Official Release

AndesCore™ AX45MP R20 Data Sheet

Document Number DS238-10
Date Issued 2023-05-05

Copyright © 2023 Andes Technology Corporation.
All rights reserved.

Copyright Notice

Copyright © 2023 Andes Technology Corporation. All rights reserved.

AndesCore™, AndeSight™, AndeShape™, AndESLive™, AndeSoft™, AndeStar™, Andes Custom Extension™, CoDense™, StackSafe™, QuickNap™, AndesClarity™, AndeSim™, AndeSysC™, AndesAIRE™, AnDLA™, NNPilot™, Andes-Embedded and Driving Innovations are trademarks owned by Andes Technology Corporation. All other trademarks used herein are the property of their respective owners.

Official Release

This document contains confidential information pertaining to Andes Technology Corporation. Use of this copyright notice is precautionary and does not imply publication or disclosure. Neither the whole nor part of the information contained herein may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Andes Technology Corporation.

The product described herein is subject to continuous development and improvement. Thus, all information herein is provided by Andes in good faith but without warranties. This document is intended only to assist the reader in the use of the product. Andes Technology Corporation shall not be liable for any loss or damage arising from the use of any information in this document, or any incorrect use of the product.

Contact Information

Should you have any problems with the information contained herein, you may contact Andes Technology Corporation through:

Email — support@andestech.com
Website — https://es.andestech.com/eservice/

Please include the following information in your inquiries:

- the document title
- the document number
- the page number(s) to which your comments apply
- a concise explanation of the problem.

General suggestions for improvements are welcome.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Revision History

Rev.	Rev. Date	Revised Content
1.0	2023-05-05	First Release for CPU Revision 20.0.0

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Contents

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

List of Figures

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

List of Tables

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1 Overview

This document describes the AndesCore AX45MP multi-core processor, and associated platform/peripheral IPs that come with the AX45MP release.

Official Release

The organization of this document is as follows: the AX45MP processor is described first, followed by descriptions regarding the associated AE350 platform in Section 23, and RISC-V specific platform IP components in Section 24, Section 25 and Section 26, followed by simulation guides in Section 29, synthesis guides in Section 30 and Section 31, and FPGA guides in Section 32.

Note

- The RISC-V spec refers to the hardware thread as “hart”. This is equivalent to “core” herein and they are interchangeably used throughout this document.

1.1 AX45MP Processor Features

The main features of the AX45MP processor are:

Cluster

- Support of 1/2/4/8 Cores
- Support of MESI cache coherence protocol with Coherence Manager
- Level-2 (L2) cache
  - Shared between cores
  - 16-way, pseudo random replacement
  - Cache size: 128KiB/256KiB/512KiB/1MiB/2MiB/4MiB/8MiB
  - Inclusive policy
  - Cache line size: 64 bytes
  - Multi-Cycle RAM support
  - ECC error protection
  - Hardware prefetch
- Core Interface
  - low-latency/synchronous/asynchronous
- Bus Interface
  - AXI4 Protocol
  - Memory Interface and Memory Mapped I/O (MMIO) Interface
  - Optional I/O Coherence Port (IOCP)
  - Configurable clock
    - Synchronous L2:bus N:1 clock ratio

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

    - Asynchronous Bus clock
  - Configurable data width: 128/256/512
  - Configurable address width: 32–64 bits
  - Optional Shared Peripheral Port (SPP)
    - 64-bit data width

Official Release

CPU Core

- 8-stage in-order dual-issue execution pipeline
- Hardware multiplier
  - radix-2/radix-4/radix-16/radix-256/fast
- Hardware divider
- Prediction
  - Dynamic branch prediction
    - 256-entry branch target buffer (BTB)
    - 768-entry branch history table
    - 8-bit global branch history
    - 4-entry return address stack (RAS)
- Machine mode, Supervisor mode and User mode
- Optional performance monitors
- Misaligned memory accesses
- RISC-V physical memory protection (PMP)
- Programmable physical memory attributes (PPMA)

AndeStar V5 ISA

- RISC-V RV64I base integer instruction set
- RISC-V “C” standard extension for compressed instructions
- RISC-V “M” standard extension for integer multiplication and division
- Optional RISC-V “P” extension for DSP/SIMD instructions
- RISC-V “A” standard extension for atomic instructions
- Optional RISC-V Bit-Manipulation ISA Extension
- Optional RISC-V “N” standard extension for user-level interrupt and exception handling
- Optional RISC-V “F” and “D” standard extensions for single/double-precision floating-point
  - FP16 half-precision floating-point extension
  - Andes BFLOAT16 Extension
- Andes Performance extension
- Andes CoDense extension

Andes Custom Extension

- Simple and flexible interface for user defined instructions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- ACE description file and the concise RTL design file in Verilog form
- COPILOT (Custom-OPtimized Instruction deveLOpment Tools) can generate extended components
  - Development tools
  - Instruction set simulator
  - AndesCore RTL

Official Release

Memory Management Unit

- sv39/sv48/bare
- 4/8-entry fully associative ITLB/DTLB
- 32/64/128/256/512-entry 4-way set-associative shared TLB
- ECC error protection

Memory Subsystem

- Support for non-blocking memory operations
- I & D-Caches
  - I-Cache is virtually indexed and physically tagged
  - D-Cache is physically indexed and physically tagged
  - Cache size: 8KiB/16KiB/32KiB/64KiB
  - Cache line size: 64 bytes
  - Set associativity: Direct-mapped/2-way/4-way
  - Custom cache control operation through CSR read/write
  - D-Cache prefetch support
  - D-Cache write-Around support
- I & D local memories
  - Size: 4KiB to 16MiB
  - Interface: RAM
- Memory subsystem soft-error protection
  - Protection scheme: parity-checking or error-checking-and-correction (ECC)
  - Automatic hardware error correction
  - Protected memories:
    - I-Cache tag RAM and data RAM
    - D-Cache tag RAM and data RAM
    - I & D local memories
    - Level-2 cache tag RAM and data RAM

Power Management

- Wait-for-interrupt (WFI) mode
- Power Domains

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Debug

- RISC-V External Debug Support
- Configurable number of breakpoints: 2/4/8
- JTAG debug port for private debug transport module
  - JTAG: IEEE Std 1149.1 style 4-wire JTAG interface
- Optional secure debug

Official Release

Trace

- Optional instruction trace

AndeStar Extension

- StackSafe hardware stack protection extension
- PowerBrake simple power/performance scaling extension
- Custom performance counter events

Platform-Level Interrupt Controller (PLIC)

- Configurable number of interrupts: 1–1023
- Configurable number of interrupt priorities: 3/7/15/31/63/127/255
- Configurable number of targets: 1–16
- Andes Vectored Interrupt extension

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1.2 Block Diagram

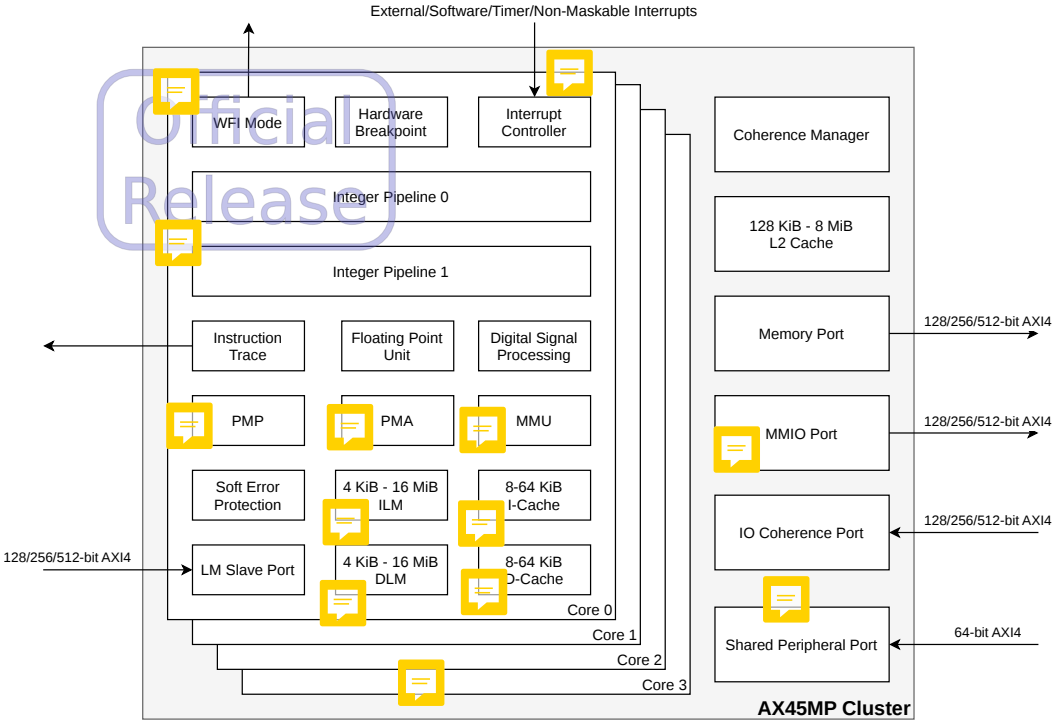


Figure 1: AX45MP Block Diagram

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1.3 Major Components

The following list describes the major components of the ax45mp_core design (the inner core of the AX45MP multi-core processor):

Official Release

ACE	Andes Custom Extension
ALU	Arithmetic Logic Unit
BIU	Bus Interface Unit
BPU	Branch Prediction Unit
CSR	Control and Status Register
DCU	Data Cache Unit
DLM	Data Local Memory Controller
DSP	Digital Signal Processing
FASTMUL	Fast Multiplier
FPU	Floating Point Unit
ICU	Instruction Cache Unit
IFU	Instruction Fetch Unit
ILM	Instruction Local Memory Controller
IPIPE	Integer Pipeline
LSU	Load Store Unit
MDU	Multiplication and Division Unit
MMU	Memory Manager Unit
ITLB	Instruction Translation Lookaside Buffer
DTLB	Data Translation Lookaside Buffer
PMA	Programmable Physical Memory Attributes Unit
PMP	Physical Memory Protection Unit
RF	Integer Register File
TRIGM	Trigger Module
L2C	Level-2 (L2) Cache
CM	Coherency Manager
IOCP	I/O Coherency Port

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1.4 Design Hierarchy

The top of the AX45MP processor design is ax45mp_cluster. The ax45mp_cluster (Figure 2) instantiates the configured number of ax45mp_core_top and the ax45mp_l2_top. The ax45mp_l2_top integrates level-2 (L2) modules, including L2 interconnect, coherence manager (CM), L2-Cache controller, L2-Cache RAMs, Memory, MMIO and IOCP interfaces.

Official Release

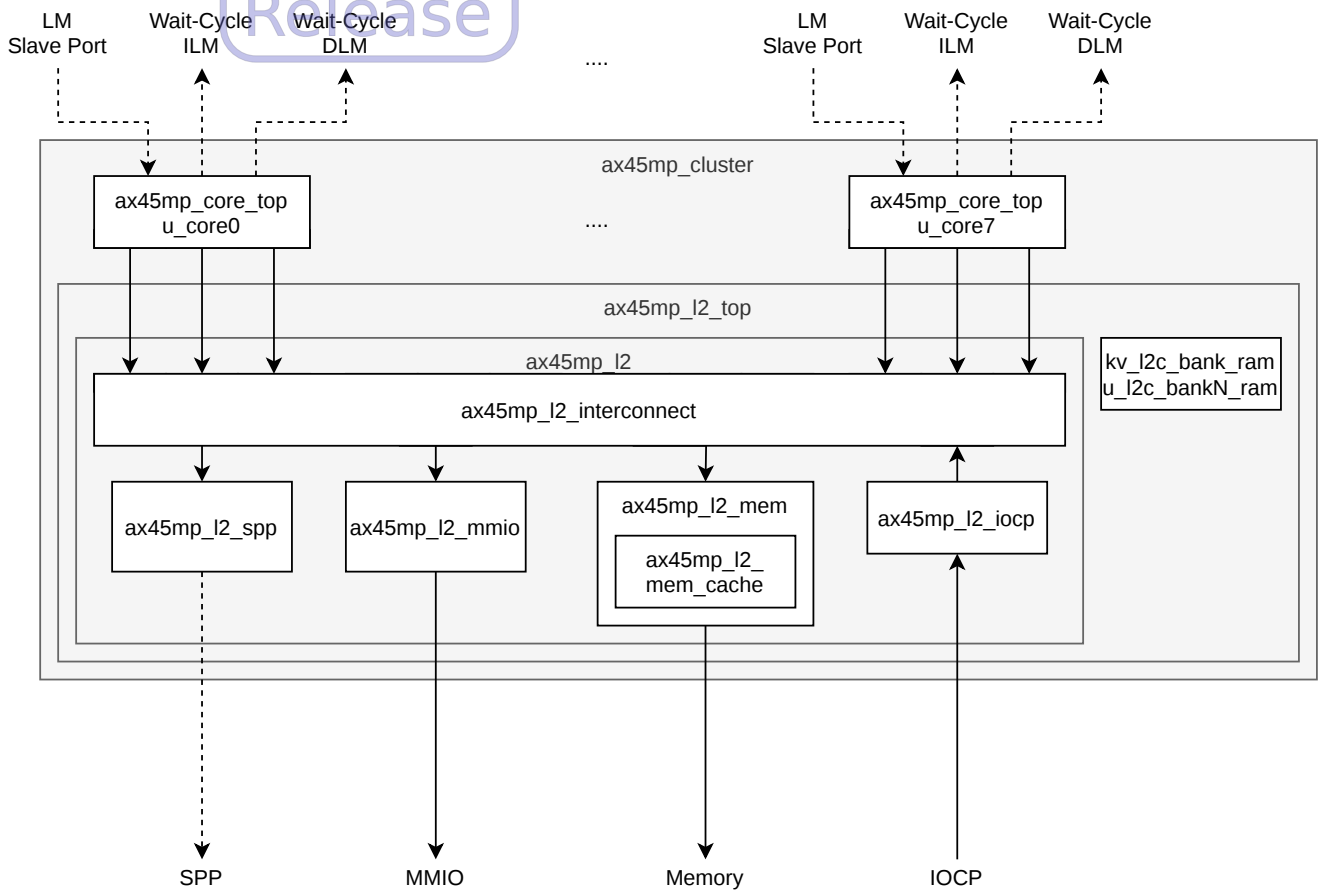


Figure 2: AX45MP Cluster Design Hierarchy

The ax45mp_core_top design instantiates one (ax45mp_core) along with all required SRAM cells to simplify the integration effort for the multi-core configuration.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

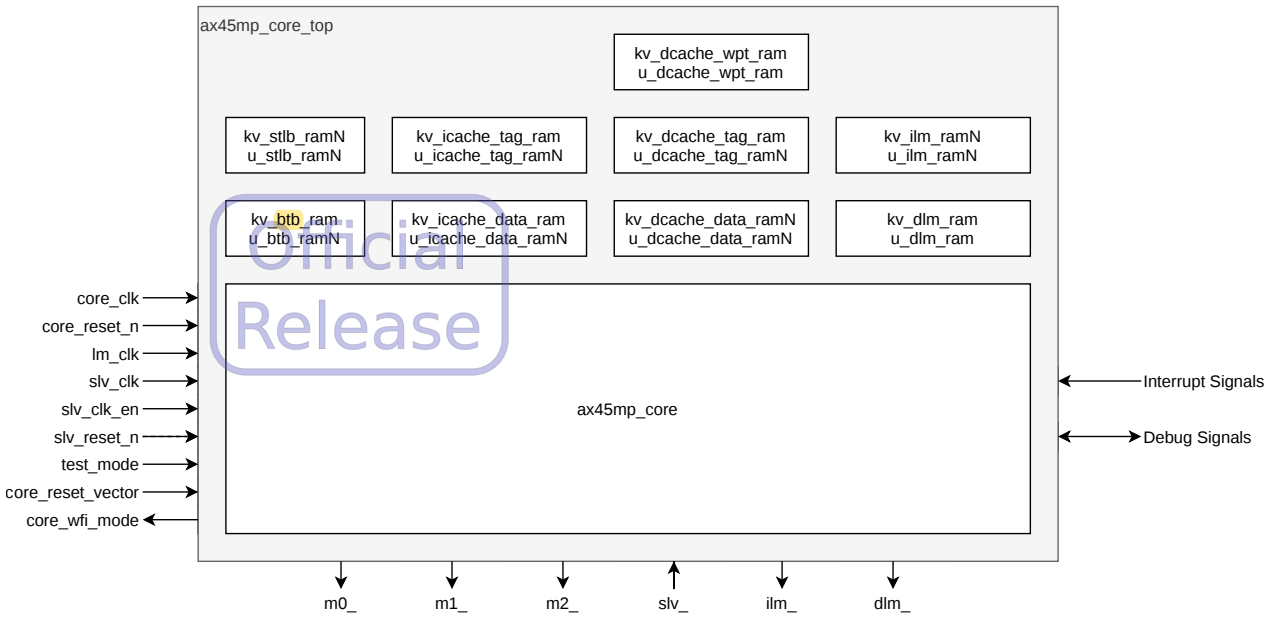


Figure 3: AX45MP Core Top Design Hierarchy

The AX45MP release package comes with the reference AXI platform design, AE350 (see Section 23). AE350 design hierarchy without and with SPP configuration are illustrated in Figure 4 and Figure 5, respectively. The top-level module of this platform is ae350_chip. The ae350_chip module instantiates the AX45MP processor.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

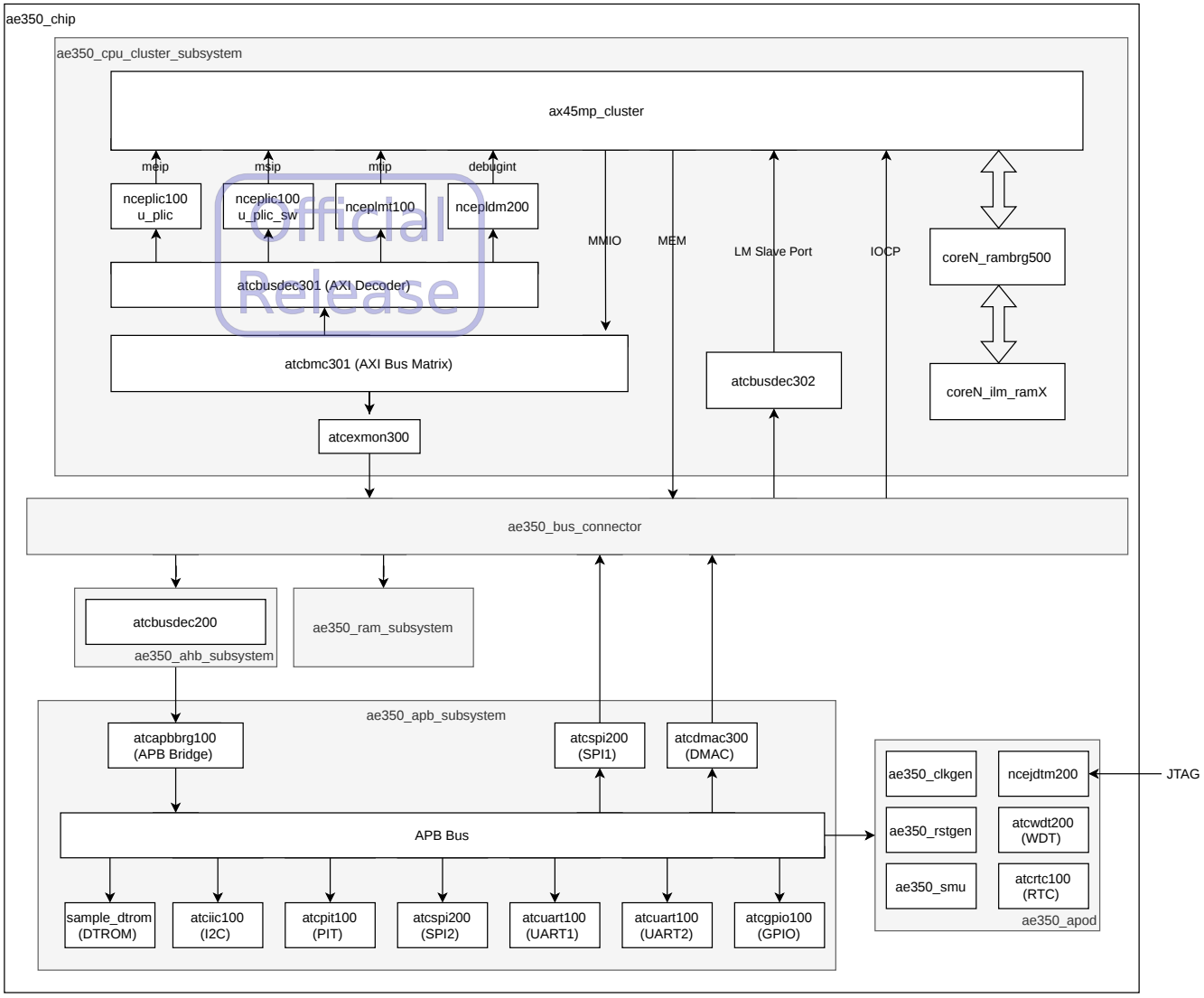


Figure 4: AX45MP Design Hierarchy Without SPP Configuration

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

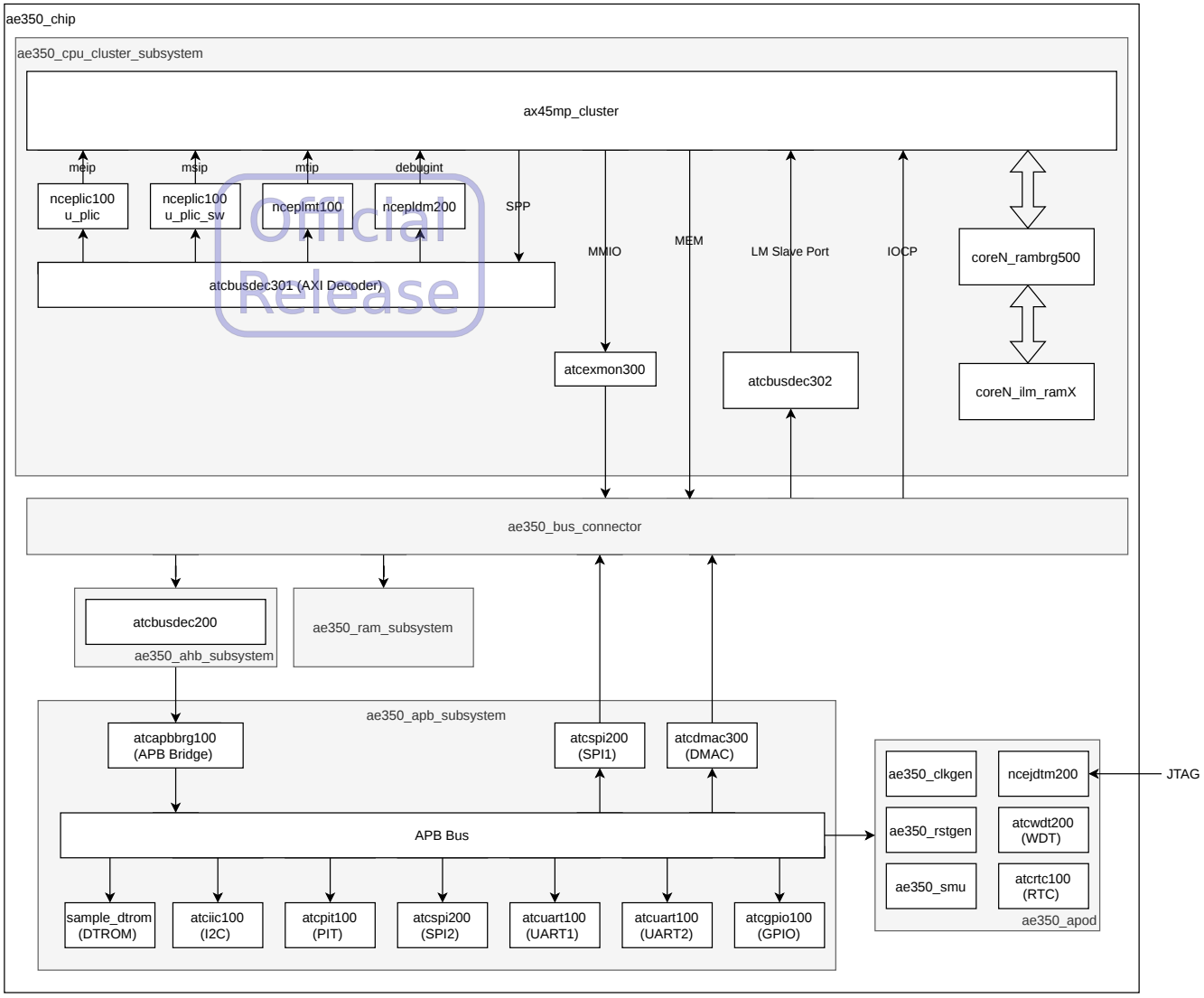


Figure 5: AX45MP Design Hierarchy With SPP Configuration

The ae350_cpu_cluster_subsystem module is a reference subsystem that instantiates ax45mp_cluster and other tightly-coupled modules such as the debug subsystem. ae350_chip and ae350_cpu_cluster_subsystem are free for modification to meet system requirements.

Note

ae350_chip and ae350_cpu_cluster_subsystem modules will be overwritten when the configuration tool is re-run. Back up local changes before re-running the configuration tool.

ncepldm200, nceplmt100, and nceplic100 modules are platform IPs as specified in the RISC-V architecture for proper operations of a RISC-V system. The ncepldm200 module implements the debug functionality. nceplmt100 implements the RISC-V machine timer, and nceplic100 implements the RISC-V platform-level interrupt controller (PLIC).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The PLIC module is instantiated twice: one (`u_plic`) for arbitrating interrupts from peripheral devices, and the other (`u_plic_sw`) for supporting software interrupts. The `u_plic_sw` instantiation only needs to use the programmability of the PLIC registers to generate (software programmable) interrupts, so all its interrupt sources are tied to zero.

atcbmc300, atcbusdec200, atcspi200, sample_ae350_smu, atcrtc100, atcapbbrg100, atcwdt200, atciic100, atcpit100, atcuart100, atcgpio100, and some other necessary modules are pre-integrated Andes platform/peripheral IPs that will be shipped together with the AX45MP package.

Official Release

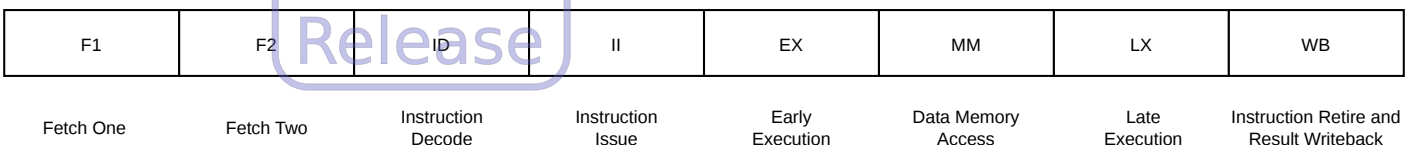
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1.5 Micro-Architecture

1.5.1 Core Pipeline Stages and Activities

AX45MP implements an 8-stage dual-issue pipeline architecture. The following figure shows the pipeline stages of the processor.

Official Release



The pipeline activities of the corresponding stages are:

F1—Instruction Fetch Stage 1

- Fetching an instruction block from ILM/Instruction cache/bus
- Dynamic branch prediction

F2—Instruction Fetch Stage 2

- Fetch block data replies from ILM/Instruction cache/bus
- Branch prediction target acquired from BPU
- Redirect fetch address to prediction target

ID—Instruction Decode

- 16/32-bit instruction alignment
- Instruction decoding

II—Instruction Issue

- Register file read
- Resolving data dependency
- Instruction issue scheduling

EX—Early Execution

- Early ALU instruction execution
- Load/Store address generation
- Division instruction execution
- Multiplication instruction execution
- DSP/SIMD instruction execution

MM—Memory Access

- DLM/D-Cache access

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Early ALU branch resolution

LX—Late Execution

- Late ALU instruction execution

WB—Instruction Retire and Result Write-Back

Official Release

- Interrupt resolution
- Instruction retire
- Register file write back
- Late ALU branch resolution
- ACE instruction issue

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1.6 Directory Structure

Unless otherwise stated, all directory paths in this document are relative to the environment variable **$NDS_HOME**, which points to the top of the distribution directory. The following table is a summary of directories in the AX45MP distribution:

Official Release

Table 1: Directory Structure

Directory	Description
$NDS_HOME/andes_ip	Design related files
$NDS_HOME/andes_ip/kv_core	AX45MP core design
$NDS_HOME/andes_ip/ae350/top/hdl	AE350 design
$NDS_HOME/andes_ip/kv_core/cluster	AX45MP design
$NDS_HOME/andes_ip/soc/ncel2c500	L2-Cache (L2C) design
$NDS_HOME/andes_ip/peripheral_ip	Peripheral IPs
$NDS_HOME/andes_ip/soc/nceplic100/hdl	PLIC design
$NDS_HOME/andes_ip/soc/nceplmt100/hdl	External machine timer design
$NDS_HOME/andes_ip/soc/ncepldm200/hdl	External debug module design
$NDS_HOME/andes_ip/soc/ncejdtm200/hdl	External JTAG debug transport module design
$NDS_HOME/testbench	Testbench related files — top-level modules
$NDS_HOME/andes_vip/models	Directory for simulator models — the external memory model, the external debug host model.
$NDS_HOME/andes_vip/patterns/samples	Sample test patterns. See Section 29.3.8.
$NDS_HOME/andes_vip/patterns/riscv-tests	RISC-V test patterns
$NDS_HOME/flists	Directory for simulation file lists
$NDS_HOME/tools/bin	Tools for simulation
$NDS_HOME/config_tools	Configuration tools

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1.7 Feature Availability

This section describes feature availability of CPU revisions. The CPU revision can be found in the `mimpid` CSR.

Official Release

Table 2: Feature Availability

Feature	CPU Revision	Description
Export Signals from Machine Timer (PLMT)	20.0.0 and later	See Section 2.15
Configurable ACE Instruction FIFO Depth	20.0.0 and later	See Section 2.4.6.1
Multiple Transactions to Non-Cacheable Memory	20.0.0 and later	See Section 2.7
Configurable L2C Register Space Size	20.0.0 and later	See Section 2.3.3
Configurable L2C Prefetch Buffer	20.0.0 and later	See Section 2.3.3
dlm_boot signal	20.0.0 and later	See Table 6
Enabling/Disabling ILM/DLM at run time	18.0.0 and later	See Section 21.12.1 and Section 21.12.2
Prefix for generating a uniquified design	18.0.0 and later	See Section 2.2
Synchronizer Stage	18.0.0 and later	See Section 2.6.1
Synchronizer for Asynchronous FIFO Read Data	18.0.0 and later	See Section 2.6.2
PMP Granularity	18.0.0 and later	See Section 2.5.4
Shared Peripheral Port (SPP)	18.0.0 and later	See Section 2.7
RISC-V Bit-Manipulation ISA Extension	18.0.0 and later	See Section 2.4.3
512-bit bus data-width	18.0.0 and later	See Section 2.7.
8-core	13.0.0 and later	See Section 2.3.1
Andes Custom Extension (ACE)	13.0.0 and later	See Section 2.4.6
Local memory local interrupt output	13.0.0 and later	Add the output, lm_local_int, for core clock gating control
Shared TLB soft error protection	13.0.0 and later	See Section 2.5.2
ILM wait cycle	13.0.0 and later	See Section 3.8.2
`mxstatus.IME` and `mxstatus.PIME`	13.0.0 and later	See Section 21.3.12

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 2: (continued)

Feature	CPU Revision	Description
l2c_clkgen_resetn	12.2.0 and later	See Table 6
4MiB/8MiB L2-Cache	12.0.0 and later	See Section 2.3.3
Configurable L2-Cache RAM setup/output cycle	12.0.0 and later	See Section 2.3.3
Asynchronous bus clock	12.0.0 and later	Support asynchronous bus clock
32-entry PMP	12.0.0 and later	Support 32-entry PMP
IOCP multiple outstanding	12.0.0 and later	See Section 2.3.4
RAM Control Signals	12.0.0 and later	See Section 2.13
PC/GPR/CSR Probing Support	12.0.0 and later	See Section 3.16
2/4-bank DLM	12.0.0 and later	See Section 8.5
DLM wait cycle interface	12.0.0 and later	See Section 2.9.2
Power Domains	8.0.0 and later	See Section 20.6.
256-bit bus data-width	8.0.0 and later	See Section 2.7.
Crash debugging	8.0.0 and later	See Section 26.8.
Ratified RISC-V processor trace interface	8.0.0 and later	See Section 3.14.
`micm_cfg.IC_REPL`	8.0.0 and later	See Section 21.5.1.
`mdcm_cfg.DC_REPL`	8.0.0 and later	See Section 21.5.2.
`ml2c_ctl_base`	8.0.0 and later	See Section 21.5.5.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2 Processor Configuration Options

The AX45MP processor is configured through the bundled configuration tool.

Official Release

2.1 Configuration Tool

The configuration tool allows interactive selection of configurable options. The tool is Tcl/Tk based and it requires the Tk interpreter `wish` to exist in the search path. To start the tool, make sure that the **$DISPLAY** environment variable is set correctly to point to a valid X server, and then type the following command to launch the configuration tool.

$NDS_HOME/config_tools/nds-softcore-config

Figure 6 shows a screenshot of the tool. The "Save" button saves your options so that it could later be loaded into the tool. The "Generate ax45mp_core" button configures the AX45MP processor with the selected options. By clicking this button, the following files are generated and overwritten. If required, back up the files before re-configuring the processor.

- **$NDS_HOME**/andes_ip/kv_core/top/hdl/ax45mp_cluster.v
  - Top module of the AX45MP processor with the selected options.
- **$NDS_HOME**/andes_ip/kv_core/top/hdl/ax45mp_core_top.v
  - Top module of the AX45MP core with the selected options.
- **$NDS_HOME**/andes_ip/kv_core/top/hdl/kv_core.v
  - Design of the AX45MP core.
- **$NDS_HOME**/andes_ip/kv_core/top/hdl/ax45mp_l2_top.v
  - Top module of the AX45MP level-2 designs with the selected options.
- **$NDS_HOME**/andes_ip/kv_core/top/hdl/ae350_cpu_cluster_subsystem.v
  - Sample CPU subsystem that instantiates the AX45MP processor. This subsystem is for the AE350 platform.
- **$NDS_HOME**/andes_ip/kv_core/top/hdl/config.inc
  - Required by the accompanying testbench.
- **$NDS_HOME**/andes_ip/ae350/top/hdl/include/ae350_config.vh
  - Configurations for the platform, required by the AE350 CPU subsystem and chip.
- **$NDS_HOME**/andes_ip/kv_core/top/hdl/ax45mp_cluster.xml

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

– IP-XACT XML for the AX45MP processor.

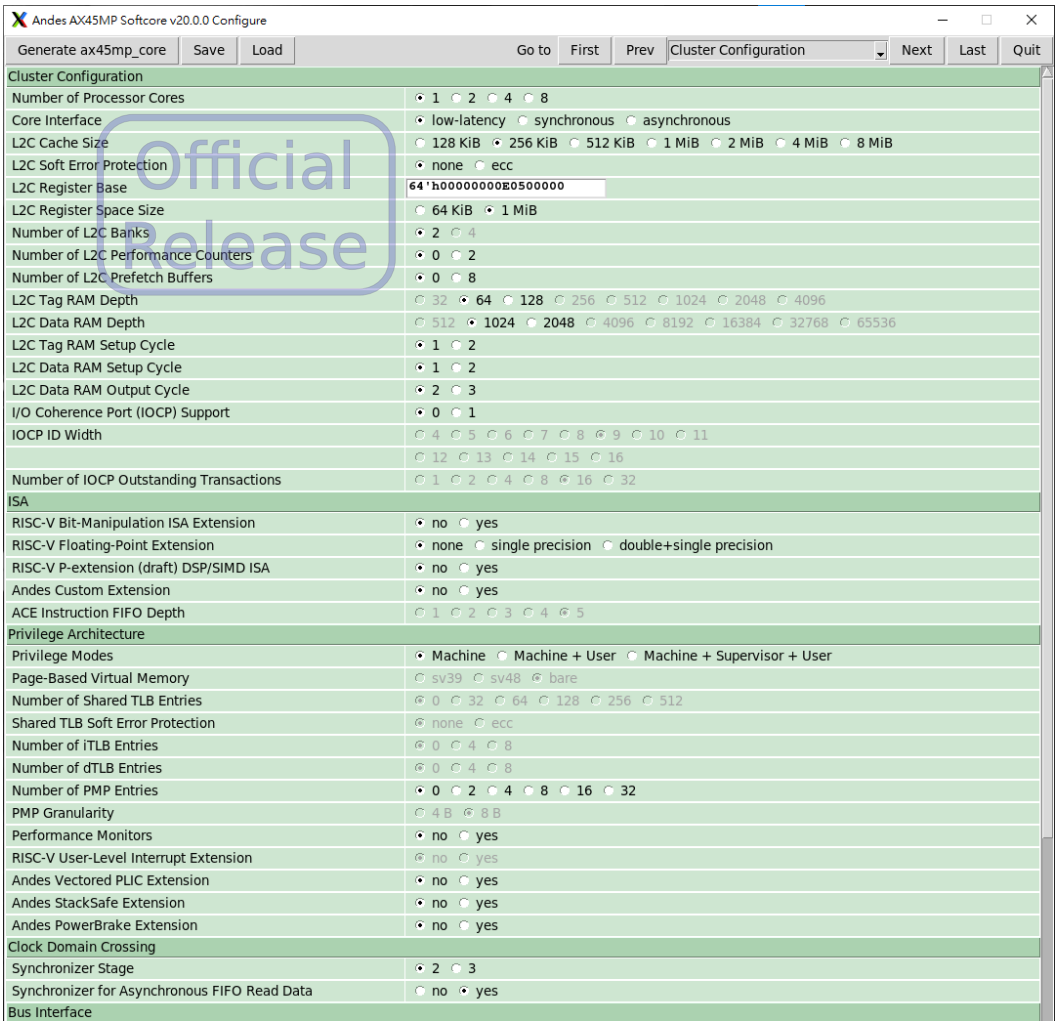


Figure 6: `nds-softcore-config` Screenshot

Note

The screenshot only serves to show how the tool *looks like* to facilitate the description above. It is not a goal for this figure to show the most updated version of the tool. The actual content of the tool may differ slightly as the tool gets updated with new features added to AX45MP. Please see Table 3 for the most updated list of all configurable options.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.2 Summary of Configuration

Table 3: AX45MP Configuration Options

Option	Valid Values	Description
Cluster Configuration		
Number of Processor Cores	1/2/4/8	Section 2.3.1
Core Interface	low-latency/synchronous/ asynchronous	Section 2.3.2
L2C Cache Size	128 KiB/256 KiB/512 KiB/ 1 MiB/2 MiB/4 MiB/8 MiB	Section 2.3.3
Number of L2C Banks	2/4	Section 2.3.3
L2C Soft Error Protection	none/ecc	Section 2.3.3
L2C Register Base		Section 2.3.3
L2C Register Space Size	64 KiB/1 MiB	Section 2.3.3
Number of L2C Performance Counters	0/2	Section 2.3.3
L2C Tag RAM Depth	32/64/128/256/512/1024/2048/4096	Section 2.3.3
L2C Data RAM Depth	512/1024/2048/4096/8192/ 16384/32768/65536	Section 2.3.3
L2C Tag RAM Setup Cycle	1/2	Section 2.3.3
L2C Data RAM Setup Cycle	1/2	Section 2.3.3
L2C Data RAM Output Cycle	2/3	Section 2.3.3
L2C Prefetch Buffer	yes/no	Section 2.3.3
I/O Coherence Port (IOCP) Support	0/1	Section 2.3.4
IOCP ID Width	4/5/6/7/8/9/10/11/12/13/14/15/16	Section 2.3.4
Number of IOCP Outstanding Transactions	1/2/4/8/16/32	Section 2.3.4
ISA		
RISC-V Bit-Manipulation ISA Extension	yes/no	Section 2.4.3
RISC-V Floating-Point Extension	double+single precision / single precision / none	Section 2.4.4
RISC-V P-extension (draft) DSP/SIMD ISA	yes/no	Section 2.4.5
Andes Custom Extension	yes/no	Section 2.4.6
ACE Instruction FIFO Depth	1/2/3/4/5	Section 2.4.6.1
Privilege Architecture		

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 3: (continued)

Option	Valid Values	Description
Privilege Modes	Machine/Machine + User/ Machine + Supervisor + User	Section 2.5.1
Page-Based Virtual Memory	sv39/sv48/bare	Section 2.5.2
Number of Shared TLB Entries	0/32/64/128/256/512	Section 2.5.2
Shared TLB Soft Error Protection	none/ecc	Section 2.5.2
Number of *i*TLB Entries	0/4/8	Section 2.5.2
Number of *d*TLB Entries	0/4/8	Section 2.5.2
Number of PMP Entries	0/2/4/8/16/32	Section 2.5.3
PMP Granularity	4 B/8 B	Section 2.5.4
Performance Monitors	yes/no	Section 2.5.6
RISC-V User-Level Interrupt Extension	yes/no	Section 2.4.1
Andes Vectored PLIC Extension	yes/no	Section 2.5.7
Andes StackSafe Extension	yes/no	Section 2.5.8
Andes PowerBrake Extension	yes/no	Section 2.5.9
Clock Domain Crossing		
Synchronizer Stage	2/3	Section 2.6.1
Synchronizer for Asynchronous FIFO Read Data	yes/no	Section 2.6.2
Bus Interface		
Bus Clock	asynchronous/synchronous	Section 2.7
Bus Data Width	128/256/512	Section 2.7
Bus Address Width	32-64	Section 2.7
Number of Transactions to Non-Cacheable Memory	1/4/8/16/32	Section 2.7
Size of Shared Peripheral Region	0 KiB/1 KiB/2 KiB/4 KiB/8 KiB/16 KiB/32 KiB/ 64 KiB/128 KiB/256 KiB/512 KiB/ 1 MiB/2 MiB/4 MiB/8 MiB/16 MiB/ 32 MiB/64 MiB/128 MiB/256 MiB/ 512 MiB/1 GiB/2 GiB	Section 2.7
Base address of Shared Peripheral Region		Section 2.7
Micro-Architecture		
Multiplier Implementation	radix2/radix4/ radix16/radix256/fast	Section 2.8.1

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 3: (continued)

Option	Valid Values	Description
Local Memory		
ILM Wait Cycle	yes/no	Section 2.9.1
ILM Size	0 KiB/4 KiB/8 KiB/16 KiB/32 KiB/ 64 KiB/128 KiB/256 KiB/512 KiB/ 1 MiB/2 MiB/4 MiB/8 MiB/16 MiB	Section 2.9.1
ILM Base		Section 2.9.1
ILM Soft Error Protection	none/ecc	Section 2.9.1
Number of DLM Banks	1/2/4	Section 2.9.2
DLM Size	0 KiB/4 KiB/8 KiB/16 KiB/32 KiB/ 64 KiB/128 KiB/256 KiB/512 KiB/ 1 MiB/2 MiB/4 MiB/8 MiB/16 MiB	Section 2.9.2
DLM Base		Section 2.9.2
DLM Soft Error Protection	none/ecc	Section 2.9.2
DLM Wait Cycle	yes/no	Section 2.9.2
Slave Port Support	yes/no	Section 2.9.3
Slave Port ID Width	4/5/6/7/8/9/10/11/12/13/14/15/16	Section 2.9.3
Cache Configuration		
I-Cache Size	8 KiB/16 KiB/32 KiB/64 KiB	Section 2.10.1
I-Cache Associativity	direct-mapped/2-way/4-way	Section 2.10.1
I-Cache Replacement Policy	random/pseudo-lru	Section 2.10.1
I-Cache Soft Error Protection	none/parity/ecc	Section 2.10.1
D-Cache Size	8 KiB/16 KiB/32 KiB/64 KiB	Section 2.10.2
D-Cache Associativity	direct-mapped/2-way/4-way	Section 2.10.2
D-Cache Replacement Policy	random/pseudo-lru	Section 2.10.2
D-Cache Soft Error Protection	none/ecc	Section 2.10.2
D-Cache Outstanding Misses	3/4/5/6/7/8	Section 2.10.2
D-Cache Prefetch Support	yes/no	Section 2.10.2
D-Cache Write-Around Support	yes/no	Section 2.10.2
Debug and Trace		
Debug Support	yes/no	Section 2.11.1
Debug Vector Address		Section 2.11.2
Number of Triggers	2/4/8	Section 2.11.3
Trace Interface	none/instruction-ratified/instruction-gen1	Section 2.11.4
Andes PC/GPR/CSR Probing Support	yes/no	Section 2.11.5

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 3: (continued)

Option	Valid Values	Description
Physical Memory Attributes		
Number of Programmable PMA Entries	0/1/2/3/4/5/6/7/8/9/ 10/11/12/13/14/15/16	Section 2.5.5
Device Region*N* Base		Section 2.12
Device Region*N* Mask		Section 2.12
Integration Settings		
RAM Control In	yes/no	Section 2.13
BTB RAM Control In Width	1 – 128	Section 2.13
ILM RAM Control In Width	1 – 128	Section 2.13
DLM RAM Control In Width	1 – 128	Section 2.13
I-Cache Tag RAM Control In Width	1 – 128	Section 2.13
I-Cache Data RAM Control In Width	1 – 128	Section 2.13
D-Cache Tag RAM Control In Width	1 – 128	Section 2.13
D-Cache Data RAM Control In Width	1 – 128	Section 2.13
D-Cache WPT RAM Control In Width	1 – 128	Section 2.13
STLB RAM Control In Width	1 – 128	Section 2.13
STLB Tag RAM Control In Width	1 – 128	Section 2.13
STLB Data RAM Control In Width	1 – 128	Section 2.13
L2C Tag RAM Control In Width	1 – 128	Section 2.13
L2C Data RAM Control In Width	1 – 128	Section 2.13
RAM Control Out	yes/no	Section 2.13
BTB RAM Control Out Width	1 – 128	Section 2.13
ILM RAM Control Out Width	1 – 128	Section 2.13
DLM RAM Control Out Width	1 – 128	Section 2.13
I-Cache Tag RAM Control Out Width	1 – 128	Section 2.13
I-Cache Data RAM Control Out Width	1 – 128	Section 2.13
D-Cache Tag RAM Control Out Width	1 – 128	Section 2.13
D-Cache Data RAM Control Out Width	1 – 128	Section 2.13
D-Cache WPT RAM Control Out Width	1 – 128	Section 2.13
STLB RAM Control Out Width	1 – 128	Section 2.13
STLB Tag RAM Control Out Width	1 – 128	Section 2.13
STLB Data RAM Control Out Width	1 – 128	Section 2.13
L2C Tag RAM Control Out Width	1 – 128	Section 2.13

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 3: (continued)

Option	Valid Values	Description
L2C Data RAM Control Out Width	1 – 128	Section 2.13
Miscellaneous		
Prefix for Generating a Uniquified Design		Section 2.14
Export Signals from Machine Timer (PLMT)	yes/no	Section 2.15
Platform Level Settings		
Platform and CPU Subsystem	ae350/custom	Section 2.16
Reset Vector Address		Section 23.2
CPU Subsystem Bus Matrix		
Interrupt Controller (PLIC) Base Address		Section 23.2
Machine Timer (PLMT) Base Address		Section 23.2
Software Interrupts (PLIC_SWINT)	yes/no	Section 2.17
PLIC_SWINT Base Address		Section 23.2
PLDM System Bus Access	yes/no	Section 2.18.1
Secure Debug	yes/no	Section 2.18.2
Debug Module (PLDM) Base Address		Section 23.2
Device in AXI Space		
AXI Bus Matrix Program Register Base Address		Section 23.2
RAM Bridge Region Size (MiB)		Section 23.2
RAM Bridge Base Address		Section 23.2
Slave Port Region Size (MiB)		Section 23.2
ILM Slave Port Base Address		Section 23.2
Slave Port DLM Selection Bit		Section 9
DLM Slave Port Base Address		Section 23.2
Dynamic Frequency Scaling Program Register Base Address		Section 23.2
Devices in AHB Space		
AHB Decoder Program Register Base Address		Section 23.2
Devices in APB Space		
UART1	yes/no	Section 2.19.1
UART2	yes/no	Section 2.19.1
Programmable Interval Timer	yes/no	Section 2.19.2

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 3: (continued)

Option	Valid Values	Description
Watchdog Timer	yes/no	Section 2.19.3
Real-Time Clock	yes/no	Section 2.19.4
GPIO	yes/no	Section 2.19.5
I2C	yes/no	Section 2.19.6
SPI1	yes/no	Section 2.19.7
DMA Controller	yes/no	Section 2.19.8
SPI2	yes/no	Section 2.19.7
DTROM	yes/no	Section 2.19.9
APB Bridge Program Register Base Address		Section 23.2
SMU Program Register Base Address		Section 23.2
UART1 Program Register Base Address		Section 23.2
UART2 Program Register Base Address		Section 23.2
Programmable Interval Timer Base Address		Section 23.2
Watchdog Timer Base Address		Section 23.2
Real-Time Clock Base Address		Section 23.2
GPIO Base Address		Section 23.2
I2C Base Address		Section 23.2
SPI1 Base Address		Section 23.2
SPI MEM Base Address		Section 23.2
DMA Controller Base Address		Section 23.2
SPI2 Base Address		Section 23.2
DTROM Base Address		Section 23.2

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.3 Cluster Configuration

2.3.1 Number of Processor Cores

This option selects the number of cores.

Official Release

Note

The availability of this "Number of Processor Cores" option depends on AX45MP licensing agreements.

2.3.2 Core Interface

This option selects the interface of cores.

- low-latency: Core interface signals are directly connected to L2 without additional buffers.
- synchronous: Core interface signals are connected to L2 with synchronous buffers, and one extra latency is introduced.
- asynchronous: Core interface signals are connected to L2 with asynchronous FIFOs. The core clock is asynchronous to L2 clock and bus clock.

2.3.3 L2-Cache (L2C) Controller

The **L2C Cache Size** option selects the size of the L2-Cache memory in KiB. Only power-of-2 sizes are supported.

The **Number of L2C Banks** option specifies the bank number of L2C.

- 2: two-bank L2C
- 4: four-bank L2C, this option is available when the **Number of Processor Cores** is greater than 4.

The **L2C Soft Error Protection** option decides whether L2-Cache RAMs are protected by ECC or not.

L2C supports built-in hardware performance counters for debugging or performance analysis purposes. The number of counters is decided by the **Number of L2C Performance Counters** option.

The base address of this space is decided by the **L2C Register Base** option. The space size is decided by the **L2C Register Space Size** option.

The following options provide trade-off between frequency, performance and area:

- **L2C Tag RAM Depth**: the depth of each L2-Cache Tag RAM instance. Greater depth has higher area density but lower frequency.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- **L2C Data RAM Depth**: the depth of each L2-Cache Data RAM instance. Greater depth has higher area density but lower frequency.
- **L2C Tag RAM Setup Cycle**: see Table 73 and Section 28.2.
- **L2C Data RAM Setup Cycle**: see Table 73 and Section 28.2.
- **L2C Data RAM Output Cycle**: see Table 73 and Section 28.2.

Official Release

Specifying **L2C Prefetch Buffer** to "yes" enables L2-Cache prefetch buffer. The prefetch buffer provides support for prefetching cache lines. It prefetches one additional cache line after a cache-miss is detected. See PB field and PBEN field for more information.

2.3.4 I/O Coherence Port (IOCP)

This option selects the number of I/O coherent ports. The IOCP provides an AXI interface for connecting external non-caching masters, such as DMA controller. The accesses from IOCP are coherent with D-Caches and L2-Cache.

The IOCP ID width is denoted as IOCP_ID_WIDTH. The IOCP_ID_WIDTH should be equal to ID width of the connected master interface.

The **Number of IOCP Outstanding Transactions** option decides the maximum outstanding by IOCP.

See Section 17 for more information.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.4 ISA

2.4.1 RISC-V User-Level Interrupt Extension

Specifying this option to "yes" enables the RISC-V User-Level Interrupt Extension (RVN). The option is available only when U-Mode is configured.

Official Release

2.4.2 RISC-V Atomic Instruction Extension

RISC-V "A" Standard Extension for Atomic Instructions (RVA) includes instructions that atomically read-modify-write memories for synchronization between multiple hardware threads (harts).

2.4.3 RISC-V Bit-Manipulation ISA Extension

Specifying this option to "yes" enables the RISC-V Bit-Manipulation ISA extensions, including Zba, Zbb, Zbc, and Zbs.

2.4.4 RISC-V Floating-Point Instruction Extension

This option determines the floating-point extension type.

- none: no floating-point extension
- single precision: "F" standard extension for single-precision floating-point instruction
- double+single precision: "F" and "D" standard extensions for single- and double-precision floating-point instructions

Note

- The availability of the floating-point extension depends on AX45MP licensing agreements.
- FP16 and BFLOAT16 Extension is also included when floating-point extension is configured in.

2.4.5 RISC-V P-Extension (Draft) DSP/SIMD ISA

Specifying this option to "yes" enables the RISC-V "P" Extension for DSP (Digital Signal Processing) instructions. DSP instructions include all the instructions of AndeStar DSP ISA Extension.

2.4.6 Andes Custom Extension

Specifying this option to "yes" enables the custom instruction extension.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The Andes Custom Extension (ACE) feature provides a simple and flexible interface to extend new instructions. Please see *Andes Custom Extension Specification* and *Andes Custom Extension User Manual* for more information.

Note

The ACE feature requires additional licenses. Please contact Andes Technology for further information.

Official Release

2.4.6.1 ACE Instruction FIFO Depth

This option determines instruction FIFO depth for ACE. It can be 1, 2, 3, 4, or 5.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.5 Privilege Architecture

2.5.1 Privilege Modes

This option determines the number of supported privileges levels. The AX45MP processor supports 1, 2 or 3 privilege levels, as shown in Table 4.

Official Release

Table 4: Supported Combinations of Privilege Modes

Number of Levels	Supported Modes	Intended Usage
1	Machine	Simple embedded systems
2	Machine, User	Secure embedded systems
3	Machine, Supervisor, User	Systems running Unix-like operating systems

The RISC-V privilege architecture specifies four privilege levels as shown in Table 5. Machine mode (M-mode) has the highest privileges and is the mandatory privilege level for a RISC-V hardware platform. User mode (U-mode) restricts privileges to protect against incorrect or malicious application codes. And Supervisor mode (S-mode) is provided for Unix-like operating systems with address translating and protection requirements.

Table 5: RISC-V Privilege Levels

Level	Encoding	Name	Abbreviation
0	00	User/Application	U
1	01	Supervisor	S
2	10	Reserved	
3	11	Machine	M

2.5.2 Page-Based Virtual Memory

Specifying this option to “sv39” or “sv48” decides the capability of the page-based virtual address translation scheme. This option is available only when **Privilege Modes** is “Machine + Supervisor + User”.

- sv39: Supervisor mode and Sv39 virtual address translation scheme
- sv48: Supervisor mode and Sv48/Sv39 virtual address translation scheme
- bare: No virtual address translation scheme

The TLB can be further configured with the following options:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- **Number of Shared TLB Entries** selects the number of supported Shared TLB entries.
- **Shared TLB Soft Error Protection** option decides whether Shared TLB RAMs are protected by ECC or not.
- **Number of *i*TLB Entries** selects the number of supported *i*TLB entries.
- **Number of *d*TLB Entries** selects the number of supported *d*TLB entries.

Official Release

2.5.3 Number of Physical Memory Protection Entries

This option selects the number of supported PMP entries. This feature is disabled if the number is set to 0.

2.5.4 Physical Memory Protection Granularity

This option selects the size of PMP granularity.

2.5.5 Number of Programmable Physical Memory Attribute Entries

This option selects the number of supported Programmable PMA (PPMA) entries. This feature is disabled if the number is set to 0.

2.5.6 Performance Monitors

Specifying this option to "yes" enables hardware performance monitors, including `mcycle` and `minstret` Control and Status Registers (CSR). The `mcycle` register counts the elapsed clock cycles, while the `minstret` register counts the number of retired instructions.

2.5.7 Andes Vectored PLIC Extension

Specifying this option to "yes" enables the Andes Vectored PLIC extension for reducing interrupt latency. See Section 24.3 for more information.

2.5.8 Andes StackSafe Extension

Specifying this option to "yes" enables the StackSafe hardware stack protection extension.

The extension is a hardware mechanism for tracking and guarding against the stack pointer overflows/underflows. See Section 21.13.1 for more information.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.5.9 Andes PowerBrake Extension

Specifying this option to “yes” enables the PowerBrake power/performance scaling extension.

The PowerBrake extension throttles performance by reducing instruction executing rate instead of slowing down clock frequency. The performance and hence power consumption can be switched to a different level in a couple of instructions. This is an ultra-low latency mechanism for performance & power scaling, as compared to the latencies of frequency scaling through PLL programming.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.6 Clock Domain Crossing

2.6.1 Synchronizer Stage

This option specifies the level N ($N = 2$ or 3) of the CDC synchronizer.

Official Release

The CDC synchronizer is designed to avoid metastability caused clock domain crossing. Increasing the level of the CDC synchronizers may reduce the risk of metastability.

All the signals on the core and the platform that cross clock domains are affected by this option.

2.6.2 Synchronizer for Asynchronous FIFO Read Data

Specifying this option to “yes” adds in a synchronizer to read data of all asynchronous FIFOs.

The synchronizer reduces the number of CDC paths from the write clock to the read clock, but the latency is increased by one read clock cycle.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.7 Bus Interface

The bus interface unit (BIU) is responsible for system bus accesses of AX45MP. The **Bus Data Width** option determines the bus data width (`BIU_DATA_WIDTH`). AX45MP supports 128/256/512 bus data width.

Official Release

The width of the bus address (`BIU_ADDR_WIDTH`) could be:

- With Sv48 page-based virtual address translation: any width between 32 and 47 bits
- With Sv39 page-based virtual address translation: any width between 32 and 38 bits

The **Bus clock** option determines whether the clock domain of core is the same as the clock domain of bus.

- asynchronous: clock domain of core is asynchronous to bus and slave port interface.
- synchronous: clock domain of core is synchronous to bus and slave port interface.

The **Number of Transactions to Non-Cacheable Memory** option specify the maximum number of transactions to Non-Cacheable Memory. See Section 15.

Size of Shared Peripheral Region and **Base address of Shared Peripheral Region** respectively specify the size and base address of shared peripheral region. The base address must be aligned with the size of shared peripheral region. The PMA of shared peripheral port must be in device region. Please see Section 13 for PMA. The shared peripheral port has one more cycle latency than the MMIO interface. The data-width of shared peripheral port is fixed at 64 bits wide. Please see Section 15.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.8 Micro Architecture

2.8.1 Multiplier Implementation

This option selects the implementation of the hardware multiplier in AX45MP. Valid values and the respective performance are:

Official Release

- radix2: 1-bit/cycle
- radix4: 2-bit/cycle
- radix16: 4-bit/cycle
- radix256: 8-bit/cycle
- fast: two-stage pipelined.

Radix multiplier types realize the multiplier through serial additions, while the fast multiplier type implements a two-stage pipelined parallel multiplier.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.9 Local Memory

2.9.1 Instruction Local Memory (ILM)

The **ILM Size** option (ILM_SIZE_KB) selects the size of the instruction local memory in KiB. Only power-of-2 sizes are supported. Specifying 0 to this option unconfigures the instruction local memory.

Official Release

The **ILM Base** option (ILM_BASE) specifies the base address of the instruction local memory, and it must be aligned to the size of the instruction local memory.

The **ILM Soft Error Protection** option selects the soft-error protection scheme for the instruction local memory:

- none: no protection
- ecc: single-error correction, and double-error detection

The **ILM Wait Cycle** option specifies the interface for ILM accesses.

- yes: RAMBRG500 interface is used for ILM accesses to support ILM access with wait cycle. Please see Section 3.8.2 for details.
- no: RAM interface is used for ILM accesses

2.9.2 Data Local Memory (DLM)

The **DLM Size** option (DLM_SIZE_KB) selects the size of the data local memory in KiB. Only power-of-2 sizes are supported. Specifying 0 to this option unconfigures the data local memory.

The **DLM Base** option (DLM_BASE) specifies the base address of the data local memory, and it must be aligned to the size of the data local memory.

The **DLM Soft Error Protection** option selects the soft-error protection scheme for the data local memory:

- none: no protection
- ecc: single-error correction, and double-error detection

The **Number of DLM Banks** option specifies the bank number of DLM.

- 1: single-bank DLM
- 2: two-bank DLM
- 4: four-bank DLM

Please see Section 8.5 for details.

The **DLM Wait Cycle** option specifies the interface for DLM accesses.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- yes: DLM wait cycle interface is used for DLM accesses to support DLM access with wait cycles. Please see Section 3.9.2 for details.
- no: RAM interface is used for DLM accesses

2.9.3 Slave Port

Official Release

Specifying **Slave Port Support** to "yes" enables the local memory slave port for bus masters to access the local memories of AX45MP. The interface protocol for the local memory slave port is AXI, and the data width is identical to bus data width. Please see Section 9 for details.

The slave port ID width is denoted as SLAVE_PORT_ID_WIDTH. For CPU revisions before 2.0.0, SLAVE_PORT_ID_WIDTH is fixed to 8. For CPU revision 2.0.0 and later, SLAVE_PORT_ID_WIDTH is specified by the **Slave Port ID Width** option. The SLAVE_PORT_ID_WIDTH should be equal to ID width of the connected master interface.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.10 Cache Configuration

2.10.1 Instruction Cache

The **I-Cache Size** option selects the size of the instruction cache in KiB. Only power-of-2 sizes are supported.

Official Release

The **I-Cache Associativity** option selects the associativity of instruction cache.

The **I-Cache Replacement Policy** option selects the replacement policy of instruction cache on cache misses.

The **I-Cache Soft Error Protection** option selects the soft-error protection scheme for the instruction cache:

- none: no protection
- parity: single-error correction. The "correction" is performed by re-fetching the clean instruction from the next-level memory.
- ecc: single-error and double error correction. The "correction" is performed by re-fetching the clean instruction from the next-level memory.

2.10.2 Data Cache

The **D-Cache Size** option selects the size of the data cache in KiB. Only power-of-2 sizes are supported.

The **D-Cache Associativity** option selects the associativity of data cache.

The **D-Cache Replacement Policy** option selects the replacement policy of data cache when cache miss.

The **D-Cache Soft Error Protection** option selects the soft-error protection scheme for the data cache:

- none: no protection
- ecc: single-error correction, and double-error detection

The **D-Cache Outstanding Misses** option selects the maximum number of outstanding D-Cache misses. There is a table of miss status handling registers (MSHR) with the configured amount of entries in D-Cache to record all outstanding bus transactions. For cacheable memory regions, all cacheable store misses to the same D-Cache cache-line will share the same entry.

The **D-Cache Write-Around** option enables the D-Cache Write-Around support. See Section 14.2 for details.

The **D-Cache Prefetch** option enables the D-Cache hardware prefetcher support. See Section 14.3 for details.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.11 Debug and Trace

2.11.1 Debug Support

Specifying this option to "yes" enables the core debug support.

Official Release

2.11.2 Debug Vector Address

The **Debug Vector Address** option specifies the entry point address of the exception handler for servicing debug exceptions in the debug mode. It should be set to the address of the Debug ROM of the NCEPLDM200 debug module, which is also the base address of the module. This address space should be a *device region* for proper operations of the external debug support. See Section 26.5 for more information.

2.11.3 Number of Trigger

The **Number of Trigger** option selects the number of hardware breakpoints.

2.11.4 Trace Interface

Specifying this option to "instruction-ratified" or "instruction-gen1" enables the instruction trace interface. See Section 3.14.

2.11.5 Andes PC/GPR/CSR Probing Support

Specifying this option to enable the interface for PC and GPR probing. Please see Section 3.16.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.12 Device Region

AX45MP can specify at most eight static device regions, where a device region N ($N = 0 – 7$) is defined by the following two configuration options:

- **Device Region*N* Base**: the base address of the region,
- **Device Region*N* Mask**: the address mask of the region.

Official Release

An address is inside a region when:

(address[PALEN-1:12] & **Device Region*N* Mask[PALEN-1:12]**) == **Device Region*N* Base[PALEN-1:12]**.

A device region can be disabled by setting **Device Region*N* Base** to all ones and **Device Region*N* Mask** to all zeros. The minimum region size is 4 KiB.

Note that PMA entries have higher priorities than the static Device Region. Please see Section 13 for details about physical memory attributes, ordering and device regions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.13 RAM Control Signals

Several SRAMs are instantiated inside the AX45MP design. Functional behavior models are shipped with the design and they need to be replaced by vendor SRAM macros. Usually the vendor SRAM cells require additional control signals beyond the functional ones, such as reset, sleep and shutdown controls. To avoid customers having to edit all intermediate files involved in these hierarchies just to propagate these additional control signals, a set of side-band control signals are pre-connected through the design hierarchy to reach the SRAMs in the leaf modules:

Official Release

- **RAM Control In** provides additional input ports that connect to SRAM models.
- **RAM Control Out** provides additional output ports that connect to SRAM models.

With this setup, integration engineers can just edit the leaf modules to instantiate vendor SRAM cells without touching all intermediate files in the design. See Section 28.1 for details of memory models, and the signals are listed in Section 3.17.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.14 Prefix for Generating a Uniquified Design

Specifying a prefix string in this option enables uniquified flows to generate a uniquified design. The prefix string is appended to the following modules:

- ax45mp_cluster and its sub-modules
- ncepldm200 and its sub-modules
- nceplic100 and its sub-modules
- nceplmt100 and its sub-modules
- ncejdtm200 and its sub-modules

Official Release

Uniquified modules will be generated with the prefix string ($NDS_RTL_PREFIX) and stored in the same directory as the original module. The following uniquified lists are generated in **$NDS_HOME**/flists.

- **$NDS_RTL_PREFIX**_ax45mp_cluster_flist.in
- **$NDS_RTL_PREFIX**_ncepldm200_flist.in
- **$NDS_RTL_PREFIX**_nceplic100_flist.in
- **$NDS_RTL_PREFIX**_nceplmt100_flist.in
- **$NDS_RTL_PREFIX**_ncejdtm200_flist.in

Note

- Only simulation is available when generating a uniquified design.

2.15 Export Signals from Machine Timer (PLMT)

Specifying this option to “yes” to pin out “mtime_shadow_64_bin” from PLMT.

Note

- Only available when specifying a prefix string to the option “Prefix for Generating a Uniquified Design”.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.16 Platform and CPU Subsystem

AX45MP provides a reference platform and a reference CPU subsystem. The memory map of the reference platform can be configured through the **Platform and CPU Subsystem** option:

- **ae350**: AE350 memory map, see Table 179.
- **custom**: Customized memory map. Most of addresses is configurable but with certain limitations.

Official Release

Note that the memory map affects only the reference platform and CPU subsystem. Regardless of the memory map, individual ax45mp_cluster can be integrated to the platform and CPU subsystem.

2.17 PLIC_SWINT for Software Interrupts

Specifying this option to “yes” enables the PLIC_SWINT for Software Interrupts.

2.18 Platform Debug Options

This group of options is used to configure the platform debug subsystem, which consists of Jtag Debug Transfer Module (JDTM), Platform Debug Module (PLDM) and the connections between these modules. Unlike other configuration options, these options will be written to the platform configuration file instead of the core configuration file, and they will also affect the platform designs.

When the core debug feature (Section 2.11.1) is disabled, the debug subsystem and the related I/O ports will be removed, and all the options in this group will be disabled.

2.18.1 PLDM System Bus Access

Specifying this option to “yes” will allow PLDM to directly access the system bus by integrating a bus master into PLDM; otherwise, the CPU core will be utilized to access the system bus.

2.18.2 Secure Debug Support

This option enables the NCEDBGLOCK100 for secure debug feature, please see Section 26.7.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.19 Platform IP Configuration

The configuration options in this section control whether the corresponding peripheral IP should be instantiated in the AE350 platform.

Official Release

2.19.1 UART Support

Specifying this option to "yes" instantiates the first (UART1) or second (UART2) UART controller individually in the platform.

UART2 is the default COM port on the FPGA board. If UART2 is not configured, CPU cannot be accessed through terminal emulators.

See Section 32.2.6 Section 32.3.2 for UART pin assignments on the FPGA board and Section 23.5 for more details about the UART controller.

2.19.2 Programmable Interval Timer (PIT) Support

Specifying this option to "yes" instantiates the PIT controller in the platform. See Section 23.5 for more details about the PIT controller.

2.19.3 Watchdog Timer (WDT) Support

Specifying this option to "yes" instantiates the WDT controller in the platform. See Section 23.5 for more details about the WDT controller.

2.19.4 Real-Time Clock (RTC) Support

Specifying this option to "yes" instantiates the RTC controller in the platform. See Section 23.5 for more details about the RTC controller.

2.19.5 GPIO Support

Specifying this option to "yes" instantiates the GPIO controller in the platform. Note that the seven-segment LED and buttons are connected to GPIO ports on the FPGA board. If the GPIO support is not configured, these LED and buttons will not be accessible. See Section 32.1.6 for GPIO pin assignments on the FPGA board, Section 23.5 for more details about the GPIO controller.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

2.19.6 I2C Support

Specifying this option to “yes” instantiates the I2C controller in the platform. Note that the I2C ROM is connected to the I2C port on the FPGA board. If the I2C support is not configured, the I2C ROM will not be accessible. See Section 32.1.7 for I2C pin assignments on the FPGA board and Section 23.5 for more details about the I2C controller.

Official Release

2.19.7 SPI Support

Specifying this option to “yes” instantiates the first (SPI1) or second (SPI2) SPI controller individually in the platform. **Please note that the reset vector points to the SPI memory interface in the AE350 platform. The CPU core will fetch instructions from the ROM through SPI1 when the CPU core boots up.** If SPI1 is not configured, it is necessary to point the reset vector to another interface. See Section 23.5 for more details about the SPI controller.

2.19.8 DMA Support

Specifying this option to “yes” instantiates the DMA (Direct-Memory-Access) controller in the platform. See Section 23.5 for more details about the DMA controller.

2.19.9 DTROM Support

Specifying this option to “yes” enables the DTROM (Device Tree) ROM, DTROM hold the binary form of the Device Tree description. See Section 23.5 for more details about the DTROM.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3 Signal Descriptions

This section describes the interface ports of the AX45MP core. All signals are Active-High unless otherwise indicated. VALEN denotes the virtual address length.

The prefix "coreN_" indicates the signal is for Core N (N = 0 – 7).

Official Release

3.1 General Signals

Table 6: General Signals

Signal Name	Direction	Description
coreN_hart_id[63:0]	input	This signal indicates the CPU hart ID. Hart IDs might not necessarily be numbered contiguously in a multiprocessor system, but at least one hart must have a hart ID of zero.
coreN_ilm_boot	input	This signal controls the reset value `milmb.IEN`. Exists when ILM is configured.
coreN_dlm_boot	input	This signal controls the reset value `mdlmb.DEN`. Exists when DLM is configured.
coreN_reset_n	input	CPU reset (Active-Low)
coreN_l2_reset_n	input	CPU reset in `l2_clk` domain. Exists when **Core Interface** is specified to "asynchronous".
coreN_l2_clk	input	`l2_clk` for each core. Exists when **Core Interface** is specified to "asynchronous".
coreN_clk	input	CPU clock input
coreN_lm_clk	input	Local RAM clock input. `coreN_lm_clk` and `coreN_clk` are synchronous but their gating conditions are different.
coreN_lm_local_int	output	Local interrupt is asserted for local RAM. `coreN_clk` should be enabled on assertion of coreN_lm_local_int.
coreN_lm_reset_n	output	Local RAM wait-cycle interface reset (Active-Low). Exists when local memory wait-cycle interface is configured

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 6: (continued)

Signal Name	Direction	Description
coreN_dcu_clk	input	D-Cache clock input. `coreN_dcu_clk` and `coreN_clk` are synchronous but their gating conditions are different.
coreN_reset_vector[VALEN-1:0]	input	Default program counter value upon reset. It should normally be a 4-byte aligned value but 2-byte aligned value is also allowed; bit 0 of this input signal should be zero. The value of VALEN is based on the capability of the MMU scheme described in Section 2.5.2.
coreN_wfi_mode	output	This signal indicates that the processor is in the wait-for-interrupt mode. See Section 20.2.
l2_clk	input	Clock input for Level 2 modules
l2_resetn	input	Reset input for Level 2 modules (Active-Low)
l2c_bank*N*_data_ram_clk	input	Clock input for L2-Cache data RAMs. The frequency is an half of l2_clk, $N = 0 - 3$
l2c_bank*N*_data_ram_clk_en	input	This is a `l2_clk` clock domain signal indicating that the signals of L2C_DATA_RAM_CLK can be sampled/toggled at the coming rising edge of `l2_clk`. See Section 4.2.4 for details.

MMU Scheme	VALEN
Sv39	39
Sv48	48

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.2 Interrupt Signals

When the clock domain of the interrupt signals is different from the `coreN_clk` clock domain, prior to connecting the interrupt signals to ax45mp_core_top, 2 or 3 stages of synchronization flip-flops are required to avoid metastability in clock domain crossing.

Official Release

Table 7: Interrupt Signals

Signal Name	Direction	Description
coreN_meip	input	External interrupt pending
coreN_meiid[9:0]	input	External interrupt source ID, used in the vector interrupt mode
coreN_meiack	output	External interrupt acknowledgment, used in the vector interrupt mode
coreN_mtip	input	Timer interrupt pending
coreN_msip	input	Software interrupt pending
coreN_nmi	input	Non-maskable interrupt
coreN_seip	input	Supervisor-mode external interrupt pending
coreN_seiid[9:0]	input	Supervisor-mode external interrupt source ID, used in the vector interrupt mode
coreN_seiack	output	Supervisor-mode external interrupt acknowledgment, used in the vector interrupt mode
coreN_ueip	input	User-mode external interrupt pending
coreN_ueiid[9:0]	input	User-mode external interrupt source ID, used in the vector interrupt mode
coreN_ueiack	output	User-mode external interrupt acknowledgment, used in the vector interrupt mode

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.3 Debug Signals

Table 8: External Debug Signals

Signal Name	Direction	Description
coreN_debugint	input	Debug interrupt
coreN_resethaltreq	input	Halt-on-reset request
coreN_hart_unavail	output	This signal indicates that the processor is not available for accesses by the external debugger. The processor may be in the reset or some kind of power-gating state.
coreN_hart_halted	output	This signal indicates that the processor is halted.
coreN_hart_under_reset	output	This signal indicates that the processor is under reset.
coreN_stoptime	output	This signal indicates that the processor is in Debug Mode and timers should stop counting. This signal is controlled by `dcsr.STOPTIME`.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.4 Bus Master Signals

In synchronous configuration, bus master signals are sampled/driven in the `core_clk` clock domain when the corresponding `clk_en` signal is HIGH. Please see Section 4.2 if the bus clock frequency needs to be lower than the `core_clk` frequency.

Official Release

In asynchronous configuration, AXI interface signals are sampled/driven in the corresponding clock domain.

MEM interface signals provide connectivity to the AXI cacheable memory. MMIO interface signals provide connectivity to the AXI non-cacheable memory and peripherals.

The optional SPP interface signals provide alternative connectivity to the AXI peripherals besides the system bus.

Table 9: Bus AxID Bit-Width

Number of Processor Cores	BIU_ID_WIDTH
1/2/4	5
8	6

Table 10: Memory AXI Interface Signals

Signal Name	Direction	Description
mem_aclk_en	input	This is a `l2_clk` clock domain signal indicating that the data from the bus clock domain can be sampled at the coming rising edge of `l2_clk`. See Section 4.2 for details.
mem_aclk	input	MEM Bus clock input. See Section 4.2 for details.
mem_awid[BIU_ID_WIDTH-1:0]	output	Write address ID
mem_awaddr[BIU_ADDR_WIDTH-1:0]	output	Write address
mem_awlen[7:0]	output	Write burst length
mem_awsize[2:0]	output	Write burst size
mem_awburst[1:0]	output	Write burst type
mem_awlock	output	Write lock type
mem_awcache[3:0]	output	Write cache type
mem_awprot[2:0]	output	Write protection type
mem_awvalid	output	Write address valid
mem_awready	input	Write address ready

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 10: (continued)

Signal Name	Direction	Description
mem_wdata[BIU_DATA_WIDTH-1:0]	output	Write data
mem_wstrb[(BIU_DATA_WIDTH/8)-1:0]	output	Write strobes
mem_wlast	output	Write last
mem_wvalid	output	Write valid
mem_wready	input	Write ready
mem_bid[BIU_ID_WIDTH-1:0]	input	Write response ID
mem_bresp[1:0]	input	Write response
mem_bvalid	input	Write response valid
mem_bready	output	Write response ready
mem_arid[BIU_ID_WIDTH-1:0]	output	Read address ID
mem_araddr[BIU_ADDR_WIDTH-1:0]	output	Read address
mem_arlen[7:0]	output	Read burst length
mem_arsize[2:0]	output	Read burst size
mem_arburst[1:0]	output	Read burst type
mem_arlock	output	Read lock type
mem_arcache[3:0]	output	Read cache type
mem_arprot[2:0]	output	Read protection type
mem_arvalid	output	Read address valid
mem_arready	input	Read address ready
mem_rid[BIU_ID_WIDTH-1:0]	input	Read ID tag
mem_rdata[BIU_DATA_WIDTH-1:0]	input	Read data
mem_rresp[1:0]	input	Read response
mem_rlast	input	Read last
mem_rvalid	input	Read valid
mem_rready	output	Read ready

Official Release

Table 11: MMIO AXI Interface Signals

Signal Name	Direction	Description
mmio_aclk_en	input	This is a `l2_clk` clock domain signal indicating that the data from the bus clock domain can be sampled at the coming rising edge of `l2_clk`. See Section 4.2 for details.
mmio_aclk	input	MMIO Bus clock input. See Section 4.2 for details.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 11: (continued)

Signal Name	Direction	Description
mmio_awid[BIU_ID_WIDTH-1:0]	output	Write address ID
mmio_awaddr[BIU_ADDR_WIDTH-1:0]	output	Write address
mmio_awlen[7:0]	output	Write burst length
mmio_awsize[2:0]	output	Write burst size
mmio_awburst[1:0]	output	Write burst type
mmio_awlock	output	Write lock type
mmio_awcache[3:0]	output	Write cache type
mmio_awprot[2:0]	output	Write protection type
mmio_awvalid	output	Write address valid
mmio_awready	input	Write address ready
mmio_wdata[BIU_DATA_WIDTH-1:0]	output	Write data
mmio_wstrb[(BIU_DATA_WIDTH/8)-1:0]	output	Write strobes
mmio_wlast	output	Write last
mmio_wvalid	output	Write valid
mmio_wready	input	Write ready
mmio_bid[BIU_ID_WIDTH-1:0]	input	Write response ID
mmio_bresp[1:0]	input	Write response
mmio_bvalid	input	Write response valid
mmio_bready	output	Write response ready
mmio_arid[BIU_ID_WIDTH-1:0]	output	Read address ID
mmio_araddr[BIU_ADDR_WIDTH-1:0]	output	Read address
mmio_arlen[7:0]	output	Read burst length
mmio_arsize[2:0]	output	Read burst size
mmio_arburst[1:0]	output	Read burst type
mmio_arlock	output	Read lock type
mmio_arcache[3:0]	output	Read cache type
mmio_arprot[2:0]	output	Read protection type
mmio_arvalid	output	Read address valid
mmio_arready	input	Read address ready
mmio_rid[BIU_ID_WIDTH-1:0]	input	Read ID tag
mmio_rdata[BIU_DATA_WIDTH-1:0]	input	Read data
mmio_rresp[1:0]	input	Read response
mmio_rlast	input	Read last

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 11: (continued)

Signal Name	Direction	Description
mmio_rvalid	input	Read valid
mmio_rready	output	Read ready

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

SPP interface signals provide connectivity to the private peripheral IPs.

Table 12: SPP Interface Signals

Signal Name	Direction	Description
spp_aclk_en	input	This is a `l2_clk` clock domain signal indicating that the data from the bus clock domain can be sampled at the coming rising edge of `l2_clk`. See Section 4.2 for details.
spp_aclk	input	SPP Bus clock input. See Section 4.2 for details.
spp_awid[SPP_ID_WIDTH-1:0]	output	Write address ID
spp_awaddr[BIU_ADDR_WIDTH-1:0]	output	Write address
spp_awlen[7:0]	output	Write burst length
spp_awsize[2:0]	output	Write burst size
spp_awburst[1:0]	output	Write burst type
spp_awlock	output	Write lock type
spp_awcache[3:0]	output	Write cache type
spp_awprot[2:0]	output	Write protection type
spp_awvalid	output	Write address valid
spp_awready	input	Write address ready
spp_wdata[63:0]	output	Write data
spp_wstrb[7:0]	output	Write strobes
spp_wlast	output	Write last
spp_wvalid	output	Write valid
spp_wready	input	Write ready
spp_bid[SPP_ID_WIDTH-1:0]	input	Write response ID
spp_bresp[1:0]	input	Write response
spp_bvalid	input	Write response valid
spp_bready	output	Write response ready
spp_arid[SPP_ID_WIDTH-1:0]	output	Read address ID
spp_araddr[BIU_ADDR_WIDTH-1:0]	output	Read address
spp_arlen[7:0]	output	Read burst length
spp_arsize[2:0]	output	Read burst size
spp_arburst[1:0]	output	Read burst type
spp_arlock	output	Read lock type
spp_arcache[3:0]	output	Read cache type
spp_arprot[2:0]	output	Read protection type

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 12: (continued)

Signal Name	Direction	Description
spp_arvalid	output	Read address valid
spp_arready	input	Read address ready
spp_rid[SPP_ID_WIDTH-1:0]	input	Read ID tag
spp_rdata[63:0]	input	Read data
spp_rresp[1:0]	input	Read response
spp_rlast	input	Read last
spp_rvalid	input	Read valid
spp_rready	output	Read ready

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.5 Bus Slave Signals

AXI Slave Port signals allow external agents to access the local memories of the core through the AXI interface. These signals are present on the core interface only when configuration option **Local Memory — Slave Port Support** is "yes".

Official Release

3.5.1 AXI Slave Port Interface Signals

In synchronous configuration, they are sampled/driven in the `coreN_lm_clk` clock domain when `coreN_slv_clk_en` is HIGH. Please see Section 4.2 if the AXI Slave Port clock frequency needs to be lower than the `coreN_lm_clk` frequency.

In asynchronous configuration, they are sampled/driven in the `coreN_slv_clk` clock domain.

Table 13: AXI Slave Port Signals

Signal Name	Direction	Description
coreN_slv_reset_n	input	Reset signal (Active-Low) for the local RAM slave port
coreN_slv_clk_en	input	This is a `coreN_lm_clk` clock domain signal indicating that the data from the Slave Port clock domain can be sampled at the coming rising edge of `coreN_lm_clk`. See Section 4.2 for details.
coreN_slv_clk	input	Slave port clock input. See Section 4.2 for details.
coreN_slv_awid[SLAVE_PORT_ID_WIDTH-1:0]	input	Write address ID
coreN_slv_awaddr[BIU_ADDR_WIDTH-1:0]	input	Write address
coreN_slv_awlen[7:0]	input	Write burst length
coreN_slv_awsize[2:0]	input	Write burst size
coreN_slv_awburst[1:0]	input	Write burst type
coreN_slv_awlock	input	Write lock type
coreN_slv_awcache[3:0]	input	Write cache type
coreN_slv_awprot[2:0]	input	Write protection type
coreN_slv_awuser	input	Write select ILM/DLM (0:ILM, 1:DLM)
coreN_slv_awvalid	input	Write address valid
coreN_slv_awready	output	Write address ready

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 13: (continued)

Signal Name	Direction	Description
coreN_slv_wdata[BIU_DATA_WIDTH-1:0]	input	Write data
coreN_slv_wstrb[(BIU_DATA_WIDTH/8)-1:0]	input	Write strobes
coreN_slv_wlast	input	Write last
coreN_slv_wvalid	input	Write valid
coreN_slv_wready	output	Write ready
coreN_slv_bid[SLAVE_PORT_ID_WIDTH-1:0]	output	Write response ID
coreN_slv_bresp[1:0]	output	Write response
coreN_slv_bvalid	output	Write response valid
coreN_slv_bready	input	Write response ready
coreN_slv_arid[SLAVE_PORT_ID_WIDTH-1:0]	input	Read address ID
coreN_slv_araddr[BIU_ADDR_WIDTH-1:0]	input	Read address
coreN_slv_arlen[7:0]	input	Read burst length
coreN_slv_arsize[2:0]	input	Read burst size
coreN_slv_arburst[1:0]	input	Read burst type
coreN_slv_arlock	input	Read lock type
coreN_slv_arcache[3:0]	input	Read cache type
coreN_slv_arprot[2:0]	input	Read protection type
coreN_slv_aruser	input	Read select ILM/DLM (0:ILM, 1:DLM)
coreN_slv_arvalid	input	Read address valid
coreN_slv_arready	output	Read address ready
coreN_slv_rid[SLAVE_PORT_ID_WIDTH-1:0]	output	Read ID tag
coreN_slv_rdata[BIU_DATA_WIDTH-1:0]	output	Read data
coreN_slv_rresp[1:0]	output	Read response
coreN_slv_rlast	output	Read last
coreN_slv_rvalid	output	Read valid
coreN_slv_rready	input	Read ready

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.6 I/O Coherence Port Signals

These signals are presented on the ax45mp_cluster interface and they are exists when configuration option **Cluster Configuration — I/O Coherence Port** is "1".

Official Release

Table 14: I/O Coherence Port Signals

Signal Name	Direction	Description
iocp0_aclk_en	input	This is a `l2_clk` clock domain signal indicating that the data from the bus clock domain can be sampled at the coming rising edge of `l2_clk`. See Section 4.2 for details.
iocp0_aclk	input	IOCP0 Bus clock input. See Section 4.2 for details.
iocp0_awid[IOCP_ID_WIDTH-1:0]	input	Write address ID
iocp0_awaddr[BIU_ADDR_WIDTH-1:0]	input	Write address
iocp0_awlen[7:0]	input	Write burst length
iocp0_awsize[2:0]	input	Write burst size
iocp0_awburst[1:0]	input	Write burst type
iocp0_awlock	input	Write lock type
iocp0_awcache[3:0]	input	Write cache type
iocp0_awprot[2:0]	input	Write protection type
iocp0_awvalid	input	Write address valid
iocp0_awready	output	Write address ready
iocp0_wdata[BIU_DATA_WIDTH-1:0]	input	Write data
iocp0_wstrb[(BIU_DATA_WIDTH/8)-1:0]	input	Write strobes
iocp0_wlast	input	Write last
iocp0_wvalid	input	Write valid
iocp0_wready	output	Write ready
iocp0_bid[IOCP_ID_WIDTH-1:0]	output	Write response ID
iocp0_bresp[1:0]	output	Write response
iocp0_bvalid	output	Write response valid
iocp0_bready	input	Write response ready
iocp0_arid[IOCP_ID_WIDTH-1:0]	input	Read address ID
iocp0_araddr[BIU_ADDR_WIDTH-1:0]	input	Read address
iocp0_arlen[7:0]	input	Read burst length
iocp0_arsize[2:0]	input	Read burst size
iocp0_arburst[1:0]	input	Read burst type

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 14: (continued)

Signal Name	Direction	Description
iocp0_arlock	input	Read lock type
iocp0_arcache[3:0]	input	Read cache type
iocp0_arprot[2:0]	input	Read protection type
iocp0_arvalid	input	Read address valid
iocp0_arready	output	Read address ready
iocp0_rid[IOCP_ID_WIDTH-1:0]	output	Read ID tag
iocp0_rdata[BIU_DATA_WIDTH-1:0]	output	Read data
iocp0_rresp[1:0]	output	Read response
iocp0_rlast	output	Read last
iocp0_rvalid	output	Read valid
iocp0_rready	input	Read ready

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.7 L2 Cache Interface Signals

L2 cache interface signals provide connectivity to L2C SRAMs. All control signals are Active-High.

Table 15: L2 Cache Signals

Official Release

Signal Name	Direction	Description
l2c_err_int	output	L2 cache asynchronous error
l2c_disable_init	input	L2 cache RAM no initialization flag indicating whether to initialize L2 cache RAMs when L2 cache comes out of reset 0: Don't disable the initialization. 1: Disable the initialization.
L2 Cache Tag RAM (concatenation of all tag SRAM interface in the bank)		
l2c_bank*N*_tag_cs[L2C_BANK_TAG_RAM_INSTS-1:0]	output	Chip select, *N* = 0, 1, 2, 3
l2c_bank*N*_tag_we[L2C_BANK_TAG_RAM_INSTS-1:0]	output	Write enable
l2c_bank*N*_tag_addr [L2C_BANK_TAG_RAM_INSTS*L2C_TAG_RAM_AW-1:0]	output	Address
l2c_bank*N*_tag_wdata [L2C_BANK_TAG_RAM_INSTS*L2C_TAG_RAM_DW-1:0]	output	Write data
l2c_bank*N*_tag_rdata [L2C_BANK_TAG_RAM_INSTS*L2C_TAG_RAM_DW-1:0]	input	Read data
L2 Cache Data RAM (concatenation of all data SRAM interface in the bank)		
l2c_bank*N*_data_cs[L2C_BANK_DATA_RAM_INSTS-1:0]	output	Chip select, *N* = 0, 1, 2, 3
l2c_bank*N*_data_we[L2C_BANK_DATA_RAM_INSTS-1:0]	output	Write enable, if a RAM macro provides only byte write enable, the signal can be unconnected.
l2c_bank*N*_data_bwe [L2C_BANK_DATA_RAM_INSTS*(L2C_DATA_RAM_DW/8)-1:0]	output	Byte write enable
l2c_bank*N*_data_addr [L2C_BANK_DATA_RAM_INSTS*L2C_DATA_RAM_AW-1:0]	output	Address
l2c_bank*N*_data_wdata [L2C_BANK_DATA_RAM_INSTS*L2C_DATA_RAM_DW-1:0]	output	Write data

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 15: (continued)

Signal Name	Direction	Description
l2c_bank*N*_data_rdata [L2C_BANK_DATA_RAM_INSTS*L2C_DATA_RAM_DW-1:0]	input	Read data

Official Release

Table 16: L2 Cache Tag RAM Address Width

L2C Tag RAM Depth	L2C_TAG_RAM_AW
64	6
128	7
256	8
512	9
1024	10
2048	11
4096	12

The number of tag RAM instances per L2-Cache bank (L2C_BANK_TAG_RAM_INSTS) is determined by **L2C Cache Size**, **L2C Tag RAM Depth** and **Number of L2C Banks**.

Table 17: Number of Tag RAM Instances Per L2-Cache Bank (**Number of L2C Banks** Is 2)

Cache Size (KiB)	L2 Cache Tag RAM Address Width						
	6	**7**	**8**	**9**	**10**	**11**	**12**
128	16	-	-	-	-	-	-
256	32	16	-	-	-	-	-
512	64	32	16	-	-	-	-
1024	128	64	32	16	-	-	-
2048	-	128	64	32	16	-	-
4096	-	-	128	64	32	16	-
8192	-	-	-	128	64	32	16

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 18: Number of Tag RAM Instances Per L2-Cache Bank (**Number of L2C Banks** Is 4)

Cache Size (KiB)	L2 Cache Tag RAM Address Width						
	5	6	7	8	9	10	11
128	16	-	-	-	-	-	-
256	-	16	-	-	-	-	-
512	-	32	16	-	-	-	-
1024	-	64	32	16	-	-	-
2048	-	-	64	32	16	-	-
4096	-	-	-	64	32	16	-
8192	-	-	-	-	64	32	16

Official Release

Table 19: L2 Cache Tag RAM Data Width

Protection Scheme	Cache Size (KiB)	Protection Width	L2C_TAG_RAM_DW
none	**128**	N/A	BIU_ADDR_WIDTH-10+NHART
	256	N/A	BIU_ADDR_WIDTH-11+NHART
	512	N/A	BIU_ADDR_WIDTH-12+NHART
	1024	N/A	BIU_ADDR_WIDTH-13+NHART
	2048	N/A	BIU_ADDR_WIDTH-14+NHART
	4096	N/A	BIU_ADDR_WIDTH-15+NHART
	8192	N/A	BIU_ADDR_WIDTH-16+NHART
ECC	**128**	(BIU_ADDR_WIDTH-10+NHART) $\leq$ 32	BIU_ADDR_WIDTH-3+NHART
		(BIU_ADDR_WIDTH-10+NHART) > 32	BIU_ADDR_WIDTH-2+NHART
	256	(BIU_ADDR_WIDTH-11+NHART) $\leq$ 32	BIU_ADDR_WIDTH-4+NHART
		(BIU_ADDR_WIDTH-11+NHART) > 32	BIU_ADDR_WIDTH-3+NHART
	512	(BIU_ADDR_WIDTH-12+NHART) $\leq$ 32	BIU_ADDR_WIDTH-5+NHART
		(BIU_ADDR_WIDTH-12+NHART) > 32	BIU_ADDR_WIDTH-4+NHART
	1024	(BIU_ADDR_WIDTH-13+NHART) $\leq$ 32	BIU_ADDR_WIDTH-6+NHART
		(BIU_ADDR_WIDTH-13+NHART) > 32	BIU_ADDR_WIDTH-5+NHART
	2048	(BIU_ADDR_WIDTH-14+NHART) $\leq$ 32	BIU_ADDR_WIDTH-7+NHART
		(BIU_ADDR_WIDTH-14+NHART) > 32	BIU_ADDR_WIDTH-6+NHART
	4096	(BIU_ADDR_WIDTH-15+NHART) $\leq$ 32	BIU_ADDR_WIDTH-8+NHART
		(BIU_ADDR_WIDTH-15+NHART) > 32	BIU_ADDR_WIDTH-7+NHART
	8192	(BIU_ADDR_WIDTH-16+NHART) $\leq$ 32	BIU_ADDR_WIDTH-9+NHART

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 19: (continued)

Protection Scheme	Cache Size (KiB)	Protection Width	L2C_TAG_RAM_DW
		(BIU_ADDR_WIDTH-16+NHART) > 32	BIU_ADDR_WIDTH-8+NHART

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

L2 Cache Data RAM Size

Table 20: L2-Cache Data RAM Address Width

L2C Data RAM Depth	L2C_DATA_RAM_AW
1024	10
2048	11
4096	12
8192	13
16384	14
32768	15
65536	16

Official Release

The number of data RAM instances per L2-Cache bank (L2C_BANK_DATA_RAM_INSTS) is determined by **L2C Cache Size**, **L2C Data RAM Depth** and **Number of L2C Banks**.

Table 21: Number of Data RAM Instances Per L2-Cache Bank (**Number of L2C Banks** Is 2)

Cache Size (KiB)	L2 Cache Data RAM Address Width						
	10	**11**	**12**	**13**	**14**	**15**	**16**
128	1	-	-	-	-	-	-
256	2	1	-	-	-	-	-
512	4	2	1	-	-	-	-
1024	8	4	2	1	-	-	-
2048	16	8	4	2	1	-	-
4096	-	16	8	4	2	1	-
8192	-	-	16	8	4	2	1

Table 22: Number of Data RAM Instances Per L2-Cache Bank (**Number of L2C Banks** Is 4)

Cache Size (KiB)	L2 Cache Data RAM Address Width						
	9	**10**	**11**	**12**	**13**	**14**	**15**
128	1	-	-	-	-	-	-
256	-	1	-	-	-	-	-
512	-	2	1	-	-	-	-
1024	-	4	2	1	-	-	-
2048	-	8	4	2	1	-	-

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 22: (continued)

Cache Size (KiB)	L2 Cache Data RAM Address Width						
	9	10	11	12	13	14	15
4096	-	-	8	4	2	1	-
8192	-	-	-	8	4	2	1

Official Release

Table 23: L2 Cache Data RAM Data Width

Protection Scheme	L2C_DATA_RAM_DW
none	512
ECC	576

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.8 Instruction Local Memory Interface Signals

ILM interface signals provide connectivity for accessing the Instruction Local Memory. AX45MP supports RAM interface and wait-cycle interface.

Official Release

3.8.1 ILM RAM Interface

When **ILM Wait Cycle** is specified to "no", interface signals described in Table 24 and Table 25. The `ILM_RAM_AW` is defined in Table 26. The timing diagram is shown as Figure 7.

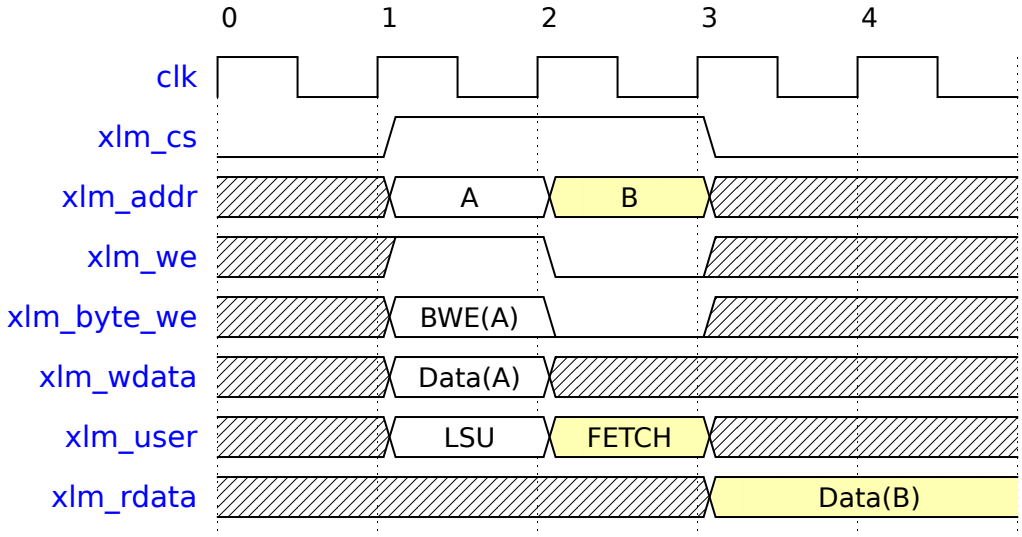


Figure 7: Timing Diagram for RAM Type LM Interface

Please see Section 28 for organization of ILM memories when RAM interface is selected.

Table 24: ILM SRAM Interface Signals (**ILM Soft Error Protection** Is None)

Signal Name	Direction	Description
ilm0_cs	output	Chip select
ilm0_we	output	Write enable. If a RAM macro provides only byte write enable, the signal can be unconnected.
ilm0_byte_we[7:0]	output	Byte write enable
ilm0_addr[ILM_RAM_AW-1:0]	output	Address
ilm0_wdata[63:0]	output	Write data

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 24: (continued)

Signal Name	Direction	Description
ilm0_user[1:0]	output	RAM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not applied.
ilm0_rdata[63:0]	input	Read data

Official Release

Table 25: ILM SRAM Interface Signals (**ILM Soft Error Protection** Is ECC)

Signal Name	Direction	Description
ilm0_cs	output	Chip select
ilm0_we	output	Write enable
ilm0_byte_we[7:0]	output	Byte write enable (not used, this signal can be unconnected)
ilm0_addr[ILM_RAM_AW-1:0]	output	Address
ilm0_wdata[71:0]	output	Write data
ilm0_user[1:0]	output	RAM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not applied.
ilm0_rdata[71:0]	input	Read data

Table 26: Instruction Local Memory RAM Address Bit-Width

Size	ILM_RAM_AW
4KiB	9
8KiB	10
16KiB	11
32KiB	12
64KiB	13
128KiB	14
256KiB	15
512KiB	16
1MiB	17
2MiB	18
4MiB	19
8MiB	20

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 26: (continued)

Size	ILM_RAM_AW
16MiB	21

Official Release

3.8.2 ILM Wait Cycle Interface

AX45MP provides a ILM wait-cycle interface, which can be selected by specifying **ILM Wait Cycle** to "yes". Table 27 shows the wait-cycle interface signals.

ILM wait-cycle interface consists of two channels:

- A-channel transmits requests from the core to ILM, and the signals are prefixed with *coreN_ilm_a*. Table 29 shows all possible operations that the core might send on A-channel.
- D-channel transmits data response from ILM to the core, and the signals are prefixed with *coreN_ilm_d*.

Both A-channel and D-channel implement valid-ready handshaking. An operation is transmitted only when valid and ready are both asserted. When the core sends multiple outstanding requests on A-channel, it expects to receive in-order responses from D-channel.

Table 27: ILM Wait-Cycle Interface Signals

Signal Name	Direction	Description
coreN_ilm_a_valid	output	Request valid on A-channel
coreN_ilm_a_ready	input	Request ready on A-channel
coreN_ilm_a_opcode[2:0]	output	Operation code (0x1: PutPartialData, 0x4: Get).
coreN_ilm_a_size[2:0]	output	Request byte size
coreN_ilm_a_mask[7:0]	output	Request byte mask
coreN_ilm_a_addr[ILM_AW-1:3]	output	Byte address for the request. Bits 2–0 are not present on the interface, and the values are zeros. `ILM_AW` is defined in Table 28.
coreN_ilm_a_data[63:0]	output	Write data
coreN_ilm_a_user[1:0]	output	ILM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not applied.
coreN_ilm_a_parity0[7:0]	output	Write data ECC code for data block 0. The signal can be unconnected if ECC is not configured.
coreN_ilm_a_parity1[7:0]	output	Write data ECC code for data block 1. The signal can be unconnected if ECC is not configured.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 27: (continued)

Signal Name	Direction	Description
coreN_ilm_d_valid	input	Response valid on D-channel
coreN_ilm_d_ready	output	Response ready on D-channel
coreN_ilm_d_data[63:0]	input	Read data
coreN_ilm_d_parity0[7:0]	input	Read data ECC code for data block 0. The signal can be unconnected if ECC is not configured.
coreN_ilm_d_parity1[7:0]	input	Read data ECC code for data block 1. The signal can be unconnected if ECC is not configured.
coreN_ilm_d_denied	input	The operation is denied (Bus error)

Official Release

Table 28: Instruction Local Memory Address Bit-Width

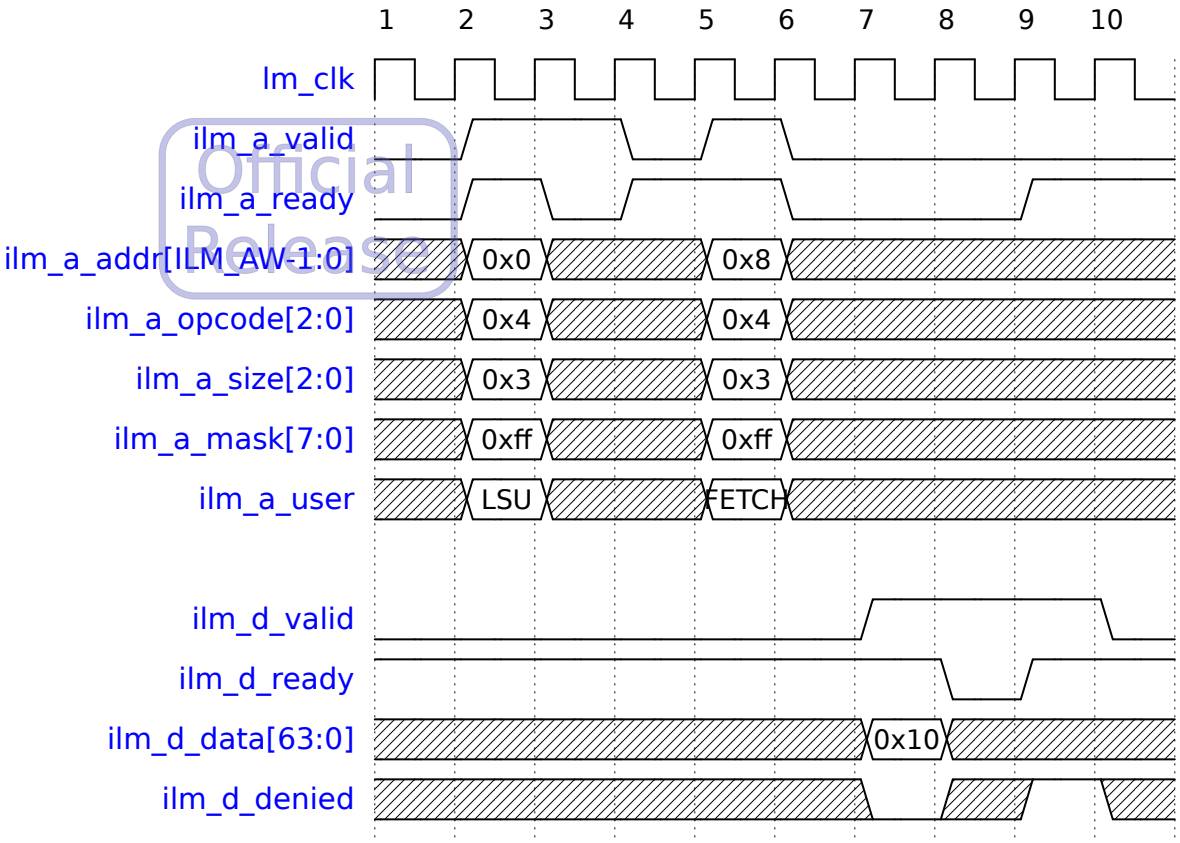
Size	ILM_AW
4KiB	12
8KiB	13
16KiB	14
32KiB	15
64KiB	16
128KiB	17
256KiB	18
512KiB	19
1MiB	20
2MiB	21
4MiB	22
8MiB	23
16MiB	24

Table 29: Possible Operations on ILM Wait Cycle Interfaces

Operation	coreN_ilm_a_opcode	coreN_ilm_a_size	coreN_ilm_a_mask
Read	0x4(Get)	0x3	0xff
Write (w/o ECC)	0x1(PutPartialData)	0x3	any value
Write (w/ ECC)	0x1(PutPartialData)	0x3	0x0 or 0xff

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Figure 8 is an example for **Get** operations.



Official Release

Figure 8: Get Operation

In the example, two requests are respectively taken at cycle 2 and 5. The *ilm_a_opcode* represents the request operation, where 0x4 is for **Get** operations. The *ilm_a_addr* indicates the address of the operation, and the values are always 8-byte aligned. The *ilm_a_size* is the size of this operation, and the values are always 3 (8 bytes). The *ilm_a_mask* selects the byte lanes to read, and the values are always 0xff. The *ilm_a_user* indicates the source of the operation.

In the example, two responses are respectively taken at cycle 7 and 9. The *ilm_d_data* represents the data of the response. ILM can assert *ilm_d_denied* to deny the operation. An fetch access fault exception is triggered for a denied IFU operation, and an load access fault exception is triggered for a denied LSU operation.

If **ILM Soft Error Protection** is configured, the *ilm_d_parity0* sideband signal is provided to transmit ECC encode data from the slave device to AX45MP core. The *ilm_d_parity1* should be unconnected.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Figure 9 is an example for **PutPartialData** operations.

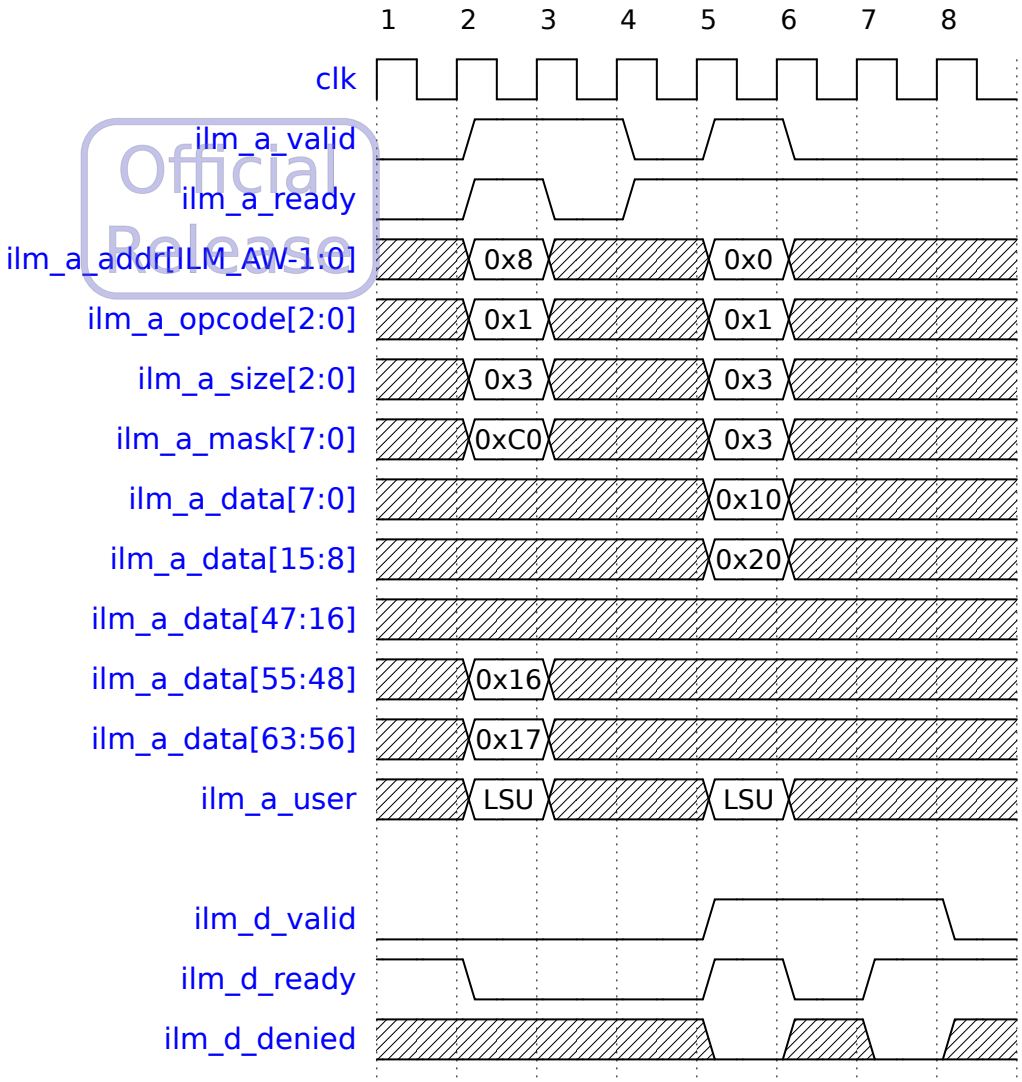


Figure 9: PutPartialData Operation

In this example, two requests are taken at cycle 2 and 5. The *ilm_a_opcode* signal represents the operation code, where 0x1 is for **PutPartialData** operations. The *ilm_a_addr* signal indicates the address of the operation, and the values are always 8-byte aligned. The *ilm_a_size* signal is the size of this operation, and the values are always 0x3 (8 bytes). The *ilm_a_mask* signal selects byte lanes to write. The *ilm_a_user* signal indicates the source of the operation.

In the example, two responses are respectively taken at cycle 5 and 7. If the *ilm_d_denied* signal is asserted, the operation is denied. A store/AMO access fault exception or store bus error local interrupt is triggered for a denied **PutPartialData** operation. The *ilm_d_data* signal is ignored for **PutPartialData** operations.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

If **ILM Soft Error Protection** is configured, the *ilm_a_parity0* sideband signal is provided to transmit ECC encode data from CPU to the slave device. *ilm_d_parity1* should be unconnected.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.9 Data Local Memory Interface Signals

DLM interface signals provide connectivity for accessing the Data Local Memory. AX45MP supports RAM interface and wait-cycle interface.

Official Release

3.9.1 DLM RAM Interface

Interface signals described in Table 30 and Table 31 provide connectivity to the DATA Local Memory RAM of the processor. The timing diagram is shown as Figure 7.

Please see Section 28 for organization of DLM memories. See Table 32 for definitions of `DLM_RAM_AW`.

Table 30: Data Local Memory Interface Signals (**DLM Soft Error Protection** Is None)

Signal Name	Direction	Description
dlm_cs	output	Chip select
dlm_we	output	Write enable. If a RAM macro provides only byte write enable, the signal can be unconnected.
dlm_byte_we[7:0]	output	Byte write enable
dlm_addr[DLM_RAM_AW-1:0]	output	Address
dlm_wdata[63:0]	output	Write data
dlm_user[1:0]	output	RAM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not used.
dlm_rdata[63:0]	input	Read data
dlm*N*_cs	output	Chip select, *N*=1, 2, 3
dlm*N*_we	output	Write enable. If a RAM macro provides only byte write enable, the signal can be unconnected.
dlm*N*_byte_we[7:0]	output	Byte write enable
dlm*N*_addr[DLM_RAM_AW-1:0]	output	Address
dlm*N*_wdata[63:0]	output	Write data
dlm*N*_user[1:0]	output	RAM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not used.
dlm*N*_rdata[63:0]	input	Read data

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 31: Data Local Memory Interface Signals (**DLM Soft Error Protection** Is ECC)

Signal Name	Direction	Description
dlm_cs	output	Chip select
dlm_we	output	Write enable
dlm_byte_we[7:0]	output	Byte write enable (not used, the signal can be unconnected)
dlm_addr[DLM_RAM_AW-1:0]	output	Address
dlm_wdata[71:0]	output	Write data
dlm_user[1:0]	output	RAM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not used.
dlm_rdata[71:0]	input	Read data
dlm*N*_cs	output	Chip select, *N*=1, 2, 3
dlm*N*_we	output	Write enable
dlm*N*_byte_we[7:0]	output	Byte write enable (not used, the signal can be unconnected)
dlm*N*_addr[DLM_RAM_AW-1:0]	output	Address
dlm*N*_wdata[71:0]	output	Write data
dlm*N*_user[1:0]	output	RAM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not used.
dlm*N*_rdata[71:0]	input	Read data

Official Release

Note

- dlm, dlm1, dlm2, and dlm3 are for DLM bank 0, bank 1, bank 2, and bank 3 respectively. dlm1 is present when **Number of DLM Banks** is 2 or 4. dlm2 and dlm3 are present when **Number of DLM Banks** is 4.

Table 32: Data Local Memory Address Bit-Width

Size	Banks	DLM_RAM_AW
4KiB	1	9
	2	
	4	7
8KiB	1	10
	2	
	4	8

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 32: (continued)

Size	Banks	DLM_RAM_AW
16KiB	1	11
	2	
	4	9
32KiB	1	12
	2	
	4	10
64KiB	1	13
	2	
	4	11
128KiB	1	14
	2	
	4	12
256KiB	1	15
	2	
	4	13
512KiB	1	16
	2	
	4	14
1MiB	1	17
	2	
	4	15
2MiB	1	18
	2	
	4	16
4MiB	1	19
	2	
	4	17
8MiB	1	20
	2	
	4	18
16MiB	1	21
	2	

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 32: (continued)

Size	Banks	DLM_RAM_AW
	4	19

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.9.2 DLM Wait Cycle Interface

AX45MP provides a DLM wait-cycle interface, which can be selected by specifying **DLM Wait Cycle** to "yes". Table 33 shows the wait-cycle interface signals.

The DLM wait-cycle interface consists of two channels:

Official Release

- A-channel transmits requests from the core to DLM, and the signals are prefixed with *dlm_a*. Table 35 shows all possible operations that the core might send on A-channel.
- D-channel transmits data responses from DLM to the core, and the signals are prefixed with *dlm_d*.

Both A-channel and D-channel implement valid-ready handshaking. An operation is transmitted only when valid and ready signals are both asserted. When the core sends multiple outstanding requests on A-channel, it expects to receive in-order responses from D-channel.

Table 33: DLM Wait-Cycle Interface Signals

Signal Name	Direction	Description
coreN_dlm_a_valid	output	Request valid on A-channel
coreN_dlm_a_ready	input	Request ready on A-channel
coreN_dlm_a_opcode[2:0]	output	Operation code (0x1: PutPartialData, 0x4: Get).
coreN_dlm_a_size[2:0]	output	Request byte size
coreN_dlm_a_mask[7:0]	output	Request byte mask
coreN_dlm_a_addr[DLM_AW-1:3]	output	Byte address for the request. Bits 2–0 are not present on the interface, and the values are zeros. `DLM_AW` is defined in Table 34.
coreN_dlm_a_data[63:0]	output	Write data
coreN_dlm_a_user[1:0]	output	DLM access source (0: LSU, 1: IFU, 2: Slave Port). The signal can be unconnected when the access source is not applied.
coreN_dlm_a_parity[7:0]	output	Write data ECC code. The signal can be unconnected if ECC is not configured.
coreN_dlm_d_valid	input	Response valid on D-channel
coreN_dlm_d_ready	output	Response ready on D-channel
coreN_dlm_d_data[63:0]	input	Read data
coreN_dlm_d_parity[7:0]	input	Read data ECC code. The signal can be unconnected if ECC is not configured.
coreN_dlm_d_denied	input	The operation is denied (Bus error)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 34: Data Local Memory Address Bit-Width

Size	DLM_AW
4KiB	12
8KiB	13
16KiB	14
32KiB	15
64KiB	16
128KiB	17
256KiB	18
512KiB	19
1MiB	20
2MiB	21
4MiB	22
8MiB	23
16MiB	24

Official Release

Table 35: Possible Operations on DLM Wait Cycle Interfaces

Operation	dlm_a_opcode	dlm_a_size	dlm_a_mask
Read	0x4(Get)	0x3	0xff
write (w/o ECC)	0x1(PutPartialData)	0x3	any value
write (w/ ECC)	0x1(PutPartialData)	0x3	0x0 or 0xff

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Figure 10 is an example for **Get** operations.

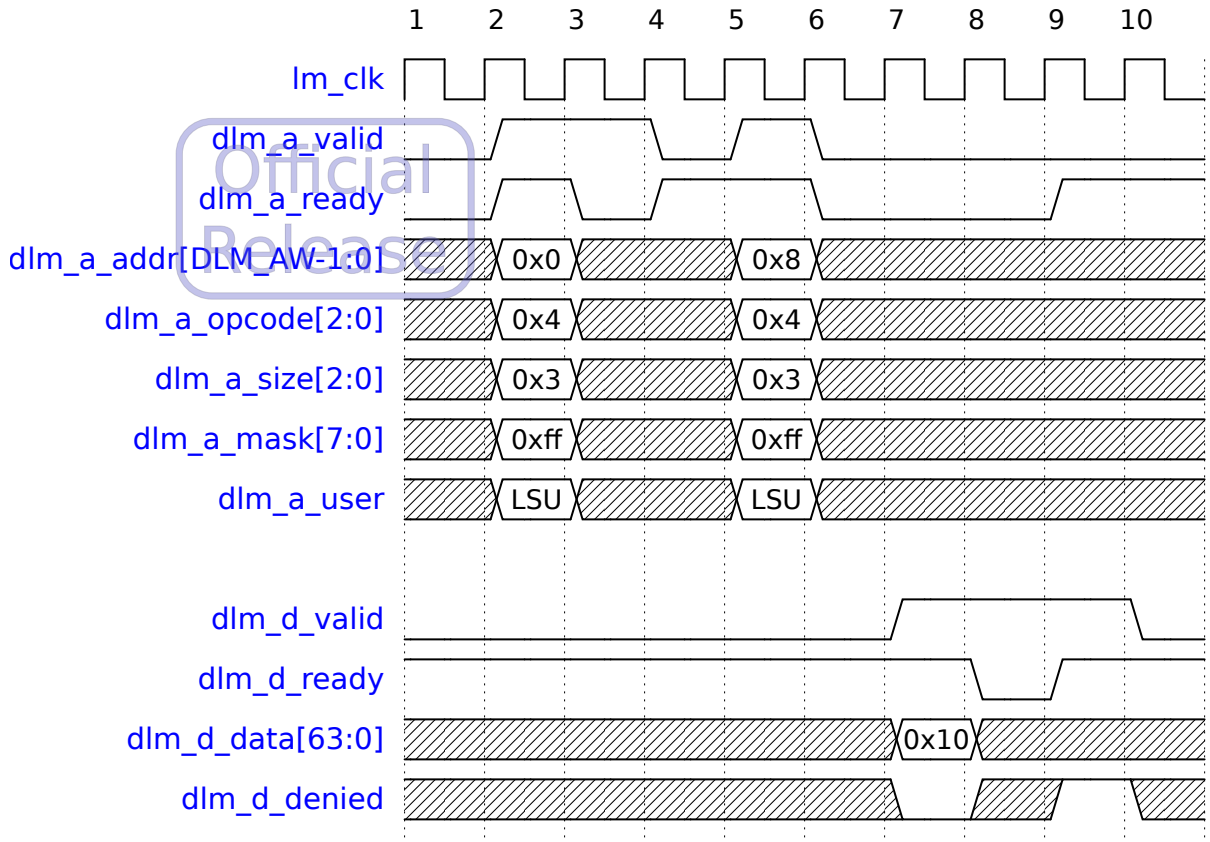


Figure 10: Get Operation

In the example, two requests are respectively taken at cycle 2 and 5. The *dlm_a_opcode* signal represents the request operation, where 0x4 is for **Get** operations. The *dlm_a_addr* signal indicates the address of the operation, and the value is always 8-byte aligned. The *dlm_a_size* signal is the size of this operation, and the value is always 3 (=8 bytes). The *dlm_a_mask* signal selects the byte lanes to read, and the value is always 0xff. The *dlm_a_user* signal indicates the source of the operation.

In the example, two responses are respectively taken at cycle 7 and 9. The *dlm_d_data* signal represents the data of the response. DLM can assert *dlm_d_denied* to deny the operation. An load access fault exception is triggered for a denied LSU operation.

If **DLM Soft Error Protection** is configured, the *dlm_d_parity* sideband signal is provided to transmit ECC encode data from the slave device to the AX45MP core.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Figure 11 is an example for **PutPartialData** operations.

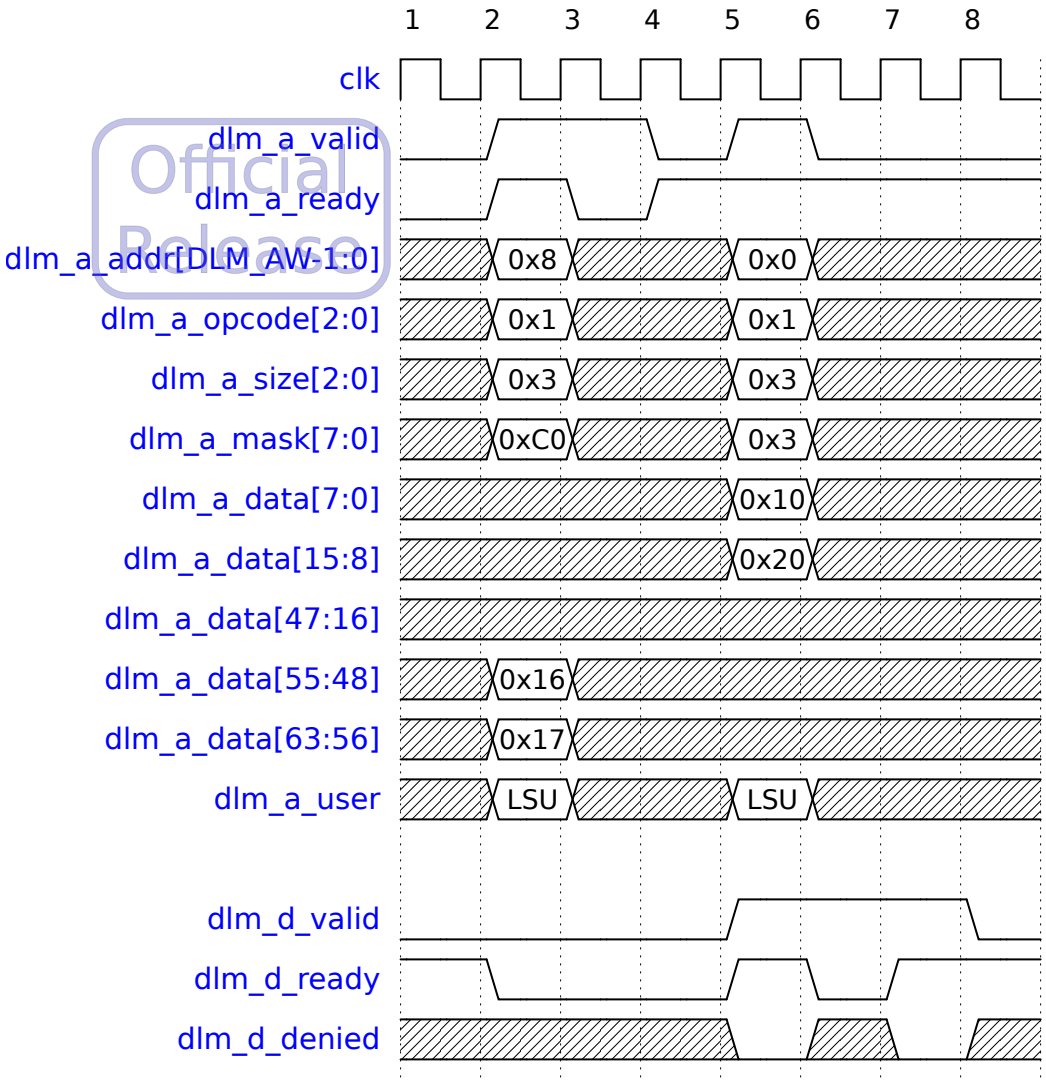


Figure 11: PutPartialData Operation

In this example, two requests are taken at cycle 2 and 5 respectively. The *dlm_a_opcode* signal represents the operation code, where 0x1 is for **PutPartialData** operations. The *dlm_a_addr* signal indicates the address of the operation, and the value is always 8-byte aligned. The *dlm_a_size* signal is the size of this operation, and the value is always 0x3 (=8 bytes). The *dlm_a_mask* signal selects byte lanes to write. The *dlm_a_user* signal indicates the source of the operation.

In the example, two responses are respectively taken at cycle 5 and 7. If *dlm_d_denied* is asserted, the operation is denied. A store/AMO access fault exception or store bus error local interrupt is triggered for a denied **PutPartialData** operation. The *dlm_d_data* signal is ignored for **PutPartialData** operations.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

If **DLM Soft Error Protection** is configured, the *dlm_a_parity* sideband signal is provided to transmit ECC encode data from CPU to the slave device.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.10 Instruction Cache Interface Signals

I-Cache interface signals provide connectivity to the I-Cache SRAMs of the processor. Please see Section 28 for organization of I-Cache SRAMs. See Table 37 to Table 40 for bit-width definitions.

Official Release

Table 36: Instruction Cache Interface Signals

Signal Name	Direction	Description
coreN_icache_disable_init	input	Disable the initialization of I-Cache RAMs when the processor exits the reset state. Assertion of this signal is to speed up the power-gating wakeup process when the content of I-Cache SRAM is preserved during power-down.
I-Cache Tag RAM		
icache_tag*N*_cs	output	Chip select, *N*=0, 1, 2, 3
icache_tag*N*_we	output	Write enable
icache_tag*N*_addr[ICACHE_TAG_RAM_AW-1:0]	output	Address
icache_tag*N*_wdata[ICACHE_TAG_RAM_DW-1:0]	output	Write data
icache_tag*N*_rdata[ICACHE_TAG_RAM_DW-1:0]	input	Read data
I-Cache Data RAM		
icache_data*N*_cs	output	Chip select, *N*=0, 1, 2, 3, 4, 5, 6, 7
icache_data*N*_we	output	Write enable
icache_data*N*_addr[ICACHE_DATA_RAM_AW-1:0]	output	Address
icache_data*N*_wdata[ICACHE_DATA_RAM_DW-1:0]	output	Write data
icache_data*N*_rdata[ICACHE_DATA_RAM_DW-1:0]	input	Read data

Table 37: I-Cache Tag Address Bit-Width

Associativity	Size (KiB)	ICACHE_TAG_RAM_AW
1	8	7
	16	8
	32	9
	64	10
2	8	6

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 37: (continued)

Associativity	Size (KiB)	ICACHE_TAG_RAM_AW
	16	7
	32	8
	64	9
4	8	5
	16	6
	32	7
	64	8

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 38: I-Cache Tag Data Bit-Width

Protection Scheme	Associativity	Size (KiB)	Protection Width > 32	ICACHE_TAG_RAM_DW
none	1	8	N/A	BIU_ADDR_WIDTH-9
		16	N/A	BIU_ADDR_WIDTH-9
		32	N/A	BIU_ADDR_WIDTH-9
		64	N/A	BIU_ADDR_WIDTH-9
	2	8	N/A	BIU_ADDR_WIDTH-9
		16	N/A	BIU_ADDR_WIDTH-9
		32	N/A	BIU_ADDR_WIDTH-9
		64	N/A	BIU_ADDR_WIDTH-9
	4	8	N/A	BIU_ADDR_WIDTH-8
		16	N/A	BIU_ADDR_WIDTH-9
		32	N/A	BIU_ADDR_WIDTH-9
		64	N/A	BIU_ADDR_WIDTH-9
parity	1	8	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		16	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		32	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		64	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
	2	8	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		16	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		32	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		64	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
	4	8	(BIU_ADDR_WIDTH-8) $\leq$ 32	BIU_ADDR_WIDTH-4
			(BIU_ADDR_WIDTH-8) > 32	BIU_ADDR_WIDTH
		16	(BIU_ADDR_WIDTH-9) $\leq$ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 38: (continued)

Protection Scheme	Associativity	Size (KiB)	Protection Width > 32	ICACHE_TAG_RAM_DW
ecc	1	32	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		64	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-5
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		8	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		16	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		32	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		64	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
	2	8	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		16	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		32	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		64	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
	4	8	(BIU_ADDR_WIDTH-8) ≤ 32	BIU_ADDR_WIDTH-1
			(BIU_ADDR_WIDTH-8) > 32	BIU_ADDR_WIDTH
		16	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		32	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1
		64	(BIU_ADDR_WIDTH-9) ≤ 32	BIU_ADDR_WIDTH-2
			(BIU_ADDR_WIDTH-9) > 32	BIU_ADDR_WIDTH-1

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 39: I-Cache Data Address Bit-Width

Size(KiB)	ICACHE_DATA_RAM_AW
8	8
16	9
32	10
64	11

Official Release

Table 40: I-Cache Data Bit-Width

Protection Scheme	ICACHE_DATA_RAM_DW
none	32
parity	36
ecc	39

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.11 Data Cache Interface Signals

D-Cache interface signals provide connectivity to D-Cache SRAMs of the processor. Please see Section 28 for organization of D-Cache SRAMs. See Table 42 to Table 44 for bit-width definitions.

Official Release

Table 41: Data Cache Interface Signals

Signal Name	Direction	Description
coreN_dcache_disable_init	input	Disable the initialization of D-Cache RAMs when the processor exits the reset state. Assertion of this signal is to speed up the power-gating wakeup process when the content of D-Cache SRAM is preserved during power-down.
D-Cache Tag RAM		
dcache_tag*N*_cs	output	Chip select, *N*=0, 1, 2, 3
dcache_tag*N*_we	output	Write enable
dcache_tag*N*_addr[DCACHE_TAG_RAM_AW-1:0]	output	Address
dcache_tag*N*_wdata[DCACHE_TAG_RAM_DW-1:0]	output	Write data
dcache_tag*N*_rdata[DCACHE_TAG_RAM_DW-1:0]	input	Read data
D-Cache Data RAM		
dcache_data*N*_cs	output	Chip select, *N*=0, 1, 2, 3
dcache_data*N*_we	output	Write enable. If a RAM macro provides only byte write enable, the signal can be unconnected.
dcache_data*N*_byte_we[DCACHE_DATA_RAM_BWEW-1:0]	output	Byte write enable
dcache_data*N*_addr[DCACHE_DATA_RAM_AW-1:0]	output	Address
dcache_data*N*_wdata[63:0]	output	Write data
dcache_data*N*_rdata[63:0]	input	Read data
D-Cache WPT RAM		
dcache_wpt_cs	output	Chip select
dcache_wpt_we	output	Write enable. If a RAM macro provides only byte write enable, the signal can be unconnected.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 41: (continued)

Signal Name	Direction	Description
dcache_wpt_byte_we[1:0]	output	Byte write enable
dcache_wpt_addr[8:0]	output	Address
dcache_wpt_wdata[15:0]	output	Write data
dcache_wpt_rdata[15:0]	input	Read data

Official Release

Note

When **D-Cache Soft Error Protection** is ecc, the processor always writes a full word. The dcache_data*N*_byte_we can be unconnected.

Table 42: D-Cache Tag Address Bit-Width

Associativity	Size (KiB)	DCACHE_TAG_RAM_AW
1	8	7
	16	8
	32	9
	64	10
2	8	6
	16	7
	32	8
	64	9
4	8	5
	16	6
	32	7
	64	8

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 43: D-Cache Tag Bit-Width

Protection Scheme	Associativity	Size (KiB)	Protection Width > 32	DCACHE_TAG_RAM_DW
none	1	8	N/A	BIU_ADDR_WIDTH-9
		16	N/A	BIU_ADDR_WIDTH-10
		32	N/A	BIU_ADDR_WIDTH-11
		64	N/A	BIU_ADDR_WIDTH-12
	2	8	N/A	BIU_ADDR_WIDTH-8
		16	N/A	BIU_ADDR_WIDTH-9
		32	N/A	BIU_ADDR_WIDTH-10
		64	N/A	BIU_ADDR_WIDTH-11
	4	8	N/A	BIU_ADDR_WIDTH-7
		16	N/A	BIU_ADDR_WIDTH-8
		32	N/A	BIU_ADDR_WIDTH-9
		64	N/A	BIU_ADDR_WIDTH-10
ecc	1	8	BIU_ADDR_WIDTH-9 $\leq$ 32	BIU_ADDR_WIDTH-2
			BIU_ADDR_WIDTH-9 > 32	BIU_ADDR_WIDTH-1
		16	BIU_ADDR_WIDTH-10 $\leq$ 32	BIU_ADDR_WIDTH-3
			BIU_ADDR_WIDTH-10 > 32	BIU_ADDR_WIDTH-2
		32	BIU_ADDR_WIDTH-11 $\leq$ 32	BIU_ADDR_WIDTH-4
			BIU_ADDR_WIDTH-11 > 32	BIU_ADDR_WIDTH-3
		64	BIU_ADDR_WIDTH-12 $\leq$ 32	BIU_ADDR_WIDTH-5
			BIU_ADDR_WIDTH-12 > 32	BIU_ADDR_WIDTH-4
	2	8	BIU_ADDR_WIDTH-8 $\leq$ 32	BIU_ADDR_WIDTH-1
			BIU_ADDR_WIDTH-8 > 32	BIU_ADDR_WIDTH
		16	BIU_ADDR_WIDTH-9 $\leq$ 32	BIU_ADDR_WIDTH-2
			BIU_ADDR_WIDTH-9 > 32	BIU_ADDR_WIDTH-1
		32	BIU_ADDR_WIDTH-10 $\leq$ 32	BIU_ADDR_WIDTH-3
			BIU_ADDR_WIDTH-10 > 32	BIU_ADDR_WIDTH-2
		64	BIU_ADDR_WIDTH-11 $\leq$ 32	BIU_ADDR_WIDTH-4
			BIU_ADDR_WIDTH-11 > 32	BIU_ADDR_WIDTH-3
	4	8	BIU_ADDR_WIDTH-7 $\leq$ 32	BIU_ADDR_WIDTH
			BIU_ADDR_WIDTH-7 > 32	BIU_ADDR_WIDTH+1
		16	BIU_ADDR_WIDTH-8 $\leq$ 32	BIU_ADDR_WIDTH-1
			BIU_ADDR_WIDTH-8 > 32	BIU_ADDR_WIDTH

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 43: (continued)

Protection Scheme	Associativity	Size (KiB)	Protection Width > 32	DCACHE_TAG_RAM_DW
		32	BIU_ADDR_WIDTH-9 $\leq$ 32	BIU_ADDR_WIDTH-2
			BIU_ADDR_WIDTH-9 > 32	BIU_ADDR_WIDTH-1
		64	BIU_ADDR_WIDTH-10 $\leq$ 32	BIU_ADDR_WIDTH-3
			BIU_ADDR_WIDTH-10 > 32	BIU_ADDR_WIDTH-2

Official Release

Table 44: D-Cache Data Address Bit-Width

Protection Scheme	Size (KiB)	DCACHE_DATA_ RAM_AW	DCACHE_DATA_ RAM_DW
none	8	8	64
	16	9	64
	32	10	64
	64	11	64
ecc	8	8	72
	16	9	72
	32	10	72
	64	11	72

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.12 BTB Interface Signals

BTB interface signals provide connectivity to the BTB SRAMs of the processor. These signals are always present on the process interface since BTB is always presented. BTB in AX45MP is 2-way associative. Please see Section 28 for organization of BTB SRAMs. See Table 46 for definition of `BTB_RAM_ADDR_WIDTH`.

Official Release

Table 45: BTB Memory Interface Signals

Signal Name	Direction	Description
btb0_cs	output	Chip select for BTB memory 0
btb0_we	output	Write enable for BTB memory 0
btb0_addr[6:0]	output	Address for BTB memory 0
btb0_wdata[VALEN + 23:0]	output	Write data for BTB memory 0
btb0_rdata[VALEN + 23:0]	input	Read data for BTB memory 0
btb1_cs	output	Chip select for BTB memory 1
btb1_we	output	Write enable for BTB memory 1
btb1_addr[6:0]	output	Address for BTB memory 1
btb1_wdata[VALEN + 23:0]	output	Write data for BTB memory 1
btb1_rdata[VALEN + 23:0]	input	Read data for BTB memory 1

Table 46: BTB RAM Address Bit-Width

BTB Size	BTB_RAM_ADDR_WIDTH	RAM Dimension
256	7	128 × (VALEN + 24)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.13 STLB Interface Signals

STLB interface signals provide connectivity to the STLB SRAMs of the processor. These signals are always present on the process interface but they are used only when STLB is configured. Otherwise, they should be left unconnected. STLB in AX45MP implements a 4-way set-associative structure. Please see Section 28 for organization of STLB SRAMs.

Official Release

See Table 47 and Table 48 for STLB interface signals definition when **Shared TLB Soft Error Protection** is selected to "none" or "ecc".

Table 47: STLB Memory Interface Signals Without ECC

Signal Name	Direction	Description
stlb*N*_cs	output	Chip select, *N* = 0, 1, 2, 3
stlb*N*_we	output	Write enable
stlb*N*_addr[STLB_RAM_AW-1:0]	output	Address
stlb*N*_wdata[STLB_RAM_DW-1:0]	output	Write data
stlb*N*_rdata[STLB_RAM_DW-1:0]	input	Read data

Table 48: STLB Memory Interface Signals With ECC

Signal Name	Direction	Description
STLB Tag RAM		
stlb_tag*N*_cs	output	Chip select, *N* = 0, 1, 2, 3
stlb_tag*N*_we	output	Write enable
stlb_tag*N*_addr[STLB_RAM_AW-1:0]	output	Address
stlb_tag*N*_wdata[STLB_TAG_RAM_DW-1:0]	output	Write data
stlb_tag*N*_rdata[STLB_TAG_RAM_DW-1:0]	input	Read data
STLB Data RAM		
stlb_data*N*_cs	output	Chip select, *N* = 0, 1, 2, 3
stlb_data*N*_we	output	Write enable
stlb_data*N*_addr[STLB_RAM_AW-1:0]	output	Address
stlb_data*N*_wdata[STLB_DATA_RAM_DW-1:0]	output	Write data
stlb_data*N*_rdata[STLB_DATA_RAM_DW-1:0]	input	Read data

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

See Table 49 for definition of STLB_RAM_AW.

Table 49: STLB RAM Address Bit-Width

MMU Scheme	STLB Size	STLB_RAM_AW
Sv39	32	3
	64	4
	128	5
	256	6
	512	7
Sv48	32	3
	64	4
	128	5
	256	6
	512	7

Official Release

See Table 50 for definition of STLB_RAM_DW.

Table 50: STLB RAM Data Bit-Width

MMU Scheme	STLB Size	STLB_RAM_DW
Sv39	32	30+BIU_ADDR_WIDTH
	64	29+BIU_ADDR_WIDTH
	128	28+BIU_ADDR_WIDTH
	256	27+BIU_ADDR_WIDTH
	512	26+BIU_ADDR_WIDTH
Sv48	32	39+BIU_ADDR_WIDTH
	64	38+BIU_ADDR_WIDTH
	128	37+BIU_ADDR_WIDTH
	256	36+BIU_ADDR_WIDTH
	512	35+BIU_ADDR_WIDTH

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

See Table 51 and Table 52 for definition of `STLB_TAG_RAM_DW` and `STLB_DATA_RAM_DW`.

Table 51: STLB Tag RAM Data Bit-Width

Protection Scheme	MMU Scheme	STLB Size	STLB_TAG_RAM_DW
ecc	Sv39	32	42
		64	41
		128	39
		256	38
		512	37
	Sv48	32	51
		64	50
		128	49
		256	48
		512	47

Official Release

Table 52: STLB Data RAM Data Bit-Width

Protection Scheme	Protection Width	STLB_DATA_RAM_DW
ecc	(BIU_ADDR_WIDTH-4) $\leq$ 32	BIU_ADDR_WIDTH+3
	(BIU_ADDR_WIDTH-4) > 32	BIU_ADDR_WIDTH+4

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.14 Trace Signals

AX45MP provides two options for instruction trace interface:

- 'instruction-ratified": trace interface defined in RISC-V processor trace specification version 1.0
- "instruction-gen1": Gen1 trace interface

Official Release

A trace signal might be a concatenation of NRET values of constant width. NRET is the maximum number of retired instructions, and the value is 2 for the AX45MP processor. See the tables below for the description of each signal.

Table 53: Signals for RISC-V Processor Trace Specification Version 1.0

Signal Name	Direction	Description
coreN_trace_enabled	input	This signal indicates the trace interface is enabled; this interface will not be active if this signal is not set.
coreN_trace_itype[7:0]	output	This signal indicates the termination type of the instruction block. An instruction block contains all the instructions retired in a cycle.
coreN_trace_cause[9:0]	output	This signal indicates the cause of an exception or an interrupt.
coreN_trace_tval[63:0]	output	This signal indicates the associated trap value.
coreN_trace_priv[1:0]	output	This signal indicates the privilege level of all instructions retired in this cycle.
coreN_trace_iaddr[NRET*VALEN-1:0]	output	This signal indicates the address of the first instruction retired in this block.
coreN_trace_iretire[3:0]	output	This signal indicates the number of halfwords represented by the instructions retired in this block.
coreN_trace_ilastsize[1:0]	output	This signal indicates that the size of the last retired instruction is $2^{ilastsize}$ half-words.
coreN_trace_trigger[5:0]	output	This signal indicates the trigger events. A pulse on bit 0 will cause the encoder to start tracing. A pulse on bit 1 will cause the encoder to stop tracing. A pulse on bit 2 is a trace notify event.
coreN_trace_halted	output	This signal indicates that the hart is halted.
coreN_trace_reset	output	This signal indicates that the hart is under reset.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 54: Signals for Gen1 Trace Interface

Signal Name	Direction	Description
coreN_gen1_trace_enabled	input	This signal enables the trace interface; this interface will not be active if this signal is not set.
coreN_gen1_trace_ivalid[NRET-1:0]	output	This signal indicates that an instruction has retired or caused a trap (exception).
coreN_gen1_trace_iexception[NRET-1:0]	output	This signal indicates that an instruction has caused a trap (exception).
coreN_gen1_trace_cause[NRET*10-1:0]	output	This signal indicates the cause of an exception.
coreN_gen1_trace_tval[NRET*64-1:0]	output	This signal indicates the exception data.
coreN_gen1_trace_interrupt[NRET-1:0]	output	This signal indicates that the exception is an interrupt.
coreN_gen1_trace_iaddr[NRET*VALEN-1:0]	output	This signal indicates the address of the instruction.
coreN_gen1_trace_instr[NRET*32-1:0]	output	This signal indicates the instruction content.
coreN_gen1_trace_priv[NRET*2-1:0]	output	This signal indicates the privilege mode during execution.

Official Release

Table 55: Trace Instruction Address Bit-Width

Virtual Memory Scheme	VALEN
Sv39	39
Sv48	48

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.15 ACE Signals

ACE signals are a set of signals for interfaces required by ACE custom instructions. Specifically, new interface signals will appear on the port list of AX45MP. The interface signals are listed in Table 56.

Official Release

Table 56: ACE Interface Signals

Signal Name	Direction	Description
ace_cmd_valid	output	ACE command valid from core, active HIGH.
ace_cmd_inst[31:7]	output	ACE instruction data.
ace_cmd_pc[VALEN-1:0]	output	ACE instruction PC.
ace_cmd_rs1[XLEN-1:0]	output	data of source register 1.
ace_cmd_rs2[XLEN-1:0]	output	data of source register 2.
ace_cmd_rs3[XLEN-1:0]	output	data of source register 3.
ace_cmd_rs4[XLEN-1:0]	output	data of source register 4.
ace_cmd_ready	input	ACE command ready from ace, active HIGH.
ace_acr_dirty_set	input	Set when ACE updates the ACR. Asserted for one cycle pulse.
ace_error	input	Set to trigger imprecise ACE local interrupt. Asserted for one cycle pulse.
ace_standby_ready	input	Set when ACE is ready to enter standby mode.
ace_xrf_rd1_ready	output	XRF rd1 from ACE is ready for update, active HIGH.
ace_xrf_rd1_valid	input	XRF rd1 is valid from ACE, active HIGH.
ace_xrf_rd1_index[4:0]	input	XRF rd1 update index
ace_xrf_rd1_data[XLEN-1:0]	input	XRF rd1 update data
ace_xrf_rd1_status	input	XRF rd1 status. 0: normal 1: error
ace_xrf_rd2_ready	output	XRF rd2 from ACE is ready for update, active HIGH.
ace_xrf_rd2_valid	input	XRF rd2 is valid from ACE, active HIGH.
ace_xrf_rd2_index[4:0]	input	XRF rd2 update index
ace_xrf_rd2_data[XLEN-1:0]	input	XRF rd2 update data
ace_xrf_rd2_status	input	XRF rd2 status. 0: normal 1: error
ace_interrupt	output	Pending CPU interrupt, active HIGH.
ace_sync_ack	input	Synchronization ack, active HIGH.
ace_sync_ack_status	input	Synchronization ack status. 0: synchronization is completed 1: synchronization is interrupted
ace_sync_req	output	Synchronization request, active HIGH.
ace_sync_type[31:0]	output	Synchronization type.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.16 PC/GPR/CSR Probing Signals

PC/GPR/CSR probing signals are a set of signals for probing PC of the most recently completed instruction, current GPRs values, and current CSRs value. The interface signals are listed in Table 57.

Official Release

Table 57: PC/GPR/CSR Probing Signals

Signal Name	Direction	Description
coreN_current_pc[VALEN-1:0]	output	This signal indicates the most recently completed instruction on the PC.(An instruction with exception will not update `coreN_current_pc`)
coreN_gpr_index[12:0]	input	This signal selects a general purpose register (GPR) or control and status registers (CSR) to be reflected on the `coreN_core_selected_gpr_value` output signal. When MSB is 1, GPRs are selected. When MSB is 0, CSRs are selected. See Table 58 for the address mapping.
coreN_selected_gpr_value[XLEN-1:0]	output	This is a multiplexer output signal showing that the register value is selected directly from the `coreN_gpr_index` signal without any flops.

Table 58: Index Mapping

coreN_gpr_index[12]	coreN_gpr_index[11:0]	Register to Be Probed
1	0x0 - 0x1F	GPR x0-x31
0	CSR address	see Section 21 for all CSRs address mapping

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.17 RAM Control Signals

Table 59 lists RAM control signals. These signals are present when the corresponding RAM is exist and **RAM Control In** or **RAM Control Out** is configured. See Section 2.13 for details.

Official Release

Table 59: RAM Control Signals

Signal Name	Direction	Description
BTB RAM Control Signals		
coreN_btb*M*_ctrl_in [BTB_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for BTB RAM, *M*=0, 1
coreN_btb*M*_ctrl_out [BTB_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for BTB RAM, *M*=0, 1
D-Cache RAM Control Signals		
coreN_dcache_data*M*_ctrl_in [DCACHE_DATA_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for D-Cache data RAM, *M*=0, 1, 2, 3
coreN_dcache_data*M*_ctrl_out [DCACHE_DATA_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for D-Cache data RAM, *M*=0, 1, 2, 3
coreN_dcache_tag*M*_ctrl_in [DCACHE_TAG_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for D-Cache tag RAM, *M*=0, 1, 2, 3
coreN_dcache_tag*M*_ctrl_out [DCACHE_TAG_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for D-Cache tag RAM, *M*=0, 1, 2, 3
coreN_dcache_wpt_ctrl_in [DCACHE_WPT_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for D-Cache WPT RAM
coreN_dcache_wpt_ctrl_out [DCACHE_WPT_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for D-Cache WPT RAM
DLM RAM Control Signals		
coreN_dlm_ctrl_in [DLM_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for DLM RAM 0
coreN_dlm*M*_ctrl_in [DLM_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for DLM RAM, *M*=1, 2, 3
coreN_dlm_ctrl_out [DLM_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for DLM RAM 0
coreN_dlm*M*_ctrl_out [DLM_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for DLM RAM, *M*=1, 2, 3
I-Cache RAM Control Signals		
coreN_icache_data*M*_ctrl_in [ICACHE_DATA_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for I-Cache data RAM, *M*=0, 1, 2, 3, 4, 5, 6, 7

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 59: (continued)

Signal Name	Direction	Description
coreN_icache_data*M*_ctrl_out [ICACHE_DATA_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for I-Cache data RAM, *M*=0, 1, 2, 3, 4, 5, 6, 7
coreN_icache_tag*M*_ctrl_in [ICACHE_TAG_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for I-Cache tag RAM, *M*=0, 1, 2, 3
coreN_icache_tag*M*_ctrl_out [ICACHE_TAG_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for I-Cache tag RAM, *M*=0, 1, 2, 3
ILM RAM Control Signals		
coreN_ilm0_ctrl_in [ILM_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for ILM RAM
coreN_ilm0_ctrl_out [ILM_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for ILM RAM
STLB RAM Control Signals		
coreN_stlb*M*_ctrl_in [STLB_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for STLB RAM, *M*=0, 1, 2, 3
coreN_stlb*M*_ctrl_out [STLB_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for STLB RAM, *M*=0, 1, 2, 3
coreN_stlb_tag*M*_ctrl_in [STLB_TAG_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for STLB tag RAM, *M*=0, 1, 2, 3
coreN_stlb_tag*M*_ctrl_out [STLB_TAG_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for STLB tag RAM, *M*=0, 1, 2, 3
coreN_stlb_data*M*_ctrl_in [STLB_DATA_RAM_CTRL_IN_WIDTH-1:0]	input	Control input for STLB data RAM, *M*=0, 1, 2, 3
coreN_stlb_data*M*_ctrl_out [STLB_DATA_RAM_CTRL_OUT_WIDTH-1:0]	output	Control output for STLB data RAM, *M*=0, 1, 2, 3
L2-Cache RAM Control Signals		
l2c_bank*M*_data_ram_ctrl_in [L2C_DATA_RAM_CTRL_IN_WIDTH*L2C_BANK_DATA_RAM_INSTS-1:0]	input	Control input for L2-Cache data RAMs, *M*=0, 1, 2, 3
l2c_bank*M*_data_ram_ctrl_out [L2C_DATA_RAM_CTRL_OUT_WIDTH*L2C_BANK_DATA_RAM_INSTS-1:0]	output	Control output for L2-Cache data RAMs, *M*=0, 1, 2, 3
l2c_bank*M*_tag_ram_ctrl_in [L2C_TAG_RAM_CTRL_IN_WIDTH*L2C_BANK_TAG_RAM_INSTS-1:0]	input	Control input for L2-Cache tag RAMs, *M*=0, 1, 2, 3

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 59: (continued)

Signal Name	Direction	Description
l2c_bank*M*_tag_ram_ctrl_out [L2C_TAG_RAM_CTRL_OUT_WIDTH*L2C_BANK_TAG_RAM_INSTS-1:0]	output	Control output for L2-Cache data RAMs, *M*=0, 1, 2, 3

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

3.18 DFT Signals

This section describes design for test (DFT) signals.

Table 60: DFT Signals

Official Release

Signal Name	Direction	Description
test_mode	input	Scan test mode. Internal synchronized reset signals are disabled when this signal is asserted.
scan_enable	input	Scan test enable. Internal gated clock signals are disabled when this signal is asserted.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

4 Reset and Clocking Scheme

4.1 Reset

Official Release

AX45MP uses input signal `coreN_reset_n` to reset the corresponding core clock domain. If the AXI slave port interface is configured, `coreN_slv_reset_n` is present to handle the reset sequence of the local memory interface. The reset input signals, `coreN_reset_n` and `coreN_slv_reset_n`, should be synchronized to their corresponding domain, `coreN_clk` and `coreN_slv_clk`, before connecting to AX45MP.

When **Core Interface** is configured to asynchronous, each core has a `coreN_l2_reset_n` input signal to reset logics in the core. The `coreN_l2_reset_n` should be synchronized from `coreN_reset_n` to `l2_clk` domain before connecting to AX45MP.

Besides CPU cores, level-2 modules also have their own reset signals: `l2_resetn` and `l2c_clkgen_resetn`. The `l2_resetn` and `l2c_clkgen_resetn` signal(s) should be synchronized to the `l2_clk` clock domain before connecting to AX45MP.

To maintain proper reset ordering, `coreN_reset_n`, `l2_resetn`, and `l2c_clkgen_resetn` should only be released after the release of the reset signal to the bus clock domain, even though AX45MP does not take the reset signal to the bus clock domain as its input. Figure 12 illustrates a reference design for reset synchronization.

Under typical usage, `l2_resetn` and `l2c_clkgen_resetn` can be connected to the same reset synchronizer. Some vendor-specific RAM macros need to enable the clock signal for initialization. For RAM initialization, `l2c_clkgen_resetn` can be released before `l2_resetn` to provide clock signals to L2-Cache data RAMs. See Section 4.2.4 for clock generation of L2-Cache data RAMs.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

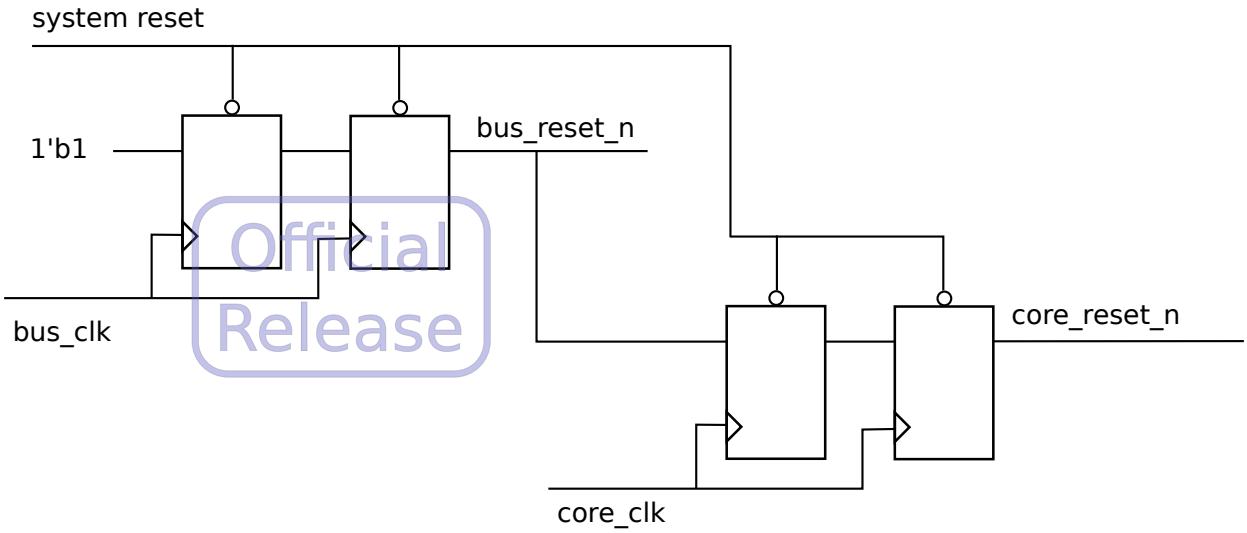


Figure 12: Reference Design for Reset Synchronization

4.2 Clock Domains

Table 61 summaries clock domains of AX45MP. Detailed clock domain constraints can be found in the synthesis script `timing_con.tcl` as described in Section 30.3.

Table 61: AX45MP Clock Domains

Clock Domain	Clock Signals
CORE_CLK	`coreN_clk`, `coreN_dcu_clk`, `coreN_lm_clk`
L2_CLK	`l2_clk`, `coreN_l2_clk`
L2C_DATA_RAM_CLK	`l2c_bankN_data_ram_clk`
BUS_CLK	`coreN_slv_clk`, `mem_aclk`, `mmio_aclk`, `iocp0_aclk`

4.2.1 BUS_CLK

The AXI interfaces signals (including MEM, MMIO, IOCP and LM Slave Port) are operated in BUS_CLK clock domain.

4.2.2 L2_CLK

Level-2 modules are in L2_CLK clock domain.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

For synchronous **Bus Clock**, L2_CLK supports integer ratios of BUS_CLK frequency, for example 1:1, 2:1, 3:1 (L2_CLK frequency:BUS_CLK frequency). Input signals `mem_aclk_en`, `mmio_aclk_en` and `iocp0_aclk_en` serve as the clock enable signals for generating the virtual BUS_CLK, and is used to determine valid cycles to sample/drive the bus interface signals. The `mem_aclk_en`, `mmio_aclk_en` and `iocp0_aclk_en` are `l2_clk` domain signals and should be asserted for one `l2_clk` cycle before the rising edge of the bus clock, as shown in Figure 13.

Official Release

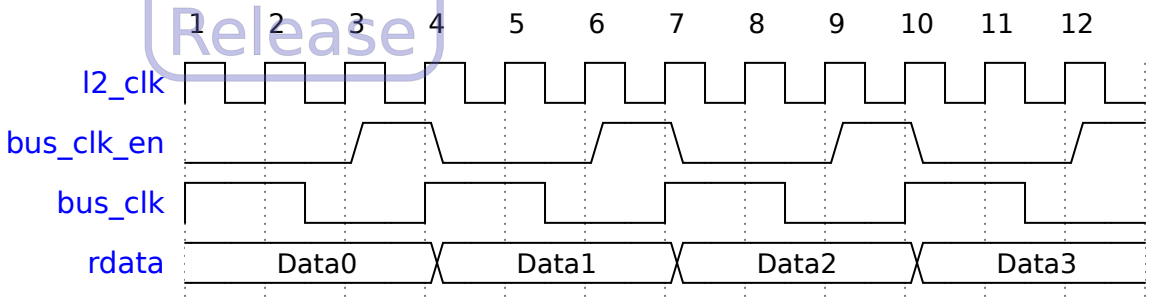


Figure 13: bus_clk_en Waveform for N:1 (3:1) Clock Ratio

For asynchronous **Bus Clock**, L2_CLK is asynchronous to BUS_CLK. Input signals `mem_aclk`, `mmio_aclk` and `iocp0_aclk` serve as the clocking signal of the bus interface.

The slave port interface signals are clocked by the input signal `coreN_slv_clk`.

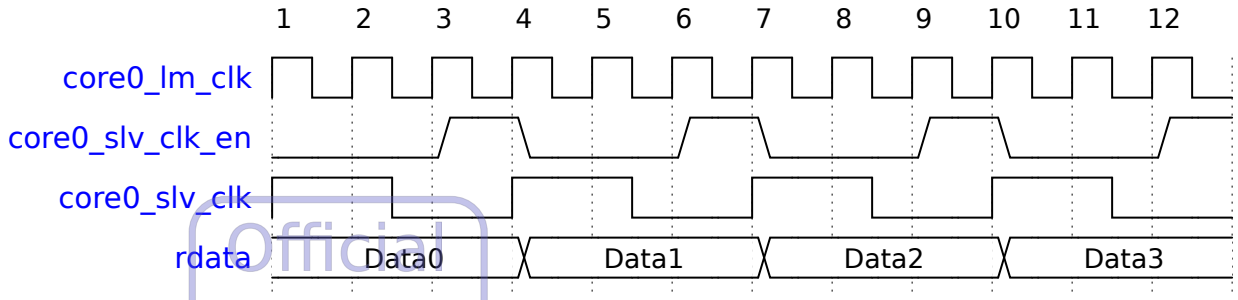
4.2.3 CORE_CLK

AX45MP provides one clock domain for each core. `coreN_clk`, `coreN_dcu_clk`, and `coreN_lm_clk` are synchronous but SoC can individually gate these clocks in different conditions.

For Low-latency and synchronous **Core Interface**, CORE_CLK and L2_CLK are synchronous.

If **Bus Clock** is also configured as synchronous, input signal `coreN_slv_clk_en` serves as the clock enable signal for generating the virtual `coreN_slv_clk` in BUS_CLK domain. `coreN_slv_clk_en` is a `coreN_lm_clk` domain signal and should be asserted for one `coreN_lm_clk` cycle before the rising edge of the bus clock, as shown in Figure 14.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



Official Release

Figure 14: slv_clk_en Waveform for N:1 (3:1) Clock Ratio

For asynchronous **Core Interface**, CORE_CLK is asynchronous to L2_CLK and BUS_CLK. The slave port interface signals are clocked by the input signal `coreN_slv_clk`.

4.2.4 L2C_DATA_RAM_CLK

The data RAMs of each L2-Cache bank are clocked by `l2c_bankN_data_ram_clk`, and their frequency is one-half of that of `l2_clk`. Figure 15 illustrates its timing diagram. The control logic part of L2-Cache are clocked by `l2_clk`, and it uses the `l2c_bankN_data_ram_clk_en` signal to synchronize with `l2c_bankN_data_ram_clk`.

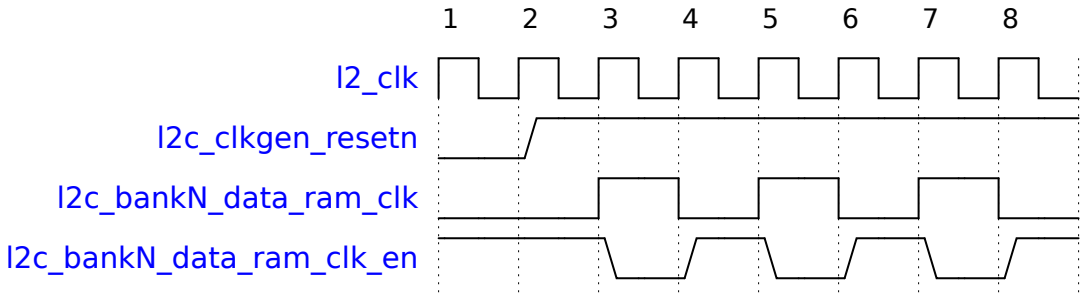


Figure 15: l2c_bankN_data_ram_clk

4.3 Race-Free Clock and Reset Generation Considerations

The RTL modeling of flops assumes zero clock-to-Q delay (no #-delay in the nonblocking assignments). Because of no #-delay flop modeling, special care must be taken to generate clocks so that there are no clock skew problems for signals that cross the two clock domains. Otherwise, simulator event ordering may cause races in simulation that are similar to hold time violations.

Either one of the rules below should be followed to avoid simulator event ordering problems crossing clock domains:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

1. All clocks are generated by blocking assignments in the same initial block.
2. When generating the bus clock by dividing CORE_CLK through flops, CORE_CLK should also be delayed through a non-blocking assignment to better align the two clock edges. See Figure 16 below for detailed information.
3. When generating the slave port clock by dividing LM_CLK through flops, LM_CLK should also be delayed through a non-blocking assignment to better align the two clock edges. See Figure 16 below for detailed information.

Official Release

These rules are also applicable to reset signals similarly.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

```
// core_clk and bus_clk clock ratio is 2:1

initial begin
        root_clk = 1'b0;
        reset_n = 1'b0;
        #(PERIOD/2) root_clk = 1'b1; #(PERIOD/2) root_clk = 1'b0;
        #(PERIOD/2) root_clk = 1'b1; #(PERIOD/2) root_clk = 1'b0;
        rstn = 1'b1;
        forever #(PERIOD/2) root_clk = ~root_clk;
end

// ---------------------------------------------------------------------
// use nonblocking assignment to align core_clk edge with bus_clk edge
// use nonblocking assignment to align lm_clk edge with slv_clk edge
// ---------------------------------------------------------------------
always @(root_clk) begin
        core_clk  <= root_clk;
        lm_clk    <= root_clk;
end

always @(root_rstn) begin
        core_rstn <= root_rstn;
        lm_rstn   <= root_rstn;
end


// divide clk by 2
always @(posedge root_clk or negedge root_rstn) begin
        if (!root_rstn) begin
                bus_clk <= 1'b1;
                slv_clk <= 1'b1;
        end
        else begin
                bus_clk <= ~bus_clk;
                slv_clk <= ~slv_clk;
        end
end

always @(posedge core_clk or negedge core_rstn) begin
        if (!core_rstn)
                bus_clk_en <= 1'b0;
        else
                bus_clk_en <= ~bus_clk_en;
end

always @(posedge lm_clk or negedge lm_rstn) begin
        if (!lm_rstn)
                slv_clk_en <= 1'b0;
        else
                slv_clk_en <= ~slv_clk_en;
end
```

Official Release

Figure 16: Race-Free CORE_CLK/BUS_CLK Generation

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

5 Instruction Set Overview

5.1 Introduction

Official Release

AX45MP implements *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA Document Version 20191213 (TD004)*. The following instruction sets are implemented:

- RV64I base integer instruction set
- RISC-V "C" standard extension
- RISC-V "M" standard extension
- RISC-V "A" standard extension
- RISC-V bit-manipulation Zba, Zbb, Zbc and Zbs extensions
- RISC-V "F" and "D" standard extensions for single/double-precision floating-point
  - FP16 half-precision floating-point extension
  - Andes BFLOAT16 Extension
- RISC-V "P" extension for DSP/SIMD instructions
- AndeStar V5 instruction extension

For detailed information, please see *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA Document Version 20191213 (TD004)* and *AndeStar V5 Instruction Extension Specification (UM165)*.

5.2 Integer Registers

Table 62 lists all general-purpose integer registers.

Table 62: Integer Registers

Register	Signal Name	Description
x0	zero	Hard-wired zero
x1	ra	Return address
x2	sp	Stack pointer
x3	gp	Global pointer
x4	tp	Thread pointer
x5	t0	Temporary/alternate link register
x6–x7	t1–t2	Temporaries

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 62: (continued)

Register	Signal Name	Description
x8	s0/fp	Saved register/frame pointer
x9	s1	Saved register
x10–x11	a0–a1	Function arguments/return values
x12–x17	a2–a7	Function arguments
x18–x27	s2–s11	Saved registers
x28–x31	t3–t6	Temporaries

Official Release

5.3 Atomic Instructions

The RVA extension includes load-reserved/store-conditional and atomic memory operation (AMO) instructions.

5.3.1 Load-Reserved/Store-Conditional Instruction

The processor tracks at most one physical address location for `LR-SC` instructions at a time. An LR instruction registers a reservation. An SC instruction succeeds only if the address of the SC instruction matches the reserved address and the reservation is still valid. The reservation is canceled under one of the following conditions:

- A load or store instruction is executed.
- An AMO instruction instruction is executed.
- `mcctlcommand` or `ucctlcommand` is written.
- A trap or an NMI is taken.
- An `FENCE.I`, `FENCE` or `SFENCE.VMA` instruction is executed.
- Entering debug mode.
- The cache line of the reservation is invalidated by coherence manager.
  - LR/SC to read-no-allocate memory (MTYP=8/10) is not supported. Hardware sets NAMO when writing 8/10 to MTYP. See Section 21.17.1.

5.3.2 Atomic Memory Operation Instruction

An atomic memory operation is expanded to `LR`-modify-`SC` sequences in the processor. The memory content is first loaded with the `LR` instruction, then the required operation is performed on the retrieved data, and the final result is written back to the memory by the `SC` instruction. If the `SC` instruction fails, the sequence will be retried until it succeeds.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

5.4 Misaligned Memory Access

AX45MP implements the misaligned memory access to support accessing misaligned addresses without triggering any Address Misaligned exceptions.

By controlling the `mmisc_ctl` CSR, the scheme can be enabled or disabled. Please see Section 21.12.7 for details.

Official Release

5.4.1 Exceptions

When the misaligned memory access scheme is enabled, Access Fault exceptions will still be triggered under the following cases:

- Accesses to device regions
- Accesses across ILM or DLM boundary
- Accesses with inconsistency PMA attributes
- Atomic accesses

If the misaligned memory access scheme is disabled, misaligned accesses will trigger Access Fault exceptions or Address Misaligned exceptions. Access fault exceptions are triggered when the following cases occur:

- Atomic accesses
- Address is located in a device region

Other misaligned accesses trigger Address Misaligned exceptions.

5.5 Floating-Point ISA Extension

AX45MP supports the “F” and “D” Standard Extensions for accelerating the performance of floating-point heavy applications. The supported configuration is indicated in the `misa` (Machine ISA) configuration register.

AX45MP supports the following FPU features:

- Fully pipelined MAC instructions
- Hardware subnormal handling
- All rounding modes

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

5.5.1 Support for Half-Precision and BFLOAT16 Formats

AX45MP also supports instructions with half-precision (FP16) data type as well as Andes extension instructions for conversion between BFLOAT16 and single-precision formats.

The support for half-precision instructions are implemented by accepting the standard RISC-V floating-point instruction formats with the width field set to "H".

Official Release

The support for conversion instructions to/from BFLOAT16 are defined in the *AndeStar V5 Instruction Extension Specification (UM165)*.

5.6 DSP ISA Extension

The processor implements the RISC-V "P" extension (draft) for DSP/SIMD ISA. The supported configuration is indicated in the `mmsc_cfg` register. With the addition of the RISC-V "P" extension (draft), the processor can run various DSP applications with lower power and higher performance.

The supported DSP features include:

- SIMD Data Processing Instructions
- Partial-SIMD Data Processing Instructions
- 64-bit Profile Instructions
- Non-SIMD Instructions
- RV64 Only Instructions
- Overflow Status Manipulation Instructions

Please see the *AndeStar V5 DSP ISA Extension Specification (UM199)* for instruction details.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

6 Branch Prediction Unit

The processor implements Branch Prediction Unit (BPU) for branch prediction in instruction fetch. BPU contains a two-way 128-entry branch target buffer (BTB), a 4-entry return address stack (RAS), and a branch history table (BHT).

Official Release

BTB is implemented to hold target addresses for unconditional jumps and conditional branches. RAS is used to keep return addresses for function calls. BHT performs the taken/not taken prediction for the conditional branches.

Branch predictions can be disabled by setting `mmisc_ctl.BRPE` to 0x0, which will make IFU fetch instructions sequentially without predictions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

7 Memory Management Unit

7.1 Introduction

Official Release

Memory Management Unit (MMU) is responsible for virtual address to physical address translation. The unit interfaces with the Instruction Fetch Control Unit (IFU) and the Load/Store Unit (LSU). An *i*TLB is implemented for IFU to speed up instruction address translation, while a *d*TLB is implemented for LSU to speed up data address translation. If the address translation information is not available in the *i*TLB or *d*TLB level, a TLB lookup request will be sent to Shared TLB (STLB), which is a bigger TLB defined in MMU. If a TLB miss happens in STLB, the hardware page table walker (PTW) will automatically traverse through page tables for the translation information.

MMU needs to be enabled and be initialized before it can be used. The default state of MMU is disabled which means there is no virtual to physical address translation. MMU is initialized through the `satp` CSR. Please see satp for details. Furthermore, `SFENCE.VMA` will be needed after `satp` is updated.

7.2 Address Translation

The virtual address to physical address translation is page based. For each virtual page, there is a page table entry (PTE) describing the physical page mapping information. Page table entries are inserted into MMU (*i*TLB/*d*TLB/STLB) by the hardware page table walker (PTW) automatically.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

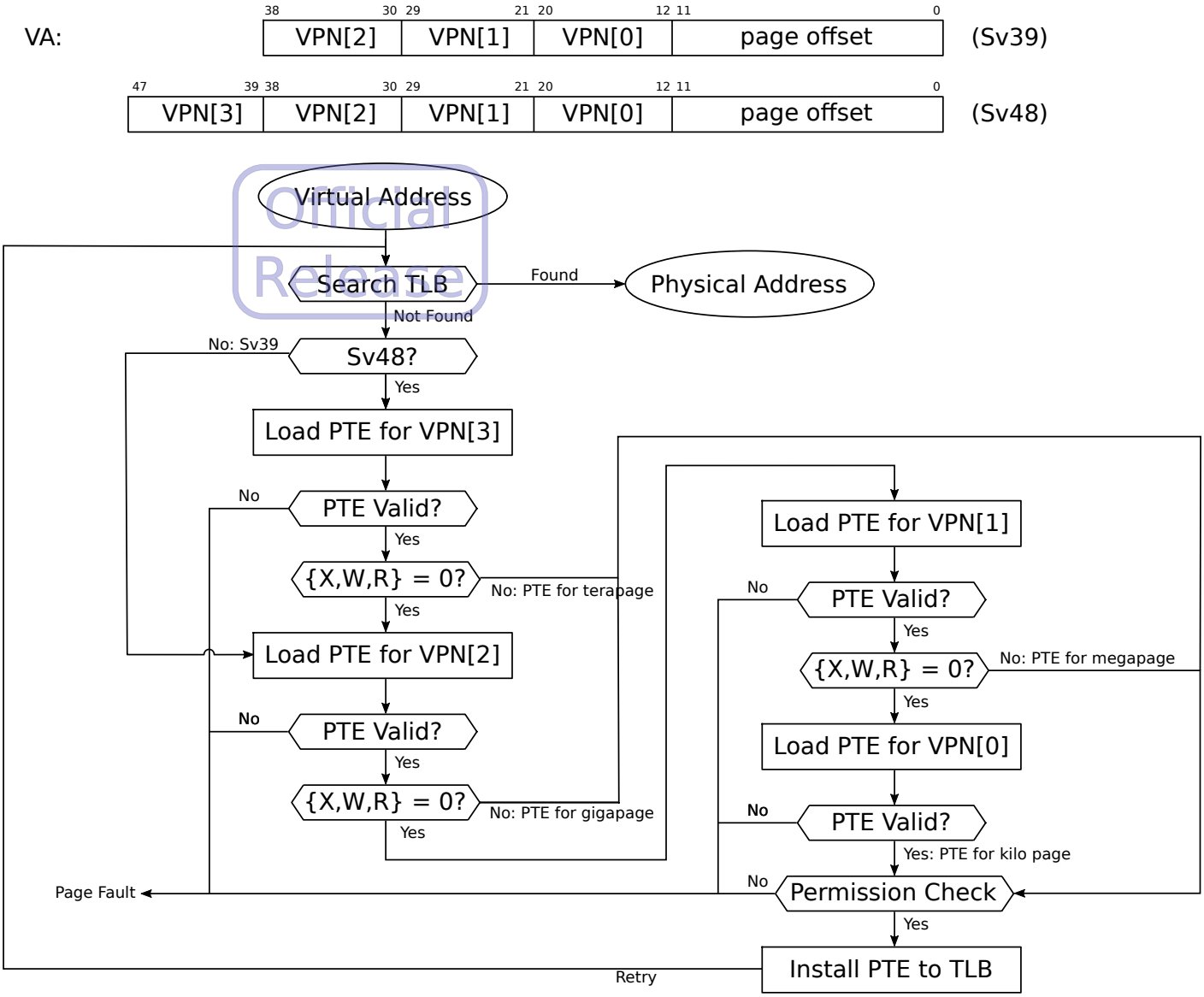


Figure 17: Virtual Address to Physical Address Translation

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

7.3 Translation Lookaside Buffer

The following sections describe the TLB operations.

7.3.1 Instruction *u*TLB (*i*TLB)

Official Release

Instruction *u*TLB(*i*TLB) is a 4/8-entry fully-associative cache that stores address translations (i.e., the Page Table Entries, PTEs) for instruction fetches. The *i*TLB gets the translation information from STLB under *i*TLB misses. `SFENCE.VMA` operations or write to satp will clear all non-global entries in *i*TLB.

7.3.2 Data *u*TLB (*d*TLB)

Data *u*TLB(*d*TLB) is a 4/8-entry fully-associative cache that stores address translations (i.e., the Page Table Entries, PTEs) for data accesses. *d*TLB gets the translation information from STLB under *d*TLB misses. `SFENCE.VMA` operations write to satp will clear all non-global entries in *d*TLB.

7.3.3 Shared TLB (STLB)

The Shared TLB (STLB) contains a 32/64/128/256/512-entry 4-way set-associative structure for 4K pages and a 4-entry fully-associative structure for superpages that store address translations (i.e., the Page Table Entries, PTEs) for both instruction fetches and data accesses.

7.3.4 Replacement Policy

TLBs (*i*TLB/*d*TLB/STLB) all implement pseudo-LRU replacement policy.

7.4 Page Table Walker (PTW)

7.4.1 Introduction

The page table walker is responsible for automatically filling STLB with PTE entries located in the system memory when there is a STLB lookup miss. Each TLB miss will require two to four memory references. The implementation caches non-leaf PTEs to speed up page table hierarchy traversal. Two sets of non-leaf PTE caches are implemented—one for misses originating from instruction fetches and the other one for misses originating from data loads/stores.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

7.4.2 Page Table Address Formation

To find the correct PTE from the page table in memory, the PTW needs to perform up to three or four memory read operations to get PTEs. The physical addresses for these read operations are formed by the PTW as follows:

Official Release

1. Let a be `satp.`$ppn \times$ PAGESIZE, and let i = LEVELS - 1. (PAGESIZE=2^{12} and for Sv39, LEVELS=3; for Sv48, LEVELS=4.)
2. The physical address for PTE of VPN[i] is $a + va.vpn[i] \times 8$. Perform a memory read to retrieve the PTE at this location. PMP checks are also applicable to memory reads accessing the PTEs. If a violation occurs, an access exception will be raised based on the access type of the instructions triggering this page table walk. Let *pte* be the value of the memory read if no access violation occurs.
3. If $pte.v = 0$, or if *(w, r)* of *pte* is *(1, 0)*, stop and raise a page-fault exception based on the access type of the instructions triggering this page table walk.
4. Otherwise, the PTE is valid.
   - If *(x, w, r)* of *pte* is *(0, 0, 0)*, this PTE is a pointer to the next level of the page table. Let $i = i$ - 1, $a = pte.ppn \times$ PAGESIZE and go to step 2 unless $i < 0$, in which case a page-fault exception should be raised based on the access type of the instructions triggering this page table walk.
   - If *(x, w, r)* of *pte* is not *(0, 0, 0)*, then the leaf PTE for the page is found.

7.5 Attributes for Address Spaces

7.5.1 Attributes for Virtual Memory Pages

Each memory page is associated with attributes controlling page accesses. These attributes are stored in the page table entries along with their physical address mappings. The following table describes the format for the page table entries.

X	W	R	Meaning
0	0	0	Pointer to next level of page table
0	0	1	Read-only page
0	1	0	Reserved for future use
0	1	1	Read-write page
1	0	0	Execute-only page
1	0	1	Read-execute page

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

X	W	R	Meaning
1	1	0	Reserved for future use
1	1	1	Read-write-execute page

Official Release

Table 63: Translated Address Space Attribute

Field	Bits	Description
PPN	53:10	Physical Page Number of the physical memory page
RSW	9:8	The RSW field is reserved for use by supervisor software.
D	7	The D bit indicates the virtual page has been written since the last time the D bit was cleared. When a virtual page is written and the D bit is clear, a page-fault exception is raised.
A	6	The A bit indicates the virtual page has been read, written, or fetched from since the last time the A bit was cleared. When a virtual page is accessed and the A bit is clear, a page-fault exception is raised.
G	5	The G bit designates a global mapping. Global mappings are those that exist in all address spaces. For non-leaf PTEs, the global setting implies that all mappings in the subsequent levels of the page table are global. Note that failing to mark a global mapping as global merely reduces performance, whereas marking a non-global mapping as global is an error.
U	4	The U bit indicates whether the page is accessible to user mode. U-mode software may only access the page when U=1. If the SUM bit in the sstatus register is set, supervisor mode software may also access pages with U=1. However, supervisor code normally operates with the SUM bit clear, in which case, supervisor code will fault on accesses to user-mode pages.
X	3	The X bit indicates whether the page is executable. Attempting to fetch an instruction from a page that does not have execute permissions raises a fetch page-fault exception.
W	2	The W bit indicates whether the page is writable. Attempting to execute a store, store-conditional (regardless of success), or AMO instruction whose effective address lies within a page without write permissions raises a store page-fault exception.
R	1	The R bit indicates whether the page is readable. Attempting to execute a load or load-reserved instruction whose effective address lies within a page without read permissions raises a load page-fault exception.
V	0	The V bit indicates whether the PTE is valid.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

8 Local Memory

8.1 Introduction

Official Release

Local memories store data or instructions that might either be accessed frequently or require deterministic access latency, such as interrupt service routines, system calls, video data, real-time systems, etc. Local memories are *memories* and accesses to them are treated the same as to the cacheable memory space. It is not suitable to map device registers in the local memories.

AX45MP supports both instruction local memory (ILM) and data local memory (DLM). They are dedicated address spaces that are independent of the memory subsystem. Accesses to them bypass the cache and memory subsystems to achieve minimal latency. The Local Memory Base Address is specified by processor configuration options described in Section 2. The details of local memory usages are described in the subsequent sections.

8.2 Local Memory Spaces

The AX45MP processor supports three address spaces: the instruction local memory, the data local memory and the system bus (AXI) address spaces. The ILM address space is defined by **ILM Size** and **ILM Base** configuration options, and the DLM address space is defined by **DLM Size** and **DLM Base** configuration options. The base address of the Andes local memory should be aligned to its size (a power-of-2 size). See Section 2 for more information regarding the configuration parameters. Any addresses outside the local memory address spaces belong to the system bus address space.

Instruction fetches go to the instruction local memory or the system bus while load/store data accesses access all three regions of spaces. The address spaces for ILM and DLM should not overlap with each other to achieve maximum compatibility across Andes processor products. The exact address space access priorities for the AX45MP processor are defined in Table 64 for instruction fetches and Table 65 for load/store data accesses.

It is not recommended to set the instruction local memory and the data local memory to have the same base address. Otherwise, UNPREDICTABLE behavior might happen.

Table 64: Priorities for Instruction Fetches

Address Hit the ILM Space	Address Hit the DLM Space	Actual Space Accessed
No	No	AXI address space
No	Yes	AXI address space

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 64: (continued)

Address Hit the ILM Space	Address Hit the DLM Space	Actual Space Accessed
Yes	No	ILM
Yes	Yes	UNPREDICTABLE (not recommended; the ILM and DLM spaces should not overlap)

Official Release

Table 65: Priorities for Data Accesses

Address Hit the ILM Space	Address Hit the DLM Space	Actual Space Accessed
No	No	AXI address space
No	Yes	DLM
Yes	No	ILM
Yes	Yes	UNPREDICTABLE (not recommended; the ILM and DLM spaces should not overlap)

8.3 Local Memory Address Range

The local memory address ranges are listed in Table 66. LM_BASE represents the base address field of the ILM and DLM local memory base address system registers (`milmb.IBPA` and `mdlmb.DBPA`).

Table 66: Local Memory Address Range (for ILM and DLM)

LM Size	Start	End
4KiB	(LM_BASE[63:12]<<12)	(LM_BASE[63:12]<<12) + 0x000000FFF
8KiB	(LM_BASE[63:13]<<13)	(LM_BASE[63:13]<<13) + 0x000001FFF
16KiB	(LM_BASE[63:14]<<14)	(LM_BASE[63:14]<<14) + 0x000003FFF
32KiB	(LM_BASE[63:15]<<15)	(LM_BASE[63:15]<<15) + 0x000007FFF
64KiB	(LM_BASE[63:16]<<16)	(LM_BASE[63:16]<<16) + 0x00000FFFF
128KiB	(LM_BASE[63:17]<<17)	(LM_BASE[63:17]<<17) + 0x00001FFFF
256KiB	(LM_BASE[63:18]<<18)	(LM_BASE[63:18]<<18) + 0x00003FFFF
512KiB	(LM_BASE[63:19]<<19)	(LM_BASE[63:19]<<19) + 0x00007FFFF
1MiB	(LM_BASE[63:20]<<20)	(LM_BASE[63:20]<<20) + 0x0000FFFFF
2MiB	(LM_BASE[63:21]<<21)	(LM_BASE[63:21]<<21) + 0x0001FFFFF
4MiB	(LM_BASE[63:22]<<22)	(LM_BASE[63:22]<<22) + 0x0003FFFFF

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 66: (continued)

LM Size	Start	End
8MiB	(LM_BASE[63:23]<<23)	(LM_BASE[63:23]<<23) + 0x0007FFFFF
16MiB	(LM_BASE[63:24]<<24)	(LM_BASE[63:24]<<24) + 0x000FFFFFF

Official Release

8.4 Local Memory Usage Constraints

Local memories are optimized for access latency. As a result, the design imposes the following usage restrictions:

- Offset addresses of VA and PA should be the same for the part of address offsets that address offset mappings for the offset part of the address that indexes into the local memory.
- Accesses to the local memory are speculative. Devices with side effects on reads should not be mapped to this region.

8.5 Multi-Bank Data Local Memory

DLM can be divided into two or four banks by specifying the **Number of DLM Banks** configuration option. Each bank applies interleaving data placement policy, and has its own read/write channel to perform read/write instructions independently.

For the two-bank DLM, bit 3 of DLM access address is used for bank selection.

- When the value of the bit is 0, the access will be performed in bank 0.
- When the value of the bit is 1, the access will be performed in bank 1.

For the four-bank DLM, LSBs (bit 3 and bit 4) of DLM access address are used for bank selection.

- When the value of two LSBs are 0, the access will be performed in bank 0.
- When the value of two LSBs are 1, the access will be performed in bank 1.
- When the value of two LSBs are 2, the access will be performed in bank 2.
- When the value of two LSBs are 3, the access will be performed in bank 3.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

9 Local Memory Slave Port

9.1 Introduction

Official Release

The LM slave port enables external bus masters to access the local memories of each core.

When an address exceeds ILM/DLM size, the higher address bits are ignored by the slave port. The `coreN_slv_awuser/coreN_slv_aruser` signal of the LM slave port interface selects which local memory to access:

Table 67: Local Memory Slave Port Selection

coreN_slv_awuser/coreN_slv_aruser	Selection
0	ILM
1	DLM

The LM slave port supports FIXED, INCR, and WRAP of AXI burst type and contains five channel FIFOs for read and write accesses. Therefore, all transfers might temporarily be stored in the FIFOs and a round-robin arbiter is used to schedule the read/write access.

Slave port accesses have a higher priority than instruction fetches and load/store operations. When the **Number of DLM Banks** option is set to two-bank or four-bank DLM, DLM might serve the slave port and load/store operations simultaneously if they access to the different banks.

Note that AX45MP does not include logics to guarantee atomicity of atomic instructions accessing the LM address space when external masters access the same location through the LM slave port, nor does it provide the protection feature on LM slave port.

9.2 Local Memory Slave Port Access Time

Table 68 and Table 69 summarize the access time of transfers accessing the LM slave port. Apply the 8-byte access time when the AXI transfer size is lower than or equal to 8-byte, otherwise apply the 16-byte access time.

Table 68: Local Memory Slave Port Read/Write Access Time (AxSIZE Is 8-Byte) (Cycle)

AXI Transfer Total Size	ILM/DLM
8-byte	8

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 68: (continued)

AXI Transfer Total Size	ILM/DLM
16-byte	9
32-byte	11
8×N-byte	7+N

Official Release

Table 69: Local Memory Slave Port Read/Write Access Time (AxSIZE Is 16-Byte) (Cycle)

AXI Transfer Total Size	ILM/DLM
16-byte	9
32-byte	11
64-byte	15
16×N-byte	7+2×N

Note

- The access time assumes that bus latency is one cycle after the request address is issued.
- The access time assumes that no core to local memory access.
- The access time assumes that the `core_clk` and `lm_clk` clock ratio is 1:1.
- The access time assumes that the latency of local memory access is 1 cycle.

9.3 Support for Soft Error Protection

The LM slave port would return bus errors to AXI masters as well as trigger local interrupts to AX45MP when a 2-bit ECC error is detected.

The behavior of ECC logic for local memories is controlled by `milmb.ECCEN/mdlmb.ECCEN`. The encoding for errors encountered through accesses from the LM slave port is summarized in the table below.

Correctable ECC errors only trigger local interrupts when `ECCEN` is equal to 3. Uncorrectable ECC errors would trigger local interrupts when `ECCEN` is equal to 2 or 3. The triggering of local interrupts is controlled by `mie.IMECCI` and the interrupt status is reported in `mip.IMECCI`. See Section 21.3.5 and Section 21.3.11 for details.

Data returned through the LM slave port is the ECC corrected version. For uncorrectable ECC errors, bus errors are reported when `ECCEN` is equal to 2 or 3.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

ECCEN	Meaning	Data Returned
0	Disable parity/ECC	Uncorrected data
1	Reserved	Reserved
2	Generate local interrupts only on uncorrectable parity/ECC errors	Corrected data or bus errors
3	Generate local interrupts on any type of parity/ECC errors	Corrected data or bus errors

Official Release

9.4 Local Memory Slave Port Operation Under WFI Mode

When one or multiple CPU cores are in WFI mode, LM data may still need to be transferred through the slave port. The accessibility of LM is controlled by `coreN_lm_clk`, which is the only clock source for the slave port. Whether `coreN_clk` is gated or not, LM is accessible when `coreN_lm_clk` is active and `coreN_slv_reset_n` is de-asserted.

AX45MP asserts `coreN_lm_local_int` to indicate that a local memory access triggers a local interrupt in the following conditions:

- An LM slave port access encounters ECC errors. See Section 9.3.
  - The error sets the interrupt pending register `mip.IMECCI`.
- ILM/DLM denies accesses from local memory store buffers with `coreN_ilm_d_denied` or `coreN_dlm_d_denied`.
  - The error sets the interrupt pending register `mip.BWEI`.

When `core_clk` is gated in the WFI mode, the interrupt pending register `mip` is not updated. The clocking gating logic should enable `coreN_clk` when `coreN_lm_local_int` is asserted.

`coreN_clk` should not be gated on assertion of lm_local_int.

To gate `coreN_lm_clk` in WFI mode as well, the following conditions should be met:

- The core is in WFI mode.
- There are no more outstanding transfers in the LM slave port.

9.5 Local Memory Initialization

In some applications, data/instructions are loaded to LM before the core fetches the first instruction. To load data/instructions, the correct de-assertion sequence of `coreN_reset_n` and `coreN_slv_reset_n` is essential.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

To allow the LM slave port to be accessible while either the core or the external bus is under reset, the reset of LM uses a merged version from coreN_reset_n and coreN_slv_reset_n. To avoid glitch on the merged reset signal, the reset sequence in system for this merged reset signal should meet the following conditions:

- coreN_slv_reset_n should be de-asserted before coreN_reset_n is de-asserted.
- coreN_reset_n and coreN_slv_reset_n never goes in opposite direction at the same time (one goes high while the other goes low).

Official Release

Between these de-assertions, data/instructions can be loaded to LM through the slave port with coreN_lm_clk being active. The length of this period depends on application requirements and is controlled by the system. coreN_reset_n and coreN_slv_reset_n are also the reset sources for local RAM modules.

The local RAM modules are in reset state when coreN_reset_n and coreN_slv_reset_n are asserted simultaneously. When one of these signals is de-asserted, local RAM modules will exit the reset state. Moreover, a synchronizer is added for generating the internal reset signal from coreN_reset_n and coreN_slv_reset_n. It leads to a 2-cycle latency for the internal reset signal to be de-asserted after any of coreN_reset_n and coreN_slv_reset_n is de-asserted.

Note

- When the LM slave port is accessing LM, the coreN_slv_reset_n should not be asserted.
- When the processor is accessing LM, the coreN_reset_n should not be asserted.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10 Level-1 Caches

10.1 Introduction

Official Release

AX45MP contains two Level-1 (L1) caches, the instruction cache and the data cache. Both can be configured to 8KiB, 16KiB, 32KiB, or 64KiB in size. The cache line size is fixed to 64 bytes.

The cache organization information can be collected from the `micm_cfg` register for the instruction cache and the `mdcm_cfg` register for the data cache. The configuration choices are listed below, and the format of the configuration registers can be found in Section 21.5.1 and Section 21.5.2.

Table 70: Configuration Choices for the I-Cache

I-Cache Size	Ways (**micm_cfg.IWAY**)	Cache lines per way (**micm_cfg.ISET**)
8 KiB	1	128
	2	64
	4	32
16 KiB	1	256
	2	128
	4	64
32 KiB	1	512
	2	256
	4	128
64 KiB	1	1024
	2	512
	4	256

Table 71: Configuration Choices for the D-Cache

D-Cache Size	Ways (**mdcm_cfg.DWAY**)	Cache lines per way (**mdcm_cfg.DSET**)
8 KiB	1	128
	2	64
	4	32

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 71: (continued)

D-Cache Size	Ways (mdcm_cfg.DWAY)	Cache lines per way (mdcm_cfg.DSET)
16 KiB	1	256
	2	128
	4	64
32 KiB	1	512
	2	256
	4	128
64 KiB	1	1024
	2	512
	4	256

Official Release

10.2 Cache Replacement Policy

The replacement policy for direct-mapped caches is irrelevant. 2-way and 4-way caches implement random or tree pseudo-LRU replacement policy.

10.3 I-Cache

I-Cache is virtually indexed and physically tagged (VIPT).

10.3.1 I-Cache Fill Operation

The instruction cache fill operation starts when a cacheable line is not in the I-Cache. After a cache miss, a burst read request for the missed cache line is issued to system bus to reduce access time for the missed cache line data. Up to two outstanding requests can be issued to system bus.

The fill operation may be aborted by system bus errors. A precise instruction access fault is triggered for the instruction fetch causing the cache miss operation if the error is on the critical word. If the error occurs on non-critical words, the fill operation will be canceled and the missed line will not be installed into I-Cache. Instruction fetches before non-critical error words will not be affected since they have received the required data.

In Debug Mode, instruction fetches will not affect I-Cache contents, and all I-Cache misses will not cause cache replacements.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.3.2 I-Cache Prefetch

After cache miss, a series of sequential cache lines will be probed to check whether they are in I-Cache. If not, streaming prefetch will be performed for these sequential cache lines. The instruction prefetch is turned off by default, and can be turned on by setting `mcache_ctl.IPREF_EN` to 0x1.

Official Release

10.3.3 I-Cache TAG SRAM Fields

Table 72: I-Cache TAG SRAM Fields

Field	Bit Position	Description
TAG	[0+:ICACHE_TAG_WIDTH]	TAG
LOCK_DUP	[ICACHE_TAG_WIDTH]	Duplicated lock bit for error protection
LOCK	[ICACHE_TAG_WIDTH+1]	Indicate the line is locked
VALID	[ICACHE_TAG_WIDTH+2]	Indicate the line is valid

Note

ICACHE_TAG_WIDTH = BIU_ADDR_WIDTH - min(12, (6 + log2(`micm_cfg.ISET`))

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.4 D-Cache

D-Cache is physically indexed and physically tagged (PIPT).

10.4.1 Cache Access Latency

Official Release

Table 73: Access Latency of the Load-to-Use Latency (Cycles) in 1/2/4 Cores Configuration

L2-Cache Tag RAM Setup Cycle	L2-Cache DATA RAM Setup Cycle	L2-Cache DATA RAM Output Cycle	D-Cache Hit	L2-Cache Hit	Modified Line in Another Core	L2-Cache Miss
1	1	2	0	11/12	24	16
1	1	3	0	12/13	24	16
1	2	2	0	13/14	24	16
1	2	3	0	14/15	24	16
2	1	2	0	12	25	17
2	1	3	0	12/13	25	17
2	2	2	0	13/14	25	17
2	2	3	0	14/15	25	17

Table 74: Access Latency of the Load-to-Use Latency (Cycles) in 8 Cores Configuration

L2-Cache Tag RAM Setup Cycle	L2-Cache DATA RAM Setup Cycle	L2-Cache DATA RAM Output Cycle	D-Cache Hit	L2-Cache Hit	Modified Line in Another Core	L2-Cache Miss
1	1	2	0	13/14	28	18
1	1	3	0	14/15	28	18
1	2	2	0	15/16	28	18
1	2	3	0	16/17	28	18
2	1	2	0	14	29	19
2	1	3	0	14/15	29	19
2	2	2	0	15/16	29	19
2	2	3	0	16/17	29	19

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

- The latency assumes that load word instructions are used to access D-Cache.
- The latency assumes that bus latency is one cycle after the request address is issued.
- The latency assumes that the dependent instruction is executed in late ALU.
- The latency assumes that the miss address is the first transfer of a cache line.
- The 0-cycle latency assumes that the dependent instruction and the load instruction are dual-issued.
- The latency assumes that the CORE_CLK and BUS_CLK clock ratio is 1:1.
- The latency assumes that the latency of memory access is 1 cycle.
- The latency assumes that the bus data width is 128-bit.
- L2-Cache hit latency is dependent on relation of L2_CLK and L2C_DATA_RAM_CLK.

Official Release

10.4.2 D-Cache Fill Operations

The D-Cache fill operation starts when a cacheable line is not in the D-Cache. A burst read request for the missed cache line is always sent first to the system bus to minimize the miss latency. The read request will be followed by a burst write request if cache eviction is required.

The fill operation may be aborted by system bus errors. A precise load access fault is triggered for the load instruction causing the cache miss operation if the bus error is for the critical word. The bus error on non-critical words does not trigger an exception, but the fill operation will be canceled and the missed line will not be installed into D-Cache. Store is always non-blocking. Bus errors on store cache miss will trigger bus-write transaction errors.

System bus errors will no longer be reported as precise exceptions if the processor is in the non-blocking mode (`mmisc_ctl.NBLD_EN`). *Bus Read/Write Transaction Error Local Interrupts* (`mip.BWEI`) will be reported instead. Please see Section 14.1 for details.

In Debug Mode, load/store instructions will minimally affect D-Cache contents. All cache misses will not cause cache replacements, and only dirty bits may be affected by accesses to cache lines that are already in D-Cache.

10.4.3 D-Cache Eviction Operations

A burst write request will be sent to L2 if a dirty line is evicted out of D-Cache. An imprecise bus-write error exception is triggered if the burst write request encounters system bus errors.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.4.4 D-Cache Write Buffers

D-Cache implements two write buffers, and each buffer is used to buffer a 64-byte line. When D-Cache evicts a line, the evicted line is temporarily buffered in one of the write buffers and then send to L2 from that write buffer.

Official Release

When the Write-Around feature is configured, store to write-no-allocate memory is also buffered in one of the write buffers. Multiple stores can be merged into a single transaction. The buffered line is flushed in any of the following conditions:

- The line is completely filled by subsequent store operations.
- A store operation writes to another line in write-no-allocate memory.
- A load operation accesses the buffered line.
- A `FENCE`, `FENCE.I`, or `SFENCE.VMA` instruction is under execution.
- A CCTL operation is under execution.
- The processor is in debug mode.
- The processor is WFI mode.
- D-Cache is disabled (`mcache_ctl.DC_EN` is cleared).

Write-no-allocate memory consists of:

- A D-Cache miss, which occurs for a store operation when D-Cache is in the write-no-allocate mode (See Section 14.2).
- PMA of the memory, which is set as (Memory, Write-back, No-allocate/Read-allocate) (MTYP=8/9).

Note

When the Write-Around feature is not configured, store to write-no-allocate memory is not buffered.

10.4.5 D-Cache TAG SRAM Fields

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 75: D-Cache TAG SRAM Fields

Field	Bit Position	Description
MESI	[2:0]	Indicate cache line states. Value / Meaning: 000 — **Invalid**. The line is not in D-Cache. 001 — **Shared**. The line is clean, and is shared by multiple caches. 011 — **Exclusive**. The cache line is present only in the current cache, but is clean. 111 — **Modified**. The line is present only in the current cache, and is dirty.
LOCK	[3]	Indicate the line is locked
TAG	[4+:DCACHE_TAG_WIDTH)]	TAG

Official Release

Note

DCACHE_TAG_WIDTH = BIU_ADDR_WIDTH - 6 - log2(`mdcm_cfg.DSET`)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.5 FENCE/FENCE.I Operations

FENCE/FENCE.I instructions may affect caches when caches are enabled. If mcache_ctl.IC_EN is 0, FENCE.I instructions will not perform any operation on the I-Cache. If mcache_ctl.DC_EN is 0, FENCE.I instructions will not perform any operation on the D-Cache. FENCE instructions do not perform any operation on either I-Cache or D-Cache. The behavior of FENCE/FENCE.I is summarized in Table 76.

Official Release

Table 76: Effects of FENCE/FENCE.I Instructions

Cache	FENCE	FENCE.I
I-Cache	None	Invalidate all cache lines
D-Cache	None	Write back all cache lines

10.6 CCTL Operations

CCTL operations provide direct control to manipulate instruction caches, data caches, or TLB tag/data RAM (cache maintenance operations). They are invoked by writing CCTL commands to the mcctlcommand CSR register. The operations can be grouped into two main types, virtual-address (VA) based or index (IX) based, and they are summarized in Table 77. These two addressing types affect how cache lines or TLB entries are specified for CCTL operations, and how the content of mcctlbeginaddr are interpreted.

Table 77: Addressing Type of CCTL Commands

Type	Usage
IX	Use the content of mcctlbeginaddr register directly as a (Way, Index), (Way, Index, Double-Word/word), or (Target, Way, Index) pair/tuple to specify a cache line in the cache or TLB entry in the TLB tag/data RAM without going through any translation mechanism. The format is defined in Table 78, Table 79, and Table 80.
VA	Use the content of mcctlbeginaddr register as a virtual address to access the cache. The virtual address has to go through the same address translation mechanism in the processor pipeline as the address of regular load/store instructions for D-Cache or instruction fetches for I-Cache. The specified operation is performed only if the addressed cache line is in the corresponding cache.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 78: Index Format for D-Cache Index Type of CCTL Operations

Field	Bit Position	Description
OFFSET	mcctlbeginaddr[A-1:3]	A=log_2(#Double-Words in a Cache Line) + 3
INDEX	mcctlbeginaddr[B-1:A]	B=log_2(Cache Size / #Ways)
WAY	mcctlbeginaddr[C-1:B]	C=Ceiling(log_2(Cache Size))

Official Release

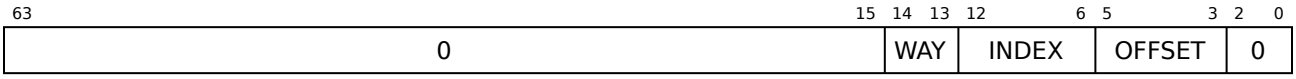
Table 79: Index Format for I-Cache Index Type of CCTL Operations

Field	Bit Position	Description
OFFSET	mcctlbeginaddr[A-1:2]	A=log_2(#Words in a Cache Line) + 2
INDEX	mcctlbeginaddr[B-1:A]	B=log_2(Cache Size / #Ways)
WAY	mcctlbeginaddr[C-1:B]	C=Ceiling(log_2(Cache Size))

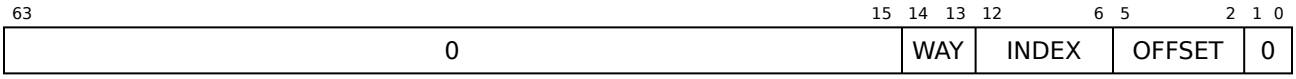
Table 80: Index Format for TLB Index Type of CCTL Operations

Field	Bit Position	Description
INDEX	mcctlbeginaddr[STLB_RAM_AW-1:0]	Index of TLB SRAM
WAY	mcctlbeginaddr[17:16]	Way of TLB SRAM
TARGET	mcctlbeginaddr[27:24]	Target of TLB SRAM

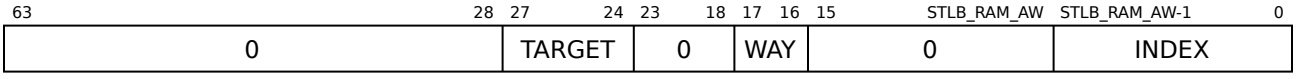
The following diagram shows an example of mcctlbeginaddr for the index type of CCTL cache operations, assuming that cache is 4-way, 32-KiB with 64-byte cache line for D-Cache.



The following diagram shows an example of mcctlbeginaddr for the index type of CCTL cache operations, assuming that cache is 4-way, 32-KiB with 64-byte cache line for I-Cache.



The following diagram shows an example of mcctlbeginaddr for the index type of CCTL TLB operations.



All available CCTL operations are summarized in Table 159. Their detailed definitions are grouped and described in the following categories.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.6.1 Invalidating Cache Blocks (L1D_VA_INVAL, L1I_VA_INVAL, L1D_IX_INVAL, L1I_IX_INVAL)

These operations invalidate the specified cache lines. Locked cache lines are unlocked and invalidated.

10.6.2 Writing Back Cache Blocks (L1D_VA_WB, L1D_IX_WB, L1D_WB_ALL)

Official Release

These operations write the data of the specified cache lines back to the system memory, if the specified cache lines are present in the cache with dirty states. The specified cache lines will still be kept in the cache and locked cache lines remain locked.

10.6.3 Writing Back & Invalidating Cache Blocks (L1D_VA_WBINVAL, L1D_IX_WBINVAL, L1D_WBINVAL_ALL)

These operations write the data of the specified cache lines back to the system memory, if the specified cache lines are present in the cache with dirty states. Then the specified cache lines will be invalidated as long as they are valid in the cache, regardless of whether their states are dirty or locked.

10.6.4 Filling and Locking Cache Blocks (L1D_VA_LOCK, L1I_VA_LOCK)

These operations lock the specified cache lines in the cache. The specified cache lines are first brought into the cache if they are not already present in the cache, then the cache lines are locked by setting their lock states. It is not an error to lock an already locked line—the same line is just locked again.

The lock state only affects the cache replacement policy. On cache miss, an unlocked lines will be replaced first. When all ways are locked, the missed line is not allocated in the cache.

The status of lock operations are written to the `mcctldata` register:

- A value of 1 indicates that the lock operation finished successfully;
- A value of 0 indicates that the lock operation aborted/failed.
  - locking an address in device or non-cacheable memory
  - locking an address in local memory (ILM/DLM)
  - locking an line when all ways are locked, and the line is not present in the cache

10.6.5 Unlocking Cache Blocks (L1D_VA_UNLOCK, L1I_VA_UNLOCK)

These operations clear the lock state of the specified cache lines if the specified cache lines are present in the cache.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.6.6 Reading Tag Data from Caches (L1D_IX_RTAG, L1I_IX_RTAG)

These operations read the contents of the tag part of the target cache line into the mcctldata register. The format of the tag data in mcctldata is defined in Section 21.12.11. The target cache line is specified in the mcctlbeginaddr register by its way and index information. Additionally, these operations observe DC_RWECC/IC_RWECC settings in mcache_ctl to read the corresponding ECC/Parity codes in the tag RAM to mecc_code.

Official Release

10.6.7 Reading Data from D-Cache (L1D_IX_RDATA)

This operation reads an 8-byte data from a cache line into the mcctldata register. The target cache line is specified in the mcctlbeginaddr register by its way, index, and double word information. Additionally, this operation observes DC_RWECC settings in mcache_ctl to read the corresponding ECC codes in the data RAM to mecc_code.

10.6.8 Reading Data from I-Caches (L1I_IX_RDATA)

This operation reads a 4-byte data from a cache line into the mcctldata register. The target cache line is specified in the mcctlbeginaddr register by its way, index, and word information. Additionally, this operation observes IC_RWECC settings in mcache_ctl to read the corresponding Parity codes in the data RAM to mecc_code.

10.6.9 Writing Tag Data to Caches (L1D_IX_WTAG, L1I_IX_WTAG)

These operations write the contents of the mcctldata register to the tag part of the target cache line. The format of the tag data in mcctldata is defined in Section 21.12.11. The target cache line is specified in the mcctlbeginaddr register by its way and index information. Additionally, these operations observe DC_RWECC/IC_RWECC settings in mcache_ctl to write the mecc_code ECC/Parity code to the corresponding tag RAM.

Note

Writing unexpected Tag data to a cache might lead to UNPREDICTABLE behavior when the cache is enable. It is recommended to clear TAG SRAM before enabling the cache.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.6.10 Writing Data to D-Cache (L1D_IX_WDATA)

This operation writes an 8-byte data in the `mcctldata` register into the target cache line. The target cache line is specified in the `mcctlbeginaddr` register by its way, index, and double word information. Additionally, this operation observes DC_RWECC settings in `mcache_ctl` to write the `mecc_code` ECC code to the corresponding data RAM.

Official Release

10.6.11 Writing Data to I-Caches (L1I_IX_WDATA)

This operation writes a 4-byte data in the `mcctldata` register into the target cache line. The target cache line is specified in the `mcctlbeginaddr` register by its way, index, and word information. Additionally, this operation observes IC_RWECC settings in `mcache_ctl` to write the `mecc_code` Parity code to the corresponding data RAM.

10.6.12 Invalidating All Cache Blocks (L1D_INVAL_ALL)

This operation invalidates all valid lines of D-Cache. Locked cache lines are unlocked and invalidated.

10.6.13 Writing Back All Cache Blocks (L1D_WB_ALL)

This operation writes the data of all dirty cache lines of D-Cache back to the system memory. All cache lines will still remain in the D-Cache and locked lines remain locked.

10.6.14 Writing Back & Invalidating All Cache Blocks (L1D_WBINVAL_ALL)

This operation writes the data of all dirty cache lines of D-Cache back to the system memory and all valid cache lines will be invalidated, including locked and/or clean cache lines.

10.6.15 Reading Tag/Data from TLB SRAMs (TLB_IX_RTAG, TLB_IX_RDATA)

These operations read the contents of the tag/data part of the target TLB entry into the `mcctldata` register. The format of the contents in `mcctldata` is defined in Section 21.12.11. The target TLB entry is specified in the `mcctlbeginaddr` register by its target, way, and index information. Additionally, these operations observe TLB_RWECC settings in `mcache_ctl` to read the corresponding ECC/Parity codes in the tag/data RAM to `mecc_code`.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

10.6.16 Writing Tag/Data to TLB SRAMs (TLB_IX_WTAG, TLB_IX_WDATA)

These operations write the contents of the `mcctldata` register to the tag/data part of the target TLB entry. The format of the contents in `mcctldata` is defined in Section 21.12.11. The target TLB entry is specified in the `mcctlbeginaddr` register by its target, way, and index information. Additionally, these operations observe TLB_RWECC settings in `mcache_ctl` to write the `mecc_code` ECC/Parity code to the corresponding tag/data RAM.

Official Release

10.7 Supervisor/User CCTL Operations

CCTL operations are available to Supervisor/User-mode software under the control of the `mcache_ctl.CCTL_SUEN` control bit. These operations are triggered in both modes by accessing `ucctlbeginaddr`, `ucctlcommand` and `scctldata` registers, while `mcache_ctl.CCTL_SUEN` controls access permission to these registers. When `CCTL_SUEN` is 0, accessing them in Supervisor and User mode would cause illegal instruction exceptions. It should be set to 1 to enable Supervisor (and User) CCTL operations.

All CCTL operations can be made available to Supervisor-mode software while only four CCTL operations listed in the table below are available to User-mode software.

Table 81: User CCTL Operations

Value		Command	Type	Exception Entry
0	0b00_000	L1D_VA_INVAL	VA	Store related Fault
1	0b00_001	L1D_VA_WB	VA	Store related Fault
2	0b00_010	L1D_VA_WBINVAL	VA	Store related Fault
8	0b01_000	L1I_VA_INVAL	VA	Store related Fault

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11 Level-2 Cache

11.1 Introduction

Official Release

A level-2 cache (L2C) improves the system performance by providing larger amount of cache line entries and reasonable access delays. Its features include:

- Inclusive policy. Any line in D-Cache has a copy in L2-Cache.
- 64 bytes cache line size
- 16-way set-associative with pseudo-random replacement policy
- Support ECC protection on L2C tag and data RAMs
- Multi-bank structure of L2C for improving physical implementation
- Configurable number of RAM setup/output cycles
- Hardware stride prefetcher for data access and instruction fetch
- Cache control (CCTL) operations for cache maintenance
- Way allocation mask for each core and IOCP

11.2 L2-Cache Multi-Bank Structure

Figure 18 is block diagram of L2-Cache. L2-Cache is partitioned into two or four banks for improving bandwidth and frequency by option **Number of L2C Banks**. These banks can be accessed in parallel. Each bank has data, and tag RAMs, and associated logic. The L2-Cache interfaces are summarized in Table 82.

Table 82: L2-Cache Interfaces

Interface	Number of L2C Banks Is 2	Number of L2C Banks Is 4
us_bank0	memory accesses whose address[6] is 0x0	memory accesses whose address[7:6] is 0x0
us_bank1	memory accesses whose address[6] is 0x1	memory accesses whose address[7:6] is 0x1
us_bank2	n.a.	memory accesses whose address[7:6] is 0x2
us_bank3	n.a.	memory accesses whose address[7:6] is 0x3
reg	L2-Cache register accesses	L2-Cache register accesses
ds	memory accesses to L3 memory subsystem	memory accesses to L3 memory subsystem

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

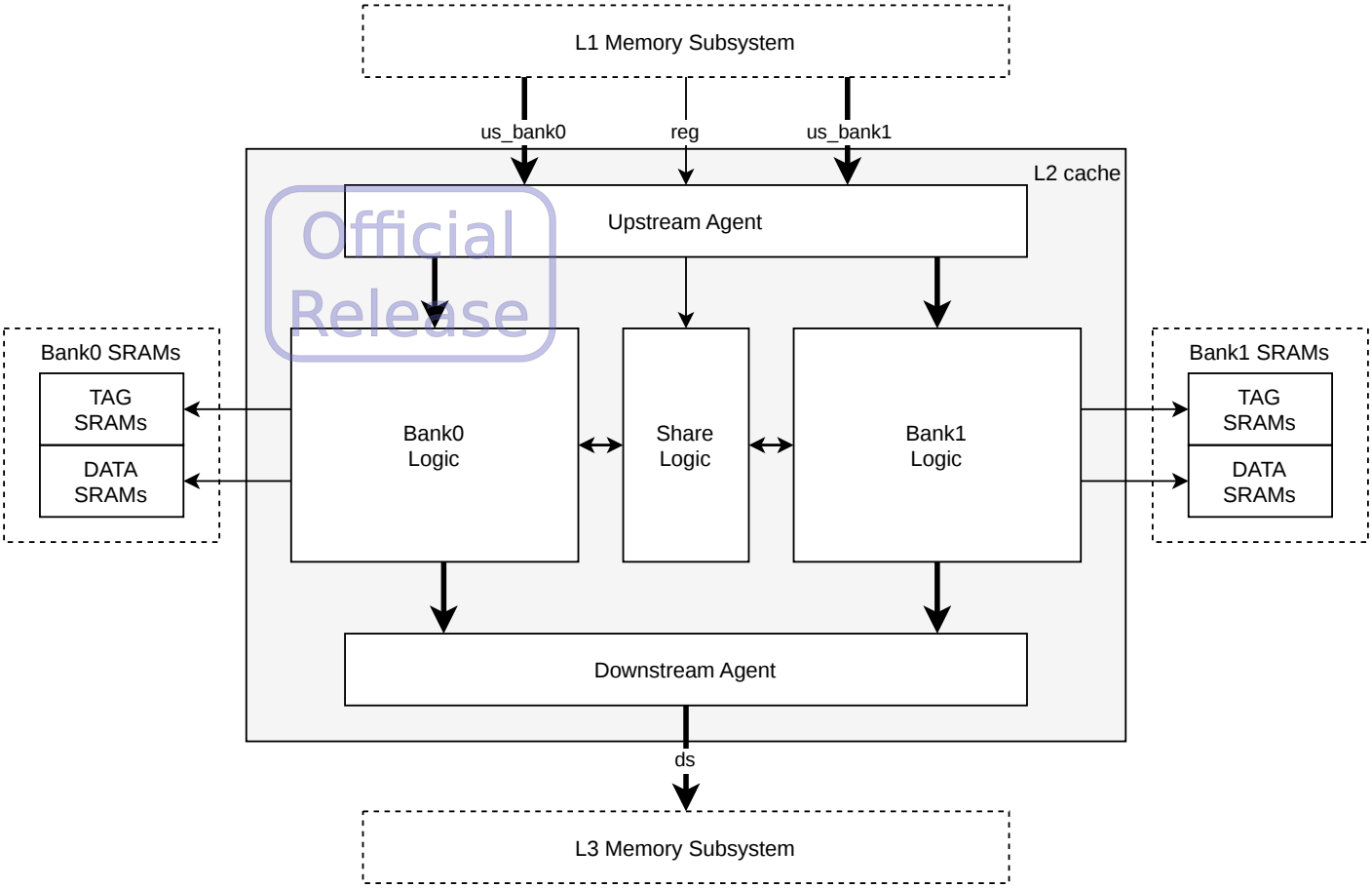


Figure 18: L2-Cache Block Diagram

11.3 L2-Cache Prefetch

L2C implements a hardware prefetch engine, and the key features of the L2-Cache prefetch include:

- Non-unit stride detection for data access stream within a 4K page
- Consecutive instruction fetch within a 4K page
- Programmable prefetch depth
  - 0, 1, 2, 3 prefetch depth for instruction fetch
  - 0, 2, 4, 8 prefetch depth for data access
- Programmable threshold of available handling register count for prefetch throttling

11.4 L2-Cache Control Operation

L2C provides a set of cache control (CCTL) operations, which are divided into four types as below:

- IX type (index)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- PA type (physical address)
- TGT type (target)
- Flush type

Table 83 lists the supported L2C CCTL operations. These CCTL operations are performed/controlled by writing the related CCTL control registers. The status of CCTL operations is reported in the CCTL Status Register.

Official Release

11.4.1 L2-Cache Control Operation

Table 83: Supported L2C CCTL Operations

OPCODE	Mnemonics	Type	Operation
0b00_000	L2_IX_INVAL	IX	Invalidate an L2-Cache entry
0b00_001	L2_IX_WB	IX	Write back an L2-Cache entry
0b00_010	L2_IX_WBINVAL	IX	Write back and invalidate an L2-Cache entry
0b01_000	L2_PA_INVAL	PA	Invalidate an L2-Cache entry
0b01_001	L2_PA_WB	PA	Write back an L2-Cache entry
0b01_010	L2_PA_WBINVAL	PA	Write back and invalidate an L2-Cache entry
0b10_000	L2_TGT_WRITE	TGT	Write an L2-Cache entry
0b10_001	L2_TGT_READ	TGT	Read an L2-Cache entry
0b10_010	L2_WBINVAL_ALL	Flush	Write back and invalidate all L2-Cache entries (Cache flush)

`L2_TGT_WRITE` and `L2_TGT_READ` are used to directly write and read the raw data of a tag or data RAM entry. If L2C is configured to support ECC, the ECC code of the RAM entry will also be written to or read through the `TGT_ECC` register. For raw data, this means the ECC checking will not be performed during the CCTL operation when ECC is enabled. However, if `L2_TGT_WRITE` is used to write a corrupted ECC code to a tag or a valid data RAM entry, the ECC error will be detected when this entry is later used by normal cache operations while ECC is enabled.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.4.2 Cache Line State Transition

The four types of L2C CCTL operations have distinct behaviors. `IX`, `PA`, and `Flush` types always access the L2C tag SRAMs to determine the line state in the D-Cache, regardless of whether the L2-Cache is enabled. When data coherence handling is necessary, the L2-Cache will write back or invalidate the copies in the D-Cache. The `TGT` type of L2C CCTL operations directly access the L2-Cache RAMs and bypass coherence management logic. Table 84 lists the line state transition of D-Cache and L2-Cache during an L2C CCTL operation.

Official Release

All the L2C CCTL operations do not affect any content in the I-Cache.

Table 84: Cache Line State Transition of L2C CCTL

Operation	Cache Line State Before Operation		Cache Line State After Operation		Transaction to L3 Memory
	D-Cache	L2-Cache	D-Cache	L2-Cache	
`L2_IX_WB`, `L2_PA_WB`	Invalid	Invalid	Invalid	Invalid	None
	Invalid	Clean	Invalid	Clean	None
	Shared	Clean	Shared	Clean	None
	Exclusive	Clean	Shared	Clean	None
	Modified	Clean	Shared	Clean	Write back the dirty data
	Invalid	Dirty	Invalid	Clean	Write back the dirty data
	Shared	Dirty	Shared	Clean	Write back the dirty data
	Exclusive	Dirty	Shared	Clean	Write back the dirty data
	Modified	Dirty	Shared	Clean	Write back the dirty data
`L2_IX_INVAL`, `L2_PA_INVAL`	Invalid, Shared, Exclusive or Modified	Invalid, Clean or Dirty	Invalid	Invalid	None
`L2_IX_WBINVAL`, `L2_PA_WBINVAL`, `L2_WBINVAL_ALL`	Invalid	Invalid	Invalid	Invalid	None
	Invalid	Clean	Invalid	Invalid	None
	Shared	Clean	Invalid	Invalid	None
	Exclusive	Clean	Invalid	Invalid	None
	Modified	Clean	Invalid	Invalid	Write back the dirty data
	Invalid	Dirty	Invalid	Invalid	Write back the dirty data
	Shared	Dirty	Invalid	Invalid	Write back the dirty data
	Exclusive	Dirty	Invalid	Invalid	Write back the dirty data
	Modified	Dirty	Invalid	Invalid	Write back the dirty data

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.5 L2-Cache Way Allocation

When a cache miss occurs, the missed line is allocated to a way. L2-Cache uses a pseudo-random replacement policy and way mask registers to allocate the line. Each core and IOCP is in an individual domain, and each domain has a programmable way mask register to specify which ways that can be allocated or be replaced by the access from the master in the corresponding domain. The domain numbers are listed in Table 85.

Official Release

Table 85: Allocation Domain Assignment

Domain Number	1 Core + IOCP	2 Cores + IOCP	4 Cores + IOCP	8 Cores + IOCP
0	Core 0 Fetch, Load/Store	Core 0 Fetch, Load/Store	Core 0 Fetch, Load/Store	Core 0 Fetch, Load/Store
1	IOCP	Core 1 Fetch, Load/Store	Core 1 Fetch, Load/Store	Core 1 Fetch, Load/Store
2		IOCP	Core 2 Fetch, Load/Store	Core 2 Fetch, Load/Store
3			Core 3 Fetch, Load/Store	Core 3 Fetch, Load/Store
4			IOCP	Core 4 Fetch, Load/Store
5				Core 5 Fetch, Load/Store
6				Core 6 Fetch, Load/Store
7				Core 7 Fetch, Load/Store
8				IOCP

11.6 L2-Cache Error Handling

There are two sources of L2-Cache requests:

- Upstream masters: the requests include core fetch, core load/store, and IOCP accesses.
- L2-Cache controller: the requests include prefetch, CCTL operations, and cache line write-back to L3.

When handling a request, L2-Cache might encounter any of the following errors:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Correctable RAM errors. These errors are silently corrected without logging.
- Uncorrectable RAM errors
- Bus errors
- Cache coherency protocol errors
- WAYMASK for a request is zero.

Official Release

Errors can be synchronous or asynchronous. Synchronous errors are reported in the upstream response of a upstream request, while asynchronous errors are reported through `l2c_err_int`.

A core ignores errors of speculative accesses (e.g. instruction prefetch and data prefetch). For demanded accesses, the following errors are reported by traps:

- Load access faults
- Store/AMO access faults
- Instruction access faults
- Local interrupts (mie.BWEI)

IOCP reports SLVERR or DECERR errors through `iocp0_bresp/iocp0_rresp`.

Asynchronous errors include:

- CCTL operation encounters uncorrectable RAM errors (RAM error).
- CCTL operation encounters bus errors (Bus Error).
- CCTL operation probes D-Cache when D-Cache coherency is disabled (Probe error).
- D-Cache writes back a line that is not in L2-Cache (Release error).
- L2-Cache sends a L3 request to write back a line, and the response has bus errors (Bus error).

When an asynchronous error occurs, the Asynchronous Error Register and Error Register are updated. The error status can be cleared by writing any value to the corresponding field of Asynchronous Error Register.

11.7 L2-Cache Way Prediction

L2-Cache is 16-way set-associative. For reducing latency and power, L2-Cache accesses all tag SRAMs and the data SRAM of a predicted way in parallel. If the prediction is correct, L2-Cache returns data without accessing data SRAMs of other ways. If the prediction is incorrect, L2-Cache accesses the data SRAM of the correct way.

The way prediction mechanism uses way prediction tables (WPT). The Way prediction table is an SRAM in D-Cache of each core. When D-Cache miss, the SRAM is read to predict the way. Then, D-Cache sends a request to access L2-Cache as well as the predicted way. The way prediction table is updated when a new line is filled into D-Cache.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

For instruction fetch and IOCP requests, the way prediction mechanism is not applied. L2-Cache accesses all tag SRAMs, and then accesses the data SRAM of the correct way.

11.8 L2-Cache Programming Guide

Official Release

11.8.1 L2-Cache Initialization

When L2-Cache exits the reset state, L2-cache TAG RAMs are initialized. The requests to L2-cache are blocked until the initialization is done. The `INITSTATUS` field of the L2C control register shows whether the initialization is done. To disable the initialization, assert the l2c_disable_init signal.

11.8.2 L2-Cache Enablement

The enablement of L2-Cache is controlled by the `CEN` field of the L2C control register. L2-Cache implements an inclusive policy and should be enabled before D-Cache is enabled. L2-Cache is enabled by default, and can be disabled by clearing the `CEN` field.

11.8.3 L2-Cache Registers

L2C controller registers are mapped to a 64KiB or 1MiB memory space which is decided by L2C Register Base and L2C Register Space Size options.

Bus transactions to this region should follow the rules below:

- a single transfer
- the access size smaller than 8 bytes
- the cacheability is Device

The result of using other burst types of transactions to access this region is UNPREDICTABLE.

11.8.4 Register Type

Term	Description
IM	Implementation dependent/determined
RO	Read-Only register/field. Any software write to RO registers/fields will be silently ignored by hardware.
RW	Read/Write register/field

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Term	Description
W1C	Write-One-Clear register/field. When written 1, the corresponding bit of the register/field is cleared to 0.
WC	Write-Clear register/field. When written any value, the register/field is cleared to 0.

Official Release

11.8.5 Summary of Registers

The memory map of the registers is specified by **Number of Processor Cores**.

- For 1/2/4 cores configuration, the summary of registers is shown in Table 86.
- For 8 cores configuration, the summary of registers is shown in Table 87.

Table 86: L2C Register Summary for 1/2/4 Cores Configuration

Address Offset	Description	Section
0x0000 – 0x0007	Configuration Register	Section 11.8.6
0x0008 – 0x000f	Control Register	Section 11.8.7
0x0010 – 0x0017	HPM Control Register 0	Section 11.8.8
0x0030 – 0x0037	Asynchronous Error Register	Section 11.8.9
0x0038 – 0x003f	Error Register	Section 11.8.10
0x0040 – 0x0047	CCTL Command Register 0	Section 11.8.11
0x0048 – 0x004f	CCTL Access Line Register 0	Section 11.8.12
0x0050 – 0x0057	CCTL Command Register 1	Section 11.8.11
0x0058 – 0x005f	CCTL Access Line Register 1	Section 11.8.12
0x0060 – 0x0067	CCTL Command Register 2	Section 11.8.11
0x0068 – 0x006f	CCTL Access Line Register 2	Section 11.8.12
0x0070 – 0x0077	CCTL Command Register 3	Section 11.8.11
0x0078 – 0x007f	CCTL Access Line Register 3	Section 11.8.12
0x0080 – 0x0087	CCTL Status Register	Section 11.8.13
0x0090 – 0x00c8	CCTL TGT Data Register 0 to 7	Section 11.8.14
0x00d0 – 0x00d7	CCTL TGT ECC Code Register	Section 11.8.15
0x0200 – 0x0207	HPM Counter Register 0	Section 11.8.16
0x0208 – 0x020f	HPM Counter Register 1	Section 11.8.16
0x0300 – 0x0307	Way Allocation Mask Register 0	Section 11.8.17
0x0308 – 0x030f	Way Allocation Mask Register 1	Section 11.8.17
0x0310 – 0x0317	Way Allocation Mask Register 2	Section 11.8.17
0x0318 – 0x031f	Way Allocation Mask Register 3	Section 11.8.17

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 86: (continued)

Address Offset	Description	Section
0x0320 – 0x0328	Way Allocation Mask Register 4	Section 11.8.17

Official Release

Table 87: L2C Register Summary for 8 Cores Configuration

Address Offset	Description	Section
0x0000 – 0x0007	Configuration Register	Section 11.8.6
0x0008 – 0x000f	Control Register	Section 11.8.7
0x0010 – 0x0017	HPM Control Register 0	Section 11.8.8
0x0030 – 0x0037	Asynchronous Error Register	Section 11.8.9
0x0038 – 0x003f	Error Register	Section 11.8.10
0x0040 – 0x0047	CCTL Command Register 0	Section 11.8.11
0x0048 – 0x004f	CCTL Access Line Register 0	Section 11.8.12
0x0080 – 0x0087	CCTL Status Register 0	Section 11.8.13
0x0090 – 0x00c8	CCTL TGT Data Register 0 to 7	Section 11.8.14
0x00d0 – 0x00d7	CCTL TGT ECC Code Register	Section 11.8.15
0x0200 – 0x0207	HPM Counter Register 0	Section 11.8.16
0x0208 – 0x020f	HPM Counter Register 1	Section 11.8.16
0x0300 – 0x0307	Way Allocation Mask Register 0	Section 11.8.17
0x0308 – 0x030f	Way Allocation Mask Register 1	Section 11.8.17
0x0310 – 0x0317	Way Allocation Mask Register 2	Section 11.8.17
0x0318 – 0x031f	Way Allocation Mask Register 3	Section 11.8.17
0x0320 – 0x0327	Way Allocation Mask Register 4	Section 11.8.17
0x0328 – 0x032f	Way Allocation Mask Register 5	Section 11.8.17
0x0330 – 0x0337	Way Allocation Mask Register 6	Section 11.8.17
0x0338 – 0x033f	Way Allocation Mask Register 7	Section 11.8.17
0x0340 – 0x0327	Way Allocation Mask Register 8	Section 11.8.17
0x1040 – 0x1047	CCTL Command Register 1	Section 11.8.11
0x1048 – 0x104f	CCTL Access Line Register 1	Section 11.8.12
0x1080 – 0x1087	CCTL Status Register 1	Section 11.8.13
0x2040 – 0x2047	CCTL Command Register 2	Section 11.8.11
0x2048 – 0x204f	CCTL Access Line Register 2	Section 11.8.12
0x2080 – 0x2087	CCTL Status Register 2	Section 11.8.13
0x3040 – 0x3047	CCTL Command Register 3	Section 11.8.11
0x3048 – 0x304f	CCTL Access Line Register 3	Section 11.8.12

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 87: (continued)

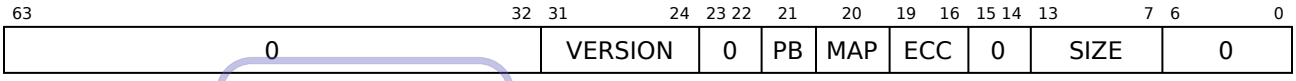
Address Offset	Description	Section
0x3080 – 0x3087	CCTL Status Register 3	Section 11.8.13
0x4040 – 0x4047	CCTL Command Register 4	Section 11.8.11
0x4048 – 0x404f	CCTL Access Line Register 4	Section 11.8.12
0x4080 – 0x4087	CCTL Status Register 4	Section 11.8.13
0x5040 – 0x5047	CCTL Command Register 5	Section 11.8.11
0x5048 – 0x504f	CCTL Access Line Register 5	Section 11.8.12
0x5080 – 0x5087	CCTL Status Register 5	Section 11.8.13
0x6040 – 0x6047	CCTL Command Register 6	Section 11.8.11
0x6048 – 0x604f	CCTL Access Line Register 6	Section 11.8.12
0x6080 – 0x6087	CCTL Status Register 6	Section 11.8.13
0x7040 – 0x7047	CCTL Command Register 7	Section 11.8.11
0x7048 – 0x704f	CCTL Access Line Register 7	Section 11.8.12
0x7080 – 0x7087	CCTL Status Register 7	Section 11.8.13

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.6 Configuration Register

Offset: 0x0000



Official Release

This register indicates the cache size, ECC type, and the version of the L2C implementation.

Field Name	Bits	Description	Type	Reset
SIZE	[13:7]	L2C cache size in KiB	RO	IM

Value	Size
0x00	0 KiB
0x01	128 KiB
0x02	256 KiB
0x04	512 KiB
0x08	1 MiB
0x10	2 MiB
0x20	4 MiB
0x40	8 MiB
Others	Reserved

Field Name	Bits	Description	Type	Reset
ECC	[19:16]	L2C ECC type	RO	IM

Value	ECC Type
0x0	No protection
0x1	One-bit error correction and two-bit error detection
Others	Reserved

Field Name	Bits	Description	Type	Reset
MAP	[20]	Memory Map	RO	IM

Value	Meaning
0x00	V0 map for 1/2/4 cores configuration
0x01	V1 map for 8 cores configuration

Field Name	Bits	Description	Type	Reset
PB	[21]	Prefetch Buffer Functionality	RO	IM

Value	Meaning
0x0	Prefetch Buffer Absent
0x1	Prefetch Buffer Present

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

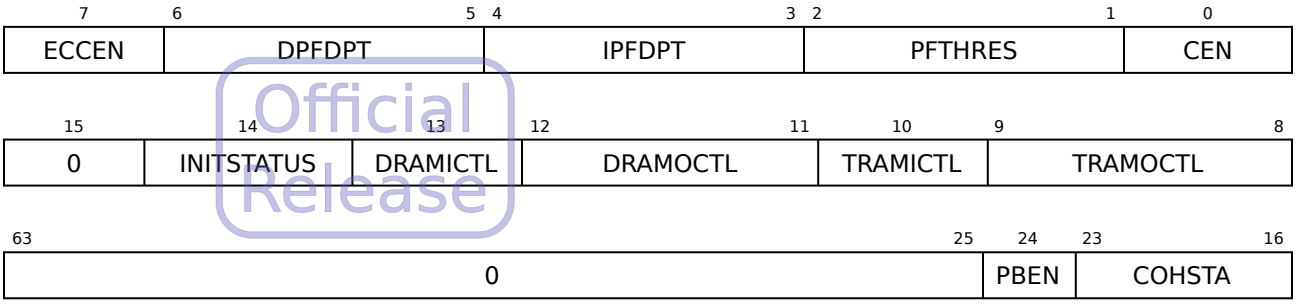
Field Name	Bits	Description	Type	Reset
VERSION	[31:24]	Version	RO	0x10

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.7 Control Register

Offset: 0x0008



Official Release

This register controls various functions of L2C.

Field Name	Bits	Description	Type	Reset
CEN	[0]	L2-Cache enable **Value** / **Meaning** 0 — Disable the L2-Cache 1 — Enable the L2-Cache	RW	1
PFTHRES	[2:1]	Prefetch threshold value. To avoid occupying too many L2C handling registers, prefetch can be throttled based on the occupancy status of handling registers. **Value** / **Meaning** 0 — Stop L2C prefetch when the number of busy handling registers exceeds 10. 1 — Stop L2C prefetch when the number of busy handling registers exceeds 8. 2 — Stop L2C prefetch when the number of busy handling registers exceeds 6. 3 — Never stop L2C prefetch regardless of the number of busy handling registers.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IPFDPT	[4:3]	Instruction prefetch depth. Indicates the depth of L2 instruction prefetch. It is the number of prefetch requests ahead of the demand request stream.	RW	0

Value	Meaning
0	0 request (disable L2 instruction prefetch)
1	1 request
2	2 requests
3	3 requests

Official Release

Field Name	Bits	Description	Type	Reset
DPFDPT	[6:5]	Data prefetch depth. Indicates the depth of L2 data prefetch. It is the number of prefetch requests ahead of the demand request stream.	RW	0

Value	Meaning
0	0 request (disable L2 data prefetch)
1	2 requests
2	4 requests
3	8 requests

Field Name	Bits	Description	Type	Reset
ECCEN	[7]	ECC enable	RW	0

Value	Meaning
0	Disable L2 ECC
1	Enable L2 ECC

Note

This field is only meaningful when the **L2C ECC** configuration option is set to “ecc”.

Field Name	Bits	Description	Type	Reset
TRAMOCTL	[9:8]	Tag RAM output cycle	RO	IM

Value	Meaning
0	1 cycle
Others	Reserved

Continued on next page. . .

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

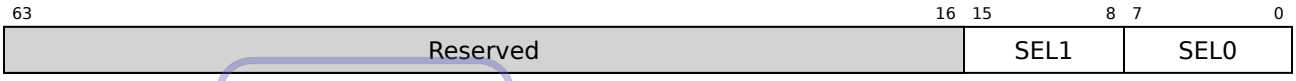
Field Name	Bits	Description	Type	Reset
TRAMICTL	[10]	Tag RAM setup cycle **Value** / **Meaning** 0 / 1 cycle 1 / 2 cycles	RO	IM
DRAMOCTL	[12:11]	Data RAM output cycle **Value** / **Meaning** 1 / 2 cycles 2 / 3 cycles Others / Reserved	RO	IM
DRAMICTL	[13]	Data RAM setup cycle **Value** / **Meaning** 0 / 1 cycle 1 / 2 cycles	RO	IM
INITSTATUS	[14]	Self-initialization status **Value** / **Meaning** 0 / Initialization done 1 / Under initializing	RO	1
COHSTA	[23:16]	Coherent status	RO	0
PBEN	[24]	Prefetch Buffer Enable. When this field is enabled, and a cache line is fetched from a 128-byte aligned address, its subsequent cache line will be prefetched from the sequential address. **Value** / **Meaning** 0 / Disable 1 / Enable	RW	IM

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.8 HPM Control Register 0

Offset: 0x0010



Official Release

This register selects events to monitor for performance counters. It is only present when the number of configured L2C Performance Counter is greater than 0.

Field Name	Bits	Description	Type	Reset
SEL0	[7:0]	Monitored event selection of performance counter 0	RW	0xff
SEL1	[15:8]	Monitored event selection of performance counter 1	RW	0xff

Table 88 lists the supported events of performance counters.

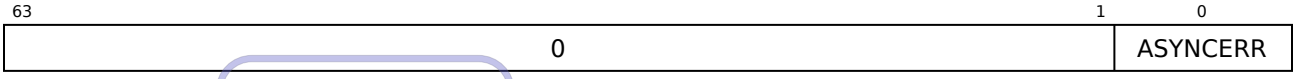
Table 88: Monitored Event Definitions of the L2C Performance Counter

Select	Event Description
0x00	Total access count
0x01	L2-Cache access count
0x02	L2-Cache miss count
0x03–0xff	Reserved

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.9 Asynchronous Error Register

Offset: 0x0030



Official Release

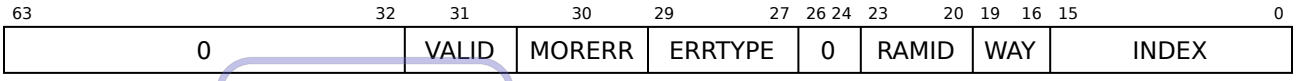
This register is write-clear and indicates the occurrence of asynchronous errors of L2C.

Field Name	Bits	Description	Type	Reset
ASYNCERR	[0]	Indicates if any asynchronous error has happened	WC	0x0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.10 Error Register

Offset: 0x0038



Official Release

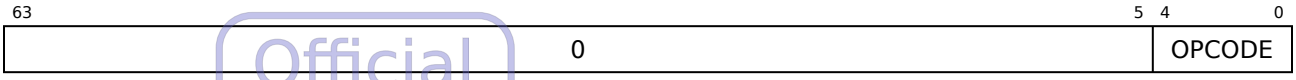
This register holds information of the first asynchronous error. If more than one errors occur, the MORERR bit is set. The INDEX and WAY fields are only meaningful when the first error is RAM error. Otherwise, the INDEX and WAY fields are "Don't Care". Writing any value to this register clears the whole register to zero.

Field Name	Bits	Description	Type	Reset
INDEX	[15:0]	Index address of the first error	WC	0
WAY	[19:16]	Way of the RAM, where the first error occurred	WC	0
RAMID	[23:20]	RAM ID, where the first error occurred **Value** / **Description** 0b0000 / L2 tag RAM 0b0001 / L2 data RAM Others / Reserved	WC	0
ERRTYPE	[29:27]	The first error type **Value** / **Description** 0b000 / RAM error 0b001 / Release error 0b010 / Probe error 0b100 / Bus error Others / Reserved	WC	0
MORERR	[30]	More error indicator, being set when more errors reported.	WC	0
VALID	[31]	Valid bit, set to 1 when an error is reported.	WC	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.11 CCTL Command Registers

Offset: 0x0040, 0x0050, 0x0060, 0x0070, 0x1040, 0x2040, 0x3040, 0x4040, 0x5040, 0x6040, 0x7040



Official Release

Writing to CCTL Command Register *n* will trigger a CCTL operation, with the operation specified by the value written and the line specified by CCTL Access Line Register *n*. Each register corresponds to one CPU core so each CPU core should only use its respective registers. If CPU core *n* does not exist, CCTL Command Register *n* is reserved. L2C internally uses the round-robin strategy to decide the run order when two or more CCTL Command Registers are written at the same time.

Field Name	Bits	Description	Type	Reset
OPCODE	[4:0]	CCTL operation	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

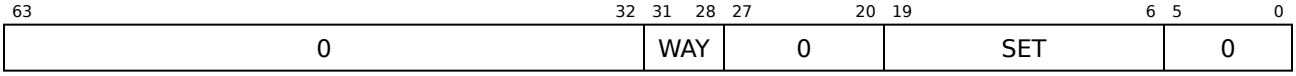
11.8.12 CCTL Access Line Registers

Offset: 0x0048, 0x0058, 0x0068, 0x0078, 0x1048, 0x2048, 0x3048, 0x4048, 0x5048, 0x6048, 0x7048

These registers are used to specify the physical address, set, way, and/or RAM ID for CCTL operations. Each register corresponds to one CPU core so each CPU core should only use its respective register. If CPU core *n* does not exist, CCTL Access Line Register *n* is reserved.

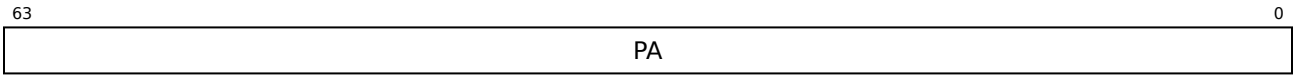
Official Release

The interpretation of these registers is different based on the CCTL operation type. The register format for IX-type CCTL operations is:



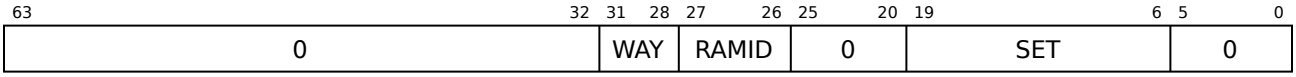
Field Name	Bits	Description	Type	Reset
SET	[19:6]	Cache set of the CCTL operation	RW	0
WAY	[31:28]	Cache way of the CCTL operation	RW	0

The register format for PA-type CCTL operations is:



Field Name	Bits	Description	Type	Reset
PA	[63:0]	Physical address of the CCTL operation	RW	0

The register format for TGT-type CCTL operations is:



Field Name	Bits	Description	Type	Reset
SET	[19:6]	Cache set of the CCTL operation	RW	0
RAMID	[27:26]	RAM ID of the CCTL operation	RW	0
WAY	[31:28]	Cache way of the CCTL operation	RW	0

Value	Meaning
0b00	L2 tag RAM
0b01	L2 data RAM

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

As for the flush type, these registers are not used.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

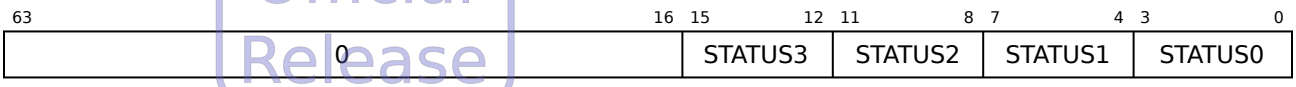
11.8.13 CCTL Status Register

Offset: 0x0080, 0x1080, 0x2080, 0x3080, 0x4080, 0x5080, 0x6080, 0x7080

This register provides the status of CCTL operations.

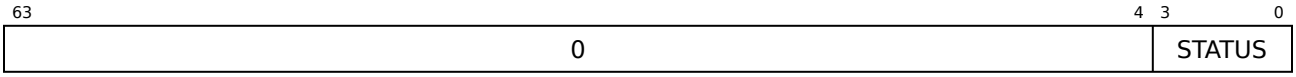
The register format for CCTL Status Register of 1/2/4 Cores Configuration is:

Official Release



Field Name	Bits	Description	Type	Reset
STATUS0	[3:0]	CCTL status 0 Value / Meaning: 0b0000 – Idle 0b0001 – A CCTL operation is running 0b0010 – An illegal CCTL operation was performed Others – Reserved	RO	0
STATUS1	[7:4]	CCTL status 1	RO	0
STATUS2	[11:8]	CCTL status 2	RO	0
STATUS3	[15:12]	CCTL status 3	RO	0

The register format for CCTL Status Register of 8 Cores Configuration is:

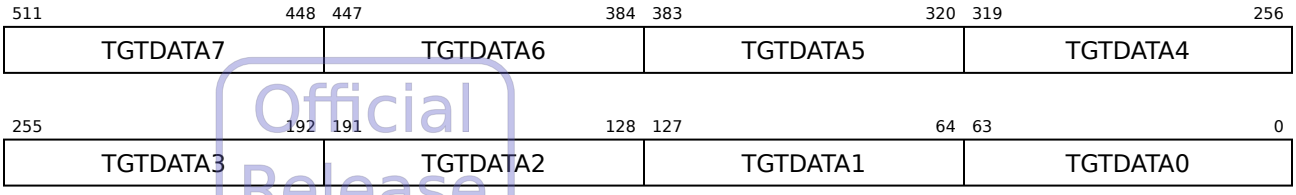


Field Name	Bits	Description	Type	Reset
STATUS	[3:0]	CCTL status of the corresponding CCTL command Value / Meaning: 0b0000 – Idle 0b0001 – A CCTL operation is running 0b0010 – An illegal CCTL operation was performed Others – Reserved	RO	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.14 CCTL TGT Data Registers 0 – 7

Offset: 0x0090 to 0x00c8

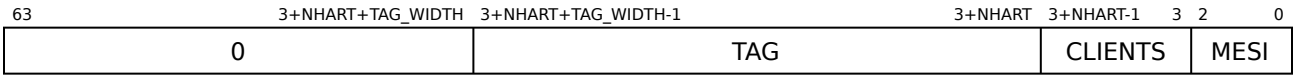


Official Release

These registers provide 512-bit write or read data for the `L2_TGT_WRITE` or `L2_TGT_READ` CCTL operation. They consist of eight 64-bit registers which are read/writable as below:

Offset	Description
0x90	CCTL TGT Data Register 0 (bits [63:0])
0x98	CCTL TGT Data Register 1 (bits [127:64])
0xa0	CCTL TGT Data Register 2 (bits [191:128])
0xa8	CCTL TGT Data Register 3 (bits [255:192])
0xb0	CCTL TGT Data Register 4 (bits [319:256])
0xb8	CCTL TGT Data Register 5 (bits [383:320])
0xc0	CCTL TGT Data Register 6 (bits [447:384])
0xc8	CCTL TGT Data Register 7 (bits [511:448])

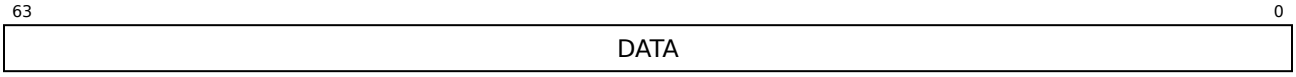
The interpretation of these registers is different based on the TGT operation. The register format for `L2_TGT_READ/L2_TGT_WRITE` accessing tag RAMs is:



Field Name	Bits	Description	Type	Reset
MESI	[2:0]	MESI state of the tag RAM entry Value / Meaning: 0b000 – Invalid 0b011 – Clean 0b111 – Dirty Others – Reserved	RW	0
CLIENTS	[3+:NHART]	Each bit indicates a core	RW	0
TAG	[(3+NHART)+: TAG_WIDTH]	Tag	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The register format for L2_TGT_READ/L2_TGT_WRITE accessing data RAMs is:



Field Name	Bits	Description	Type	Reset
DATA	[63:0]	Read data from data RAM / Write data to data RAM	RW	0

Official Release

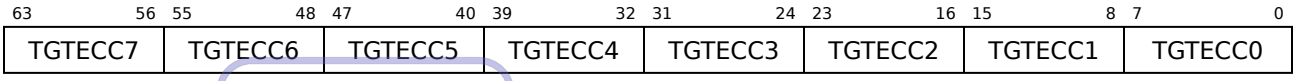
If an L2_TGT_WRITE operation is applied to the data RAM, the corresponding line of the data RAM is filled with the content of CCTL Data Registers 0 – 7. Otherwise, the corresponding line of the tag RAM is filled.

If an L2_TGT_READ operation is applied to the data RAM, CCTL Data Registers 0 – 7 are filled with the corresponding line of the data RAM. Otherwise, they are filled with the corresponding line of the tag RAM.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.15 CCTL TGT ECC Code Register

Offset: 0x00d0



Official Release

This register is used to provide ECC codes accompanying the write or read data when the `L2_TGT_WRITE` or `L2_TGT_READ` CCTL operation is performed.

Field Name	Bits	Description	Type	Reset
TGTECC0	[7:0]	ECC code of CCTL TGT data 0	RW	0
TGTECC1	[15:8]	ECC code of CCTL TGT data 1	RW	0
TGTECC2	[23:16]	ECC code of CCTL TGT data 2	RW	0
TGTECC3	[31:24]	ECC code of CCTL TGT data 3	RW	0
TGTECC4	[39:32]	ECC code of CCTL TGT data 4	RW	0
TGTECC5	[47:40]	ECC code of CCTL TGT data 5	RW	0
TGTECC6	[55:48]	ECC code of CCTL TGT data 6	RW	0
TGTECC7	[63:56]	ECC code of CCTL TGT data 7	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

11.8.16 HPM Counter Registers 0–1

Offset: 0x0200 and 0x0208

Official Release

These read/writable registers provide the counter values of monitored events. The **Number of L2C Performance Counters** configuration option determines how many of them are present.

Offset	Description
0x0200	Performance Counter 0
0x0208	Performance Counter 1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

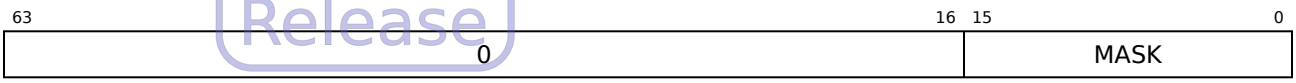
11.8.17 Way Allocation Mask

Offset: 0x0300, 0x0308, 0x0310, 0x0318, 0x0320, 0x0328, 0x0330, 0x0338, 0x0340

A WAYMASK register controls when a request from a domain can be allocated in ways. A domain is used to classify requests.

Official Release

The detail of domain classification is shown in Section 11.5.



For example, if bit-2 in WAYMASK1 is set, the request from domain 1 can be allocated to way-2.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

12 L1 Memory Error Protection

AX45MP provides ECC (error correction code) and parity protection schemes to protect RAMs against soft errors. Soft errors are caused by some external particles that temporarily corrupts value stored in the RAMs, and can be recovered or detected with different protection schemes on the RAMs. Supported protection schemes include:

Official Release

- Parity for I-Cache: Single Error Detection (SED),
- ECC for I-Cache: Double Error Detection (DED), and
- ECC for ILM, DLM, and D-Cache: Single Error Correct and Double Error Detection (SEC-DED).

Table 89: Memory Protection Types

RAM	Protection Scheme	Protection Granularity	Behavior
ILM	ECC	64 bits	See Section 12.2
DLM	ECC	64 bits	See Section 12.2
I-Cache tag	Parity	8 bits	See Section 12.3
I-Cache data	Parity	8 bits	See Section 12.3
I-Cache tag	ECC	32 bits/64 bits	See Section 12.3
I-Cache data	ECC	32 bits	See Section 12.3
D-Cache tag	ECC	64 bits	See Section 12.4
D-Cache data	ECC	64 bits	See Section 12.4
D-Cache WPT	Way Resolution	All bits	See Section 12.4
BTB	Branch Resolution	All bits	See Section 12.5
STLB tag	ECC	32 bits/64 bits	See Section 12.6
STLB data	ECC	32 bits/64 bits	See Section 12.6

Note

There are two possibilities of ECC Protection Granularity for I-Cache tag, STLB tag, and STLB data:

- 32 bits when the SRAM data width is equal to or less than 39 bits, or
- 64 bits when the SRAM data width is larger than 39 bits.

12.1 Parity/ECC Control Mode

`milmb.ECCEN`, `mdlmb.ECCEN`, `mcache_ctl.IC_ECCEN`, `mcache_ctl.DC_ECCEN`, and `mcache_ctl.TLB_ECCEN` CSR fields control memory protection modes for ILM, DLM, I-Cache, D-Cache, and STLB respectively. The memory protection modes include:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Disable parity/ECC.
- Generate exceptions on uncorrectable parity/ECC errors.
- Generate exceptions on all parity/ECC errors.

See the following subsections for detailed information on each mode.

Official Release

12.1.1 Disable Parity/ECC

The behavior of the parity/ECC logic in the "Disable parity/ECC" mode is:

- The logic disables parity/ECC checking, but still updates the newly-generated parity/ECC code into the RAM for each write access.
- The received data is always uncorrected.
- The processor uses read-modify-writes to store a partial double word.
- If any parity/ECC error happens,
  - No exception will be generated.
  - The error will not be corrected.

12.1.2 Generate Exceptions on Uncorrectable Parity/ECC Errors

The behavior of the parity/ECC logic in the "Generate exceptions on uncorrectable parity/ECC errors" mode is:

- The logic enables parity/ECC checking and updates the newly-generated parity/ECC code into the RAM for each write access.
- Corrected data is always given.
- The processor uses read-modify-writes to store a partial double word.
- No exceptions will be generated if correctable errors are detected, and those correctable errors will be recovered.
- An exception will still be generated if an uncorrectable error is detected.
- For speculative accesses, e.g. instruction prefetch and data prefetch,
  - Uncorrectable errors does not generate exceptions
  - Correctable errors are corrected

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

12.1.3 Generate Exceptions on Parity/ECC Errors

The behavior of the parity/ECC logic in the "Generate exceptions on parity/ECC errors" mode is:

- The logic enables parity/ECC checking and updates the newly-generated parity/ECC code into the RAM for each write access.
- Corrected data is always given.
- The processor uses read-modify-writes to store a partial double word.
- For correctable Parity/ECC errors, exceptions are generated, and the corrupted data in the RAM will be corrected before generating exceptions.
- For uncorrectable Parity/ECC errors, exceptions are generated.
- For speculative accesses, e.g. instruction prefetch and data prefetch,
  - Uncorrectable and correctable errors does not generate exceptions
  - Correctable errors are corrected

Official Release

12.2 Local Memory Protection

SECDED is applied to ILM and DLM when ILM or DLM Soft Error Protection is specified to ECC. One-bit ECC errors found on ILM/DLM access are correctable errors, while two-bit ECC errors found on load or store instructions on ILM/DLM are uncorrectable errors. For ILM/DLM ECC error exceptions, `mecc_code` will be updated with the ECC error information.

12.3 I-Cache Protection

SED and DED are applied to I-Cache when I-Cache Soft Error Protection is specified to Parity and ECC respectively. Data in I-Cache are always clean since there are no store accesses to I-Cache. If a parity or ECC error is found, I-Cache can get the correct copy from the next-level memory system. Therefore, if one-bit parity errors or one-bit/two-bit ECC errors are found on I-Cache accesses, they will be correctable errors.

In order to check the soft-error on lock bit, another lock mirror bit is implemented. If a parity or ECC error occurs on locked cache line, or the lock bit and the lock mirror bit are mismatched, that parity or ECC error will be an uncorrectable error.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

A correctable error will always be corrected after the error is found, even though the error will cause an exception. Error Cache data of an uncorrectable error will be kept in I-Cache without correction. On an I-Cache parity/ECC error exceptions, `mecc_code` will be updated with the parity/ECC error information. If lock bit information is needed , user can get the lock bit and the lock mirror bit by the CCTL command, L1I_IX_RTAG.

Official Release

12.4 D-Cache Protection

SEC-DED is applied to D-Cache tag and data SRAMs when D-Cache Soft Error Protection is specified to ECC. Correctable errors include:

- A cache line with a one-bit error on tag or data SRAMs is correctable, and the hardware will automatically repair the error.
- A clean line with a two-bit error on data SRAMs is also correctable, and the hardware will invalidate the line.
  - If `mcache_ctl.DC_ECCEN` is 2, the line is refilled from the next-level memory system.
  - If `mcache_ctl.DC_ECCEN` is 3, an exception is raised without refilling the line.

An access to a cache line with a two-bit error on tag SRAMs is uncorrectable. An access to a modified line with a two-bit error on data SRAMs is also uncorrectable. For D-Cache ECC error exceptions, `mecc_code` will be updated with the ECC error information.

When D-Cache write-back encounters ECC errors on data SRAMs, the following operations are applied:

- The errors are trapped by imprecise exceptions or local interrupts.
  - The `mecc_code.CODE` field is always written to 0.
- Correctable ECC errors are repaired and written to the next-level memory.
- Uncorrectable ECC errors are written to the next-level memory.

D-Cache WPT RAMs save the predicted way of L2-Cache lines for reducing the hit latency of L2-Cache. For a D-Cache miss request, L2-Cache accesses all ways of TAG RAMs and the predicted way of DATA RAMs. If the predicted way is correct, the data can be returned to D-Cache. If the predicted way is incorrect, L2-Cache access the correct way of DATA RAMs. The L2-Cache way resolution logic can tolerate errors in D-Cache WPT RAMs.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

12.5 BTB Protection

BTB RAMs save predicted branch target addresses for improving the throughput of the instruction fetch. When a branch instruction is resolved, the branch resolution logic compares the correct branch target with the predicted target address. If a mismatch is found, the logic updates the BTB entry and fetches instructions from the correct target address.

Official Release

12.6 STLB Protection

SEC and DED are applied to STLB tag and data SRAMs when STLB Soft Error Protection is specified to ECC.

Table 90 summaries STLB memory protection. The hardware operations are:

- For instruction fetch, load, store, or speculative accesses with `mcache_ctl.TLB_ECCEN` as 2,
  - repair the entry when it is a one-bit error.
  - invalidate the entry and be refilled from the next-level memory system when it is a two-bit error.
- For instruction fetch, load, or store with `mcache_ctl.TLB_ECCEN` as 3,
  - repair the entry and an exception is raised when it is a one-bit error.
  - invalidate the entry and an exception is raised without refilling the entry when it is a two-bit error.
- For speculative accesses with `mcache_ctl.TLB_ECCEN` as 3,
  - repair the entry when it is a one-bit error.
  - invalidate the entry when it is a two-bit error.
- For `SFENCE.VMA` with `mcache_ctl.TLB_ECCEN` as 2 or 3, invalidate the entry when it is a one-bit or two-bit error.

Table 90: STLB Memory Protection

Case	mcache_ctl.TLB_ECCEN	1-Bit Error	2-Bit Error	Exception
Instruction Fetch/Load/Store	2	Repair	Invalidate and refill	No
	3	Repair	Invalidate	Yes
Speculative accesses	2	Repair	Invalidate and refill	No
	3	Repair	Invalidate	No
SFENCE.VMA	2 or 3	Invalidate	Invalidate	No

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

12.7 Soft Error Injection

12.7.1 ILM/DLM ECC Error Injection

Use the following steps to inject ECC errors on ILM/DLM for implementing the ECC exception handler.

Official Release

- Specify ILM/DLM to "Disable ECC" mode.
- Turn on `milmb.RWECC` or `mdlmb.RWECC` for ILM/DLM respectively.
- Perform a load instruction on an address in ILM/DLM region. Then, `mecc_code.CODE` will contain the ECC code of the read data.
- Flip one or two bits in `mecc_code.CODE` to inject one-bit or two-bit ECC errors.
- Perform a store instruction to the address in ILM/DLM region. Then, the ECC code with error will be written to ILM/DLM.
- Turn off `milmb.RWECC` or `mdlmb.RWECC` for ILM/DLM respectively.
- Specify ILM/DLM to "Generate Exceptions on Uncorrectable Parity/ECC Errors" mode or "Generate Exceptions on parity/ECC Errors" mode.
- Perform a load instruction to the address to trigger the ECC error.

12.7.2 I-Cache Parity/ECC Error Injection

Use the following steps to inject ECC errors on I-Cache for implementing the parity/ECC exception handler.

- Specify I-Cache to "Disable ECC" mode.
- Turn on `mcache_ctl.IC_RWECC`.
- Perform L1I_IX_RTAG or L1I_IX_RDATA CCTL command to a specific address to read parity/ECC data from tag RAM or data RAM respectively.
- Flip one bit in `mecc_code.CODE` or `mcctldata` to inject one bit parity/ECC error or flip the lock duplicate in `mcctldata` to generate a lock mismatch error.
- Perform L1I_IX_WTAG or L1I_IX_WDATA CCTL command to the specific address to inject a parity/ECC error.
- Turn off `mcache_ctl.IC_RWECC`.
- Specify I-Cache to "Generate Exceptions on Uncorrectable Parity/ECC Errors" mode or "Generate Exceptions on parity/ECC Errors" mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Perform a fetch to that specific address to trigger the ECC error.

12.7.3 D-Cache ECC Error Injection

Official Release

Use the following steps to inject ECC errors on D-Cache for implementing the ECC exception handler.

- Specify D-Cache to "Disable ECC" mode.
- Turn on `mcache_ctl.DC_RWECC`.
- Perform L1D_IX_RTAG or L1D_IX_RDATA CCTL command to a specific address to read ECC data from tag RAM or data RAM respectively.
- Flip one bit or two bits in `mecc_code.CODE` and `mcctldata` to inject one-bit or two-bit ECC errors.
- Perform L1D_IX_WTAG or L1D_IX_WDATA CCTL command to the specific address to inject an ECC error.
- Turn off `mcache_ctl.DC_RWECC`.
- Specify D-Cache to "Generate Exceptions on Uncorrectable Parity/ECC Errors" mode or "Generate Exceptions on parity/ECC Errors" mode.
- Perform load/store fetch to that specific address to trigger the ECC error.

12.7.4 STLB ECC Error Injection

Use the following steps to inject ECC errors on Shard TLB SRAMs for implementing the ECC exception handler.

- Specify `mcache_ctl.TLB_ECCEN` to "Disable ECC" mode.
- Turn on `mcache_ctl.TLB_RWECC`.
- Perform TLB_IX_RTAG or TLB_IX_RDATA CCTL command to a specific address to read ECC data from tag RAM or data RAM respectively.
- Flip one bit or two bits in `mecc_code.CODE` and `mcctldata` to inject one-bit or two-bit ECC errors.
- Perform TLB_IX_WTAG or TLB_IX_WDATA CCTL command to the specific address to inject an ECC error.
- Turn off `mcache_ctl.TLB_RWECC`.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Specify `mcache_ctl.TLB_ECCEN` to “Generate Exceptions on Uncorrectable Parity/ECC Errors” mode or “Generate Exceptions on parity/ECC Errors” mode.
- Perform a fetch or load/store to that specific address to trigger the ECC error.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

13 Physical Memory Attributes

13.1 Introduction

Official Release

Memory locations can have various attributes associated with them, and are basically categorized into either one of the two types: device region and memory region. While memory regions may be cacheable or non-cacheable locations, device regions are non-cacheable locations where accesses to these locations may cause side effects. AX45MP uses MTYP (memory type) to define the cacheability and idempotency of memory regions. Please see descriptions of the MTYP field in Section 21.17.1 for possible values of MTYP.

AX45MP provides two mechanisms for physical memory attributes:

- Static physical memory attributes
- Programmable physical memory attributes

13.2 Static Physical Memory Attributes

Up to 8 device regions could be statically configured in the processor through the Device Region configuration options. The memory type attributes for device regions are (device, non-bufferable).

Device regions and ILM/DLM should not overlap with each other. The behavior is UNDEFINED when they overlap. See Section 2.12 for how to configure static device regions.

13.3 Programmable Physical Memory Attributes

Programmable PMA allows dynamic adjustment of memory attributes in the runtime. It contains a configurable amount of PMA entries implemented as CSR registers (see Section 2.5.5) to control the attributes of memory locations in interest. If the settings in those entries conflict with the static Device Region settings, PMA entries will have higher priorities.

PMA entries themselves are statically prioritized. The lowest-numbered PMA entry that matches any physical address (PA) of the access determines the attribute type and whether to support AMO instructions. If no PMA entries match the address, the attribute type is determined by the statically configured PMA. See Section 21.17.1 for more information about PMA entries.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

13.4 Memory Access Ordering

Accesses to device regions are strongly-ordered. They are guaranteed to be non-speculative and issued in program order. An access to a device region is not issued until all preceding accesses to device regions are finished.

Official Release

On the other hand, accesses to the memory regions could be speculative and the order of accessing memory regions is not guaranteed. A load access to a cacheable memory region might bypass an earlier store access if there is no data dependency. In such a scenario, explicit `FENCE` instructions are required to guarantee the order.

Table 91 shows ordering of two instructions A and B, where A < B (A comes earlier than B) in program order.

Table 91: Memory Access Ordering for CPU Revision 20.0.0 and Later

A<B in Program Order	B		
A	Cacheable Memory	Non-Cacheable Memory	Device
Cacheable Memory	-	-	-
Non-Cacheable Memory	-	-	-
Device	-	-	<

13.5 Non-Cacheable Memory and Device Accesses

Accessing to non-cacheable memory and device will bypass I-Cache, D-Cache and L2-Cache no matter the data is cached or not. The cache states are not changed by these accesses.

Note

For CPU revisions before 10.0.0, instruction fetch will access I-Cache when data in non-cacheable memory and device regions are cached into I-Cache. Under normal usages, data in non-cacheable memory and device regions are not cached unless PMA is changed from cacheable memory to non-cacheable memory or device regions.

13.6 Cacheable Memory Accesses to L1-Caches

This section describes operations to I-Cache and D-Cache.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

13.6.1 Write-Back

Table 92: Behaviors upon Accessing Write-Back and No-Allocate Regions (MTYP=8)

Scenario	Operations
Read-Hit	Read from L1
Write-Hit	Write to L1 Change the line state to modified
Read-Miss	Read from L2
Write-Miss	Write to L2

Official Release

Table 93: Behaviors upon Accessing Write-Back and Read-Allocate Regions (MTYP=9)

Scenario	Operations
Read-Hit	Read from L1
Write-Hit	Write to L1 Change the line state to modified
Read-Miss	Read from L2 Allocate the line in L1 with exclusive/shared state
Write-Miss	Write to L2

Table 94: Behaviors upon Accessing Write-Back and Write-Allocate Regions (MTYP=10)

Scenario	Operations
Read-Hit	Read from L1
Write-Hit	Write to L1 Change the line state to modified
Read-Miss	Read from L2
Write-Miss	Allocate the line in L1 with modified state Write to L1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 95: Behaviors upon Accessing Write-Back and Read-and-Write-Allocate Regions (MTYP=11)

Scenario	Operations
Read-Hit	Read from L1
Write-Hit	Write to L1 Change the line state to modified
Read-Miss	Read from L2 Allocate the line in L1 with exclusive/shared state
Write-Miss	Allocate the line in L1 with modified state Write to L1

Official Release

When a write-miss happens for a write-no-allocate region (MTYP=8/9), the missed line is not allocated into D-Cache. The write data is written to L2. When Write-Around feature is configured, AX45MP implements write buffers to improve write bandwidth to L2. See Section 10.4.4 for more details.

AX45MP does not improve the read bandwidth of read-no-allocate region (MTYP=8/10). When a read-miss happens for read-no-allocate region (MTYP=8/10), D-Cache sends a single transfer to L2. The read to the same line is blocked until the transfer is completed.

13.6.2 I-Cache Disabled Behaviors

When `mcache_ctl.IC_EN` is 0, all fetches to cacheable memory regions are treated as non-cacheable and bufferable. For CPU revision before 2.0.0, the ARCACHE is not affected. For CPU revision 2.0.0 and later, ARCACHE is affected by `mcache_ctl.IC_EN`. When I-Cache is disabled, the requests are sent to MMIO port without passing through L2-Cache.

13.6.3 D-Cache Disabled Behaviors

When `mcache_ctl.DC_EN` is 0, all load and store instructions to cacheable memory regions are treated as non-cacheable and bufferable. When D-Cache is disabled, the requests are sent to MMIO port without passing through L2-Cache.

13.6.4 Debug Mode

In the debug mode, allocate attributes are ignored. AX45MP does not allocate lines to D-Cache and I-Cache in the debug mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

13.7 Cacheable Memory Accesses to L2-Caches

This section describes operations to L2-Cache. L2-Cache read requests include cache filling and non-allocate read. L2-Cache write requests include cache eviction and non-allocate write.

Official Release

Table 96: Behaviors upon Accessing Write-Back and No-Allocate Regions (MTYP=8)

Scenario	Operations
Read-Hit	Read from L2
Write-Hit	Write to L2 Change the line state to dirty
Read-Miss	Read from L3
Write-Miss	Write to L3

Table 97: Behaviors upon Accessing Write-Back and Read-Allocate Regions (MTYP=9)

Scenario	Operations
Read-Hit	Read from L2
Write-Hit	Write to L2 Change the line state to dirty
Read-Miss	Read from L3 Allocate the line in L2 with clean state
Write-Miss	Write to L3

Table 98: Behaviors upon Accessing Write-Back and Write-Allocate Regions (MTYP=10)

Scenario	Operations
Read-Hit	Read from L2
Write-Hit	Write to L2 Change the line state to dirty
Read-Miss	Read from L3
Write-Miss	Allocate the line in L2 with dirty state Write to L2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 99: Behaviors upon Accessing Write-Back and Read-and-Write-Allocate Regions (MTYP=11)

Scenario	Operations
Read-Hit	Read from L2
Write-Hit	Write to L2 Change the line state to dirty
Read-Miss	Read from L3 Allocate the line in L2 with clean state
Write-Miss	Allocate the line in L2 with dirty state Write to L2

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

14 MemBoost

14.1 Non-Blocking Memory Access

Official Release

Non-blocking memory access hides memory latency by concurrently executing instructions and memory accesses. Store instructions in AX45MP are always non-blocking. Load instructions in AX45MP are blocking by default, meaning that they will block all subsequent instructions until the required bus accesses are finished (e.g., on cache misses, or when D-Cache is disabled or non-present).

The control bit mmisc_ctl.NBLD_EN controls the blocking behavior of load instructions. Once set, load instructions to cacheable memory will be non-blocking, waiting in the background for bus accesses to finish, and allow subsequent instructions to continue execution until data dependency hazards or structural hazards occur.

Data dependency hazards are hazards where subsequent instructions accessing the destination registers of the load instructions. Structural hazards are the resource limitations to allow load/store instructions to wait in the background. The number of cacheable accesses is limited by the maximum outstanding D-Cache misses.

When the processor operates in the non-blocking mode, the bus read transactions for the outstanding load/stores are each assigned a different AxID to allow the data to return out-of-order.

When load instructions operate in the blocking mode, bus aborts/errors are reported synchronously as *Load Access Faults* (mcause = 5). On the other hand, bus aborts/errors on load instructions will no longer be reported synchronously in the non-blocking mode. They will be treated as imprecise exceptions and reported as *Bus Read/Write Transaction Error Local Interrupts* (mip.BWEI).

Regardless of mmisc_ctl.NBLD_EN, the following memory access instructions are always blocking:

- Load instructions to device and non-cacheable memory
- Load instructions to local memory
- Load-Reserved/Store-Conditional instructions
- Atomic memory access (AMO) instructions

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

14.2 D-Cache Write-Around Support

Store operations to write-allocate memory will allocate a D-Cache entry on cache misses, in the hope that the entry will later be used again. However, in some streaming cases, such as memory copy or memory set operations for a large amount of data, where data is used once, allocating D-Cache entries is detrimental to performances. When Write-Around feature is configured, the store operations to the same write-no-allocate line are merged to a burst write transfer for improving performance. Write-no-allocate lines are memory regions with the following physical memory attributes:

Official Release

- Cacheable memory, write-back, no allocate
- Cacheable memory, write-back, read-allocate
- D-Cache is in write-no-allocate mode (see below)
  - Cacheable memory, write-back, write-allocate
  - Cacheable memory, write-back, read and write-allocate

When `mcache_ctl.DC_WAROUND` is set, the processor automatically detects streaming memory write patterns and makes D-Cache switch to write-no-allocate mode when the following conditions are met:

- A D-Cache miss happens for a store operation.
- This missing cache line is then completely filled by subsequent store operations. During this process, no cache miss for any other cache line happens for any store operation. Note that the same bytes in the missing cache line may be overwritten multiple times.
- The above mentioned conditions happen consecutively for a configured amount of cache lines. The corresponding addresses of those cache lines do not need to be consecutive.

In write-no-allocate mode, write-allocate attribute is overridden to write-no-allocate. For CPU revisions before 2.0.0, the overridden attribute affects D-Cache allocation policy, but AWCACHE of bus requests is not affected. For CPU revision 2.0.0 and later, both D-Cache allocation policy and AWCACHE are affected.

The processor stays in write-no-allocate mode until any of the following conditions happens:

- A D-Cache miss happens for a store operation while previous store operations have not completely filled their cache line.
- A load operation accesses the same cache line, which is being filled by store operations honored by Write-Around and is not full yet.
- `mcache_ctl.DC_WAROUND` is changed.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The Write-Around feature should be configured (see Section 2.10.2) and enabled (see Section 21.12.6) to take effect.

Note

- Stores qualified for the Write-Around behavior may produce AXI write transactions with partial or even zero write strobes in some corner case scenarios. As a result, Write-Around support should not be enabled for accessing regions containing designs that do not support AXI partial and zero write strobes, such as the Andes ATCAXI2AHB100 and ATCAXI2AHB200 bridges.
- For CPU revision 10.0.0 and later, the processor does not produce AXI write transactions with zero write strobes.

Official Release

14.3 D-Cache Prefetch

AX45MP implements a hardware prefetcher for hiding memory access latency. The hardware prefetcher has a 4-entry reference prediction table, and each entry can detect an access sequence with a fixed stride. When `mcache_ctl.DPREF_EN` is set, the following access pattern enables the hardware prefetcher:

- A load instruction that accesses to a cacheable and read-allocate memory region
- The access causes cache misses
- The same load instruction accesses another two cache lines with a fixed stride (positive or negative).

After the pattern is detected, the hardware prefetcher automatically fills the next cache lines (with the same stride) into D-Cache. The hardware prefetcher does not send requests to system bus for the lines that have been in the D-Cache.

The prefetcher is stopped in one of the following conditions:

- The stride of the load access is changed
- Entering debug mode
- The load accesses
  - Cross LM boundary
  - Have inconsistent PMA
  - Cause access fault
  - Bus error
  - ECC error
  - PMP violations
  - PMA empty hole
  - Page fault
- SFENCE.VMA is executed
- SATP is updated

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Hit a load/store debug trigger
- A trap is taken
- A trap return instruction is executed

The prefetcher is also stopped when prefetch accesses encounter one of the following conditions. These errors will be ignored without triggering any exception.

Official Release

- The accesses cross LM boundary
- The PMA is inconsistent with the load access
- Bus error
- ECC error
- PMP violations
- PMA empty hole
- Page fault

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15 Bus Master Interface

15.1 Introduction

Official Release

AX45MP provides three AXI master interfaces for sending requests to L3. The requests are routed by physical memory attributes.

- MEM
  - Cacheable memory
- MMIO
  - Non-cacheable memory
  - Device, and the address is not in the shared peripheral region
- SPP
  - Device, and the address is in the shared peripheral region

15.2 AXI4 Memory Type Encoding

Table 100 shows the encoding for AXI4 ARCACHE and AWCACHE in memory types.

Table 100: AXI4 Memory Type Encoding

Memory Type	ARCACHE[3:0]	AWCACHE[3:0]
Device, Non-Bufferable	0000	0000
Device, Bufferable	0001	0001
Memory, Non-cacheable, Non-bufferable	0010	0010
Memory, Non-cacheable, Bufferable	0011	0011
Memory, Write-back, No-allocate	1011	0111
Memory, Write-back, Read-allocate	1111	0111
Memory, Write-back, Write-allocate	1011	1111
Memory, Write-back, Read and Write-allocate	1111	1111

AWCACHE of cache line write back (eviction) is always 1111 (Write-back, Read and Write-allocate).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.3 MEM Interface

MEM interface is used to send cacheable requests to L3 memory system. When L2-Cache is disabled, L1 requests are bypassed to L3. When L2-Cache is enabled, the L2-Cache controller looks up TAG SRAM to determine L2 hit or miss. When a L2 miss occurs, the controller sends a request to L3.

Official Release

15.3.1 MEM Transaction Types

Table 101, Table 102, and Table 103 summarize the transactions of MEM interface.

Table 101: MEM Transactions with 128-Bit Data Width

Request Types	AxBURST	AxLEN	AxSIZE
Load to read-no-allocate memory	INCR	0 (1 transfer)	3
Store to write-no-allocate memory (**D-Cache Write-Around Support** is not configured)	INCR	0 (1 transfer)	0
			1
			2
			3
Store to write-no-allocate memory (**D-Cache Write-Around Support** is configured)	INCR	3 (4 transfer)	4
Fetch to read-no-allocate memory	INCR	0 (1 transfer)	3
IOCP read	INCR	0 (1 transfer)	0
			1
			2
			3
			4
		3 (4 transfer)	4
IOCP write	INCR	0 (1 transfer)	0
			1
			2
			3
			4
		3 (4 transfer)	4
L2-Cache Fill	INCR	3 (4 transfer)	4
L2-Cache Writeback	INCR	3 (4 transfer)	4
D-Cache Fill (L2-Cache is disabled)	INCR	3 (4 transfer)	4
D-Cache Writeback (L2-Cache is disabled)	INCR	3 (4 transfer)	4
I-Cache Fill (L2-Cache is disabled)	INCR	3 (4 transfer)	4

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 102: MEM Transactions with 256-Bit Data Width

Request Types	AxBURST	AxLEN	AxSIZE
Load to read-no-allocate memory	INCR	0 (1 transfer)	3
Store to write-no-allocate memory (**D-Cache Write-Around Support** is not configured)	INCR	0 (1 transfer)	0
			1
			2
			3
Store to write-no-allocate memory (**D-Cache Write-Around Support** is configured)	INCR	1 (2 transfer)	5
Fetch to read-no-allocate memory	INCR	0 (1 transfer)	3
IOCP read	INCR	0 (1 transfer)	0
			1
			2
			3
			4
			5
		1 (2 transfer)	5
IOCP write	INCR	0 (1 transfer)	0
			1
			2
			3
			4
			5
		1 (2 transfer)	5
L2-Cache Fill	INCR	1 (2 transfer)	5
L2-Cache Writeback	INCR	1 (2 transfer)	5
D-Cache Fill (L2-Cache is disabled)	INCR	1 (2 transfer)	5
D-Cache Writeback (L2-Cache is disabled)	INCR	1 (2 transfer)	5
I-Cache Fill (L2-Cache is disabled)	INCR	1 (2 transfer)	5

Official Release

Table 103: MEM Transactions with 512-Bit Data Width

Request Types	AxBURST	AxLEN	AxSIZE
Load to read-no-allocate memory	INCR	0 (1 transfer)	3

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 103: (continued)

Request Types	AxBURST	AxLEN	AxSIZE
Store to write-no-allocate memory (**D-Cache Write-Around Support** is not configured)	INCR	0 (1 transfer)	0 1 2 3
Store to write-no-allocate memory (**D-Cache Write-Around Support** is configured)	INCR	0 (1 transfer)	6
Fetch to read-no-allocate memory	INCR	0 (1 transfer)	3
IOCP read	INCR	0 (1 transfer)	0 1 2 3 4 5 6
IOCP write	INCR	0 (1 transfer)	0 1 2 3 4 5 6
L2-Cache Fill	INCR	0 (1 transfer)	6
L2-Cache Writeback	INCR	0 (1 transfer)	6
D-Cache Fill (L2-Cache is disabled)	INCR	0 (1 transfer)	6
D-Cache Writeback (L2-Cache is disabled)	INCR	0 (1 transfer)	6
I-Cache Fill (L2-Cache is disabled)	INCR	0 (1 transfer)	6

Official Release

Note

- D-Cache fills include cache misses and D-Cache prefetch.
- L2-Cache fills include cache misses and L2-Cache prefetch.
- D-Cache writebacks include D-Cache eviction and L1D CCTL operations.
- L2-Cache writebacks include L2-Cache eviction and L2 CCTL operations.
- See Section 10.4.4 for details about store to write-no-allocate memory.
- See Section 13.6 and Section 13.7 for cacheable memory accesses.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.3.2 Number of MEM Outstanding Transactions

Table 106 shows the read/write issuing capability of the MEM interface. The upper bound of transactions is 32 in 2-bank L2-Cache or 64 in 4-bank L2-Cache, which is corresponding to the number of handling registers in L2-Cache.

Official Release

Table 104: The Maximum Number of L1 Cacheable Requests

Attribute	Value
The maximum number of ICU transactions	2
The maximum number of DCU transactions	3-8 (**D-Cache Outstanding Misses**)
The maximum number of IOCP transactions	1, 2, 4, 8, 16, 32 (**Number of IOCP Outstanding Transactions**)

Table 105: The Maximum Number of MEM Outstanding Transactions of 1-Core Configuration with 2-Bank L2-Cache

Attribute	L2-Cache Is Disabled	L2-Cache Is Enabled
Number of Reads	min(14, (2 + x) + y)	14
Number of Writes	16	14

Table 106: The Maximum Number of MEM Outstanding Transactions of 2/4/8-Core Configurations with 2-Bank L2-Cache

Attribute	L2-Cache Is Disabled	L2-Cache Is Enabled
Number of Reads	min(30, n * (2 + x) + y)	30
Number of Writes	32	30

Table 107: The Maximum Number of MEM Outstanding Transactions of 8-Core Configuration with 4-Bank L2-Cache

Attribute	L2-Cache Is Disabled	L2-Cache Is Enabled
Number of Reads	min(60, 8 * (2 + x) + y)	60
Number of Writes	64	60

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

- n is the number of cores.
- x is D-Cache Outstanding Misses.
- y is the number of IOCP outstanding transactions.
- When L2-Cache is enabled, the maximum number of outstanding transactions includes L2-Cache prefetch requests. See Section 11.3.

Official Release

15.3.3 MEM AxID Assignments

MEM ARID/AWID are not corresponding to transaction sources or core operations. The following tables show the ARID/AWID assignments. Each AxID can have only one outstanding transaction. Each ID value is reused for different transactions.

Table 108: MEM ARID/AWID Assignment of 1-Core Configuration with 2-Bank L2-Cache

Value	Description
0x00–0x0F	Transactions from cores, IOCP, L2-Cache eviction, L2-Cache fill, and L2 CCTL command: L2_IX_WB/L2_IX_WBINVAL/L2_PA_WB/L2_PA_WBINVAL/L2_WBINVAL_ALL

Table 109: MEM ARID/AWID Assignment of 2/4/8-Core Configurations with 2-Bank L2-Cache

Value	Description
0x00–0x1F	Transactions from cores, IOCP, L2-Cache eviction, L2-Cache fill, and L2 CCTL command: L2_IX_WB/L2_IX_WBINVAL/L2_PA_WB/L2_PA_WBINVAL/L2_WBINVAL_ALL

Table 110: MEM ARID/AWID Assignment of 8-Core Configuration with 4-Bank L2-Cache

Value	Description
0x00–0x3F	Transactions from cores, IOCP, L2-Cache eviction, L2-Cache fill, and L2 CCTL command: L2_IX_WB/L2_IX_WBINVAL/L2_PA_WB/L2_PA_WBINVAL/L2_WBINVAL_ALL

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.3.4 MEM Protection Type Value

Interconnect and peripherals should not use AxPROT signals from MEM interface. For CPU Revision 5.0.0, *mem_arprot* and *mem_awprot* are arbitrary values. For CPU Revision 8.0.0 and later, *mem_arprot* and *mem_awprot* are hardwired to 000.

Official Release

15.3.5 MEM Exclusive Accesses

AX45MP does not send exclusive accesses to the MEM interface. Atomic memory operations to cacheable memory are implemented with the coherence manager. See Section 5.3 for more details.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.4 MMIO Interface

15.4.1 MMIO Transaction Types

Table 111, Table 112, and Table 113 summarize the transactions of MMIO interface. IOCP requests are present only when IOCP is configured.

Official Release

Table 111: MMIO Transactions with 128-Bit Data Width

Request Types	AxBURST	AxLEN	AxSIZE
LSU (Device and Non-Cacheable)	INCR	0 (1 transfer)	0
			1
			2
			3
Fetch (Device and Non-Cacheable)	INCR	0 (1 transfer)	3
IOCP Single (Device and Non-Cacheable)	INCR	0 (1 transfer)	0
			1
			2
			3
			4
IOCP Burst (Non-Cacheable)	INCR	3 (4 transfers)	4

Table 112: MMIO Transactions with 256-Bit Data Width

Request Types	AxBURST	AxLEN	AxSIZE
LSU (Device and Non-Cacheable)	INCR	0 (1 transfer)	0
			1
			2
			3
Fetch (Device and Non-Cacheable)	INCR	0 (1 transfer)	3
IOCP Single (Device and Non-Cacheable)	INCR	0 (1 transfer)	0
			1
			2
			3
			4
			5
IOCP Burst (Non-Cacheable)	INCR	1 (2 transfers)	5

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 113: MMIO Transactions with 512-Bit Data Width

Request Types	AxBURST	AxLEN	AxSIZE
LSU (Device and Non-Cacheable)	INCR	0 (1 transfer)	0
			1
			2
			3
Fetch (Device and Non-Cacheable)	INCR	0 (1 transfer)	3
IOCP (Device and Non-Cacheable)	INCR	0 (1 transfer)	0
			1
			2
			3
			4
			5
			6

Official Release

15.4.2 Number of MMIO Outstanding Transactions

The following table shows the read/write issuing capability of the MMIO interface.

Table 114: MMIO Interface Attributes

Attribute	Values	Comments
Number of read	(n * 3) + y	Each core can issue 2 read transactions for instruction fetch and 1 read transaction.
Number of writes	(n * x) + y	Each core can issue x write transaction for store instructions to non-cacheable memory.

Note

- n is the number of cores.
- x is the number of transactions to non-cacheable memory.
- y is the number of IOCP outstanding transactions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.4.3 MMIO AxID Assignments

Each AxID can have only one outstanding transaction. The assignment is dependent on configurations.

Official Release

Number of Processor Cores	Number of Transactions to Non-Cacheable Memory	Number of IOCP Outstanding Transactions	Assignment
1/2/4	1	1/2/4/8/16	Assignment 1
1/2/4	1	32	Assignment 2
1/2/4	4	1/2/4/8/16/32	Assignment 2
1/2/4	8	1/2/4/8/16/32	Assignment 3
1/2/4	16	1/2/4/8/16/32	Assignment 4
1/2/4	32	1/2/4/8/16/32	Assignment 5
8	1	1/2/4/8/16	Assignment 6
8	1	32	Assignment 7
8	4	1/2/4/8/16/32	Assignment 7
8	8	1/2/4/8/16/32	Assignment 8
8	16	1/2/4/8/16/32	Assignment 9
8	32	1/2/4/8/16/32	Assignment 10

Table 115: MMIO AxID Assignments of 1/2/4-Core Configurations

Source	Assignment 1	Assignment 2	Assignment 3	Assignment 4	Assignment 5
Core 0 LSU	0x0	0x00–0x03	0x00–0x07	0x00–0x0F	0x000–0x01F
Core 0 ICU	0x2–0x3	0x04–0x05	0x08–0x09	0x10–0x11	0x020–0x021
Core 1 LSU	0x4	0x08–0x0B	0x10–0x17	0x20–0x2F	0x040–0x05F
Core 1 ICU	0x6–0x7	0x0C–0x0D	0x18–0x19	0x30–0x31	0x060–0x061
Core 2 LSU	0x8	0x10–0x13	0x20–0x27	0x40–0x4F	0x080–0x09F
Core 2 ICU	0xA–0xB	0x14–0x15	0x28–0x29	0x50–0x51	0x0A0–0x0A1
Core 3 LSU	0xC	0x18–0x1B	0x30–0x37	0x60–0x6F	0x0C0–0x0DF
Core 3 ICU	0xE–0xF	0x1C–0x1D	0x38–0x39	0x70–0x71	0x0E0–0x0E1
IOCP	0x10–0x1F	0x20–0x3F	0x40–0x5F	0x80–0x9F	0x100–0x11F

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 116: MMIO AxID Assignments for 8-Core Configuration

Source	Assignment 6	Assignment 7	Assignment 8	Assignment 9	Assignment 10
Core 0 LSU	0x00	0x00–0x03	0x00–0x07	0x000–0x00F	0x000–0x01F
Core 0 ICU	0x02–0x03	0x04–0x05	0x08–0x09	0x010–0x011	0x020–0x021
Core 1 LSU	0x04	0x08–0x0B	0x10–0x17	0x020–0x02F	0x040–0x05F
Core 1 ICU	0x06–0x07	0x0C–0x0D	0x18–0x19	0x030–0x031	0x060–0x061
Core 2 LSU	0x08	0x10–0x13	0x20–0x27	0x040–0x04F	0x080–0x09F
Core 2 ICU	0x0A–0x0B	0x14–0x15	0x28–0x29	0x050–0x051	0x0A0–0x0A1
Core 3 LSU	0x0C	0x18–0x1B	0x30–0x37	0x060–0x06F	0x0C0–0x0DF
Core 3 ICU	0x0E–0x0F	0x1C–0x1D	0x38–0x39	0x070–0x071	0x0E0–0x0E1
Core 4 LSU	0x10	0x20–0x23	0x40–0x47	0x080–0x08F	0x100–0x11F
Core 4 ICU	0x12–0x13	0x24–0x25	0x48–0x49	0x090–0x091	0x120–0x121
Core 5 LSU	0x14	0x28–0x2B	0x50–0x57	0x0A0–0x0AF	0x140–0x15F
Core 5 ICU	0x16–0x17	0x2C–0x2D	0x58–0x59	0x0B0–0x0B1	0x160–0x161
Core 6 LSU	0x18	0x30–0x33	0x60–0x67	0x0C0–0x0CF	0x180–0x19F
Core 6 ICU	0x1A–0x1B	0x34–0x35	0x68–0x69	0x0D0–0x0D1	0x1A0–0x1A1
Core 7 LSU	0x1C	0x38–0x3B	0x70–0x77	0x0E0–0x0EF	0x1C0–0x1DF
Core 7 ICU	0x1E–0x1F	0x3C–0x3D	0x78–0x79	0x0F0–0x0F1	0x1E0–0x1E1
IOCP	0x20–0x2F	0x40–0x5F	0x80–0x9F	0x100–0x11F	0x200–0x21F

Official Release

15.4.4 MMIO Exclusive Accesses

AX45MP uses AXI exclusive accesses to MMIO interface for atomic operations. `ARLOCK` is set for `LR` instructions, and `AWLOCK` is set for `SC` instructions. Please see the usage descriptions for `LR` and `SC` instructions in *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA Document Version 20191213 (TD004)*.

The AX45MP product package provides ATCEXMON300 that supports AXI exclusive accesses. See Section 23.5.19.

15.4.5 MMIO Protection Type

Table 117 shows the encoding for AXI4 ARPROT and AWPROT in combination with access types and privilege modes.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 117: MMIO Protection Encoding

Access Type	Privilege Mode	AxPROT[2]	AxPROT[1]	AxPROT[0]
Device, Non-Cacheable Load/Store	M	0 (Data)	0 (Secure)	1 (Privileged)
	S	0 (Data)	1 (Non-Secure)	1 (Privileged)
	U	0 (Data)	1 (Non-Secure)	0 (Unprivileged)
Instruction Fetch	M	1 (Instruction)	0 (Secure)	1 (Privileged)
	S	1 (Instruction)	1 (Non-Secure)	1 (Privileged)
	U	1 (Instruction)	1 (Non-Secure)	0 (Unprivileged)
Device/Non-Cacheable, Page Table Walker	S, U	0 (Data)	0 (Secure)	1 (Privileged)

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.5 SPP Interface

15.5.1 SPP Transaction Types

Table 118 summarizes the transactions of SPP interface. IOCP requests are present only when IOCP is configured.

Official Release

Table 118: SPP Transactions

Request Types	AxBURST	AxLEN	AxSIZE
LSU (Device)	INCR	0 (1 transfer)	0
			1
			2
			3
Fetch (Device)	INCR	0 (1 transfer)	3
IOCP Single (Device)	INCR	0 (1 transfer)	0
			1
			2
			3
IOCP Burst (Device)	INCR	1 (2 transfer)	3
IOCP Burst (Device) when **Bus Data Width** $\geq$ 256	INCR	3 (4 transfer)	3
IOCP Burst (Device) when **Bus Data Width** $\geq$ 512	INCR	7 (8 transfer)	3

15.5.2 Number of SPP Outstanding Transactions

The following table shows the read/write issuing capability of the SPP interface.

Table 119: SPP Interface Attributes

Attribute	Values	Comments
Number of read	n * 3 + y	Each core can issue 2 read transactions for instruction fetch and 1 read transaction for load instructions.
Number of writes	n * 1 + y	Each core can issue 1 write transaction for store instructions.

Note

- n is the number of cores.
- y is the number of IOCP outstanding transactions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

15.5.3 SPP AxID Assignments

Table 120: SPP AxID Assignments of 1/2/4 Cores

Value	Description
0x00–0x03	Reserved
0x04	Transactions from Core 0 LSU (Device)
0x05	Transactions from Core 1 LSU (Device)
0x06	Transactions from Core 2 LSU (Device)
0x07	Transactions from Core 3 LSU (Device)
0x08–0x09	Transactions from Core 0 ICU (Device)
0x0A–0x0B	Transactions from Core 1 ICU (Device)
0x0C–0x0D	Transactions from Core 2 ICU (Device)
0x0E–0x0F	Transactions from Core 3 ICU (Device)
0x10–0x1F	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** < 32
0x20–0x3F	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 32

Official Release

Table 121: SPP AxID Assignments for 8 Cores

Value	Description
0x00	Transactions from Core 0 LSU (Device)
0x01	Transactions from Core 1 LSU (Device)
0x02	Transactions from Core 2 LSU (Device)
0x03	Transactions from Core 3 LSU (Device)
0x04	Transactions from Core 4 LSU (Device)
0x05	Transactions from Core 5 LSU (Device)
0x06	Transactions from Core 6 LSU (Device)
0x07	Transactions from Core 7 LSU (Device)
0x08–0x0F	Reserved
0x10, 0x11	Transactions from Core 0 ICU (Device)
0x12, 0x13	Transactions from Core 1 ICU (Device)
0x14, 0x15	Transactions from Core 2 ICU (Device)
0x16, 0x17	Transactions from Core 3 ICU (Device)
0x18, 0x19	Transactions from Core 4 ICU (Device)
0x1A, 0x1B	Transactions from Core 5 ICU (Device)

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 121: (continued)

Value	Description
0x1C, 0x1D	Transactions from Core 6 ICU (Device)
0x1E, 0x1F	Transactions from Core 7 ICU (Device)
0x20	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 1
0x21	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 2
0x22–0x23	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 4
0x24–0x27	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 8
0x28–0x2F	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 16
0x30–0x3F	Transactions from IOCP (Device) when **Number of IOCP Outstanding Transactions** $\geq$ 32

Official Release

15.5.4 SPP Exclusive Accesses

AX45MP uses AXI exclusive accesses to SPP interface for atomic operations. `ARLOCK` is set for `LR` instructions, and `AWLOCK` is set for `SC` instructions. Please see the usage descriptions for `LR` and `SC` instructions in *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA Document Version 20191213 (TD004)*.

The AX45MP product package provides ATCEXMON300 that supports AXI exclusive accesses. See Section 23.5.19.

15.5.5 SPP Protection Type

Table 122 shows the encoding for AXI4 ARPROT and AWPROT in combination with access types and privilege modes.

Table 122: SPP Protection Encoding

Access Type	Privilege Mode	AxPROT[2]	AxPROT[1]	AxPROT[0]
Device Load/Store	M	0 (Data)	0 (Secure)	1 (Privileged)
	S	0 (Data)	1 (Non-Secure)	1 (Privileged)

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 122: (continued)

	U	0 (Data)	1 (Non-Secure)	0 (Unprivileged)
Instruction Fetch	M	1 (Instruction)	0 (Secure)	1 (Privileged)
	S	1 (Instruction)	1 (Non-Secure)	1 (Privileged)
	U	1 (Instruction)	1 (Non-Secure)	0 (Unprivileged)
Device Table Walker	S, U	0 (Data)	0 (Secure)	1 (Privileged)

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

16 Coherence Manager (CM)

AX45MP supports cache coherency of 1/2/4/8 cores with D-Cache and L2-Cache. To enforce cache coherency, D-Cache and L2-Cache settings should be properly enabled. Table 123 summarized behaviors of different settings. D-Cache is enabled by setting `mcache_ctl.DC_EN`. L2-Cache is enabled by setting `CEN` in L2-Cache control register. D-Cache coherency is enabled by setting `mcache_ctl.DC_COHEN`. See Section 20.4 for sequences of enabling/disabling cache coherency.

Official Release

Table 123: Behaviors of Cache Enable/Disable Settings

D-Cache	L2-Cache	D-Cache Coherency	Description
Disabled	Don't Care	Don't Care	Cache coherence is not supported. All load/store requests are sent to MMIO port.
Enabled	Disabled	Don't Care	Cache coherence is not supported. All cacheable load/store requests bypass L2-Cache, and are sent to MEM port.
Enabled	Enabled	Disabled	Illegal usage. Accesses to shared lines might trigger access fault exceptions.
Enabled	Enabled	Enabled	Cache coherence is enforced by the coherence manager.

AX45MP implements MESI protocol in L2-Cache to maintain coherency of cacheable accesses. If L2-Cache is enabled, L2-Cache tracks line states of both L2-Cache and D-Cache to maintain the cache coherency. For each read or write request to L2-Cache, the tag SRAMs of L2-Cache are accessed to determine the cache line state. When data coherence handling is necessary, L2-Cache sends probe messages to write back or invalidate copies in D-Cache.

In some cases, L2-Cache handles the request without intervening D-Cache:

- The D-Cache coherency is disabled.
- The cache line has no valid copy in D-Cache.
- The cache line state says coherence handling is not required.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

17 I/O Coherence Port (IOCP)

The IOCP is implemented as an AXI4 slave interface for communicate between I/O devices and L2 memory subsystem with following restrictions:

Official Release

- 128/256/512 bits read and write interfaces.
- Supports `iocp0_arcache` and `iocp0_awcache` for dynamic cacheability.
- Supports static physical memory attribute for static cacheability.
- Supports FIXED, INCR, and WRAP transfer type.
- Supports 1-byte to 16-byte transfer size when configured to 128-bit interface.
- Supports 1-byte to 32-byte transfer size when configured to 256-bit interface.
- Supports 1-byte to 64-byte transfer size when configured to 512-bit interface.
- Supports 1 to 256 transfer length.
- Exclusive accesses are not supported.
- Round-robin arbiter is used to schedule the read/write accesses.

17.1 IOCP Transfer Type

Table 124, Table 125, and Table 126 summarize the supported IOCP transactions.

Table 124: Supported IOCP Transactions (128-Bit Interface)

Burst Type (iocp0_arburst/ iocp0_awburst)	**Transfer Size (iocp0_arsize/ iocp0_awsize)**	**Transfer Length (iocp0_arlen/ iocp0_awlen)**
FIXED	0-4	0-15 (1 transfer - 16 transfers)
INCR	0-4	0-255 (1 transfer - 256 transfers)
WRAP	0-4	1,3,7,15 (2 transfers, 4 transfers, 8 transfers, 16 transfers)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 125: Supported IOCP Transactions (256-Bit Interface)

Burst Type (iocp0_arburst/ iocp0_awburst)	Transfer Size (iocp0_arsize/ iocp0_awsize)	Transfer Length (iocp0_arlen/ iocp0_awlen)
FIXED	0-5	0-15 (1 transfer - 16 transfers)
INCR	0-5	0-255 (1 transfer - 256 transfers)
WRAP	0-5	1,3,7,15 (2 transfers, 4 transfers, 8 transfers, 16 transfers)

Official Release

Table 126: Supported IOCP Transactions (512-Bit Interface)

Burst Type (iocp0_arburst/ iocp0_awburst)	Transfer Size (iocp0_arsize/ iocp0_awsize)	Transfer Length (iocp0_arlen/ iocp0_awlen)
FIXED	0-6	0-15 (1 transfer - 16 transfers)
INCR	0-6	0-255 (1 transfer - 256 transfers)
WRAP	0-6	1,3,7,15 (2 transfers, 4 transfers, 8 transfers, 16 transfers)

Note that a burst must not cross a 4KiB address boundary, and the early termination of bursts is not supported. The IOCP also does not include logic to guarantee atomicity of atomic instructions accessing when external masters access the same location through the IOCP, nor does it provide the protection feature on IOCP.

17.2 IOCP Cacheability

The cacheability of IOCP requests are determined by static physical memory attribute, `iocp0_arcache` and `iocp0_awcache` signals. If the address match the Device Region (see Section 2.12), the attribute is device non-bufferable. Otherwise, the memory type attribute is determined by `iocp0_arcache` and `iocp0_awcache` signals.

Device and non-cacheable memory are sent to MMIO bus interface.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Cacheable memory are coherent with L1 D-Caches, and the requests are sent to L2-Cache. If L2-Cache is not configured, the requests is sent to Memory bus interface. The write-through memory type is not supported, and the behaviors are unpredictable.

The accesses to L2-Cache registers are sent to L2-Cache controller when L2-Cache is configured, otherwise sent to MMIO bus interface or Memory bus interface depends on the cacheability.

Official Release

Table 127: IOCP Write Behaviors

Memory Type	L2-Hit	L2-Miss
Write-Back, Read and Write Allocate (AWCACHE==0b1111)	Write to L2-Cache	Allocate a line in L2-Cache Write to L2-Cache
Write-Back, No Allocate (AWCACHE==0b0111)	Write to L2-Cache	Write to L3 (MEM)
Non-cacheable Memory (AWCACHE[3:1]==0b001)	Write to L3 (MMIO)	Write to L3 (MMIO)
Device (AWCACHE[3:1]==0b000)	Write to L3 (MMIO)	Write to L3 (MMIO)
Write-Through	Not Supported	Not Supported

Table 128: IOCP Read Behaviors

Memory Type	L2-Hit	L2-Miss
Write-Back, Read and Write Allocate (ARCACHE==0b1111)	Read from L2-Cache	Allocate a line in L2-Cache Read from L2-Cache
Write-Back, No Allocate (ARCACHE==0b1011)	Read from L2-Cache	Read from L3 (MEM)
Non-cacheable Memory (ARCACHE[3:1]==0b001)	Read from L3 (MMIO)	Read from L3 (MMIO)
Device (ARCACHE[3:1]==0b000)	Read from L3 (MMIO)	Read from L3 (MMIO)
Write-Through	Not Supported	Not Supported

17.3 IOCP Performance

The AX45MP supports up to 32 outstanding IOCP requests. When an IOCP request is received, the request is decomposed to multiple accesses of 64-byte lines. The performance is dependent on cacheability.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

17.3.1 Cacheable Accesses

Table 129: IOCP Access Time (Cycle) of a 64-Byte Cacheable Access

AxBURST	Condition	Read Latency	Write Latency
INCR (NHART is 1, 2, 3 or 4)	L2_MISS	22	32/33
	L1D_SHARE	28	34/35
	L1D_EXCLUSIVE	28	34/35
	L1D_MODIFIED	30	40/41
	L2_HIT	20/21	23/24
INCR (NHART is 8)	L2_MISS	24	34/35
	L1D_SHARE	32	38/39
	L1D_EXCLUSIVE	32	38/39
	L1D_MODIFIED	34	44/45
	L2_HIT	22/23	25/26
FIXED	L2_MISS	60/61	87
	L1D_SHARE	88	89/90
	L1D_EXCLUSIVE	88	89/90
	L1D_MODIFIED	177	94
	L2_HIT	59/60	77
WRAP	L2_MISS	60/61	87
	L1D_SHARE	88	89/90
	L1D_EXCLUSIVE	88	89/90
	L1D_MODIFIED	177	94
	L2_HIT	59/60	77

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

- The latency assumes that memory Type is “Write-Back, Read and Write Allocate”.
- The latency assumes that `iocp_awburst/iocp_arburst` is 0b01 (INCR).
- The latency assumes that AxSIZE is 4 and AxLEN is 3.
- The latency assumes that address is 64-byte aligned.
- The latency assumes that L2_CLK and BUS_CLK clock are synchronous and the ratio is 1:1.
- The latency assumes that core interface is synchronous.
- The latency assumes that bus latency is 1 cycle.
- The latency assumes that bus data width is 128.
- The latency assumes that the Tag RAM output cycle of L2-Cache is 1 cycle.
- The latency assumes that the Tag RAM setup cycle of L2-Cache is 1 cycle.
- The latency assumes that the Data RAM output cycle of L2-Cache is 2 cycle.
- The latency assumes that the Data RAM setup cycle of L2-Cache is 1 cycle.

Official Release

17.3.2 Device and Non-Cacheable Accesses

Table 130: Access Time (Cycle) of Device or Non-Cacheable Accesses From IOCP

AxBURST	Latency
INCR/FIXED/WRAP Read	40
INCR/FIXED/WRAP Write	37

Note

- The latency assumes that core interface is synchronous.
- The latency assumes that L2_CLK and BUS_CLK clock are synchronous and the ratio is 1:1.
- The latency assumes that AxSIZE is 4 and AxLEN is 0.
- The latency assumes that bus latency is 1 cycle.
- The latency assumes that bus data width is 128.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

18 Trap

18.1 Introduction

Official Release

According to the RISC-V Privileged Architecture, a trap is a control flow change of normal instruction execution caused by an interrupt or an exception. An interrupt is initiated by an external source, while an exception is generated as a by-product of instruction execution. When a trap happens, the processor stops processing the current flow of instructions, disables interrupts, saves enough states for later resumption, and starts executing a trap handler.

Interrupts can be local or external. The external interrupts are global interrupts that are arbitrated externally by a platform level interrupt controller (PLIC) and the selected external interrupt joins the rest of local interrupts for arbitration to take a trap.

Exceptions can be precise or imprecise. The instruction causing precise exceptions and all its subsequent instructions in the program order will not have affected the architectural state when precise exceptions are triggered. Furthermore, the events that cause these precise exceptions have to be precisely attributed to the causing instruction. The value of `mcause` register will be greater than zero for precise exceptions. Exceptions not meeting these criteria can only be imprecise and they are delivered as local interrupts (`mcause` < 0) instead. That is, the standard RISC-V privileged architecture exceptions are only triggered for precise exceptions, and local interrupts are triggered for imprecise exceptions.

For precise exceptions, `mepc` is the PC of the faulting instruction. For imprecise exceptions, `mepc` is pointing to the interrupted instruction. Regardless of preciseness of exceptions, `mtval` records the effective faulting address for exceptions related to memory operations.

18.2 Interrupt

AX45MP provides three interrupt inputs: timer interrupt, software interrupt, and external interrupt. Timer interrupts and software interrupts are local interrupts in a RISC-V platform, which means each processor in the platform receives its own timer/software interrupts. External interrupts are global interrupts in a RISC-V platform shared by all processors in a RISC-V platform. External interrupts are arbitrated and distributed by a platform-level interrupt controller (PLIC) to a processor. Each external interrupt source can be assigned its own priority, and each interrupt target (i.e., RISC-V processors) can select which external interrupt sources it will handle. PLIC routes the highest priority interrupt source to the target processor. See Section 24 for more descriptions on PLIC.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

18.2.1 Additional Local Interrupts

In addition to external interrupts, AX45MP may generate internal interrupts for the following events (imprecise exceptions):

- Local memory slave port parity/ECC error (Section 9.3. See mie.IMECCI and mip.IMECCI)
- Cache write back parity/ECC error (See mie.IMECCI and mip.IMECCI)
- Bus read/write transaction error (See mie.BWEI, mip.BWEI, and mdcause)
- Performance monitor overflow (See mie.PMOVI, mip.PMOVI, and mdcause)

Official Release

18.2.2 Interrupt Status and Masking

The `mip` CSR contains pending bits of these interrupts, with the `mie` CSR contains enable bits of the respective interrupts. The processor can selectively enable interrupts by manipulating the `mie` CSR, or globally disable interrupts by clearing the `mstatus.MIE` bit.

18.3 Exception

AX45MP implements the following (precise) exceptions (`mcause` > 0). See the tables in Section 21.3.13 (and Section 21.3.9) for how these exceptions can be identified by trap handlers.

- Instruction address misaligned exceptions
  - Jump to misaligned addresses
- Instruction access faults
  - Bus errors caused by instruction fetches
  - Uncorrectable ECC errors when fetching
  - PMP faults caused by instruction fetches
  - PMA faults caused by instruction fetches
- Illegal instructions
  - Unsupported instructions
  - Privileged instructions
  - Accessing non-existent CSRs
  - Accessing privileged CSRs
  - Writing to read-only CSRs
  - Executing Andes-specific instructions in the RISC-V compatibility mode (`mmisc_ctl.RVCOMPM` == 1).
  - Executing floating point instructions without enabling FPU context (`mstatus.FS` == 0).
  - Executing ACE instructions without enabling ACE context (`mmisc_ctl.ACES` == 0).
- Breakpoint exceptions

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Load address misaligned exceptions
- Load access faults
  - Bus errors caused by load instructions
  - ECC errors caused by load instructions
  - PMP fault caused by load instructions
  - PMA fault caused by load instructions
- Store/AMO address misaligned exceptions
- Store/AMO access faults
  - Bus errors caused by store instructions
  - ECC errors caused by store instructions
  - PMP fault caused by store instructions
  - PMA fault caused by store instructions
- Environment calls
- Stack overflow/underflow exceptions with StackSafe supported
- ACE exceptions

Official Release

Some events (for example, parity/ECC and bus errors) in this list may cause imprecise exceptions in some circumstances instead. Imprecise exceptions are delivered through local interrupts (`mcause` < 0) instead of the standard RISC-V exceptions (`mcause` > 0). It all depends on the ability of the pipeline to attribute the errors to the faulting instruction and keep the architectural state clean from being polluted by the faulting instruction and all of its subsequent instructions. For example, bus read errors on non-critical word cannot be attributed to any of the executed instructions and will be imprecise. As another example, when the processor is in the non-blocking mode (Section 14.1), load instructions could have been retired before data returns and bus errors will always be imprecise.

Most of errors related to address checks are precise, unless the instruction is split into micro-operations and the error is found not on the first micro-operation. For example, PMP checks errors for the second micro-operation of a misaligned memory accesses.

18.4 Trap Handling

18.4.1 Entering the Trap Handler

When a trap occurs, the following operations are applied:

- `mepc` is set to the current program counter.
- `mstatus` is updated.
  - The `MPP` field is set to the current privilege mode.
  - The `MPIE` field is set to the `MIE` field.
  - The `MIE` field is set to 0.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- mcause is updated.
- mtval is updated on any of address-misaligned, access-fault, or page-fault exceptions.
- The privilege mode is changed to M-mode.
- When mmisc_ctl.VEC_PLIC is 0, the program counter is set to the address specified by mtvec.
- When mmisc_ctl.VEC_PLIC is 1, the mtvec register will be the base address register of a vector table with 4-byte entries storing addresses pointing to interrupt service routines.
  - mtvec[0] is for exceptions and non-external local interrupts. For these traps, the mcause register records the trap type based on RISC-V definitions.
  - mtvec[*i*] is for external PLIC interrupt source *i* triggered through the mip.MEIP pending condition.
  - mtvec[1024+*i*] is for external PLIC interrupt source *i* triggered through
    * the mip.SEIP pending condition when mideleg.SEI == 0 for M/S/U systems.
    * the mip.UEIP pending condition when mideleg.UEI == 0 for M/U systems.
  - mtvec[2048+*i*] is for external PLIC interrupt source *i* triggered through the mip.UEIP pending condition when mideleg.UEI == 0 for M/S/U systems.
  - For external PLIC interrupts, the mcause register records the interrupt source ID. The RISC-V architecture defines a two-level stack of interrupt enable bits and privilege modes. To support nested traps, the trap handler should back up trap handling CSRs and enable the interrupt enable bit.

Official Release

18.4.2 Returning from the Trap Handler

After handling a trap, the MRET instruction can be executed for returning to the instruction and the privilege context before the trap happened. Alternatively, the trap handler could assign new PC, privilege level and/or interrupt enable status to mepc, mstatus.MPP and mstatus.MPIE before MRET. Specifically, the following operations take place when an MRET instruction is executed:

- The program counter is set to mepc
- The privilege mode is set to mstatus.MPP
- mstatus is updated
  - The MPP field is set to U-mode (or M-mode if U-mode is not supported)
  - The MIE field is set to the MPIE field
  - The MPIE field is set to 1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

19 Reset and Non-Maskable Interrupts

19.1 Reset

Official Release

When the coreN_reset_n input signal of the processor is deasserted, the following operations are applied:

- CSRs are set to their reset values.
- All integer registers (listed in Table 62) are set to zero.
- BTB is initialized.
- Program execution starts with the address specified by the coreN_reset_vector input signal.

19.2 Non-Maskable Interrupts

Non-maskable interrupts (NMIs) are intended for handling hardware error conditions and are assumed to be non-resumable. They are triggered through the coreN_nmi input signal. The rising edge of the signal causes an immediate jump to an address stored in the mnvec register and transition of the privilege level to M-mode, regardless of the state of a hart's interrupt enable bit.

The following operations are applied when an NMI is taken:

- The mepc register is written with the address of the next instruction when the NMI was taken.
- The mcause register is set to 1, indicating that NMI is caused by the coreN_reset_vector input signal.
- The mstatus.MPP field records the privilege mode before NMI was taken.
- The mstatus.MPIE field is set to the value of mstatus.*x*IE before NMI was taken. The "x" is the active privilege mode before the NMI was taken.
- The mstatus.MIE field is set to 0.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

20 Power Management

20.1 Power Management Unit

Official Release

A power management unit is used to control clocks, resets, isolation cells and power switches. AX45MP does not provide the power management unit. To demonstrate the power management sequences, AX45MP uses a reference design in AE350 SMU. Please see Section 23.9.34.2 for details.

20.2 Wait-For-Interrupt Mode

The processor enters the wait-for-interrupt (WFI) mode with the `WFI` instruction for reducing power consumption, and clock-gating or power-gating of the processor should only happen when the processor is in the WFI mode.

Upon execution of the `WFI` instruction, the processor stops all activities and asserts the `coreN_wfi_mode` output signal to indicate that this processor is in the WFI mode.

Once in the WFI mode, memory transactions that are started before the execution of `WFI` are guaranteed to have been completed. All transient states of memory handling are flushed, and no new memory accesses will take place.

In this period, the `coreN_clk` and `bus_clk_en` input signals can be safely gated by an external clock generator to reduce the power consumption or changed for frequency scaling. This is also the safe period to power-gate the processor and leave the I/D-Cache SRAMs entering the state retention mode.

Slave port accesses are not affected by WFI mode. The availability depends on whether `coreN_lm_clk` is active. If slave port accesses are still needed in WFI mode, `coreN_lm_clk` should still be clocked while `coreN_clk` may be gated off. Please note that a core cannot leave the WFI mode when the `coreN_clk` is gated off. The external clock generator should resume the `coreN_clk` when detecting wakeup events, e.g interrupts and nmi.

`nmi`, `debugint`, and interrupts defined in the `mip` CSR may cause the processor to leave the WFI mode.

The `nmi` or `debugint` signals cause the processor to leave the WFI mode unconditionally. The processor will resume and start to execute from the first instruction of NMI or debug-interrupt service routine.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

All interrupts defined in the mip CSR may cause the processor to leave the WFI mode, depending on the setting of the mie CSR: interrupts disabled by the mie CSR cannot wake up the processor. However, the processor can be awoken by these interrupts regardless the value of the global interrupt enable bit (mstatus.MIE).

When the processor is awoken by a pending interrupt and mstatus.MIE is enabled, it will resume and start to execute from the corresponding interrupt service routine. When the processor is awoken by a pending interrupt and mstatus.MIE is disabled, it will resume and start to execute from the instruction after the WFI instruction.

Official Release

Please note that the RISC-V ISA only defines the WFI instruction as a hint instruction. For portability, WFI instructions should not be assumed to always cause the processor to pause until an interrupt arrives. They may be implemented as NOPs by other implementations and should be inside loops that stop when mie & mip are not zero.

20.3 Dynamic Frequency Scaling (DFS)

When the **Core Interface** is "asynchronous", the core clocks are asynchronous to L2 clock and bus clock. An external clock generator can be implemented for scaling the core clocks, coreN_clk, coreN_dcu_clk, and coreN_lm_clk.

Note

Andes does not provide the external clock generator. The clock generator should generate stable clocks.

20.4 D-Cache Coherency

AX45MP supports cache coherency with D-Cache and coherence manager (CM). When a core is in WFI mode, CM might still send probe requests to the core. When the core receives probe requests, coreN_wfi_mode is still asserted. For responding the probe requests, coreN_dcu_clk of the core should be enabled. Before disabling all clocks of a core or powering down the core, D-Cache coherency should be disabled.

20.4.1 Disable D-Cache Coherency

To disable D-Cache coherency of a core, apply the following sequence:

- Disable D-Cache by clearing mcache_ctl.DC_EN. This step changes accesses to non-cacheable regions for preventing D-Cache from allocating more lines.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Write back and invalidate D-Cache by executing the CCTL command L1D_WBINVAL_ALL. All modified lines are flushed and D-Cache will not cache stale lines after power-up.
- Disable D-Cache coherency by clearing `mcache_ctl.DC_COHEN`. CM will not send probe requests while D-Cache is powered down.
- Wait for `mcache_ctl.DC_COHSTA` to be cleared to ensure the previous step is completed.

Official Release

20.4.2 Enable D-Cache Coherency

To enable coherency of the core, apply the following sequence:

- Enable D-Cache coherency by setting `mcache_ctl.DC_COHEN`.
- Wait for `mcache_ctl.DC_COHSTA` to be set.
- Enable D-Cache by setting `mcache_ctl.DC_EN`.

For maintaining a coherent view between cores, D-Cache coherency should be enabled before D-Cache is enabled.

20.5 Disable All Clocks of a Core

To disable all clocks of a core, apply the following sequence:

- Disable D-Cache coherency by applying the sequence in Section 20.4.1.
- Execute the WFI instruction and wait for `coreN_wfi_mode` to be asserted.
- Clocks should not be disabled until the processor enters WFI mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

20.6 Power Up and Down

20.6.1 Power Domains

Table 131 and Figure 19 show the recommended power domains of AX45MP. Isolation cells are recommended to be placed in PD_CLUSTER domain.

Official Release

Table 131: AX45MP Recommended Power Domains

Power Domain	Power State	Description
PD_CLUSTER	ON	ax45mp_cluster power domain
PD_L2	ON/OFF	ax45mp_l2_top power domain
PD_CORE<n>	ON/OFF	ax45mp_core_top power domain, where <n> is 0–7.

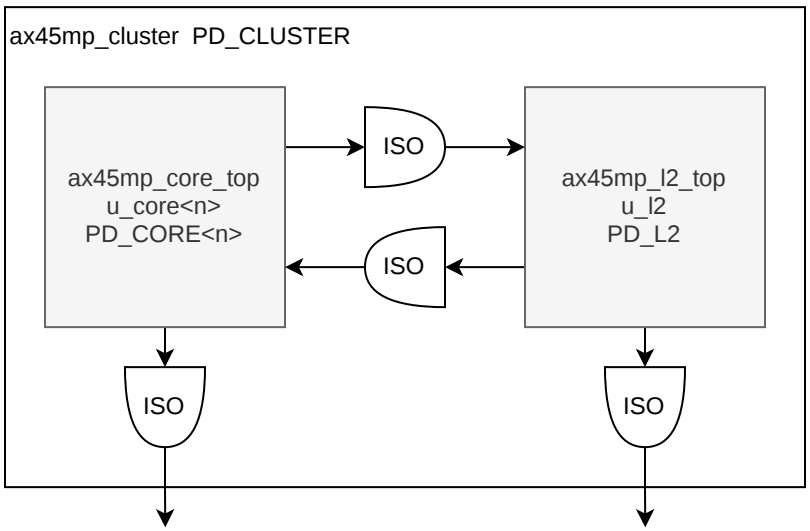


Figure 19: AX45MP Power Domains

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

20.6.1.1 Power Modes

Table 132 shows the power modes of ax45mp_cluster.

Table 132: AX45MP Power Modes

Power Domains			Description
PD_CLUSTER	PD_L2	PD_CORE<n>	
ON	ON	ON	L2-Cache and all cores are on.
ON	ON	OFF	CPU hotplug. L2-Cache and only some cores are on.
ON	OFF	OFF	Suspend to memory. L2-Cache and all cores are off.

Official Release

20.6.2 Power Down An Individual Core

To power down a core, apply the following sequence:

- Disable D-Cache coherency by applying the sequence in Section 20.4.1
- Execute the WFI instruction and wait for `coreN_wfi_mode` to be asserted.
- Enable the isolation cells for outputs of PD_CORE<n>
- Remove power from PD_CORE<n>

20.6.3 Power Up An Individual Core

To power up of a core, apply the following sequence:

- Assert coreN_reset_n of the core
- Apply power to PD_CORE<n>
- Disable the isolation cells for output of PD_CORE<n>
- De-assert coreN_reset_n of the core
- Enable D-Cache coherency by applying the sequence in Section 20.4.2

20.6.4 Power Down the Cluster

The primary core is responsible for flushing the shared L2-Cache, and other cores are secondary. To power down the cluster, apply the following sequence:

- For secondary cores
  - Power down the core by applying the sequence in Section 20.6.2.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- For the primary core
  - Disable D-Cache coherency by applying the sequence in Section 20.4.1
  - Flush L2-Cache by writing L2_WBINVAL_ALL to CCTL command register in Section 11.8.11.
  - Wait for L2_WBINVAL_ALL to be done by checking CCTL status register in Section 11.8.13.
  - Disable L2-Cache by writing L2 control register in Section 11.8.7.
- Execute the WFI instruction and wait for `coreN_wfi_mode` to be asserted.
- Enable the isolation cells for outputs of PD_CORE<n>.
- Remove power from PD_CORE<n>.
- Enable the isolation cells for outputs of PD_L2.
- Remove power from PD_L2.

Official Release

20.6.5 Power Up the Cluster

To power up the cluster, apply the following sequence:

- For the primary core
  - Assert coreN_reset_n of the core
  - Apply power to PD_CORE<n>
  - Disable the isolation cells for output of PD_CORE<n>
  - De-assert coreN_reset_n of the core
  - Enable L2-Cache by writing L2 control register in Section 11.8.7
  - Enable D-Cache coherency by applying the sequence in Section 20.4.2
- For secondary cores
  - Power up the core by applying the sequence in Section 20.6.3.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21 Control and Status Registers

21.1 Introduction

The sections below describe the registers in detail.

Official Release

21.1.1 System Register Type

Term	Description
IM	Implementation dependent/determined
RO	Read-Only register/field. Any software write to RO register/field will be silently ignored by hardware.
RW	Read/Write register/field
W1	Write-only. Only writing 1 has an effect.
W1S	Write 1 to Set
W1C	Write 1 to Clear
WLRL	Write/Read Only Legal Values
WARL	Write-Any-Read-Legal

21.1.2 Reset Value

Term	Description
DC	The reset value is “Don’t Care”.

21.1.3 CSR Listing

Table 133: Machine Information Registers

Mnemonic Name	CSR Address	Definition
mvendorid	0xf11	Section 21.2.1
marchid	0xf12	Section 21.2.2
mimpid	0xf13	Section 21.2.3
mhartid	0xf14	Section 21.2.4

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 134: Machine Trap Related Registers

Mnemonic Name	CSR Address	Definition
mstatus	0x300	Section 21.3.1
misa	0x301	Section 21.3.2
medeleg	0x302	Section 21.3.3
mideleg	0x303	Section 21.3.4
mie	0x304	Section 21.3.5
mtvec	0x305	Section 21.3.6
mscratch	0x340	Section 21.3.7
mepc	0x341	Section 21.3.8
mcause	0x342	Section 21.3.9
mtval	0x343	Section 21.3.10
mip	0x344	Section 21.3.11
mxstatus	0x7c4	Section 21.3.12
mdcause	0x7c9	Section 21.3.13
mslideleg	0x7D5	Section 21.3.14

Official Release

Table 135: Counter Related Registers

Mnemonic Name	CSR Address	Definition
mcycle	0xb00	Section 21.4.1
minstret	0xb02	Section 21.4.2
mhpmcounter3	0xb03	Section 21.4.3
mhpmcounter4	0xb04	Section 21.4.3
mhpmcounter5	0xb05	Section 21.4.3
mhpmcounter6	0xb06	Section 21.4.3
mcounteren	0x306	Section 21.4.6
mhpmevent3	0x323	Section 21.4.5
mhpmevent4	0x324	Section 21.4.5
mhpmevent5	0x325	Section 21.4.5
mhpmevent6	0x326	Section 21.4.5
mcountinhibit	0x320	Section 21.4.4
mcounterwen	0x7ce	Section 21.4.7
mcounterinten	0x7cf	Section 21.4.8
mcountermask_m	0x7d1	Section 21.4.9
mcountermask_s	0x7d2	Section 21.4.10

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 135: (continued)

Mnemonic Name	CSR Address	Definition
mcountermask_u	0x7d3	Section 21.4.11
mcounterovf	0x7d4	Section 21.4.12

Official Release

Table 136: Configuration Control & Status Registers

Mnemonic Name	CSR Address	Definition
micm_cfg	0xfc0	Section 21.5.1
mdcm_cfg	0xfc1	Section 21.5.2
mmsc_cfg	0xfc2	Section 21.5.3
mrvarch_cfg	0xfca	Section 21.5.4
ml2c_ctl_base	0xfcf	Section 21.5.5

Table 137: Trigger Registers

Mnemonic Name	CSR Address	Definition
tselect	0x7a0	Section 21.6.1
tdata1	0x7a1	Section 21.6.2
tdata2	0x7a2	Section 21.6.3
tdata3	0x7a3	Section 21.6.4
tinfo	0x7a4	Section 21.6.5
tcontrol	0x7a5	Section 21.6.6
mcontext	0x7a8	Section 21.6.7
scontext	0x7aa	Section 21.6.8
mcontrol	0x7a1	Section 21.6.9
icount	0x7a1	Section 21.6.10
itrigger	0x7a1	Section 21.6.11
etrigger	0x7a1	Section 21.6.12
textra	0x7a3	Section 21.6.13

Table 138: Debug Registers

Mnemonic Name	CSR Address	Definition
dcsr	0x7b0	Section 21.7.1
dpc	0x7b1	Section 21.7.2
dscratch0	0x7b2	Section 21.7.3

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 138: (continued)

Mnemonic Name	CSR Address	Definition
dscratch1	0x7b3	Section 21.7.4
dexc2dbg	0x7e0	Section 21.7.5
ddcause	0x7e1	Section 21.7.6

Official Release

Table 139: Supervisor Trap Related Registers

Mnemonic Name	CSR Address	Definition
sstatus	0x100	Section 21.8.1
sedeleg	0x102	Section 21.8.2
sideleg	0x103	Section 21.8.3
sie	0x104	Section 21.8.4
stvec	0x105	Section 21.8.5
scounteren	0x106	Section 21.8.6
sscratch	0x140	Section 21.8.7
sepc	0x141	Section 21.8.8
scause	0x142	Section 21.8.9
stval	0x143	Section 21.8.10
sip	0x144	Section 21.8.11
slie	0x9c4	Section 21.8.12
slip	0x9c5	Section 21.8.13
sdcause	0x9c9	Section 21.8.14

Table 140: Supervisor Page Translation Related Registers

Mnemonic Name	CSR Address	Definition
satp	0x180	Section 21.9.1

Table 141: Supervisor Counter Related Registers

Mnemonic Name	CSR Address	Definition
scountermask_m	0x9d1	Section 21.10.1
scountermask_s	0x9d2	Section 21.10.2
scountermask_u	0x9d3	Section 21.10.3
scounterinten	0x9cf	Section 21.10.4
scounterovf	0x9d4	Section 21.10.5

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 141: (continued)

Mnemonic Name	CSR Address	Definition
scountinhibit	0x9e0	Section 21.10.6
shpmevent3	0x9E3	Section 21.10.7
shpmevent4	0x9E4	Section 21.10.7
shpmevent5	0x9E5	Section 21.10.7
shpmevent6	0x9E6	Section 21.10.7

Official Release

Table 142: User Trap Related Registers

Mnemonic Name	CSR Address	Definition
ustatus	0x000	Section 21.11.1
uie	0x004	Section 21.11.2
utvec	0x005	Section 21.11.3
uscratch	0x040	Section 21.11.4
uepc	0x041	Section 21.11.5
ucause	0x042	Section 21.11.6
utval	0x043	Section 21.11.7
uip	0x044	Section 21.11.8
udcause	0x809	Section 21.11.9

Table 143: Memory and Miscellaneous Registers

Mnemonic Name	CSR Address	Definition
milmb	0x7c0	Section 21.12.1
mdlmb	0x7c1	Section 21.12.2
mecc_code	0x7c2	Section 21.12.3
mnvec	0x7c3	Section 21.12.4
mpft_ctl	0x7c5	Section 21.12.5
mcache_ctl	0x7ca	Section 21.12.6
mcctlbeginaddr	0x7cb	Section 21.12.9
mcctlcommand	0x7cc	Section 21.12.10
mcctldata	0x7cd	Section 21.12.11
scctldata	0x9cd	Section 21.12.12
ucctlbeginaddr	0x80b	Section 21.12.13
ucctlcommand	0x80c	Section 21.12.14

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 143: (continued)

Mnemonic Name	CSR Address	Definition
mmisc_ctl	0x7d0	Section 21.12.7
mclk_ctl	0x7df	Section 21.12.8

Official Release

Table 144: Hardware Stack Protection and Recording Registers

Mnemonic Name	CSR Address	Definition
mhsp_ctl	0x7c6	Section 21.13.1
msp_bound	0x7c7	Section 21.13.2
msp_base	0x7c8	Section 21.13.3

Table 145: CoDense Registers

Mnemonic Name	CSR Address	Definition
uitb	0x800	Section 21.14.1

Table 146: DSP Registers

Mnemonic Name	CSR Address	Definition
ucode	0x801	Section 21.15.1

Table 147: PMP Registers

Mnemonic Name	CSR Address	Definition
pmpcfg0	0x3a0	Section 21.16.1
pmpcfg2	0x3a2	Section 21.16.1
pmpcfg4	0x3a4	Section 21.16.1
pmpcfg6	0x3a6	Section 21.16.1
pmpaddr0	0x3b0	Section 21.16.2
pmpaddr1	0x3b1	Section 21.16.2
pmpaddr2	0x3b2	Section 21.16.2
pmpaddr3	0x3b3	Section 21.16.2
pmpaddr4	0x3b4	Section 21.16.2
pmpaddr5	0x3b5	Section 21.16.2
pmpaddr6	0x3b6	Section 21.16.2
pmpaddr7	0x3b7	Section 21.16.2

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 147: (continued)

Mnemonic Name	CSR Address	Definition
pmpaddr8	0x3b8	Section 21.16.2
pmpaddr9	0x3b9	Section 21.16.2
pmpaddr10	0x3ba	Section 21.16.2
pmpaddr11	0x3bb	Section 21.16.2
pmpaddr12	0x3bc	Section 21.16.2
pmpaddr13	0x3bd	Section 21.16.2
pmpaddr14	0x3be	Section 21.16.2
pmpaddr15	0x3bf	Section 21.16.2
pmpaddr16	0x3c0	Section 21.16.2
pmpaddr17	0x3c1	Section 21.16.2
pmpaddr18	0x3c2	Section 21.16.2
pmpaddr19	0x3c3	Section 21.16.2
pmpaddr20	0x3c4	Section 21.16.2
pmpaddr21	0x3c5	Section 21.16.2
pmpaddr22	0x3c6	Section 21.16.2
pmpaddr23	0x3c7	Section 21.16.2
pmpaddr24	0x3c8	Section 21.16.2
pmpaddr25	0x3c9	Section 21.16.2
pmpaddr26	0x3ca	Section 21.16.2
pmpaddr27	0x3cb	Section 21.16.2
pmpaddr28	0x3cc	Section 21.16.2
pmpaddr29	0x3cd	Section 21.16.2
pmpaddr30	0x3ce	Section 21.16.2
pmpaddr31	0x3cf	Section 21.16.2

Official Release

Table 148: PMA Registers

Mnemonic Name	CSR Address	Definition
pmacfg0	0xbc0	Section 21.17.1
pmacfg2	0xbc2	Section 21.17.1
pmaaddr0	0xbd0	Section 21.17.2
pmaaddr1	0xbd1	Section 21.17.2
pmaaddr2	0xbd2	Section 21.17.2
pmaaddr3	0xbd3	Section 21.17.2

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 148: (continued)

Mnemonic Name	CSR Address	Definition
pmaaddr4	0xbd4	Section 21.17.2
pmaaddr5	0xbd5	Section 21.17.2
pmaaddr6	0xbd6	Section 21.17.2
pmaaddr7	0xbd7	Section 21.17.2
pmaaddr8	0xbd8	Section 21.17.2
pmaaddr9	0xbd9	Section 21.17.2
pmaaddr10	0xbda	Section 21.17.2
pmaaddr11	0xbdb	Section 21.17.2
pmaaddr12	0xbdc	Section 21.17.2
pmaaddr13	0xbdd	Section 21.17.2
pmaaddr14	0xbde	Section 21.17.2
pmaaddr15	0xbdf	Section 21.17.2

Official Release

Table 149: Floating-Point CSRs

Mnemonic Name	CSR Address	Definition
fflags	0x001	Section 21.18.1
frm	0x002	Section 21.18.2
fcsr	0x003	Section 21.18.3

Table 150: User Mode Counter Related Registers

Mnemonic Name	CSR Address	Definition
cycle	0xc00	Section 21.19.1
time	0xc01	Section 21.19.2
instret	0xc02	Section 21.19.3
hpmcounter3	0xc03	Section 21.19.4
hpmcounter4	0xc04	Section 21.19.4
hpmcounter5	0xc05	Section 21.19.4
hpmcounter6	0xc06	Section 21.19.4

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.2 Machine Information Registers

21.2.1 Machine Vendor ID Register

Mnemonic Name: mvendorid
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xf11 (standard read only)

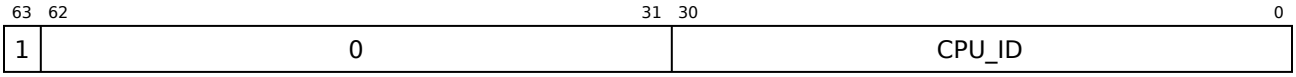
Official Release

This read-only register provides the Andes JEDEC manufacturer ID: 0x0000031e.

Field Name	Bits	Description	Type	Reset
MVENDORID	[63:0]	The manufacturer ID of Andes	RO	0x0000031e

21.2.2 Machine Architecture ID Register

Mnemonic Name: marchid
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xf12 (standard read only)



This register provides the micro-architecture id of AndesCore processor implementations. For AX45MP, `marchid.CPU_ID` will be 0x8a45. Note that the MSB of this register is 1 for commercial implementations of RISC-V processors.

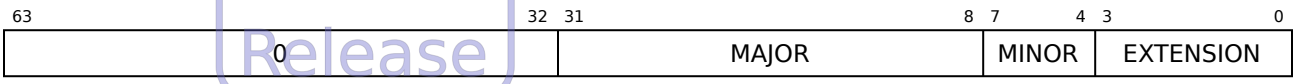
Field Name	Bits	Description	Type	Reset
CPU_ID	[30:0]	Andes CPU ID	RO	0x8a45

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.2.3 Machine Implementation ID Register

Mnemonic Name: mimpid
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xf13 (standard read only)

Official Release



This register provides the revision number of the AX45MP processor. Please see *AndesCore AX45MP R20 Release Note (RN371)* for exact values. It is documented in the release note as MAJOR.MINOR.EXTENSION.

Field Name	Bits	Description	Type	Reset
EXTENSION	[3:0]	Revision extension	RO	IM
MINOR	[7:4]	Revision minor	RO	IM
MAJOR	[31:8]	Revision major	RO	IM

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.2.4 Hart ID Register

Mnemonic Name: mhartid
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xf14 (standard read only)

Official Release

This register provides the ID of the hardware thread. It is required that one of the hart IDs must be zero on a RISC-V platform.

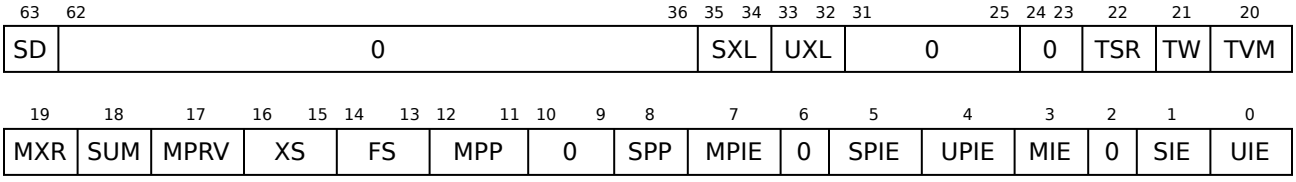
The value of this register is determined by the `hart_id` input port, and the value should not repeat in the system.

Field Name	Bits	Description	Type	Reset
MHARTID	[63:0]	Hart ID	RO	IM

21.3 Machine Trap Related CSRs

21.3.1 Machine Status

Mnemonic Name: mstatus
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x300 (standard read/write)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
UIE	[0]	U-mode interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
SIE	[1]	S-mode interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
MIE	[3]	M-mode interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
UPIE	[4]	`UPIE` holds the value of the `UIE` bit prior to a trap.	RW	0
SPIE	[5]	`SPIE` holds the value of the `SIE` bit prior to a trap.	RW	0
MPIE	[7]	`MPIE` holds the value of the `MIE` bit prior to a trap.	RW	0
SPP	[8]	`SPP` holds the privilege mode prior to a trap. Encoding is 1 for S-mode and 0 for U-mode.	RW	0
MPP	[12:11]	`MPP` holds the privilege mode prior to a trap. Encoding for privilege mode is described in Table 5. When U-mode is not available, this field is hardwired to 3.	WARL	3

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
FS	[14:13]	FS holds the status of the architectural states of the floating-point unit, including the fcsr CSR and f0 – f31 floating-point data registers. The value of this field is zero and read-only if the processor does not have FPU. This field is primarily managed by software. The processor hardware assists the state managements in two regards: • Attempts to access fcsr or any f register raise an illegal-instruction exception when FS is Off. • FS is updated to the Dirty state with the execution of any instruction that updates fcsr or any f register when FS is Initial or Clean. Changing the setting of this field has no effect on the contents of the floating-point register states. In particular, setting FS to Off does not destroy the states, nor does setting FS to Initial clear the contents.	WLRL	0

Value	Meaning
0	Off
1	Initial
2	Clean
3	Dirty

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
XS	[16:15]	XS holds the status of the architectural states (ACE registers) of ACE instructions. The value of this field is zero if ACE extension is not configured. This field is primarily managed by software. The processor hardware assists the state managements in two regards: • Illegal instruction exceptions are triggered when XS is Off. • XS is updated to the Dirty state with the execution of ACE instructions when XS is not Off. Changing the setting of this field has no effect on the contents of ACE states. In particular, setting XS to Off does not destroy the states, nor does setting XS to Initial clear the contents.	RO	0
MPRV	[17]	When the MPRV bit is set, the memory access privilege for load and store are specified by the MPP field. When U-mode is not available, this field is hardwired to 0.	RW	0

Value	Meaning
0	Off
1	Initial
2	Clean
3	Dirty

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SUM	[18]	SUM controls whether a S-mode load/store instruction to a user accessible page is allowed or not when page translation is enabled. It is in effect in two scenarios: (a) M-mode with MPRV=1 and MPP=S, and (b) in S-mode. It has no effect when page-based virtual memory is not in effect. A page is user accessible when the U bit of the corresponding PTE entry is 1. It is hardwired to 0 when S-mode is not supported.	RW	0

Value	Meaning
0	Not Allowed
1	Allowed

Field Name	Bits	Description	Type	Reset
MXR	[19]	MXR controls whether execute-only pages are readable. It has no effect when page-based virtual memory is not in effect.	RW	0

Value	Meaning
0	Execute-only pages are not readable
1	Execute-only pages are readable

Field Name	Bits	Description	Type	Reset
TVM	[20]	TVM controls whether performing certain virtual memory operations in S-mode will raise illegal instruction exceptions. The operations include accessing the satp register and executing the SFENCE.VMA instruction. It is hardwired to 0 when S-mode is not supported.	RW	0

Value	Meaning
0	Normal execution
1	Raising exceptions

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
TW	[21]	TW controls whether executing WFI instructions in S-mode will raise illegal instruction exceptions. It is hardwired to 0 when S-mode is not supported. **Value** / **Meaning** 0 — Normal execution 1 — Raising exceptions	RW	0
TSR	[22]	TSR controls whether executing SRET instructions in S-mode will raise illegal instruction exceptions. It is hardwired to 0 when S-mode is not supported. **Value** / **Meaning** 0 — Normal execution 1 — Raising exceptions	RW	0
UXL	[33:32]	UXL controls the value of XLEN for U-mode. When U-mode is not available, this field is hardwired to 0. **Value** / **Meaning** 1 — 32 2 — 64	RO	2/0
SXL	[35:34]	SXL controls the value of XLEN for S-mode. When S-mode is not available, this field is hardwired to 0. **Value** / **Meaning** 1 — 32 2 — 64	RO	2/0
SD	[63]	SD summarizes whether either the FS or XS field is dirty.	RO	0

Official Release

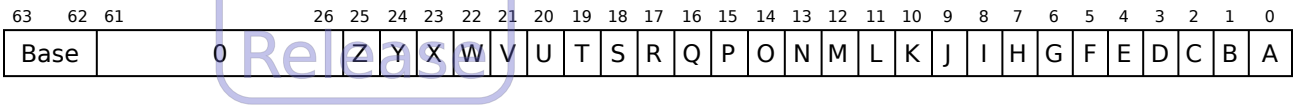
When supervisor mode or N extension is not supported, the corresponding bits in mstatus are hardwired to zero.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.3.2 Machine ISA Register

Mnemonic Name: misa
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x301 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
A	[0]	Atomic extension	RO	1
B	[1]	*Tentatively reserved for Bit operations extension*	RO	0
C	[2]	Compressed extension	RO	1
D	[3]	Double-precision floating-point extension **Value** / **RISC-V Floating-Point Instruction Extension** 0 / single precision / none 1 / double+single precision	RO	IM
E	[4]	RV32E base ISA	RO	0
F	[5]	Single-precision floating-point extension **Value** / **RISC-V Floating-Point Instruction Extension** 0 / none 1 / double+single precision / single precision	RO	IM
G	[6]	Additional standard extensions present	RO	0
H	[7]	Reserved	RO	0
I	[8]	RV32I/64I/128I base ISA	RO	1
J	[9]	*Tentatively reserved for Dynamically Translated Languages extension*	RO	0
K	[10]	Reserved	RO	0
L	[11]	*Tentatively reserved for Decimal Floating-Point extension*	RO	0
M	[12]	Integer Multiply/Divide extension	RO	1

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

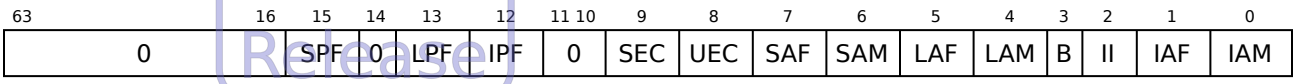
Field Name	Bits	Description	Type	Reset
N	[13]	User-level interrupts supported **Value** / **RISC-V User-Level Interrupt Extension** 0 / no 1 / yes	RO	IM
O	[14]	Reserved	RO	0
P	[15]	*Tentatively reserved for Packed-SIMD extension*	RO	0
Q	[16]	Quad-precision floating-point extension	RO	0
R	[17]	Reserved	RO	0
S	[18]	Supervisor mode implemented **Value** / **Privilege Modes** 0 / Machine / Machine + User 1 / Machine + Supervisor + User	RO	IM
T	[19]	*Tentatively reserved for Transactional Memory extension*	RO	0
U	[20]	User mode implemented **Value** / **Privilege Modes** 0 / Machine 1 / Machine + User / Machine + Supervisor + User	RO	IM
V	[21]	*Tentatively reserved for Vector extension*	RO	0
W	[22]	Reserved	RO	0
X	[23]	Non-standard extensions present	RO	1
Y	[24]	Reserved	RO	0
Z	[25]	Reserved	RO	0
Base	[63:62]	The general-purpose register width of the native base integer ISA. **Value** / **Meaning** 0 / Reserved 1 / 32 2 / 64 3 / 128	RO	2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.3.3 Machine Exception Delegation

Mnemonic Name: medeleg
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x302 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
IAM	[0]	IAM indicates whether an Instruction Address Misaligned exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
IAF	[1]	IAF indicates whether an Instruction Access Fault exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
II	[2]	II indicates whether an Illegal Instruction exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
B	[3]	B indicates whether an exception triggered by breakpoint will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
LAM	[4]	LAM indicates whether a Load Address Misaligned exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
LAF	[5]	LAF indicates whether a Load Access Fault exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
SAM	[6]	SAM indicates whether a Store/AMO Address Misaligned exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
SAF	[7]	SAF indicates whether a Store/AMO Access Fault exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
UEC	[8]	UEC indicates whether an exception triggered by environment call from U-mode will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

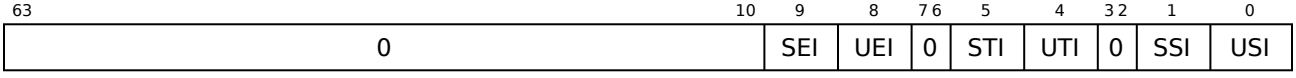
Field Name	Bits	Description	Type	Reset
SEC	[9]	SEC indicates whether an exception triggered by environment call from S-mode will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
IPF	[12]	IPF indicates whether an Instruction Page Fault exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
LPF	[13]	LPF indicates whether a Load Page Fault exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
SPF	[15]	SPF indicates whether a Store/AMO Page Fault exception will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0

Official Release

When supervisor mode or N extension is not supported, the corresponding bits in medeleg are hard-wired to zero.

21.3.4 Machine Interrupt Delegation

Mnemonic Name: mideleg
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x303 (standard read/write)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
USI	[0]	USI indicates whether an U-mode software interrupt will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
SSI	[1]	SSI indicates whether an S-mode software interrupt will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
UTI	[4]	UTI indicates whether an U-mode timer interrupt will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
STI	[5]	STI indicates whether an S-mode timer interrupt will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
UEI	[8]	UEI indicates whether an U-mode external interrupt will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0
SEI	[9]	SEI indicates whether an S-mode external interrupt will be delegated to S-mode. Value 0: Not delegate Value 1: delegate	RW	0

Official Release

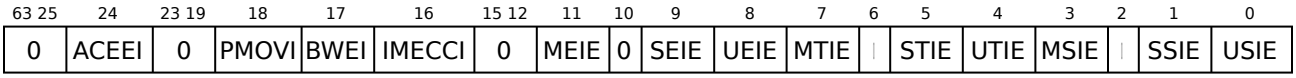
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

When supervisor mode or N extension is not supported, the corresponding bits in `mideleg` are hard-wired to zero.

21.3.5 Machine Interrupt Enable

Official Release

Mnemonic Name: mie
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x304 (standard read/write)



63 25	24	23 19	18	17	16	15 12	11	10	9	8	7	6	5	4	3	2	1	0
0	ACEEI	0	PMOVI	BWEI	IMECCI	0	MEIE	0	SEIE	UEIE	MTIE		STIE	UTIE	MSIE		SSIE	USIE

Field Name	Bits	Description	Type	Reset
USIE	[0]	U-mode software interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
SSIE	[1]	S-mode software interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
MSIE	[3]	M-mode software interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
UTIE	[4]	U-mode timer interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
STIE	[5]	S-mode timer interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MTIE	[7]	M-mode timer interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
UEIE	[8]	U-mode external interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
SEIE	[9]	S-mode external interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
MEIE	[11]	M-mode external interrupt enable bit. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
IMECCI	[16]	Imprecise ECC error local interrupt enable bit. The processor may receive imprecise ECC errors on slave port accesses or cache writebacks. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0
BWEI	[17]	Bus read/write transaction error local interrupt enable bit. The processor may receive bus errors on load/store instructions or cache writebacks. **Value** / **Meaning**: 0 Disabled; 1 Enabled	RW	0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
PMOVI	[18]	Performance monitor overflow local interrupt enable bit. 0: Disabled 1: Enabled	RW	0
ACEEI	[24]	ACE error local interrupt enable bit. 0: Disabled 1: Enabled	RW	0

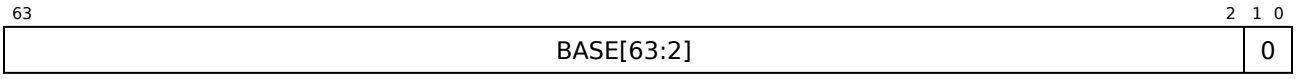
Official Release

When supervisor mode or N extension is not supported, the corresponding bits in `mie` are hardwired to zero.

Each local interrupt can be configured with a local interrupt number. Bit location of interrupts are the same as their interrupt numbers. Register fields above show the default bit location.

21.3.6 Machine Trap Vector Base Address

Mnemonic Name: mtvec
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x305 (standard read/write)



This register determines the base address of the trap vector. The least significant 2 bits are hardwired to zeros. When the configured address width is less than 64, the upper bits are hardwired to zeros. When `mmisc_ctl.VEC_PLIC` is 0 (PLIC is not in the vector mode), this register indicates the entry points for the trap handler and it may point to any 4-byte aligned location in the memory space.

On the other hand, when `mmisc_ctl.VEC_PLIC` is 1 (PLIC is in the vector mode), this register will be the base address of a vector table with 4-byte entries storing addresses pointing to interrupt service routines.

- This register should be aligned to $2^{\log_2 N+2}$-byte boundary for PLIC with N interrupt sources. For example, if N is 1023, the minimum alignment requirement is 4096 bytes (4 KiB).
- `mtvec[0]` is for exceptions and non-external local interrupts.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- `mtvec[i]` is for external PLIC interrupt source *i* triggered through the `mip.MEIP` pending condition.
- `mtvec[1024+i]` is for external PLIC interrupt source *i* triggered through
  - the `mip.SEIP` pending condition when `mideleg.SEI` == 0 for M/S/U systems.
  - the `mip.UEIP` pending condition when `mideleg.UEI` == 0 for M/U systems.
- `mtvec[2048+i]` is for external PLIC interrupt source *i* triggered through the `mip.UEIP` pending condition when `mideleg.UEI` == 0 for M/S/U systems.

Official Release

Field Name	Bits	Description	Type	Reset
BASE[63:2]	[63:2]	Base address for interrupt and exception handlers. See description above for alignment requirements when PLIC is in the vector mode.	RW	0

21.3.7 Machine Scratch Register

Mnemonic Name: mscratch
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x340 (standard read/write)

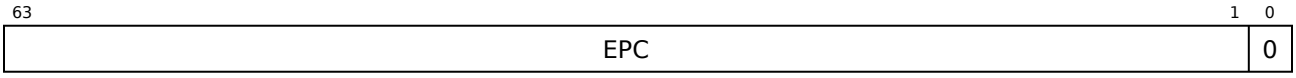
This is a scratch register for temporary data storage, which is typically used by the M-mode trap handler.

Field Name	Bits	Description	Type	Reset
MSCRATCH	[63:0]	Scratch register storage.	RW	0

21.3.8 Machine Exception Program Counter

Mnemonic Name: mepc
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x341 (standard read/write)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



This register is written with the virtual address of the instruction that encountered traps and/or NMIs when these events occurred.

When **Page-Based Virtual Memory** is configured to "sv39", bits 63–40 of this register is hardwired to bit 39. AX45MP ignores bits 63–40 of written values.

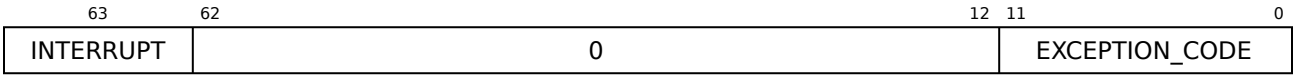
When **Page-Based Virtual Memory** is configured to "sv48", bits 63–49 of this register is hardwired to bit 48. AX45MP ignores bits 63–49 of written values.

When **Page-Based Virtual Memory** is configured to "bare", bits 63–BIU_ADDR_WIDTH of this register is hardwired to 0. AX45MP ignores bits 63–BIU_ADDR_WIDTH of written values.

Field Name	Bits	Description	Type	Reset
EPC	[63:1]	Exception program counter.	RW	0

21.3.9 Machine Cause Register

Mnemonic Name: mcause
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x342 (standard read/write)



This register indicates the cause of trap, reset, NMI or the interrupt source ID of a vector interrupt. This register is updated when a trap, reset, NMI or vector interrupt occurs. Please see Section 18 for an overview of how interrupts and exceptions are handled by AX45MP. When multiple events may cause a trap to be taken with the same `mcause` value, the value of `mdcause` records the exact event that causes the trap.

Exceptions can be precise or imprecise. Only precise exceptions are triggered as the standard RISC-V exceptions with the `mcause.INTERRUPT` bit clear. Imprecise exceptions are triggered as local interrupts, with the `mcause.INTERRUPT` bit set.

Field Name	Bits	Description	Type	Reset
EXCEPTION_CODE	[11:0]	Exception code	RW	0
INTERRUPT	[63]	Interrupt	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The following tables show the possible values of `mcause`:

Table 151: Possible Values of mcause After Trap

Interrupt	Exception Code	Description
1	0	User software interrupt
1	1	Supervisor software interrupt
1	3	Machine software interrupt
1	4	User timer interrupt
1	5	Supervisor timer interrupt
1	7	Machine timer interrupt
1	8	User external interrupt
1	9	Supervisor external interrupt
1	11	Machine external interrupt
1	16	Imprecise ECC error interrupt (slave port accesses, D-Cache evictions, and nonblocking load/stores) (M-mode)
1	17	Bus read/write transaction error interrupt (M-mode)
1	18	Performance monitor overflow interrupt (M-mode)
1	24	ACE error (M-mode)
1	256+16	Imprecise ECC error interrupt (slave port accesses, D-Cache evictions, and nonblocking load/stores) (S-mode)
1	256+17	Bus write transaction error interrupt (S-mode)
1	256+18	Performance monitor overflow interrupt (S-mode)
1	256+24	ACE error (S-mode)
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9	Environment call from S-mode
0	11	Environment call from M-mode
0	12	Instruction page fault

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 151: (continued)

Interrupt	Exception Code	Description
0	13	Load page fault
0	15	Store/AMO page fault
0	32	Stack overflow exception
0	33	Stack underflow exception
0	40–47	Andes Custom Extension exception (see *Andes Custom Extension Specification* for more details)

Official Release

Table 152: Possible Values of mcause After Reset

Interrupt	Exception Code	Description
0	0	Initial value when the processor comes out of reset (by `coreN_reset_n`)

Table 153: Possible Values of mcause After NMI

Interrupt	Exception Code	Description
0	0x001	NMI triggered

Table 154: Possible Values of mcause After Vector Interrupt

mcause	Description
Interrupt source ID	Interrupt source ID when a vector interrupt occurs

21.3.10 Machine Trap Value

Mnemonic Name: mtval
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x343 (standard read/write)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

This register is updated when a trap is taken to M-mode. The updated value is dependent on the cause of traps:

- For Hardware Breakpoint exceptions, Address Misaligned exceptions, Page Fault exceptions, or Access Fault exceptions, it is the effective faulting addresses.
- For illegal instruction exceptions, the updated value is the faulting instruction. If the length of the instruction is less than XLEN bits long, the upper bits of `mtval` are cleared to zero. For the EXEC.IT instructions triggering illegal instruction exceptions, the faulting instruction is the translated instruction. Please note that if a EXEC.IT instruction is translated to a 16-bit instruction, the translated instruction is considered an illegal instruction even if it is normally a valid one.
- For other exceptions, `mtval` is set to zero.

Official Release

For instruction-fetch access faults, this register will be updated with the address pointing to the portion of the instruction that caused the fault, while the `mepc` register will be updated with the address pointing to the beginning of the instruction.

When the configured address width is less than 64, the upper bits of `mtval` are hardwired to zeros.

Field Name	Bits	Description	Type	Reset
MTVAL	[63:0]	Exception-specific information for software trap handling.	RW	0

21.3.11 Machine Interrupt Pending

Mnemonic Name: mip
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x344 (standard read/write)

63 25	24	23 19	18	17	16	15 12	11	10	9	8	7	6	5	4	3	2	1	0
0	ACEEI	0	PMOVI	BWEI	IMECCI	0	MEIP	0	SEIP	UEIP	MTIP		STIP	UTIP	MSIP		SSIP	USIP

Field Name	Bits	Description	Type	Reset
USIP	[0]	U-mode software interrupt pending bit.	RW	0

Value	Meaning
0	Not pending
1	Pending

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SSIP	[1]	S-mode software interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
MSIP	[3]	M-mode software interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RO	0
UTIP	[4]	U-mode timer interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
STIP	[5]	S-mode timer interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
MTIP	[7]	M-mode timer interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RO	0
UEIP	[8]	U-mode external interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
SEIP	[9]	S-mode external interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MEIP	[11]	M-mode external interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RO	0
IMECCI	[16]	Imprecise ECC error local interrupt pending bit. The processor may receive imprecise ECC errors on slave port accesses or cache writebacks. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
BWEI	[17]	Bus read/write transaction error local interrupt pending bit. The processor may receive bus errors on load/store instructions or cache writebacks. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
PMOVI	[18]	Performance monitor overflow local interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0
ACEEI	[24]	ACE error local interrupt pending bit. **Value** / **Meaning** 0 — Not pending 1 — Pending	RW	0

Official Release

When supervisor mode or N extension is not supported, the corresponding bits in `mip` are hardwired to zero.

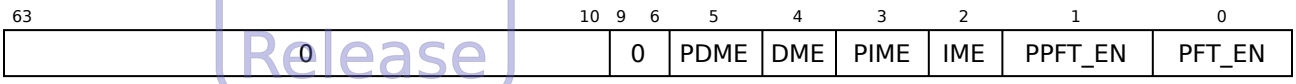
Each local interrupt can be configured with a local interrupt number. Bit location of interrupts are the same as their interrupt numbers. Register fields above show the default bit location.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.3.12 Machine Extended Status

Mnemonic Name: mxstatus
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x7c4 (non-standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
PFT_EN	[0]	Enable performance throttling. When throttling is enabled, the processor executes instructions at the performance level specified in `mpft_ctl.T_LEVEL`. On entering a trap: • `PPFT_EN` ⇐ `PFT_EN`; • `PFT_EN` ⇐ `mpft_ctl.FAST_INT` ? 0 : `PFT_EN`; On executing an `MRET` instruction: • `PFT_EN` ⇐ `PPFT_EN`; This field is hardwired to 0 if the PowerBrake feature is not supported.	RW	0
PPFT_EN	[1]	For saving previous `PFT_EN` state on entering a trap. This field is hardwired to 0 if the PowerBrake feature is not supported.	RW	0
IME	[2]	Instruction Machine Error flag. It indicates an ECC exception occurred at the instruction cache or instruction local memory (ILM). Instruction accesses will bypass I-Cache when this bit is set. The exception handler should clear this bit after the machine error has been dealt with. Note that when load/store to ILM access causes ECC error exception, DME is set instead of IME. IME is overwritten by PIME on MRET.	RW	0
PIME	[3]	For saving previous `IME` state on entering a trap. This field is hardwired to 0 if instruction cache and instruction local memory are not supported.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

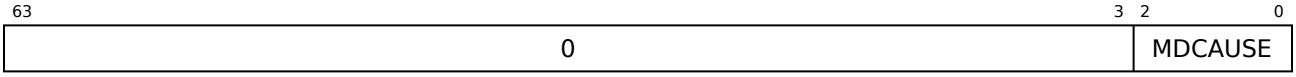
Field Name	Bits	Description	Type	Reset
DME	[4]	Data Machine Error flag. It indicates an ECC exception occurred at the data cache or data local memory (DLM). Load/store accesses will bypass D-Cache when this bit is set. The exception handler should clear this bit after the machine error has been dealt with. Note that when load/store to ILM access causes ECC error exception, DME is set instead of IME. DME is overwritten by PDME on MRET.	RW	0
PDME	[5]	For saving previous DME state on entering a trap. This field is hardwired to 0 if data cache and data local memory are not supported.	RW	0

Official Release

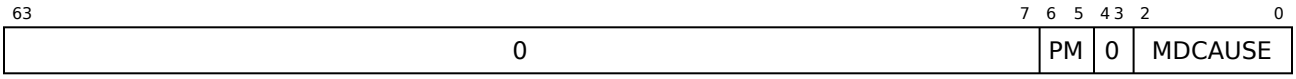
21.3.13 Machine Detailed Trap Cause

Mnemonic Name: mdcause
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x7c9 (non-standard read/write)

For precise exceptions:



For imprecise exceptions (local interrupts):



When multiple events cause traps to be taken with the same mcause value, this register helps to further disambiguate them. Some events could cause either precise exceptions or imprecise exceptions (local interrupts) depending on when they are detected, so they can appear in multiple tables below.

Imprecise exceptions are triggered as local interrupts, so the tables below for mcause == Local Interrupt *n* summarizes imprecise exceptions delivered as local interrupt *n*. The mcause == Local Interrupt *n* notation stands for (INTERRUPT, EXCEPTION_CODE) fields of mcause is (1, *n*).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MDCAUSE	[2:0]	This register further disambiguates causes of traps recorded in the `mcause` register. See the list below for details.	RW	0
PM	[6:5]	When `mcause` is imprecise exception (in the form of an interrupt), the PM field records the current privileged mode. The PM field encoding is defined as follows:	RW	0

Official Release

Value	Meaning
0	User mode
1	Supervisor mode
2	Reserved
3	Machine mode

The value of `MDCAUSE` for precise exception:

- When `mcause` == 1 (Instruction access fault):

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP instruction access violation
3	Bus error
4	PMA empty hole access

- When `mcause` == 2 (Illegal instruction):

Value	Meaning
0	The actual faulting instruction is stored in the `mtval` CSR.
1	FP disabled exception
2	ACE disabled exception
3	Reserved

- When `mcause` == 5 (Load access fault)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP load access violation
3	Bus error
4	Misaligned address
5	PMA empty hole access
6	PMA attribute inconsistency
7	PMA NAMO exception

Official Release

- When `mcause` == 7 (Store access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP store access violation
3	Bus error
4	Misaligned address
5	PMA empty hole access
6	PMA attribute inconsistency
7	PMA NAMO exception

The value of `MDCAUSE` for imprecise exception:

- When `mcause` == *Local Interrupt 16* or *Local Interrupt 272 (16 + 256)* (ECC error local interrupt)

Value	Meaning
0	Reserved
1	LM slave port ECC/Parity error
2	Imprecise store ECC/Parity error
3	Imprecise load ECC/Parity error

- When `mcause` == *Local Interrupt 17* or *Local Interrupt 273 (17 + 256)* (Bus read/write transaction error local interrupt)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Value	Meaning
0	Reserved
1	Bus read error
2	Bus write error
3	PMP error caused by load instructions
4	PMP error caused by store instructions
5	PMA error caused by load instructions"
6	PMA error caused by store instructions

Official Release

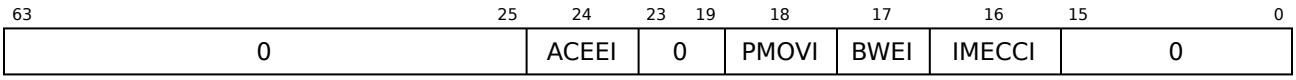
- For PMOVI, MDCAUSE will be written 0. For other exceptions and interrupts, this register will not be updated.

21.3.13.1 Detailed Exception Priority

Within Instruction/Load/Store access fault exceptions, the priority of a PMP exception is higher than the priority of a PMA exception, when both types of exceptions happen on the same instruction.

21.3.14 Machine Supervisor Local Interrupt Delegation

Mnemonic Name: mslideleg
IM Requirement: misa[18]==1
Access Mode: Machine
CSR Address: 0x7D5 (non-standard read/write)



This register controls the delegation of supervisor local interrupts. If a supervisor local interrupt is not delegated, the supervisor local interrupt will be taken in M-mode. If a supervisor local interrupt is delegated, the supervisor local interrupt will be taken in S-mode.

The privileged mode of IMECCI, BWEI, and ACEEI are determined by the current privileged mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

For M/S/U configuration, if the current privileged mode is User or Supervisor, the local interrupt generated is a supervisor local interrupt. For M/U configuration, if the current privileged mode is User, the local interrupt generated is a machine local interrupt. The privileged mode of the above `PMOVI` interrupt is determined by the counter state of `mcountermask_m`.

Each local interrupt can be configured with a local interrupt number. Bit location of interrupts are the same as their interrupt numbers. Register fields below show the default bit location.

Official Release

Field Name	Bits	Description	Type	Reset
IMECCI	[16]	Delegate S-mode imprecise ECC error local interrupt to S-mode. **Value** / **Meaning** 0 — Do not delegate to S-mode. 1 — Delegate to S-mode.	RW	0
BWEI	[17]	Delegate S-mode bus read/write transaction error local interrupt to S-mode. **Value** / **Meaning** 0 — Do not delegate to S-mode. 1 — Delegate to S-mode.	RW	0
PMOVI	[18]	Delegate S-mode performance monitor overflow local interrupt to S-mode. **Value** / **Meaning** 0 — Do not delegate to S-mode. 1 — Delegate to S-mode. Delegate S-mode ACE error local interrupt to S-mode. **Value** / **Meaning** 0 — Do not delegate to S-mode. 1 — Delegate to S-mode.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.4 Counter Related CSRs

21.4.1 Machine Cycle Counter

Mnemonic Name: mcycle
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xb00 (standard read/write)

Official Release

The `mcycle` CSR counts the number of cycles that the hart has executed since some arbitrary time in the past. The `mcycle` register has 64-bit precision.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.4.2 Machine Instruction-Retired Counter

Mnemonic Name: minstret
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xb02 (standard read/write)

Official Release

The `minstret` CSR counts the number of instructions that the hart has retired since some arbitrary time in the past. The `minstret` register has 64-bit precision.

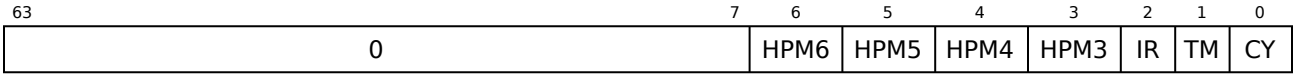
21.4.3 Machine Performance Monitoring Counter

Mnemonic Name: mhpmcounter3–mhpmcounter6
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xb03 to 0xb06 (standard read/write)

The `mhpmcounter3-mhpmcounter6` CSRs count the number of events selected by `mhpmevent3-mhpmevent6`.

21.4.4 Machine Counter-Inhibit

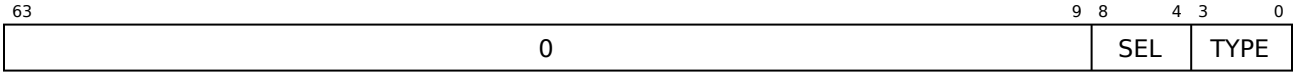
Mnemonic Name: mcountinhibit
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x320 (non-standard read/write)



The counter-inhibit register controls which counters should not be incremented. When the `CY`, `IR`, or `HPMn` bit is set, the corresponding counter will not be incremented on the event.

21.4.5 Machine Performance Monitoring Event Selector

Mnemonic Name: mhpmevent3–mhpmevent6
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x323 to 0x326 (standard read/write)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The event selectors are defined in Table 155. Micro-architectural events are mostly speculative in nature. The counted events include events caused by speculative actions, unless they are defined to be non-speculative in the comment section. In particular, *retired* instruction counts are non-speculative.

Table 155: Event Selectors

Official Release

TYPE	SEL	Event Name	Comment
0	1	Cycle count	Number of elapsed processor clock cycles.
0	2	Retired instruction count	Number of retired instructions.
0	3	Integer load instruction count	Number of retired load instructions (including `LR`).
0	4	Integer store instruction count	Number of retired store instructions (including `SC`).
0	5	Atomic instruction count	Number of retired atomic instructions (not including `LR` and `SC`).
0	6	System instruction count	Number of retired SYSTEM instructions (instructions with major opcode equal to 0b1110011).
0	7	Integer computational instruction count	Number of retired integer computational instructions.
0	8	Conditional branch instruction count	Number of retired conditional branch instructions.
0	9	Taken conditional branch instruction count	Number of retired conditional branch instructions that are taken.
0	10	`JAL` instruction count	Number of retired `JAL` instructions.
0	11	`JALR` instruction count	Number of retired `JALR` instructions. This event selector also counts the events monitored by the *return instruction count* event selector defined in the next row.
0	12	Return instruction count	Number of retired return instructions. Return instructions are `JALR` instructions with zero immediate offset and the following operands: • (rd != x1/x5) and (rs1 == x1/x5) • rd == x1 and rs1 == x5 • rd == x5 and rs1 == x1
0	13	Control transfer instruction count	Number of retired unconditional jumps (`JAL` and `JALR`) and conditional branch instructions.
0	14	`EXEC.IT` instruction count	Number of retired `EXEC.IT` instructions.
0	15	Integer multiplication instruction count	Number of retired integer multiplication instructions.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 155: (continued)

TYPE	SEL	Event Name	Comment
0	16	Integer division instruction count	Number of retired integer division/remainder instructions.
0	17	Floating-point load instruction count	Number of retired floating-point load instructions.
0	18	Floating-point store instruction count	Number of retired floating-point store instructions.
0	19	Floating-point addition instruction count	Number of retired floating-point addition/subtraction instructions.
0	20	Floating-point multiplication instruction count	Number of retired floating-point multiplication instructions.
0	21	Floating-point fused multiply-add instruction count	Number of retired floating-point fused multiply-add/subtraction instructions (`FMADD`, `FMSUB`, `FNMSUB`, `FNMADD`).
0	22	Floating-point division or square-root instruction count	Number of retired floating-point division/square-root instructions.
0	23	Other floating-point instruction count	Number of retired floating-point instructions not counted by the previous floating-point instruction event selectors.
0	24	Integer multiplication and add/sub instruction count	Number of retired integer multiplication and add/sub instructions.
0	25	Retired operation count	Number of retired operations. • a floating-point multiply-add instruction is counted as 2 operations. • other instructions are counted as 1 operation.
1	0	ILM access	Number of ILM transfers, including speculative instruction fetch, load/store accesses, ECC repair and slave port accesses.
1	1	DLM access	Number of DLM transfers, including speculative load/store accesses, ECC repair and slave port accesses.
1	2	I-Cache access	Number of completed I-Cache fetch access.
1	3	I-Cache miss	Number of I-Cache fetch miss.

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 155: (continued)

TYPE	SEL	Event Name	Comment
1	4	D-Cache access*	Number of completed D-Cache load-and-store access. Misaligned load/store accesses might increase this counter by either one or two, depending on access sizes and alignments. Only misaligned accesses crossing two cache lines are guaranteed to result in increment of two.
1	5	D-Cache miss*	The event counts the number of D-Cache load-and-store miss. Misaligned load/store accesses might increase this counter by either zero, one or two, depending on access sizes, alignments and whether the accessed lines are in D-Cache.
1	6	D-Cache load access*	Number of completed D-Cache load access. See the D-Cache access count event selector for the handling of misaligned load accesses.
1	7	D-Cache load miss*	Number of D-Cache load miss. See the D-Cache miss count event selector for the handling of misaligned load accesses.
1	8	D-Cache store access*	Number of completed D-Cache store access. See the D-Cache access count event selector for the handling of misaligned load accesses.
1	9	D-Cache store miss*	Number of D-Cache store miss. See the D-Cache miss count event selector for the handling of misaligned load accesses.
1	10	D-Cache writeback*	Number of D-Cache writeback.
1	11	Cycles waiting for I-Cache fill data*	Number of cycles waiting for the return of the critical word of I-Cache misses from the system bus. This event selector does not monitor accesses to I/O regions or accesses to cacheable regions when I-Cache is turned off.

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 155: (continued)

TYPE	SEL	Event Name	Comment
1	12	Cycles waiting for D-Cache fill data*	Number of cycles waiting for the return of the critical word of D-Cache misses from the system bus. This event selector does not monitor accesses to I/O regions or accesses to cacheable regions when D-Cache is turned off. Additionally, non-blocking loads do not increment this counter since they do not cause pipeline stalls under D-Cache misses.
1	13	Uncached fetch data access from bus*	Number of accesses of uncached instruction data returning from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when I-Cache is not configured or off.
1	14	Uncached load data access from bus*	Number of accesses of uncached load data returning from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when D-Cache is not configured or off.
1	15	Cycles waiting for uncached fetch data from bus*	Number of cycles waiting for the instruction data to return from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when I-Cache is not configured or off.
1	16	Cycles waiting for uncached load data from bus*	Number of cycles waiting for the load data to return from the system bus. This event selector monitors accesses to I/O regions or accesses to cacheable regions when D-Cache is not configured or off.
1	17	Main ITLB access	Number of address translation requests performed by the shared TLB for instruction fetches
1	18	Main ITLB miss	Number of address translation requests from instruction fetch that miss Shared TLB and invoke the hardware page table walker.
1	19	Main DTLB access	Number of address translation requests performed by the shared TLB for load/store accesses

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 155: (continued)

TYPE	SEL	Event Name	Comment
1	20	Main DTLB miss	Number of address translation requests from load/store accesses that miss Shared TLB and invoke the hardware page table walker.
1	21	Cycles waiting for Main ITLB fill data	Number of instruction fetch stall cycles attributable to TLB misses.
1	22	Pipeline stall cycles caused by Main DTLB miss	Number of pipeline stall cycles attributable to address translation for load/store accesses.
1	23	Hardware prefetch bus access	This event counts the bus accesses generated by the hardware data prefetcher.
1	24	Cycles waiting for source operand ready in the integer register file scoreboard	Number of cycles waiting for source operand ready in the integer register file scoreboard. This event monitors the waiting cycles caused by nonblocking execution instructions. The event is present for CPU revision 10.0.0 and later.
2	0	Misprediction of conditional branches (direction)	Number of misprediction of committed conditional branches.
2	1	Misprediction of taken conditional branches (direction)	Number of misprediction of committed taken conditional branches.
2	2	Misprediction of targets of Return instructions	Number of misprediction of committed Return instruction.
3	0-31	MSHR entry accumulated miss counts	Number of miss counts in the selected MSHR entry. If the SEL number is a nonexistent MSHR entry, the counter value will not change. For the MSHR entry number configuration, please see Section 2.10.2. The event is present for CPU revision 10.0.0 and later.
4	0-31	MSHR entry accumulated miss latencies	Number of cycles waiting filling data in the selected MSHR entry. If the SEL number is a nonexistent MSHR entry, the counter value will not change. For the MSHR entry number configuration, please see Section 2.10.2. The event is present for CPU revision 10.0.0 and later.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

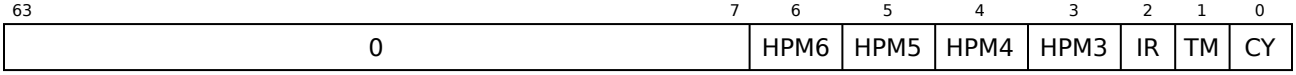
Note

- Interrupts are expected to be disabled when monitoring D-Cache related events and cycles waiting related events.

Official Release

21.4.6 Machine Counter Enable

Mnemonic Name: mcounteren
IM Requirement: Required if User mode is implemented
Access Mode: Machine
CSR Address: 0x306 (standard read/write)

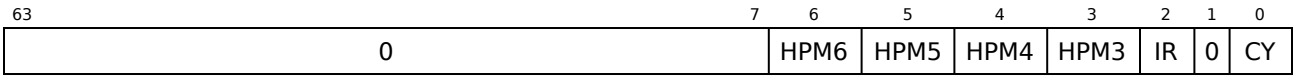


The machine counter-enable register controls the availability of the hardware performance monitoring counters to the next-lowest privileged mode. The default value of this register is 0.

When `CY`, `TM`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `mcounteren` register is 0, attempts to read the `cycle`, `time`, `instret`, `hpmcounter3`, `hpmcounter4`, `hpmcounter5`, or `hpmcounter6` registers while executing in U-mode (M/U configuration) or S-mode (M/S/U configuration) will cause an illegal instruction exception. When one of these bits is set, accessing to the corresponding register is permitted in the next implemented privilege mode.

21.4.7 Machine Counter Write Enable

Mnemonic Name: mcounterwen
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[20] == 1
Access Mode: Machine
CSR Address: 0x7CE (non-standard read/write)



The machine counter write enable register controls the permission of writing the hardware performance monitoring counters in the next-lowest privileged mode and M-mode itself. The default value of this register is 0.

When `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `mcounterwen` register is 0, attempts to write the `cycle`, `time`, `instret`, `hpmcounter3`, `hpmcounter4`, `hpmcounter5`, or `hpmcounter6` registers while executing in U-mode or M-mode (M/U configuration) or S-mode (M/S/U configuration) will cause an illegal instruction exception. When one of these bits is set, writing to the corresponding register is permitted in M-mode and the next implemented privilege mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

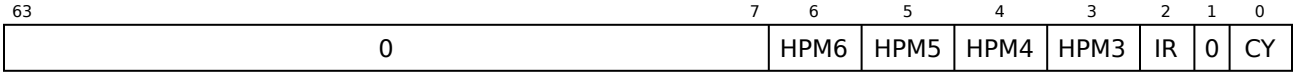
Note

The register does not affect permissions of M-Mode counters, i.e. `mcycle`, `mtime`, `minstret`, `mhpmcounter3`, `mhpmcounter4`, and `mhpmcounter5`.

Official Release

21.4.8 Machine Counter Interrupt Enable

Mnemonic Name: mcounterinten
IM Requirement: mmsc_cfg.PMNDS == 1
Access Mode: Machine
CSR Address: 0x7CF (non-standard read/write)

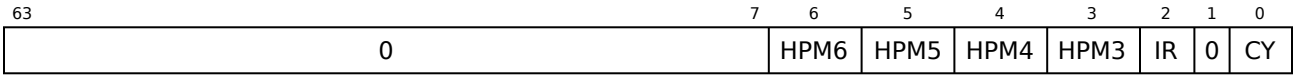


The machine counter interrupt enable register controls whether a counter overflow interrupt is generated or not. The default value of this register is 0.

When `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `mcounterinten` register is 0, no overflow interrupt is generated for the corresponding counter. When one of these bits is set, an interrupt will be generated when the corresponding counter overflows (the counter value wraps around back to 0).

21.4.9 Machine Counter Mask for Machine Mode

Mnemonic Name: mcountermask_m
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[20] == 1
Access Mode: Machine
CSR Address: 0x7D1 (non-standard read/write)



The machine counter mask for M-mode register controls the performance counter behavior in M-mode. The default value of this register is 0.

When `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `mcountermask_m` register is set, the specific counter will not be incremented in M-mode.

The setting in this register also controls the privileged mode of the overflow local interrupt when the corresponding counter overflows for the M /S /U configuration: For any bit in this register, overflow of the corresponding counter will trigger an M-mode interrupt if the bit is zero and an S-mode interrupt if the bit is one .

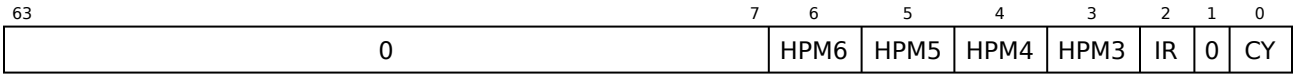
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

On the other hand, a counter overflow will always generate an M-mode interrupt for the M/U configuration, regardless of the settings in this register.

21.4.10 Machine Counter Mask for Supervisor Mode

Official Release

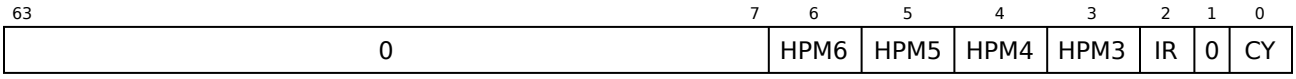
Mnemonic Name: mcountermask_s
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[18] == 1
Access Mode: Machine
CSR Address: 0x7D2 (non-standard read/write)



The machine counter mask for S-mode register controls the performance counter behavior in S-mode. The default value of this register is 0.

21.4.11 Machine Counter Mask for User Mode

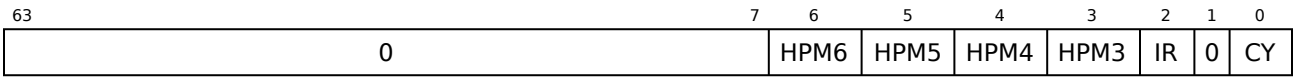
Mnemonic Name: mcountermask_u
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[20] == 1
Access Mode: Machine
CSR Address: 0x7D3 (non-standard read/write)



The machine counter mask for U-mode register controls the performance counter behavior in U-mode. The default value of this register is 0.

21.4.12 Machine Counter Overflow Status

Mnemonic Name: mcounterovf
IM Requirement: mmsc_cfg.PMNDS == 1
Access Mode: Machine
CSR Address: 0x7D4 (non-standard read/write)



The machine counter overflow status register records the overflow status of performance counters. When a bit is set, it indicates that an overflow has happened to the corresponding counter. Writing 0 will deassert the corresponding bit and writing 1 will have no effect.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.5 Configuration Control & Status Registers

21.5.1 Instruction Cache/Memory Configuration Register

Mnemonic Name: micm_cfg
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xfc0 (non-standard read only)

Official Release

63 27	26 25	24	23	22 21	20	19 15	14 12	11 10	9	8 6	5 3	2 0
0	IC_REPL	SETH	0	ILM_ECC	0	ILMSZ	ILMB	IC_ECC	ILCK	ISZ	IWAY	ISET

This register provides information about configurations of instruction cache and instruction memory.

Field Name	Bits	Description	Type	Reset
ISET	[2:0]	I-Cache sets (# of cache lines per way): When `micm_cfg.SETH`==0:	RO	IM

Value	Meaning
0	64
1	128
2	256
3	512
4	1024
5	2048
6	4096
7	Reserved

When `micm_cfg.SETH`==1:

Value	Meaning
0	32
1	16
2	8
3~7	Reserved

- When instruction cache is not configured, this field should be ignored.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IWAY	[5:3]	Associativity of I-Cache	RO	IM

Value	Meaning
0	Direct-mapped
1	2-way
2	3-way
3	4-way
4	5-way
5	6-way
6	7-way
7	8-way

- When instruction cache is not configured, this field should be ignored.

Official Release

Field Name	Bits	Description	Type	Reset
ISZ	[8:6]	I-Cache block (line) size	RO	IM

Value	Meaning
0	No I-Cache
1	8 bytes
2	16 bytes
3	32 bytes
4	64 bytes
5	128 bytes
6,7	Reserved

- When instruction cache is not configured, this field should be ignored.

Field Name	Bits	Description	Type	Reset
ILCK	[9]	I-Cache locking support	RO	IM

Value	Meaning
0	No locking support
1	With locking support

- When instruction cache is not configured, this field should be ignored.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IC_ECC	[11:10]	I-Cache soft-error protection scheme **Value** / **Meaning** 0 — No parity/ECC 1 — Parity 2 — ECC 3 — Reserved • When instruction cache is not configured, this field should be ignored.	RO	IM
ILMB	[14:12]	Number of ILM base registers present **Value** / **Meaning** 0 — No ILM base register present 1 — One ILM base register present 2-7 — Reserved • When ILM is not configured, this field should be ignored.	RO	IM

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
ILMSZ	[19:15]	ILM Size **Value** / **Meaning** 0 — 0 Byte 1 — 1 KiB 2 — 2 KiB 3 — 4 KiB 4 — 8 KiB 5 — 16 KiB 6 — 32 KiB 7 — 64 KiB 8 — 128 KiB 9 — 256 KiB 10 — 512 KiB 11 — 1 MiB 12 — 2 MiB 13 — 4 MiB 14 — 8 MiB 15 — 16 MiB 16-31 — Reserved • When ILM is not configured, this field should be ignored.	RO	IM
ILM_ECC	[22:21]	ILM soft-error protection scheme **Value** / **Meaning** 0 — No parity/ECC 1 — Parity 2 — ECC 3 — Reserved • When ILM is not configured, this field should be ignored.	RO	IM
SETH	[24]	This bit extends the `ISET` field. • When instruction cache is not configured, this field should be ignored.	RO	IM

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IC_REPL	[26:25]	Indicates I-Cache replacement policy **Value** / **Meaning** 0 — Unknown 1 — **I-Cache Replacement Policy** is pseudo-lru 2 — **I-Cache Replacement Policy** is random 3 — Reserved • When I-Cache is not configured, this field should be ignored.	RO	IM

Official Release

21.5.2 Data Cache/Memory Configuration Register

Mnemonic Name: mdcm_cfg
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xfc1 (non-standard read only)

63 27	26 25	24	23	22 21	20	19 15	14 12	11 10	9	8 6	5 3	2 0
0	DC_REPL	SETH	0	DLM_ECC	0	DLMSZ	DLMB	DC_ECC	DLCK	DSZ	DWAY	DSET

This register provides information about the configurations of data cache and data local memory.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
DSET	[2:0]	D-Cache sets (# of cache lines per way): When `mdcm_cfg.SETH`==0: **Value** / **Meaning** 0 / 64 1 / 128 2 / 256 3 / 512 4 / 1024 5 / 2048 6 / 4096 7 / Reserved When `mdcm_cfg.SETH`==1: **Value** / **Meaning** 0 / 32 1 / 16 2 / 8 3~7 / Reserved • When data cache is not configured, this field should be ignored.	RO	IM
DWAY	[5:3]	Associativity of D-Cache **Value** / **Meaning** 0 / Direct-mapped 1 / 2-way 2 / 3-way 3 / 4-way 4 / 5-way 5 / 6-way 6 / 7-way 7 / 8-way • When data cache is not configured, this field should be ignored.	RO	IM

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DSZ	[8:6]	D-Cache block (line) size **Value** / **Meaning** 0 — No D-Cache 1 — 8 bytes 2 — 16 bytes 3 — 32 bytes 4 — 64 bytes 5 — 128 bytes 6,7 — Reserved • When data cache is not configured, this field should be ignored.	RO	IM
DLCK	[9]	D-Cache locking support **Value** / **Meaning** 0 — No locking support 1 — With locking support • When data cache is not configured, this field should be ignored.	RO	IM
DC_ECC	[11:10]	D-Cache soft-error protection scheme **Value** / **Meaning** 0 — No parity/ECC support 1 — Has parity support 2 — Has ECC support 3 — Reserved • When data cache is not configured, this field should be ignored.	RO	IM
DLMB	[14:12]	Number of DLM base registers present **Value** / **Meaning** 0 — No DLM base register present 1 — One DLM base register present 2-7 — Reserved • When DLM is not configured, this field should be ignored.	RO	IM

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DLMSZ	[19:15]	DLM Size **Value** / **Meaning** 0 / 0 Byte 1 / 1 KiB 2 / 2 KiB 3 / 4 KiB 4 / 8 KiB 5 / 16 KiB 6 / 32 KiB 7 / 64 KiB 8 / 128 KiB 9 / 256 KiB 10 / 512 KiB 11 / 1 MiB 12 / 2 MiB 13 / 4 MiB 14 / 8 MiB 15 / 16 MiB 16-31 / Reserved • When DLM is not configured, this field should be ignored.	RO	IM
DLM_ECC	[22:21]	DLM soft-error protection scheme **Value** / **Meaning** 0 / No parity/ECC 1 / Parity 2 / ECC 3 / Reserved • When DLM is not configured, this field should be ignored.	RO	IM
SETH	[24]	This bit extends the `DSET` field. • When D-Cache is not configured, this field should be ignored.	RO	IM

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DC_REPL	[26:25]	Indicates D-Cache replacement policy **Value** / **Meaning** 0 — Unknown 1 — **D-Cache Replacement Policy** is pseudo-lru 2 — **D-Cache Replacement Policy** is random 3 — Reserved • When D-Cache is not configured, this field should be ignored.	RO	IM

Official Release

21.5.3 Misc. Configuration Register

Mnemonic Name: mmsc_cfg
IM Requirement: Required
Access Mode: Machine
CSR Address: 0xfc2 (non-standard read only)

14	13	12	11 … 7	6	5	4	3	2 … 1	0
LMSLVP	EV5PE	VPLIC	0	ACE	HSP	PFT	ECD	TLB_ECC	ECC

37	36 34	33	32	31	30	29	28 26	25	24 23	22	21 20	19	18	17	16 … 15
FINV	0	ZFH	BF16CVT	0	PPMA	EDSP	0	0	0	NOPMC	0	VCCTL	EFHW	CCTLCSR	PMNDS

63 … 54	53 … 52	51 … 48	47	46	45 … 44	43 … 42	41 … 40	39 … 38
0	TLB_RAM_CMD	RVARCH	CORE_PCLUS	IOCP	L2C	L2CMP_CFG	0	0	0	0

This register provides information regarding miscellaneous processor configurations.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
ECC	[0]	Indicates whether the parity/ECC soft-error protection is implemented or not. **Value** / **Meaning** 0 — Not implemented. 1 — Implemented. The specific parity/ECC scheme used for each protected RAM is specified by the control bits in the following list. • `micm_cfg.IC_ECC` • `micm_cfg.ILM_ECC` • `mdcm_cfg.DC_ECC` • `mdcm_cfg.DLM_ECC` • `mmsc_cfg.TLB_ECC`	RO	IM
TLB_ECC	[2:1]	TLB parity/ECC support configuration. **Value** / **Meaning** 0 — No parity/ECC support. 1 — Reserved. 2 — Has ECC support. 3 — Reserved. When **Shared TLB Soft Error Protection** is configured to “none”, the field is 0. When **Shared TLB Soft Error Protection** is configured to “ecc”, the field is 2.	RO	IM
ECD	[3]	Indicates whether the Andes CoDense Extension is implemented or not. **Value** / **Meaning** 0 — Not implemented. 1 — Implemented.	RO	1
PFT	[4]	Indicates whether the Andes PowerBrake (Performance Throttling) power/performance scaling extension is implemented or not. **Value** / **Meaning** 0 — Not implemented. 1 — Implemented.	RO	IM

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
HSP	[5]	Indicates whether the Andes StackSafe hardware stack protection extension is implemented or not. **Value** / **Meaning** 0 — Not implemented. 1 — Implemented.	RO	IM
ACE	[6]	Indicates whether Andes Custom Extension is implemented or not. **Value** / **Meaning** 0 — Not implemented. 1 — Implemented.	RO	IM
VPLIC	[12]	Indicates whether the Andes Vectored PLIC Extension is implemented or not. **Value** / **Meaning** 0 — Not implemented. 1 — Implemented.	RO	IM
EV5PE	[13]	Indicates whether AndeStar V5 Performance Extension is implemented or not. AX45MP always implements AndeStar V5 Performance Extension.	RO	1
LMSLVP	[14]	Indicates if local memory slave port is present or not. **Value** / **Meaning** 0 — Local memory slave port is not present. 1 — Local memory slave port is implemented. Note that atomicity of atomic instructions accessing local memory address space is not guaranteed if external masters modify the same data through the local memory slave port.	RO	IM

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
PMNDS	[15]	Indicates if Andes performance monitor extension is present or not. This extension should be present when Performance Monitors are configured. **Value** / **Meaning** 0 — Andes-enhanced performance monitoring feature is not supported. 1 — Andes-enhanced performance monitoring feature is supported.	RO	IM
CCTLCSR	[16]	Indicates the presence of CSRs for CCTL operations. **Value** / **Meaning** 0 — Feature of CSRs for CCTL operations is not supported. 1 — Feature of CSRs for CCTL operations is supported.	RO	IM
EFHW	[17]	Indicates the support of `FLHW` and `FSHW` instructions. **Value** / **Meaning** 0 — `FLHW` and `FSHW` instructions are not supported. 1 — `FLHW` and `FSHW` instructions are supported.	RO	IM
VCCTL	[19:18]	Indicates the version number of CCTL command operation scheme supported by an implementation. **Value** / **Meaning** 0 — Instruction cache and data cache are not configured. 1 — Instruction cache or data cache is configured.	RO	IM

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
NOPMC	[22]	Indicates if Performance Monitoring Counters are implemented or not. When Hardware Performance Monitors is not configured, the counter will be hardwired to 1. **Value** / **Meaning** 0 — Performance monitoring counters are implemented. 1 — All performance monitoring counters CSRs are hardwired to 0	RO	IM
EDSP	[29]	Indicates if the DSP extension is supported or not. **Value** / **Meaning** 0 — The DSP extension is not supported. 1 — The DSP extension is supported.	RO	IM
PPMA	[30]	Indicates if programmable PMA setup with PMA region CSRs is supported or not. **Value** / **Meaning** 0 — Programmable PMA setup is not supported. 1 — Programmable PMA setup is supported.	RO	IM
BF16CVT	[32]	Indicates if the BFLOAT16 conversion extension is supported or not. **Value** / **Meaning** 0 — The BFLOAT16 conversion extension is not supported 1 — The BFLOAT16 conversion extension is supported	RO	IM

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
ZFH	[33]	Indicates if the FP16 half-precision floating-point extension (Zfh) is supported or not. **Value** / **Meaning** 0 — The FP16 extension is not supported. 1 — The FP16 extension is supported.	RO	IM
FINV	[37]	Indicates if scalar FPU is implemented in VPU. **Value** / **Meaning** 0 — Scalar FPU is not implemented in VPU. 1 — Scalar FPU is implemented in VPU.	RO	IM
L2CMP_CFG	[45]	Indicates whether cluster configuration info is implemented or not. **Value** / **Meaning** 0 — L2C, IOCP, CORE_PCLUS is not implemented 1 — L2C, IOCP, CORE_PCLUS is implemented	RO	IM
L2C	[46]	Indicates L2-Cache is present or not. **Value** / **Meaning** 0 — L2-Cache is not present 1 — L2-Cache is present	RO	IM
IOCP	[47]	Indicates IO Coherence Port is present or not. **Value** / **Meaning** 0 — IOCP is not present 1 — IOCP is present	RO	IM

Continued on next page...

Official Release

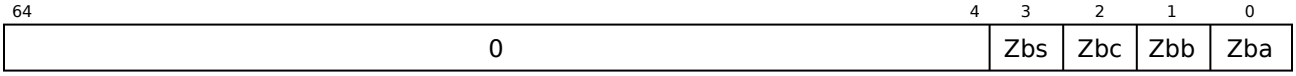
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
CORE_PCLUS	[51:48]	Indicates the number of cores in a cluster. Value: 0 — Meaning: 1 core Value: 1 — Meaning: 2 cores Value: 3 — Meaning: 4 cores Value: 7 — Meaning: 8 cores	RO	IM
RVARCH	[52]	Indicates if `mrvarch_cfg` CSR is present or not. Value: 0 — Meaning: `mrvarch_cfg` CSR is not present. Value: 1 — Meaning: `mrvarch_cfg` CSR is present.	RO	IM
TLB_RAM_CMD	[53]	Indicates the presence of TLB RAM command for index type of CCTL TLB operations. Value: 0 — Meaning: Index type of CCTL TLB operations is not supported. Value: 1 — Meaning: Index type of CCTL TLB operations is supported. When **Shared TLB Soft Error Protection** is configured to “none”, the field is 0. When **Shared TLB Soft Error Protection** is configured to “ecc”, the field is 1.	RO	IM

Official Release

21.5.4 RISC-V Architecture Configuration Register

Mnemonic Name: mrvarch_cfg
IM Requirement: mmsc_cfg.RVARCH==1
Access Mode: Machine
CSR Address: 0xfca (non-standard read only)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

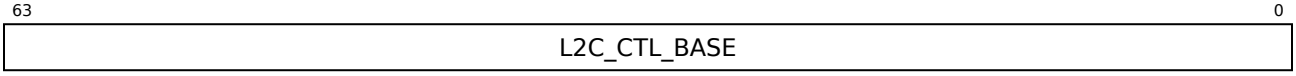
This register provides information regarding RISC-V Architecture. Note that the below table shows all valid values, but some of them may not be implemented.

Field Name	Bits	Description	Type	Reset
Zba	[0]	Indicates the RISC-V Zba ISA extension is implemented or not. **Value** / **Meaning** 0 – Zba is not implemented. 1 – Zba is implemented.	RO	IM
Zbb	[1]	Indicates the RISC-V Zbb ISA extension is implemented or not. **Value** / **Meaning** 0 – Zbb is not implemented. 1 – Zbb is implemented.	RO	IM
Zbc	[2]	Indicates the RISC-V Zbc ISA extension is implemented or not. **Value** / **Meaning** 0 – Zbc is not implemented. 1 – Zbc is implemented.	RO	IM
Zbs	[3]	Indicates the RISC-V Zbs ISA extension is implemented or not. **Value** / **Meaning** 0 – Zbs is not implemented. 1 – Zbs is implemented.	RO	IM

Official Release

21.5.5 L2-Cache Control Base Register

Mnemonic Name: ml2c_ctl_base
IM Requirement: mmsc_cfg.L2C
Access Mode: Machine
CSR Address: 0xfcf (non-standard read only)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

This register indicates the base address of the memory-mapped L2-Cache control registers.

Field Name	Bits	Description	Type	Reset
L2C_CTL_BASE	[63:0]	Indicates **L2C Register Base**	RO	IM

Official Release

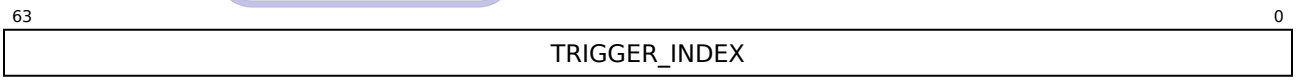
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.6 Trigger Registers

21.6.1 Trigger Select

Mnemonic Name: tselect
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a0 (standard read/write)

Official Release

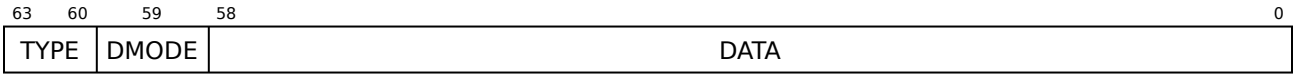


This register determines which trigger is accessible through other trigger registers. The setting of accessible triggers must start at 0, and be contiguous. Writes of values greater than or equal to the number of supported triggers might result in a different value in this register than what was written. Debuggers should read back the value to confirm that what they wrote was a valid index.

Since triggers can be used by both Debug mode and Machine mode, the debugger must restore this register after the modification.

21.6.2 Trigger Data 1

Mnemonic Name: tdata1
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a1 (standard read/write)



This register provides access to the `tdata1` register of the currently selected trigger registers selected by the `tselect` register.

Field Name	Bits	Description	Type	Reset
DATA	[58:0]	Trigger-specific data	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DMODE	[59]	Setting this field to indicate the trigger is used by Debug Mode. **Value** / **Meaning** 0 — Both Debug-mode and M-mode can write the currently selected trigger registers. 1 — Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.	RW	0
TYPE	[63:60]	Indicates the trigger type. **Value** / **Meaning** 0 — The selected trigger is invalid. 2 — The selected trigger is an address/data match trigger. 3 — The selected trigger is an instruction count trigger. 4 — The selected trigger is an interrupt trigger. 5 — The selected trigger is an exception trigger.	RW	2

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.6.3 Trigger Data 2

Mnemonic Name: tdata2
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a2 (standard read/write)

Official Release

This register provides accesses to the `tdata2` register of the currently selected trigger registers selected by the `tselect` register, and it holds trigger-specific data.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.6.4 Trigger Data 3

Mnemonic Name: tdata3
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a3 (standard read/write)

Official Release

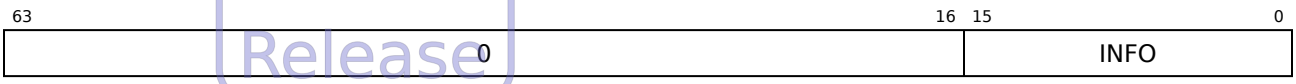
This register provides access to the `tdata3` register of the currently selected trigger registers selected by the `tselect` register, and it holds trigger-specific data.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.6.5 Trigger Info

Mnemonic Name: tinfo
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a4 (standard read/write)

Official Release



This register provides accesses to the `tinfo` register of the currently selected trigger registers selected by the `tselect` register, and it indicates the supported trigger types of the currently selected trigger.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
INFO	[15:0]	One bit for each possible type in `tdata1`. Bit *N* corresponds to type *N*. If the bit is set, then that type is supported by the currently selected trigger. If the currently selected trigger does not exist, this field contains 1.	RO	IM

Official Release

Bit *N*	Descriptions
0	When this bit is set, there is no trigger at this `tselect`.
1	Reserved and hardwired to 0.
2	When this bit is set, the selected trigger supports type of address/data match trigger.
3	When this bit is set, the selected trigger supports type of instruction count trigger.
4	When this bit is set, the selected trigger supports type of interrupt trigger.
5	When this bit is set, the selected trigger supports type of exception trigger.
15	When this bit is set, the selected trigger exists (so enumeration shouldn't terminate), but is not currently available.
Others	Reserved for future use.

The detailed correlation between trigger types and Number of Trigger (Section 2.11.3) are as follows:

- INFO[2] = 1, all trigger types are supported.
- INFO[3] = 1, trigger 0 or 1 (`tselect` = 0 or 1) is supported.
- INFO[4] = 1, trigger 0 (`tselect` = 0) or trigger 4 (`tselect` = 4) is supported when Number of Triggers is 8.
- INFO[5] = 1,
  - trigger 1 (`tselect` = 1) is supported when Number of Triggers is 2,

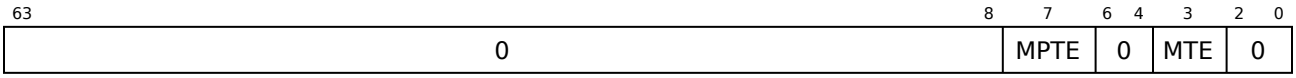
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

– trigger 3 (`tselect` = 3) is supported when Number of Triggers is 4, or
– trigger 3 or 7 (`tselect` = 3 or 7) is supported when Number of Triggers is 8.

21.6.6 Trigger Control

Official Release

Mnemonic Name: tcontrol
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a5 (standard read/write)



This register provides accesses to the `tcontrol` register, and it indicates the current native M-Mode debugging settings.

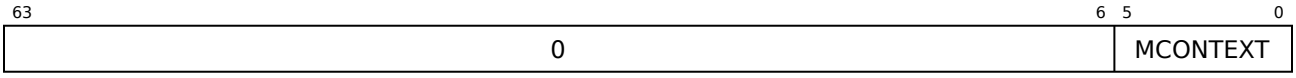
Field Name	Bits	Description	Type	Reset
MTE	[3]	M-mode trigger enable field. When a trap into M-mode is taken, `MTE` is set to 0. When the MRET instruction is executed, `MTE` is set to the value of `MPTE`.	RW	0
MPTE	[7]	M-mode previous trigger enable field. When a trap into M-mode is taken, `MPTE` is set to the value of `MTE`.	RW	0

Value	Meaning
0	Triggers do not match/fire while the hart is in M-mode.
1	Triggers do match/fire while the hart is in M-mode.

21.6.7 Machine Context

Mnemonic Name: mcontext
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a8 (standard read/write)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



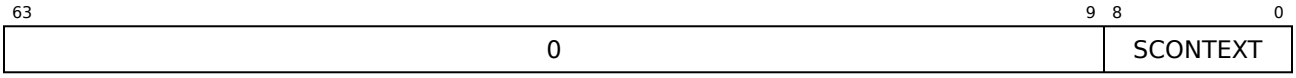
This register provides access to the `mcontext` register.

Field Name	Bits	Description	Type	Reset
MCONTEXT	[5:0]	Machine mode software can write a context number to this register, which can be used to set triggers that only fire in that specific context.	RW	0

Official Release

21.6.8 Supervisor Context

Mnemonic Name: scontext
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug, Machine, Supervisor
CSR Address: 0x7aa (standard read/write)



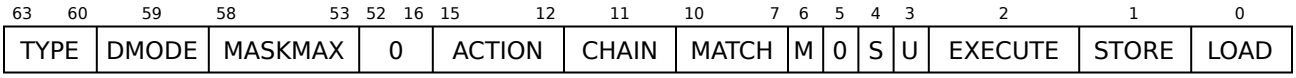
This register provides access to the `scontext` register.

Field Name	Bits	Description	Type	Reset
SCONTEXT	[8:0]	Machine mode software can write a context number to this register, which can be used to set triggers that only fire in that specific context.	RW	0

21.6.9 Match Control

Mnemonic Name: mcontrol
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a1 (standard read/write)

This register is accessible as `tdata1` when `TYPE` is 0 or 2.



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
LOAD	[0]	Setting this field to enable this trigger to compare virtual address of a load.	RW*	0
STORE	[1]	Setting this field to enable this trigger to compare virtual address of a store.	RW*	0
EXECUTE	[2]	Setting this field to enable this trigger to compare virtual address of an instruction.	RW	0
U	[3]	Setting this field to enable this trigger in U-mode.	RW	0
S	[4]	Setting this field to enable this trigger in S-mode.	RW	0
M	[6]	Setting this field to enable this trigger in M-mode.	RW	0
MATCH	[10:7]	Setting this field to select the matching scheme.	RW	0

Value	Meaning
0	Matches when the value equals `tdata2`.
1	Matches when the top *M* bits of the value match the top *M* bits of `tdata2`. *M* is 63 minus the index of the least-significant bit containing 0 in `tdata2`.
2	Matches when the value is greater than (unsigned) or equal to `tdata2`.
3	Matches when the value is less than (unsigned) `tdata2`.

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
CHAIN	[11]	Setting this field to enable trigger chain. **Value** / **Meaning** 0 — When this trigger matches, the configured action is taken. 1 — While this trigger does not match, it prevents the trigger with the next index from matching. If Number of Triggers is 2, this field is hardwired to 0 on trigger 1 (`tselect` = 1). If Number of Triggers is 4, this field is hardwired to 0 on trigger 3 (`tselect` = 3). If Number of Triggers is 8, this field is hardwired to 0 on trigger 3 and trigger 7 (`tselect` = 3 or 7).	RW	0
ACTION	[15:12]	Setting this field to select what happens when this trigger matches. **Value** / **Meaning** 0 — Raise a breakpoint exception. 1 — Enter Debug Mode. The value is legal only when `DMODE` is 1. 2 — Trace-on. The value is legal only when the trace interface is configured. 3 — Trace-off. The value is legal only when the trace interface is configured. 4 — Trace-notify. The value is legal only when the trace interface is configured.	RW	0
MASKMAX	[58:53]	Indicates the largest naturally aligned range supported by the hardware is 2^12 bytes.	RO	12

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DMODE	[59]	Setting this field to indicate the trigger is used by Debug Mode. **Value** / **Meaning** 0 — Both Debug-mode and M-mode can write the currently selected trigger registers. 1 — Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.	RW	0
TYPE	[63:60]	Indicates the trigger type. **Value** / **Meaning** 0 — The selected trigger is invalid. 2 — The selected trigger is an address/data match trigger.	RW	2

Official Release

Note

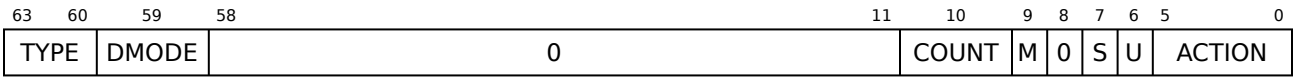
The LOAD/STORE fields take no effect and are cleared if the EXECUTE field is set at the same time.

21.6.10 Instruction Count

Mnemonic Name: icount
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a1 (standard read/write)

This register is accessible as tdata1 when TYPE is 3.

This register exists just for single-stepping support so COUNT is hardwired to 1. After this trigger fires, the mode bits the mode bits (M, S, U) will be cleared instead of causing the COUNT bits to be decremented.



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
ACTION	[5:0]	Setting this field to select what happens when this trigger matches. **Value** / **Meaning** 0 — Raise a breakpoint exception. 1 — Enter Debug Mode. (Only supported when DMODE is 1.)	RW	0
U	[6]	Setting this field to enable this trigger in U-mode.	RW	0
S	[7]	Setting this field to enable this trigger in S-mode.	RW	0
M	[9]	Setting this field to enable this trigger in M-mode.	RW	0
COUNT	[10]	This field is hardwired to 1 for single-stepping support	RO	1
DMODE	[59]	Setting this field to indicate the trigger is used by Debug Mode. **Value** / **Meaning** 0 — Both Debug-mode and M-mode can write the currently selected trigger registers. 1 — Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.	RW	0
TYPE	[63:60]	The selected trigger is an instruction count trigger.	RW	2

Official Release

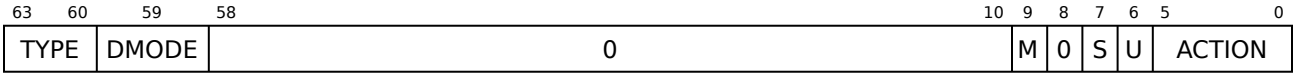
21.6.11 Interrupt Trigger

Mnemonic Name: itrigger
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a1 (standard read/write)

This register is accessible as `tdata1` when `TYPE` is 4.

This trigger may fire on any of the interrupts configurable in `mie`. The interrupts to fire on are configured by setting the same bit in `tdata2` as would be set in `mie` to enable the interrupt.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



Field Name	Bits	Description	Type	Reset
ACTION	[5:0]	Setting this field to select what happens when this trigger matches. **Value** / **Meaning** 0 — Raise a breakpoint exception. 1 — Enter Debug Mode. (Only supported when `DMODE` is 1.)	RW	0
U	[6]	Setting this field to enable this trigger in U-mode.	RW	0
S	[7]	Setting this field to enable this trigger in S-mode.	RW	0
M	[9]	Setting this field to enable this trigger in M-mode.	RW	0
DMODE	[59]	Setting this field to indicate the trigger is used by Debug Mode. **Value** / **Meaning** 0 — Both Debug-mode and M-mode can write the currently selected trigger registers. 1 — Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.	RW	0
TYPE	[63:60]	The selected trigger is an interrupt trigger.	RW	2

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

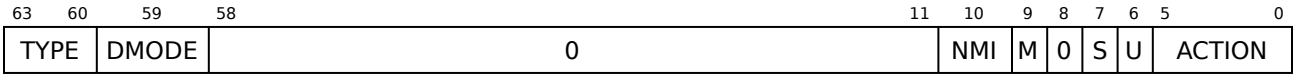
21.6.12 Exception Trigger

Mnemonic Name: etrigger
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a1 (standard read/write)

Official Release

This register is accessible as `tdata1` when `TYPE` is 5.

This trigger may fire on up to XLEN of the Exception Codes defined in `mcause` (with Interrupt=0). Those causes are configured by writing the corresponding bit in `tdata2`. (E.g. to trap on an illegal instruction, the debugger sets bit 2 in `tdata2`.)



Field Name	Bits	Description	Type	Reset
ACTION	[5:0]	Setting this field to select what happens when this trigger matches. **Value** / **Meaning** 0 — Raise a breakpoint exception. 1 — Enter Debug Mode. (Only supported when `DMODE` is 1.)	RW	0
U	[6]	Setting this field to enable this trigger in U-mode.	RW	0
S	[7]	Setting this field to enable this trigger in S-mode.	RW	0
M	[9]	Setting this field to enable this trigger in M-mode.	RW	0
NMI	[10]	Setting this field to enable this trigger in non-maskable interrupts, regardless of the values of s, u, and m.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DMODE	[59]	Setting this field to indicate the trigger is used by Debug Mode. **Value** / **Meaning** 0 — Both Debug-mode and M-mode can write the currently selected trigger registers. 1 — Only Debug Mode can write the currently selected trigger registers. Writes from M-mode is ignored.	RW	0
TYPE	[63:60]	The selected trigger is an exception trigger.	RW	2

Official Release

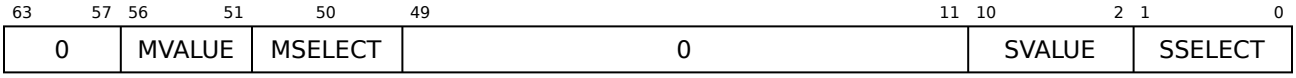
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.6.13 Trigger Extra

Mnemonic Name: textra
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug and Machine
CSR Address: 0x7a3 (standard read/write)

Official Release

This register is accessible as `tdata3` when `TYPE` is 2, 3, 4, or 5 of the currently selected trigger registers selected by the `tselect` register, and it indicates the context matching scheme of the currently selected trigger.



Field Name	Bits	Description	Type	Reset
SSELECT	[1:0]	<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Ignore MVALUE.</td></tr><tr><td>1</td><td>This trigger will only match if the lower bits of scontext equal SVALUE.</td></tr><tr><td>2</td><td>This trigger will only match if satp.ASID equals SVALUE.</td></tr></table>	RW	0
SVALUE	[10:2]	Data used together with SSELECT.	RW	0
MSELECT	[50]	<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Ignore MVALUE.</td></tr><tr><td>1</td><td>This trigger will only match if the lower bits of mcontext equal MVALUE.</td></tr></table>	RW	0
MVALUE	[56:51]	Data used together with MSELECT.	RW	0

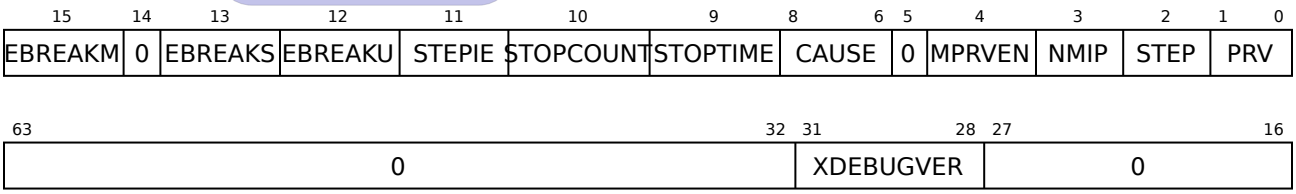
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.7 Debug and Trigger Registers

21.7.1 Debug Control and Status Register

Mnemonic Name: dcsr
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug
CSR Address: 0x7b0 (debug-mode-only)

Official Release



Field Name	Bits	Description	Type	Reset
PRV	[1:0]	The privilege level that the hart was operating in when Debug Mode was entered. The external debugger can modify this value to change the hart's privilege level when exiting Debug Mode.	WARL	3
STEP	[2]	This bit controls whether non-Debug Mode instruction execution is in the single step mode. When set, the hart returns to Debug Mode after a single instruction execution. If the instruction does not complete due to an exception, the hart will immediately enter Debug Mode before executing the trap handler, with appropriate exception registers set.	RW	0

Value	Meaning
0	User/Application
1	Supervisor
2	Reserved
3	Machine

Value	Meaning
0	Single Step Mode is off
1	Single Step Mode is on

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
NMIP	[3]	When this bit is set, there is a Non-Maskable-Interrupt (NMI) pending for the hart. Since an NMI can indicate a hardware error condition, reliable debugging may no longer be possible once this bit becomes set.	RO	0
MPRVEN	[4]	This bit controls whether `mstatus.MPRV` takes effect in Debug Mode. **Value** / **Meaning** 0 — `MPRV` in `mstatus` is ignored in Debug Mode. 1 — `MPRV` in `mstatus` takes effect in Debug Mode.	RW	0
CAUSE	[8:6]	Reason why Debug Mode was entered. When there are multiple reasons to enter Debug Mode, the priority to determine the `CAUSE` value will be: trigger module > `EBREAK` > halt-on-reset > halt request > single step. Halt requests are requests issued by the external debugger. **Value** / **Meaning** 0 — Reserved 1 — EBREAK 2 — Trigger module 3 — Halt request 4 — Single step 5 — Halt-on-reset 6–7 — Reserved	RO	0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
STOPTIME	[9]	This bit controls whether timers are stopped in Debug Mode. The processor only drives its `stoptime` output pin to 1 if it is in Debug Mode and this bit is set. Integration effort is required to make timers in the platform observe this pin to really stop them.	RW	1

Value	Meaning
0	Do not stop timers in Debug Mode
1	Stop timers in Debug Mode

Field Name	Bits	Description	Type	Reset
STOPCOUNT	[10]	This bit controls whether performance counters are stopped in Debug Mode.	RW	1

Value	Meaning
0	Do not stop counters in Debug Mode
1	Stop counters in Debug Mode

Field Name	Bits	Description	Type	Reset
STEPIE	[11]	This bit controls whether interrupts are enabled during single stepping.	RW	0

Value	Meaning
0	Disable interrupts during single stepping
1	Allow interrupts in single stepping

Field Name	Bits	Description	Type	Reset
EBREAKU	[12]	This bit controls the behavior of `EBREAK` instructions in User/Application Mode.	RW	0

Value	Meaning
0	Generate a regular breakpoint exception
1	Enter Debug Mode

Continued on next page...

Official Release

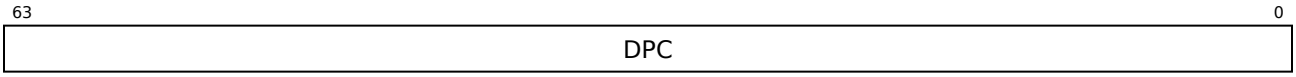
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
EBREAKS	[13]	This bit controls the behavior of EBREAK instructions in Supervisor Mode. 0: Generate a regular breakpoint exception 1: Enter Debug Mode	RW	0
EBREAKM	[15]	This bit controls the behavior of EBREAK instructions in Machine Mode. 0: Generate a regular breakpoint exception 1: Enter Debug Mode	RW	0
XDEBUGVER	[31:28]	Version of the external debugger. 0 indicates that no external debugger exists and 4 indicates that the external debugger conforms to the *RISC-V External Debug Support (TD003) V0.13*.	RO	4

Official Release

21.7.2 Debug Program Counter

Mnemonic Name: dpc
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug
CSR Address: 0x7b1 (debug-mode-only)



When entering Debug Mode, the dpc CSR is updated with the virtual address of the next instruction to be executed. The behavior is described in more detail in Table 156. When leaving Debug Mode, the hart's PC is updated to the value stored in this register. The external debugger may write this register to change where the hart resumes.

Field Name	Bits	Description	Type	Reset
DPC	[63:0]	Debug Program Counter. Bit 0 is hardwired to 0.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 156: Virtual Address in DPC upon Debug Mode Entry

Cause	Virtual Address in DPC
EBREAK	Address of the EBREAK instruction
single step	Address of the instruction that would be executed next if no debugging was going on.
trigger module	Address of the instruction which caused the trigger module to fire.
halt request	Address of the next instruction to be executed at the time that Debug Mode was entered

Official Release

21.7.3 Debug Scratch Register 0

Mnemonic Name: dscratch0
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug
CSR Address: 0x7b2 (debug-mode-only)

A scratch register that is reserved for use by Debug Module.

21.7.4 Debug Scratch Register 1

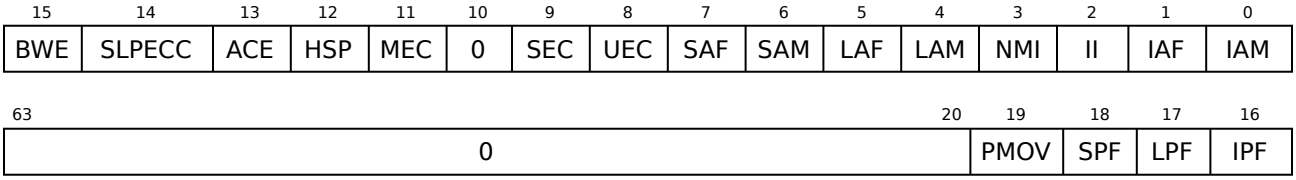
Mnemonic Name: dscratch1
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug
CSR Address: 0x7b3 (debug-mode-only)

A scratch register that is reserved for use by Debug Module.

21.7.5 Exception Redirection Register

Mnemonic Name: dexc2dbg
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug
CSR Address: 0x7e0 (non-standard read/write)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



Official Release

This register redirects selected exceptions to cause the hart to enter Debug Mode instead of performing the standard trap handling.

When an exception is redirected to enter Debug Mode, the `dpc` CSR will be updated with the virtual address of the instruction causing the exception. The `dcsr.CAUSE` field will be updated with a value of 1 (EBREAK). The actual cause of the exception is saved to the `ddcause` CSR. The required updates to `mepc`, `mcause`, `mtval`, `mstatus`, and `mxstatus` CSRs for exceptions will not be affected by the redirection and these CSRs continue to provide information associated with the corresponding exceptions.

Field Name	Bits	Description	Type	Reset
IAM	[0]	Indicates whether Instruction Access Misaligned exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 — Do not redirect; 1 — Redirect	RW	0
IAF	[1]	Indicates whether Instruction Access Fault exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 — Do not redirect; 1 — Redirect	RW	0
II	[2]	Indicates whether Illegal Instruction exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 — Do not redirect; 1 — Redirect	RW	0
NMI	[3]	Indicates whether Non-Maskable Interrupt exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 — Do not redirect; 1 — Redirect	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
LAM	[4]	Indicates whether Load Access Misaligned exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 / Do not redirect 1 / Redirect	RW	0
LAF	[5]	Indicates whether Load Access Fault exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 / Do not redirect 1 / Redirect	RW	0
SAM	[6]	Indicates whether Store Access Misaligned exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 / Do not redirect 1 / Redirect	RW	0
SAF	[7]	Indicates whether Store Access Fault exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 / Do not redirect 1 / Redirect	RW	0
UEC	[8]	Indicates whether U-mode Environment Call exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 / Do not redirect 1 / Redirect	RW	0
SEC	[9]	Indicates whether S-mode Environment Call exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 / Do not redirect 1 / Redirect	RW	0

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MEC	[11]	Indicates whether M-mode Environment Call exceptions are redirected to enter Debug Mode. **Value** / **Meaning** 0 — Do not redirect 1 — Redirect	RW	0
HSP	[12]	Indicates whether Stack Protection exceptions are redirected to enter Debug Mode. This bit is present only when `mmsc_cfg.HSP` is set. **Value** / **Meaning** 0 — Do not redirect 1 — Redirect	RW	0
ACE	[13]	Indicates whether ACE-related exceptions are redirected to enter Debug Mode. This bit is present only when `mmsc_cfg.ACE` is set. **Value** / **Meaning** 0 — Do not redirect 1 — Redirect	RW	0
SLPECC	[14]	Indicates whether local memory slave port ECC Error local interrupts are redirected to enter Debug Mode. **Value** / **Meaning** 0 — Do not redirect 1 — Redirect	RW	0
BWE	[15]	Indicates whether Bus-write Transaction Error local interrupts are redirected to enter Debug Mode. **Value** / **Meaning** 0 — Do not redirect 1 — Redirect	RW	0

Continued on next page...

Official Release

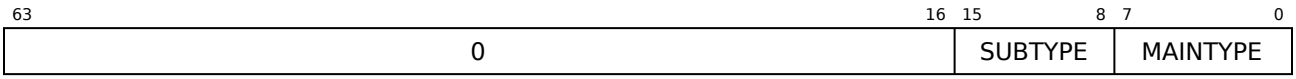
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IPF	[16]	Indicates whether instruction page fault exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 = Do not redirect; 1 = Redirect	RW	0
LPF	[17]	Indicates whether load fault exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 = Do not redirect; 1 = Redirect	RW	0
SPF	[18]	Indicates whether store page fault exceptions are redirected to enter Debug Mode. **Value** / **Meaning**: 0 = Do not redirect; 1 = Redirect	RW	0
PMOV	[19]	Indicates whether performance counter overflow interrupts are redirected to enter Debug Mode. **Value** / **Meaning**: 0 = Do not redirect; 1 = Redirect	RW	0

Official Release

21.7.6 Debug Detailed Cause

Mnemonic Name: ddcause
IM Requirement: DEBUG_SUPPORT
Access Mode: Debug
CSR Address: 0x7e1 (non-standard read/write)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MAINTYPE	[7:0]	Cause for redirection to Debug Mode.	RO	0

Value	Meaning
0	Software Breakpoint (EBREAK)
1	Instruction Access Misaligned (IAM)
2	Instruction Access Fault (IAF)
3	Illegal Instruction (II)
4	Non-Maskable Interrupt (NMI)
5	Load Access Misaligned (LAM)
6	Load Access Fault (LAF)
7	Store Access Misaligned (SAM)
8	Store Access Fault (SAF)
9	U-mode Environment Call (UEC)
10	S-mode Environment Call (SEC)
11	Instruction page fault
12	M-mode Environment Call (MEC)
13	Load page fault
14	Reserved
15	Store/AMO page fault
16	Imprecise ECC error
17	Bus write transaction error
18	Performance Counter overflow
19–31	Reserved
32	Stack overflow exception
33	Stack underflow exception
34	ACE disabled exception
35–39	Reserved
40–47	ACE exception
≥48	Reserved

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SUBTYPE	[15:8]	Subtypes for main type. The table below lists the subtypes for DCSR.CAUSE==1 and DDCAUSE.MAINTYPE==3.	RO	0

Value	Meaning
0	Illegal instruction
1	Privileged instruction
2	Non-existent CSR
3	Privilege CSR access
4	Read-only CSR update

Official Release

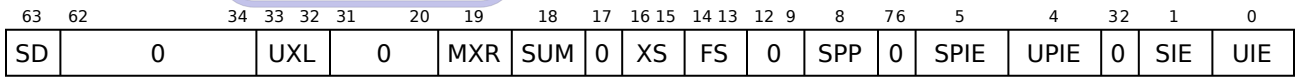
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8 Supervisor Trap Related CSRs

21.8.1 Supervisor Status

Mnemonic Name: sstatus
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x100 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
UIE	[0]	U-mode interrupt enable bit. Value: 0 = Disabled; 1 = Enabled	RW	0
SIE	[1]	S-mode interrupt enable bit. Value: 0 = Disabled; 1 = Enabled	RW	0
UPIE	[4]	`UPIE` holds the value of the `UIE` bit prior to a trap.	RW	0
SPIE	[5]	`SPIE` holds the value of the `SIE` bit prior to a trap.	RW	0
SPP	[8]	`SPP` holds the privilege mode prior to a trap. Encoding is 1 for S-mode and 0 for U-mode.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
FS	[14:13]	FS holds the status of the architectural states of the floating-point unit, including the fcsr CSR and f0 – f31 floating-point data registers. The value of this field is zero and read-only if the processor does not have FPU. This field is primarily managed by software. The processor hardware assists the state managements in two regards: • Attempts to access fcsr or any f register raise an illegal-instruction exception when FS is Off. • Otherwise, FS is updated to the Dirty state by any instruction that updates fcsr or any f register. Changing the setting of this field has no effect on the contents of the floating-point register states. In particular, setting FS to Off does not destroy the states, nor does setting FS to Initial clear the contents. The same copy of FS bits are shared by both mstatus and sstatus. Normally the supervisor mode privileged software would use the FS bits to manage deferred context switches of FPU states. Machine mode software should be more conservative in managing context switches using FS bits.	WLRL	0

Value	Meaning
0	Off
1	Initial
2	Clean
3	Dirty

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
XS	[16:15]	XS holds the status of the architectural states (ACE registers) of ACE instructions. The value of this field is zero if ACE extension is not configured. This field is primarily managed by software. The processor hardware assists the state managements in two regards: • Illegal instruction exceptions are triggered when XS is Off. • XS is updated to the Dirty state with the execution of ACE instructions when XS is not Off. Changing the setting of this field has no effect on the contents of ACE states. In particular, setting XS to Off does not destroy the states, nor does setting XS to Initial clear the contents. The same copy of XS bits are shared by both mstatus and sstatus. Normally the supervisor mode privileged software would use the XS bits to manage deferred context switches of ACE states. Machine mode software should be more conservative in managing context switches using XS bits.	RO	0

Value	Meaning
0	Off
1	Initial
2	Clean
3	Dirty

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SUM	[18]	SUM controls whether a S-mode load/store instruction to a user accessible page is allowed or not when page translation is enabled. It is in effect in two scenarios: (a) M-mode with MPRV=1 and MPP=S, and (b) in S-mode. It has no effect when page-based virtual memory is not in effect. A page is user accessible when the U bit of the corresponding PTE entry is 1.	RW	0
MXR	[19]	MXR controls whether execute-only pages are readable. It has no effect when page-based virtual memory is not in effect.	RW	0
UXL	[33:32]	UXL controls the value of XLEN for U-mode. When U-mode is not available, this field is hardwired to 0.	RO	2/0
SD	[63]	SD summarizes whether either the FS field or XS field is dirty.	RO	0

Value	Meaning
0	Not Allowed
1	Allowed

Value	Meaning
0	Execute-only pages are not readable
1	Execute-only pages are readable

Value	Meaning
1	32
2	64

Official Release

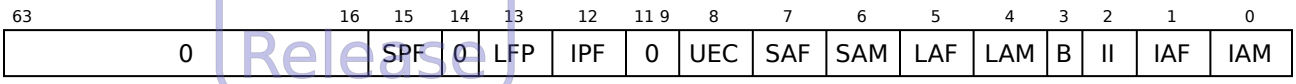
When N extension is not supported, the corresponding bits in sstatus are hardwired to zero.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.2 Supervisor Exception Delegation

Mnemonic Name: sedeleg
IM Requirement: misa[18]==1 and misa[13]==1
Access Mode: Supervisor
CSR Address: 0x102 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
IAM	[0]	IAM indicates whether an Instruction Address Misaligned exception will be delegated to U-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0
IAF	[1]	IAF indicates whether an Instruction Access Fault exception will be delegated to U-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0
II	[2]	II indicates whether an Illegal Instruction exception will be delegated to U-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0
B	[3]	B indicates whether an exception triggered by breakpoint will be delegated to U-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
LAM	[4]	LAM indicates whether a Load Address Misaligned exception will be delegated to U-mode. Value 0: Not delegate Value 1: delegate	RW	0
LAF	[5]	LAF indicates whether a Load Access Fault exception will be delegated to U-mode. Value 0: Not delegate Value 1: delegate	RW	0
SAM	[6]	SAM indicates whether a Store/AMO Address Misaligned exception will be delegated to U-mode. Value 0: Not delegate Value 1: delegate	RW	0
SAF	[7]	SAF indicates whether a Store/AMO Access Fault exception will be delegated to U-mode. Value 0: Not delegate Value 1: delegate	RW	0
UEC	[8]	UEC indicates whether an exception triggered by environment call from U-mode will be delegated to U-mode. Value 0: Not delegate Value 1: delegate	RW	0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IPF	[12]	`IPF` indicates whether an Instruction Page Fault exception will be delegated to U-mode. **Value** / **Meaning** 0 / Not delegate 1 / delegate	RW	0
LPF	[13]	`LPF` indicates whether a Load Page Fault exception will be delegated to U-mode. **Value** / **Meaning** 0 / Not delegate 1 / delegate	RW	0
SPF	[15]	`SPF` indicates whether a Store/AMO Page Fault exception will be delegated to U-mode. **Value** / **Meaning** 0 / Not delegate 1 / delegate	RW	0

Official Release

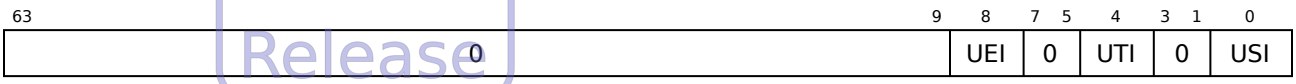
When N extension is not supported, the corresponding bits in `sedeleg` are hardwired to zero.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.3 Supervisor Interrupt Delegation

Mnemonic Name: sideleg
IM Requirement: misa[18]==1 and misa[13]==1
Access Mode: Supervisor
CSR Address: 0x103 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
USI	[0]	`USI` indicates whether an U-mode software interrupt will be delegated to S-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0
UTI	[4]	`UTI` indicates whether an U-mode timer interrupt will be delegated to S-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0
UEI	[8]	`UEI` indicates whether an U-mode external interrupt will be delegated to S-mode. **Value** / **Meaning**: 0 = Not delegate; 1 = delegate	RW	0

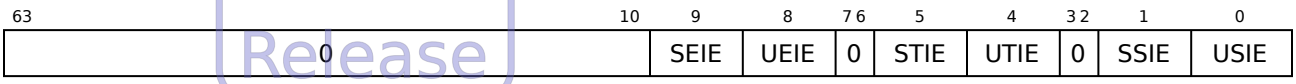
When N extension is not supported, the corresponding bits in `sideleg` are hardwired to zero.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.4 Supervisor Interrupt Enable

Mnemonic Name: sie
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x104 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
USIE	[0]	U-mode software interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
SSIE	[1]	S-mode software interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
UTIE	[4]	U-mode timer interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
STIE	[5]	S-mode timer interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Field Name	Bits	Description	Type	Reset
UEIE	[8]	U-mode external interrupt enable bit.	RW	0

Value	Meaning
0	Disabled
1	Enabled

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SEIE	[9]	S-mode external interrupt enable bit. 0: Disabled 1: Enabled	RW	0

Official Release

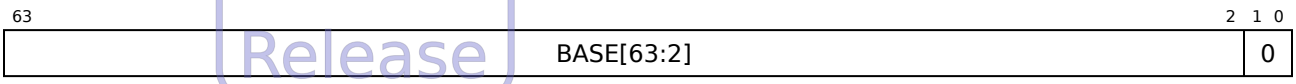
When N extension is not supported, the corresponding bits in `sie` are hardwired to zero.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.5 Supervisor Trap Vector Base Address

Mnemonic Name: stvec
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x105 (standard read/write)

Official Release



This register determines the base address of the trap vector for S-mode trap handling. The least significant 2 bits are hardwired to zeros. When the width of the configured address is less than 64, the upper bits are hardwired to zeros.

When `mmisc_ctl.VEC_PLIC` is 0 (PLIC is not in the vector mode), this register indicates the entry points for the trap handler and it may point to any 4-byte aligned location in the memory space.

On the other hand, when `mmisc_ctl.VEC_PLIC` is 1 (PLIC is in the vector mode), this register will be the base address of a vector table with 4-byte entries storing addresses pointing to interrupt service routines.

- This register should be aligned to $2^{\log_2 N+2}$-byte boundary for PLIC with N interrupt sources. For example, if N is 1023, the minimum alignment requirement is 4096 bytes (4 KiB).
- `stvec[0]` is for exceptions and non-external local interrupts.
- `stvec[`*i*`]` is for external PLIC interrupt source *i* triggered through the `sip.SEIP` pending condition when `mideleg.SEI` == 1.
- `stvec[1024+`*i*`]` is for external PLIC interrupt source *i* triggered through the `sip.UEIP` pending condition when `sideleg.UEI` == 0.

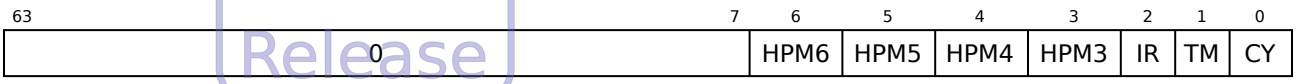
Field Name	Bits	Description	Type	Reset
BASE[63:2]	[63:2]	Base address for interrupt and exception handlers. See description above for alignment requirements when PLIC is in the vector mode.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.6 Supervisor Counter Enable Register

Mnemonic Name: scounteren
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x106 (standard read/write)

Official Release



The supervisor counter-enable register controls the availability of the hardware performance monitoring counters to U-mode.

If S-mode is not permitted to access a counter register or when the corresponding bit in the `scounteren` register is zero, attempts to read the corresponding register while executing in U-mode will cause an illegal instruction exception. If S-mode is permitted to access a counter register and the corresponding bit is set, access to the counter register is permitted in U-mode.

In summary, a counter can be accessed in U-mode only when (`mcounteren.counter==1`) and (`scounteren.counter==1`).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.7 Supervisor Scratch Register

Mnemonic Name: sscratch
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x140 (standard read/write)

Official Release

A scratch register for temporary data storage, which is typically used by the S-mode trap handler.

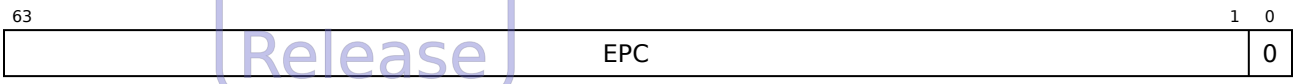
Field Name	Bits	Description	Type	Reset
SSCRATCH	[63:0]	Scratch register storage.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.8 Supervisor Exception Program Counter

Mnemonic Name: sepc
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x141 (standard read/write)

Official Release



This register is written with the virtual address of the instruction that raises a trap and the trap is taken to S-mode.

When **Page-Based Virtual Memory** is configured to "sv39", bits 63–40 of this register is hardwired to bit 39. AX45MP ignores bits 63–40 of written values.

When **Page-Based Virtual Memory** is configured to "sv48", bits 63–49 of this register is hardwired to bit 48. AX45MP ignores bits 63–49 of written values.

When **Page-Based Virtual Memory** is configured to "bare", bits 63–BIU_ADDR_WIDTH of this register is hardwired to 0. AX45MP ignores bits 63–BIU_ADDR_WIDTH of written values.

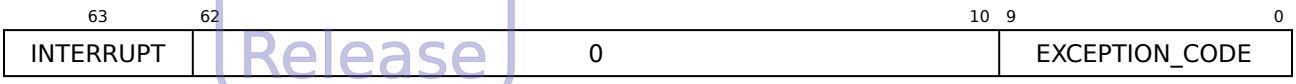
Field Name	Bits	Description	Type	Reset
EPC	[63:1]	Exception program counter.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.9 Supervisor Cause Register

Mnemonic Name: scause
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x142 (standard read/write)

Official Release



This register indicates the cause of traps when they are taken to S-mode.

Field Name	Bits	Description	Type	Reset
EXCEPTION_CODE	[9:0]	Exception Code.	RW	0
INTERRUPT	[63]	Interrupt.	RW	0

Each local interrupt can be configured with a local interrupt number. For S-mode local interrupts, cause numbers will be (local interrupt number + 256). Cause numbers below show the default cause numbers of local interrupts.

Table 157: AX45MP scause Value After Trap

Interrupt	Exception Code	Description
1	0	User software interrupt
1	1	Supervisor software interrupt
1	4	User timer interrupt
1	5	Supervisor timer interrupt
1	8	User external interrupt
1	9	Supervisor external interrupt
1	256+16	Slave port ECC error interrupt (S-mode)
1	256+17	Bus write transaction error interrupt (S-mode)
1	256+18	Performance monitor overflow interrupt(S-mode)
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 157: (continued)

Interrupt	Exception Code	Description
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9	Environment call from S-mode
0	11:10	Reserved
0	12	Instruction page fault
0	13	Load page fault
0	14	Reserved
0	15	Store/AMO page fault
0	32	Stack overflow exception
0	33	Stack underflow exception
0	40-47	Andes Custom Extension exception (see *Andes Custom Extension Specification* for more details)

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.10 Supervisor Trap Value

Mnemonic Name: stval
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x143 (standard read/write)

Official Release

This register is updated when a trap is taken to S-mode. The updated value is dependent on the cause of traps:

- For Hardware Breakpoint exceptions, Address Misaligned exceptions, Access Fault exceptions, or Page Fault exceptions, it is the effective faulting addresses.
- For illegal instruction exceptions, the updated value is the faulting instruction. If the length of the instruction is less than XLEN bits long, the upper bits of `stval` are cleared to zero.
- For other exceptions, `stval` is set to zero.

For instruction-fetch access faults, this register will be updated with the address pointing to the portion of the instruction that caused the fault, while the `sepc` register will be updated with the address pointing to the beginning of the instruction.

When the width of the configured address is less than 64, the upper bits are hardwired to zeros.

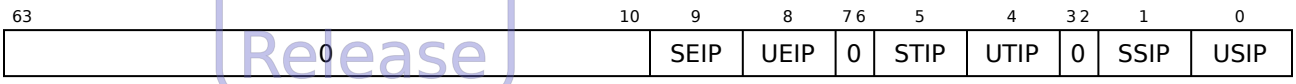
Field Name	Bits	Description	Type	Reset
STVAL	[63:0]	Exception-specific information for software trap handling.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.11 Supervisor Interrupt Pending

Mnemonic Name: sip
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x144 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
USIP	[0]	U-mode software interrupt pending bit.	RW	0

Value	Meaning
0	Not pending
1	Pending

Field Name	Bits	Description	Type	Reset
SSIP	[1]	S-mode software interrupt pending bit.	RW	0

Value	Meaning
0	Not pending
1	Pending

Field Name	Bits	Description	Type	Reset
UTIP	[4]	U-mode timer interrupt pending bit.	RO	0

Value	Meaning
0	Not pending
1	Pending

Field Name	Bits	Description	Type	Reset
STIP	[5]	S-mode timer interrupt pending bit.	RO	0

Value	Meaning
0	Not pending
1	Pending

Field Name	Bits	Description	Type	Reset
UEIP	[8]	U-mode external interrupt pending bit.	RW	0

Value	Meaning
0	Not pending
1	Pending

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SEIP	[9]	S-mode external interrupt pending bit. 0: Not pending 1: Pending	RO	0

Official Release

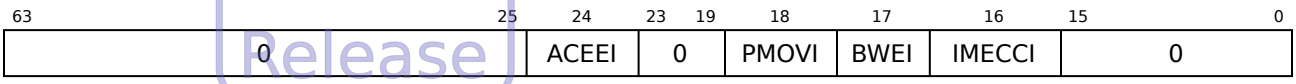
When N extension is not supported, the corresponding bits in `sip` are hardwired to zero.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.12 Supervisor Local Interrupt Enable

Mnemonic Name: slie
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x9C4 (non-standard read/write)

Official Release



If a supervisor local interrupt is enabled, the supervisor local interrupt can be taken in the mode depending on the `mslideleg` CSR. When `mslideleg` for a local interrupt is set, the supervisor local interrupt will be served in S-mode. Otherwise, it will be served in M-mode.

The privileged mode of `IMECCI`, `BWEI`, and `ACEEI` are determined by the current privileged mode. For M/S/U configuration, if the current privileged mode is User or Supervisor, the local interrupt generated is a supervisor local interrupt. For M/U configuration, if the current privileged mode is User, the local interrupt generated is a machine local interrupt. The privileged mode of the above `PMOVI` interrupt is determined by the counter state of `mcountermask_m`.

Each local interrupt can be configured with a local interrupt interrupt number. Bit location of interrupts are the same as their interrupt numbers. Register fields below show the default bit location.

Field Name	Bits	Description	Type	Reset
IMECCI	[16]	Enable S-mode imprecise ECC error local interrupt. The processor may receive imprecise ECC errors on slave port accesses or cache writebacks. **Value** / **Meaning** 0 — Local interrupt is not enabled. 1 — Local interrupt is enabled.	RW	0
BWEI	[17]	Enable S-mode bus read/write transaction error local interrupt. The processor may receive bus errors on load/store instructions or cache writebacks. **Value** / **Meaning** 0 — Local interrupt is not enabled. 1 — Local interrupt is enabled.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
PMOVI	[18]	Enable S-mode performance monitor overflow local interrupt. **Value** / **Meaning** 0 — Local interrupt is not enabled. 1 — Local interrupt is enabled.	RW	0
ACEEI	[24]	Enable S-mode ACE error local interrupt. **Value** / **Meaning** 0 — Local interrupt is not enabled. 1 — Local interrupt is enabled.	RW	0

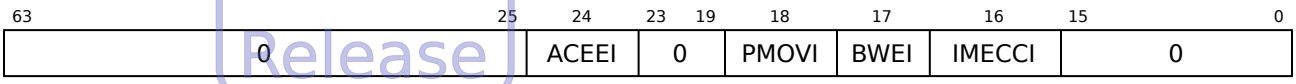
Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.13 Supervisor Local Interrupt Pending

Mnemonic Name: slip
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x9C5 (non-standard read/write)

Official Release



This register indicates whether a supervisor local interrupt is pending or not. If an enabled supervisor local interrupt is pending, the supervisor local interrupt will be taken in the mode depending on the `mslideleg` CSR. When `mslideleg` for a local interrupt is set, the supervisor local interrupt will be served in S-mode. Otherwise, it will be served in M-mode.

The privileged mode of `IMECCI`, `BWEI`, and `ACEEI` are determined by the current privileged mode. For M/S/U configuration, if the current privileged mode is User or Supervisor, the local interrupt generated is a supervisor local interrupt. For M/U configuration, if the current privileged mode is User, the local interrupt generated is a machine local interrupt. The privileged mode of the above `PMOVI` interrupt is determined by the counter state of `mcountermask_m`.

Each local interrupt can be configured with a local interrupt interrupt number. Bit location of interrupts are the same as their interrupt numbers. Register fields below show the default bit location.

Field Name	Bits	Description	Type	Reset
IMECCI	[16]	Pending status of S-mode imprecise ECC error local interrupt. The processor may receive imprecise ECC errors on slave port accesses or cache writebacks. **Value** / **Meaning** 0 — Local interrupt is not pending. 1 — Local interrupt is pending.	RW	0
BWEI	[17]	Pending status of S-mode bus read/write transaction error local interrupt. The processor may receive bus errors on load/store instructions or cache writebacks. **Value** / **Meaning** 0 — Local interrupt is not pending. 1 — Local interrupt is pending.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
PMOVI	[18]	Pending status of S-mode performance monitor overflow local interrupt. **Value** / **Meaning** 0 — Local interrupt is not pending. 1 — Local interrupt is pending.	RW	0
ACEEI	[24]	Pending status of S-mode ACE error local interrupt. **Value** / **Meaning** 0 — Local interrupt is not pending. 1 — Local interrupt is pending.	RW	0

Official Release

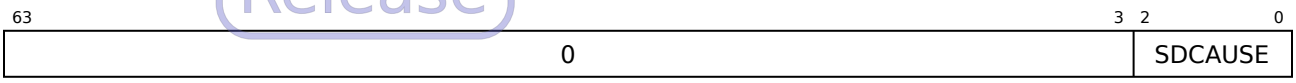
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.14 Supervisor Detailed Trap Cause

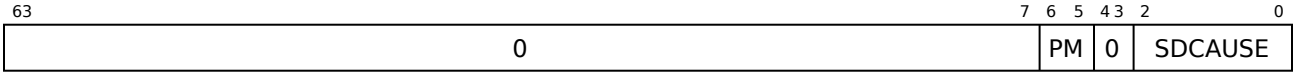
Mnemonic Name: sdcause
IM Requirement: Required if supervisor mode is implemented
Access Mode: Supervisor
CSR Address: 0x9c9 (non-standard read/write)

Official Release

For precise exceptions:



For imprecise exceptions (local interrupts):



When multiple events cause traps to be taken with the same scause value, this register helps to further disambiguate them. Some events could cause either precise exceptions or imprecise exceptions (local interrupts) depending on when they are detected, so they can appear in multiple tables below.

Imprecise exceptions are triggered as local interrupts, so the tables below for `scause` == Local Interrupt *n* summarizes imprecise exceptions delivered as local interrupt *n*. The `scause` == Local Interrupt *n* notation stands for (`INTERRUPT`, `EXCEPTION_CODE`) fields of `scause` is (1, *n*).

Field Name	Bits	Description	Type	Reset
SDCAUSE	[2:0]	This register further disambiguates causes of traps recorded in the `scause` register. See the list below for details.	RW	0
PM	[6:5]	When `scause` is imprecise exception (in the form of an interrupt), the PM field records the privileged mode of the instruction that caused the imprecise exception. The PM field encoding is defined as follows:	RW	0

Value	Meaning
0	User mode
1	Supervisor mode
2	Reserved
3	Machine mode

The value of `SDCAUSE` for precise exception:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- When `scause` == 1 (Instruction access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP instruction access violation
3	Bus error
4	PMA empty hole access

Official Release

- When `scause` == 2 (Illegal instruction)

Value	Meaning
0	Please parse the `stval` CSR
1	FP disabled exception
2	ACE disabled exception

- When `scause` == 5 (Load access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP load access violation
3	Bus error
4	Misaligned address
5	PMA empty hole access
6	PMA attribute inconsistency
7	PMA NAMO exception

- When `scause` == 7 (Store access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP store access violation
3	Bus error
4	Misaligned address

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Value	Meaning
5	PMA empty hole access
6	PMA attribute inconsistency
7	PMA NAMO exception

Official Release

The value of SDCAUSE for imprecise exception:

- When scause == *Local Interrupt 272 (16 + 256)* (ECC error local interrupt)

Value	Meaning
0	Reserved
1	LM slave port ECC/Parity error
2	Imprecise store ECC/Parity error
3	Imprecise load ECC/Parity error

- When scause == *Local Interrupt 273 (17 + 256)* (Bus read/write transaction error local interrupt)

Value	Meaning
0	Reserved
1	Bus read error
2	Bus write error
3	PMP error caused by load instructions
4	PMP error caused by store instructions
5	PMA error caused by load instructions"
6	PMA error caused by store instructions

- For PMOVI, SDCAUSE will be written 0. For other exceptions and interrupts, this register will not be updated.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.8.14.1 Detailed Exception Priority

Within Instruction/Load/Store access fault exceptions, the priority of a PMP exception is higher than the priority of a PMA exception, when both types of exceptions happen on the same instruction.

Official Release

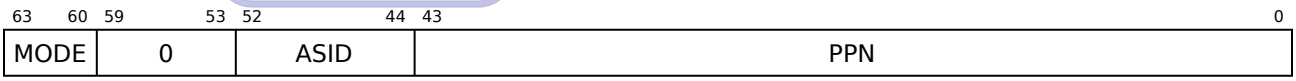
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.9 Supervisor Translation Related CSRs

21.9.1 Supervisor Address Translation and Protection

Mnemonic Name: satp
IM Requirement: misa[18]==1
Access Mode: Supervisor
CSR Address: 0x180 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
PPN	[43:0]	PPN holds the physical page number of the root page table.	RW	0
ASID	[52:44]	ASID holds the address space identifier.	RW	0
MODE	[63:60]	MODE holds the page translation mode. When MODE is Bare, virtual addresses are equal to physical addresses in S-mode. When MMU is not supported in the product, this CSR will be hardwired to 0.	RW	0

Value	Name	Meaning
0	Bare	No page translation
8	Sv39	Page-based 39-bit virtual addressing
9	Sv48	Page-based 48-bit virtual addressing

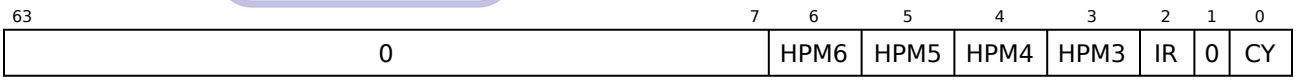
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10 Supervisor Counter Related CSRs

21.10.1 Supervisor Counter Mask for Machine Mode

Mnemonic Name: scountermask_m
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[18] == 1
Access Mode: Supervisor
CSR Address: 0x9D1 (non-standard read/write)

Official Release



The supervisor counter mask for M-mode register is used to disable M-mode counting for each counter.

This register is an alias of the `mcountermask_m` CSR, and its default value is 0. Each bit of this register is read-only if the corresponding bit in the `mcounterwen` CSR is 0. Writing a read-only bit of this register with a CSR write instruction will not generate any exception. This register is not controlled by the `mcounteren` register and can always be read.

Each bit of this register is writable if the corresponding bit in the `mcounterwen` CSR is 1.

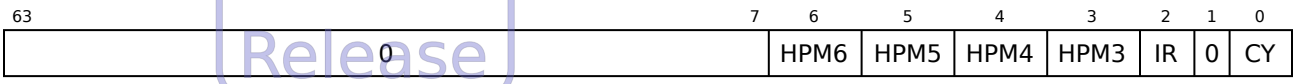
When the `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `scountermask_m` register is set, the specific counter will not be incremented in M-mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10.2 Supervisor Counter Mask for Supervisor Mode

Mnemonic Name: scountermask_s
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[18] == 1
Access Mode: Supervisor
CSR Address: 0x9D2 (non-standard read/write)

Official Release



The supervisor counter mask for S-mode register controls the performance counter behavior in S-mode. The default value of this register is 0.

This register is an alias of the `mcountermask_s` CSR, and its default value is 0. Each bit of this register is read-only if the corresponding bit in the `mcounterwen` CSR is 0. Writing a read-only bit of this register with a CSR write instruction will not generate any exception. This register is not controlled by the `mcounteren` register and can always be read.

Each bit of this register is writable if the corresponding bit in the `mcounterwen` CSR is 1.

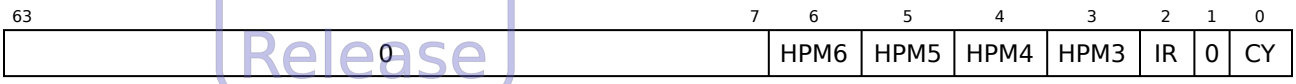
When the `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `scountermask_s` register is set, the specific counter will not be incremented in S-mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10.3 Supervisor Counter Mask for User Mode

Mnemonic Name: scountermask_u
IM Requirement: mmsc_cfg.PMNDS == 1 and misa[18] == 1
Access Mode: Supervisor
CSR Address: 0x9D3 (non-standard read/write)

Official Release



The supervisor counter mask for U-mode register controls the performance counter behavior in U-mode. The default value of this register is 0.

This register is an alias of the `mcountermask_u` CSR, and its default value is 0. Each bit of this register is read-only if the corresponding bit in the `mcounterwen` CSR is 0. Writing a read-only bit of this register with a CSR write instruction will not generate any exception. This register is not controlled by the `mcounteren` register and can always be read.

Each bit of this register is writable if the corresponding bit in the `mcounterwen` CSR is 1.

When the `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in the `scountermask_u` register is set, the specific counter will not be incremented in S-mode.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10.4 Supervisor Counter Interrupt Enable

Mnemonic Name: scounterinten
IM Requirement: mmsc_cfg.PMNDS==1 & misa[18]==1
Access Mode: Supervisor
CSR Address: 0x9CF (non-standard read/write)

Official Release

63 ... 7	6	5	4	3	2	1	0
0	HPM6	HPM5	HPM4	HPM3	IR	0	CY

The supervisor counter interrupt enable register controls whether a counter overflow interrupt is generated or not. This register is an alias of the `mcounterinten` CSR, and its default value is 0. Each bit of this register is read-only if the corresponding bit in the `mcounterwen` CSR is 0. Writing a read-only bit of this register with a CSR write instruction will not generate any exception. This register is not controlled by the `mcounteren` register and can always be read.

Each bit of this register is writable if the corresponding bit in the `mcounterwen` CSR is 1.

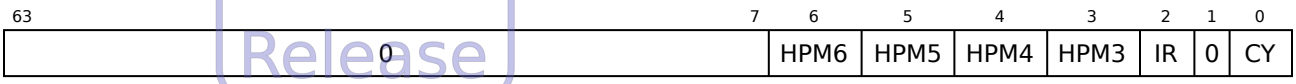
When the `CY`, `IR`, `HPM3`, `HPM4`, `HPM5`, or `HPM6` in this register is 0, no overflow interrupt is generated for the corresponding counter. When one of these bits is set, an interrupt will be generated when the corresponding counter overflows (the counter value wraps around back to 0).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10.5 Supervisor Counter Overflow Status

Mnemonic Name: scounterovf
IM Requirement: mmsc_cfg.PMNDS==1 & misa[18]==1
Access Mode: Supervisor
CSR Address: 0x9D4 (non-standard read/write)

Official Release



The supervisor counter overflow status register records the overflow status of each counter. When a bit is set, it indicates that an overflow has happened to the corresponding counter. This register is an alias of the `mcounterovf` CSR. Each bit of this register is read-only if the corresponding bit in the `mcounterwen` CSR is 0. Writing a read-only bit of this register with a CSR write instruction will not generate any exception. This register is not controlled by the `mcounteren` register and can always be read. If the corresponding bit in the `mcounterwen` CSR is 1, writing 0 will deassert the bit and writing 1 will have no effect.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10.6 Supervisor Counter-Inhibit

Mnemonic Name: scountinhibit
IM Requirement: mmsc_cfg.PMNDS==1 & misa[18]==1
Access Mode: Supervisor
CSR Address: 0x9E0 (non-standard read/write)

Official Release

63 ... 7	6	5	4	3	2	1	0
0	HPM6	HPM5	HPM4	HPM3	IR	TM	CY

This register is an alias of the `mcountinhibit` CSR. Each bit of this register is read-only if the corresponding bit in the `mcounterwen` CSR is 0. Writing a read-only bit of this register with a CSR write instruction will not generate any exception. This register is not controlled by the `mcounteren` register and can always be read.

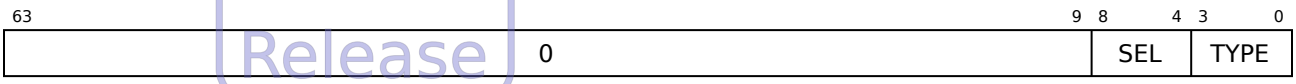
Each bit of this register is writable if the corresponding bit in the `mcounterwen` CSR is 1.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.10.7 Supervisor Performance Monitoring Event Selector

Mnemonic Name: shpmevent3–shpmevent6
IM Requirement: mmsc_cfg.PMNDS==1 & misa[18]==1
Access Mode: Supervisor
CSR Address: 0x9E3 to 0x9E6 (non-standard read/write)

Official Release



The monitoring event is defined in Section 21.4.5.

The read behavior of this register is controlled by the `mcounteren` register. The write behavior of this register is controlled by the `mcounterwen` register.

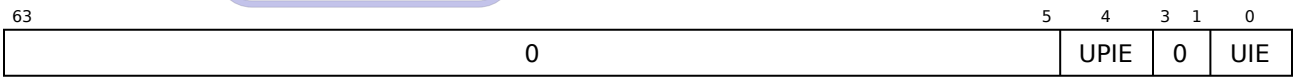
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11 User Trap Related CSRs

21.11.1 User Status

Mnemonic Name: ustatus
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x000 (standard read/write)

Official Release



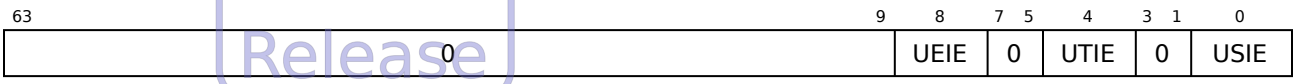
Field Name	Bits	Description	Type	Reset
UIE	[0]	U-mode interrupt enable bit. **Value** / **Meaning** 0 — Disabled 1 — Enabled	RW	0
UPIE	[4]	`UPIE` holds the value of the `UIE` bit prior to a trap.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.2 User Interrupt Enable

Mnemonic Name: uie
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x004 (standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
USIE	[0]	U-mode software interrupt enable bit.	RW	0
UTIE	[4]	U-mode timer interrupt enable bit.	RW	0
UEIE	[8]	U-mode external interrupt enable bit.	RW	0

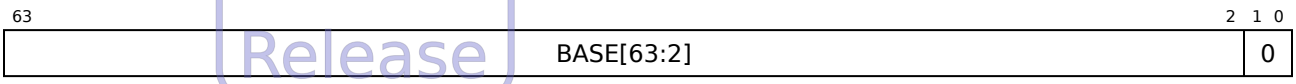
Value	Meaning
0	Disabled
1	Enabled

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.3 User Trap Vector Base Address

Mnemonic Name: utvec
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x005 (standard read/write)

Official Release



This register determines the base address of the trap vector for U-mode trap handling. The least significant 2 bits are hardwired to zeros. When the width of the configured address is less than 64, the upper bits are hardwired to zeros.

When `mmisc_ctl.VEC_PLIC` is 0 (PLIC is not in the vector mode), this register indicates the entry points for the trap handler and it may point to any 4-byte aligned location in the memory space.

On the other hand, when `mmisc_ctl.VEC_PLIC` is 1 (PLIC is in the vector mode), this register will be the base address of a vector table with 4-byte entries storing addresses pointing to interrupt service routines. And this register should be aligned to $2^{\log_2 N+2}$-byte boundary for PLIC with N interrupt sources. For example, if N is 1023, the minimum alignment requirement is 4096 bytes (4 KiB).

Field Name	Bits	Description	Type	Reset
BASE[63:2]	[63:2]	Base address for interrupt and exception handlers. See description above for alignment requirements when PLIC is in the vector mode.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.4 User Scratch Register

Mnemonic Name: uscratch
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x040 (standard read/write)

Official Release

A scratch register for temporary data storage, which is typically used by the U-mode trap handler.

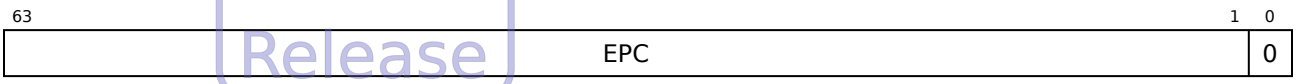
Field Name	Bits	Description	Type	Reset
USCRATCH	[63:0]	Scratch register storage.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.5 User Exception Program Counter

Mnemonic Name: uepc
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x041 (standard read/write)

Official Release



This register is written with the virtual address of the instruction that raises a trap and the trap is taken to U-mode.

When **Page-Based Virtual Memory** is configured to "sv39", bits 63–40 of this register is hardwired to bit 39. AX45MP ignores bits 63–40 of written values.

When **Page-Based Virtual Memory** is configured to "sv48", bits 63–49 of this register is hardwired to bit 48. AX45MP ignores bits 63–49 of written values.

When **Page-Based Virtual Memory** is configured to "bare", bits 63–BIU_ADDR_WIDTH of this register is hardwired to 0. AX45MP ignores bits 63–BIU_ADDR_WIDTH of written values.

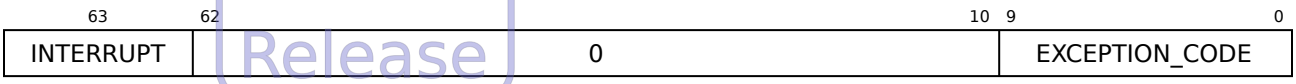
Field Name	Bits	Description	Type	Reset
EPC	[63:1]	Exception program counter.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.6 User Cause Register

Mnemonic Name: ucause
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x042 (standard read/write)

Official Release



This register indicates the cause of traps when they are taken to U-mode.

Field Name	Bits	Description	Type	Reset
EXCEPTION_CODE	[9:0]	Exception Code.	RW	0
INTERRUPT	[63]	Interrupt.	RW	0

Table 158: AX45MP ucause Value After Trap

Interrupt	Exception Code	Description
1	0	User software interrupt
1	4	User timer interrupt
1	8	User external interrupt
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9-11	Reserved
0	12	Instruction page fault
0	13	Load page fault
0	14	Reserved
0	15	Store/AMO page fault
0	32	Stack overflow exception

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 158: (continued)

Interrupt	Exception Code	Description
0	33	Stack underflow exception
0	40-47	Andes Custom Extension exception (see *Andes Custom Extension Specification* for more details)

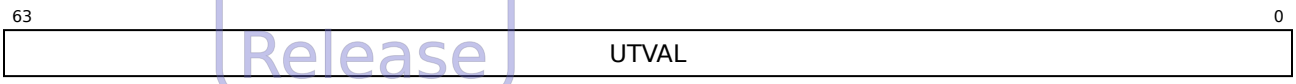
Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.7 User Trap Value

Mnemonic Name: utval
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x043 (standard read/write)

Official Release



This register is updated when a trap is taken to U-mode. The updated value is dependent on the cause of traps:

- For hardware breakpoint exceptions, address-misaligned exceptions, access-fault exceptions, or page-fault exceptions , it is the effective faulting addresses.
- For illegal instruction exceptions, the updated value is the faulting instruction. If the length of the instruction is less than XLEN bits long, the upper bits of `utval` are cleared to zero.
- For other exceptions, `utval` is set to zero.

For instruction-fetch access faults, this register will be updated with the address pointing to the portion of the instruction that caused the fault, while the `sepc` register will be updated with the address pointing to the beginning of the instruction.

When the width of the configured address is less than 64, the upper bits are hardwired to zeros.

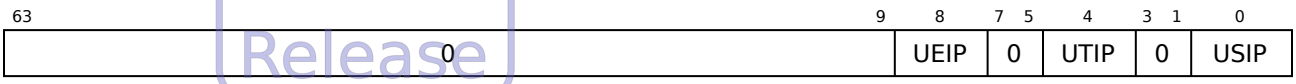
Field Name	Bits	Description	Type	Reset
UTVAL	[63:0]	Exception-specific information for software trap handling.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.8 User Interrupt Pending

Mnemonic Name: uip
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x044 (standard read/write)

Official Release



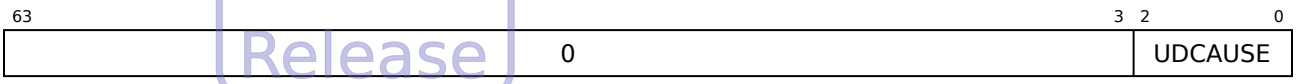
Field Name	Bits	Description	Type	Reset
USIP	[0]	U-mode software interrupt pending bit.	RW	0
		Value / **Meaning**: 0 = Not pending; 1 = Pending		
UTIP	[4]	U-mode timer interrupt pending bit.	RO	0
		Value / **Meaning**: 0 = Not pending; 1 = Pending		
UEIP	[8]	U-mode external interrupt pending bit.	RO	0
		Value / **Meaning**: 0 = Not pending; 1 = Pending		

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.11.9 User Detailed Trap Cause

Mnemonic Name: udcause
IM Requirement: misa[13]==1
Access Mode: User
CSR Address: 0x809 (non-standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
UDCAUSE	[2:0]	This register further disambiguates causes of traps recorded in the `ucause` register. See the list below for details.	RW	0

The value of `UDCAUSE` for precise exception:

- When `ucause` == 1 (Instruction access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP instruction access violation
3	Bus error
4	PMA empty hole access

- When `ucause` == 2 (Illegal instruction)

Value	Meaning
0	Please parse the `utval` CSR
1	FP disabled exception
2	ACE disabled exception

- When `ucause` == 5 (Load access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Value	Meaning
2	PMP load access violation
3	Bus error
4	Misaligned address
5	PMA empty hole access
6	PMA attribute inconsistency
7	PMA NAMO exception

Official Release

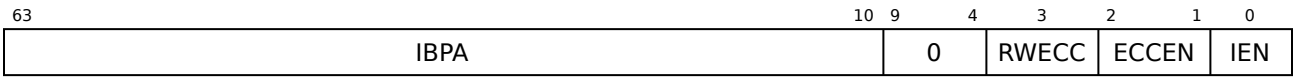
- When `ucause` == 7 (Store access fault)

Value	Meaning
0	Reserved
1	ECC/Parity error
2	PMP store access violation
3	Bus error
4	Misaligned address
5	PMA empty hole access
6	PMA attribute inconsistency
7	PMA NAMO exception

21.12 Memory and Miscellaneous Registers

21.12.1 Instruction Local Memory Base Register

Mnemonic Name: milmb
IM Requirement: ILM_SIZE_KB > 0
Access Mode: Machine
CSR Address: 0x7c0 (non-standard read/write)



This register controls instruction local memory.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

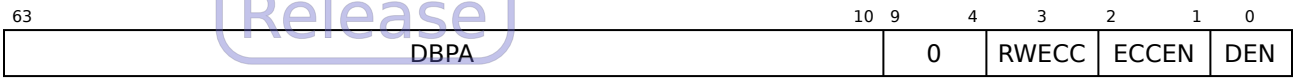
<table>
<tr><th>Field Name</th><th>Bits</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td>IEN</td><td>[0]</td><td>ILM enable control:<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>ILM is disabled</td></tr><tr><td>1</td><td>ILM is enabled</td></tr></table>The reset value is controlled by <code>ilm_boot</code> input signal of the AX45MP processor.</td><td>RW</td><td>IM</td></tr>
<tr><td>ECCEN</td><td>[2:1]</td><td>Parity/ECC enable control:<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Disable parity/ECC</td></tr><tr><td>1</td><td>Reserved</td></tr><tr><td>2</td><td>Generate exceptions only on uncorrectable parity/ECC errors</td></tr><tr><td>3</td><td>Generate exceptions on any type of parity/ECC errors</td></tr></table></td><td>RW</td><td>0</td></tr>
<tr><td>RWECC</td><td>[3]</td><td>Controls diagnostic accesses of ECC codes of the ILM RAMs. When set, load/store to ILM reads/writes ECC codes to the <code>mecc_code</code> register. This bit can be set for injecting ECC errors to test the ECC handler.<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Disable diagnostic accesses of ECC codes</td></tr><tr><td>1</td><td>Enable diagnostic accesses of ECC codes</td></tr></table></td><td>RW</td><td>0</td></tr>
<tr><td>IBPA</td><td>[63:10]</td><td>The base physical address of ILM. It has to be an integer multiple of the ILM size.</td><td>RO</td><td>ILM_ BASE[63:10]</td></tr>
</table>

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.2 Data Local Memory Base Register

Mnemonic Name: mdlmb
IM Requirement: DLM_SIZE_KB > 0
Access Mode: Machine
CSR Address: 0x7c1 (non-standard read/write)

Official Release



This register controls data local memory.

Field Name	Bits	Description	Type	Reset
DEN	[0]	DLM enable control: **Value** / **Meaning** 0 — DLM is disabled 1 — DLM is enabled The reset value is controlled by `dlm_boot` input signal of the AX45MP processor.	RW	IM
ECCEN	[2:1]	Parity/ECC enable control: **Value** / **Meaning** 0 — Disable parity/ECC 1 — Reserved 2 — Generate exceptions only on uncorrectable parity/ECC errors 3 — Generate exceptions on any type of parity/ECC errors	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
RWECC	[3]	Controls diagnostic accesses of ECC codes of the DLM RAMs. When set, load/store to DLM reads/writes ECC codes to the `mecc_code` register. This bit can be set for injecting ECC errors to test the ECC handler. **Value** / **Meaning** 0 — Disable diagnostic accesses of ECC codes 1 — Enable diagnostic accesses of ECC codes	RW	0
DBPA	[63:10]	The base physical address of DLM. It has to be an integer multiple of the DLM size.	RO	DLM_BASE[63:10]

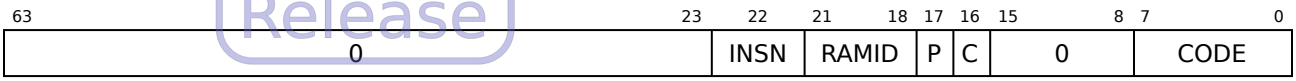
Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.3 ECC Code Register

Mnemonic Name: mecc_code
IM Requirement: mmsc_cfg.ECC == 1
Access Mode: Machine
CSR Address: 0x7c2 (non-standard read/write)

Official Release



This register is used for accessing ECC array.

Field Name	Bits	Description	Type	Reset
CODE	[7:0]	This field records the ECC value on ECC error exceptions. This field is also used to read/write the ECC codes when diagnostic access of ECC codes are enabled (`milmb.RWECC`, `mdlmb.RWECC`, `mcache_ctl.IC_RWECC`, `mcache_ctl.DC_RWECC`, or `mcache_ctl.TLB_RWECC` is 1).	RW	0
C	[16]	Correctable error. This bit is updated on parity/ECC error exceptions. **Value** / **Meaning** 0 — Uncorrectable error 1 — Correctable error	RO	0
P	[17]	Precise error. This bit is updated on parity/ECC error exceptions. **Value** / **Meaning** 0 — Imprecise error 1 — Precise error	RO	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
RAMID	[21:18]	The ID of RAM that caused parity/ECC errors. This bit is updated on parity/ECC error exceptions.	RO	0

Value	Meaning
0–1	Reserved
2	Tag RAM of I-Cache
3	Data RAM of I-Cache
4	Tag RAM of D-Cache
5	Data RAM of D-Cache
6	Tag RAM of TLB
7	Data RAM of TLB
8	ILM
9	DLM
10–15	Reserved

Field Name	Bits	Description	Type	Reset
INSN	[22]	Indicates if the parity/ECC error is caused by instruction fetch or data access.	RO	0

Value	Meaning
0	Data access
1	Instruction fetch

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.4 NMI Vector Base Address Register

Mnemonic Name: mnvec
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x7c3 (non-standard read/write)

Official Release

This register indicates the entry point when an NMI occurs.

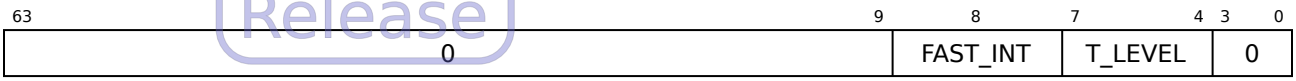
Field Name	Bits	Description	Type	Reset
MNVEC	[63:0]	Base address of the NMI handler. Its value is the zero-extended value of the `coreN_reset_vector[VALEN-1:0]` input signal to this core (Core N).	RO	Pin Configured

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.5 Performance Throttling Control Register

Mnemonic Name: mpft_ctl
IM Requirement: POWERBRAKE_SUPPORT = "yes"
Access Mode: Machine
CSR Address: 0x7c5 (non-standard read/write)

Official Release



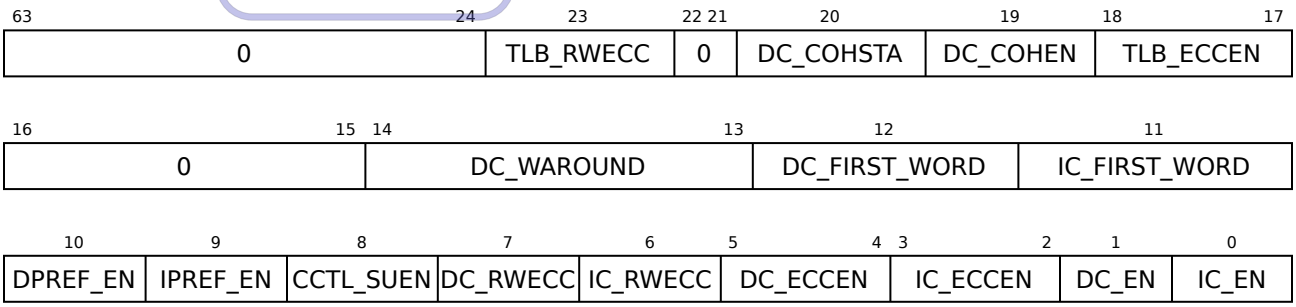
Field Name	Bits	Description	Type	Reset
T_LEVEL	[7:4]	Throttling Level. The processor has the highest performance at throttling level 0 and the lowest performance at throttling level 15. **Value** / **Meaning** 0 — Level 0 (the highest performance) 1-14 — Level 1-14 15 — Level 15 (the lowest performance)	RW	0
FAST_INT	[8]	Fast interrupt response. If this field is set, `mxstatus.PFT_EN` will be automatically cleared when the processor enters an interrupt handler.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.6 Cache Control Register

Mnemonic Name: mcache_ctl
IM Requirement: Cache optional (micm_cfg.ISZ != 0, mdcm_cfg.DSZ != 0, or mmsc_cfg.TLB_ECC != 0)
Access Mode: Machine
CSR Address: 0x7ca (non-standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
IC_EN	[0]	Controls if the instruction cache is enabled or not. **Value** / **Meaning** 0 — I-Cache is disabled 1 — I-Cache is enabled	RW	0
DC_EN	[1]	Controls if the data cache is enabled or not. **Value** / **Meaning** 0 — D-Cache is disabled 1 — D-Cache is enabled	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
IC_ECCEN	[3:2]	Parity/ECC error checking enable control for the instruction cache. **Value** / **Meaning** 0 — Disable parity/ECC 1 — Reserved 2 — Generate exceptions only on uncorrectable parity/ECC errors 3 — Generate exceptions on any type of parity/ECC errors This field is hardwired to zeros when **I-Cache Soft Error Protection** is none.	RW	0
DC_ECCEN	[5:4]	Parity/ECC error checking enable control for the data cache. **Value** / **Meaning** 0 — Disable parity/ECC 1 — Reserved 2 — Generate exceptions only on uncorrectable parity/ECC errors 3 — Generate exceptions on any type of parity/ECC errors This field is hardwired to zeros when **D-Cache Soft Error Protection** is none.	RW	0
IC_RWECC	[6]	Controls diagnostic accesses of ECC codes of the instruction cache RAMs. It is set to enable CCTL operations to access the ECC codes (see Section 10.6). This bit can be set for injecting ECC errors to test the ECC handler. **Value** / **Meaning** 0 — Disable diagnostic accesses of ECC codes 1 — Enable diagnostic accesses of ECC codes This field is hardwired to zeros when **I-Cache Soft Error Protection** is none.	RW	0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
DC_RWECC	[7]	Controls diagnostic accesses of ECC codes of the data cache RAMs. It is set to enable CCTL operations to access the ECC codes (see Section 10.6). This bit can be set for injecting ECC errors to test the ECC handler. **Value** / **Meaning** 0 — Disable diagnostic accesses of ECC codes 1 — Enable diagnostic accesses of ECC codes This field is hardwired to zeros when **D-Cache Soft Error Protection** is none.	RW	0
CCTL_SUEN	[8]	Enable bit for Superuser-mode and User-mode software to access `ucctlbeginaddr` and `ucctlcommand` CSRs. **Value** / **Meaning** 0 — Disable `ucctlbeginaddr` and `ucctlcommand` accesses in S/U mode 1 — Enable `ucctlbeginaddr` and `ucctlcommand` accesses in S/U mode	RW	0
IPREF_EN	[9]	This bit controls hardware prefetch for instruction fetches to cacheable memory regions when I-Cache size is not 0. **Value** / **Meaning** 0 — Disable hardware prefetch on instruction fetches 1 — Enable hardware prefetch on instruction fetches	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DPREF_EN	[10]	This bit controls hardware prefetch for load/store accesses to cacheable memory regions when D-Cache size is not 0.	RW	0

Value	Meaning
0	Disable hardware prefetch on load/store memory accesses.
1	Enable hardware prefetch on load/store memory accesses.

Official Release

Field Name	Bits	Description	Type	Reset
IC_FIRST_WORD	[11]	I-Cache miss allocation filling policy	RO	1

Value	Meaning
0	Cache line data returns the critical (double) word first
1	Cache line data returns the lowest address (double) word first

Field Name	Bits	Description	Type	Reset
DC_FIRST_WORD	[12]	D-Cache miss allocation filling policy	RO	1

Value	Meaning
0	Cache line data returns the critical (double) word first
1	Cache line data returns the lowest address (double) word first

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
DC_WAROUND	[14:13]	D-Cache Write-Around threshold **Value** / **Meaning** 0: Disables streaming. All cacheable write misses allocate a cache line according to PMA settings. 1: Stop allocating D-Cache entries regardless of PMA settings after consecutive stores to 4 cache lines. 2: Stop allocating D-Cache entries regardless of PMA settings after consecutive stores to 64 cache lines. 3: Stop allocating D-Cache entries regardless of PMA settings after consecutive stores to 128 cache lines.	RW	0
TLB_ECCEN	[18:17]	ECC error checking enable control for the STLB tag and data SRAMs. **Value** / **Meaning** 0: Disable ECC 1: Reserved 2: Generate exceptions only on unrepairable ECC errors 3: Generate exceptions on any type of ECC errors This field is hardwired to 0, when **Shared TLB Soft Error Protection** is configured to “none”.	RW	0

Continued on next page. . .

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DC_COHEN	[19]	Enable data cache to participate in the coherence management. **Value** / **Meaning** 0: Disable Data cache to participate in the coherence management 1: Enable Data cache to participate in the coherence management	RW	0
DC_COHSTA	[20]	Indicate if data cache is participated in the coherence management **Value** / **Meaning** 0: Data cache is not participated in the coherence management 1: Data cache is participated in the coherence management	RO	0
TLB_RWECC	[23]	Controls diagnostic accesses of ECC codes of the STLB tag and data RAMs. It is set to enable CCTL operations to access the ECC codes (see Section 10.6). This bit can be set for injecting ECC errors to test the ECC handler. **Value** / **Meaning** 0: Disable diagnostic accesses of ECC codes 1: Enable diagnostic accesses of ECC codes This field is hardwired to 0, when **Shared TLB Soft Error Protection** is configured to “none”.	RW	0

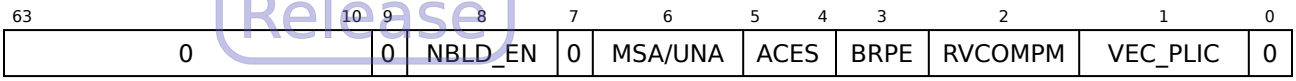
Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.7 Machine Miscellaneous Control Register

Mnemonic Name: mmisc_ctl
IM Requirement: Required
Access Mode: Machine
CSR Address: 0x7d0 (non-standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
VEC_PLIC	[1]	Selects the operation mode of PLIC: **Value** / **Meaning**: 0 = Regular mode; 1 = Vector mode Please note that both this bit and the vector mode enable bit (VECTORED) of the Feature Enable Register in NCEPLIC100 should be turned on for the vectored interrupt support to work correctly. See Section 24.5.2 for the definition of the VECTORED bit. This bit is hardwired to 0 if the vectored PLIC feature is not supported.	RW	0
RVCOMPM	[2]	RISC-V compatibility mode enable bit. If the compatibility mode is turned on, all Andes-specific instructions become reserved instructions. **Value** / **Meaning**: 0 = Disabled; 1 = Enabled For CPU revisions before v20.0.0, ACE instructions are not disabled when mmisc_ctl.RVCOMPM is set to Disabled. However, for v20.0.0 and later CPU revisions, ACE instructions will be disabled when mmisc_ctl.RVCOMPM is set to Disabled.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

<table>
<tr><th>Field Name</th><th>Bits</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td>BRPE</td><td>[3]</td><td>Branch prediction enable bit. This bit controls all branch prediction structures.
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table>This bit is hardwired to 0 if branch prediction structure is not supported.</td><td>RW</td><td>1</td></tr>
<tr><td>ACES</td><td>[5:4]</td><td>Andes Custom Extension (ACE) extension context status field:
<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Off</td></tr><tr><td>1</td><td>Initial</td></tr><tr><td>2</td><td>Clean</td></tr><tr><td>3</td><td>Dirty</td></tr></table>• This field should not be Off (0) for ACE instructions to execute normally. ACE unit enters Off state by software program through CSRW instructions.
• A normal flow to turn on ACE unit will be as follows:
– ACES is in the Off state.
– An ACE instruction executed in the Off state triggers an illegal instruction with sdcause/mdcause == 2 (ACE disabled exception).
– The exception handler initializes all ACE register states, changes this field to the Initial state, and then returns from exception.
– The ACE instruction is executed again. Since ACES is not in the Off state this time, it shall execute correctly. If any ACE register states are modified, ACES will be updated to the Dirty state automatically by hardware.
This field is hardwired to 0 if ACE extension is not configured.</td><td>RW</td><td>0</td></tr>
</table>

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MSA/UNA	[6]	This field controls whether the load/store instructions can access misaligned memory locations without generating Address Misaligned exceptions. Supported instructions: LW/LH/LHU/SW/SH/LD/LWU/SD **Value** / **Meaning** 0 — Misaligned accesses generate Address Misaligned exceptions. 1 — Misaligned accesses are allowed.	RW	1
NBLD_EN	[8]	This field controls the blocking behavior of load instructions. When this bit is clear, load instructions accessing non-device regions are blocking. When this bit is set, load instructions will not be blocking on such occasions and bus errors will no longer be reported synchronously. See Section 14.1 for details. **Value** / **Meaning** 0 — Load to memory regions are blocking. 1 — Load to memory regions are non-blocking.	RW	0

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.8 Clock Control Register

Mnemonic Name: mclk_ctl
IM Requirement: (misa.V == 1) | (mmsc_cfg.BF16CVT == 1)
Access Mode: Machine
CSR Address: 0x7df (non-standard read/write)

Official Release

This register is hardwired to zeros.

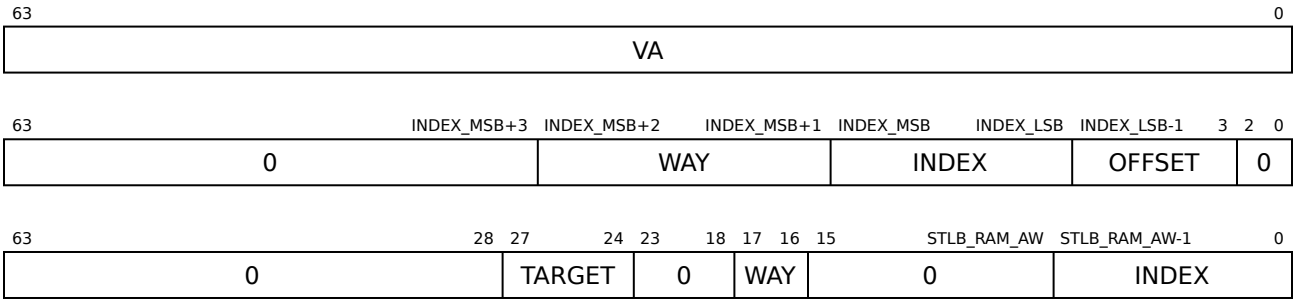
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.9 Machine CCTL Begin Address

Mnemonic Name: mcctlbeginaddr
IM Requirement: Cache optional
Access Mode: Machine
CSR Address: 0x7cb (non-standard read/write)

Official Release

This register holds the address information required by CCTL operations. It is only present when (`micm_cfg.ISZ`!=0, `mdcm_cfg.DSZ`!=0, or `mmsc_cfg.TLB_RAM_CMD`!=0) and (`mmsc_cfg.CCTLCSR`==1).



- For "VA" type of CCTL operations:

  The `mcctlbeginaddr` register contains the starting virtual address for CCTL operations triggered by writes to the `mcctlcommand` register. For CCTL lock operations, the `mcctldata` register will be updated with a status value (0:fail, 1:success) when the operations complete.

  After an update to the `mcctlcommand` register with a VA-type command, the value of this register will be incremented with the byte size of the corresponding cache line.

- For "Index" type of CCTL operations:

  The `mcctlbeginaddr` register contains the cache index for CCTL operations triggered by writes to the `mcctlcommand` register.

  – For all Index-type commands other than "L1*_IX_RDATA", "L1*_IX_WDATA", and "TLB_IX_*":

    The `WAY` field in this register will be incremented to the value of the next way. If the incremented `WAY` wraps to 0 (i.e., the first way of a set), then the `INDEX` field in this register will be incremented.

  – For "L1*_IX_RDATA" and "L1*_IX_WDATA" commands:

    The `OFFSET` field in this register will be incremented first to the next `OFFSET` value. If the incremented `OFFSET` field wraps around to 0 (i.e., the first double word of a cache line), then the `WAY` field in this register will be incremented. If the incremented `WAY` field wraps around to 0 (i.e., the first way of a set), then the `INDEX` field in this register will be incremented.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

– For “TLB_IX_*” commands:

  After an update to the `mcctlcommand` CSR with an TLB type command, the value of this register will not be changed. The `TARGET` field should be set to 3 for accessing the STLB RAMs. Otherwise, the CCTL command will be treated like NOP.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.10 Machine CCTL Command

Mnemonic Name: mcctlcommand
IM Requirement: Cache optional
Access Mode: Machine
CSR Address: 0x7cc (non-standard read/write)

Official Release

Writing to this register will trigger a CCTL operation, with the type of the operation specified by the value written. Valid CCTL operations are defined in Table 159. See Section 10.6 for more information. CCTL operations are inherently not atomic, see notes below for usage limitations.

This register is only present when (`micm_cfg.ISZ`!=0, `mdcm_cfg.DSZ`!=0, or `mmsc_cfg.TLB_RAM_CMD`!=0) and
(`mmsc_cfg.CCTLCSR`==1).

Table 159: CCTL Command Definition

	Value	Command	Type
0	0b00000_000	L1D_VA_INVAL	VA
1	0b00000_001	L1D_VA_WB	VA
2	0b00000_010	L1D_VA_WBINVAL	VA
3	0b00000_011	L1D_VA_LOCK	VA
4	0b00000_100	L1D_VA_UNLOCK	VA
6	0b00000_110	L1D_WBINVAL_ALL	-
7	0b00000_111	L1D_WB_ALL	-
8	0b00001_000	L1I_VA_INVAL	VA
11	0b00001_011	L1I_VA_LOCK	VA
12	0b00001_100	L1I_VA_UNLOCK	VA
16	0b00010_000	L1D_IX_INVAL	Index
17	0b00010_001	L1D_IX_WB	Index
18	0b00010_010	L1D_IX_WBINVAL	Index
19	0b00010_011	L1D_IX_RTAG	Index
20	0b00010_100	L1D_IX_RDATA	Index
21	0b00010_101	L1D_IX_WTAG	Index
22	0b00010_110	L1D_IX_WDATA	Index
23	0b00010_111	L1D_INVAL_ALL	-
24	0b00011_000	L1I_IX_INVAL	Index
27	0b00011_011	L1I_IX_RTAG	Index

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 159: (continued)

	Value	Command	Type
28	0b00011_100	L1I_IX_RDATA	Index
29	0b00011_101	L1I_IX_WTAG	Index
30	0b00011_110	L1I_IX_WDATA	Index
147	0b10011_011	TLB_IX_RTAG	Index
148	0b10011_100	TLB_IX_RDATA	Index
149	0b10011_101	TLB_IX_WTAG	Index
150	0b10011_110	TLB_IX_WDATA	Index

Official Release

Note

CCTL operations take parameters from multiple CSR registers, thus they are inherently not atomic. In addition, the CSRs share common storage across privilege levels, making them vulnerable to be overwritten across context switches. This implies that higher privilege level software should back up the content of CCTL CSR registers with interrupts disabled before using them, and restore their values afterwards if CCTL operations will be invoked in multiple privilege levels. The same is true if CCTL operations will be used both in non-interrupt codes and in the interrupt handlers.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.11 Machine CCTL Data

Mnemonic Name: mcctldata
IM Requirement: Cache optional
Access Mode: Machine
CSR Address: 0x7cd (non-standard read/write)

Official Release

This register holds data required/returned by some CCTL operations. It is only present when (`micm_cfg.ISZ`!=0, `mdcm_cfg.DSZ`!=0, or `mmsc_cfg.TLB_RAM_CMD`!=0) and (`mmsc_cfg.CCTLCSR`==1). The complete list of CCTL operations are summarized in Table 160 and described below.

Table 160: CCTL Commands Which Access `mcctldata`

Value of `mcctlcommand`		Command	Type
3	0b00000_011	L1D_VA_LOCK	VA
11	0b00001_011	L1I_VA_LOCK	VA
19	0b00010_011	L1D_IX_RTAG	Index
20	0b00010_100	L1D_IX_RDATA	Index
21	0b00010_101	L1D_IX_WTAG	Index
22	0b00010_110	L1D_IX_WDATA	Index
27	0b00011_011	L1I_IX_RTAG	Index
28	0b00011_100	L1I_IX_RDATA	Index
29	0b00011_101	L1I_IX_WTAG	Index
30	0b00011_110	L1I_IX_WDATA	Index
147	0b10011_011	TLB_IX_RTAG	Index
148	0b10011_100	TLB_IX_RDATA	Index
149	0b10011_101	TLB_IX_WTAG	Index
150	0b10011_110	TLB_IX_WDATA	Index

- For CCTL lock operations: The `mcctldata` register will be updated with a status value (0: fail, 1:success) when the operations complete.
- For CCTL index read/write-data operations: `mcctldata[63:0]` holds the cache data for the operations.
- For CCTL index read/write-tag operations, the `mcctldata` register holds the cache tag for the operations.
  - The bit positions of the fields for I-Cache CCTL index read/write-tag is as follows:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

* The TAG field does not hold every bit of PA[(PALEN-1):10] for the cache line. The cache tag RAMs do not hold all physical addresses down to bit 10. They only hold enough bits for tag look-ups. The unused lower order TAG bits will be reported as *Don't Care* value for tag-read operations and ignored on tag-write operations. The full PA value should be constructed using the corresponding index bits.
* I-Cache TAG RAM holds PA[(PA_LEN-1):A].
  · A = min(12:(log_2(`micm_cfg.ISET`)+6))
* If A is 11, bit 0 is unused. If A is 12, bit 0 and bit 1 are unused.
* Bit XLEN-3 holds the duplicated lock bit for I-Cache to better tolerate soft-errors.
* Bit XLEN-2 holds the lock bit.
* Bit XLEN-1 holds the valid bit.

– The bit positions of the fields for D-Cache CCTL index read/write-tag to tag RAM is as follows:
* MESI field holds the cache line status.
* Bit 3 holds the lock bit.
* The TAG field holds the tag data for the cache line. DTW = PALEN-6- log_2(`mdcm_cfg.DSET`)

Table 161: I-Cache CCTL Index Read/Write TAG Bit Fields

Field	Bit Position
TAG	[0+:PALEN-11]
LOCK_DUP	[XLEN-3]
LOCK	[XLEN-2]
VALID	[XLEN-1]

Table 162: D-Cache CCTL Index Read/Write TAG Bit Fields

Field	Bit Position
MESI	[0+:2]
LOCK	[3]
TAG	[4+:DTW]

- For index type of CCTL TLB operations:
  – The bit position of the fields for TLB CCTL index read/write-tag is as follows:

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

* The TAG field holds Partial Virtual Page Number (Partial-VPN), Address Space ID (ASID), and VALID.

– The bit position of the fields for TLB CCTL index read/write-data is as follows:

* The DATA field holds Physical Page Number (PPN), PTE information (V, R, W, X, U, G, A, and D).

Official Release

Table 163: TLB CCTL Index Read/Write TAG Bit Fields

Field	Bit Position
VALID	[0]
ASID	[1+:9]
Partial-VPN	[10+:VALEN-STLB_RAM_AW-12]

Table 164: TLB CCTL Index Read/Write DATA Bit Fields

Field	Bit Position
V	[0]
R	[1]
W	[2]
X	[3]
U	[4]
G	[5]
A	[6]
D	[7]
PPN	[8+:(PALEN-12)]

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.12 Supervisor CCTL Data

Mnemonic Name: scctldata
IM Requirement: Cache optional
Access Mode: Supervisor and above
CSR Address: 0x9cd (non-standard read/write)

Official Release

This register is only present when ((`micm_cfg.ISZ`!=0, `mdcm_cfg.DSZ`!=0, or `mmsc_cfg.TLB_RAM_CMD`!=0) and `mmsc_cfg.CCTLCSR`==1 and `misa[18]`==1). It is an alias to the `mcctldata` register and it is only accessible to Supervisor-mode software when `mcache_ctl.CCTL_SUEN` is 1. Otherwise, illegal instruction exceptions will be triggered.

Note

- S-mode software triggers CCTL operations through writing the `ucctlcommand` register, and the associated addresses for the CCTL operations are specified in the `ucctlbeginaddr` register.
- Due to the sharing of storage with `mcctldata`, interrupt-handler/M-mode software should back up `mcctldata` before use. See notes in Section 21.12.10 for usage limitations.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.13 User CCTL Begin Address

Mnemonic Name: ucctlbeginaddr
IM Requirement: Cache optional
Access Mode: User and above
CSR Address: 0x80b (non-standard read/write)

Official Release

This register is only present when ((`micm_cfg.ISZ`!=0, `mdcm_cfg.DSZ`!=0, or `mmsc_cfg.TLB_RAM_CMD`!=0) and `mmsc_cfg.CCTLCSR`==1 and `misa[20]`==1). It is an alias to the `mcctlbeginaddr` register and it is only accessible to Supervisor-mode and User-mode software when `mcache_ctl.CCTL_SUEN` is 1. Otherwise, illegal instruction exceptions will be triggered.

Note

- Both S-mode and U-mode software trigger CCTL operations through writing the `ucctlcommand` register; the associated addresses for CCTL operations are specified in the `ucctlbeginaddr` register.
- Due to the sharing of storage with `mcctlbeginaddr`, interrupt-handler/privileged-mode software should back up `mcctlbeginaddr` before use. See notes in Section 21.12.10 for usage limitations.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.12.14 User CCTL Command

Mnemonic Name: ucctlcommand
IM Requirement: Cache optional
Access Mode: User and above
CSR Address: 0x80c (non-standard read/write)

Official Release

Writing to this register will trigger a CCTL operation, with the type of the operation specified by the value written. Valid User CCTL commands are defined in Table 165. Both S-mode and U-mode software trigger CCTL operations through this register. However, if `ucctlcommand` is written by U-mode software with a command whose "U-Mode Allowed" field is 0 in Table 165, an illegal instruction exception will still be generated.

This register is only present when ((`micm_cfg.ISZ`!=0, `mdcm_cfg.DSZ`!=0, or `mmsc_cfg.TLB_RAM_CMD`!=0) and `mmsc_cfg.CCTLCSR`==1 and `misa[20]`==1) and it is an alias to the `mcctlcommand` register. This register is only accessible to Supervisor-mode and User-mode software when `mcache_ctl.CCTL_SUEN` is 1. Otherwise, illegal instruction exceptions will be triggered.

Table 165: User CCTL Command Definition

Value of `ucctlcommand`		Command	Type	U-Mode Allowed
0	0b0000_000	L1D_VA_INVAL	VA	1
1	0b0000_001	L1D_VA_WB	VA	1
2	0b0000_010	L1D_VA_WBINVAL	VA	1
3	0b0000_011	L1D_VA_LOCK	VA	0
4	0b0000_100	L1D_VA_UNLOCK	VA	0
6	0b0000_110	L1D_WBINVAL_ALL	-	0
7	0b0000_111	L1D_WB_ALL	-	0
8	0b0001_000	L1I_VA_INVAL	VA	1
11	0b0001_011	L1I_VA_LOCK	VA	0
12	0b0001_100	L1I_VA_UNLOCK	VA	0
16	0b0010_000	L1D_IX_INVAL	Index	0
17	0b0010_001	L1D_IX_WB	Index	0
18	0b0010_010	L1D_IX_WBINVAL	Index	0
19	0b0010_011	L1D_IX_RTAG	Index	0
20	0b0010_100	L1D_IX_RDATA	Index	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 165: (continued)

Value of `ucctlcommand`		Command	Type	U-Mode Allowed
21	0b0010_101	L1D_IX_WTAG	Index	0
22	0b0010_110	L1D_IX_WDATA	Index	0
23	0b0010_111	L1D_INVAL_ALL	-	0
24	0b0011_000	L1I_IX_INVAL	Index	0
27	0b0011_011	L1I_IX_RTAG	Index	0
28	0b0011_100	L1I_IX_RDATA	Index	0
29	0b0011_101	L1I_IX_WTAG	Index	0
30	0b0011_110	L1I_IX_WDATA	Index	0
147	0b10011_011	TLB_IX_RTAG	Index	0
148	0b10011_100	TLB_IX_RDATA	Index	0
149	0b10011_101	TLB_IX_WTAG	Index	0
150	0b10011_110	TLB_IX_WDATA	Index	0

Official Release

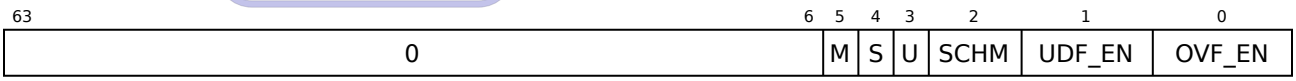
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.13 Hardware Stack Protection and Recording Registers

21.13.1 Machine Hardware Stack Protection Control

Mnemonic Name: mhsp_ctl
IM Requirement: STACKSAFE_SUPPORT = "yes" (mmsc_cfg.HSP == 1)
Access Mode: Machine
CSR Address: 0x7C6 (non-standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
OVF_EN	[0]	Enable bit for the stack overflow protection and recording mechanism. This bit will be cleared to 0 automatically by hardware when a stack protection (overflow or underflow) exception is taken.	RW	0

Value	Meaning
0	The stack overflow protection and recording mechanism are disabled.
1	The stack overflow protection and recording mechanism are enabled.

Field Name	Bits	Description	Type	Reset
UDF_EN	[1]	Enable bit for the stack underflow protection mechanism. This bit will be cleared to 0 automatically by hardware when a stack protection (overflow or underflow) exception is taken.	RW	0

Value	Meaning
0	The stack underflow protection is disabled.
1	The stack underflow protection is enabled.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
SCHM	[2]	Selects the operating scheme of the stack protection and recording mechanism. **Value** / **Meaning** 0 — Stack overflow/underflow detection 1 — Top-of-stack recording	RW	0
U	[3]	Enables the SP protection and recording mechanism in User mode. **Value** / **Meaning** 0 — The mechanism is disabled in User mode. 1 — The mechanism is enabled in User mode.	RW	0
S	[4]	Enables the SP protection and recording mechanism in Supervisor mode. **Value** / **Meaning** 0 — The mechanism is disabled in Supervisor mode. 1 — The mechanism is enabled in Supervisor mode.	RW	0
M	[5]	Enables the SP protection and recording mechanism in Machine mode. **Value** / **Meaning** 0 — The mechanism is disabled in Machine mode. 1 — The mechanism is enabled in Machine mode.	RW	0

Official Release

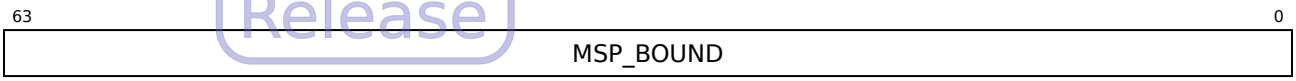
Note: Stack protection and recording support ALU, MUL, DIV and LOAD instructions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.13.2 Machine SP Bound Register

Mnemonic Name: msp_bound
IM Requirement: STACKSAFE_SUPPORT = "yes" (mmsc_cfg.HSP == 1)
Access Mode: Machine
CSR Address: 0x7c7 (non-standard read/write)

Official Release



When the `SP` overflow detection mechanism is properly selected and enabled, any updated value to the `SP` register (via any instruction) is compared with the `msp_bound` register. If the updated value to the `SP` register is smaller than the `msp_bound` register, a stack overflow exception is generated. The stack overflow exception has an exception code of 32 in the `mcause` register.

When the top of stack recording mechanism is properly selected and enabled, any updated value to the `SP` register on any instruction is compared with the `msp_bound` register. If the updated value to the `SP` register is smaller than the `msp_bound` register, the `msp_bound` register is updated with this updated value. It is an RW-type register with the all-one reset value (0xFFFFFFFF for RV32 and 0xFFFFFFFFFFFFFFFF for RV64).

Programming Note:

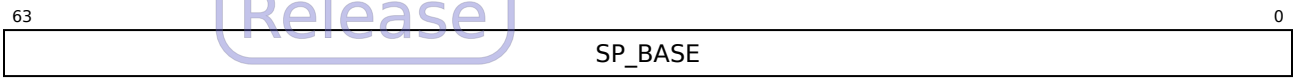
- The "`CSRRW sp, msp_bound, rs`" instruction updates both `sp` and `msp_bound` registers at the same time. When the stack overflow detection mechanism is enabled, using this instruction may generate unpredictable exception behavior.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.13.3 Machine SP Base Register

Mnemonic Name: msp_base
IM Requirement: STACKSAFE_SUPPORT = "yes" (mmsc_cfg.HSP == 1)
Access Mode: Machine
CSR Address: 0x7c8 (non-standard read/write)

Official Release



When the `SP` underflow detection mechanism is properly selected and enabled, any updated value to the `SP` register (via any instruction) is compared with the `msp_base` register. If the updated value to the `SP` register is greater than the `msp_base` register, a stack underflow exception is generated. The stack underflow exception has an exception code of 33 in the `mcause` register.

It is an RW-type register with the all-one reset value (0xFFFFFFFF for RV32 and 0xFFFFFFFFFFFFFFFF for RV64).

Programming Note:

- The "`CSRRW sp, msp_base, rs`" instruction updates both `sp` and `msp_base` registers at the same time. When the stack underflow detection mechanism is enabled, using this instruction may generate unpredictable exception behavior.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

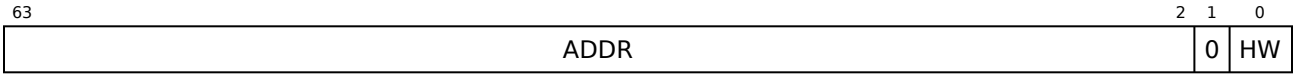
21.14 CoDense Registers

21.14.1 Instruction Table Base Address Register

Mnemonic Name: uitb
IM Requirement: CODENSE_SUPPORT = "yes"
Access Mode: User and above
CSR Address: 0x800 (non-standard read/write)

Official Release

This register defines the base address of the CoDense instruction table. Each entry in the table contains a 32-bit instruction, which can be looked up and executed by a CoDense instruction. The table is typically generated by the compiler for replacing 32-bit instructions with the shorter 16-bit Andes CoDense instructions, hence reducing the code size.



Field Name	Bits	Description	Type	Reset
HW	[0]	This bit specifies if the CoDense instruction table is hardwired. **Value** / **Meaning** 0 — The instruction table is located in memory. `uitb.ADDR` should be initialized to point to the table before using the CoDense instructions. 1 — The instruction table is hardwired. Initialization of `uitb.ADDR` is not needed before using the CoDense instructions.	RO	0
ADDR	[63:2]	The base address of the CoDense instruction table. This field is reserved if `uitb.HW` == 1.	RW	0

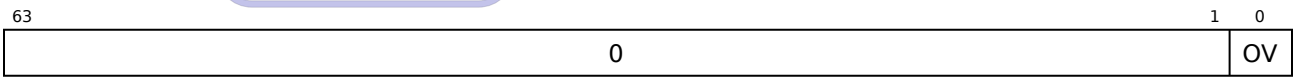
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.15 DSP Registers

21.15.1 Code Register

Mnemonic Name: ucode
IM Requirement: DSP_SUPPORT = "yes" (mmsc_cfg.EDSP == 1)
Access Mode: User
CSR Address: 0x801 (non-standard read/write)

Official Release



Field Name	Bits	Description	Type	Reset
OV	[0]	Overflow flag. It will be set by DSP instructions with a saturated result. **Value** / **Meaning** 0 — A saturated result is not generated. 1 — A saturated result is generated.	RW	0

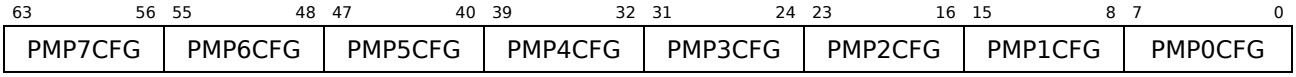
21.16 Physical Memory Protection Unit Configuration & Address Registers

21.16.1 PMP Configuration Registers

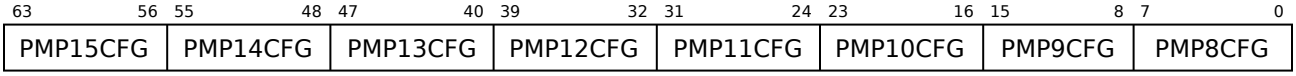
Mnemonic Name: pmpcfg0, pmpcfg2, pmpcfg4 and pmpcfg6
IM Requirement: PMP_SUPPORT
Access Mode: Machine
CSR Address: 0x3A0, 0x3A2, 0x3A4, and 0x3A6 (standard read/write)

Official Release

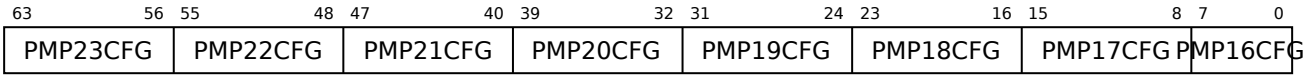
0x3A0



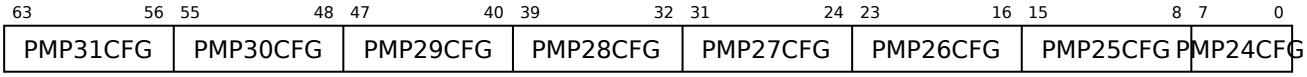
0x3A2



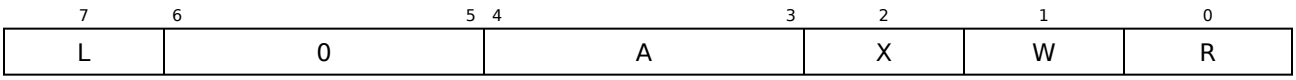
0x3A4



0x3A6



PMP Configuration Format (for PMP$_i$CFG)



Field Name	Bits	Description	Type	Reset
R	[0]	Read access control.	RW	0

Value	Meaning
0	Read accesses are not allowed.
1	Read accesses are allowed.

Field Name	Bits	Description	Type	Reset
W	[1]	Write access control.	RW	0

Value	Meaning
0	Write accesses are not allowed.
1	Write accesses are allowed.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
X	[2]	Instruction execution control. 0: Instruction execution is not allowed. 1: Instruction execution is allowed.	RW	0
A	[4:3]	Address matching mode. 0: OFF: Null region. 1: TOR: Top of range. For PMP entry 0, it matches any address *A* < `pmpaddr0`. For PMP entry *i*, it matches any address *A* such that $pmpaddr_i > A >= pmpaddr_{i-1}$. But the 4-byte range is not supported. 2: Naturally aligned four-byte region. This value is reserved when the PMP granularity is larger than 4 bytes. When the PMP granularity is larger than 4 bytes, hardware converts the written value to 0. 3: NAPOT: Naturally aligned power-of-2 region. This mode makes use of the low-order bits of the associated address register to encode the size of the range. See Table 166 for range encoding from the value of a PMP address register. The minimal size of NAPOT regions must be 8 bytes.	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
L	[7]	Write lock and permission enforcement bit for Machine mode.	W1S*	0

Value	Meaning
0	Machine mode writes to PMP entry registers are allowed. R/W/X permissions apply to S and U modes.
1	For PMP entry *i*, writes to $\texttt{PMP}_i\texttt{CFG}$ and $\texttt{PMPADDR}_i$ are ignored. Additionally, if $\texttt{PMP}_i\texttt{CFG.A}$ is set to TOR, writes to $\texttt{pmpaddr}_{i-1}$ are ignored as well. As for permission enforcement, R/W/X permissions apply to all modes. This bit can only be cleared to 0 with a system reset.

Official Release

Note

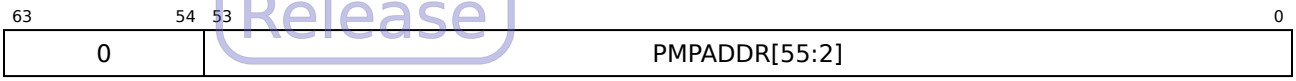
The register type of the `L` field is W1S because only a system reset can clear this bit.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.16.2 PMP Address Register

Mnemonic Name: pmpaddr0–pmpaddr31
IM Requirement: PMP_SUPPORT
Access Mode: Machine
CSR Address: 0x3b0 to 0x3cf (standard read/write)

Official Release



Each PMP address register encodes bits 55–2 of a 56-bit physical address, as shown in the register format. Different address matching mode also decides the way how PMP addressing and memory size.

If PMP configuration field A is set as 1, address matching mode becomes TOR (Top of range) mode. At TOR mode, PMP entry 0 matches any address of A < pmpaddr0. PMP entry i matches any address of pmpaddr(i-1) <= A < pmpaddr(i), where i ranges from 1 and 31. When i is 1, PMP entry 1 matches any address of pmpaddr0 <= A < pmpaddr1. The start address of PMP entry 1 is bit[55:2] of pmpaddr0, the end address is (bit[55:2]-1) of pmpaddr1 and the Memory size is (end address - start address + 1)*4 bytes.

If PMP configuration field A is set as 3, address matching mode becomes NAPOT (Naturally aligned power-of-2 region) mode. At NAPOT mode, not all physical address bits may be implemented. The encoding is described in Table 166. “a” is an arbitrary value representing one bit value of the physical address.

Table 166: AX45MP NAPOT Range Encoding in PMP Address and Configuration Registers

Register Content	Match Size(Byte)
aaaa...aaa0	8
aaaa...aa01	16
aaaa...a011	32
...	...
aa01...1111	2^{XLEN}
a011...1111	2^{XLEN+1}
0111...1111	2^{XLEN+2}
1111...1111	2^{XLEN+3} *1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

1. In Andes implementation, the behavior of this register content is the same as that of 0111...1111 and hence the match size is equivalent to 2^{XLEN+2}.

The match size of a PMP region may be coarser than the granularity (G) of PMP regions. The granularity can be 4 or 8 bytes. When the mode is OFF or TOR, and the granularity is 8, then bit 0 of $pmpaddr_i$ is hardwired to zero. The PMP granularity can be determined by writing zero to pmp_icfg, all ones to $pmpaddr_i$, then reading back $pmpaddr_i$ and detecting the lowest zero-bit to look up the size using Table 166.

Official Release

Note

When PMP is used in the cacheable memory space, a PMP region must also naturally align to the size of the cache line. Otherwise, a deliberate load to a cache line partially covered by a PMP region may bring the full cache line to the Data Cache and allow CCTL operations to access the reset of the cache line that may have a different access restriction.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

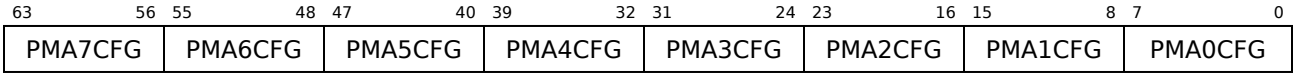
21.17 Physical Memory Attribute Unit Configuration & Address Registers

21.17.1 PMA Configuration Registers

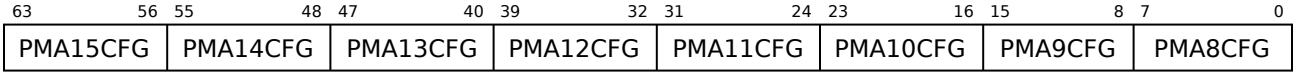
Mnemonic Name: pmacfg0 and pmacfg2
IM Requirement: PPMA_SUPPORT
Access Mode: Machine
CSR Address: 0xBC0 and 0xBC2 (standard read/write)

Official Release

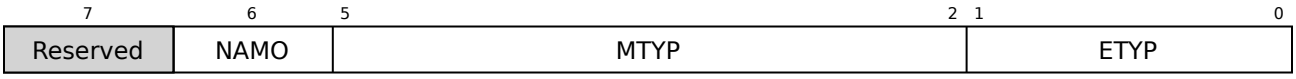
0xBC0



0xBC2



PMA Configuration Format (for PMA$_i$CFG)



Field Name	Bits	Description	Type	Reset
ETYP	[1:0]	Entry address matching mode.	RW	0

Value	Meaning
0	OFF: This PMA entry is disabled.
1	Reserved.
2	Reserved.
3	NAPOT: Naturally aligned power-of-2 region. The granularity is 4K bytes. This mode makes use of the low-order bits of the associated address register to encode the size of the range. See Table 167 for range encoding from the value of a PMA address register.

This field will be 0 when it is set to 1 or 2.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MTYP	[5:2]	Memory type attribute. This field defines the cacheability and idempotency of memory regions. In the table below, “Device” regions are non-idempotent regions and “Memory” regions are idempotent. The non-cacheable memory regions (type 2 and 3) are also referred to as uncached regions by this document.	RW	0

Official Release

Value	Meaning
0	Device, Non-bufferable
1	Device, bufferable
2	Memory, Non-cacheable, Non-bufferable
3	Memory, Non-cacheable, Bufferable
4	Reserved. Hardware converts the written value to 3
5	Reserved. Hardware converts the written value to 3
6	Reserved. Hardware converts the written value to 3
7	Reserved. Hardware converts the written value to 3
8	Memory, Write-back, No-allocate
9	Memory, Write-back, Read-allocate
10	Memory, Write-back, Write-allocate
11	Memory, Write-back, Read and Write-allocate
12 – 14	Reserved. Hardware converts the written value to 15
15	Empty hole, nothing exists. The instruction fetch will generate an instruction access fault. The load instruction access will generate a load access fault. The store instruction access will generate a store access fault.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.
Continued on next page...

Field Name	Bits	Description	Type	Reset
NAMO	[6]	Indicate whether Atomic Memory Operation instructions (including LR/SC) are not supported in this region. When this bit is set and an AMO instruction is encountered in this region, an access fault PMA NAMO exception will be generated. Atomic Memory Operation instructions (including LR/SC) to read-no-allocate memory (MTYP=8/10) are not supported. Hardware automatically sets NAMO when writing 8/10 to MTYP.	RW	0

Value	Meaning
0	AMO instructions (including LR/SC) are supported in the region.
1	AMO instructions (including LR/SC) are not supported in the region.

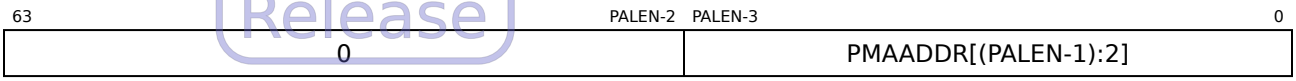
Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.17.2 PMA Address Register

Mnemonic Name: pmaaddr0–pmaaddr15
IM Requirement: PPMA_SUPPORT
Access Mode: Machine
CSR Address: 0xBD0 to 0xBDf (standard read/write)

Official Release



Each PMA address register encodes bits (PALEN-1)–2 physical address, as shown in the register format. Not all physical address bits may be implemented. The encoding is described in Table 167. "a" in the table represents one bit address, with arbitrary values.

Table 167: AX45MP NAPOT Range Encoding in PMA Address and Configuration Registers

Register Content	Match Size(Byte)
aaaa...aaaaaaaaaaa	Reserved
...	Reserved
aaaa...aa011111111	Reserved
aaaa...a0111111111	2^{12}
aaaa...01111111111	2^{13}
...	...
aa01...11111111111	2^{XLEN}
a011...11111111111	2^{XLEN+1}
0111...11111111111	2^{XLEN+2}
1111...11111111111	Reserved

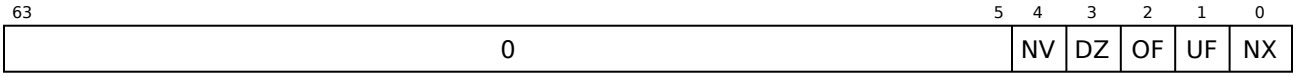
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.18 Floating-Point CSRs

21.18.1 Floating-Point Accrued Exception Flags

Mnemonic Name: fflags
IM Requirement: F or D ISA extension optional
Access Mode: User
CSR Address: 0x001 (standard read/write)

Official Release

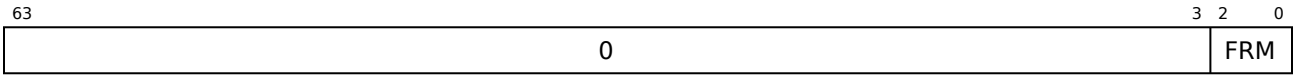


Field Name	Bits	Description	Type	Reset
NX	[0]	Inexact exception flag	RW	0
UF	[1]	Under flow exception flag	RW	0
OF	[2]	Over flow exception flag	RW	0
DZ	[3]	Divide by zero flag	RW	0
NV	[4]	Invalid operation flag	RW	0

This register is an aliased portion of the `fcsr` register.

21.18.2 Floating-Point Rounding Mode

Mnemonic Name: frm
IM Requirement: F or D ISA extension optional
Access Mode: User
CSR Address: 0x002 (standard read/write)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
FRM	[2:0]	Floating-point instruction dynamic round mode control:	RW	0

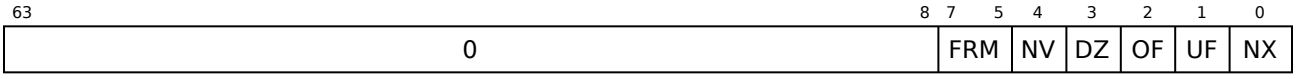
Value	Meaning
0	RNE: Round to nearest, ties to even.
1	RTZ: Round towards zero.
2	RDN: Round down (towards -infinity)
3	RUP: Round up (towards +infinity)
4	RMM: Round to nearest, ties to max magnitude
5	Reserved
6	Reserved
7	Reserved

Official Release

This register is an aliased portion of the `fcsr` register with the field position shifted to LSB.

21.18.3 Floating-Point Control and Status

Mnemonic Name: fcsr
IM Requirement: F or D ISA extension optional
Access Mode: User
CSR Address: 0x003 (standard read/write)



Field Name	Bits	Description	Type	Reset
NX	[0]	Inexact exception flag	RW	0
UF	[1]	Under flow exception flag	RW	0
OF	[2]	Over flow exception flag	RW	0
DZ	[3]	Divide by zero flag	RW	0
NV	[4]	Invalid operation flag	RW	0

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
FRM	[7:5]	Floating-point instruction dynamic round mode control:	RW	0

Value	Meaning
0	RNE: Round to nearest, ties to even.
1	RTZ: Round towards zero.
2	RDN: Round down (towards -infinity)
3	RUP: Round up (towards +infinity)
4	RMM: Round to nearest, ties to max magnitude
5	Reserved
6	Reserved
7	Reserved

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.19 User Counter Related CSRs

21.19.1 Cycle Counter

Mnemonic Name: cycle
IM Requirement: misa.U==1
Access Mode: User
CSR Address: 0xc00 (standard read only)

Official Release

This register is the read-only shadow of `mcycle` register. Writing of this register in any mode will cause an illegal instruction exception (mcause==2). When the `mcounteren.CY` bit is cleared, attempts to read this register in User mode will cause an illegal instruction exception (mcause==2).

When `mmsc_cfg.PMNDS`==1, writing of this register in any mode is controlled by the `mcounterwen` CSR. When the corresponding counter bit in `mcounterwen` is 0, writing of this register in any mode will cause an illegal instruction exception (mcause==2). When the corresponding counter bit in `mcounterwen` is 1, writing of this register in M-mode or S-mode is allowed for an M/S/U system.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.19.2 User Time Register

Mnemonic Name: time
IM Requirement: misa.U==1
Access Mode: User
CSR Address: 0xc01 (software emulation)

Official Release

This is the register for RDTIME instruction. An implementation can trap the RDTIME instruction and use a machine mode trap handler to get the timer value to emulate the correct behavior of a RDTIME instruction.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.19.3 Instruction-Retired Counter

Mnemonic Name: instret
IM Requirement: misa.U==1
Access Mode: User
CSR Address: 0xc02 (standard read only)

Official Release

This register is the read-only shadow of `mcycle` register. Writing of this register in any mode will cause an illegal instruction exception (mcause==2). When the `mcounteren.CY` bit is cleared, attempts to read this register in User mode will cause an illegal instruction exception (mcause==2).

When `mmsc_cfg.PMNDS`==1, writing of this register in any mode is controlled by the `mcounterwen` CSR. When the corresponding counter bit in `mcounterwen` is 0, writing of this register in any mode will cause an illegal instruction exception (mcause==2). When the corresponding counter bit in `mcounterwen` is 1, writing of this register in M-mode or S-mode is allowed for an M/S/U system.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

21.19.4 Performance Monitoring Counter

Mnemonic Name: hpmcounter3–hpmcounter6
IM Requirement: misa.U==1
Access Mode: User
CSR Address: 0xc03 to 0xc06 (standard read only)

Official Release

These registers are the read-only shadow of `mhpmcounter3-6` registers. Writing of these registers in any mode will cause an illegal instruction exception (mcause==2). When the `mcounteren.HPM3-6` bits are cleared, attempts to read these registers in User mode will cause an illegal instruction exception (mcause==2).

When `mmsc_cfg.PMNDS==1`, writing of this register in any mode is controlled by the `mcounterwen` CSR. When the corresponding counter bit in `mcounterwen` is 0, writing of this register in any mode will cause an illegal instruction exception (mcause==2). When the corresponding counter bit in `mcounterwen` is 1, writing of this register in M-mode or S-mode is allowed for an M/S/U system.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

22 Instruction Throughput and Latency

This chapter lists instruction throughput and latency. The instruction throughput is the number of cycles before executing the next independent instruction of the same kind. The instruction latency is the number of cycles before executing the next instruction with read-after-write dependency.

Official Release

Note

- For instruction latency, dependent instructions are assumed to be ALU instructions.

22.1 ALU Instructions

The latency and the throughput of ALU instructions are both 1 cycle. ALU instructions include:

- Add/Sub: ADD, SUB, ADDI, ADDW, SUBW, ADDIW
- Shift: SLL, SRL, SRA, SLLI, SRLI, SRAI, SLLW, SRLW, SRAW, SLLIW, SRLIW, SRAIW
- Logical: AND, OR, XOR, ANDI, ORI, XORI
- Compare: SLT, SLTU, SLTI, SLTIU
- LUI and AUIPC
- Load effective address instructions
- ADDIGP
- String processing: FFB, FFZMISM, FFMISM, FLMISM
- Bit field operation: BFOS, BFOZ
- Bit-manipulation ISA extensions
  - Zba and Zbs instructions
  - Logical with negate: ANDN, ORN, XNOR
  - Integer minimum/maximum: MAX, MAXU, MIN, MINU
  - Sign-extension and zero-extension: SEXT.B, SEXT.H, ZEXT.H
  - Bitwise rotation: ROL, ROLW, ROR, RORI, RORIW, RORW
  - OR combine: ORC.B
  - Byte-reverse: REV8

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

22.2 BITMANIP Instructions

The latency and the throughput of BITMANIP instructions are both 1 cycle. BITMANIP instructions include:

- Count leading/trailing zero bits: CLZ, CLZW, CTZ, CTZW
- Count population: CPOP, CPOPW
- Carry-less multiplication: CLMUL, CLMULH, CLMULR

Official Release

22.3 Dual-Issue Capability

This section is only for CPU revision 10.0.0 and later. For earlier revisions, see earlier documents.

22.3.1 Dual-Issue Capability of Integer Instructions

AX45MP can simultaneously execute two independent instructions. Table 168 shows dual-issue capability of integer instruction pairs.

Table 168: Dual-Issue Capability

The First Instruction	The Second Instruction	Dual-Issue
ALU/BITMANIP	ALU/BITMANIP	Yes
ALU/BITMANIP	Branch/Jump	Yes
ALU/BITMANIP	Load/Store	Yes
ALU/BITMANIP	MUL/DIV	Yes
ALU/BITMANIP	DSP (2R/1W)	Yes
ALU/BITMANIP	DSP (3R/4R/2W)	No
ALU/BITMANIP	ACE(2R/1W)	Yes
ALU/BITMANIP	ACE(3R/4R/2W)	No
Branch/Jump	ALU/BITMANIP	Yes
Branch/Jump	Branch/Jump	Yes
Branch/Jump	Load/Store	Yes
Branch/Jump	MUL/DIV	Yes
Branch/Jump	DSP (2R/1W)	Yes
Branch/Jump	DSP (3R/4R/2W)	No
Branch/Jump	ACE(2R/1W)	Yes
Branch/Jump	ACE(3R/4R/2W)	No

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 168: (continued)

The First Instruction	The Second Instruction	Dual-Issue
Load/Store	ALU/BITMANIP	Yes
Load/Store	Branch/Jump	Yes
Load/Store	Load/Store	No
Load/Store	MUL/DIV	Yes
Load/Store	DSP (2R/1W)	Yes
Load/Store	DSP (3R/4R/2W)	No
Load/Store	ACE(2R/1W)	Yes
Load/Store	ACE(3R/4R/2W)	No
MUL/DIV	ALU/BITMANIP	Yes
MUL/DIV	Branch/Jump	Yes
MUL/DIV	Load/Store	Yes
MUL/DIV	MUL/DIV	No
MUL/DIV	DSP (2R/1W)	Yes
MUL/DIV	DSP (3R/4R/2W)	No
MUL/DIV	ACE(2R/1W)	Yes
MUL/DIV	ACE(3R/4R/2W)	No
DSP (2R/1W)	ALU/BITMANIP	Yes
DSP (2R/1W)	Branch/Jump	Yes
DSP (2R/1W)	Load/Store	Yes
DSP (2R/1W)	MUL/DIV	Yes
DSP (2R/1W)	DSP (2R/1W)	No
DSP (2R/1W)	DSP (3R/4R/2W)	No
DSP (2R/1W)	ACE(2R/1W)	Yes
DSP (2R/1W)	ACE(3R/4R/2W)	No
DSP (3R/4R/2W)	ALU/BITMANIP	No
DSP (3R/4R/2W)	Branch/Jump	No
DSP (3R/4R/2W)	Load/Store	No
DSP (3R/4R/2W)	MUL/DIV	No
DSP (3R/4R/2W)	DSP (2R/1W)	No
DSP (3R/4R/2W)	DSP (3R/4R/2W)	No
DSP (3R/4R/2W)	ACE(2R/1W)	No
DSP (3R/4R/2W)	ACE(3R/4R/2W)	No
ACE(2R/1W)	ALU/BITMANIP	Yes

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 168: (continued)

The First Instruction	The Second Instruction	Dual-Issue
ACE(2R/1W)	Branch/Jump	Yes
ACE(2R/1W)	Load/Store	Yes
ACE(2R/1W)	MUL/DIV	Yes
ACE(2R/1W)	DSP (2R/1W)	Yes
ACE(2R/1W)	DSP (3R/4R/2W)	No
ACE(2R/1W)	ACE(2R/1W)	No
ACE(2R/1W)	ACE(3R/4R/2W)	No
ACE(3R/4R/2W)	ALU/BITMANIP	No
ACE(3R/4R/2W)	Branch/Jump	No
ACE(3R/4R/2W)	Load/Store	No
ACE(3R/4R/2W)	MUL/DIV	No
ACE(3R/4R/2W)	DSP (2R/1W)	No
ACE(3R/4R/2W)	DSP (3R/4R/2W)	No
ACE(3R/4R/2W)	ACE(2R/1W)	No
ACE(3R/4R/2W)	ACE(3R/4R/2W)	No

Official Release

Note

- An ACE (2R/1W) instruction reads 2 integer registers, and writes 1 register.
- An ACE (3R/4R/2W) instruction reads 3/4 integer registers, or writes 2 register.
- A DSP (2R/1W) instruction reads 2 integer registers, and writes 1 register.
- A DSP (3R/4R/2W) instruction reads 3/4 integer registers, or writes 2 register.

22.3.2 Dual-Issue Capability of Floating-Pointing Instructions for CPU Revision 10.0.0 and Later

Floating-point instructions are executed in 5 functional units:

- FMAC: FADD.x, FSUB.x, FMUL.x, FMADD.x, FMSUB.x, FNMADD.x, and FNMSUB.x
- FDIV: FDIV.x and FSQRT.x
- FMV: FSGNJ.x, FSGNJN.x, FSGNJX.x, FMV.D.X, FMV.W.X, FMV.H.X, FMV.X.D, FMV.X.W, and FMV.X.H
- FMISC: FMIN.x, FMAX.x, FCLASS.x, FEQ.x, FLT.x, FLE.x, FCVT.D.x, FCVT.S.x, FCVT.H.x, FCVT.BF16.x, FCVT.L[U].x, and FCVT.W[U].x
- LSU: FLH, FSH, FLW, FSW, FLD, FSD, C.FLWSP, C.FSWSP, C.FLDSP, C.FSDSP, C.FLW, C.FSW, C.FLD, and C.FSD

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

AX45MP can issue a floating-point instruction with another instruction as an instruction pair unless it encounters an issue restriction. The floating-point instruction can be either the first or the second instruction of the instruction pair. The issue restrictions include

- Any instruction of the instruction pair read 3/4 integer registers.
- Any instruction of the instruction pair writes 2 integer registers.
- The two instructions use the same functional unit
- The first instruction reads 3 floating-point registers and the second instruction reads 1/2/3 floating-point registers.
- The first instruction reads 2/3 floating-point registers and the second one reads 3 floating-point registers.

Official Release

22.4 Throughput and Latency for Aligned Load Instructions

The throughput and latency of load instructions are summarized in the following table.

Table 169: Load Instruction Throughput and Latency

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles)
Load word/dword from DLM/D-Cache	1	0
Load byte/halfword from DLM/D-Cache	1	1
Load word/dword from Device/Non-cacheable memory (NHART is 1, 2 or 4)	10	9
Load byte/halfword from Device/Non-cacheable (NHART is 1, 2 or 4)	10	10
Load word/dword from Device/Non-cacheable memory (NHART is 8)	12	11
Load byte/halfword from Device/Non-cacheable (NHART is 8)	12	12
Load word/dword from ILM	2	2
Load byte/halfword from ILM	2	3

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

- The latency assumes that the dependent instruction is executed in late ALU.
- The latency assumes that non-blocking load is disabled.
- The latency assumes that load hits cache.
- The 0-cycle latency assumes that the dependent instruction and the load instruction are dual-issued.
- The latency assumes that the CORE_CLK and BUS_CLK clock ratio is 1:1.
- The latency assumes that the latency of memory access is 1 cycle.
- Load instructions from the shared peripheral port region have one more cycle latency than those from the device region.

Official Release

22.5 Throughput and Latency for Misaligned Load Instructions

For misaligned data access, it will incur extra overhead in latency and throughput. Please refer to the following table for the details.

Table 170: Misaligned Load Throughput and Latency for CPU Revision 8.0.0 and Later

Instruction	Misaligned Address	Throughput (Cycles/Instruction)	Latency (Cycles)
Load halfword from DLM/D-Cache	addr[2:0] = 1/3/5	1	1
	addr[2:0] = 7	2	2
Load word from DLM/D-Cache	addr[2:0] = 1/2/3	1	1
	addr[2:0] = 5/6/7	2	2
Load dword from DLM/D-Cache	addr[2:0] != 0	2	2
Load halfword from Non-cacheable memory (NHART is 1, 2 or 4)	addr[2:0] = 1/3/5	10	10
	addr[2:0] = 7	20	20
Load word from Non-cacheable memory (NHART is 1, 2 or 4)	addr[2:0] = 1/2/3	10	10
	addr[2:0] = 5/6/7	20	20
Load dword from Non-cacheable memory (NHART is 1, 2 or 4)	addr[2:0] != 0	20	20
Load halfword from Non-cacheable memory (NHART is 8)	addr[2:0] = 1/3/5	12	12
	addr[2:0] = 7	24	24
Load word from Non-cacheable memory (NHART is 8)	addr[2:0] = 1/2/3	12	12
	addr[2:0] = 5/6/7	24	24

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 170: (continued)

Instruction	Misaligned Address	Throughput (Cycles/Instruction)	Latency (Cycles)
Load dword from Non-cacheable memory (NHART is 8)	addr[2:0] != 0	24	24
Load halfword from ILM	addr[2:0] = 1/3/5	2	3
	addr[2:0] = 7	4	4
Load word from ILM	addr[2:0] = 1/2/3	2	3
	addr[2:0] = 5/6/7	4	4
Load dword from ILM	addr[2:0] != 0	4	4

Official Release

22.6 Multiply Instructions

The latency and throughput of multiply instructions depend on the multiplier implementation.

Table 171: Multiply Instruction Throughput and Latency: Radix Multiplier

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles)
MULHU, MULW	4 + 64 / LOG2(MUL_RADIX)	4 + 64 / LOG2(MUL_RADIX)
MUL, MULH, MULHSU without negative operands	4 + 64 / LOG2(MUL_RADIX)	4 + 64 / LOG2(MUL_RADIX)
MUL, MULH, MULHSU with negative operands	5 + 64 / LOG2(MUL_RADIX)	5 + 64 / LOG2(MUL_RADIX)

Note

- When **Multiplier Implementation** is configured to “radix2”, MUL_RADIX is 2.
- When **Multiplier Implementation** is configured to “radix4”, MUL_RADIX is 4.
- When **Multiplier Implementation** is configured to “radix16”, MUL_RADIX is 16.
- When **Multiplier Implementation** is configured to “radix256”, MUL_RADIX is 256.

Table 172: Multiply Instruction Throughput and Latency: Fast Multiplier

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles)
MUL, MULH, MULHU, MULHSU, MULW	1	1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

22.7 Divide and Remainder Instructions

The divide and remainder instructions are implemented using the non-restoring division algorithm with early termination detection.

Official Release

Table 173: Divide and Remainder Instruction Throughput and Latency

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles)
DIVU, REMU	6–69	6–69
DIV, REM	6–69	6–69

Note

- The latency of 0x8000000000000001 / 0x1 is 69 cycles.
- The latency of 0x0000000000000000 / 0x1 is 6 cycles.

22.8 Branch and Jump Instruction

The branch and jump instruction throughput is 1 cycle/instruction. Branch mis-prediction penalty is 5 cycles when the branch is resolved in EX stage. Branch mis-prediction penalty is 7 cycles when the branch is resolved in LX stage.

22.9 EXEC.IT Instruction

When an EXEC.IT instruction is decoded, the pipeline takes extra 4 cycles to fetch the instruction from CoDense instruction table. The 4-cycle penalty might be hidden if execution pipeline is stalled by earlier instructions.

22.10 CSR Instruction

22.10.1 Latency Type

Accesses to CSR with CSRRW/S/C instructions may introduce extra latencies, depending on the latency type of the CSR and the operation. The following attributes are used to describe the latency type that CSR may possess.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Term	Description
WA	Wait for the previous operation to complete in all cases. Accessing a CSR with this attribute may introduce extra latency to the CSR instruction it self.
WH	Wait for the previous operation to complete only to avoid hazards in the pipeline. Accessing a CSR with this attribute may introduce extra latency to the CSR instruction it self.
RF	Re-fetch instruction after the completion of the current CSR access. Accessing a CSR with this attribute may introduce extra latency to the instruction next to the CSR instruction.

Official Release

The following table summarizes the latency due to WA or WH with respect to CSR operations.

CSR Operation	CSR With WA	CSR With WH (w/ Hazard)	CSR with WH (w/o Hazard)
CSR read-write	1 cycle	1 cycle	0 cycle
CSR read-only	1 cycle	1 cycle	0 cycle
CSR write-only	0 cycle	0 cycle	0 cycle

The following table summarizes the latency due to RF with respect to CSR operations.

CSR Operation	CSR With RF	CSR Without RF
CSR read-write	8 cycles	0 cycle
CSR read-only	0 cycle	0 cycle
CSR write-only	8 cycles	0 cycle

Note

- Assuming the instruction is re-fetched from ILM. If re-fetching requires accesses to the bus, the wait cycles can be longer than 8.

22.10.2 List of CSRs with Latency Type WH

`mvendorid`	`marchid`	`mimpid`	`mhartid`
`mstatus`	`misa`	`medeleg`	`mideleg`
`mie`	`mtvec`	`mip`	`mxstatus`
`mslideleg`	`sstatus`	`sedeleg`	`sideleg`
`sie`	`stvec`	`sip`	`slie`
`slip`	`ustatus`	`uie`	`utvec`

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

uscratch	uepc	ucause	utval
uip	mscratch	mepc	mcause
mtval	mdcause	sscratch	sepc
scause	stval	sdcause	

Official Release

22.10.3 List of CSRs with Latency Type WA

WA and WH are mutually exclusive, i.e., all CSRs that are not WH are WA.

22.10.4 List of CSRs with Latency Type RF

mvendorid	marchid	mimpid	mhartid
mstatus	misa	medeleg	mideleg
mie	mtvec	mip	mxstatus
mslideleg	mcounteren	mhpmevent3	mhpmevent4
mhpmevent5	mhpmevent6	mcounterwen	mcounterinten
mcountermask_m	mcountermask_s	mcountermask_u	mcounterovf
micm_cfg	mdcm_cfg	mmsc_cfg	fcsr
tdata1	tdata2	tdata3	tinfo
tcontrol	mcontext	scontext	mcontrol
icount	itrigger	etrigger	textra
dcsr	dexc2dbg	sstatus	sedeleg
sideleg	sie	stvec	scounteren
sip	slie	slip	satp
scountermask_m	scountermask_s	scountermask_u	scounterinten
scounterovf	shpmevent3	shpmevent4	shpmevent5
shpmevent6	ustatus	uie	utvec
uscratch	uepc	ucause	utval
uip	milmb	mdlmb	mecc_code
mnvec	mpft_ctl	mcache_ctl	mcctlcommand
scctldata	ucctlbeginaddr	ucctlcommand	mmisc_ctl
mhsp_ctl	msp_bound	msp_base	uitb
ucode	pmpcfg0	pmpcfg2	pmpcfg4
pmpcfg5	pmpcfg6	pmpcfg7	pmpaddr0
pmpaddr1	pmpaddr2	pmpaddr3	pmpaddr4
pmpaddr5	pmpaddr6	pmpaddr7	pmpaddr8
pmpaddr9	pmpaddr10	pmpaddr11	pmpaddr12

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

pmpaddr13 pmpaddr14 pmpaddr15 pmpaddr16
pmpaddr17 pmpaddr18 pmpaddr19 pmpaddr20
pmpaddr21 pmpaddr22 pmpaddr23 pmpaddr24
pmpaddr25 pmpaddr26 pmpaddr27 pmpaddr28
pmpaddr29 pmpaddr30 pmpaddr31 pmacfg0
pmacfg2 pmaaddr0 pmaaddr1 pmaaddr2
pmaaddr3 pmaaddr4 pmaaddr5 pmaaddr6
pmaaddr7 pmaaddr8 pmaaddr9 pmaaddr10
pmaaddr11 pmaaddr12 pmaaddr13 pmaaddr14
pmaaddr15

Official Release

22.11 Trap Return Instruction

The trap return instruction flushes the entire pipeline, and the penalty is 8 cycles.

22.12 FENCE Instruction

FENCE instruction flushes the entire pipeline and waits for outstanding memory access. The penalty is 13 cycles if there is no outstanding memory access.

22.13 Scalar Floating-Point Instructions for CPU Revision 10.0.0 and Later

The instruction latencies of floating-point instructions are shown in the following tables.

Table 174: Throughput and Latency for Instructions Executed in FMAC When Double-Precision Is Supported

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles) to FMAC, FDIV, FMV, FMISC	Latency (Cycles) to LSU
FADD.x, FSUB.x	1	4	2
FMUL.x	1	4	2
FMADD.x, FMSUB.x, FNMADD.x, FNMSUB.x	1	4	2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 175: Throughput and Latency for Instructions Executed in FMAC When Double-Precision Is Not Supported

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles) to FMAC, FDIV, FMV, FMISC	Latency (Cycles) to LSU
FADD.x, FSUB.x	1	3	1
FMUL.x	1	3	1
FMADD.x, FMSUB.x, FNMADD.x, FNMSUB.x	1	3	1

Official Release

Table 176: Throughput and Latency for Instructions Executed in FDIV/FMV/FMISC

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles) to FMAC, FDIV, FMV, FMISC	Latency (Cycles) to LSU/ALU
FDIV.D	32	32	33
FSQRT.D	31	31	32
FDIV.S	18	18	19
FSQRT.S	17	17	18
FDIV.H	11	11	12
FSQRT.H	10	10	11
FSGNJ.x, FSGNJN.x, FSGNJX.x	1	1	1
FMV.D.X, FMV.W.X, FMV.H.X	1	1	1
FMV.X.D, FMV.X.W, FMV.X.H	1	1	1
FMIN.x, FMAX.x	1	2	1
FCLASS.x, FEQ.x, FLT.x, FLE.x	1	2	1
FCVT.D.x, FCVT.S.x, FCVT.H.x, FCVT.BF16.x	1	2	1
FCVT.L[U].x, FCVT.W[U].x	1	2	1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

22.14 DSP Instructions

The instruction latencies of DSP instruction groups are shown in the following tables.

DSP instructions include:

Official Release

- Non-MUL Arithmetic Operation (8/16/32/64-bit): ADD8, SUB16, CLZ32, RSUB64, ...
- MUL8*8: KHM8, ...
- MUL8*8 with 32-bit ADD/SUB: SMAQA, ...
- MUL16*16: KHM16, ...
- MUL16*16 with 32-bit ADD/SUB: KMDA, ...
- MUL16*16 with 64-bit ADD/SUB: SMAL, ...
- MUL32*32: SMMUL, ...
- MUL32*32 with 32-bit ADD/SUB: KMMAC, ...
- MUL32*32 with 64-bit ADD/SUB: KMAR64, ...
- MUL32*16: SMMWB, ...
- MUL32*16 with 32-bit ADD/SUB: KMMAWB, ...

Table 177: DSP Instruction Throughput and Latency

Instruction	Throughput (Cycles/Instruction)	Latency (Cycles)
Non-MUL Arithmetic Operation	1	1
MUL8*8	1	1
MUL8*8 with 32-bit ADD/SUB	1	2
MUL16*16	1	2
MUL16*16 with 32-bit ADD/SUB	1	2
MUL16*16 with 64-bit ADD/SUB	1	3
MUL32*32	1	2
MUL32*32 with 32-bit ADD/SUB	1	2
MUL32*32 with 64-bit ADD/SUB	1	3
MUL32*16	1	2
MUL32*16 with 32-bit ADD/SUB	1	2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23 AE350 Platform

The AE350 platform is a pre-integrated AXI reference platform in Verilog implementing the AE350 memory map that contains an AX45MP processor. The block diagram is depicted in Figure 4 and Figure 5. The peripheral platform IPs may be available as either unencrypted RTL or encrypted RTL depending on the AX45MP licensing agreements.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.1 I/O Signals

The top-level module of this platform is ae350_chip. I/O signals of ae350_chip are described in Table 178. Signal types are listed below:

Official Release

Term	Description
I	Input signals
O	Output signals
I/O	Bi-directional signals

Table 178: I/O Signals

Interface	Signal Name	Type	Description
General	X_om	I	Operation mode
	X_aopd_por_b	I	Power-on reset in the always-on power domain
	X_por_b	I	Power-on reset in the main power domain
	X_hw_rstn	I	Hardware reset
	X_oschio	I/O	High frequency oscillator output
	X_oschin	I	High frequency oscillator input
	X_osclio	I/O	Low frequency oscillator output
	X_osclin	I	Low frequency oscillator input
	X_mpd_pwr_off	O	Main power domain power off indication
	X_rtc_wakeup	O	Alarm wake-up event
	X_wakeup_in	I	External wake-up event
SPI 1	X_spi1_clk	I/O	SPI clock
	X_spi1_csn	I/O	SPI chip select (Active-Low)
	X_spi1_mosi	I/O	SPI bus master output / slave input
	X_spi1_miso	I/O	SPI bus master input / slave output
	X_spi1_holdn	I/O	SPI hold (Active-Low)
	X_spi1_wpn	I/O	SPI WP (Active-Low)
SPI 2	X_spi2_clk	I/O	SPI Clock
	X_spi2_csn	I/O	SPI chip select (Active-Low)
	X_spi2_mosi	I/O	SPI bus master output / slave input
	X_spi2_miso	I/O	SPI bus master input / slave output
	X_spi2_holdn	I/O	SPI hold (Active-Low)
	X_spi2_wpn	I/O	SPI WP (Active-Low)
I2C	X_i2c_scl	I/O	I^2C clock

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 178: (continued)

Interface	Signal Name	Type	Description
	X_i2c_sda	I/O	I^2C data
JTAG	X_tdi	I	Test data input*
	X_tdo	O	Test data output*
	X_tms	I	Test mode select*
	X_tck	I	Test clock*
	X_trst	I	Test reset*
Secure Debug	X_secure_mode[1:0]	I	Secure mode selection*
GPIO	X_gpio[31:0]	I/O	General purpose I/O
UART 1	X_uart1_rxd	I	UART1 serial data input
	X_uart1_txd	O	UART1 serial data output
	X_uart1_ctsn	I	UART1 modem clear to send (Active-Low)
	X_uart1_rtsn	O	UART1 modem request to send (Active-Low)
	X_uart1_dcdn	I	UART1 modem data carrier detect (Active-Low)
	X_uart1_dsrn	I	UART1 modem data set ready (Active-Low)
	X_uart1_dtrn	O	UART1 modem data terminal ready (Active-Low)
	X_uart1_out1n	O	UART1 user-defined output 1 (Active-Low)
	X_uart1_out2n	O	UART1 user-defined output 2 (Active-Low)
	X_uart1_rin	I	UART1 modem ring indicator (Active-Low)
UART 2	X_uart2_rxd	I	UART2 serial data input
	X_uart2_txd	O	UART2 serial data output
	X_uart2_ctsn	I	UART2 modem clear to send (Active-Low)
	X_uart2_rtsn	O	UART2 modem request to send (Active-Low)
	X_uart2_dcdn	I	UART2 modem data carrier detect (Active-Low)
	X_uart2_dsrn	I	UART2 modem data set ready (Active-Low)
	X_uart2_dtrn	O	UART2 modem data terminal ready (Active-Low)
	X_uart2_out1n	O	UART2 user-defined output 1 (Active-Low)
	X_uart2_out2n	O	UART2 user-defined output 2 (Active-Low)
	X_uart2_rin	I	UART2 modem ring indicator (Active-Low)
PWM	X_pwm_ch0	O	PWM channel 0
	X_pwm_ch1	O	PWM channel 1
	X_pwm_ch2	O	PWM channel 2
	X_pwm_ch3	O	PWM channel 3

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

- All JTAG ports will be removed when macro `PLATFORM_NO_DEBUG_SUPPORT` is defined.
- JTAG ports X_tdi, X_tdo, and X_trst will be removed when macro `PLATFORM_JTAG_TWOWIRE` is defined even if `PLATFORM_NO_DEBUG_SUPPORT` is not defined.
- Secure Debug ports will be removed when `PLATFORM_NCEDBGLOCK100_SUPPORT` is not defined or `PLATFORM_NO_DEBUG_SUPPORT` is defined.
- In the AE350 platform, the reset value of `milmb.IEN` is set to 0 by ilm_boot.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.2 AE350 Memory Map

The default memory map is shown in Table 179. This map is based on the default Device Region configuration. If the Device Region configuration is modified, this map should be adjusted accordingly so that the spaces of I/O devices still reside in the device region. If any peripheral IP is not configured in Section 2, the related device region will not exist.

Official Release

Table 179: AE350 Memory Map

Address		Description
Begin	End	
0x00000000	0x7FFFFFFF	RAM Bridge
0x80000000	0x87FFFFFF	SPI1 AHB Memory/ Reset Vector
0xA0000000	0xA01FFFFF	Hart 0 Local Memory Slave Port: ILM
0xA0200000	0xA03FFFFF	Hart 0 Local Memory Slave Port: DLM
0xA0400000	0xA05FFFFF	Hart 1 Local Memory Slave Port: ILM
0xA0600000	0xA07FFFFF	Hart 1 Local Memory Slave Port: DLM
0xA0800000	0xA09FFFFF	Hart 2 Local Memory Slave Port: ILM
0xA0A00000	0xA0BFFFFF	Hart 2 Local Memory Slave Port: DLM
0xA0C00000	0xA0DFFFFF	Hart 3 Local Memory Slave Port: ILM
0xA0E00000	0xA0FFFFFF	Hart 3 Local Memory Slave Port: DLM
0xC0000000	0xC00FFFFF	BMC
0xC0200000	0xC02FFFFF	DFS
0xE0000000	0xE00FFFFF	AHB Decoder
0xE0500000	0xE05FFFFF	L2C
0xE4000000	0xE43FFFFF	PLIC
0xE6000000	0xE60FFFFF	Machine Timer
0xE6400000	0xE67FFFFF	PLIC-SWINT
0xE6800000	0xE68FFFFF	Debug Module/ Debug Vector
0xF0000000	0xF00FFFFF	APBBRG
0xF0100000	0xF01FFFFF	SMU
0xF0200000	0xF02FFFFF	UART1
0xF0300000	0xF03FFFFF	UART2
0xF0400000	0xF04FFFFF	PIT
0xF0500000	0xF05FFFFF	WDT
0xF0600000	0xF06FFFFF	RTC
0xF0700000	0xF07FFFFF	GPIO

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 179: (continued)

Address		Description
Begin	End	
0xF0A00000	0xF0AFFFFF	I2C
0xF0B00000	0xF0BFFFFF	SPI1
0xF0C00000	0xF0CFFFFF	DMAC
0xF0F00000	0xF0FFFFFF	SPI2
0xF2000000	0xF20FFFFF	DTROM
0x100000000	0x1FFFFFFFF	Uncacheable alias to region 0x00000000 - 0xFFFFFFFF. The data in cacheable regions will be cached into L1 caches of the processor. This region is an uncacheable alias to the cacheable regions. Accesses to this region will bypass the L1 caches. This is useful when the processor and external bus masters need to share data in the same memory space. This region is present only when BIU_ADDR_WIDTH is 33 bits.

Official Release

Note

- The RAM bridge space indicates the size of the RAM behind this bridge. It can be different from the size of the address space allocated to the bridge on the bus. The default setting allocates a 2GiB space (0x00000000 – 0x7FFFFFFF) to the bridge on the bus. When the bridge sees a transaction for addresses outside of the addressable RAM, it will return an error response.
- In addition to the bus view described here, ILM/DLM are accessible by the processor through private address spaces visible only to the processor:
  - The address ranges of these private address spaces are controlled by `milmb.IBPA` and `mdlmb.DBPA`.
  - The private address spaces have higher priority than the bus address spaces in the processor. Accesses will be directed to go through the local memory interfaces and bypass the bus address spaces if they hit the private address spaces.
  - If overlapping of address spaces is not desired, the `milmb` and `mdlmb` control registers could be programmed to avoid overlapping.
  - ILM is visible to the processor at 0x00000000 in the default setting.
  - DLM is visible to the processor at 0x00200000 in the default setting.
  - Debug Module region will be mapped to the default slave when macro `PLATFORM_NO_DEBUG_SUPPORT` is defined

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The base address could be modified when **Platform and CPU Subsystem** is specified to “custom”. In this case, the base address should meet the following requirements or a warning message will be displayed when ‘Generate ax45mp_core” button is pressed:

- The base address should be aligned to the minimum space requirement.
- The base address of a peripheral should be allocated in the Device Region.
- The base address should be allocated in correct region.
  - PLIC, PLMT, PLIC_SWINT, and PLDM should be allocated within PLIC base address + 64 MiB.
  - Device in AHB space (Include APB bridge program register base address + 256 MiB) should be allocated within AHB decoder program register base address + 512 MiB.
  - Device in APB space should be allocated within APB bridge program register base address + 256 MiB.
  - Reset Vector Address should be allocated in ILM, RAM bridge, or SPI memory space.
  - Debug Vector Address should be equal to PLDM base address.
- The base address should not overlap.

If any warning message shows on terminal after ‘Generate ax45mp_core” button is pressed. Any implementation on AE350 platform will return with unpredictable results.

Note

When an Error Box is pop-up. Users must resolve the configuration issue since the error is non-waivable for the CPU level settings.

23.3 Interrupt Assignment

Interrupts in a RISC-V platform are classified into two types: local interrupts and global interrupts. Local interrupts are interrupts that go directly into a RISC-V processor, and global interrupts get arbitrated through a platform-level interrupt controller (PLIC) before going into the RISC-V processor as the *external interrupts*.

Local interrupts supported by each core include non-maskable interrupts (`coreN_nmi`), machine timer interrupts (`coreN_mtip`) and software interrupts (`coreN_msip`). Additionally, external interrupt pins (`coreN_meip` and `coreN_seip`) accept arbitrated interrupt signaling from PLIC.

Table 180 summarizes the interrupt source connectivity.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

In the AE350 platform, the PLIC module is instantiated a second time with all interrupt sources tied to zero as the software interrupt controller (`PLIC_SW`). The capability of the PLIC controller to generate interrupts through its programming registers is used for generating software interrupts.

The global interrupt sources in the AE350 platform and their connectivity to PLIC are summarized in Table 181.

If any peripheral IP is not configured in Section 2, the related interrupt signal will be tied to 0.

Official Release

Table 180: AX45MP Interrupt Assignment

Interrupt Signal	Description
nmi	WDT
mtip	Machine Timer
meip	PLIC
seip	PLIC
msip	PLIC_SW

Table 181: PLIC Interrupt Source

Interrupt Source	Description
1	RTC period
2	RTC alarm
3	PIT
4	SPI1
5	SPI2
6	IIC
7	GPIO
8	UART1
9	UART2
10	DMAC
16	L2C

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.4 DMA Hardware Handshake ID

The source/destination IDs are needed for programming the handshake pairs when initiating DMA transfers. Table 182 assigns the handshake ID for all devices of the AE350 platform. The source and destination ID should not be the same for a handshake pair since the DMA controller does not support transferring data back to the same device.

Official Release

Table 182: DMA Hardware Handshake ID

DMA Handshake ID	Devices
0	SPI1 TX
1	SPI1 RX
2	SPI2 TX
3	SPI2 RX
4	UART1 TX
5	UART1 RX
6	UART2 TX
7	UART2 RX
8	I2C

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.5 Platform IP Functional Description

23.5.1 ATCAPBBRG100 – AHB-to-APB Bridge

The AHB-to-APB bridge translates AHB transactions to APB transactions targeting a specific APB slave according to the slave base address and address space size configurations. Features of the bridge include:

Official Release

- Compliant with AMBA 4 APB
- Support of 24/32 bits address width
- Support of up to 32 APB slaves
  - Including one internal slave for slave information registers and register programming
- Configurable base/size for each downstream APB slave
- Support of various synchronous AHB/APB clock ratios (N:1, N = 1, 2, 3, . . .)
- Support of write buffering

23.5.2 ATCAXI2AHB200 – AXI-to-AHB Asynchronous Bridge

ATCAXI2AHB200 is a protocol converter that translates the AMBA AXI4 protocol to the AMBA AHB-Lite protocol. Features of the AXI-to-AHB asynchronous bridge include:

- Compliant with AMBA AXI4
- Compliant with AMBA AHB-Lite
- Support of 24–64 bits address width
- Support of 32/64 data width
- Asynchronous clock and reset domains between AXI4 and AHB-Lite interfaces
- Support of narrow transfers

23.5.3 ATCBMC300 – AXI Bus Matrix

The AXI bus matrix (BMC) provides a backbone for All functional units are connected through the AXI bus matrix. Features of ATCBMC300 include:

- Compliant with AMBA AXI4
- Support of 24–64 bits address width
- Support of 32/64/128/256/512 bits data width
- Support of configurable AxID width on master ports
  - A single configurable width for all masters
  - The width of AxID on slave ports is this width plus 4
- Support of up to 16 AXI masters

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Support of up to 32 AXI slaves
  - Including one internal slave for slave information registers and register programming
- Configurable connectivity between masters and slaves
- Configurable number of outstanding transactions
- 3 programmable priority levels with round-robin arbitration

Official Release

23.5.4 ATCBMC301 – Limited Edition of ATCBMC300

Feature of ATCBMC301 include:

- Compliant with AMBA AXI4
- Support of 24–64 bits address width
- Support of 32/64/128/256/512 bits data width
- Support of configurable AxID width on master ports
  - A single configurable width for all masters
  - The width of AxID on slave ports is this width plus 4
- Support of up to 2 AXI masters
- Support of up to 3 AXI slaves
  - No internal slave device
- Configurable connectivity between masters and slaves
- Configurable number of outstanding transactions
- 3 programmable priority levels with round-robin arbitration

23.5.5 ATCBUSDEC200 – AHB Bus Decoder

ATCBUSDEC200 is an AMBA AHB-Lite decoder. It receives bus transactions from the upstream port and dispatches the transactions to the downstream ports according to the slave base address and space size configurations. This decoder also provides slave information registers for software to look up the memory space assignment information. Features of the AHB bus decoder include:

- Compliant with AMBA AHB-Lite
- One upstream AHB-Lite port
- Support of 24/32–64 bits address width
- Support of 32/64/128/256 bits data width
- Support of up to 32 downstream AHB-Lite slaves
  - Slave 0 is reserved as the internal slave for slave information registers
- Configurable base/size for each downstream AHB-Lite slave

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.5.6 ATCBUSDEC301/ATCBUSDEC302 – Limited Edition of ATCBUSDEC300

Feature of ATCBUSDEC301/ATCBUSDEC302 include:

- Compliant with the AMBA AXI4 protocol
- Supports one upstream port
- Supports up to 4 downstream ports
  - No internal slave device
- Configurable base/size for each downstream port
- Configurable address width: 24, 32-64 bits
- Configurable data width: 32, 64, 128, 256 bits

Official Release

23.5.7 ATCDMAC300 – DMA Controller

The Direct Memory Access Controller (DMAC) enhances system performance by transferring large data blocks between devices in the background to offload the processor. Features of DMAC include:

- Compliant with AMBA AXI4 and APB4
- Support of up to 8 DMA channels
- Support of up to 16 DMA request/acknowledge pairs for hardware handshake
- Support of up to two AXI master ports for data transfers
- Support of up to two configurable DMA cores
- Support of an APB slave port for DMA register programming
- Support of 24–64 bits AXI address width
- Support of 32/64/128/256 bits AXI data width
- Support of narrow transfers on the AXI bus
- Support of group round-robin arbitration scheme with 2 priority levels
- Support of chain transfers

23.5.8 ATCGPIO100 – GPIO Controller

The General Purpose I/O (GPIO) controller supports up to 32 channels with independently programmable input/output control. Features of the GPIO controller include:

- Support of up to 32 GPIO channels (pins)
- Independent control of each channel
- Programmable I/O direction
- Optional pull-up/down control
- Optional support of interrupt trigger control

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Flexible combination of interrupt trigger modes: high/low level trigger and rising/falling/both edge trigger
- Optional de-bounce functionality for input channels

23.5.9 ATCIIC100 – I2C Controller

Official Release

The I2C controller handles communications to the Inter-Integrated Circuit (IIC or I2C) serial interface. Features of the I2C controller include:

- Programmable to be either a master or a slave device
- Programmable clock/data timing
- Support of the I2C-bus Standard-mode (100 kb/s), Fast-mode (400 kb/s) and Fast-mode plus (1 Mb/s)
- Support of hardware handshaking to the DMA controller
- Support of the master-transmit, master-receive, slave-transmit and slave-receive modes
- Support of the multi-master mode
- Support of 7-bit and 10-bit addressing modes
- Support of general call addressing mode
- Support of auto clock stretch

23.5.10 ATCPIT100 – PIT Controller

The Programmable Interval Timer (PIT) controller is a set of compact multi-function timers, which can be used as pulse width modulators (PWM) or simple timers. Each multi-function timer provides the following 6 usage scenarios:

- One 32-bit timer
- Two 16-bit timers
- Four 8-bit timers
- One 16-bit PWM
- One 16-bit timer and one 8-bit PWM
- Two 8-bit timers and one 8-bit PWM

Features of the PIT controller include:

- Support of AMBA 2.0 APB bus protocol
- Support of up to 4 multi-function timers
- Six usage scenarios (combination of timer and PWM) for each multi-function timer
- Programmable source of timer clock

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.5.11 ATCRAMBRG300 – RAM Bridge

The RAM bridge controller allows standard SRAMs to be accessed on the AXI bus. Features of the RAM bridge include:

- Support of 10 – 64 bits address width
- Support of 32/64/128/256 bits data width
- Support of exclusive accesses

Official Release

23.5.12 ATCRAMBRG500 – RAM Bridge

The RAM bridge controller allows standard SRAMs to be accessed on the ILM wait-cycle interface. Features of the RAM bridge include:

- Support of 10 – 64 bits address width
- Support of 64 bits data width
- Prefetch buffers are implemented for read accesses performance improvement
- Support synchronous clock with integer clock ratio (N:1, ILM clock frequency:SRAM clock frequency)

23.5.13 ATCRTC100 – Real-Time Clock

Real-time clock (RTC) keeps track of current time relative to a base time. The time is stored in a RTC counter which records the amount of elapsed time since RTC is enabled. Features of RTC include:

- The frequency of clock source (before the clock divider) for the counter is 32.768KHz.
- Separate second, minute, hour and day counters.
- Periodic interrupts: half-second, second, minute, hour and day interrupts.
- Programmable alarm interrupt with specified second, minute and hour values.

RTC duplicates in functionality the RISC-V Machine Timer (Section 25). The RTC module is offered for compatibility of existing Andes platform software environment. It may be configured out for a pure RISC-V platform.

23.5.14 SAMPLE_DTROM – Device Tree ROM

SAMPLE_DTROM is a read-only device on the APB bus to hold the binary form of the Device Tree description. Device Tree description is a standard way for Linux and embedded software to discover platform level configurations. SAMPLE_DTROM is not enabled by default, and it can be enabled by defining `AE350_DTROM_SUPPORT` macro.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

It is okay to build a platform without SAMPLE_DTROM as software platforms can usually take alternative device tree information from other media. However, built-in device tree information in the hardware lessens the burden to manage hardware variations.

The ROM data is expected to be provided by file `sample_dtrom.data` located in the working directory of synthesis tools.

Official Release

A script **$NDS_HOME**/andes_ip/scripts/gen_dtrom.pl is provided to automatically generate a Device Tree description and binary based on ae350 platform configuration settings. It will save the generated description in the `ae350.dts` file and the compiled binary in `sample_dtrom.data`. The Device Tree Compiler `dtc` should be in the search path to run this script.

23.5.15 ATCSIZEDN300 – AXI Downsizer

The ATCSIZEDN300 bridge converts transactions between the upstream AMBA AXI4 bus of wider data width and the downstream AMBA AXI4 bus of narrower data width. Features of the AXI downsizer include:

- Compliant with AMBA AXI4
- Support of 24–64 bits address width
- Support of 4–32 bits ID width
- Support of 512-to-32, 512-to-64, 256-to-32, 256-to-64, 256-to-128, 128-to-32, 128-to-64, 64-to-32 data width conversion

23.5.16 ATCSPI200 – SPI Controller

The SPI controller handles communications to the Serial Peripheral Interface (SPI). The supported serial data formats range from 4 bits to 32 bits in length. Features of the SPI controller include:

- Compliant with AMBA 2 AHB protocol specification
- Compliant with AMBA 3 APB protocol specification
- Support of MSB/LSB first transfer
- Support of Direct Memory Access (DMA) data transfer
- Support of programmable SPI SCLK
- Support of memory-mapped access (read-only) through AHB bus or EILM bus
- Support of SPI slave mode
- Configurable Dual and Quad I/O SPI interfaces
- Configurable TX/RX FIFO depth (The depth could be 2, 4, 8, 16, 32, 64, or 128)
- Configurable programming port location on AHB/APB/EILM interfaces

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.5.17 ATCUART100 – UART Controller

The UART controller handles communications to the Universal Asynchronous Receiver/Transmitter (UART) serial interface. It has the following features:

- Compatible with the 16C550A register structure
- Support of the hardware flow control (CTS/RTS)
- Support of hardware handshaking to the DMA controller
- Option of by-8 or by-16 over-sampling frequency
- Support of 16/32/64/128-entry transmit/receive FIFO depth

Official Release

23.5.18 ATCWDT200 – Watchdog Timer

The Watchdog Timer (WDT) controller prevents the system from hanging if software is trapped in a deadlock condition. A decrementing counter (the watchdog timer) is maintained in WDT and a watchdog interrupt will be generated once the watchdog timer reaches zero. The timer should be restarted in the watchdog interrupt service routine. A secondary timer called system reset timer starts ticking after the watchdog interrupt, and it gets canceled upon restart of the watchdog timer. Should the watchdog timer be not restarted in time after the watchdog interrupt is triggered, system reset will be triggered by the system reset timer to reset the system. Features of WDT include:

- Internal/external clock source selection
- Separate timers for the watchdog interrupt and the system reset
  - Eight choices of watchdog timer intervals
  - Four choices of reset timer intervals
- Register write protection for watchdog timer control register and restart register
  - Configurable magic number for register write protection
  - Configurable magic number to restart the watchdog timer

23.5.19 ATCEXMON300 – AXI Exclusive Monitor

ATCEXMON300 is an exclusive monitor that implements AXI4 exclusive access mechanism. The monitor forwards requests from an upstream interface to a downstream interface. ATCEXMON300 monitors exclusive reads and exclusive writes and ensures that an exclusive write is only successful when the write is related to an exclusive sequence.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.6 Duplicated Copies of Platform IPs

In AE350, some IP modules need to be instantiated more than once but with different macro settings. To avoid the macro name conflict, one module is duplicated to another module. For example, atcbusdec200_rom is duplicated from atcbusdec200 with all occurrences of string "atcbusdec200" inside all related file names and file contents changed to "atcbusdec200_rom". Current duplicated modules are listed below.

Official Release

New Module Name	Duplicated From
atcbusdec200_rom	ATCBUSDEC200

Furthermore, the ae350_cpu_cluster_subsystem design additionally instantiates modules atcbmc301, atcbusdec301, and atcbusdec302. Those modules are not only duplicated but also specialized versions of other IPs as listed in the following table.

New Module Name	Specialized From
atcbmc301	ATCBMC300
atcbusdec301	ATCBUSDEC300
atcbusdec302	ATCBUSDEC300

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.7 IP Configurations

Find the configuration files of peripheral IPs under the following directory:

$NDS_HOME/andes_ip/ae350/define/

The configuration file name is **${IP_NAME}**_config.vh. In the release package, two configuration files can be found for each peripheral IP. The effective one is located under the above mentioned directory and it is specialized for AE350. The other one is located under **$NDS_HOME**/andes_ip/peripheral_ip/**$IP_NAME**/hdl/include/, which is a generic version provided by each individual IP for reference only and is not used by the AE350 platform. For example, for ATCGPIO100, two configuration files with the same name exist as:

$NDS_HOME/andes_ip/ae350/define/atcgpio100_config.vh
$NDS_HOME/andes_ip/peripheral_ip/atcgpio100/hdl/include/atcgpio100_config.vh

Only the configuration files under **$NDS_HOME**/andes_ip/ae350/define/ take effect since this directory is listed first with the +incdir+ option in the file list input to the simulator and synthesizer. However, the paths **$NDS_HOME**/andes_ip/peripheral_ip/**$IP_NAME**/hdl/include/ are still present in the file list as the +incdir+ options. The purpose is to specify the location of the **${IP_NAME}**_const.vh files, which contain constant/non-configurable macro settings for each IP. Please see the respective data sheets of the component IPs for configuration details.

23.8 Platform Configurations

The platform-level configurations are defined in the platform configuration file:

$NDS_HOME/andes_ip/ae350/top/hdl/include/ae350_config.vh

Configurations defined in ae350_config.vh are listed as follows:

Table 183: AE350 Configuration Options

Macro Name	Description
PLATFORM_PLDM_SYS_BUS_ACCESS	See Section 2.18.1.
PLATFORM_NCEDBGLOCK100_SUPPORT	See Section 2.18.2.

The platform debug related options should not be modified manually since these options are automatically generated and overwritten by the configuration tool.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.9 System Management Unit (SMU)

SMU provides versatile system management capabilities, including clock, reset and power control based on power domain partitions.

Official Release

23.9.1 Summary of Registers

SMU registers are summarized as follows:

Table 184: SMU Register Summary

Address Offset	Name	Description	Section
0x00	SYSTEMVER	SYSTEM ID & revision register	Section 23.9.2
0x04	BOARDVER	BOARD ID & revision register	Section 23.9.3
0x08	SYSTEMCFG	SYSTEM configuration register	Section 23.9.4
0x0C	SMUVER	SMU version register	Section 23.9.5
0x10	WRSR	Wake-up and reset status register	Section 23.9.6
0x14	SMUCR	SMU command register	Section 23.9.7
0x20	CER	Clock enable register	Section 23.9.8
0x24	CRR	Clock ratio register	Section 23.9.9
0x40	SCRATCH	Scratch pad register	Section 23.9.10
0x44	HART_RESET_CTL	Hart reset control register	Section 23.9.11
0x50	HART0_RESET_VECTOR_LO	Hart 0 reset vector register[31:0]	Section 23.9.12
0x54	HART1_RESET_VECTOR_LO	Hart 1 reset vector register[31:0]	Section 23.9.13
0x58	HART2_RESET_VECTOR_LO	Hart 2 reset vector register[31:0]	Section 23.9.14
0x5C	HART3_RESET_VECTOR_LO	Hart 3 reset vector register[31:0]	Section 23.9.15
0x60	HART0_RESET_VECTOR_HI	Hart 0 reset vector register[63:32]	Section 23.9.16
0x64	HART1_RESET_VECTOR_HI	Hart 1 reset vector register[63:32]	Section 23.9.17
0x68	HART2_RESET_VECTOR_HI	Hart 2 reset vector register[63:32]	Section 23.9.18
0x6C	HART3_RESET_VECTOR_HI	Hart 3 reset vector register[63:32]	Section 23.9.19

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 184: (continued)

Address Offset	Name	Description	Section
(0x80+0x20×*m*)	PCS*m*_CFG	Power control slot *m* configuration register	Section 23.9.28
(0x84+0x20×*m*)	PCS*m*_SCRATCH	Power control slot *m* scratch pad	Section 23.9.29
(0x88+0x20×*m*)	*Reserved*	Reserved	-
(0x8C+0x20×*m*)	*Reserved*	Reserved	-
(0x90+0x20×*m*)	PCS*m*_WE	Power control slot *m* wakeup enable	Section 23.9.30
(0x94+0x20×*m*)	PCS*m*_CTL	Power control slot *m* control	Section 23.9.31
(0x98+0x20×*m*)	PCS*m*_STATUS	Power control slot *m* status	Section 23.9.32
(0x9C+0x20×*m*)	*Reserved*	Reserved	-
0x200	HART4_RESET_VECTOR_LO	Hart 4 reset vector register[31:0]	Section 23.9.20
0x204	HART5_RESET_VECTOR_LO	Hart 5 reset vector register[31:0]	Section 23.9.21
0x208	HART6_RESET_VECTOR_LO	Hart 6 reset vector register[31:0]	Section 23.9.22
0x20C	HART7_RESET_VECTOR_LO	Hart 7 reset vector register[31:0]	Section 23.9.23
0x210	HART4_RESET_VECTOR_HI	Hart 4 reset vector register[63:32]	Section 23.9.24
0x214	HART5_RESET_VECTOR_HI	Hart 5 reset vector register[63:32]	Section 23.9.25
0x218	HART6_RESET_VECTOR_HI	Hart 6 reset vector register[63:32]	Section 23.9.26
0x21C	HART7_RESET_VECTOR_HI	Hart 7 reset vector register[63:32]	Section 23.9.27

Official Release

Note

1. *m* is Power control slot, 3–10.
2. Hart 0 is controlled by PCS3, Hart 1 is controlled by PCS4, and so on.
3. Power control slot 0–2 are reserved.

23.9.2 SYSTEM ID & Revision Register (SYSTEMVER) (0x00)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
MINOR	[3:0]	Minor revision number	RO	0x0
MAJOR	[7:4]	Major revision number	RO	0x0
ID	[31:8]	ID for AE350	RO	0x414535

Official Release

23.9.3 BOARD ID & Revision Register (BOARDVER) (0x04)

Field Name	Bits	Description	Type	Reset
MINOR	[3:0]	Minor revision number	RO	0x0
MAJOR	[7:4]	Major revision number	RO	0x1
ID	[31:8]	BOARD ID for AE350	RO	0x0174b0

23.9.4 SYSTEM Configuration Register (SYSTEMCFG) (0x08)

Field Name	Bits	Description	Type	Reset
CORENUM	[7:0]	Configured the number of CPU cores	RO	Configuration dependent
L2C	[8]	Indicates whether the L2-Cache controller exists	RO	Configuration dependent
DFS	[9]	Indicates whether the dynamic frequency scaling presents	RO	Configuration dependent

23.9.5 SMU Version Register (SMUVER) (0x0c)

Field Name	Bits	Description	Type	Reset
SMUVER	[31:0]	SMU Version	RO	0x100

Value	Meaning
0x0000	SMU V0
0x0100	SMU V1

23.9.6 Wake-Up and Reset Status Register (WRSR) (0x10)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
APOR	[0]	AOPD Power-On Reset	W1C	Note1

Value	Meaning
0	No action
1	Reset has occurred

Official Release

MPOR	[1]	MPD Power-On Reset	W1C	Note2

Value	Meaning
0	No action
1	Reset has occurred

HW	[2]	Hardware Reset	W1C	Note3

Value	Meaning
0	Reset didn't occur
1	Reset has occurred

WDT	[3]	Watchdog Reset	W1C	Note3

Value	Meaning
0	Reset didn't occur
1	Reset has occurred

SW	[4]	Software Reset	W1C	Note3

Value	Meaning
0	Reset didn't occur
1	Reset has occurred

WI	[8]	Wake-up by external events	W1C	0

Value	Meaning
0	Wake-up event didn't occur
1	Wake-up event has occurred

ALM	[9]	Wake-up by RTC alarm events	W1C	0

Value	Meaning
0	Wake-up event didn't occur
1	Wake-up event has occurred

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
DBG	[10]	Wake-up by debug requests	W1C	0

Value	Meaning
0	Wake-up event didn't occur
1	Wake-up event has occurred

Official Release

Note

1. APOR is reset to 1 during the AOPD power-on reset.
2. MPOR is reset to 1 during the MPD power-on reset.
3. HW, WDT, and SW are reset to 0 during the AOPD power-on reset.

23.9.7 SMU Command Register (SMUCR) (0x14)

Field Name	Bits	Description	Type	Reset
SMUCMD	[7:0]	SMU command	WO	0

Value	Meaning
0x3c	Software reset to reset the whole system.

23.9.8 Clock Enable Register (CER) (0x20)

This register controls all clocks in the platform. Processor and AHB/APB bus clocks should be turned on/off according to the programming sequence in Section 23.9.34.

Field Name	Bits	Description	Type	Reset
CCLK_EN	[0]	Processor clock enable.	RW	1

Value	Meaning
0	Disable clock
1	Enable clock

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
HCLK_EN	[1]	AHB bus clock enable. 0: Disable clock 1: Enable clock	RW	1
PCLK_EN	[2]	Main APB bus clock enable. 0: Disable clock 1: Enable clock	RW	1
Reserved	[10:3]	Reserved	RW	0
ACLK_EN	[11]	AXI bus clock enable 0: Disable clock 1: Enable clock	RW	1

Official Release

23.9.9 Clock Ratio Register (CRR) (0x24)

Field Name	Bits	Description	Type	Reset
CCLKSEL	[0]	Processor clock select 0: OSCH (Default) 1: Divide OSCH by 2	RW	0
HPCLKSEL	[3:1]	HCLK and PCLK clock ratio select 0: 1:1:1:1 1: 1:1:1:1/2 2: 1:1:1:1/4 3: 1:1:1/2:1/2 4: 1:1:1/2:1/4 5-7: Reserved	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.9.10 Scratch Pad Register (SCRATCH) (0x40)

This is a scratch register, which retains values when the rest of the system is powered down. It could be used to hold some parameters during the power down period.

Field Name	Bits	Description	Type	Reset
SCRATCH	[31:0]	Scratch register	RW	0

Official Release

23.9.11 Hart Reset Control Register (HART_RESET_CTL) (0x44)

This register is used to generate resets to harts other than Hart 0. When a bit is written 0, the corresponding hart will be reset. Write 1 to stop the reset. The bit only exists when the corresponding hart is configured.

Field Name	Bits	Description	Type	Reset
HART0_RESET	[0]	Hardwired to 1	RO	1
HART1_RESET	[1]	Hardwired to 1	RO	1
HART2_RESET	[2]	Hardwired to 1	RO	1
HART3_RESET	[3]	Hardwired to 1	RO	1

23.9.12 Hart 0 Reset Vector Register (HART0_RESET_VECTOR_LO) (0x50)

This register controls the value driven to the `hart0_reset_vector[31:0]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.13 Hart 1 Reset Vector Register (HART1_RESET_VECTOR_LO) (0x54)

This register controls the value driven to the `hart1_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 1 is configured.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.14 Hart 2 Reset Vector Register (HART2_RESET_VECTOR_LO) (0x58)

Official Release

This register controls the value driven to the `hart2_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 2 is configured.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.15 Hart 3 Reset Vector Register (HART3_RESET_VECTOR_LO) (0x5C)

This register controls the value driven to the `hart3_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 3 is configured.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.16 Hart 0 Reset Vector Register High Part (HART0_RESET_VECTOR_HI) (0x60)

This register controls the value driven to the `hart0_reset_vector[63:32]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.17 Hart 1 Reset Vector Register High Part (HART1_RESET_VECTOR_HI) (0x64)

This register controls the value driven to the `hart1_reset_vector[63:32]` input signal of the AX45MP processor.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.18 Hart 2 Reset Vector Register High Part (HART2_RESET_VECTOR_HI) (0x68)

Official Release

This register controls the value driven to the `hart2_reset_vector[63:32]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.19 Hart 3 Reset Vector Register High Part (HART3_RESET_VECTOR_HI) (0x6c)

This register controls the value driven to the `hart3_reset_vector[63:32]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.20 Hart 4 Reset Vector Register (HART4_RESET_VECTOR_LO) (0x200)

This register controls the value driven to the `hart4_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 4 is configured.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.21 Hart 5 Reset Vector Register (HART5_RESET_VECTOR_LO) (0x204)

This register controls the value driven to the `hart5_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 5 is configured.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.22 Hart 6 Reset Vector Register (HART6_RESET_VECTOR_LO) (0x208)

Official Release

This register controls the value driven to the `hart6_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 6 is configured.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.23 Hart 7 Reset Vector Register (HART7_RESET_VECTOR_LO) (0x20C)

This register controls the value driven to the `hart7_reset_vector[31:0]` input signal of the AX45MP processor. This register is present only when Hart 7 is configured.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x80000000

23.9.24 Hart 4 Reset Vector Register High Part (HART4_RESET_VECTOR_HI) (0x210)

This register controls the value driven to the `hart4_reset_vector[63:32]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.25 Hart 5 Reset Vector Register High Part (HART5_RESET_VECTOR_HI) (0x214)

This register controls the value driven to the `hart5_reset_vector[63:32]` input signal of the AX45MP processor.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.26 Hart 6 Reset Vector Register High Part (HART6_RESET_VECTOR_HI) (0x218)

Official Release

This register controls the value driven to the `hart6_reset_vector[63:32]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

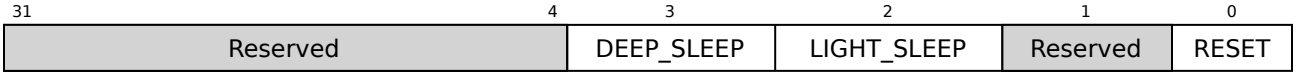
23.9.27 Hart 7 Reset Vector Register High Part (HART7_RESET_VECTOR_HI) (0x21C)

This register controls the value driven to the `hart7_reset_vector[63:32]` input signal of the AX45MP processor.

Field Name	Bits	Description	Type	Reset
RESET_VECTOR	[31:0]	Entry address upon processor reset	RW	0x00000000

23.9.28 Power Control Slot *m* Configuration Register

Mnemonic Name: PCS*m*_CFG
Address Offset: 0x80 + 0x20 × *m*
IM Requirement: Optional



Field Name	Bits	Description	Type	Reset
RESET	[0]	Power domain *m* reset control capability	RO	IM

Value	Meaning
0	Power domain *m* can not be reset.
1	Power domain *m* can be reset.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
LIGHT_SLEEP	[2]	Power domain *m* light sleep mode capability **Value** / **Meaning** 0 — Power domain *m* light sleep mode is not supported. 1 — Power domain *m* light sleep mode is supported.	RO	IM
DEEP_SLEEP	[3]	Power domain *m* deep sleep mode capability **Value** / **Meaning** 0 — Power domain *m* deep sleep mode is not supported. 1 — Power domain *m* deep sleep mode is supported.	RO	IM

Official Release

23.9.29 Power Control Slot *m* Scratch Pad

Mnemonic Name: PCS*m*_SCRATCH
Address Offset: 0x84 + 0x20 × *m*
IM Requirement: Optional

This is a scratch register, which retains values when the rest of the system is powered down. It could be used to hold some parameters during the power down period.

Field Name	Bits	Description	Type	Reset
SCRATCH	[31:0]	Scratch register	RW	0x0

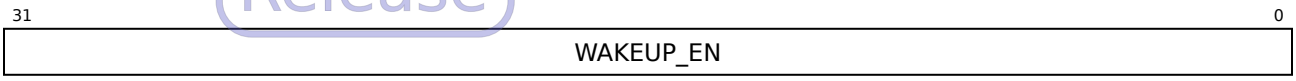
The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

23.9.30 Power Control Slot *m* Wakeup Enable

Mnemonic Name: PCS*m*_WE
Address Offset: 0x90 + 0x20 × *m*
IM Requirement: Optional

Official Release

The enable registers for wakeup event in Section 23.9.33.



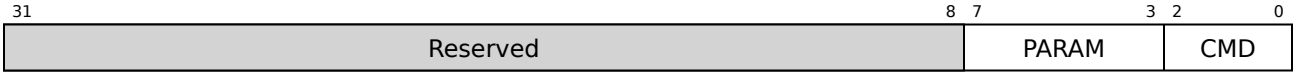
Field Name	Bits	Description	Type	Reset
WAKEUP_EN	[31:0]	Each bit indicates one wakeup event	RW	0xffffffff

Bit[n] value	Meaning
0	Disable corresponding wakeup event
1	Enable corresponding wakeup event

23.9.31 Power Control Slot *m* Control

Mnemonic Name: PCS*m*_CTL
Address Offset: 0x94 + 0x20 × *m*
IM Requirement: Optional

The control register for power control slot.



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

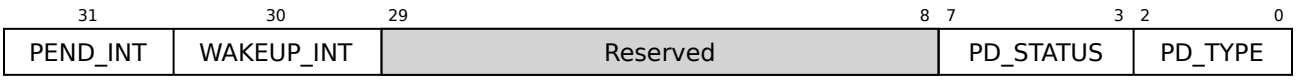
Field Name	Bits	Description	Type	Reset
CMD	[2:0]	Power control command **Value** / **Meaning** 0x0 — Active 0x1 — PCS reset 0x3 — Sleep Others — Reserved	RW	0x0
PARAM	[7:3]	The detail for power command When the CMD field is “Active”: **Value** / **Meaning** 0x1 — Wakeup Command Others — Reserved When the cmd field is “PCS reset”: **Value** / **Meaning** 0x0–0x1f — Minimal reset cycles When the cmd field is “Sleep”: **Value** / **Meaning** 0x0 — Light sleep mode 0x1 — Deep sleep mode Others — Reserved	RW	0x0

Official Release

23.9.32 Power Control Slot *m* Status

Mnemonic Name: PCS*m*_STATUS
Address Offset: 0x98 + 0x20 × *m*
IM Requirement: Optional

The status register for power control slot



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
PD_TYPE	[2:0]	Power domain status	RW	0x1

Value	Meaning
0x0	Active
0x1	Reset
0x2	Sleep
0x3	Busy_Wait
Others	Reserved

Official Release

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

<table>
<tr><th>Field Name</th><th>Bits</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td>PD_STATUS</td><td>[7:3]</td><td>The detail for power domain status This field indicates the wakeup reason when PD_TYPE is “Active”:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x0</td><td>Wakeup from system reset or PCS Wakeup Command</td></tr>
<tr><td>Others</td><td>Wakeup from other wakeup events, the value is least significant bit in wakeup event. For example, 0x8 means the PCS is wakeup from UART1 wakeup event.</td></tr>
</table>
This field indicates reset reason when PD_TYPE is “Reset”:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x4</td><td>PCS_reset (SW reset)</td></tr>
<tr><td>Others</td><td>Reserved</td></tr>
</table>
This field indicates sleep mode type when PD_TYPE is “Sleep”:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x0</td><td>Light sleep</td></tr>
<tr><td>0x10</td><td>Deep sleep</td></tr>
<tr><td>Others</td><td>Reserved</td></tr>
</table>
This field indicates wait reason when PD_TYPE is “Busy_Wait”:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0x0</td><td>Waiting WFI to enter reset state</td></tr>
<tr><td>0x2</td><td>Waiting WFI to enter light sleep state</td></tr>
<tr><td>0x3</td><td>Waiting WFI to enter deep sleep state</td></tr>
<tr><td>0x10</td><td>Busy on reset</td></tr>
<tr><td>0x12</td><td>Busy on light sleep</td></tr>
<tr><td>0x13</td><td>Busy on deep sleep</td></tr>
</table></td><td>RW</td><td>0x0</td></tr>
</table>

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
WAKEUP_INT	[30]	Pending wakeup interrupt. This bit is asserted when PCS receives a wakeup event.	W1C	0x0
PEND_INT	[31]	Pending interrupt. This bit is asserted when the PCS command is invalid.	W1C	0x0

Official Release

23.9.33 Wakeup Event

Each power domain is arranged to correlate to a set of wakeup events which are used to resume the domain. For example, SMU would wakeup a powered-down domain upon receiving a timer interrupt event which is associated to this domain.

Most SMU wakeup events come from either harts or peripherals. The wakeup events and the corresponding PCS are summarized in Table 185.

Table 185: Wakeup Events Summary

Bits	PCS	Descriptions
[31]	3	Hart 0: meip/ueip/seip
	4	Hart 1: meip/ueip/seip
	5	Hart 2: meip/ueip/seip
	6	Hart 3: meip/ueip/seip
[30]	3	Hart 0: mtip
	4	Hart 1: mtip
	5	Hart 2: mtip
	6	Hart 3: mtip
[29]	3	Hart 0: msip
	4	Hart 1: msip
	5	Hart 2: msip
	6	Hart 3: msip
[28]	3	Hart 0: debugint
	4	Hart 1: debugint
	5	Hart 2: debugint
	6	Hart 3: debugint
[27]	3	WDT, NMI is only included in PCS3 wakeup events. When the system hangs unexpectedly, PCS3 is expected to be resumed by NMI and resets the whole system.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 185: (continued)

Bits	PCS	Descriptions
[26:12]	-	Reserved
[11]	3–6	BMC
[10]	3–6	DMA
[9]	3–6	UART2
[8]	3–6	UART1
[7]	3–6	GPIO
[6]	3–6	I2C
[5]	3–6	SPI2
[4]	3–6	SPI1
[3]	3–6	PIT
[2]	3–6	RTC alarm interrupt
[1]	3–6	RTC period interrupt
[0]	-	Reserved

Official Release

23.9.34 SMU Programming Sequence

23.9.34.1 SMU Clock Control Flow

SMU supports simple clock control with the following programming sequence:

PROCESSOR CLOCK OPERATION SEQUENCE

1. Set RTC alarm in the RTC programming register (optional).
2. Set CCLK_EN to 0.
3. Set standby command in the SMU command register.
   - SMU issues an external interrupt to the processor to notify the standby request.
   - The processor should execute the `WFI` instruction to enter the WFI mode.
4. The processor clock is disabled after the processor enters the WFI mode.
5. The processor clock is enabled and the processor is waked up when a wakeup event arrives.
6. Clear the SMU command and status registers.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

BUS CLOCK OPERATION SEQUENCE

1. Set RTC alarm in the RTC programming register (optional).
2. Set PCLK_EN, HCLK_EN or CCLK_EN to 0.
3. Set standby command in the SMU command register.
   - SMU issues an external interrupt to the processor to notify the standby request.
   - The processor should execute the `WFI` instruction to enter the WFI mode.
4. The bus clock or processor clock is disabled after the processor enters the WFI mode, depending on the PCLK_EN, HCLK_EN, and CCLK_EN setting.
5. The bus clock is enabled and the processor is waked up when a wakeup event arrives.
6. Clear the SMU command and status registers.

Official Release

23.9.34.2 Power Control Slot

The system is partitioned into 3 + *n* power control slot (*n* indicates the number of harts in the platform) as below:

Table 186: Power Control Slot Summary

Power Control Slot	Descriptions
PCS3	Control clocks, resets, and power of Core 0.
PCS4	Control clocks, resets, and power of Core 1. Present when the number of cores > 1.
PCS5	Control clocks, resets, and power of Core 2. Present when the number of cores > 2.
PCS6	Control clocks, resets, and power of Core 3. Present when the number of cores > 3.

The PCS-based power control operates under the following recommended software programming sequence and scenarios:

1. The software enables proper interrupts as wakeup events for power control operations.
2. The main core informs other cores to be ready for the coming power operation by initiating the IPI (Inter Processor Interrupt). When other cores enter WFI mode, the main core issues the power operation via `PCS`*m*`_CTL` followed by a `WFI` instruction.
3. When the corresponding PCS in SMU receives the command and captures the ready-to-execute condition (e.g., the processor enters WFI mode), SMU performs power operation accordingly. The corresponding processor is finally waken up by preset wakeup events.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Wakeup events can be interrupts listed in Table 185, IPI, or writing the wakeup command to PCS*m*_CTL.

23.9.34.3 Light Sleep

Official Release

PCS 3–6 provide the light sleep command (0x3) to demonstrate the sequence defined in Section 20.5. To execute the light sleep command, apply the following sequence:

1. Set proper interrupts in PLIC and wakeup events in PCS*m*_WE.
2. Write the light sleep command to PCS*m*_CTL.

After the value is written to PCS*m*_CTL, the PCS waits until coreN_wfi_mode is asserted. Then, the PCS disables all clocks in the CORE_CLK domain. When the light sleep command is executed for all cores, SMU will automatically disable the clocks of the L2_CLK domain.

When a wakeup event occurs, the PCS enables all clocks in the CORE_CLK domain.

23.9.34.4 Deep Sleep

PCS 3–6 provide the deep sleep command (0xb) to demonstrate the sequence defined in Section 20.6.2 and Section 20.6.4. To execute the deep sleep command, apply the following sequence:

1. Set proper interrupts in PLIC and wakeup events in PCS*m*_WE.
2. Write the deep sleep command to PCS*m*_CTL.

If the processor is not the last one to be powered down, the PCS performs the following operations:

1. Wait until coreN_wfi_mode is asserted
2. Disable all clocks of CORE_CLK domain
3. Enable isolation cells of PD_CORE<n>
4. Remove power of PD_CORE<n>
5. Assert reset of the processor
6. Wait for the wakeup event

If the processor is the last one to be powered down, the PCS performs the following operations:

1. Wait until coreN_wfi_mode is asserted
2. Disable all clocks of CORE_CLK domain
3. Disable all clocks of L2_CLK domain
4. Enable isolation cells of PD_CORE<n>
5. Enable isolation cells of PD_L2
6. Remove power of PD_CORE<n>

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

7. Remove power of PD_L2
8. Assert reset of the processor
9. Assert reset of the L2-Cache
10. Wait for the wakeup event

Then, PCS waits until a wakeup event occurs before powering up the corresponding power domain.

Official Release

If the processor is the first one to be powered up, the PCS performs the following operations:

1. Apply power to the PD_CORE<n>
2. Apply power to the PD_L2
3. Disable isolation cells of PD_CORE<n>
4. Disable isolation cells of PD_L2
5. Enable all clocks in CORE_CLK domain
6. Enable all clocks in L2_CLK domain
7. De-assert the reset signal of L2-Cache
8. De-assert the reset signal of the processor

If the processor is not first one to be powered up, the PCS performs the following operations:

1. Apply power to the PD_CORE<n>
2. Disable isolation cells of the processor
3. Enable all clocks in CORE_CLK domain
4. De-assert the reset signal of the processor

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24 Platform-Level Interrupt Controller (PLIC)

24.1 Introduction

Official Release

Andes Platform-Level Interrupt Controller (NCEPLIC100) prioritizes and distributes global interrupts. It is compatible with RISC-V PLIC with the following features:

- Configurable interrupt trigger types
- Software-programmable interrupt generation
- Preemptive priority interrupt extension
- Vectored interrupt extension

See Section 24.2 for information regarding the Andes preemptive priority interrupt extension and Section 24.3 for information regarding the Andes vectored PLIC extension.

The block diagram of NCEPLIC100 is shown in Figure 20. Interrupt sources (e.g., devices) send interrupt requests to NCEPLIC100 through `int_src` signals. The signals can be level-triggered or edge-triggered, and they are converted to interrupt requests by the interrupt gateway. Interrupt requests are prioritized and routed to interrupt targets (e.g., AndesCore processor cores) according to interrupt settings. Interrupt settings include enable bits, priorities, and priority thresholds, and these settings are programmable through the bus interface. Note that interrupt targets should not modify enable bits, priorities and priority thresholds if there are any un-serviced interrupts.

`tx_eip` (*x* stands for the target number) is an external interrupt pending notification signal to the targets. It is a level signal summarizing the interrupt pending (IP) status of all interrupt sources to target *x*. When a target takes the external interrupt, it should send an interrupt claim request (bus read request) to retrieve the interrupt ID, upon which the corresponding interrupt pending status bit will be cleared and `tx_eip` will be deasserted. `tx_eip` is guaranteed to be deasserted for at least one cycle even if there are pending interrupt sources still remaining. This is done to ensure that the interrupt detection logic of the target processor can see the remaining interrupt pending status.

The interrupt gateway stops processing newer interrupt requests from its interrupt sources once it reports an interrupt request. When the target has serviced the interrupt, it should send the interrupt completion message (bus write request) to NCEPLIC100 such that the interrupt gateway resumes processing newer interrupt requests.

The interrupt pending bit array of the PLIC registers provides a summary of all interrupt sources status. In addition, it is also writable for setting software-programmed interrupts for the corresponding interrupt sources. See Section 24.5.4 for more information.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

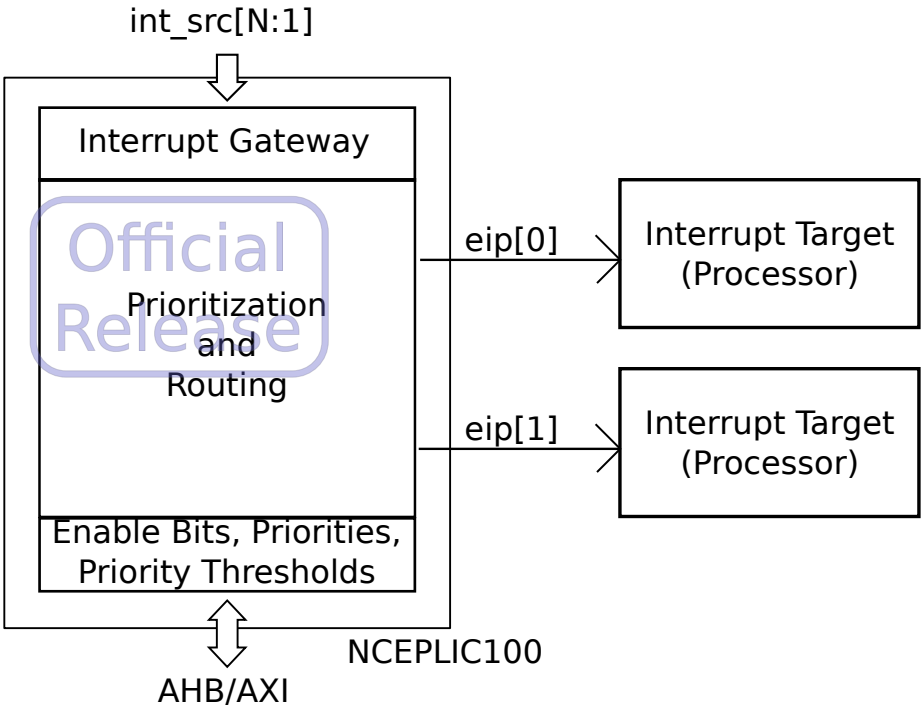


Figure 20: NCEPLIC100 Block Diagram

24.2 Support for Preemptive Priority Interrupt

NCEPLIC100 implements the Andes preemptive priority interrupt extension which enables faster responses for high-priority interrupts. This feature is enabled by setting the `PREEMPT` field (bit 0) of the Feature Enable Register (offset: 0x0000) to 1.

With this extension, if a high-priority interrupt arrives and the global interrupt is enabled (i.e.,`mstatus.MIE` or `mstatus.SIE` are 1), the processor will stop servicing the current low-priority interrupt and begin servicing this new high-priority interrupt. The handling of the suspended lower-priority interrupts will resume only after the handling of the higher-priority interrupt ends. Interrupts of same or lower priorities will not cause preemption to take effect and interfere the handling of the current interrupt. They have to wait until the handling of the current interrupt finishes.

To support this feature, the PLIC core is enhanced with a preempted priority stack for each target. The stack saves and restores priorities of the nested/preempted interrupts for the target it is associated. The operation of the preempted stack is implicitly performed through two regular PLIC operations (*Interrupt Claim* and *Interrupt Completion*). See the next two subsections for more information.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.2.1 Interrupt Claims with Preemptive Priority

When a target sends an interrupt claim message to the PLIC core, the PLIC core will atomically determine the ID of the highest-priority pending interrupt for the target and then deassert the corresponding source's IP bit. The PLIC core will then return the ID to the target.

Official Release

At the same time, the priority number in the target's Priority Threshold Register will be saved to a preempted priority stack for that target and the new priority number of the claimed interrupt will be written to the Priority Threshold Register.

24.2.2 Interrupt Completion with Preemptive Priority

When a target sends an interrupt completion message to the PLIC core, in addition to forwarding the completion message to the associated gateway, the PLIC core will restore the highest priority number in the preempted priority stack back to the Priority Threshold Register of the target.

Note that out-of-order completion of interrupts is not allowed when this feature is turned on — the latest claimed interrupt should be completed first.

24.2.3 Programming Sequence to Allow Preemption of Interrupts

Turning on the global interrupt enable flag (`mstatus.MIE` or `mstatus.SIE`) is all it takes to allow the current interrupt handler to be preempted by higher priority interrupts. However, as the preemptive priority stack operations do not allow out-of-order completion, some care should be taken to make sure that the claim and completion operations are nested properly.

For the non-vectored mode single-entry interrupt handler, the global interrupt enable flag could be turned on after the processor context are saved and *Interrupt Claim* is performed to allow preemption of the current interrupt handler. At the end of interrupt handler, an *Interrupt Completion* message is performed to signal that the handler has processed the interrupt and PLIC may deliver the next interrupt from the same interrupt source again. As both claim and completion messages are done through load/store instructions to device regions, they should automatically be ordered correctly. Compared with the vectored mode interrupt handler two paragraphs below, the global interrupt flag does not need to be disabled and no `FENCE` needs to be inserted after sending the completion message.

In summary, below is the suggested sequence for a non-vector mode interrupt handler for supporting preemptive priority interrupts:

1. Save registers/CSRs to stack
2. Sends *Interrupt Claim* message to PLIC (device-load)
3. Enable global interrupt (`mstatus.MIE/SIE`)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

4. Handle the expected interrupt
5. Sends *Interrupt Completion* message to PLIC (device-store)
6. Restore registers/CSRs
7. Return from interrupt

Official Release

For vector mode interrupt handlers (see the next section), *Interrupt Claim* is implicit when the external interrupt is taken. The global interrupt enable flag could be turned on as long as the processor context are saved to allow preemption of the current interrupt handler. However, the global interrupt flag should be turned off *before Interrupt Completion* operations are performed, since the processor will trigger the next implicit *Interrupt Claim* operation as soon as the global interrupt enable flag is turned on and cause races between *Interrupt Claim* and *Interrupt Completion*. Additionally, a `FENCE io,io` operation should be inserted after the *Interrupt Completion* operation to make sure that the completion message reaches PLIC before the interrupt handler returns, which turns on the interrupt enable flag again and cause the next *Interrupt Claim* to be performed.

In summary, below is the suggested sequence for a vector mode interrupt handler for supporting preemptive priority interrupts:

1. Save registers/CSRs to stack
2. Wait for PLIC to de-asserts the interrupt pending signal (See NOTE)
3. Enable global interrupt (`mstatus.MIE/SIE`)
4. Handle the expected interrupt
5. Disable global interrupt (`mstatus.MIE/SIE`)
6. Sends *Interrupt Completion* message to PLIC (device-store)
7. Restore registers/CSRs
8. Use a `FENCE io,io` instruction to ensure that the completion message has reached PLIC.
9. Return from interrupt

Note

Typically, PLIC de-asserts the interrupt pending signal within a few clock cycles after the core enters the trap handler. If PLIC runs at a slow clock frequency, the interrupt pending signal might still be asserted after the registers/CSRs are saved. In such cases, the core needs to wait for the EIP signal to be de-asserted before enabling the global interrupt.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.3 Vectored Interrupts

NCEPLIC100 enhances the RISC-V PLIC functionality with the vector mode extension to allow the interrupt target to receive the interrupt source ID without going through the target claim request protocol. This feature can shorten the latency of interrupt handling by enabling the interrupt target to run the corresponding interrupt handler directly upon accepting the external interrupt. It is enabled by setting the VECTORED field of the Feature Enable Register (offset: 0x0000) to 1.

Official Release

Two extra interface signals, tx_eiid and tx_eiack, are added to facilitate interrupt handling in the vector mode. When a valid interrupt is sent to PLIC, PLIC would send tx_eip with tx_eiid. Upon accepting the interrupt, the target asserts tx_eiack to PLIC and takes tx_eiid as the interrupt source ID. The assertion of tx_eiack would cause the deassertion of tx_eip and clearing of the tx_IP bit of corresponding interrupt source as does the handling of the interrupt claim request.

Note that interrupt *completion* messages are still required to notify the interrupt gateway the completion of interrupt handling and to forward additional interrupts to the PLIC core.

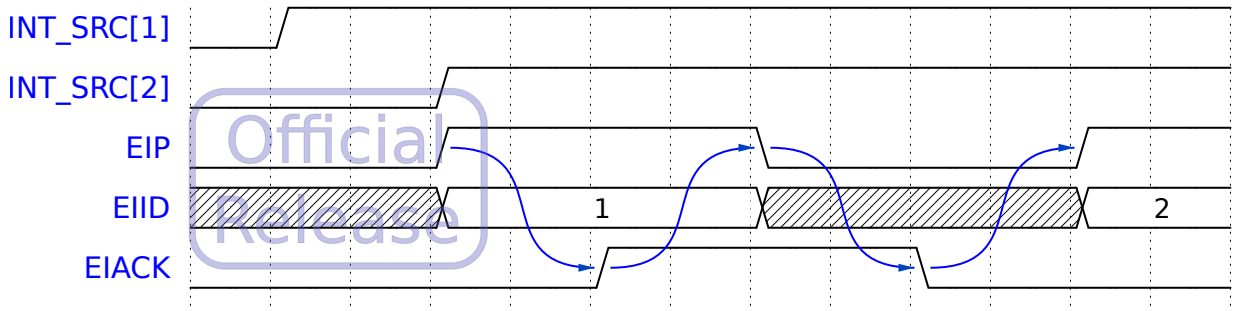
The interrupt priority arbitration works differently under the vector mode. In the non-vector mode, PLIC continues to arbitrate among all active interrupts even after the target is notified of occurrence of *some* interrupts (tx_eip sent to the target). Arbitration is not final until the claim request message is processed. In the vector mode, interrupt arbitration is final as soon as tx_eip is posted to the interrupt target. Interrupt arbitration resumes after tx_eiack is replied to PLIC and tx_eip is deasserted. The protocol does not change tx_eiid on the fly to allow PLIC and the interrupt target to operate at different clock domains.

The vector mode extension is designed such that each interrupt source is statically assigned to a single target. No two targets should compete servicing (claiming) the same interrupt source through the handshaking interface signals (tx_eiack). Otherwise, unpredictable results may occur.

Despite automatic dispatching, the PLIC interrupt claim request protocol still works under the vector mode for the interrupt handler to claim additional interrupts.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.3.1 Vector Mode Protocol



Official Release

Figure 21: NCEPLIC100 Vector Mode Protocol

- `tx_eiid` remains stable when `tx_eip` is asserted.
- When `tx_eip` is asserted, it remains asserted until `tx_eiack` is asserted or an interrupt claim request is sent to PLIC.
- The assertion of `tx_eiack` causes the deassertion of `tx_eip`, which in turn causes the deassertion of `tx_eiack`.
- If there are more pending interrupts, `tx_eip` is asserted again after deassertion of `tx_eiack`.

24.4 PLIC Configuration Options

Table 187 summarizes all supported configuration parameters and the subsections describe the parameters in detail.

Table 187: PLIC Configuration Parameters

Parameter Name	Type	Valid Values	Default Value
INT_NUM	Integer	1–1023	63
TARGET_NUM	Integer	1–16	1
MAX_PRIORITY	Integer	3/7/15/31/63/127/255	15
EDGE_TRIGGER	Integer	See Section 24.4.4	0
ASYNC_INT	Integer	See Section 24.4.5	0
ADDR_WIDTH	Integer	≥22	32
DATA_WIDTH	Integer	32/64	32
VECTOR_PLIC_SUPPORT	String	yes/no	yes
PLIC_BUS	String	ahb/axi	axi
ID_WIDTH	Integer	4–32	4
SYNC_STAGE	Integer	2/3	2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.4.1 Number of Interrupts

INT_NUM determines the number of interrupts, and the maximal value is 1023.

24.4.2 Number of Targets

Official Release

TARGET_NUM determines the number of interrupt targets, and the maximal value is 16.

24.4.3 Maximum Interrupt Priority

MAX_PRIORITY determines the valid priority levels of the interrupt sources and the target threshold. The priority value 0 is reserved to mean “never interrupt”, and the larger the priority value, the higher the interrupt priority.

24.4.4 Edge Trigger

EDGE_TRIGGER is regarded as a bit vector and each bit controls whether the corresponding interrupt source is level-triggered or edge-triggered:

- Value 0 means level-triggered; and
- Value 1 means edge-triggered.

The bit width of EDGE_TRIGGER should be (INT_NUM+1).

For example, value 0 means none of the interrupt source is edge-triggered.

24.4.5 Asynchronous Interrupt Source

ASYNC_INT is a bit vector where each bit indicates whether the corresponding interrupt source is asynchronous or synchronous.

- Value 0 means the interrupt source is synchronous.
- Value 1 means the interrupt source is asynchronous.

The bit width of ASYNC_INT should be (INT_NUM+1).

For example, value 0xc000000 means interrupt 26 and 27 of the interrupt source are asynchronous interrupts, and the rest are all synchronous ones.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.4.6 Address Width of PLIC Bus Interface

ADDR_WIDTH determines the address width of PLIC bus. The address width should be greater than or equal to 22 to encompass all addressable PLIC memory space.

Official Release

24.4.7 Data Width of PLIC Bus Interface

DATA_WIDTH determines the data width of PLIC bus.

24.4.8 Support For Vectored PLIC Extension

VECTOR_PLIC_SUPPORT controls whether to include Andes Vectored PLIC Extension or not. Note that while VECTOR_PLIC_SUPPORT is supported, harts can still perform *Interrupt Claim* operations through the AXI bus and the gate count difference of enabling VECTOR_PLIC_SUPPORT is minor to PLIC.

For PLIC integrated on the AE350 platform, VECTOR_PLIC_SUPPORT is automatically aligned to the CPU core setting by the configuration tool.

24.4.9 Bus Type of PLIC

PLIC_BUS determines the bus type of PLIC.

- String “ahb” indicates PLIC is an AHB slave device.
- String “axi” indicates PLIC is an AXI4 slave device.

24.4.10 ID Width of PLIC Bus Interface

ID_WIDTH determines the ID width of PLIC Bus Interface.

24.4.11 Synchronizer Level

SYNC_STAGE configures the level of the CDC synchronizer.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.5 PLIC Registers

24.5.1 Summary of Registers

NCEPLIC100 registers are accessed through bus transfers, and the summary of registers is shown in Table 188.

Official Release

Please note that NCEPLIC100 supports only 32-bit transfers. Behaviors of 8-bit and 16-bit transfers are UNDEFINED, and the transfers might be ignored or result in error responses.

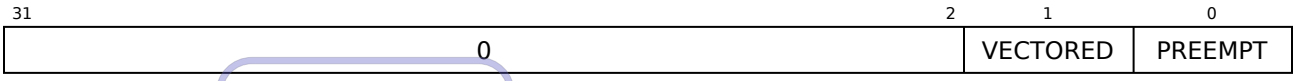
Table 188: PLIC Register Summary

Address Offset		Description	Section
Begin	End		
0x000000	0x000003	Feature enable register	Section 24.5.2
0x000004	0x000007	Source 1 priority	Section 24.5.3
0x000008	0x00000B	Source 2 priority	
...	...	...	
0x000FFC	0x000FFF	Source 1023 priority	
0x001000	0x00107F	Pending array	Section 24.5.4
0x001080	0x0010FF	Trigger type array	Section 24.5.5
0x001100	0x001103	Number of interrupts and targets	Section 24.5.6
0x001104	0x001107	Version and max priority register	Section 24.5.7
0x002000	0x00207F	Target 0 interrupt enable bits	Section 24.5.8
0x002080	0x0020FF	Target 1 interrupt enable bits	
...	...	...	
0x002780	0x0027FF	Target 15 interrupt enable bits	
0x200000	0x200003	Target 0 priority threshold	Section 24.5.9
0x200004	0x200007	Target 0 claim/complete	Section 24.5.10
0x200400	0x20041F	Target 0 preempted priority stack	Section 24.5.11
0x201000	0x20141F	Target 1 priority threshold, claim/complete, preempted priority stack	
...	...	...	
0x20F000	0x20F41F	Target 15 priority threshold, claim/complete, preempted priority stack	

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.5.2 Feature Enable Register

Offset: 0x0



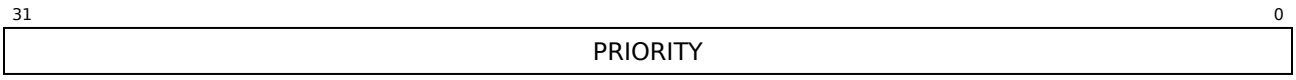
Official Release

This register enables preemptive priority interrupt feature and the vector mode.

Field Name	Bits	Description	Type	Reset
PREEMPT	[0]	Preemptive priority interrupt enable **Value** / **Meaning** 0 — Disabled 1 — Enabled	RW	0
VECTORED	[1]	Vector mode enable **Value** / **Meaning** 0 — Disabled 1 — Enabled Please note that both this bit and the `mmisc_ctl.VEC_PLIC` bit of the processor should be turned on for the vectored interrupt support to work correctly. See Section 21.12.7 for the definition of the `VEC_PLIC` bit.	RW	0

24.5.3 Interrupt Source Priority

Offset: n*4



This register determines the priority for interrupt source n ($1 \leq n \leq 1023$).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
PRIORITY	[31:0]	Interrupt source priority. The valid range of this field is determined by the MAX_PRIORITY field of the Version & Maximum Priority Configuration Register.	RW	1

Value	Meaning
0	Never interrupt.
1–255	Interrupt source priority. The larger the value, the higher the priority.

Official Release

24.5.4 Interrupt Pending

Offset: 0x1000 to 0x107F

These registers provide the interrupt pending status of interrupt sources, and a way for software to trigger an interrupt without relying on external devices. Every interrupt source occupies 1 bit. There are a total of 32 registers, each 32-bit wide, for 1023 interrupt sources. Note that zero is not a valid interrupt source number so bit 0 of the first register is hardwired to 0.

When these registers are read, the interrupt pending status of interrupt sources are returned. The pending bits could be set by writing a bit mask that specifies the bit positions to be set, and this action would result in software-programmed interrupts of the corresponding interrupt sources. The pending bits could only be cleared through the *Interrupt Claim* requests.

The location of the interrupt pending bit for interrupt source n ($1 \leq n \leq 1023$) can be determined by the following equations:

- Word offset address: 0x1000 + 4 * floor(n/32)
- Bit Position: n modulo 32

24.5.5 Interrupt Trigger Type

Offset: 0x1080 to 0x10FF

These registers are read-only and indicate the configured interrupt trigger type of interrupt sources. Every interrupt source occupies 1 bit. There are a total of 32 registers, each 32-bit wide, for 1023 interrupt sources. The location of the interrupt trigger type bit for interrupt source n ($1 \leq n \leq 1023$) can be determined by the following equations:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Word offset address: 0x1080 + 4 * floor(n/32)
- Bit Position: n modulo 32

The meaning of each bit is shown in Table 189. Note that zero is not a valid interrupt source number so bit 0 of the first register is hardwired to 0.

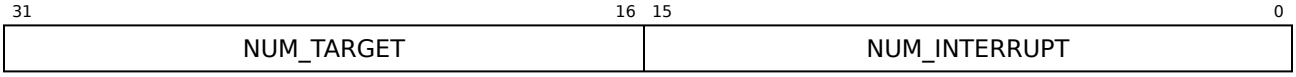
Official Release

Table 189: Meaning of Trigger Type

Value	Meaning
0	Level-triggered interrupt
1	Edge-triggered interrupt

24.5.6 Number of Interrupt and Target Configuration Register

Offset: 0x1100

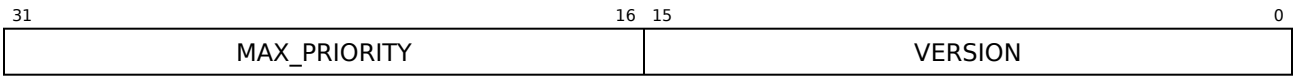


This register indicates the number of supported interrupt sources and supported targets.

Field Name	Bits	Description	Type	Reset
NUM_INTERRUPT	[15:0]	The number of supported interrupt sources	RO	IM
NUM_TARGET	[31:16]	The number of supported targets	RO	IM

24.5.7 Version & Maximum Priority Configuration Register

Offset: 0x1104



This register indicates the version and the maximum priority of PLIC implementation.

Field Name	Bits	Description	Type	Reset
VERSION	[15:0]	The version of the PLIC design	RO	IM
MAX_PRIORITY	[31:16]	The maximum priority supported	RO	IM

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.5.8 Interrupt Enable Bits for Target *m*

Offset: (0x2000+*m**128) to (0x207F+*m**128)

These registers control the routing of interrupt source n to target m ($1 \leq n \leq 1023$ and $m \geq 0$). Each bit controls one interrupt source. For each target, there are a total of 32 word-sized registers for controlling 1023 interrupt sources to that target. Note that zero is not a valid interrupt source number so bit 0 of the first register is hardwired to 0.

Official Release

The location of the interrupt enable bit for interrupt source n to target m can be determined by the following equations:

- Word offset address: 0x2000 + 128 * m + 4 * floor(n/32)
- Bit position: n modulo 32

The following pseudo code demonstrates how to enable interrupt n for target m:

```
intptr_t reg_offset = 0x2000 + 128 * m + 4 * (n/32);
int bit_position = n % 32;

volatile uint32_t *reg_pointer = (volatile uint32_t *)(PLIC_BASE+reg_offset);

*reg_pointer = *reg_pointer | (1 << bit_position);
```

24.5.9 Priority Threshold for Target *m*

Offset: 0x200000+4096**m*

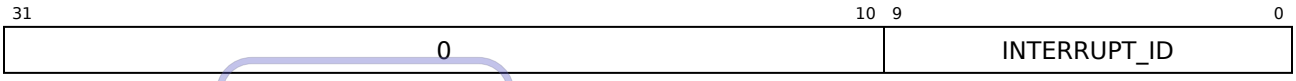
Each interrupt target m ($m \geq 0$) is associated with one Priority Threshold Register. Only active interrupts with priorities strictly greater than the threshold will cause an interrupt notification to be sent to the target.

Field Name	Bits	Description	Type	Reset
THRESHOLD	[31:0]	Interrupt priority threshold. The valid range of this field is determined by the MAX_PRIORITY field of the Version & Maximum Priority Configuration Register.	RW	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

24.5.10 Claim and Complete Register for Target *m*

Offset: 0x200004+4096*m



Official Release

There is one Claim and Complete Register for each interrupt target m ($m \geq 0$). Reading this register claims an interrupt source and returns the ID of that interrupt source.

The interrupt gateway stops processing newer interrupt requests from its interrupt sources until the earlier interrupt request completes. Writing this register with an interrupt ID serves as the interrupt completion message acknowledging to PLIC that the handling of the claimed interrupt has been serviced in target m and the associated interrupt gateway may resume processing newer interrupt requests.

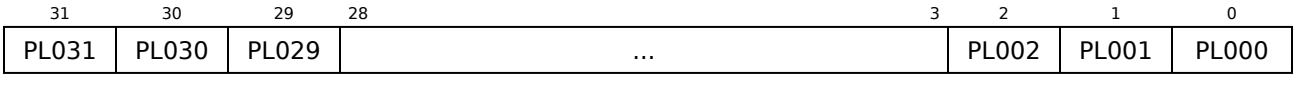
The interrupt gateway only resumes processing of newer interrupt requests if the enable bit of the interrupt source for target m is set. If the enable bit is not set, the interrupt completion message will be ignored.

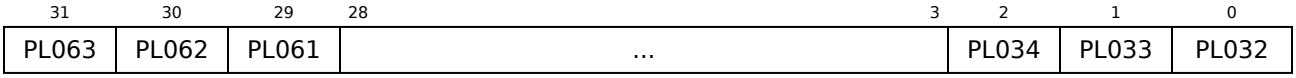
Generally there are no limitations to the order of interrupt claims and completions except when the preemptive priority mode is enabled. When PLIC is in the preemptive priority mode, the latest claimed interrupt should be completed first.

Field Name	Bits	Description	Type	Reset
INTERRUPT_ID	[9:0]	On reads, indicating the interrupt source that has being claimed. On writes, indicating the interrupt source that has been handled (completed).	RW	0

24.5.11 Preempted Priority Stack Registers for Target *m*

Offset: (0x200400+4096*m) to (0x20041F+4096*m)





...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

31	30	29	28 ... 3	2	1	0
PL255	PL254	PL253	...	PL226	PL225	PL224

These registers are read/writable registers for accessing the preempted priority stack for target *m* (*m* $\geq$ 0). These registers are used for saving and restoring priorities of the nested/preempted interrupts for a particular target by hardware. They are made available to software primarily for diagnostic purposes.

Official Release

There are a total of 8 registers, each 32-bit wide, for 255 priority levels. Each bit in these registers indicates if the corresponding priority level has been preempted by a higher-priority interrupt. The location of the priority level bit for priority *p* of target *m* (Word offset Address, Bit Position) can be determined by the following equations:

- Word offset Address: 0x20_0400 + 4096 * *m* + 4 * floor(p/32)
- Bit Position: P modulo 32

Field Name	Bits	Description	Type	Reset
PL_p	[n]	This bit indicates that an interrupt at priority level *p* has been preempted by a higher-priority interrupt. The bit position n = p modulo 32.	RW	0

Value	Meaning
0	No interrupts of priority level *p* are preempted.
1	An interrupt of priority level *p* has been preempted.

24.6 Interrupt Latency

Figure 22 illustrates the minimum timing for the processor to execute the first instruction.

When a device asserts `INT_SRC[n]`, it takes 2 `BUS_CLK` cycles for NCEPLIC100 to arbitrate interrupts and assert its `MEIP` output signal, where `BUS_CLK` is the clock source of NCEPLIC100. When the `MEIP` signal is asserted, it takes one `CORE_CLK` cycle for the processor to latch the signal into the `mip` register. The interrupt is usually taken immediately at the same cycle, unless current operations cannot be interrupted (e.g., accesses to device regions).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

How the processor fetches the trap handler for handling the associated external interrupt depends on its vector interrupt setting. When `mmisc_ctl.VEC_PLIC` is 0, the processor fetches the instruction pointed by `mtvec`. When `mmisc_ctl.VEC_PLIC` is 1 (vector mode), `mtvec` points to a table of entry point addresses, one entry for each external interrupt. It takes the processor 3 additional `CORE_CLK` cycles to fetch the entry point address, before fetching the first instruction of the associated trap handler. The waveform assumes that instruction fetch returns immediately without wait states.

Official Release

The processor implements a 8-stage pipeline, so it takes at least 8 `CORE_CLK` cycles for the first instruction of the trap handler to execute and retire.

In summary, the minimum latency is 2 `BUS_CLK` and 10 `CORE_CLK` cycles. The latency is counted from assertion of `INT_SRC[n]` to the end of execution of the first instruction of the trap handler.

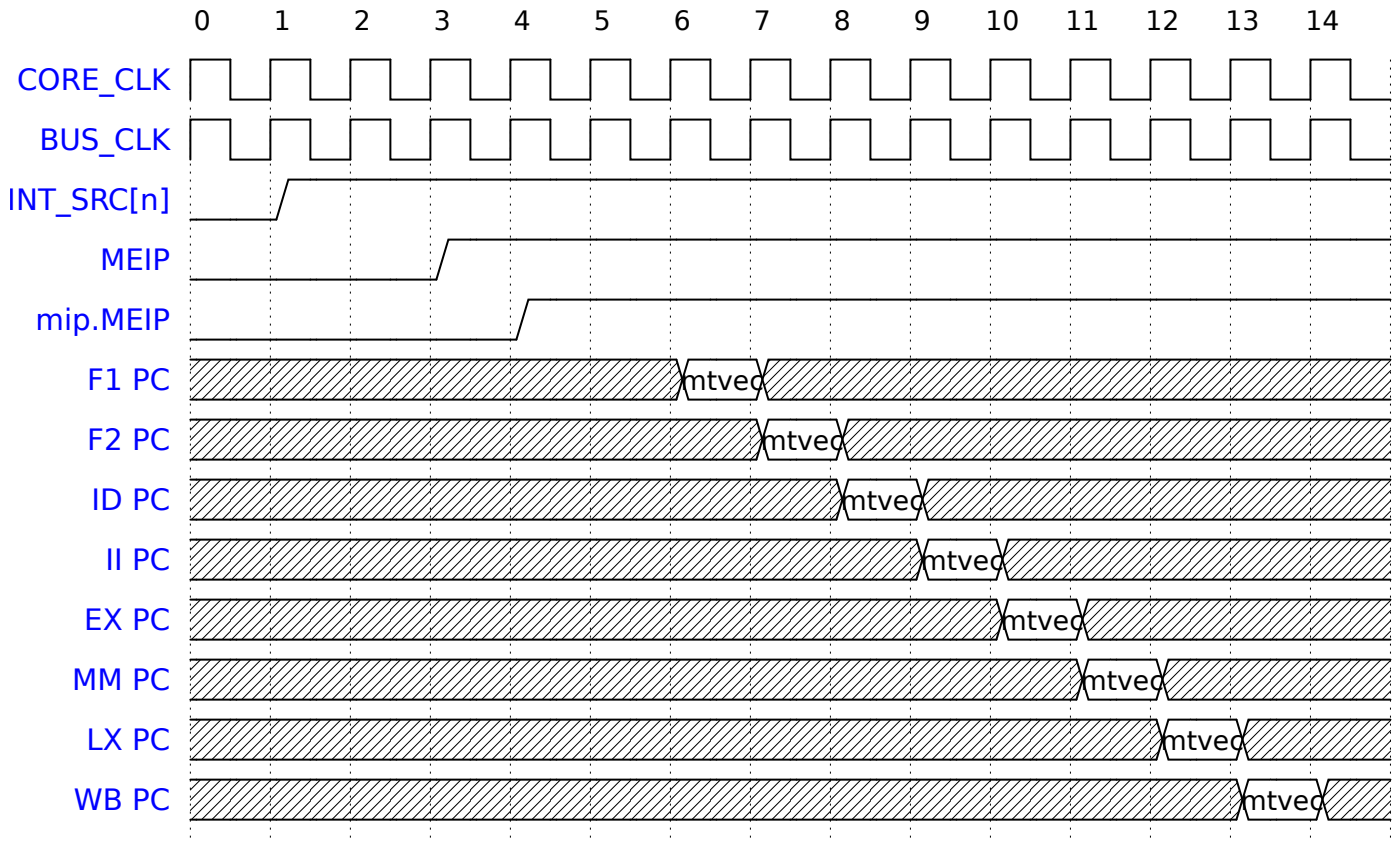


Figure 22: Minimum Interrupt Latency

24.7 Interface Signals

On the AXI interface, `rresp` or `bresp` will also be set to `SLVERR` for invalid transactions.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 190: General Signals of NCEPLIC100

Signal Name	Direction	Description
clk	input	Clock
reset_n	input	Reset (Active-Low)
int_src[INT_NUM:1]	input	Interrupt sources. The sources could be configured to be asynchronous inputs through the ASYNC_INT parameter. See Section 24.4.5 for details.
t*x*_eip	output	External interrupt pending for target *x*.
t*x*_eiid[9:0]	output	External interrupt id for target *x*, see Section 24.3 for details.
t*x*_eiack	input	Interrupt acknowledgment from target *x*, see Section 24.3 for details.

Official Release

Table 191: AXI Interface Signals of NCEPLIC100

Signal Name	Direction	Description
awid[3:0]	input	Write address ID
awaddr[ADDR_WIDTH-1:0]	input	Write address
awlen[7:0]	input	Write burst length
awsize[2:0]	input	Write burst size
awburst[1:0]	input	Write burst type
awlock	input	Write lock type
awcache[3:0]	input	Write cache type
awprot[2:0]	input	Write protection type
awvalid	input	Write address valid
awready	output	Write address ready
wdata[DATA_WIDTH-1:0]	input	Write data
wstrb[(DATA_WIDTH/8-1:0]	input	Write strobes
wlast	input	Write last
wvalid	input	Write valid
wready	output	Write ready
bid[3:0]	output	Write response ID
bresp[1:0]	output	Write response
bvalid	output	Write response valid
bready	input	Write response ready

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 191: (continued)

Signal Name	Direction	Description
arid[3:0]	input	Read address ID
araddr[ADDR_WIDTH-1:0]	input	Read address
arlen[7:0]	input	Read burst length
arsize[2:0]	input	Read burst size
arburst[1:0]	input	Read burst type
arlock	input	Read lock type
arcache[3:0]	input	Read cache type
arprot[2:0]	input	Read protection type
arvalid	input	Read address valid
arready	output	Read address ready
rid[3:0]	output	Read ID tag
rdata[DATA_WIDTH-1:0]	output	Read data
rresp[1:0]	output	Read response
rlast	output	Read last
rvalid	output	Read valid
rready	input	Read ready

Official Release

Table 192: Valid Transactions for NCEPLIC100

Bus	Transaction Type
AXI	Single WORD

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

25 Machine Timer

25.1 Introduction

Official Release

The RISC-V architecture defines a machine timer that provides a real-time counter and generates timer interrupts. NCEPLMT100 is an implementation of the machine timer, and the block diagram is shown in Figure 23. This timer is not to be confused with the real-time clock timer (RTC) of typical computing platforms. Per RISC-V privileged specification, the timer clock (`mtime_clk`) could be clocked at any arbitrary frequency as long as it is a fixed frequency clock that is not affected by clock gating or frequency scaling of clocks in the rest of the platform. On the other hand, RTC is usually clocked by a 32768Hz clock. The Linux kernel expects microsecond resolutions for `mtime`, so it imposes an additional requirement that the frequency of the timer clock has to be greater than 1MHz. For non-Linux applications, the timer clock could share the same clock source as the real-time clock timer.

The RISC-V privileged specification expects that software discovers the frequency of the timer clock through a platform specific mechanism. For Linux kernels, this is achieved through the Device Tree specification.

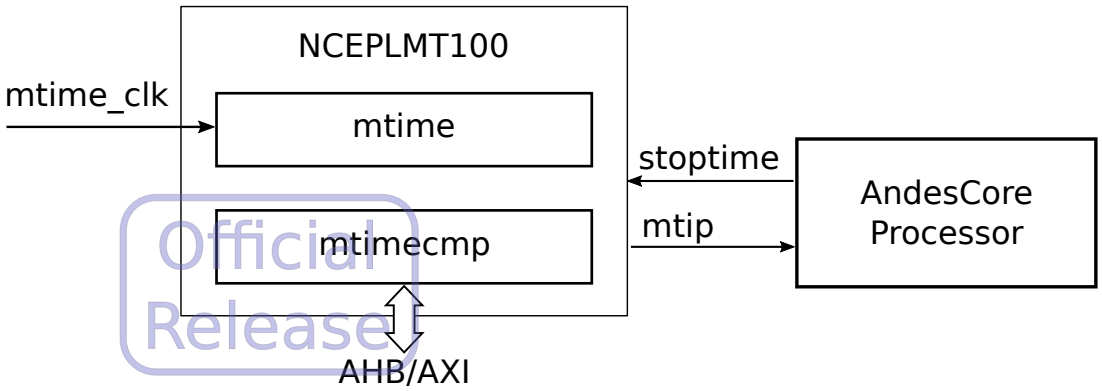
NCEPLMT100 imposes a frequency limitation on the frequency of the `mtime_clk` clock. The `mtime` update synchronization logic requires that the period of the `mtime_clk` should be larger than or equal to that of the bus clock. Please take clock period variations into consideration when evaluating this constraint. Generally speaking, the slowest frequency of the bus interface of NCEPLMT100 should be faster than that of the fastest `mtime_clk` clock. This limitation could be further relaxed through the GRAY_WIDTH configuration parameter. See Section 25.3.6 for details.

On Andes evaluation platforms, the frequency of `mtime_clk` is set to be the same as the regular operating frequency of APB clocks to minimize the number of clock sources in FPGA.

NCEPLMT100 primarily consists of these memory-mapped registers: `mtime` and `mtimecmp`*n* (*n*: 0 – (NHART-1)). The `mtime` register is a 64-bit real-time counter clocked by `mtime_clk`.

The `mtimecmp`*n* register stores a 64-bit value for comparing with `mtime`. When the value in `mtime` is greater than or equal to the value in `mtimecmp`*n*, the `mtip[n]` signal is asserted for generating a timer interrupt. When `mtimecmp`*n* is written, the interrupt is cleared and the `mtip[n]` signal is deasserted.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



Official Release

Figure 23: NCEPLMT100 Block Diagram

25.2 Machine Timer Registers

NCEPLMT100 registers are accessed through the bus interface, and their memory map is shown in Table 193.

Please note that NCEPLMT100 supports only 32-bit or 64-bit transfers. Behaviors of 8-bit and 16-bit transfers are UNDEFINED, and these transfers might be ignored as well as result in error responses or unexpected register updates.

Table 193: AX45MP NCEPLMT100 Memory Map

Address Offset	Description
0x0 – 0x3	mtime[31:0]
0x4 – 0x7	mtime[63:32]
0x8 – 0xB	mtimecmp0[31:0]
0xC – 0xF	mtimecmp0[63:32]
0x10 – 0x13	mtimecmp1[31:0]
0x14 – 0x17	mtimecmp1[63:32]
0x18 – 0x1B	mtimecmp2[31:0]
0x1C – 0x1F	mtimecmp2[63:32]
0x8+*n**8 – 0xB+*n**8	mtimecmp*n*[31:0]
0xC+*n**8 – 0xF+*n**8	mtimecmp*n*[63:32]

The `mtime` register is driven by `mtime_clk`, which is assumed to be slower than the clock of the bus interface. The `mtime_shadow` shadow register is maintained in the bus clock domain to reduce the latency of accessing the `mtime` register in the slow clock domain. The values of `mtime` and `mtime_shadow` registers are constantly synchronized such that `mtime_shadow` maintains the most

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

up-to-date values of mtime. The value in mtime_shadow is instantly returned when reading the mtime register. When writing the mtime register, bus write transactions finish when the values are written to the mtime_shadow register, and NCEPLMT100 handles the synchronization to mtime in the background.

Official Release

25.2.1 Machine Timer Initialization

The mtime counter is a 64-bit value and it increments non-stop on every machine timer clock except the first few cycles after its control register updates. When the data bus of NCEPLMT100 is 64-bit, it is recommended to program the mtime counter through a single double-word store to guarantee the value is updated atomically. Otherwise, the following programming sequence should be followed to initialize the counter through two separate 32-bit updates to mtime[31:0] and mtime[63:32].

- If bit[31:29] of the intended value is 7:
  1. Write *zero* to mtime[31:0].
  2. Write high part of the intended value to mtime[63:32].
  3. Write low part of the intended value to mtime[31:0].
- Otherwise:
  1. Write low part of the intended value to mtime[31:0].
  2. Write high part of the intended value to mtime[63:32].

25.3 Machine Timer Configuration Options

Table 194 summarizes all supported configuration parameters and the subsections describe the parameters in detail.

Table 194: NCEPLMT100 Configuration Parameters

Parameter Name	Type	Valid Values	Default Value
ADDR_WIDTH	Integer	≥10	32
DATA_WIDTH	Integer	32 or 64	32
BUS_TYPE	String	ahb/axi	ahb
ID_WIDTH	Integer	≥1	4
NHART	Integer	1–32	4
GRAY_WIDTH	Integer	2–31	2
SYNC_STAGE	Integer	2 or 3	2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

25.3.1 Address Width

ADDR_WIDTH configures the address width of Machine Timer bus. The address width should be greater than or equal to 10 to encompass all addressable Machine timer memory space.

Official Release

25.3.2 Data Width

DATA_WIDTH configures the data width of Machine Timer bus. It is either 32-bit or 64-bit.

25.3.3 Number of Supported Harts

NHART configures the number of supported harts. This essentially configures the number of timer comparison registers `mtimecmp`*N* as well as the width of `mtip` interface, where $N = 0$ – (NHART-1).

25.3.4 Bus Type

BUS_TYPE configures the bus type of Machine Timer.

- String “ahb” indicates Machine Timer is an AHB slave device.
- String “axi” indicates Machine Timer is an AXI4 slave device.

25.3.5 AXI ID Width

ID_WIDTH configures the width of the AXI ID signals. This parameter only takes effect when BUS_TYPE is “axi”.

25.3.6 Range of Clock Ratio Between MTIME_CLK and Bus Clock and MTIME_CLK

GRAY_WIDTH configures how slow the bus clock could be with respect to `mtime_clk`. The default setting of 2 is designed to work well when the bus clock is faster than `mtime_clk`. If the frequency of the bus clock may be slowed down to below that of `mtime_clk`, this parameter should be adjusted for proper synchronization between the two clock domains.

The exact requirement is $\text{period}_{\text{clk}} \leq \text{period}_{\text{mtime\_clk}} * 2^{\text{GRAY\_WIDTH-2}}$; or in terms of frequency, see the table below. Please be reminded that clock jitters should be taking into consideration when determining GRAY_WIDTH values. Larger GRAY_WIDTH values only impose slightly larger overhead in the critical path of NCEPLMT100 in the Gray-to-Binary conversion circuit. So if unsure about the amount of clock jitters, it is always prudent to select one value larger than the expected clock ratio.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 195: Supported Clock Ratio (Frequency) Configuration

GRAY_WIDTH	Supported Clock Ratio (f_{clk} : f_{mtime_clk})
2	$\geq$ 1 : 1
3	$\geq$ 1 : 2
4	$\geq$ 1 : 4
5	$\geq$ 1 : 8
6	$\geq$ 1 : 16
7	$\geq$ 1 : 32
8	$\geq$ 1 : 64
9	$\geq$ 1 : 128
W	$\geq$ 1 : 2^{W-2}

Official Release

25.3.7 Synchronizer Level

SYNC_STAGE configures the level of the CDC synchronizer.

25.4 Interface Signals

NCEPLMT100 offers two types of bus interfaces: AHB and AXI interfaces. The interfaces are selected by the `PLMT_BUS` parameter. The interface signals to both interfaces are simultaneously present on its module port list, and only the selected one will be used. The other group of signals will be unused and left floating.

The tables below describe the interface signals of NCEPLMT100, and the clock to NCEPLMT100 should be synchronous to that of AX45MP.

The valid transactions that NCEPLMT100 accepts are summarized in Table 199.

It responds to invalid transactions by returning undefined values for read transactions and ignoring writes. On the AXI interface, non-zero `rresp` or `bresp` is also set to `SLVERR` for invalid transactions.

Table 196: General Signals of NCEPLMT100

Signal Name	Direction	Description
clk	input	Clock for the bus interface
resetn	input	Reset for the bus interface (Active-Low)
mtime_clk	input	Clock for the `mtime` counter. See Section 25.1

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 196: (continued)

Signal Name	Direction	Description
por_rstn	input	Power on reset (Active-Low) for initializing the `mtime` counter
stoptime	input	Stop counting the `mtime` counter
mtip[NHART-1:0]	output	Timer interrupt pending

Official Release

Table 197: AHB Interface Signals of NCEPLMT100

Signal Name	Direction	Description
hsel	input	Slave select
hrdata[DATA_WIDTH-1:0]	output	Read data bus
hready	input	Transfer done signal of the AHB bus
hreadyout	output	Transfer done signal of Machine Timer
hresp[1:0]	output	Transfer response
haddr[ADDR_WIDTH-1:0]	input	Address bus
hburst[2:0]	input	Burst type
hprot[3:0]	input	Protection control
hsize[2:0]	input	Transfer size
htrans[1:0]	input	Transfer type
hwdata[DATA_WIDTH-1:0]	input	Write data bus
hwrite	input	Transfer direction

Table 198: AXI Interface Signals of NCEPLMT100

Signal Name	Direction	Description
awid[3:0]	input	Write address ID
awaddr[ADDR_WIDTH-1:0]	input	Write address
awlen[7:0]	input	Write burst length
awsize[2:0]	input	Write burst size
awburst[1:0]	input	Write burst type
awlock	input	Write lock type
awcache[3:0]	input	Write cache type
awprot[2:0]	input	Write protection type
awvalid	input	Write address valid
awready	output	Write address ready

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 198: (continued)

Signal Name	Direction	Description
wdata[DATA_WIDTH-1:0]	input	Write data
wstrb[(DATA_WIDTH/8)-1:0]	input	Write strobes
wlast	input	Write last
wvalid	input	Write valid
wready	output	Write ready
bid[3:0]	output	Write response ID
bresp[1:0]	output	Write response
bvalid	output	Write response valid
bready	input	Write response ready
arid[3:0]	input	Read address ID
araddr[ADDR_WIDTH-1:0]	input	Read address
arlen[7:0]	input	Read burst length
arsize[2:0]	input	Read burst size
arburst[1:0]	input	Read burst type
arlock	input	Read lock type
arcache[3:0]	input	Read cache type
arprot[2:0]	input	Read protection type
arvalid	input	Read address valid
arready	output	Read address ready
rid[3:0]	output	Read ID tag
rdata[DATA_WIDTH-1:0]	output	Read data
rresp[1:0]	output	Read response
rlast	output	Read last
rvalid	output	Read valid
rready	input	Read ready

Official Release

Table 199: Valid Transactions for NCEPLMT100

Bus	Configuration	Transaction Type
AXI	DATA_WIDTH == 32	Single WORD
AXI	DATA_WIDTH == 64	Single WORD, Double WORD

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26 Debug Subsystem

26.1 Overview

Official Release

The AX45MP debug subsystem implements *RISC-V External Debug Support (TD003) V0.13*. Figure 24 shows the block diagram of the debug subsystem, which contains two components: NCEPLDM200 and NCEJDTM200. NCEPLDM200 is the debug module, which could be accessed through its two AHB slave ports. One is the system interface port, which is for an AndesCore processor to access the debug module through the system bus. The other one is the Debug Memory Interface (DMI) port, which is accessed by NCEJDTM200 (JTAG Debug Transport Module). NCEJDTM200 converts debug commands in JTAG interfaces of external debuggers to bus read/write requests to the DMI port.

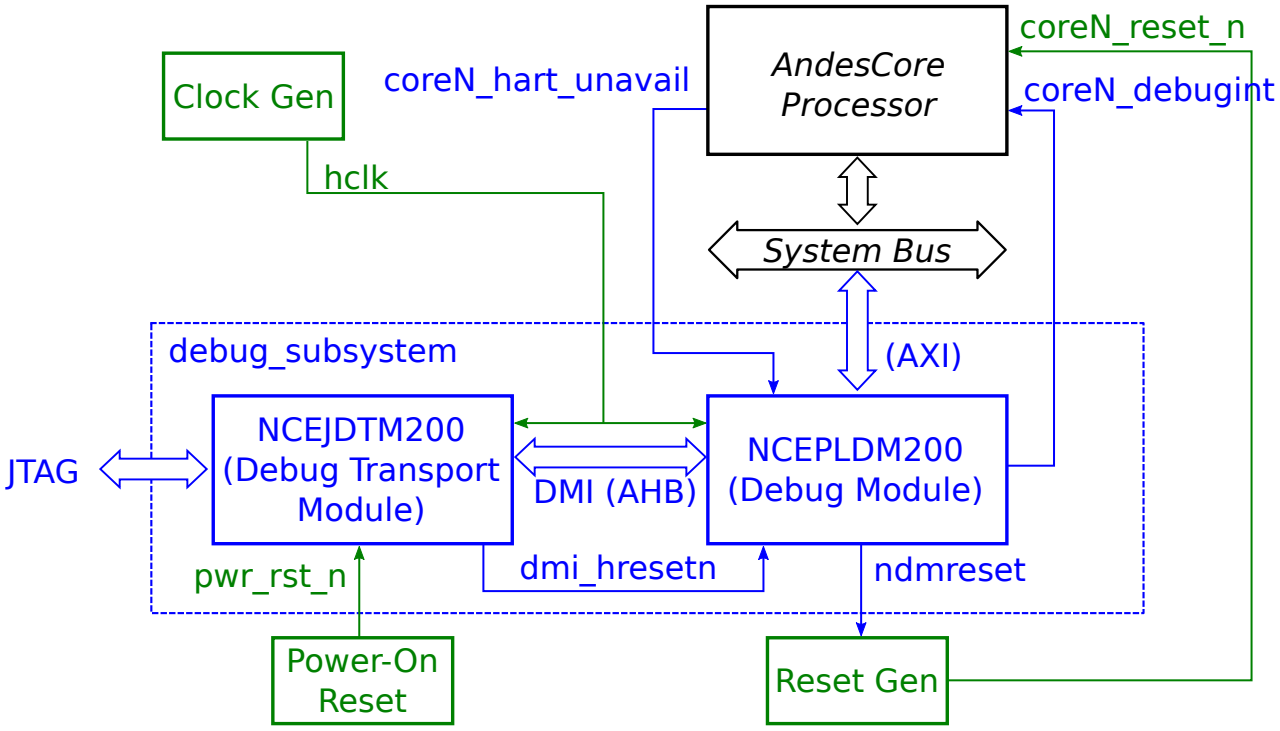


Figure 24: Debug Subsystem Block Diagram

Debug interrupts cause AX45MP to enter the debug mode and redirect the instruction fetch to the debug exception handler, whose entry point should be the base address of NCEPLDM200 and defined by the Debug Module Base option of AX45MP.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

NCEPLDM200 includes a Debug ROM at its base address that defines the debug exception handler. When invoked, the debug handler polls NCEPLDM200 internal registers to process commands issued by the external debugger through the NCEJDTM200 module. Typical debug commands include accessing processor registers, accessing memories, and executing programs written in the Program Buffer, which is a memory region writable by the external debugger.

Official Release

Correct operations of the debug subsystem require clocks, resets and I/O interfaces to be connected in a certain way, as well as making sure Debug Module Base is within a device region (Section 2.12). Please see the next subsection for information.

26.2 Integration Requirements

For proper operations of the debug subsystem, the following platform-level requirements of reset, clock and I/O signals should be met:

- NCEJDTM200 should be reset by power-on resets, which will not be triggered by other resets in the system.
- NCEPLDM200 should be reset by `dmi_hresetn` that is generated by NCEJDTM200.
- `ndmreset` (non-debug-module reset) should connect to the platform reset generator, which triggers platform-wide resets. Through the reset generator, `ndmreset` should be able to control `coreN_reset_n` of AndesCore processors. As the name suggests, `ndmreset` should reset neither NCEJDTM200 nor NCEPLDM200. Please also note that the platform reset should not affect the pin-muxing of the external debug interface (JTAG) pins to preserve connectivity to the external debugger throughout resets.
- `ndmreset` should not cause assertion of power-on resets (`pwr_rst_n`).
- The clock signal (`clk`) to NCEJDTM200 and NCEPLDM200 should keep running during the assertion of `ndmreset`. Stopped clocks will impede the deassertion of `ndmreset`.
- The bus interfaces of NCEJDTM200 and NCEPLDM200 should be in the same clock domain.
- Both `tck` and `tms` pins of the JTAG interface should not be floating when the external debugger is not attached:
  - `tck` can be either pulled up to HIGH or pulled down to LOW.
  - `tms` should be pulled up to HIGH.

26.3 Optional Debug Subsystem

By default, the debug subsystem is embedded in the platform.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

For the AE350 platform, the debug subsystem follows the core debug setting (Section 2.11.1). The debug subsystem can be removed by simply disabling the core debug feature in the configuration tool.

26.4 Debug Subsystem Configuration Options

Official Release

Table 200: Debug Subsystem Configuration Parameters

Parameter Name	Type	Valid Values	Default Value
NHART	Integer	1–1024	1
SYS_ADDR_WIDTH	Integer	9–64	32
SYS_DATA_WIDTH	Integer	32, 64, or 128	64
ADDR_WIDTH	Integer	9–64	32
DATA_WIDTH	Integer	32 or 64	64
SYS_BUS_ACCESS	String	“yes” or “no”	yes*
DEBUG_INTERFACE	String	“jtag” or “serial”	jtag*
PROGBUF_SIZE	Integer	1–8	8*
SYNC_STAGE	Integer	2/3	2

Note

- These options should be configured in configuration tool, see Section 2.18.

26.4.1 Number of Harts

`NHART` determines how many harts can be connected to a NCEPLDM200 module. The maximum value is 1024.

26.4.2 Bus Slave Configurations

`ADDR_WIDTH` and `DATA_WIDTH` determine the address width and the data width of the bus slave interface. This interface is used for debug flow control and data exchange with all the connected harts. All the ports on this interface use “rv” as prefix.

26.4.3 System Bus Access

`SYS_BUS_ACCESS` determines whether the optional feature, system bus access is supported. With system bus access support, PLDM can access the memory without involving any hart.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.4.4 System Bus Master Configurations

SYS_ADDR_WIDTH and SYS_DATA_WIDTH determine the address width and the data width of the bus master. This interface is only used for system bus access feature. All the ports on this interface use “sys” as prefix.

Official Release

26.4.5 Debug Interface

DEBUG_INTERFACE determines the interface between debug subsystem and the external device for debug.

- String “jtag” implies the standard 4-wire JTAG interface.
- String “serial” implies Andes 2-wire serial debug interface.

26.4.6 Program Buffer Size

PROGBUF_SIZE is used to configure the size of program buffer, in 32-bit words.

26.4.7 Synchronizer Level

SYNC_STAGE configures the level of the CDC synchronizer.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5 NCEPLDM200

Table 201 summarizes the memory map within the NCEPLDM200 address space as viewed from the system bus interface. Please note that the offset for the program buffer could be discovered by the external debugger through execution of the `AUIPC` instruction as the first instruction in the program buffer, and the starting offset of Abstract Data is defined as hartinfo.DATAADDR. The offsets could be used as offsets of load/store instructions with the `zero` register as the base register to access this memory space. The `zero` register is automatically mapped to the base of NCEPLDM200 for load/store instructions in Debug Mode.

Official Release

This system bus address space of NCEPLDM200 is defined through the Debug Module Base option. It should be within a device region for the proper operation of the Debug ROM and the external debugger support. Please see Section 2.12 for information on how to set up this address space as device regions.

Table 202 summarizes the memory map as viewed from the DMI interface. The address[8:2] value in the table follows the address value assignment of the debug module debug bus registers as described in *RISC-V External Debug Support (TD003) V0.13*.

Table 201: System Memory Map of NCEPLDM200

Address Offset	Description	Definition
0x0000 – 0x007F	Debug ROM	Internal Use Only
0x0080 – 0x00BF	Program Buffer	Section 26.5.12
0x00C0 – 0x00CF	Abstract Data 0–3	Section 26.5.1
0x00D0 – 0x01FF	Reserved for internal use	N/A

Table 202: DMI Memory Map of NCEPLDM200

Address[8:2]	Description	Definition
0x04 – 0x07	Abstract Data 0–3	Section 26.5.1
0x10	Debug Module Control	Section 26.5.2
0x11	Debug Module Status	Section 26.5.3
0x12	Hart Info	Section 26.5.4
0x13	Halt Summary	Section 26.5.5
0x14	Hart Array Window Select	Section 26.5.6
0x15	Hart Array Window	Section 26.5.7
0x16	Abstract Control and Status	Section 26.5.8
0x17	Abstract Command	Section 26.5.9
0x18	Abstract Command Autoexec	Section 26.5.10

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 202: (continued)

Address[8:2]	Description	Definition
0x19 – 0x1C	Device Tree Addr 0–3	Section 26.5.11
0x20 – 0x2F	Program Buffer 0–15	Section 26.5.12
0x30	Authentication Data	Section 26.5.13
0x38	System Bus Access Control and Status	Section 26.5.14
0x39 – 0x3B	System Bus Address	Section 26.5.15
0x3C – 0x3F	System Bus Data	Section 26.5.16

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

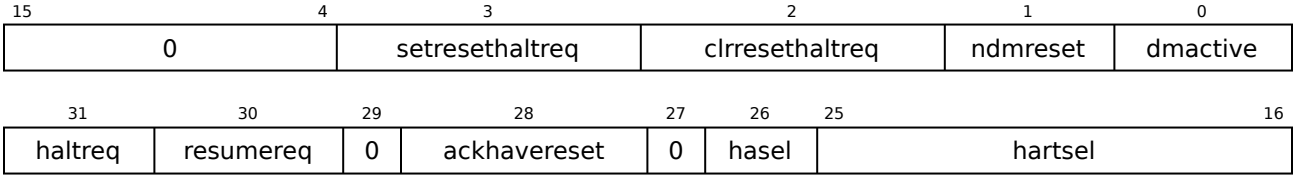
26.5.1 Abstract Data 0–3 (`data0` – `data3`)

Basic read/write registers that may be read or changed by abstract commands.

The registers are accessible from both the DMI interface and the system bus interface to support data exchanges between the external debugger and the processor (i.e., instructions in the program buffer).

Official Release

26.5.2 Debug Module Control (`dmcontrol`)



Field Name	Bits	Description	Type	Reset
dmactive	[0]	Controlling reset signal for Debug Module itself. **Value** / **Meaning** 0 — The Debug Module's state takes its reset values. 1 — The Debug Module functions normally.	RW	0x0
ndmreset	[1]	Controlling reset signal from the Debug Module to the rest of the system. **Value** / **Meaning** 0 — Deassert system reset signal. 1 — Assert system reset signal.	RW	0x0
clrresethaltreq	[2]	This optional field clears the halt-on-reset request bit for all currently selected harts. Writes apply to the new value of `hartsel` and `hasel`.	W1	-

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
setresethaltreq	[3]	This optional field writes the halt-on-reset request bit for all currently selected harts, unless `clrresethaltreq` is simultaneously set to 1. When set to 1, each selected hart will halt upon the next deassertion of its reset. The halt-on-reset request bit is not automatically cleared. The debugger must write to `clrresethaltreq` to clear it. Writes apply to the new value of `hartsel` and `hasel`. If `hasresethaltreq` is 0, this field is not implemented.	W1	-
hartsel	[25:16]	Selecting the target hart to be debugged.	RW	0x0
hasel	[26]	Selects the definition of currently selected harts. **Value** / **Meaning** 0 — There is a single currently selected hart, that is selected by `hartsel`. 1 — There may be multiple currently selected harts selected by `hartsel`, plus those selected by the hart array mask register.	RW	0x0
ackhavereset	[28]	Writing 1 to this bit clears the havereset bits for any selected harts. Harts are selected based on the new value of `hartsel` and `hasel` being written.	W1	0x0
resumereq	[30]	Writes the resume request bit for all currently selected harts. When set to 1, each selected hart will resume if it is currently halted. The resume request bit is ignored while the halt request bit is set. Harts are selected based on the new value of `hartsel` and `hasel` being written.	WO	0x0

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
haltreq	[31]	Writes the halt request bit for all currently selected harts. When set to 1, each selected hart will halt if it is not currently halted. Writing 1 or 0 has no effect on a hart which is already halted, but the bit must be cleared to 0 before the hart is resumed. Harts are selected based on the new value of `hartsel` and `hasel` being written.	WO	0x0

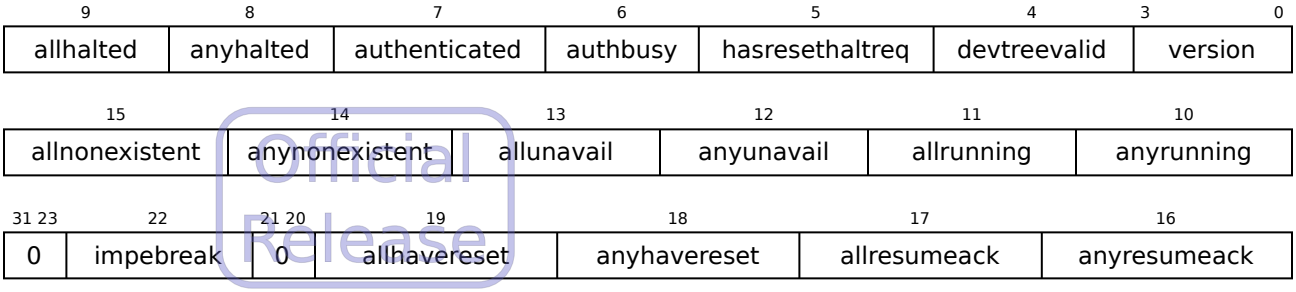
Official Release

Note

- NCEPLDM200 uses wraparound hart ID to select harts, which can reduce the integration effort on a multi-hart system with multiple debug modules. When a hart ID is larger than the effective bit value, it will be mapped to a smaller value. For example, hart 10 will be remapped to hart 2 on a debug module instance supporting 8 harts (NHART=8). Essentially, the effective bits of hart ID used in NCEPLDM200 is the ceiling of log2 value of `NHART`.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5.3 Debug Module Status (`dmstatus`)



Official Release

Field Name	Bits	Description	Type	Reset
version	[3:0]	Version of the implemented RISC-V External Debug Support. 0x2 indicates that the current implemented version is 0.13.	RO	0x2
devtreevalid	[4]	Whether the information in devtreeaddr0 – devtreeaddr3 registers holds the address of the Device Tree.	RO	0x0
hasresethaltreq	[5]	This bit indicates the supported status of `resethaltreq`.	RO	0x1
authbusy	[6]		RO	0x0
authenticated	[7]		RO	0x1
anyhalted	[8]	Indicates whether any currently selected hart is halted.	RO	0x0
allhalted	[9]	Indicates whether all currently selected harts are halted.	RO	0x0

Value	Meaning
0	The authentication module is ready to process the next read/write to *authdata*.
1	The authentication module is busy.

Value	Meaning
0	Authentication is required before using the Debug Module.
1	Authentication check has passed.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
anyrunning	[10]	Indicates whether any currently selected hart is running.	RO	0x0
allrunning	[11]	Indicates whether all currently selected harts are running.	RO	0x0
anyunavail	[12]	Indicates whether any currently selected hart is unavailable.	RO	0x0
allunavail	[13]	Indicates whether all currently selected harts are unavailable.	RO	0x0
anynonexistent	[14]	Indicates whether any currently selected hart does not exist in this system.	RO	0x0
allnonexistent	[15]	Indicates whether all currently selected harts do not exist in this system.	RO	0x0
anyresumeack	[16]	Indicates whether any currently selected hart has acknowledged the previous resume request.	RO	0x0
allresumeack	[17]	Indicates whether all currently selected harts have acknowledged the previous resume request.	RO	0x0
anyhavereset	[18]	Indicates whether any currently selected hart has been reset but the reset has not been acknowledged	RO	0x0
allhavereset	[19]	Indicates whether all currently selected harts have been reset but the reset has not been acknowledged	RO	0x0
impebreak	[22]	Indicates whether there is an implicit `EBREAK` instruction at the nonexistent word immediately after the program buffer.	RO	0x1

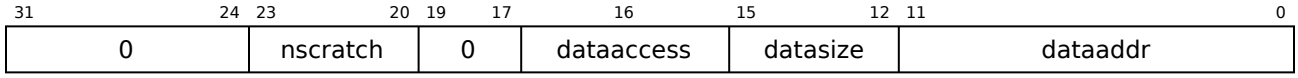
Official Release

Note

- The reset values with * mark may not reflect the transient hart states when NCEPLDM200 is under reset or `dmcontrol.DMACTIVE` is not set.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5.4 Hart Info (`hartinfo`)



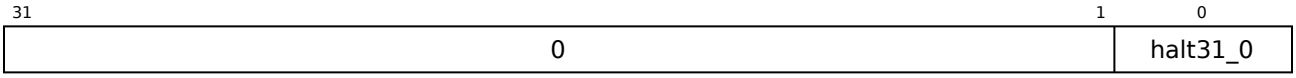
Official Release

Field Name	Bits	Description	Type	Reset
dataaddr	[11:0]	Signed offset for accessing the shadowed *data* registers by the processor, to be used as offsets for load/store instructions with the `zero` register as the base register in Debug Mode.	RO	0xC0
datasize	[15:12]	Number of 32-bit words in the memory map dedicated to shadowing the data registers.	RO	0x4
dataaccess	[16]	The method for accessing the shadowed *data* registers. The value of this field is 0x1 for AX45MP, indicating that the *data* registers are shadowed in the memory map of the selected hart under Debug Mode.	RO	0x1
nscratch	[23:20]	Number of dscratch registers available for the debugger to use during program buffer execution, starting from `dscratch0`.	RO	0x2

Value	Meaning
0	The *data* registers are shadowed in the hart by CSR registers.
1	The *data* registers are shadowed in the hart's memory map. Each register takes up 4 bytes in the memory map.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

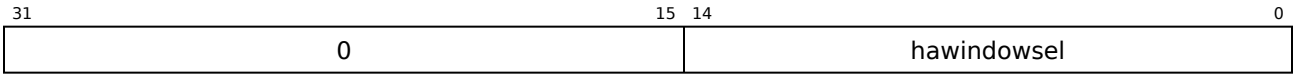
26.5.5 Halt Summary (haltsum)



Bit 0 contains the logical OR of 32 halt bits for hart 0 – hart 31.

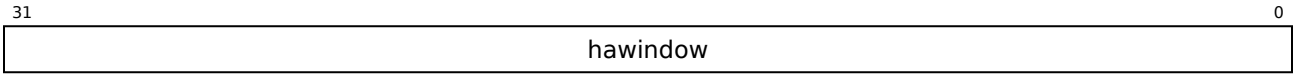
Official Release

26.5.6 Hart Array Window Select (hawindowsel)



Field Name	Bits	Description	Type	Reset
hawindowsel	[14:0]	This register selects which of the 32-bit portion of the hart array mask register is accessible in hawindow.	RW	0x0

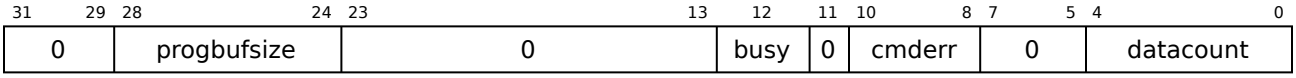
26.5.7 Hart Array Window (hawindow)



Field Name	Bits	Description	Type	Reset
hawindow	[31:0]	This register provides R/W accesses to a 32-bit portion of the hart array mask register. The position of the window is determined by `hawindowsel`. That is, bit 0 refers to hart (`hawindowsel` * 32), while bit 31 refers to hart (`hawindowsel` * 32 + 31).	RW	0x0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5.8 Abstract Control and Status (abstractcs)

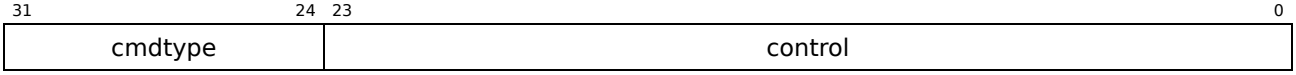


Official Release

Field Name	Bits	Description	Type	Reset
datacount	[4:0]	Number of *data* registers that are implemented.	RO	0x4
cmderr	[10:8]	Error code indicating that an abstract command fails. The bits in this field remain set until they are cleared by writing 1 to them. No abstract command is started until the value is reset to 0.	R/W1C	0x0
busy	[12]	Flag indicating an abstract command is currently being executed.	RO	0x0
progbufsize	[28:24]	Size of the program buffer, in 32-bit words.	RO	0x8

Value	Meaning
0	None: no error
1	Busy: an abstract command was executing while command, abstractcs, or abstractauto was written, or when one of the data or progbuf registers was read or written.
2	Not supported: the requested command is not supported.
3	Exception: an exception occurred while executing the command.
4	Halt/resume: an abstract command cannot execute because the hart wasn't in the expected state (running/halted).

26.5.9 Abstract Command (command)



The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
control	[23:0]	The field is interpreted in a command-specific manner.	WO	0x0
cmdtype	[31:24]	Controlling the overall functionality of this abstract command.	WO	0x0

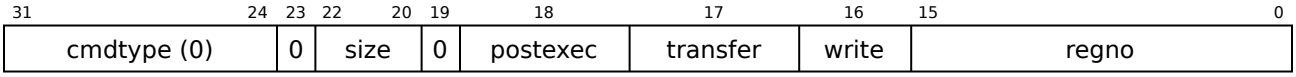
Official Release

If a command takes arguments or returns values, they must be written to or placed in the data registers. Which data registers are used for the arguments is described in Table 203.

Table 203: Use of Data Registers in PLDM

Argument Width	arg0/Return Value	arg1	arg2
32	data0	data1	data2
64	data0, data1	data2, data3	data4, data5

26.5.9.1 Access Register



This command gives the debugger access to CPU registers and allows CPU to execute the program buffer.

Field Name	Bits	Description	Type	Reset
regno	[15:0]	Number of the specified register to be accessed.	WO	0x0
write	[16]	The direction of data transfer.	WO	0x0

Value	Meaning
0	Copy data from the specified register into *arg0* portion of the Abstract Data registers.
1	Copy data from *arg0* portion of Abstract Data registers into the specified register.

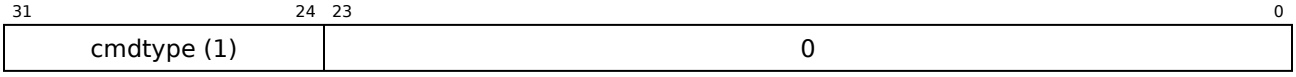
Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Field Name	Bits	Description	Type	Reset
transfer	[17]	Indicates whether to perform data transfer. Value: 0 — Don't do the operation specified by write. Value: 1 — Do the operation specified by write.	WO	0x0
postexec	[18]	Indicates whether to execute the program in the program buffer. If this field is set, execute the program in the Program Buffer exactly once after performing the transfer, if any.	WO	0x0
size	[22:20]	Value: 2 — Access the lowest 32 bits of the register Value: 3 — Access the lowest 64 bits of the register.	WO	0x0

Official Release

26.5.9.2 Quick Access

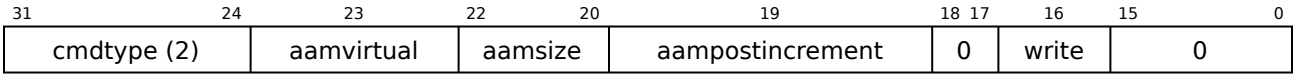


Perform the following sequence of operations:

- If the hart is halted already, the command sets cmderr to *halt/resume* and does not continue.
- Halt the hart. If the hart halts for some other reason (e.g., breakpoint), the command sets cmderr to *halt/resume* and does not continue.
- Execute the program buffer. If an exception occurs, cmderr is set to exception and the program buffer execution ends, but the quick access command continues.
- Resume the hart.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5.9.3 Access Memory



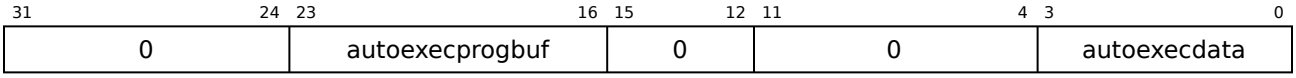
This command lets the debugger perform memory accesses with the memory view of the selected hart.

Official Release

Field Name	Bits	Description	Type	Reset
write	[16]	The direction of data transfer. **Value** / **Meaning** 0: Copy data from the memory location specified in *arg1* into *arg0* portion of data. 1: Copy data from *arg0* portion of data into the memory location specified in *arg1*.	WO	0x0
aampostincrement	[19]	Increment *arg1* by the number of bytes encoded in `aamsize` after a memory access completes.	WO	0x0
aamsize	[22:20]	Size of memory accesses. **Value** / **Meaning** 0: Access the lowest 8 bits of the memory location. 1: Access the lowest 16 bits of the memory location. 2: Access the lowest 32 bits of the memory location. 3: Access the lowest 64 bits of the memory location.	WO	0x0
aamvirtual	[23]	Virtual or physical address accesses **Value** / **Meaning** 0: Addresses are physical 1: No action	WO	0x0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5.10 Abstract Command Autoexec (abstractauto)



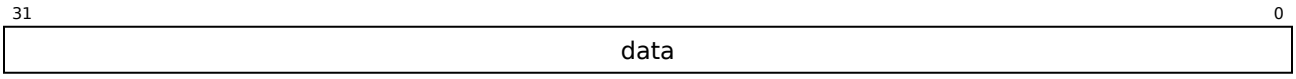
Official Release

Field Name	Bits	Description	Type	Reset
autoexecdata	[3:0]	When a bit in this field is 1, read or write accesses to the corresponding data word cause the command in command to be executed again.	RW	0x0
autoexecprogbuf	[23:16]	When a bit in this field is 1, read or write accesses to the corresponding progbuf word cause the command in command to be executed again.	RW	0x0

26.5.11 Device Tree Addr 0–3 (devtreeaddr0 – devtreeaddr3)

The *devicetreeaddr* registers are hardwired to zeros in NCEPLDM200.

26.5.12 Program Buffer 0–15 (progbuf0 – progbuf15)



The *progbuf* registers provide read/write accesses to the program buffer. NCEPLDM200 supports program buffer 0–7 only. Program buffer 8–15 are hardwired to 0x00100073 (the EBREAK instruction).

These registers are read/write accessible from both the DMI interface and the system bus, in addition to being a valid region for instruction fetches. They hold small programs written by the external debugger and these small programs will be fetched and executed by the processor upon execution of the abstract commands which require execution of the program buffer.

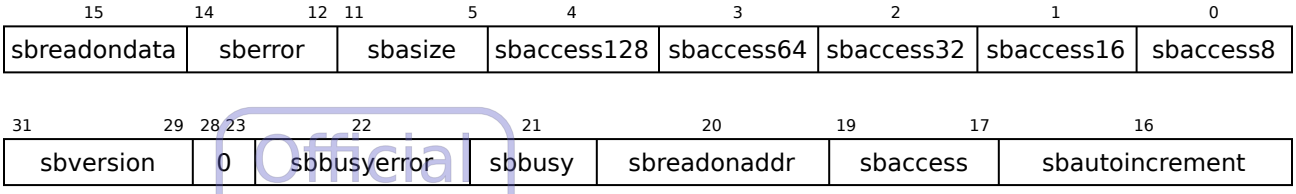
Programs in *progbuf* must end with EBREAK or C.EBREAK instructions if the program size is less than 8 words (32 bytes).

26.5.13 Authentication Data (authdata)

The *authdata* register is hardwired to zeros in NCEPLDM200.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.5.14 System Bus Access Control and Status (sbcs)



Official Release

The *sbcs* register holds the control bits and status flags of System Bus Accesses. It is only valid when the SYSTEM_BUS_ACCESS_SUPPORT parameter of NCEPLDM200 is set. It is otherwise hardwired to zero.

Field Name	Bits	Description	Type	Reset
sbaccess8	[0]	This bit indicates the supported status of 8-bit system bus data accesses.	RO	0x1
sbaccess16	[1]	This bit indicates the supported status of 16-bit system bus data accesses.	RO	0x1
sbaccess32	[2]	This bit indicates the supported status of 32-bit system bus data accesses.	RO	0x1
sbaccess64	[3]	This bit indicates the supported status of 64-bit system bus data accesses. This bit is 1 if SYS_DATA_WIDTH == 64.	RO	Configuration Dependent
sbaccess128	[4]	This bit indicates the supported status of 128-bit system bus data accesses.	RO	0x0
sbasize	[11:5]	Address width of System bus addresses in bits. This field is 0 if there is no bus access support. Otherwise, it is the value of the SYS_ADDR_WIDTH parameter of NCEPLDM200.	RO	Configuration Dependent

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Official Release

Field Name	Bits	Description	Type	Reset
sberror	[14:12]	Error code indicating the failure type of the system bus accesses. Write 1 to clear the status. **Value** / **Meaning** 0 — No bus error 1 — Timeout error 2 — Bad address 3 — Alignment error 4 — Unsupported size was requested 7 — Others	R/W1C	0x0
sbreadondata	[15]	When 1, Every read from *sbdata0* automatically triggers a system bus read at the (possibly auto-incremented) address.	RW	0x0
sbautoincrement	[16]	*sbaddress* is incremented by the size specified in *sbaccess_* after every system bus access.	RW	0x0
sbaccess	[19:17]	Access size. **Value** / **Meaning** 0 — 8-bit 1 — 16-bit 2 — 32-bit 3 — 64-bit 4 — 128-bit	RW	0x2
sbreadonaddr	[20]	When 1, Every write to *sbaddress0* automatically triggers a system bus read at the new address.	RW	0x0
sbbusy	[21]	Indicate the system bus master is busy.	RO	0x0
sbbusyerror	[22]	Indicate the debugger attempts to execute system bus access before *sbbusy* is cleared.	R/W1C	0x0
sbversion	[31:29]	The version of the supported System Bus Access version. It is currently 1 for v0.13 of the RISC-V External Debug Support specification.	RO	0x1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

The states of *sbaccess8* – *sbaccess128* are set based on the SYS_ADDR_WIDTH parameter in NCE-PLDM200.

Official Release

26.5.15 System Bus Address (sbaddress0 – sbaddress2)

The *sbaddress* registers are only valid when the SYSTEM_BUS_ACCESS_SUPPORT parameter of NCE-PLDM200 is set. They are otherwise hardwired to zeros.

Table 204: System Bus Address Register

Address Register	Description
sbaddress0	bit[31:0] of the address in *sbaddress*
sbaddress1	bit[63:32] of the address in *sbaddress*
sbaddress2	bit[95:64] of the address in *sbaddress*
sbaddress3	bit[127:96] of the address in *sbaddress*

26.5.16 System Bus Data (sbdata0 – sbdata3)

The *sbdata* register is supported as below when SYSTEM_BUS_ACCESS_SUPPORT parameter of NCE-PLDM200 is set. Otherwise it is hardwired to zeros.

Table 205: System Bus Data Register

Address Register	Description
sbdata0	bit [31:0] of the address in *sbdata*
sbdata1	bit [63:32] of the address in *sbdata*
sbdata2	bit [95:64] of the address in *sbdata*
sbdata3	bit [127:96] of the address in *sbdata*

26.5.17 Interface Signals

The interface signals consist of four groups of interfaces: a DMI interface, a bus slave interface, a system bus master interface and the General Signals interface.

The DMI interface is a dedicated bus interface for communicating with debug transport modules (NCEJDTM200).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

The bus slave interface is the bus interface for the target processor and the Debug Module to exchange data.

The system bus master interface is an optional interface for supporting the direct System Bus Access feature of the RISC-V External Debug Specification. This feature is not enabled by default and it can be turned on through the SYSTEM_BUS_ACCESS_SUPPORT parameter of NCEPLDM200. The interface signals, however, are always present on the module port list regardless of configuration. They should be left floating and ignored when not enabled.

Official Release

NCEPLDM200 offers two types of bus interfaces for the bus slave interface and the system bus interface: AHB interface and AXI interface. The interface types are selected by the RV_BUS_TYPE and SYS_BUS_TYPE parameters. The interface signals of both types are simultaneously present on the module port list and only the selected one will be used. The other group of signals will be unused and left floating. The values of RV_BUS_TYPE and SYS_BUS_TYPE parameters should match the bus type of the processor. Note that the bus protocol of the DMI interface is not configurable and unconditionally uses the AHB interface.

The General Signals include the clock and reset signals, processor status and debug-interrupt signals. NCEPLDM200 supports multi-hart (multi-target) debugging. The number of harts supported is controlled by its NHART parameter, and the width of all hart-specific signals are NHART-wide, one bit per hart.

The tables below describe the interface signals of NCEPLDM200, and the clock to NCEPLDM200 should be synchronous to that of the processor. All signals are Active-High unless otherwise indicated.

Table 206: General Signals of NCEPLDM200

Signal Name	Direction	Description
clk	input	Clock input. This clock should also drive the DMI interface of NCEJDTM200. Furthermore, this clock should not be stopped during ndmreset. See Section 26.2 for integration requirements.
dmi_resetn	input	Reset (Active-Low). The reset signal should be driven by NCEJDTM200. It should not be active when ndmreset is asserted. See Section 26.2 for integration requirements.
hart_halted[NHART-1:0]	input	Hart halted, one bit per hart. Each bit will be 1 if the corresponding hart is halted (in the debug mode). Note: Removed in 2.0.0 or later version.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 206: (continued)

Signal Name	Direction	Description
hart_unavail[NHART-1:0]	input	Hart unavailable, one bit per hart. Each bit will be 1 if the corresponding hart is not available for accesses by the external debugger. The hart could be in the reset or some kind of power-gating state.
hart_under_reset[NHART-1:0]	input	Hart under reset, one bit per hart. Each bit will be 1 if the corresponding hart is under reset.
debug_int[NHART-1:0]	output	Debug interrupt, one bit per hart. Each bit will be 1 if the external debugger makes a debug request to the corresponding hart. Each hart should respond to the request by entering the debug mode.
dmactive	output	Debug module active. This signal reflects the value of dmcontrol.DMACTIVE.
ndmreset	output	Non-debug module reset. This signal should be routed to the platform reset controller, so that the external debugger could trigger system reset for the platform. See Section 26.2 for integration requirements.
resethaltreq[NHART-1:0]	output	Halt-on-reset request, one bit per hart. Each bit will be 1 if the external debugger sets the corresponding hart to enter the debug mode after reset (the halt-on-reset requests).

Official Release

Table 207: DMI Interface Signals of NCEPLDM200

Signal Name	Direction	Description
dmi_hrdata[31:0]	output	DMI read data bus
dmi_hreadyout	output	DMI transfer done of NCEPLDM200
dmi_hresp[1:0]	output	DMI transfer response
dmi_hsel	input	DMI selection
dmi_htrans[1:0]	input	DMI transfer type
dmi_haddr[9:0]	input	DMI address bus
dmi_hburst[2:0]	input	DMI burst type
dmi_hprot[3:0]	input	DMI protection control
dmi_hsize[2:0]	input	DMI transfer size
dmi_hready	input	DMI transfer done

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 207: (continued)

Signal Name	Direction	Description
dmi_hwrite	input	DMI transfer direction
dmi_hwdata[31:0]	input	DMI write data bus

Official Release

Table 208: AHB Slave Signals of NCEPLDM200

Signal Name	Direction	Description
rv_hrdata[63:0]	output	Processor read data bus
rv_hreadyout	output	Processor transfer done of NCEPLDM200
rv_hresp[1:0]	output	Processor transfer response
rv_haddr[BIU_ADDR_WIDTH-1:0]	input	Processor address bus
rv_hburst[2:0]	input	Processor burst type
rv_hprot[3:0]	input	Processor protection control
rv_hsize[2:0]	input	Processor transfer size
rv_htrans[1:0]	input	Processor transfer type
rv_hwdata[63:0]	input	Processor write data bus
rv_hwrite	input	Processor transfer direction
rv_hsel	input	Processor selection
rv_hready	input	Processor transfer done

Table 209: AHB Master Signals of NCEPLDM200

Signal Name	Direction	Description
sys_hrdata[63:0]	input	System bus read data bus
sys_hready	input	System bus transfer done
sys_hgrant	input	System bus bus grant
sys_hresp[1:0]	input	System bus transfer response
sys_haddr[BIU_ADDR_WIDTH-1:0]	output	System bus address bus
sys_hburst[2:0]	output	System bus burst type
sys_hprot[3:0]	output	System bus protection control
sys_hsize[2:0]	output	System bus transfer size
sys_htrans[1:0]	output	System bus transfer type
sys_hwdata[63:0]	output	System bus write data bus
sys_hwrite	output	System bus transfer direction
sys_hbusreq	output	System bus bus request

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 210: AXI Slave Signals of NCEPLDM200

Signal Name	Direction	Description
rv_awid[3:0]	input	Processor write address ID
rv_awaddr[BIU_ADDR_WIDTH-1:0]	input	Processor write address
rv_awlen[7:0]	input	Processor write burst length
rv_awsize[2:0]	input	Processor write burst size
rv_awburst[1:0]	input	Processor write burst type
rv_awlock	input	Processor write lock type
rv_awcache[3:0]	input	Processor write cache type
rv_awprot[2:0]	input	Processor write protection type
rv_awvalid	input	Processor write address valid
rv_awready	output	Processor write address ready
rv_wdata[DATA_WIDTH:0]	input	Processor write data
rv_wstrb[(DATA_WIDTH/8)-1:0]	input	Processor write strobes
rv_wlast	input	Processor write last
rv_wvalid	input	Processor write valid
rv_wready	output	Processor write ready
rv_bid[3:0]	output	Processor write response ID
rv_bresp[1:0]	output	Processor write response
rv_bvalid	output	Processor write response valid
rv_bready	input	Processor write response ready
rv_arid[3:0]	input	Processor read address ID
rv_araddr[BIU_ADDR_WIDTH-1:0]	input	Processor read address
rv_arlen[7:0]	input	Processor read burst length
rv_arsize[2:0]	input	Processor read burst size
rv_arburst[1:0]	input	Processor read burst type
rv_arlock	input	Processor read lock type
rv_arcache[3:0]	input	Processor read cache type
rv_arprot[2:0]	input	Processor read protection type
rv_arvalid	input	Processor read address valid
rv_arready	output	Processor read address ready
rv_rid[3:0]	output	Processor read ID tag
rv_rdata[DATA_WIDTH:0]	output	Processor read data
rv_rresp[1:0]	output	Processor read response
rv_rlast	output	Processor read last

Continued on next page...

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 210: (continued)

Signal Name	Direction	Description
rv_rvalid	output	Processor read valid
rv_rready	input	Processor read ready

Official Release

Table 211: AXI Master Signals of NCEPLDM200

Signal Name	Direction	Description
sys_awid[3:0]	output	System bus write address ID
sys_awaddr[BIU_ADDR_WIDTH-1:0]	output	System bus write address
sys_awlen[7:0]	output	System bus write burst length
sys_awsize[2:0]	output	System bus write burst size
sys_awburst[1:0]	output	System bus write burst type
sys_awlock	output	System bus write lock type
sys_awcache[3:0]	output	System bus write cache type
sys_awprot[2:0]	output	System bus write protection type
sys_awvalid	output	System bus write address valid
sys_awready	input	System bus write address ready
sys_wdata[SYS_DATA_WIDTH:0]	output	System bus write data
sys_wstrb[(SYS_DATA_WIDTH/8)-1:0]	output	System bus write strobes
sys_wlast	output	System bus write last
sys_wvalid	output	System bus write valid
sys_wready	input	System bus write ready
sys_bid[3:0]	input	System bus write response ID
sys_bresp[1:0]	input	System bus write response
sys_bvalid	input	System bus write response valid
sys_bready	output	System bus write response ready
sys_arid[3:0]	output	System bus read address ID
sys_araddr[BIU_ADDR_WIDTH-1:0]	output	System bus read address
sys_arlen[7:0]	output	System bus read burst length
sys_arsize[2:0]	output	System bus read burst size
sys_arburst[1:0]	output	System bus read burst type
sys_arlock	output	System bus read lock type
sys_arcache[3:0]	output	System bus read cache type
sys_arprot[2:0]	output	System bus read protection type
sys_arvalid	output	System bus read address valid

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 211: (continued)

Signal Name	Direction	Description
sys_arready	input	System bus read address ready
sys_rid[3:0]	input	System bus read ID tag
sys_rdata[SYS_DATA_WIDTH:0]	input	System bus read data
sys_rresp[1:0]	input	System bus read response
sys_rlast	input	System bus read last
sys_rvalid	input	System bus read valid
sys_rready	output	System bus read ready

Official Release

Table 212: AXI Master Transactions Used by NCEPLDM200 Bus Access

Configuration	Transaction Type
SYS_DATA_WIDTH >= 32	Single BYTE
	Single HALF WORD
	Single WORD
SYS_DATA_WIDTH >= 64	Single BYTE
	Single HALF WORD
	Single WORD
	Single DOUBLE WORD

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.6 NCEJDTM200

NCEJDTM200 is an implementation of the JTAG debug transport module (DTM), as defined by the spec: *RISC-V External Debug Support (TD003) V0.13*. It implements the IEEE 1149.1 style test access port controller (TAP). The supported instructions are summarized in Table 213.

Official Release

Table 213: Supported TAP Instructions of NCEJDTM200

Encoding	Instruction	Definition
b11111	BYPASS	Section 26.6.2
b00001	IDCODE	Section 26.6.3
b10000	dtmcs	Section 26.6.4
b10001	dmi	Section 26.6.5

26.6.1 Interface Signals

Table 214: NCEJDTM200 Interface Signals

Signal Name	Direction	Description
pwr_rst_n	Input	Power on reset for NCEJDTM200
tck	Input	JTAG TCK clock
tms	Input	JTAG TMS signal
tdi	Input	JTAG TDI signal
tdo	Output	JTAG TDO signal
tdo_out_en	Output	JTAG TDO output enable signal
dmi_hresetn	Output	The reset signal for NCEPLDM200
dmi_hclk	Input	DMI clock
dmi_hsel	Output	DMI selection
dmi_haddr	Output	DMI address bus
dmi_htrans	Output	DMI transfer type
dmi_hsize	Output	DMI transfer size
dmi_hburst	Output	DMI burst type
dmi_hprot	Output	DMI protection control
dmi_hwrite	Output	DMI transfer direction
dmi_hwdata	Output	DMI write data bus
dmi_hrdata	Input	DMI read data bus
dmi_hresp	Input	DMI transfer response

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 214: (continued)

Signal Name	Direction	Description
dmi_hready	Output	DMI transfer done of NCEJDTM200
dbg_wakeup_req	Output	System wakeup request (experimental)

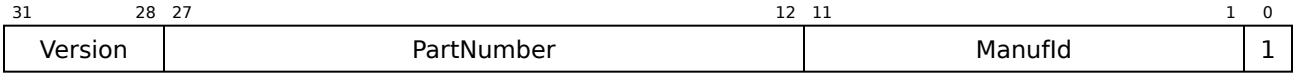
Official Release

26.6.2 BYPASS

When the TAP instruction is BYPASS, a single-bit register is connected to *tdi* and *tdo*. In Capture-DR state, the register is loaded by 0. In Shift-DR state, data is transfered from *tdi* to *tdo* through the single-bit register.

26.6.3 IDCODE

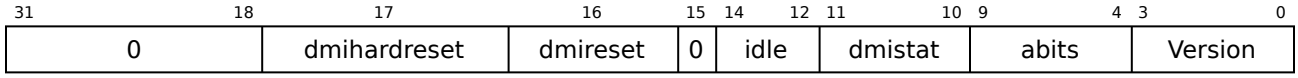
This register contains device identification code: 0x1000563D.



Field Name	Bits	Description	Type	Reset
ManufId	[11:1]	Identifies the designer/manufacturer of this part.	RO	0x31E
PartNumber	[27:12]	Identifies the designer's part number of this part.	RO	0x0005
Version	[31:28]	Identifies the release version of this part.	RO	0x1

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.6.4 DTM Control and Status (dtmcs)



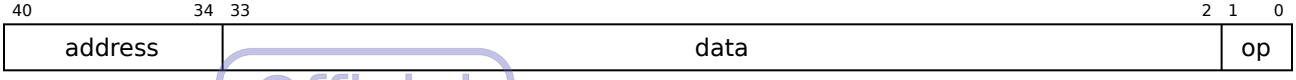
Official Release

Field Name	Bits	Description	Type	Reset
Version	[3:0]	Version of the implemented DTM. 0x1 indicates that the current implementation conforms to *RISC-V External Debug Support (TD003) V0.13*.	RO	0x1
abits	[9:4]	Bit width of DMI address is 7	RO	0x7
dmistat	[11:10]	State of DMI **Value** / **Meaning** 0 — No error 1 — Reserved 2 — An operation failed (resulted in op of 2) 3 — An operation was attempted while a DMI access was still in progress (resulted in op of 3)	RO	0x0
idle	[14:12]	This is a hint to the debugger of the minimum number of cycles a debugger should spend in RunTest/Idle after every DMI scan to avoid a *busy* return code (dmistat of 3).	RO	0x7
dmireset	[16]	Writing 1 to this bit clears the sticky error state and allows the DTM to retry or complete the previous transaction.	W1	0
dmihardreset	[17]	Writing 1 to this bit does a hard reset of the DTM, causing the DTM to forget about any outstanding DMI transactions. In general, this should only be used when the debugger has reasons to expect that the outstanding DMI transaction will never complete (e.g., a reset condition causes an inflight DMI transaction to be canceled).	W1	0

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

26.6.5 Debug Module Interface Access (dmi)

The Debug Module Interface (DMI) is accessed through this register.



Official Release

Field Name	Bits	Description	Type	Reset
op	[1:0]	Write operation: Value / Meaning: 0 – Ignore data and address. (nop) 1 – Read from address. (read) 2 – Write data to address. (write) 3 – Reserved Read operation: Value / Meaning: 0 – The previous operation completed successfully. 1 – Reserved 2 – A previous operation failed. 3 – An operation was attempted while a DMI request is still in progress. The data scanned into dmi in this access will be ignored.	RW	2
data	[33:2]	The data to send to the DM over the DMI during Update-DR, and the data returned from the DM as a result of the previous operation.	RW	0
address	[40:34]	Address used for DMI access. In Update-DR this value is used to access the DM over the DMI.	RW	0

26.6.6 Debug Wake Up Request (dbg_wakeup_req)

(This is an experimental feature in this release. Please contact Andes for more information.)

This signal is typically connected to the system management unit (SMU) to allow NCEPLDM200 to be powered off in the normal system operation unless an external debugger ever connects to the chip.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

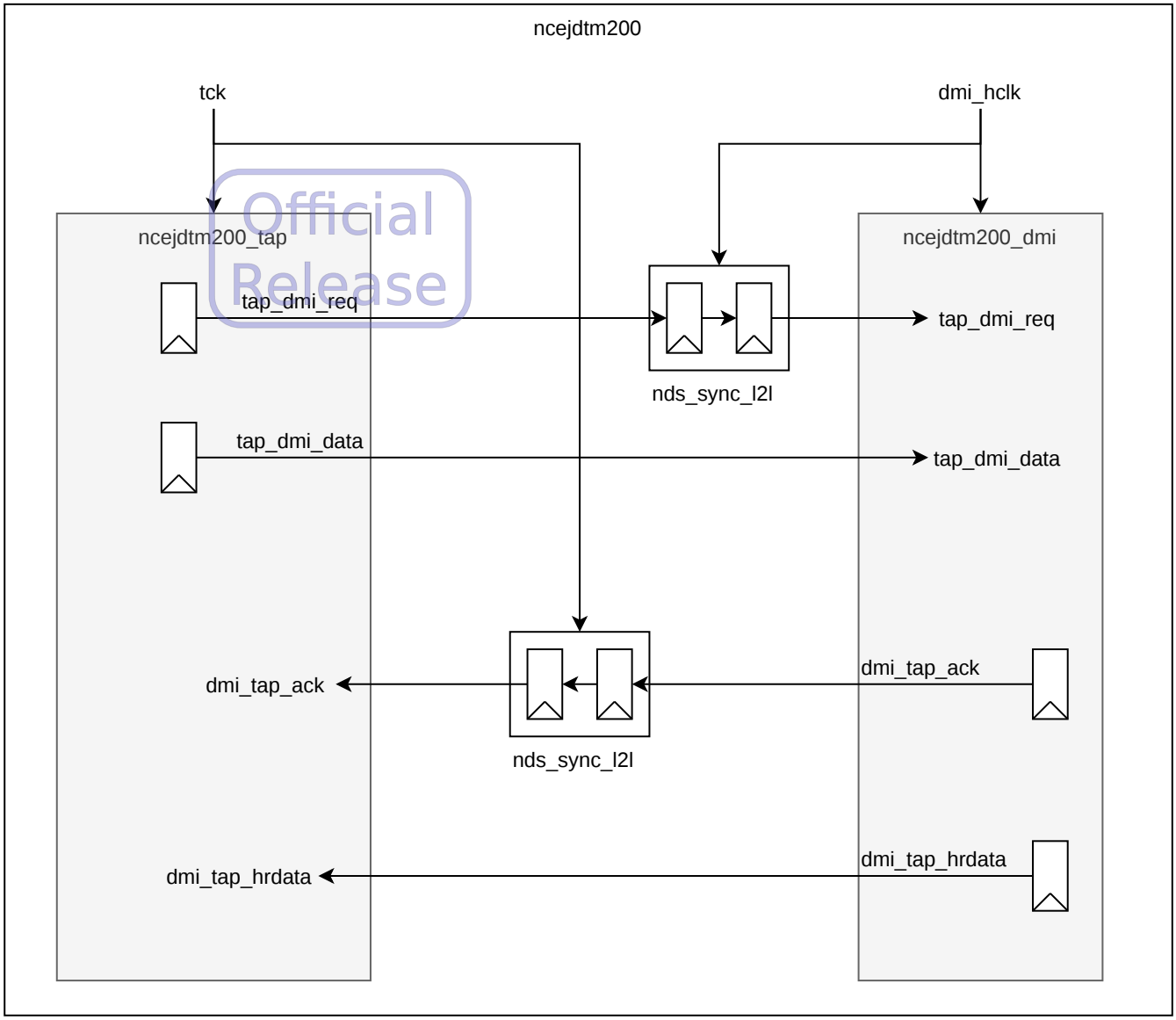
Once JTAG controller leaves the Test-Logic-Reset state, this signal will be set to indicate that the external debugger has connected to the JTAG controller. All debug related logics should therefore be powered on to handle debug requests from the external debugger. This signal is never cleared once it is set and the only way to to clear it is through power-cycling.

Official Release

26.6.7 Clock Domain Crossing (CDC)

NCEJDTM200 has two asynchronous clocks: `tck` and `dmi_hclk`. Figure 25 shows the NCEJDTM200 clock domain crossing paths. The ncejdtm200_tap is in `tck` clock domain, and ncejdtm200_dmi is in `dmi_hclk` domain. The `tap_dmi_req` and `dmi_tap_ack` are synchronized by nds_sync_l2l modules. Figure 26 shows the handshake protocol for guaranteeing tap_dmi_data and dmi_tap_hrdata are stable at the sampling clock edge.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.



Official Release

Figure 25: NCEJDTM200 Clock Domain Crossing

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

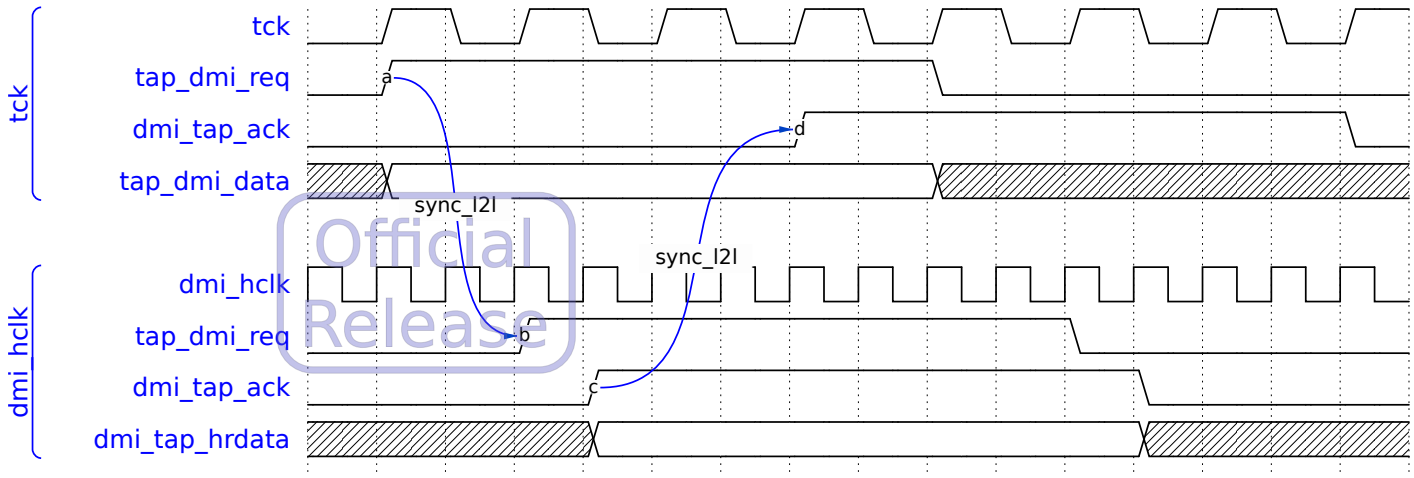


Figure 26: Handshake Protocol for tap_dmi_data and dmi_tap_hrdata

26.7 NCEDBGLOCK100

NCEDBGLOCK100 implements hardware debug security architecture that locks the JTAG signals between debug interfaces of an external debugger and the AndesCore JTAG Debug Transport Module (NCEJDTM200).

NCEDBGLOCK100 supports three debug security modes:

- Disabled Mode: the JTAG interface is permanently locked,
- Password Mode: a 128-bit password is required to unlock the JTAG interface, and
- Enabled Mode: the JTAG interface is always unlocked.

This NCEDBGLOCK100 is enabled by setting `Secure Debug Support` to `yes`.

Please see AndeShape NCEDBGLOCK100 Data Sheet for more information and see Section 23 and Section 32 for platform integration information.

26.8 Crash Debugging

AX45MP supports crash debugging for retaining states before the reset. When the processor enters debug mode immediately after the reset,

- `dpc` points to the reset vector,
- `mepc` records the program counter before reset, and
- register files retain the values before reset.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

Crash debugging is available in CPU revision 8.0.0 and later.

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

27 Andes Custom Extension (ACE)

This section describes required integration efforts needed for integrating ACE with the base AX45MP design.

Official Release

27.1 Generated Files for ACE

The verilog design and synthesis script files generated by COPILOT need to be copied to the directories of the AX45MP distribution.

A `install_ace_files` script is generated automatically under the directory where COPILOT is invoked. The script selects the relevant files, assigns the future location of the files to be copied, and then copies the files to the required directories.

27.2 Simulation with ACE

Please follow the instructions and steps in the *RTL Simulation on CPU Product Package* chapter of the *Andes Custom Extension User Manual* to install the ACE sample test cases to the directory of the AX45MP distribution and to run the RTL simulation.

27.3 Synthesis with ACE

ACE design is integrated with the processor automatically. Please follow the standard synthesis flow provided in the AX45MP product package (see Section 30 for more details).

27.4 FPGA with ACE

ACE design is integrated with the processor automatically. Please follow the standard FPGA flow provided in the AX45MP product package (see Section 32 for more details).

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

28 Models

28.1 Memory Models

Official Release

AX45MP requires several memory cells. Behavioral SRAM models are instantiated for these memory cells in the release package. These memory cells should be replaced with the cells in your library based on the actual configuration. Figure 4 illustrates the SRAM models connected to the AX45MP processor design.

Reference memory instantiations with behavioral SRAM models can be found in the **$NDS_HOME**/andes_ip/kv_core/memory/model directory. They are only good for simulations, and they should be replaced by files that instantiate the SRAM macros from the targeted technology library. It is suggested that a dedicated new directory **$NDS_HOME**/andes_ip/kv_core/memory/syn is created to hold the synthesizable copy of these memory instantiations.

28.1.1 Important Assumptions on SRAMs

All SRAMs used by AX45MP are single cycle latency: all control bits (chip-selects, addresses and write-enable) and write-data appear on the interface in the first cycle and (read) data returns on the next cycle. Please note that in case a single SRAM needs to be composed of multiple smaller SRAMs, the select signals for the read data selection mux need to be coming from flopped version of the address bus in order to align to the data cycle.

The data output ports of SRAMs are expected to hold the values from the most recent accesses. For SRAMs that do not hold the output value, wrappers must be added to remember the output values.

Combining the previous two requirements, the flopped version of the address bus for muxing data among outputs of smaller SRAMS should only be updated when the top chip select signal is asserted. Otherwise, a free-running flop for the delayed version of the address bus may toggle and change the value of output data when chip select is not asserted. Figure 27 shows how the output data mux should be implemented for a SRAM module is composed of two smaller SRAM cells.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

```
assign bank0_cs = cs & ~addr[N];
assign bank1_cs = cs & addr[N];
assign rdata = data_select ? bank1_rdata : bank0_rdata;

always @(posedge clk) begin
        if (cs) begin
                data_select <= addr[N];
        end
end
sram bank0(
        .cs (bank0_cs),
        .rdata (bank0_rdata),
        ...
);
sram bank1(
        .cs (bank1_cs),
        .rdata (bank1_rdata),
        ...
);
```

Official Release

Figure 27: Output Muxing for Composing a Larger SRAM with Smaller Ones

28.1.2 Branch Target Buffer (BTB) Organization

BTB is implemented as a two-way associative array to store the branch prediction data. BTB reference memory instantiations with behavioral SRAM models can be found in the **$NDS_HOME**/andes_ip/kv_core/memory/model/kv_btb_ram.v directory.

The SRAMs for BTB (kv_btb_ram) are instantiated in the ax45mp_core_top.

The I/O ports for btb_ram.v and SRAM dimension are summarized in Section 3.12. The signals with prefix “btb0_” and “btb1_” are for BTB way 0 and way 1, respectively.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

28.1.3 Instruction Local Memory Organization

ILM in AX45MP consists of vanilla SRAMs. The behavioral SRAM model can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model, and is dependent on **ILM Soft Error Protection** option:

- none: kv_ilm_ram.v
- ecc: kv_ilm_ecc_ram.v

Official Release

The ILM SRAMs are instantiated in the ax45mp_core_top. The I/O ports and its SRAM dimension are summarized in Section 3.8. The signals with prefix "ilm" are for ILM SRAM.

28.1.4 Data Local Memory Organization

DLM in AX45MP consists of vanilla SRAMs. The behavioral SRAM model can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model, and is dependent on **DLM Soft Error Protection** option:

- none: kv_dlm_ram.v
- ecc: kv_dlm_ecc_ram.v

The DLM SRAMs are instantiated in the ax45mp_core_top. The I/O ports and its SRAM dimension are summarized in Section 3.9. The signals with prefix "dlm" are for DLM SRAM.

28.1.5 Instruction Cache Organization

I-Cache in AX45MP consists of tag and data RAMs. Sample tag and data RAM wrappers with the behavioral SRAM model instantiation can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model/. The file name is dependent on **I-Cache Soft Error Protection** option:

- none: kv_icache_tag_ram.v and kv_icache_data_ram.v
- parity: kv_icache_tag_par_ram.v and kv_icache_data_par_ram.v

The SRAMs for I-Cache are instantiated in the ax45mp_core_top. The I/O ports for icache SRAMs and the SRAM dimensions are summarized in Section 3.10. The signals with prefix "icache_" are for I-Cache SRAMs.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

28.1.6 Data Cache Organization

D-Cache in AX45MP consists of tag and data RAMs. Sample tag and data RAM wrappers with the behavioral SRAM model instantiation can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model/. The file name is dependent on **D-Cache Soft Error Protection** option:

Official Release

- none: kv_dcache_tag_ram.v and kv_dcache_data_ram.v
- ecc: kv_dcache_tag_ecc_ram.v and kv_dcache_data_ecc_ram.v

For L2-Cache way prediction logic, D-Cache in AX45MP consists of WPT RAM, see Section 11.7 for details. Sample WPT RAM wrapper with the behavioral SRAM model instantiation can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model/. The file name is kv_dcache_wpt_ram.v.

The SRAMs for D-Cache are instantiated in the ax45mp_core_top. The I/O ports for dcache SRAMs and the SRAM dimensions are summarized in Section 3.11. The signals with prefix "dcache_" are for D-Cache SRAMs.

28.1.7 L2-Cache Organization

L2-Cache in AX45MP consists of tag and data RAMs. Sample tag and data RAM wrappers with the behavioral SRAM model instantiation can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model/. The file name is kv_l2c_bank_ram.v, kv_l2c_tag_ram.v and kv_l2c_data_ram.v.

The SRAMs for L2C are instantiated in the ax45mp_l2_top. The I/O ports for L2C SRAMs and the SRAM dimensions are summarized in Section 3.7. The signals with prefix "l2c_" are for L2C SRAMs.

28.1.8 MMU Shared TLB (STLB) Organization

STLB in AX45MP consists of tag and data vanilla SRAMs. Sample tag and data RAM wrappers with the behavioral SRAM model instantiation can be found in **$NDS_HOME**/andes_ip/kv_core/memory/model/. The file name is dependent on the **Shared TLB Soft Error Protection** option:

- none: kv_stlb_ram.v
- ecc: kv_stlb_tag_ecc_ram.v and kv_stlb_data_ecc_ram.v

The SRAMs for STLB are instantiated in ax45mp_core_top. The I/O ports for STLB SRAMs and the SRAM dimensions are summarized in Section 3.13. The signals with the prefix "stlb_" are for STLB SRAMs.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

28.2 L2-Cache RAM Setup Cycle and Output Cycle

Figure 28 shows two options of L2-Cache Data RAM output cycle:

- 2-cycle: T0 - T2
- 3-cycle: T0 - T3. This option adds one additional register slice for Data RAM output. It helps data signals to travel and converge to the L2-Cache control logic when the size of L2-Cache is very large.

Official Release

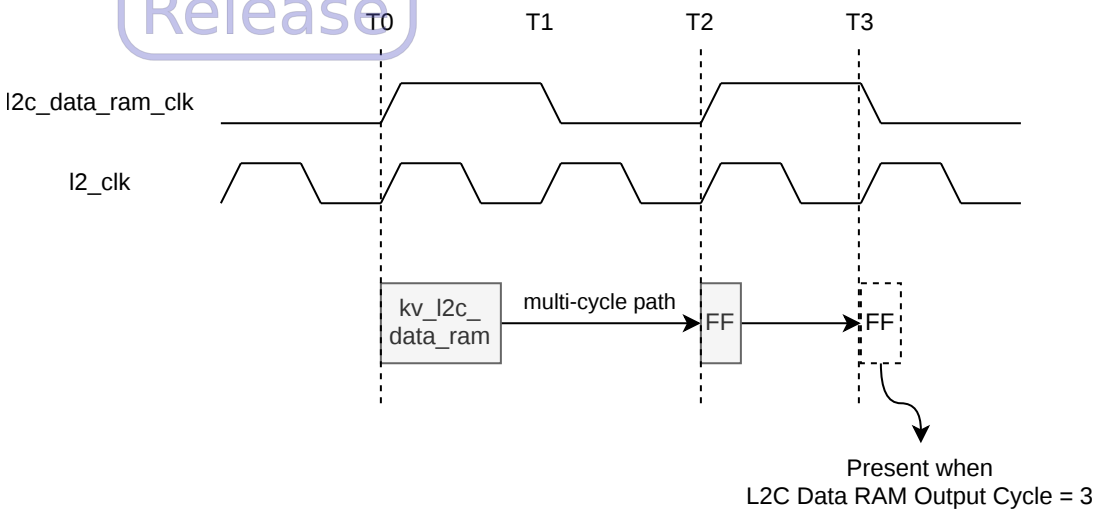


Figure 28: L2-Cache Data RAM Output Cycle

Figure 29 shows two options of L2-Cache Data RAM setup cycle:

- 1-cycle: T1 - T2
- 2-cycle: T0 - T2. This option uses multi-cycle path for L2-Cache Data RAM input signals.

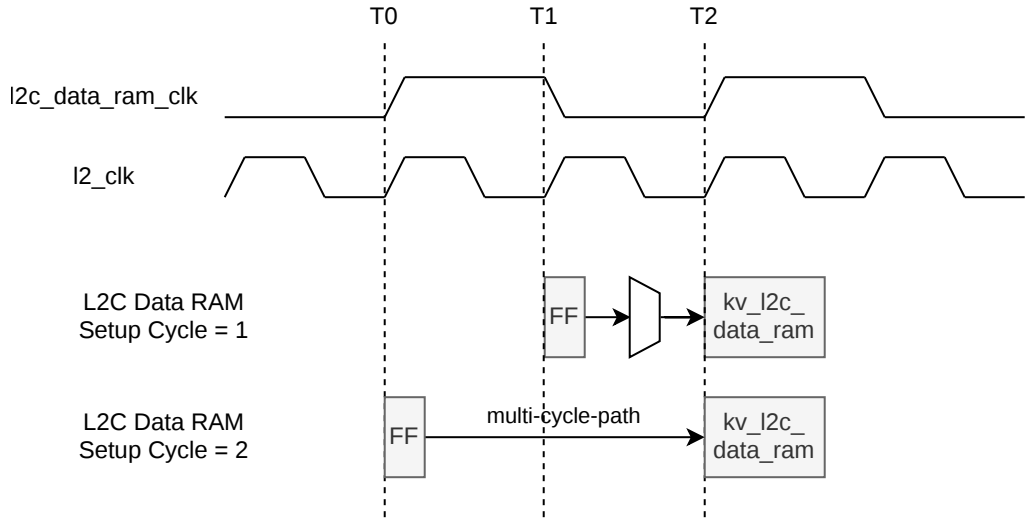


Figure 29: L2-Cache Data RAM Setup Cycle

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Figure 29 shows two options of L2-Cache Tag RAM setup cycle:

- 1-cycle: T1 - T2
- 2-cycle: T0 - T2. This option adds additional register slice for Tag RAM input signals. It helps address and control signals to distribute to the L2-Cache SRAM cells when the size of L2-Cache is very large.

Official Release

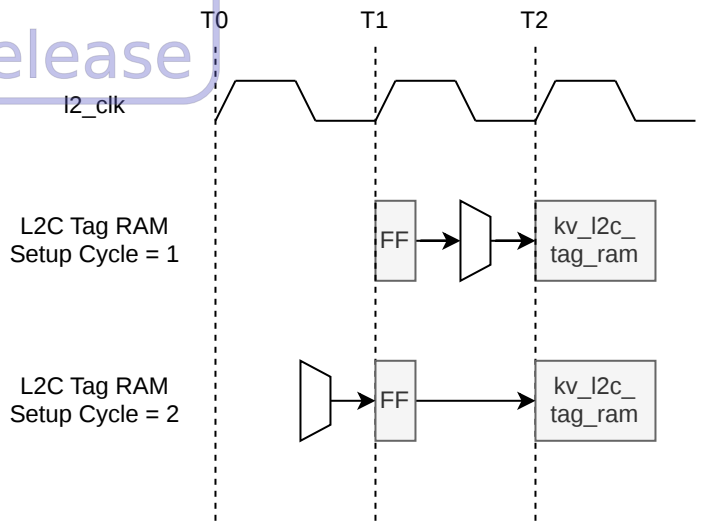


Figure 30: L2-Cache Tag RAM Setup Cycle

28.3 Clock Generation Model

AX45MP requires clock generation model when configuration option **Core Interface** is selected to "asynchronous". Behavioral PLL models is instantiated for clock generation in the release package.

Reference clock generation model with behavioral PLL models can be found at **$NDS_HOME**/andes_vip/models/axi/hdl/pll_model.v. The PLL control logic can be found at **$NDS_HOME**/andes_vip/models/axi/hdl/mmcm_dfs_model.v. They are only good for simulations, and they should be replaced by files that is generated by FPGA Macros Generation when FPGA synthesis.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

29 Simulation

29.1 Prerequisites

Official Release

Please check the following items regarding your simulation environment before performing the verification:

- The AX45MP softcore RTL requires SystemVerilog compatible simulators.
- To run simulations, the simulation environment requires standard Unix tools like `make`, `sed`, `grep` and `perl`.
- The softcore GUI configuration tool requires the Tcl/Tk interpreter `wish` to be installed in the system.
- This document assumes that environment variable **$NDS_HOME** points to the top directory of the AX45MP distribution.
- This document assumes that environment variable **$NDS_TOOLCHAIN** points to the bin directory containing Andes toolchains.
  - Note that the Andes toolchains are not part of this distribution. Please find them in the AndeSight software development tools. This distribution ships precompiled test patterns to eliminate the dependency.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

29.2 AE350 Testbench

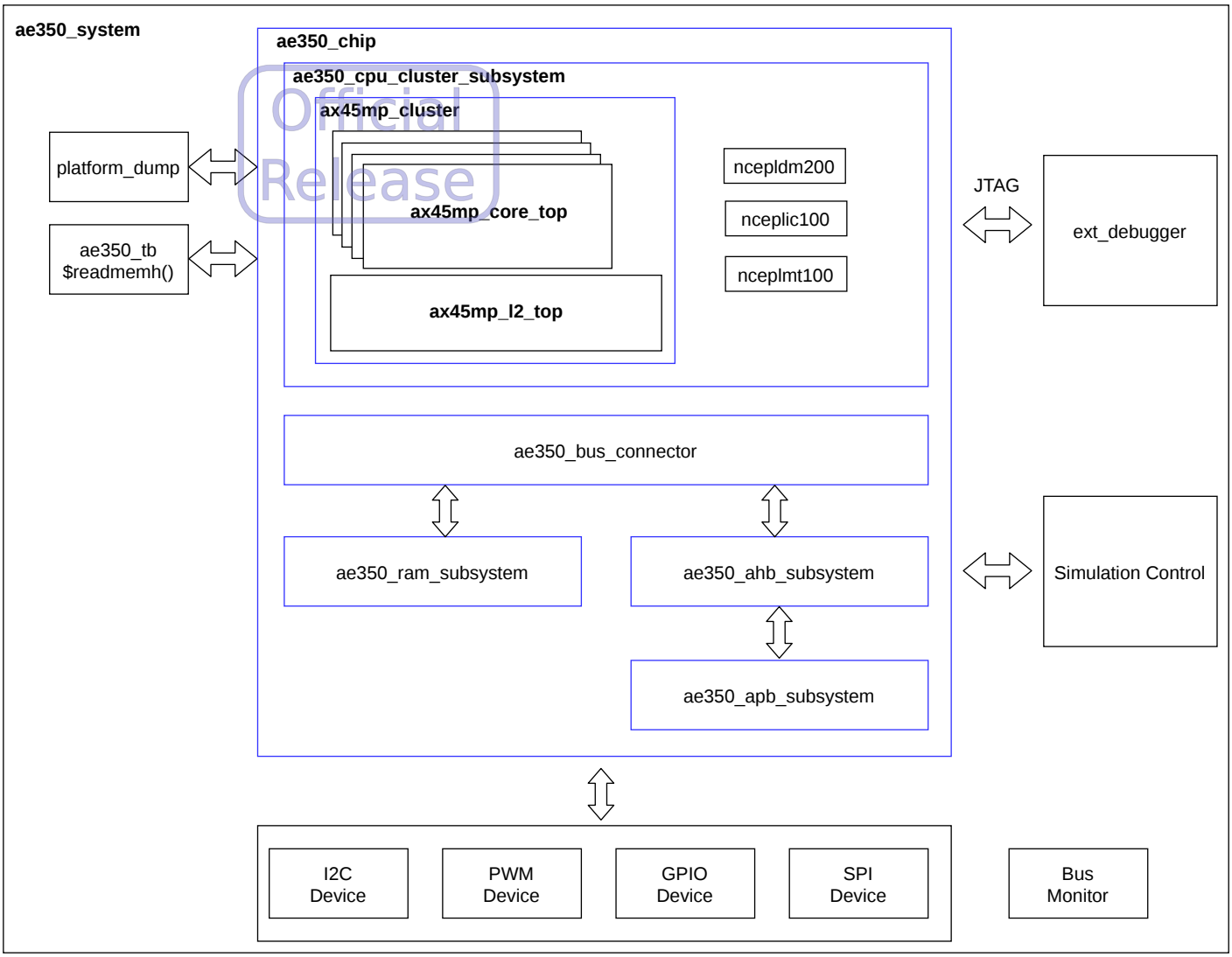


Figure 31: AX45MP Simulation Environment

Figure 31 shows the block diagram of the AX45MP simulation environment on the AE350 platform. The ax45mp_core_top module in the block diagram is the AX45MP core. The ae350_tb module provides the clock sources and the reset sequences. It also loads the ROM image (`NDSROM.dat`) of the test case to ILM, ae350_ram_subsystem and the SPI device. The simulation control module (ahb_sim_control) is connected to ae350_bus_connector and is responsible mainly for the termination of simulations. The platform_dump module controls the waveform dumping. The ext_debugger module simulates external debugger activities and interfaces with the RISC-V debug subsystem within the ae350_chip module.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

29.3 Sample Test Cases

A set of sample test cases are shipped with AX45MP for reference. The test cases are located under directory

$NDS_HOME/andes_vip/patterns/samples

Official Release

29.3.1 Quick Start

A simple test bench is included in the AX45MP distribution. To start a simulation with the generated image file,

1. Set **$NDS_HOME** to the top directory of the package:

```
bash: export NDS_HOME=<top directory of this package>
csh:  setenv NDS_HOME <top directory of this package>
```

2. Change directory to the sample pattern directory:

```
cd $NDS_HOME/andes_vip/patterns/samples
```

3. Edit Make.vars to set **$(VERILOG)** to your favorite SystemVerilog simulator.

4. Select a test case and run it:

```
cd test_dhrystone_v5
make
```

Output of the simulation should look like Figure 32. Upon a successful simulation, the "SIMULATION PASSED" string shall be observed at the end of the output file. Otherwise, simulations will either hang forever (most likely due to X-propagation) or "SIMULATION FAILED" string will appear if errors are detected.

The output message is intentionally terse to speed up simulation time. It could be decoded into assembly outputs with `ipipe_decode.pl`. The `ipipe_decode.pl` command can be found as

$NDS_HOME/tools/bin/ipipe_decode.pl

Figure 33 shows a sample decoded output.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

```
linux$ make
xrun -l verilog.log -exit +licq +nowarn+CUVWSP +nowarn+LIBNOU +nowarn+SPDUSD   -f flist
...
318446.30 ns:ipipe:0:@80000000=00000297
327821.30 ns:ipipe:0:@80000004=0102a283
327896.30 ns:ipipe:0:@80000008=00008282
328046.30 ns:ipipe:0:@00000080=0000a009
328196.30 ns:ipipe:0:@00000082=342022f3
328271.30 ns:ipipe:0:@00000086=00028463
328421.30 ns:ipipe:0:@0000008e=00004081
...
1131221.30 ns:ipipe:0:---- SIMULATION PASSED ----
1131226.00 ns:ipipe:sim_ctrl finish=0
...
linux$
```

Confidential (Release)

Figure 32: Simulation Output for Test Case `test_dhrystone_v5`

```
linux$ export PATH=$NDS_HOME/tools/bin:$PATH
linux$ ipipe_decode.pl < verilog.log
...
318446.30 ns:ipipe:0:@80000000=00000297 auipc t0, 0x00000000 (0x80000000)
327821.30 ns:ipipe:0:@80000004=0102a283 lw t0, 0x010 (t0)
327896.30 ns:ipipe:0:@80000008=00008282 c.jr zero, 0 (t0)
328046.30 ns:ipipe:0:@00000080=0000a009 c.j +0x0002 (_reset_handler:0x00000082)
328196.30 ns:ipipe:0:@00000082=342022f3 csrrs t0, mcause, zero
328271.30 ns:ipipe:0:@00000086=00028463 beq t0, zero, +0x0008 (0x0000008e)
328421.30 ns:ipipe:0:@0000008e=00004081 c.li ra, 0x00
...
1131221.30 ns:ipipe:0:---- SIMULATION PASSED ----
1131226.00 ns:ipipe:sim_ctrl finish=0
...
linux$
```

Figure 33: `ipipe_decode.pl` Output

29.3.2 SystemVerilog Simulator Selection

The test cases are launched through Makefiles. Before starting the simulation, please edit

$NDS_HOME/andes_vip/patterns/samples/Make.vars

such that the make variable **$(VERILOG)** points to a valid SystemVerilog simulator.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

29.3.3 Test Case Organization

Test cases are organized as a hierarchy of directory tree through Makefiles. The default make target compiles and runs all test cases. Typing `make` at the topmost level will run all test cases under the directory. Individual test cases can also be run by starting make at the specific test case subdirectory. Examples as below:

Official Release

```
# run all test cases under the "samples" subdirectory
cd $NDS_HOME/andes_vip/patterns/samples; make

# or, run test_dhrystone_v5 only
cd $NDS_HOME/andes_vip/patterns/samples/test_dhrystone_v5; make
```

29.3.4 Extra Options for SystemVerilog Simulators

To pass extra options to the simulator, the **$(VPLUSDEFINES)** variable could be modified through the `make` command line or through modifying

$NDS_HOME/andes_vip/patterns/samples/Make.vars

For example, the following command could be used to enable dumping waveforms:

```
make VPLUSDEFINES="+define+DUMP+TRN +access+rc"
```

29.3.5 Simulation File List

The simulation file list is defined at:

$NDS_HOME/flists/flist.in

The `flist.in` file must be processed to expand the **$NDS_HOME** variable to the actual path value before SystemVerilog simulators could accept it as a valid command line switch file. This is handled by the following rule in `Makefile`:

```
@rm -f flist
@sed -e "s,\$$NDS_HOME,$$NDS_HOME," < $$NDS_HOME/flists/flist.in \
 | grep -v "#" | sed -e "s,//*,/,g" > flist
```

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

29.3.6 NDSROM.dat Image File

The ROM image file NDSROM.dat serves as the test pattern for each test case. Its format is defined by SystemVerilog $readmemh() system task. The default make target does not attempt to regenerate the ROM file. Instead, the make rom target should be used to regenerate the NDSROM.dat file.

Official Release

Note

Some test patterns, such as Dhrystone and Coremark, have their own make targets for compiling and creating ROM images. Please see Section 29.3.8 for details.

For the make rom target to work, the toolchain programs (riscv64-elf-ar and riscv64-elf-gcc) should exist under the directory specified by **$NDS_TOOLCHAIN**, e.g.,

```
setenv NDS_TOOLCHAIN <directory of toolchain>/riscv64-elf-mculib/bin
```

Note that the toolchain programs (risv64-elf-ar and risv64-elf-gcc) are not included in the AX45MP release package. Please find them in the AndeSight software development tools.

The toolchains convert the assembly/C programs to a.out files. To make the executable files loadable, a.out must be further converted into flat binary data, and then converted to the ASCII format readable by the $readmemh() SystemVerilog system task. riscv-elf-aout2mem demonstrates how that could be done for simple a.out formats with simple .text and .data sections. riscv-elf-aout2mem comes with the AX45MP distribution and it could be found as **$NDS_HOME**/tools/bin/riscv-elf-aout2mem. It is a straightforward sample Perl script. If advanced linker sections are used, the riscv-elf-aout2mem program might need to be modified to support extra sections.

Each element of the array in NDSROM.dat is in the *big-endian* format regardless of the system endian. That is, byte 0 (0x00), 1 (0x11), 2 (0x22), and 3 (0x33) of the binary image will be represented as 0x00112233 at index 0 of the array.

29.3.7 Clean Up of Simulation Results

The target make clean can be used to clean up the simulation results.

29.3.8 Description of Test Cases

The test cases that come with this distribution could be found under the **$NDS_HOME**/andes_vip/patterns/samples directory. The test cases are described in this section.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

Some of the reference test cases may be designed for certain configurations only and may not work for all configurations. For example, the local memory related test cases are designed for local memory sizes larger than 4KiB and obviously they require the corresponding local memory support.

Please contact Andes Technology for further supports on specific test case issues.

Official Release

test_dhrystone_v5

This pattern is the precompiled version of the Dhrystone benchmark. To compile the test pattern, please get the C source code for the Dhrystone benchmark from http://www.netlib.org/benchmark/dhry-c, place it in the pattern directory, and type the command below to generate `NDSROM.dat` for later simulation:

```
make dhry
```

To show Dhrystone numbers, type:

```
make; make getdmips
```

test_coremark_v5

This pattern is the precompiled version of the CoreMark benchmark. The Makefile is setup to automatically get coremark from its official github source: https://github.com/eembc/coremark. Please make sure your https_proxy setting is correctly setup for git and type the command below to generate `NDSROM.dat` for later simulation:

```
make coremark
```

To show CoreMark numbers, type:

```
make; make getcoremark
```

test_whetstone_v5

This pattern is only available when FPU extension supported. It is the precompiled version of the Whetstone benchmark. To compile the test pattern, please get and port Whetstone source code from http://www.roylongbottom.org.uk/whets.c, place it in the pattern directory, and type the command below to generate `NDSROM.dat` for later simulation:

```
make whet
```

To change the floating-point precision (single or double), the **$(FPU_TEST_TYPE)** variable could be modified through the `make` command line or through modifying

```
$NDS_HOME/andes_vip/patterns/samples/test_whetstone_v5/Makefile
```

For example, the following command could be used to change to double precision:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

```
make whet FPU_TEST_TYPE=dp
```

To show WISP numbers, type:

```
make; make getwips
```

Official Release

test_mem_macro

This pattern tests integrity and connectivity of instantiated memory macro.

test_meminfo

This pattern extracts information for used memory blocks in the design.

```
make getmeminfo
```

test_icache_sram

This pattern tests the connectivity of SRAM memories. This pattern touches every data and address bit of I-Cache memories.

test_dcache_sram

This pattern tests the connectivity of SRAM memories. This pattern touches every data and address bit of D-Cache memories.

test_btb_sram, test_dlm_sram, and test_ilm_sram

These patterns test the connectivity of SRAM memories. These patterns touch every data and address bit of BTB, DLM and ILM memories.

test_wpt_sram

This pattern tests the connectivity of SRAM memories. This pattern touches every data and address bit of D-Cache WPT memories.

test_l2_cache_sram

This pattern tests the connectivity of L2-Cache SRAM memories. This pattern touches every data and address bit of L2-Cache memories.

test_load_performance

This pattern measures the throughput and latency for aligned and misaligned load instructions in different conditions.

- Load from cacheable memory
  - Data hits in D-Cache
  - Data hits in another hart's D-Cache with share state
  - Data hits in another hart's D-Cache with exclusive state

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

  - Data hits in another hart's D-Cache with modified state
  - Data hits in L2-Cache
  - Data in memory and D-Caches/L2-Cache are in miss state
- Load from non-cacheable memory
- Load from device memory

Official Release

To generate and see the report for the performance of load instructions, type:

```
make rpt; vi performance.log
```

test_iocp_latency

This pattern measures latency of access through IOCP. The pattern measures the latency of memory access with user defined cacheability and the latency of device access. In memory access, the pattern tests cache line in different conditions.

- Conditions:
  - Data hits in D-Cache with share state
  - Data hits in D-Cache with exclusive state
  - Data hits in D-Cache with modified state
  - Data hits in L2-Cache
  - Data in memory and D-Caches/L2-Cache are in miss state

  The tested cacheability of memory access can be modified by the following command. Values of cacheability should be assigned in decimal representation. See Table 128 and Table 127 for the supported values of ARCACHE and AWCACHE respectively.

```
make VPLUSDEFINES="+define+TB_IOCP_ARCACHE=15 +define+TB_IOCP_AWCACHE ↩
   =15"
```

  To generate and see the report of access latency, type:

```
make rpt; vi latency.log
```

test_slvp_latency

This pattern tests the latency of access through slave port. The pattern measures latency of ILM/DLM access if related memory is configured.
To generate and see the report of access latency, type:

```
make; make rpt; vi latency.log
```

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

test_mtip_all_cores

This pattern demonstrates mtip interrupts all cores simultaneously. This pattern is only available for multi-core processors.

test_int_all_cores

This pattern demonstrates meip interrupts all cores simultaneously. This pattern is only available for multi-core processors.

Official Release

test_mmcm_dfs

This pattern demonstrates dynamic frequency control of core clock generation.

test_caches

This pattern turns on both I-Cache and D-Cache. The test pattern causes various corners of the caches to be accessed.

test_pmp

This pattern turns on physical memory protection for both instruction fetch and data accesses.

test_mmu

This pattern turns on paging for both instruction fetch and data accesses.

test_stlb_sram

This pattern tests the connectivity of SRAM memories. This pattern touches every data and address bit of STLB memories.

test_mutual_exclusion

This pattern implements the Dekker algorithm which is a mutual exclusion algorithm in concurrent programming. The pattern serves to demonstrate cache coherency capability of the AX45MP system and it requires the configuration of at least two CPU cores with L1 and L2 caches.

test_exclusive_access

This pattern tests exclusive accesses to non-cacheable memory.

test_debugger

This pattern demonstrates external debugger accesses.

To test secure debug feature, type:

```
make VPLUSDEFINES="+define+PLATFORM_NCEDBGLOCK100_PASSWD_MODE"
```

test_wfi_resume

This pattern tests entering and leaving the WFI mode.

test_atcgpio100

This pattern tests the interrupt generated by GPIO.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

test_atcpit100
This pattern tests the timer in 8/16/32-bit modes.

test_atcrtc100
This pattern tests the RTC interrupts.

Official Release

test_atcwdt200
This pattern tests accesses to the registers of the watchdog timer.

test_atcapbbrg100
This pattern tests accesses to the registers of the APB bridge.

test_atcbmc300
This pattern tests accesses to the registers of the bus matrix.

test_atcuart100
This pattern tests UART read/write transactions.

test_atcspi200
This pattern tests SPI read/write transactions through register programming.

test_atcspi200_slave
This pattern tests SPI read/write transactions in the slave mode.

test_atciic100
This pattern tests I2C read/write transactions with interrupts.

test_atcdmac300
This pattern tests DMA accesses.

test_dsp
This pattern tests the RISC-V P extension (draft) DSP/SIMD instruction.

test_riscv_bitmanip
This pattern tests RISC-V bit-manipulation instructions.

test_suspend_to_standby
This pattern tests light-sleep (clock-gated) control flow. It demonstrates that the core clock is gated when the core enters WFI mode. The core resumes after its clock is recovered from interrupt events.

test_cpu_hotplug
This pattern tests deep-sleep (power-gated) control flow. It demonstrates that the core is powered down after entering WFI mode. The core wakes up after its power is restored from wake up command of corresponding power control slot.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

test_suspend_to_mem

This pattern tests deep-sleep (power-gated) control flow. It demonstrates that the cluster are powered down after all cores entering WFI mode. The core wakes up after its power is restored from interrupt events.

test_power_core

This pattern is to test max power of processor cores. Please see **$NDS_HOME**/andes_vip/patterns/samples/test_power_core/README.txt for details.

Official Release

test_power_l2c

This pattern is to test max power of L2-Cache. Please see **$NDS_HOME**/andes_vip/patterns/samples/test_power_l2c/README.txt for details.

29.3.9 Simulation Control

These patterns run in a self check manner. Upon detecting any unexpected error, the simulation will be early terminated by the program writing a specific value to ahb_sim_control to abort the simulation. Or, if everything goes fine, the program in the end writes another specific value to ahb_sim_control to gracefully terminate the simulation.

On the hardware side, ahb_sim_control achieves this by snooping AHB traffics of the internal slave (slave 0) of AHB Decoder. This internal slave is allocated a 1MiB space (see Table 179) but actually it only uses less than two hundred bytes. The BASE parameter of ahb_sim_control is set by default to 0x80000, meaning it will only check offset addresses behind 512KiB in this 1MiB space. This guarantees the existence of ahb_sim_control will not interfere the normal operation of this internal slave.

If the base memory address of this 1MiB space is changed, the monitored space of ahb_sim_control will also be effectively changed since the internal signals after address decoding inside AHB Decoder are directly used to do the snooping. This 1MiB space must reside in the device region and this guarantees the memory space of ahb_sim_control is also inside the device region.

When ahb_sim_control sees an AHB write transaction for the BASE offset with some recognized values of write data, it prints related pass/fail information and calls Verilog system task $finish to terminate the simulation. The control register information of ahb_sim_control is listed in Table 215.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 215: Simulation Control Registers

Address	I/O Type	Description
(Base address of AHB Decoder) + ahb_sim_control.BASE	Write only	Write 0x01234568 to finish simulation. Write 0x01234569 to abort simulation. Write 0x01234571 to skip simulation.

Official Release

On the software side, the program calls `exit()` with the appropriate argument. `exit()` is defined inside **$NDS_HOME**`/andes_vip/patterns/samples/src/ae350_isr.c`. The address of ahb_sim_control is decided by macro `DEV_SIM_CONTROL` which is equal to macro `SIM_CONTROL_BASE` in value. These macros are defined inside `$NDS_HOME/andes_vip/patterns/samples/include/platform.h`. When the memory map is changed, both hardware and software settings must still match each other.

29.4 RISC-V Verification Suite

The RISC-V verification suite is a set of unit tests provided by the RISC-V foundation that could be obtained from https://github.com/riscv/riscv-tests. A copy of the test suite is packaged and integrated in this release under the following directory to enable simulations with the AX45MP design:

$NDS_HOME/andes_vip/patterns/riscv-tests

Some tests of the RISC-V verification suite currently fail under RISC-V configurations that they do not expect, so a set of enhancement patches is provided to fix them. Please note that the patches may have conflicts that need to be resolved when applied to the newer RISC-V verification suite.

29.4.1 Quick Start

A simple test bench is included in the AX45MP distribution. To start a simulation with the generated image file,

1. Set **$NDS_HOME** to the top directory of the package:

```
bash: export NDS_HOME=<top directory of this package>
csh:  setenv NDS_HOME <top directory of this package>
```

2. Change directory to the directory for the RISC-V verification suite:

```
cd $NDS_HOME/andes_vip/patterns/riscv-tests
```

3. Edit Make.vars to set **$(VERILOG)** to your favorite SystemVerilog simulator.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

4. Select a test case and run it:

```
cd rundir/test_rv64ui_add
make
```

Output of the simulation should look like Figure 34. Upon a successful simulation, the "SIMULATION PASSED" string shall be observed at the end of the output file. Otherwise, simulations will either hang forever (most likely due to X-propagation) or "SIMULATION FAILED" string will appear if errors are detected.

Official Release

The output message is intentionally terse to speed up simulation time. It could be decoded into assembly outputs with `ipipe_decode.pl`. See Section 29.3.1 for how the script works.

```
linux$ make
xrun +licq +nowarn+CUVWSP  -l verilog.log -f flist +notiminechecks
...
68644.53 ns:ipipe:reset 80000000
...
95007.03 ns:ipipe:@00000044=f1402573
95082.03 ns:ipipe:@00000048=00051063
95107.03 ns:ipipe:@0000004c=30102573
95182.03 ns:ipipe:@00000050=02054063
95282.03 ns:ipipe:@00000070=00000193
95307.03 ns:ipipe:@00000074=00000297
95332.03 ns:ipipe:@00000078=f9028293
...
109682.00 ns:ipipe:sim_ctrl finish=0
109682.03 ns:ipipe:0:---- SIMULATION PASSED ----
...
linux$
```

Figure 34: Simulation Output for Test Case `test_rv64ui_add`

29.4.2 Updating to the Latest Test Suite

The latest RISC-V verification suite can be found at https://github.com/riscv/riscv-tests. Please copy the newer "isa" directory to replace

$NDS_HOME/andes_vip/patterns/riscv-tests/isa

After the test suite is updated, please execute `setup.sh` and generate `NDSROM.dat` again.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

29.4.3 Creating Makefile and Test Case Directory

Execute `setup.sh` and it will create `Makefile` and test case directories depending on `Makefile.in` and configuration files. Please note that `setup.sh` should be executed each time the processor configuration changes or the latest test suite updates.

$NDS_HOME/andes_vip/patterns/riscv-tests/setup.sh

Official Release

29.4.4 `NDSROM.dat` Image File

Please see Section 29.3.6 for the description of how to compile and generate the `NDSROM.dat` image file for simulation.

In addition to the descriptions in Section 29.3.6, the patch.sh script will be run to apply enhancement patches to the RISC-V verification suites when building image files by using `make rom` target. Please note that the patches may have conflicts with the newer RISC-V verification suite downloaded from Internet.

29.4.5 SystemVerilog Simulator Selection

Before starting the simulation, please edit

$NDS_HOME/andes_vip/patterns/riscv-tests/Make.vars

such that the make variable **$(VERILOG)** points to a valid SystemVerilog simulator.

Note
The Makefiles for the RISC-V verification suite do not share the same settings used by the sample test patterns described earlier in Section 29.3. So settings of all variables should be assigned separately.

29.4.6 Test Case Organization

Test cases are organized as a hierarchy of directory tree through Makefiles. The default make target compiles and runs the test cases. Typing `make` at the topmost level will run all test cases under the directory. Individual test cases can also be run by starting make at the specific test case subdirectory. Examples as below:

```
# run all test cases under the "riscv-tests/rundir" directory
cd $NDS_HOME/andes_vip/patterns/riscv-tests; make
```

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

```
# or, run test_rv64ui_add only
cd $NDS_HOME/andes_vip/patterns/riscv-tests/rundir/test_rv64ui_add; make
```

29.4.7 Extra Options for SystemVerilog Simulators

Official Release

To pass extra options to the simulator, the **$(VPLUSDEFINES)** variable could be modified through the `make` command line or through modifying

$NDS_HOME/andes_vip/patterns/riscv-tests/Make.vars

For example, the following command could be used to enable dumping waveforms:

```
make VPLUSDEFINES="+define+DUMP+TRN +access+rc"
```

29.5 Simulation with UPF

The following tests demonstrate power management sequences with UPF simulation:

- test_suspend_to_standby
- test_cpu_hotplug
- test_suspend_to_mem

Simulation with “Synopsys VCS”, the `Makefile` of these test cases enable the simulator option **-upf** and load the top-level of hierarchical UPF **sample_system.upf**, and the child UPF files are loaded by **load_upf** command inside this file.

Simulation with “Cadence Xcelium”, the `Makefile` of these test cases enable the simulator option **-lps_1801** and load the top-level of flatten UPF **sample_system_flat.upf**.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

30 Synthesis of AX45MP

There are sets of synthesis scripts to support the following tools:

- Synopsys DC (Design Compiler)
- Cadence Genus

Official Release

The AX45MP core synthesis working directory is **$NDS_HOME**/andes_ip/kv_core/syn.

The reference core synthesis flow is a bottom-up flow that requires synthesizing the ax45mp_core_top, ax45mp_l2_top, and ax45mp_cluster first, before performing the synthesis of the ae350_cpu_cluster_subsystem itself.

30.1 Synopsys DC Synthesis

30.1.1 Introduction

The master run script that drives the entire synthesis flow is at

$NDS_HOME/andes_ip/kv_core/syn/run_syn

And it is expected to be invoked under the **$NDS_HOME**/andes_ip/kv_core/syn directory. This script will set up working directories and invoke the synthesizer.

The DC scripts and constraint files are under the directory **$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE**/syn/script, where the **$SUB-MODULE** are ax45mp_core_top, ax45mp_l2_top, ax45mp_cluster, and ae350_cpu_cluster_subsystem. The top-level synthesis scripts of these **$SUB-MODULE** are ax45mp_core_top.tcl, ax45mp_l2_top.tcl, ax45mp_cluster.tcl, and ae350_cpu_cluster_subsystem.tcl. They calls the rest of the synthesis scripts.

The tunable TCL variables that the synthesis scripts use are collected in the following files:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/core_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script/syn_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/syn_setup_dc.tcl

See Section 30.1.2 for more details.

After the synthesis completes, the results will be saved in the directories shown in the following table. These directories will be created if they do not exist.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 216: Synthesis Result Directories

Directory	Description
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn	Synthesis working directory
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/ddc	Directory for DDC files
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/log	Directory for log files
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/netlist	Directory for netlist files
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/rpt	Directory for synthesis reports
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/work	Directory for temp files generated during synthesis

Official Release

30.1.2 Synthesis Environment Setup

Table 217 and Table 218 show all TCL variables that can be adjusted, and they are discussed in the following subsections.

30.1.2.1 Technology Library and Memory Macros

Technology library and memory macros are specified in

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/syn_setup_dc.tcl

In addition, this TCL file also contains settings of technology-related TCL variables, which include operating_cond, loading_cell, driving_cell, max_trans (max transitions), dont_use_cells, and wire_load_group (wire load model selection).

30.1.2.2 Synthesis Configuration

Three configuration scripts needed for performing synthesis are located separately at:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/core_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script/syn_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script/io_delay.tcl

The `core_env.tcl` configuration script sets TCL variables related to root design, target frequencies, and I/O timing. The `syn_env.tcl` script parses `config.inc` for processor configurations. The `io_delay.tcl` script applies the actual I/O constraints.

Note that the `core_env.tcl` script is located one level higher than other scripts to be shared by all synthesis tools.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Setting the Target Frequencies

The target frequencies are specified in `core_env.tcl`. The target frequencies define the timing for various clock domains used by the processor core: **$core_clk_period**, **$l2_clk_period**, **$bus_clk_period**.

Official Release

When **ILM Wait Cycle** is configured, **$ilm_clk_ratio** is used to define the clock frequency for SRAMs connected to RAMBRG500. This parameter is only effective in ae350_cpu_cluster_subsystem synthesis. When **$ilm_clk_ratio** is set as 1, the clock period for the SRAMs are constrained as **$core_clk_period**. When **$ilm_clk_ratio** is set as 2, the clock for the SRAMs are constrained as 2 * **core_clk_period**. **$ilm_clk_ratio** can only be 1 or 2.

The **$max_trans** TCL variable is used to specify the desired max transition rate (slew rate). The **$apr_margin**, **$l2_apr_margin** and **$bus_apr_margin** TCL variable reserves additional margins for the backend implementation; it is the amount of margin for place & route. The **$clock_uncertainty** TCL variable describes the expected clock uncertainty after clock tree synthesis. Those TCL variables are summed up together in the script (**$synthesis_margin** = **$apr_margin** + **$clock_uncertainty**), (**$l2_synthesis_margin** = **$l2_apr_margin** + **$clock_uncertainty**) and (**$bus_synthesis_margin** = **$bus_apr_margin** + **$clock_uncertainty**) to set the synthesis tool's clock uncertainty value. Therefore, the actual cycle time for the synthesis will be (**$core_clk_period** - **$synthesis_margin**), (**$l2_clk_period** - **$l2_synthesis_margin**) and (**$bus_clk_period** - **$bus_synthesis_margin**).

After the synthesis completes, the `output_netlist.tcl` script will reset the clock uncertainty to just **$clock_uncertainty** before writing out the SDC constraint file. However, if a shorter clock period is used for reserving margins for the backend implementation already, both **$clock_uncertainty** and **$apr_margin** should be set to 0 to avoid double counting.

Selecting Processor Configurations

The processor configuration is defined by the `config.inc` file that the configuration tool generates. The script `syn_env.tcl` automatically scans `config.inc` to determine and react to the selected configuration. The `config.inc` file should be properly saved to

$NDS_HOME/andes_ip/kv_core/top/hdl/config.inc

When executing the synthesis, `config.inc` will be copied to

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/config.inc

for the synthesis script to find it.

Setting I/O Port Timing Constraints

The I/O delay constraints are set in `io_delay.tcl`. For bus signals, two thirds of the bus clock period is allocated to the external logic.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Setting Synthesis Environment

The synthesis environment setting in Table 217 could be configured with the following corresponding TCL script:

- syn_setup_dc.tcl

Official Release

Table 217: Adjustable TCL Variables in AX45MP Synthesis Scripts

Parameter	Description
tech_lib	Technology library name.
tech_lib_path	Path to the technology library.
operating_cond	Chip operating condition.
loading_cell	The input of library cell for output load estimation. For example, set loading_cell BUFX4.
driving_cell	The library cell for input driving estimation. For example, set driving_cell BUFX4.
dont_use_cells	The cells that should be excluded from the specified library during the synthesis.
wire_load_group	Wire load model selection group.
memory_lib_path	Path to memory library cells.
mem_cond	Memory macro file name suffix. Specify the file name suffix for searching the target memory library files in the memory path. The matched memory macro library file will be used for the synthesis.

Setting Synthesis Clock

The clock setting of the designs in Table 218 could be found in `core_env.tcl`.

Table 218: Adjustable TCL Variables in AX45MP Synthesis Scripts

Parameter	Description
core_clk_period	CPU clock period in nanoseconds.
l2_clk_period	L2 clock period in nanoseconds.
bus_clk_period	BUS clock period in nanoseconds.
test_clk_period	Test clock period in nanoseconds.
ilm_clk_ratio	Ratio of **core clock frequency** to **clock frequency for SRAMs connected to RAMBRG500**.
clock_uncertainty	Expected clock uncertainty in nanoseconds. The clock period will be deducted by (**$clock_uncertainty** + **$apr_margin**) for synthesis.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 218: (continued)

Parameter	Description
apr_margin	The timing margin for APR in nanoseconds. The CPU clock period will be deducted by (**$clock_uncertainty** + **$apr_margin**) for the synthesis.
l2_apr_margin	The timing margin for APR in nanoseconds. The L2 clock period will be deducted by (**$clock_uncertainty** + **$l2_apr_margin**) for the synthesis.
bus_apr_margin	The timing margin for APR in nanoseconds. The BUS clock period will be deducted by (**$clock_uncertainty** + **$bus_apr_margin**) for the synthesis.
synthesis_derate	The derating factor for cell delay.

Official Release

30.1.2.3 Reading Designs and Adding Memories

The script `read_design.tcl` is responsible for reading the AX45MP RTL design. The script is located at

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script/read_design.tcl

In addition, the synthesizable definition of core memories (`kv_l2c_tag_ram.v`,`kv_l2c_data_ram.v`, `kv_icache_tag_ram.v`, `kv_icache_tag_par_ram.v`, `kv_icache_data_ram.v`, `kv_icache_data_par_ram.v`, `kv_dcache_tag_ram.v`, `kv_dcache_tag_ecc_ram.v`, `kv_dcache_data_ram.v`, `kv_dcache_data_ecc_ram.v`, `kv_dcache_wpt_ram.v`, `kv_btb_ram.v` `kv_stlb_ram.v`, `kv_stlb_tag_ecc_ram.v`, `kv_stlb_data_ecc_ram.v`, `kv_ilm_ram.v`, `kv_ilm_ecc_ram.v`, `kv_dlm_ram.v`, and `kv_dlm_ecc_ram.v`) should be created and saved under

$NDS_HOME/andes_ip/kv_core/memory/syn/

The SRAM cells for these memories should also be saved into the same directory and added to `read_design.tcl`.

The dimension of the used memories could be got by running `test_meminfo`. (See Section 29.3.8.)

30.1.3 Starting to Synthesize

Execute the run script, `run_syn`, under directory **`$NDS_HOME`**`/andes_ip/kv_core/syn` to start the synthesis. For example,

```
cd $NDS_HOME/andes_ip/kv_core/syn
./run_syn
```

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

30.1.4 Synthesis Result

30.1.4.1 Check Log File

Execute `check_log` to scan for synthesis error messages. The script must be run under **$NDS_HOME**/`andes_ip/kv_core/syn/`**$SUB-MODULE**`/syn`. For example,

Official Release

```
cd $NDS_HOME/andes_ip/kv_core/syn/$SUB-MODULE/syn
./check_log
```

30.1.4.2 Check Report

After the synthesis completes, the timing and area reports are saved under directory **$NDS_HOME**/`andes_ip/kv_core/syn/`**$SUB-MODULE**`/syn/rpt`. The final reports could be found in the files described below:

- Area report: `area`**${itr}**`.rpt`
- Timing summary report: `timing_summary`**${itr}**`.rpt`
- Detailed timing report: `timing`**${itr}**`.rpt`

Where **${itr}** is the iteration number of incremental compiles.

30.1.4.3 Netlist, SDC, DB, and DDC Files

The netlist and SDC files are saved under the following directory:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/netlist

Note that if DC is in XG mode, the DDC file will also be saved in the directory below:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/ddc

When whole synthesis flow is completed, there will have the netlists, SDC files and DDC files be saved in the directory below:

$NDS_HOME/andes_ip/kv_core/syn/ip_database

and

$NDS_HOME/andes_ip/kv_core/syn/ip_database/ip_ddc

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

30.2 Cadence Genus Synthesis

30.2.1 Introduction

Official Release

The master run script that drives the entire synthesis flow is at

$NDS_HOME/andes_ip/kv_core/syn/run_syn_genus

And it is expected to be invoked under the **$NDS_HOME**/andes_ip/kv_core/syn directory. This script will set up working directories and invoke the synthesizer.

The Genus scripts and constraint files are under the directory **$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE**/syn/script_rc, where the **$SUB-MODULE** are ax45mp_core_top, ax45mp_l2_top, ax45mp_cluster, and ae350_cpu_cluster_subsystem. The top-level synthesis scripts of these **$SUB-MODULE** are ax45mp_core_top.tcl, ax45mp_l2_top.tcl, ax45mp_cluster.tcl, and ae350_cpu_cluster_subsystem.tcl. They calls the rest of the synthesis scripts.

The tunable TCL variables that the synthesis scripts use are collected in the following files:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/core_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script_rc/syn_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/syn_setup_genus.tcl

See Section 30.2.2 for more details. The only difference against Synopsys DC scripts is that there is no output_netlist.tcl for Cadence Genus, and the relevant code are directly inlined in ax45mp_core_top.tcl, ax45mp_l2_top.tcl, ax45mp_cluster.tcl, and ae350_cpu_cluster_subsystem.tcl.

After the synthesis completes, the results will be saved in the directories shown in the following table. These directories will be created if they do not exist.

Table 219: Synthesis Result Directories

Directory	Description
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn	Synthesis working directory
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/ilm	Directory for ilm files
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/log	Directory for log files
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/netlist	Directory for netlist files
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/rpt	Directory for synthesis reports

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

30.2.2 Synthesis Environment Setup

Table 220 and Table 221 show all TCL variables that can be adjusted, and they are discussed in the following subsections.

Official Release

30.2.2.1 Technology Library and Memory Macros

Technology library and memory macros are specified in

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/syn_setup_genus.tcl

In addition, this TCL file also contains settings of technology-related TCL variables, which include operating_cond, loading_cell, driving_cell, max_trans (max transitions), dont_use_cells, and wire_load_group (wire load model selection).

30.2.2.2 Synthesis Configuration

Three configuration scripts needed for performing synthesis are located separately at:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/core_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script_rc/syn_env.tcl
$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script_rc/io_delay.tcl

The `core_env.tcl` configuration script sets TCL variables related to root design, target frequencies, and I/O timing. The `syn_env.tcl` script parses `config.inc` for processor configurations. The `io_delay.tcl` script applies the actual I/O constraints.

Note that the `core_env.tcl` script is located one level higher than other scripts to be shared by all synthesis tools.

Setting the Target Frequencies
The target frequencies are specified in `core_env.tcl`. The target frequencies define the timing for various clock domains used by the processor core: **$core_clk_period**, **$l2_clk_period**, **$bus_clk_period**. When **ILM Wait Cycle** is configured, **$ilm_clk_ratio** is used to define the clock frequency for SRAMs connected to RAMBRG500. This parameter is only effective in ae350_cpu_cluster_subsystem synthesis. When **$ilm_clk_ratio** is set as 1, the clock period for the SRAMs are constrained as **$core_clk_period**. When **$ilm_clk_ratio** is set as 2, the clock for the SRAMs are constrained as 2 * **$core_clk_period**. **$ilm_clk_ratio** can only be 1 or 2.

The **$max_trans** TCL variable is used to specify the desired max transition rate (slew rate). The **$apr_margin**, **$l2_apr_margin** and **$bus_apr_margin** TCL variable reserves additional margins for the backend implementation; it is the amount of margin for place & route. The **$clock_uncertainty** TCL variable describes the expected clock uncertainty after clock tree synthesis. Those TCL vari-

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

ables are summed up together in the script (**$synthesis_margin** = **$apr_margin** + **$clock_uncertainty**), (**$l2_synthesis_margin** = **$l2_apr_margin** + **$clock_uncertainty**), and (**$bus_synthesis_margin** = **$bus_apr_margin** + **$clock_uncertainty**) to set the synthesis tool's clock uncertainty value. Therefore, the actual cycle time for the synthesis will be (**$core_clk_period** - **$synthesis_margin**), (**$l2_clk_period** - **$l2_synthesis_margin**) and (**$bus_clk_period** - **$bus_synthesis_margin**).

Official Release

After the synthesis completes, the **$SUB-MODULE**.tcl script will reset the clock uncertainty to just **$clock_uncertainty** before writing out the SDC constraint file. However, if a shorter clock period is used for reserving margins for the backend implementation already, both **$clock_uncertainty** and **$apr_margin** should be set to 0 to avoid double counting.

Selecting Processor Configurations

The processor configuration is defined by the config.inc file that the configuration tool generates. The script syn_env.tcl automatically scans config.inc to determine and react to the selected configuration. The config.inc file should be properly saved to

$NDS_HOME/andes_ip/kv_core/top/hdl/config.inc

When executing the synthesis, config.inc will be copied to

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/config.inc

for the synthesis script to find it.

Setting I/O Port Timing Constraints

The I/O delay constraints are set in io_delay.tcl. For bus signals, two thirds of the bus clock period is allocated to the external logic.

Setting Synthesis Environment

The synthesis environment setting in Table 220 could be configured with the following corresponding TCL script:

- syn_setup_genus.tcl

Table 220: Adjustable TCL Variables in AX45MP Synthesis Scripts

Parameter	Description
tech_lib	Technology library name.
tech_lib_path	Path to the technology library.
operating_cond	Chip operating condition.
loading_cell	The input of library cell for output load estimation. For example, set loading_cell BUFX4.
driving_cell	The library cell for input driving estimation. For example, set driving_cell BUFX4.

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 220: (continued)

Parameter	Description
dont_use_cells	The cells that should be excluded from the specified library during the synthesis.
wire_load_group	Wire load model selection group.
memory_lib_path	Path to memory library cells.
mem_cond	Memory macro file name suffix. Specify the file name suffix for searching the target memory library files in the memory path. The matched memory macro library file will be used for the synthesis.

Official Release

Setting Synthesis Clock

The clock setting of the designs in Table 221 could be found in `core_env.tcl`.

Table 221: Adjustable TCL Variables in AX45MP Synthesis Scripts

Parameter	Description
core_clk_period	CPU clock period in nanoseconds.
l2_clk_period	L2 clock period in nanoseconds.
bus_clk_period	BUS clock period in nanoseconds.
test_clk_period	Test clock period in nanoseconds.
ilm_clk_ratio	Ratio of **core clock frequency** to **clock frequency for SRAMs connected to RAMBRG500**.
clock_uncertainty	Expected clock uncertainty in nanoseconds. The clock period will be deducted by (**$clock_uncertainty** + **$apr_margin**) for synthesis.
apr_margin	The timing margin for APR in nanoseconds. The CPU clock period will be deducted by (**$clock_uncertainty** + **$apr_margin**) for the synthesis.
l2_apr_margin	The timing margin for APR in nanoseconds. The L2 clock period will be deducted by (**$clock_uncertainty** + **$l2_apr_margin**) for the synthesis.
bus_apr_margin	The timing margin for APR in nanoseconds. The BUS clock period will be deducted by (**$clock_uncertainty** + **$bus_apr_margin**) for the synthesis.
synthesis_derate	The derating factor for cell delay.

30.2.2.3 Reading Designs and Adding Memories

The script `read_design.tcl` is responsible for reading the AX45MP RTL design. The script is located at

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/script_rc/read_design.tcl

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

In addition, the synthesizable definition of core memories (kv_l2c_tag_ram.v,kv_l2c_data_ram.v, kv_icache_tag_ram.v, kv_icache_tag_par_ram.v, kv_icache_data_ram.v, kv_icache_data_par_ram.v, kv_dcache_tag_ram.v, kv_dcache_tag_ecc_ram.v, kv_dcache_data_ram.v, kv_dcache_data_ecc_ram.v, kv_dcache_wpt_ram.v, kv_btb_ram.v kv_stlb_ram.v, kv_stlb_tag_ecc_ram.v, kv_stlb_data_ecc_ram.v, kv_ilm_ram.v, kv_ilm_ecc_ram.v, kv_dlm_ram.v, and kv_dlm_ecc_ram.v) should be created and saved under

$NDS_HOME/andes_ip/kv_core/memory/syn/

Official Release

The SRAM cells for these memories should also be saved into the same directory and added to read_design.tcl.

The dimension of the used memories could be got by running test_meminfo. (See Section 29.3.8.)

30.2.3 Starting to Synthesize

Execute the run script, run_syn_genus, under directory **$NDS_HOME**/andes_ip/kv_core/syn to start the synthesis. For example,

```
cd $NDS_HOME/andes_ip/kv_core/syn

./run_syn_genus
```

30.2.4 Synthesis Result

30.2.4.1 Check Log File

Execute check_log_genus to scan for synthesis error messages. The script must be run under **$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE**/syn. For example,

```
cd $NDS_HOME/andes_ip/kv_core/syn/$SUB-MODULE/syn

./check_log_genus
```

30.2.4.2 Check Report

After the synthesis completes, the timing and area reports are saved under directory **$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE**/syn/rpt. The final reports could be found in the files described below:

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- Area report: `area${itr}.rpt`
- Timing summary report: `timing_summary${itr}.rpt`
- Detailed timing report: `timing${itr}.rpt`

Where **${itr}** is the iteration number of incremental compiles.

Official Release

30.2.4.3 Netlist, SDC, and DB Files

The netlist, SDC, and DB files are saved under the following directory:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/netlist

The ILM file will also be saved in the directory below:

$NDS_HOME**/andes_ip/kv_core/syn/**$SUB-MODULE/syn/ilm

When whole synthesis flow is completed, there will have the netlists, SDC files and ILM files be saved in the directory below:

$NDS_HOME/andes_ip/kv_core/syn/ip_database

and

$NDS_HOME/andes_ip/kv_core/syn/ip_database/ip_ilm

30.3 Timing Constraints

All AX45MP timing constraints are collected in `timing_con.tcl` under the `script` or `script_rc` directory.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

31 Synthesis of the Platform

31.1 Overview

Official Release

This section discusses the reference synthesis flow for the accompanying AE350 platform. The reference flow is a bottom-up flow that requires synthesizing the component IPs first, before performing the synthesis of the platform itself.

31.2 Reference Scripts

The reference scripts include the following:

- Synthesis scripts (see Table 222) that are applicable to all platform IPs under:

  $NDS_HOME/andes_ip/peripheral_ip/design_flow/samples

- The file lists and constraint files for the synthesis of the chip-level platform module under:

  $NDS_HOME/andes_ip/ae350/syn

- The synthesizable definition of memory (ae350_rambrg_ram.v) should be created and saved under:

  $NDS_HOME/andes_ip/ae350/memory/syn

- A file list and a constraint file for the synthesis of each IP under:

  $NDS_HOME**/andes_ip/peripheral_ip/**$IP_NAME/syn

Table 222: Reference Synthesis Scripts

File Name	Description
ip_env.tcl	Providing TCL variables used by the reference scripts for the synthesis of the chip-level module of the selected platform
Cadence Genus	
run_genus.sh	Main script to drive syntheses of all peripheral IPs and the platform for Genus
syn_genus.tcl	Top script for synthesizing one component IP
syn_setup_genus.tcl	Synthesis environment setup
syn_run_genus.tcl	Main synthesis script
Synopsys DC	

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 222: (continued)

File Name	Description
run_dc.sh	Main script to drive syntheses of all peripheral IPs and the platform for DC
syn_dc.tcl	Top script for synthesizing one component IP
syn_setup_dc.tcl	Synthesis environment setup
syn_run_dc.tcl	Main synthesis script

Official Release

31.3 Setting Environment Variables and TCL Variables

The environment variables listed in Table 223 are used by the reference scripts and must be set properly before invoking the synthesis command.

Table 223: Variables for Synthesis

Variable Name	Description
OS environment variable	
SCRIPT_PATH	Path of the synthesis scripts (**$NDS_HOME**/andes_ip/peripheral_ip/design_flow/samples)
TCL variables in ip_env.tcl	
env(core_clk_period)	Period of the AX45MP clock
env(aclk_period)	Period of the AXI clock
env(pclk_period)	Period of the APB clock
env(jdtm_clk_period)	Period of clock for the external debugger interface (NCEJDTM200)
env(pclk_period)	Period of the APB clock
env(osch_clk_period)	Period of the main clock for chip
env(spi_clk_period)	Period of the SPI clock
env(ip_def_search_path)	Search path for include files
syn_define	Macro definitions for synthesis
compile_itr	Number of synthesis iterations
report_path	Path of reports
output_path	Path of the output netlists/constraints
ip_database	Path to all netlists/constraints of component IPs for the synthesis of the chip-level module of the selected platform.
TCL variables in syn_setup_XXX.tcl(See Section 30.1.2 for more information)	

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 223: (continued)

Variable Name	Description
tech_lib	Name of the standard cell library
tech_lib_path	Path of the standard cell library
operating_cond	Operating condition of the standard cell library
mem_lib	Name of the memory library
memory_lib_path	Path of the memory library
mem_cond	Operating condition of the memory library
pad_lib	Name of the PAD library
pad_lib_path	Path of the PAD library
loading_cell	Loading cell name
driving_cell	Driving cell name
dont_use_cells	List of cells which should not be used
wire_load_group	Name of the wire-load selection group
syn_effort	Synthesis effort

Official Release

31.4 Batch Script

A reference batch script is available for driving the whole chip synthesis in one shot, including AX45MP processor, all peripheral IPs and the chip-level of the platform. These scripts contain the **$itr** variable, which assigns the iteration of AX45MP processor synthesis result for copying and renaming the necessary netlist files to the **$ip_database** directory. The **$itr** variable can be revised based on the requirement (The default itr variable value is 2).

- For Synopsys DC

  $SCRIPT_PATH/run_dc.sh

- For Cadence Genus

  $SCRIPT_PATH/run_genus.sh

31.5 Synthesizing the AX45MP Processor

The synthesis result of the selected AX45MP processor must be ready before the synthesis of the chip-level module of the platform. Please see Section 30 for more information. The following table lists the necessary files which are copied and renamed from the best synthesis iteration result in the **$ip_database** directory for the synthesis of the platform.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

File Name	Source Directory	Applied EDA Tool
AE350 Platform		
ae350_cpu_cluster_subsystem.ddc	**$NDS_HOME**/andes_ip/kv_core/syn/**ae350_cpu_cluster_subsystem**/syn/ddc	Synopsys DC
ae350_cpu_cluster_subsystem.vg	**$NDS_HOME**/andes_ip/kv_core/syn/**ae350_cpu_cluster_subsystem**/syn/netlist	Cadence Genus
ae350_cpu_cluster_subsystem.sdc	**$NDS_HOME**/andes_ip/kv_core/syn/**ae350_cpu_cluster_subsystem**/syn/netlist	Cadence Genus

Official Release

31.5.1 Synthesis with UPF

The AX45MP provides a reference UPF synthesis flow for "Synopsys Design Compiler" that is described in Section 30.

There are two files need to be modified manually to enable this flow when synthesizing the ax45mp_cluster:

- Set the variable **synthesis_upf** to 1 in:

  $NDS_HOME/andes_ip/kv_core/syn/ax45mp_cluster/syn/core_env.tcl

- Set the variable **switch_lib**, **switch_lib_path**, and **iso_lvl_lib** to target library for power switchs and isolation cells.

  $NDS_HOME/andes_ip/kv_core/syn/ax45mp_cluster/syn/syn_setup_dc_upf.tcl

There are two files need to be modified manually to enable this flow when synthesizing the ae350_cpu_cluster_subsystem:

- Set the variable **synthesis_upf** to 1 in:

  $NDS_HOME/andes_ip/kv_core/syn/ae350_cpu_cluster_subsystem/syn/core_env.tcl

- Set the variable **switch_lib**, **switch_lib_path**, and **iso_lvl_lib** to target library for power switchs and isolation cells.

  $NDS_HOME/andes_ip/kv_core/syn/ae350_cpu_cluster_subsystem/syn/syn_setup_dc_upf.tcl

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

31.6 Synthesizing Peripheral IPs

The synthesis result of the peripheral IPs must also be ready before the synthesis of the chip-level module of the platform. To synthesize a peripheral IP, environment variable **$DESIGN_NAME** should be set to the IP name in the lower case. For example, the following command sets the environment variable for the GPIO controller, ATCGPIO100:

Official Release

- For Bourne shell:

```
DESIGN_NAME=atcgpio100; export DESIGN_NAME
```

- For C shell:

```
setenv DESIGN_NAME atcgpio100
```

Each IP should be synthesized under its own working directory, by creating directories as follows:

```
mkdir $DESIGN_NAME
cd $DESIGN_NAME
```

Under the working directory, start the synthesis with the following command:

- For Cadence Genus

```
genus -f $SCRIPT_PATH/syn_genus.tcl -log ./log/genus.log
```

- For Synopsys DC

```
dc_shell-t -f $SCRIPT_PATH/syn_dc.tcl | tee dc.log
```

When the synthesis completes successfully, the synthesis report will be generated in the directory **$report_path**. The netlist, SDC file and DDC file will be generated in the directory **$output_path** and copied to the **$ip_database** directory.

31.7 Synthesizing the Chip-Level Module of the Platform

When the syntheses of the AX45MP processor and peripheral IPs are complete, the chip-level of the platform can synthesized as follows:

- Set environment variable **$DESIGN_NAME** to the chip-level module name of the selected platform, i.e., ae350_chip.
- Follow the same procedures as described in Section 31.6 for the synthesis.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

32 FPGA

32.1 FPGA Block Diagram

Official Release

The AE350 modules and the external components on the evaluation board (EVB) are illustrated in Figure 35. AE350 interfaces with external components by two UART ports, two SPI ports, up to 4 PWM channels, 32-bit GPIO, I2C, JTAG debug port, and a Secure debug port. Please see *AndeShape ADP-XC7K160/410 User Manual (UM098)* and *VCU118 Board User Guide(UG1224)* for more information. Because ADP-XC7K410 and VCU118 FPGA boards have different components and IO pins, these two boards do not share all the features. Please see *AndesCore 45-Series FPGA Bitmap User Manual (UM223)* for supported features.

32.1.1 UART

UART1 is a reserved port. UART2 is connected to the UART DB9 male connector for connecting to terminal emulators.

32.1.2 JTAG Debug Port

The JTAG debug port is connected to the JTAG header for communicating with the AICE-MICRO probe.

32.1.3 Secure Debug Port

The Secure debug port is connected to the Switch 13 (SW13) for setting the secure mode. The SW13.1 is connected to the port `X_secure_mode[0]`, and the SW13.2 is connected to the port `X_secure_mode[1]`. Table 224 shows the information for secure modes.

Table 224: Supported Debug Security Modes

X_secure_mode[1:0]	Description
0bx0	Enabled mode
0b01	Disabled mode
0b11	Password mode

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

The secure code is pre-defined in AE350, and the default value is 0x524953432d5640416e64657354656368.

Official Release

32.1.4 SPI

SPI1 is connected to the on-board flash ROM. SPI2 is connected to the connector pins shown in Table 230. Another SPI ROM could be connected to SPI2 through these pins.

32.1.5 PWM

Four PWM channels are connected to the connector pins shown in Table 234.

32.1.6 GPIO

Two seven-segment LEDs are connected to part of the GPIOs for displaying diagnostic codes during booting or GPIO testing. The rest of GPIOs are connected to the buttons and the on-board connector pins. See Table 235 for pin assignments. Table 240 for pin assignments.

32.1.7 I2C

The I2C port is connected to an on-board I2C ROM.

32.1.8 Clock Generator

The oscillators on the ADP-XC7KFF676 EVB generate a 20MHz clock source and a 32.768KHz clock source. About VCU118 EVB, the oscillator generates a 125MHz clock source. The clock generator produces the following clocks by default:

- CPU clock (60 MHz)
- AXI clock (60 MHz)
- AHB clock (60 MHz)
- APB clock (60 MHz)
- UART clock (20 MHz)
- SPI clock (66 MHz)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

- RTC 32K clock (32.768 KHz)

When configuration option **Core Interface** is selected to “asynchronous”, the CPU clocks are generated with configurable clock sources. The control register is mapped in AE350 memory space 0xC0200000 – 0xC02FFFFF. The following table shows the detail clock routing.

Official Release

Table 225: AE350 Clock Routing

Clock Root	Description
CLKOUT0	Hart 0 CPU clock
CLKOUT1	Hart 1 CPU clock
CLKOUT2	Hart 2 CPU clock
CLKOUT3	Hart 3 CPU clock

More details about the clock control register, please see Xilinx Clocking Wizard v6.0.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

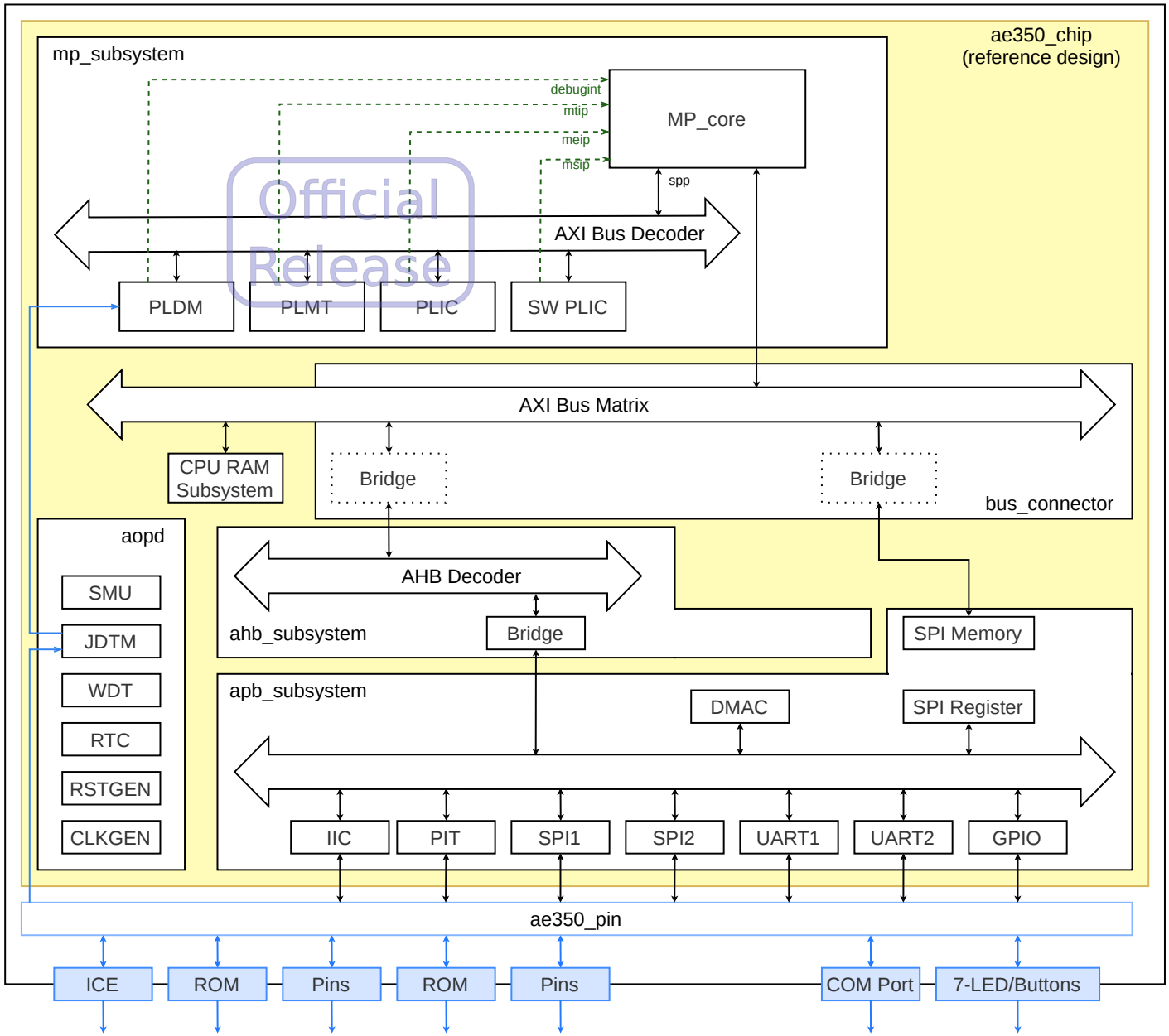


Figure 35: AE350 FPGA Block Diagram

Official Release

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

32.2 ADP-XC7KFF676 EVB FPGA Pin Assignment

32.2.1 Global Signals

Official Release

Table 226: Pin Assignment of Global Signals

Signal Name	FPGA Pin #	Board Pin Name	Component	Remark
X_hw_rstn	U22	SYS_RSTn	HW_RST_SW1	
X_oschin	G24	OSCCLK1	X1	
X_osclin	H17	RTC_32K	X2	
X_wakeup_in	D9	GPIO8	SW8	
X_por_b	U21	PORn	ALIVE Power On Reset	
X_aopd_por_b	-	-	-	Internally connected to X_por_b
X_om	-	-	-	Internally hardwired to 0
X_oschio	-	-	-	Not used on FPGA
X_osclio	-	-	-	Not used on FPGA
X_rtc_wakeup	-	-	-	Not used on FPGA
X_mpd_pwr_off	-	-	-	Not used on FPGA

32.2.2 JTAG Signals

The JTAG signals are for connecting to the external debugger.

Table 227: Pin Assignment of JTAG Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_tck	Y22	ICE_TCK	AICE_CON1.9	
X_tms	AA24	ICE_TMS	AICE_CON1.7	
X_tdo	AA22	ICE_TDO	AICE_CON1.13	
X_tdi	AC23	ICE_TDI	AICE_CON1.5	
X_trst	W20	ICE_TRSTn	AICE_CON1.3	

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

32.2.3 Secure Debug Signals

The Secure debug signals are for the security mode selection to the external debugger.

Table 228: Pin Assignment of Secure Debug Signals

Official Release

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_secure_mode[0]	A8	CFC_DATA0	SW13.1	
X_secure_mode[1]	C9	CFC_DATA1	SW13.2	

32.2.4 SPI 1: For Flash ROM

Table 229: Pin Assignment of SPI1 Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_spi1_mosi	AC6	SPI_SI	SPI ROM U17.5	
X_spi1_miso	AC4	SPI_SO_SIO1	SPI ROM U17.2	
X_spi1_clk	AA5	SPI_SCLK	SPI ROM U17.6	
X_spi1_csn	Y5	SPI_CSN	SPI ROM U17.1	
X_spi1_holdn	AB6	SPI_SIO3	SPI ROM U17.7	
X_spi1_wpn	AB5	SPI_WP#_SIO2	SPI ROM U17.3	

32.2.5 SPI 2

Table 230: Pin Assignment of SPI2 Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_spi2_mosi	H14	CFC_ADDR2	IDE_CON1.36	
X_spi2_miso	C14	CFC_NCE1	IDE_CON1.38	
X_spi2_clk	J11	CFC_ADDR0	IDE_CON1.35	
X_spi2_csn	E12	CFC_NCE0	IDE_CON1.37	
X_spi2_holdn	J10	CFC_ADDR1	IDE_CON1.33	
X_spi2_wpn	H12	CFC_PDIAG	IDE_CON1.34	

32.2.6 UART1 & UART2

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 231: Pin Assignment of UART1 Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_uart1_rxd	G25	S2_RXD	RS-232 U15.17	
X_uart1_txd	D25	S2_TXD	RS-232 U15.12	
X_uart1_ctsn		-	-	
X_uart1_rtsn		-	-	
X_uart1_dcdn		-	-	Not used on FPGA
X_uart1_dsrn		-	-	Not used on FPGA
X_uart1_dtrn		-	-	Not used on FPGA
X_uart1_out1n		-	-	Not used on FPGA
X_uart1_out2n		-	-	Not used on FPGA
X_uart1_rin		-	-	Not used on FPGA

Official Release

Table 232: Pin Assignment of UART2 Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_uart2_rxd	R18	S1_RXD	RS-232 U15.19	
X_uart2_txd	T17	S1_TXD	RS-232 U15.14	
X_uart2_ctsn		-	-	
X_uart2_rtsn		-	-	
X_uart2_dcdn		-	-	Not used on FPGA
X_uart2_dsrn		-	-	Not used on FPGA
X_uart2_dtrn		-	-	Not used on FPGA
X_uart2_out1n		-	-	Not used on FPGA
X_uart2_out2n		-	-	Not used on FPGA
X_uart2_rin		-	-	Not used on FPGA

32.2.7 I2C

Table 233: Pin Assignment of I2C Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_i2c_scl	M25	I2C_SCL	I2C FLASH U16.6	
X_i2c_sda	L25	I2C_SDA	I2C FLASH U16.5	

32.2.8 PWM

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 234: Pin Assignment of PWM Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_pwm_ch0	A15	CFC_RESET	IDE_CON1.1	
X_pwm_ch1	C12	CFC_DATA7	IDE_CON1.3	
X_pwm_ch2	D10	CFC_DATA6	IDE_CON1.5	
X_pwm_ch3	E10	CFC_DATA5	IDE_CON1.7	

Official Release

32.2.9 GPIO

Table 235: Pin Assignment of GPIO Signals

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_gpio[0]	E21	GPIO1	SW1	
X_gpio[1]	F24	GPIO2	SW2	
X_gpio[2]	J21	GPIO3	SW3	
X_gpio[3]	L23	GPIO4	SW4	
X_gpio[4]	A13	GPIO5	SW5	
X_gpio[5]	A12	GPIO6	SW6	
X_gpio[6]	J14	GPIO7	SW7	
X_gpio[7]	C11	CFC_DATA8	IDE_CON1.4	
X_gpio[8]	E11	CFC_DATA9	IDE_CON1.6	
X_gpio[9]	D11	CFC_DATA10	IDE_CON1.8	
X_gpio[10]	F14	CFC_DATA11	IDE_CON1.10	
X_gpio[11]	F13	CFC_DATA12	IDE_CON1.12	
X_gpio[12]	G12	CFC_DATA13	IDE_CON1.14	
X_gpio[13]	F12	CFC_DATA14	IDE_CON1.16	
X_gpio[14]	D14	CFC_DATA15	IDE_CON1.18	
X_gpio[15]	J8	CFC_DATA4	IDE_CON1.9	
X_gpio[16]	V26	7SEG1_A	7SEG1.A	
X_gpio[17]	W25	7SEG1_B	7SEG1.B	
X_gpio[18]	W26	7SEG1_C	7SEG1.C	
X_gpio[19]	V21	7SEG1_D	7SEG1.D	
X_gpio[20]	W21	7SEG1_E	7SEG1.E	
X_gpio[21]	AA25	7SEG1_F	7SEG1.F	
X_gpio[22]	AB25	7SEG1_G	7SEG1.G	
X_gpio[23]	W23	7SEG1_P	7SEG1.P	

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 235: (continued)

Signal Name	FPGA Pin #	Board Pin Name	Component Pin	Remark
X_gpio[24]	W24	7SEG2_A	7SEG2.A	
X_gpio[25]	AB26	7SEG2_B	7SEG2.B	
X_gpio[26]	AC26	7SEG2_C	7SEG2.C	
X_gpio[27]	Y25	7SEG2_D	7SEG2.D	
X_gpio[28]	Y26	7SEG2_E	7SEG2.E	
X_gpio[29]	AD21	7SEG2_F	7SEG2.F	
X_gpio[30]	AD23	7SEG2_G	7SEG2.G	
X_gpio[31]	AB24	7SEG2_P	7SEG2.P	

Official Release

32.3 VCU118 EVB FPGA Pin Assignment

For Board Pin Name and Component, please see Xilinx VCU118 Board User Guide (UG1224).

32.3.1 Global Signals

Table 236: Pin Assignment of Global Signals

Signal Name	FPGA Pin #	Remark
X_hw_rstn	L19	
X_oschin	AY24	
X_osclin	AR14	
X_wakeup_in	BA10	
X_por_b	BF11	
X_aopd_por_b	-	Internally connected to X_por_b
X_om	-	Internally hardwired to 0

32.3.2 UART2

Table 237: Pin Assignment of UART2 Signals

Signal Name	FPGA Pin #	Remark
X_uart2_rxd	AW25	
X_uart2_txd	BB21	

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

32.3.3 SPI 1: For Flash ROM

Table 238: Pin Assignment of SPI1 Signals

Signal Name	FPGA Pin #	Remark
X_spi1_mosi	BE12	
X_spi1_miso	BD12	
X_spi1_clk	BF9	
X_spi1_csn	BF10	
X_spi1_holdn	BD11	
X_spi1_wpn	BC11	

Official Release

32.3.4 JTAG

The JTAG signals are for connecting to the external debugger.

Table 239: Pin Assignment of JTAG Signals

Signal Name	FPGA Pin #	Remark
X_tck	AP13	
X_tms	AR13	
X_tdo	AN16	
X_tdi	AP12	
X_trst	AR12	

32.3.5 GPIO

Table 240: Pin Assignment of GPIO Signals

Signal Name	FPGA Pin #	Remark
X_gpio[0]	BD23	
X_gpio[1]	BE23	
X_gpio[2]	BF22	
X_gpio[3]	BE22	
X_gpio[4]	BB24	
X_gpio[5]	B17	
X_gpio[6]	G16	

Continued on next page...

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Table 240: (continued)

Signal Name	FPGA Pin #	Remark
X_gpio[7]	J16	
X_gpio[8]	AT32	
X_gpio[9]	AV34	
X_gpio[10]	AY30	
X_gpio[11]	BB32	
X_gpio[12]	BF32	
X_gpio[13]	AU37	
X_gpio[14]	AV36	
X_gpio[15]	BA37	
X_gpio[16]	D21	
X_gpio[17]	AY14	
X_gpio[18]	AY15	
X_gpio[19]	AW15	
X_gpio[20]	AV15	
X_gpio[21]	AV16	
X_gpio[22]	AU16	
X_gpio[23]	AT15	
X_gpio[24]	AT16	
X_gpio[25]	N28	
X_gpio[26]	M30	
X_gpio[27]	N30	
X_gpio[28]	P30	
X_gpio[29]	P29	
X_gpio[30]	L31	
X_gpio[31]	M31	

Official Release

32.4 IO Constraints

The chip-level IO pins of the AX45MP platform consist of pins mainly from peripheral controllers (e.g., SPI, UART) which communicate with off-chip components. Apart from them, the RISC-V debug transport module NCEJDTM200 also requires some IO pins for interfacing with the external debugger. As for AX45MP, all its interface signals are directly connected to other on-chip components.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

This section only describes the IO constraints for the external debug interface of NCEJDTM200. Constraints related to other peripheral IPs can be found in respective data sheets of those peripheral IPs.

32.4.1 IO Constraints for the External Debug Interface

Official Release

It is expected that the external debug interface should normally work with frequency no higher than 25MHz, so the FPGA I/O constraint for this interface could be set as follows:

```
create_clock -name {X_tck} [get_ports {X_tck}] -period 40.0 -waveform {0  ↩
   20.0}
set_clock_groups -asynchronous -name {X_tck_async_SDC} -group [get_clocks { ↩
   X_tck}]

set_output_delay   9.0 [get_ports {X_tdo}] -clock {X_tck} -add_delay
set_input_delay   30.0 [get_ports {X_tdi}] -clock {X_tck} -add_delay
set_output_delay   9.0 [get_ports {X_tms}] -clock {X_tck} -add_delay
set_input_delay   30.0 [get_ports {X_tms}] -clock {X_tck} -add_delay
```

32.4.2 IO Constraints Except the External Debug Interface

For other I/O constraints, please refer to the following constraint files.

- For AE350 Platform

  $NDS_HOME/andes_ip/ae350/fpga/vivado_flow/constraint/ae350_fpga_orca_pre_synth.sdc
  $NDS_HOME/andes_ip/ae350/fpga/vivado_flow/constraint/ae350_fpga_orca_post_synth.sdc
  $NDS_HOME/andes_ip/ae350/fpga/vivado_flow/constraint/ae350_fpga_vcu118_pre_synth.sdc
  $NDS_HOME/andes_ip/ae350/fpga/vivado_flow/constraint/ae350_fpga_vcu118_post_synth.sdc

32.5 FPGA Netlist Generation

This section describes FPGA synthesis flow to create the bitmap for the AX45MP platform.

The FPGA synthesis flow requires the Xilinx VIVADO Design Suite and it generates the bitmap for the AndeShape ADP-XC7KFF676 and VCU118 evaluation board.

The FPGA synthesis environment and the working directory are at:

$NDS_HOME/andes_ip/ae350/fpga

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

Note

Logic elements of ADP-XC7KFF676 evaluation board is not enough for multi-core configuration. If synthesizing multi-core configuration on ADP-XC7KFF676 evaluation board, the synthesis flow will be stopped and reminds “ERROR : Can’t synthesize multiple cores on xc7k410t”.

Official Release

32.5.1 FPGA Macros Generation

FPGA memory macros and DCM macros are required for the synthesis of the AE350 platform. These macros are not part of the AX45MP platform so they need to be generated separately using Xilinx tools. The following script file is included to generate all the required macros automatically:

- For AE350 Platform on ADP-XC7KFF676

```
cd $NDS_HOME/andes_ip/ae350/fpga
./gen_fpga_lib clk mem ila -part xc7k410t
```

- For AE350 Platform on VCU118

```
cd $NDS_HOME/andes_ip/ae350/fpga
./gen_fpga_lib clk mem ila -part vcu118
```

The generated macros and the file list for FPGA synthesis will be saved under the following directory:

xc7k410t:
$NDS_HOME/vendor_ip/xilinx_ip/xc7k410tffg676-2
vcu118:
$NDS_HOME/vendor_ip/xilinx_ip/xcvu9p-flga2104-2L-e

Note

To generate FPGA required memory macros, please list these macros in `gen_fpga_lib` before executing the script.

32.5.2 FPGA Synthesis

Invoke the following commands to start the FPGA synthesis:

- For AE350 Platform on ADP-XC7KFF676

```
cd $NDS_HOME/andes_ip/ae350/fpga
./syn_fpga_vivado -part xc7k410t
```

- For AE350 Platform on VCU118

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

```
cd $NDS_HOME/andes_ip/ae350/fpga
./syn_fpga_vivado -part vcu118
```

32.5.3 FPGA Synthesis Result

Official Release

The synthesis results will be saved in the following folders of the working directory:

- fpga_ae350_vivado

  xc7k410t:
  Xilinx FPGA BIN file (ae350_chip.bin)
  Xilinx FPGA MCS file (ae350_chip.mcs)
  vcu118:
  Xilinx FPGA BIN file (ae350_chip_primary.bin)
  Xilinx FPGA MCS file (ae350_chip_primary.mcs)

- fpga_ae350_vivado/ae350_chip

  Xilinx FPGA BIT file (ae350_chip.bit)

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.

33 DFT and MBIST

33.1 DFT Considerations for the Clock Gating Cells

Official Release

A requirement by ATPG test is that during ATPG scan/shift phase, clocks to all flops should not be gated off. ATPG test needs the clocks to be free running to shift in ATPG patterns and shift out the captured results. AX45MP connects `scan_enable` to all gck cells to serve this purpose. `scan_enable` can be active only during ATPG scan/shift phase and it should be off during ATPG capture phase. This allows the clock gating enable pin to be observable and achieve the best ATPG test coverage.

The DFT tool should be constrained such that the top level scan_enable pin serves the `scan_enable` purpose.

33.2 MBIST

The AX45MP design does not include MBIST logic circuit. It is up to the implementation to decide the most suitable testing strategy.

The information contained herein is the exclusive property of Andes Technology Co. and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of Andes Technology Corporation.