

Andes QEMU Emulator Reference Manual



Document Number UM284-12

Date Issued 2025-09-02

Copyright © 2024–2025 Andes Technology Corporation.
All rights reserved.



Copyright Notice

Copyright © 2024–2025 Andes Technology Corporation. All rights reserved.

AndesCore, AndeSight, AndeShape, AndeSoft, AndeStar, AndesClarity, Andes AutoOpTune, AndeSentry Secure Boot, AndeSentry MCU-TEE, AndeSim and AndeSysC are trademarks owned by Andes Technology Corporation. Please see <https://www.andestech.com/en/trademark> for a complete list of Andes trademarks and logos. All other logos and product names are the property of their respective owners.

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.

Revision History

Rev.	Revision Date	Revised Content
1.2	2025/09/02	<ol style="list-style-type: none"> Updated the document template to V19. Fixed value typos in Table 2 and Table 5. Removed the options for controlling the debug trigger in a designated execution context – “<code>sdtrig_scontext</code>”, “<code>sdtrig_hcontext</code>” and “<code>sdtrig_mcontext</code>”. (Section 2.2.3)
1.1	2025/03/31	<ol style="list-style-type: none"> Updated the document template to V17. Added A(X)46MP(V) and AX66 to the supported processors. (Section 2.1) Removed the description of accessing the semihosting console and streamlined the required options for enabling the semihosting support in the Andes QEMU emulator. (Section 2.1 and Section 2.2.5.1) Removed descriptions of the ACE debugger tool <code>libacedbg.so</code> and associated configuration option. (Section 2.2.5 and 2.2.5.1) Added how to specify IOPMP IP(s) for an emulated Andes AE350 secure platform. (Section 2.2.6) Removed the option to enable/disable the user-level interrupt support “<code>isa-n</code>” in Andes QEMU config files (Section 2.3 and 2.3.1.1) Removed descriptions of the option “<code>isa-zfh</code>” for Andes QEMU config files and updated the value indications for the option “<code>codense-ver</code>”. (Section 2.3.1.2 and Chapter 3) Added an option “<code>mrvarch-ext3</code>” to configure the <code>MRVARCH_EXT3</code> field of the <code>mrvarch_cfg</code> and <code>mrvarch_cfg2</code> registers in Andes QEMU config files. (Section 2.3.1.4)
1.0	2024/11/21	Initial Release

Table of Contents

COPYRIGHT NOTICE	I
CONTACT INFORMATION	I
REVISION HISTORY	II
LIST OF TABLES	IV
LIST OF FIGURES	V
1. INTRODUCTION	1
1.1. ANDES QEMU SOURCE OR EXECUTABLES.....	1
2. ANDES IMPLEMENTATIONS	3
2.1. FEATURE LIST	3
2.2. CPU OR SoC CONFIGURATION OPTIONS.....	6
2.2.1. <i>CPU Local Memory</i>	6
2.2.2. <i>High-Bandwidth Vector Memory (HVM)</i>	8
2.2.3. <i>Debug Trigger Module</i>	9
2.2.4. <i>RISC-V Vector Extension (RVV)</i>	10
2.2.5. <i>Andes Custom Extension (ACE)</i>	11
2.2.6. <i>Secure Platform</i>	14
2.3. CONFIGURATION FILE.....	15
2.3.1. <i>CSR Settings</i>	17
3. KNOWN ISSUES AND LIMITATIONS	27

List of Tables

TABLE 1. OPTIONS FOR ILM/DLM SETTINGS6

TABLE 2. OPTIONS FOR HVM SETTINGS8

TABLE 3. OPTION FOR DEBUG TRIGGER SETTINGS9

TABLE 4. CSRS FOR SDTRIG EXTENSION9

TABLE 5. OPTIONS FOR RVV SETTINGS10

TABLE 6. OPTIONS FOR ACE SETTINGS11

TABLE 7. OPTION FOR SECURE PLATFORM14

TABLE 8. CONFIGURATIONS FOR MACHINE ISA REGISTER17

TABLE 9. CONFIGURATIONS FOR MISC. CONFIGURATION REGISTER AND MISC. CONFIGURATION 2 REGISTER.....20

TABLE 10. CONFIGURATION FOR MISC. CONFIGURATION REGISTER324

TABLE 11. CONFIGURATIONS FOR RVA CONFIGURATION REGISTER AND CONFIGURATION 2 REGISTER.....24

TABLE 12. CONFIGURATIONS FOR VECTOR CONFIGURATION REGISTER25

TABLE 13. CONFIGURATION FOR MACHINE ARCHITECTURE ID REGISTER AND MACHINE IMPLEMENTATION ID REGISTER
.....26



List of Figures

FIGURE 1. ANDES QEMU EMULATOR WORKING WITH ACE 11



Typographical Convention Index

Document Element	Font	Font Style	Size	Color
Normal text	Georgia	Normal	12	Black
Command line, source code or file paths	Lucida Console	Normal	11	Indigo
VARIABLES OR PARAMETERS IN COMMAND LINE, SOURCE CODE OR FILE PATHS	LUCIDA CONSOLE	BOLD + ALL-CAPS	11	INDIGO
Hyperlink	Georgia	<u>Underlined</u>	12	Blue



1. Introduction

The Andes QEMU emulator is a QEMU system emulator to simulate Andes target architectures. Based on the upstream QEMU, it incorporates Andes hardware features and capabilities, allowing developers to run and test software on Andes-based virtualized environment without the need for physical hardware. The Andes QEMU emulator supports simulating Andes AE350 System on Chip (SoC) platform, associated peripherals, and a wide range of Andes processors. It offers flexible configuration options for seamless simulation.

This document outlines Andes-specific implementations and features on the Andes QEMU emulator and details the options for simulating Andes target systems. It also provides some application notes to ensure a smooth emulation process and collects known issues and limitations for developers' reference.

1.1. Andes QEMU Source or Executables

The Andes QEMU source can be obtained from the GitHub server. Follow below to copy it to a desired destination directory.

```
git clone https://github.com/andestech/qemu
```

The destination directory is the root directory of the Andes QEMU package and is represented as **QEMU_ROOT** in subsequent descriptions. To configure a QEMU emulator for your interested Andes target system, see Chapter 2 for detailed configurations.

The Andes QEMU emulator can be built and run following procedures on the QEMU website. See [QEMU Build and Test System](#) for detailed build instructions and follow the [Running](#) instructions to launch the emulator.

Andes also provides prebuilt Andes QEMU executables within the AndeSight package. If you are an AndeSight licensee, find the ready-to-use QEMU executables in **ANDESIGHT_ROOT/QEMU**, where `qemu-system-riscv32.exe` is used to simulate 32-bit Andes targets and `qemu-system-riscv64.exe` to simulate 64-bit ones. AndeSight also offers a friendly environment for launching the emulator, through the IDE or the command line, to execute programs.

You can see *AndeSight STD IDE User Manual* for instructions on launching an Andes QEMU emulator using the AndeSight IDE. As to the command-line launch procedure, see *AndeSight BSP User Manual*. For booting the Linux kernel on an Andes QEMU emulator, detailed instructions are available in *Linux Development User Manual*.



2. Andes Implementations

2.1. Feature List

- Supported platform: Andes AE350 SoC platform
Specify “-M andes_ae350” or “--machine andes_ae350” for emulating the Andes AE350 platform. Note that the AE350 platform only supports up to 2GB of memory.

- Supported peripherals:
 - Platform Level Interrupt Controller (PLIC)
 - PLIC Software
 - Machine Timer
 - Watchdog
 - Real-Time Clock (RTC)
 - Direct Memory Access (DMA) Controller
 - Network Interface Controller (NIC)
 - SD Controller
 - UART/UART2
 - Programmable Interval Timer (PIT)
 - System Management Unit (SMU)
 - Virtual I/O Device (VirtIO)

- Supported processors: Andes RISC-V processors, which include

	Supported Andes Processors
RV32	A25, A27, A45, A46MP, A46MPV, D23, D25, D25F, D45, N225, N25, N25F, N45
RV64	AX25, AX27, AX45, AX45MPV, AX46MP, AX46MPV, AX65, AX66, NX25, NX25F, NX27V, NX45, NX45V

To emulate an Andes processor, use the command “-cpu andes-CORE” to designate the processor for the emulator. For example, you can specify “-cpu andes-a25” to simulate the AndesCore A25.

To configure the processor emulator, specify options associated with desired CPU properties after “-cpu andes-CORE”. You can also define these properties in a

configuration file and use the command “`-readconfig`” to read the file. For more about the Andes QEMU configuration file, see Section 2.3.

■ Supported AndesCore features, which contain

- AndeStar V5 ISA extensions, including
 - Performance Extension
 - Andes Code Dense (CoDense) Extension
 - INT4 Vector Load Extension
 - Vector Small INT Handling Extension
 - Scalar BFLOAT16 Conversion Extension
 - Vector BFLOAT16 Conversion Extension
 - Vector Packed FP16 Extension
 - Vector Dot Product Extension
 - Vector Quad-Widening Integer Multiply-Add Extension
- Dynamic conversion for half-precision floating-point operations
The `FP_MODE` bits are used to convert between FP16 and BF16.
- AndeStar V5 Control and System Registers (CSRs)
For details about CSRs, see the data sheet of the AndesCore processor being emulated.
- Andes CPU Local Memory
See Section 2.2.1 for local memory configuration options.
- Debug Trigger Module
See Section 2.2.3 for the debug trigger control options.
- Andes Custom Extension (ACE)
See Section 2.2.5 for ACE configuration options.

■ Semihosting support

The Andes QEMU emulator supports directing semihosting system calls via itself or through a GDB client. To enable the semihosting support, program binaries must be compiled with the `-mvh` option first and the Andes QEMU emulator must be launched with the appropriate options, as listed below.

- Options for directing semihosting calls via the Andes QEMU emulator itself to the host system:
 - semihosting-config enable=on,target=native
- Options for redirecting semihosting calls via a GDB client to a GDB:
 - semihosting-config enable=on,target=gdb



2.2. CPU or SoC Configuration Options

The following subsections outline the command-line options available to configure the emulated Andes processor or System on Chip (SoC) within an Andes QEMU emulator. Additionally, they illustrate how these configurations can be defined in a configuration file tailored for the emulator. For more about supported configurations in the QEMU configuration file, see Section 2.3.

2.2.1. CPU Local Memory

To configure the ILM/DLM for the emulated Andes processor, use the command “`-global driver=riscv.andes.ae350.soc`” followed by configuration options for Instruction Local Memory (ILM) or Data Local Memory (DLM). The available options to set the ILM/DLM are as follows:

Table 1. Options for ILM/DLM Settings

Option	Description	Valid Value	Default Value
<code>ilm_default_enable</code>	Enables/Disables the ILM	<code>on/off</code>	<code>off</code>
<code>ilm_base</code>	Specifies the ILM base address	Hexadecimal	<code>0x0</code>
<code>ilm_size</code>	Specifies the ILM size	Hexadecimal	<code>0x200000</code>
<code>d1m_default_enable</code>	Enables/Disable the DLM	<code>on/off</code>	<code>off</code>
<code>d1m_base</code>	Specifies the DLM base address	Hexadecimal	<code>0x200000</code>
<code>d1m_size</code>	Specifies the DLM size	Hexadecimal	<code>0x200000</code>

The following is a command example to specify the ILM base address.

```
-global driver=riscv.andes.ae350.soc,property=ilm_base,value=0x0
```

You can also define ILM/DLM properties in a configuration file for the Andes QEMU emulator. The following is an example code snippet to set the ILM size to `0x10000` in the configuration file.

```
[global]
driver = "riscv.andes.ae350.soc"
property = "ilm_size"
value = "0x10000"
```

Note that if a specified address in local memory is the same as that in main memory, such as having the ILM base address and the main memory address on the AE350 platform both set to 0x0, it may cause problems when loading programs.

In such a case, make sure to use the command “`-device loader`”, rather than “`-bios`”, to load programs to the ILM/DLM. The command `-bios` notifies the QEMU emulator to load the assigned file (e.g., a program in ELF format) and get the entry point of the program for execution after the CPU starts/resets. It is processed during the initialization phase of the SoC platform when the CPU and the ILM/DLM are not yet ready, leading the assigned program being loaded into the main memory address at 0x0 in this example.

To ensure that your data/program can be loaded into the ILM/DLM successfully, use the command “`-device loader`” instead. This command is processed after the CPU is initialized and thus can effectively load data/programs into the specified CPU core. For more details about the option, see [Generic Loader](#) on the QEMU website.

Here is a command example to enable the ILM. It uses the command “`-device loader`” to load a program `Demo.elf` to the address 0x0 of CPU0, and set the CPU0 program counter to the entry point of `Demo.elf`.

```
-M andes_ae350 -bios none -device loader,file=/tmp/Demo.elf,cpu-num=0
-global driver=riscv.andes.ae350.soc,property=ilm_default_enable,value=on
```

2.2.2. High-Bandwidth Vector Memory (HVM)

To configure the HVM of the emulated Andes processor, specify the Table 2 options respectively after the command “`-global driver=riscv.andes.ae350.soc`”.

Table 2. Options for HVM Settings

Option	Description	Valid Value	Note
<code>hvm_base</code>	Specifies the HVM base address	Hexadecimal	
<code>hvm_size_pow_2</code>	Specifies the HVM size	A power of 2	The HVM size is 2 raised to the power of the specified value N (i.e. size = 2^N).

For example, to set the HVM base address to `0x90000000` and the size to 512 KB (2^{19}), issue the following commands:

```
-global driver=riscv.andes.ae350.soc,property=hvm_base,value=0x90000000 -global driver=riscv.andes.ae350.soc,property=hvm_size_pow_2,value=19
```

Alternatively, you can define the HVM settings in a configuration file for the Andes QEMU emulator. For instance, the HVM settings from the above command example can be defined as follows in a configuration file:

```
[global]
driver = "riscv.andes.ae350.soc"
property = "hvm_base"
value = "0x90000000"

[global]
driver = "riscv.andes.ae350.soc"
property = "hvm_size_pow_2"
value = "19"
```

2.2.3. Debug Trigger Module

The Andes QEMU emulator supports the RISC-V Sdtrig ISA extension. You can use the command “`-cpu andes-CORE`” followed by the option “`sdtrig_tcontrol`” to enable or disable the debug trigger. For details about this option, see Table 3 below.

Table 3. Option for Debug Trigger Settings

Option	Description	Valid Value	Default Value
<code>sdtrig_tcontrol</code>	Enables/Disables the trigger control.	on/off	off

For example, to enable the debug trigger on an emulated Andes AX45 processor, just issue as follows:

```
-cpu andes-ax45,sdtrig_tcontrol=on
```

Alternatively, you can configure the debug trigger in a configuration file for the Andes QEMU emulator as follows.

```
[global]
driver = "andes-ax45-riscv-cpu"
property = "sdtrig_tcontrol"
value = "on"
```

For your reference, the following are Andes CPU CSRs associated with the Sdtrig extension.

Table 4. CSRs for Sdtrig Extension

Number	Privilege	Name	Description
0x7A8	MRW	<code>mcontext</code>	Machine context register
0x6A8	MRW	<code>hcontext</code>	Hypervisor context register
0x5A8	MRW	<code>scontext</code>	Supervisor context register
0x7A5	MRW	<code>tcontrol</code>	Trigger control register

2.2.4. RISC-V Vector Extension (RVV)

For an emulated Andes processor enabled with the RISC-V Vector ISA extension, use the command “`-cpu andes-CORE`” together with the following options to configure the RVV settings.

Table 5. Options for RVV settings

Option	Description	Valid Value	Note
<code>vlen</code>	Specifies the vector length, i.e., the maximum number of bits in a single vector register.	Decimal	Its value must be no smaller than the <code>elen</code> value ($vlen \geq elen$).
<code>elen</code>	Specifies the element length, i.e., the maximum bit-width of an individual element in a vector register.	Decimal	Its value must be no less than 8 ($elen \geq 8$) and no larger than the specified <code>vlen</code> value ($vlen \geq elen$).

The following is an example to specify the vector length and element length for an emulated AndesCore AX45 that is enabled with the RVV extension.

```
-cpu andes-ax45,vlen=1024,elen=64
```

Alternatively, you can configure these RVV settings in a configuration file for the Andes QEMU emulator, as shown below.

```
[global]
driver = "andes-ax45-riscv-cpu"
property = "vlen"
value = "1024"
```

```
[global]
driver = "andes-ax45-riscv-cpu"
property = "elen"
value = "64"
```

2.2.5. Andes Custom Extension (ACE)

The Andes QEMU emulator supporting ACE enables you to design custom CPU instructions on the optimized Andes processors. As illustrated below, the emulator needs to work with an ACE simulator and the QEMU ACE agent to allow the development of ACE instructions. In the process, the ACE simulator is a simulation library generated by Custom-OPTimized Instruction deVeLOpment Tools (COPILOT) and the ACE agent is a dynamic library that bridges the QEMU emulator and the ACE simulator. The following illustration, with dashed lines and blocks to indicate optional processes and components, shows the capability of the emulator to support multiple ACE simulators. Note that only one ACE agent is allowed at a time during ACE development with the Andes QEMU emulator, even when multiple ACE simulators are involved.

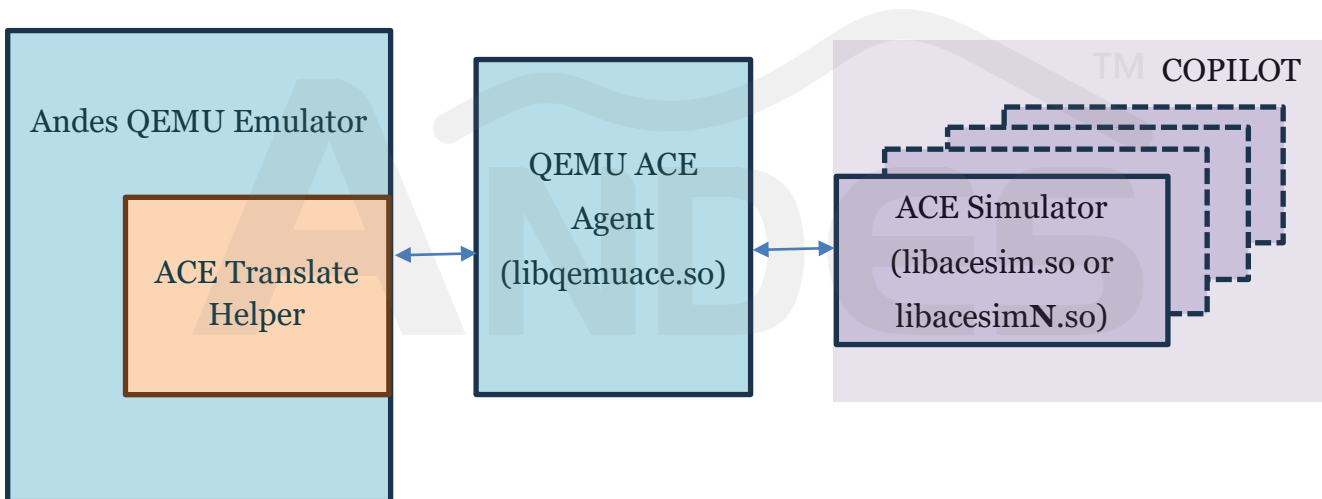


Figure 1. Andes QEMU Emulator working with ACE

The following lists ACE configuration options.

Table 6. Options for ACE Settings

Option	Description	Valid Value	Default Value
<code>xandesace</code>	Enables/Disables the ACE feature	on/off	off
<code>xandesacelib</code>	Specifies the path and file name of the ACE agent (e.g. <code>libqemuace.so</code>) in the Andes QEMU package. See Note 1 below for more information.	String	
<code>xandesaceextlibpath</code>	Specifies the location of the ACE	String	

Option	Description	Valid Value	Default Value
	simulator generated by COPILOT from an external directory. See Note 1 below for more information.		
<code>xandesacemulti</code>	Enables/Disables the support on multiple ACE architectures. Note that this option can only be enabled when multiple ACE simulators are available.	<code>on/off</code>	<code>off</code>

NOTE

1. The ACE simulator is named `libacesim.so` when there is only one ACE architecture, and `libacesimN.so` in the case of multiple ACE architectures, where **N** is a 2-digit hart ID starting from `00`.
-

To configure ACE settings, set these ACE options following the command “`-cpu andes-CORE`”. Here is an example to enable the ACE feature and specify its settings for an emulator that simulates AndesCore AX45MPV.

```
-cpu andes-ax45mpv,xandesace=on,xandesacelib="lib/libqemuace.so",xandesaceextlibpath=lib
```

You can also define ACE properties in a configuration file for the Andes QEMU emulator. The following is an example code snippet related to the ACE settings in a configuration file.

```
[global]
driver = "andes-ax45mpv-riscv-cpu"
property = "xandesace"
value = "on"
```

```
[global]
driver = "andes-ax45mpv-riscv-cpu"
property = "xandesacelib"
value = "lib/libqemuace.so"
```

```
[global]
driver = "andes-ax45mpv-riscv-cpu"
property = "xandesaceextlibpath"
value = "lib"
```

2.2.5.1 Running Andes QEMU Emulator with ACE

The following gives an example about how to run an Andes QEMU emulator with ACE. With a built Andes QEMU emulator and a QEMU ACE agent, your working directory may look like below when an ACE demo project is given and its `/include` and `/lib` directories are generated by COPILOT. The ACE demo project in this example contains a program “demo” that uses semihosting as its standard outputs.

```
workDir/  
  +-qemu/  
    +---build  
  +-qemu-ace-agent  
  +-demo-ace/  
    +---include  
    +---lib
```

You can follow below to run the Andes QEMU emulator with ACE.

Step 1 Change the current directory to the directory of the ACE demo:

```
$ cd demo-ace
```

Step 2 Copy the QEMU ACE agent library to `/lib` of the ACE demo.

```
$ cp ../qemu-ace-agent/libqemuace.so ./lib/
```

Step 3 Issue as follows to run the Andes QEMU emulator with ACE configurations.

```
$ ../qemu/build/qemu-system-riscv64 -nographic -smp 1 -m 512M  
-machine andes_ae350 --semihosting-config enable=on,target=native  
-cpu andes-ax45mpv,v=on,vlen=512,elen=32,vext_spec=v1.0,  
xandesace=on,xandesacelib="lib/libqemuace.so",xandesaceextlibpath=lib  
-bios demo
```

Also, to debug the Andes QEMU emulator with ACE, make sure you use the dedicated commands for debugging applications using ACE Registers (ACR) and ACE Memory (ACM). For more about the commands, see the section “Debug Application Software” in *Andes Custom Extension User Manual*.

2.2.6. Secure Platform

To configure the I/O Physical Memory Protection Units (IOPMPs) on an emulated Andes AE350 secure platform, use the command “`-global driver=riscv.andes.ae350.soc property=secure_platform`” to specify the associated IP(s). For details on the secure platform configuration option “`secure_platform`”, see Table 7 below.

Table 7. Option for Secure platform

Option	Description	Valid value	Default Value
<code>secure_platform</code>	Specifies the IOPMP IP(s) on the secure platform	<code>none</code> , <code>atciopmp100</code> , <code>atciopmp200</code> , <code>atciopmp300</code> , <code>atciopmp300+atciopmp100</code> , <code>atciopmp200+atciopmp100</code>	<code>none</code>

The following command example demonstrates how to use this configuration option to configure the AE350 secure platform with two IOPMP IPs, `atciopmp200` and `atciopmp100`.

```
-global driver=riscv.andes.ae350.soc,property=secure_platform,value=atciopmp200+atciopmp100
```

You can also define the `secure_platform` settings in a configuration file for the Andes QEMU emulator. The following is an example code snippet in the configuration file to configure the AE350 platform with the `atciopmp300` IOPMP IP.

```
[global]
driver = "riscv.andes.ae350.soc"
property = "secure_platform"
value = "atciopmp300"
```

2.3. Configuration File

You can define your target configurations in a configuration file and use `-readconfig` to load the file and apply the configurations all at once when launching the Andes QEMU emulator. This simplifies the setup process and makes it easier to manage complex target configurations.

For your reference, the Andes QEMU package provides a sample configuration file, `andes.cfg`, under `QEMU_ROOT/docs/config/`. If you are an AndeSight user, you can also find predefined QEMU configuration files for a number of Andes targets within `ANDESIGHT_ROOT/QEMU/qemu-cfgs/`.

To create or customize an Andes QEMU configuration file, make sure it follows the required format to organize configuration options. Specifically, the file is divided into two types of sections – one is for CPU or SoC configurations and the other is for Control/Status Register (CSR)-related configurations.

Andes CPU or SoC configurations, which can also be specified via command-line options, must be defined individually in sections labelled `[global]` following the format below:

```
[global]
driver = "CPU/SOC_QEMU_DRIVER"
property = "CPU/SOC_COMMAND_OPTION"
value = "VALUE"
```

where

- `CPU/SOC_QEMU_DRIVER` is the QEMU driver that runs the emulated Andes processor (named `andes-CPU-riscv-cpu`, such as `andes-ax45-riscv-cpu`) or SoC on the Andes AE350 platform (i.e., `riscv.andes.ae350.soc`).
- `CPU/SOC_COMMAND_OPTION` corresponds to the specific command-line option for the targeted CPU or SoC property (e.g., `ilm_base` or `sdtrig_tcontrol`).
- `VALUE` represents the specified configuration for the CPU/SoC property.

For examples of defined CPU/SoC configurations in a configuration file, see subsections in Section 2.2.

On the other hand, configurations associated with specific Andes processor CSR or bit field need to be grouped together in a different section labelled `[andes-config "cpu"]`. For

example,

```
[andes-config "cpu"]
isa-a = "on"
isa-c = "on"
isa-d = "on"
isa-f = "on"
mmsc-cfg-rvarch = "on"
isa-codense = "on"
pfm-nds = "on"
powerbrake = "on"
cctl = "on"
stack-protection = "on"
vectored-pllc = "on"
isa-dsp = "on"
```

The following subsections introduce supported CRS or bit field configurations in the `[andes-config "cpu"]` section. Note that these CRS-related configurations, unlike CPU or SoC configurations, can't be specified via command line options. For more about CSRs of a specific Andes processor and their bit field configurations, see the corresponding data sheet.

2.3.1. CSR Settings

2.3.1.1 Machine ISA Register (*misa*)

The following are options to specify a particular extension by configuring bits in the *Extensions* field of the machine ISA register (*misa*).

Table 8. Configurations for Machine ISA Register

Option	Description	Valid Value
<i>isa-a</i>	Enables/Disables the Atomic extension.	on/off
<i>isa-b</i>	Enables/Disables the Bit operations extension.	on/off
<i>isa-c</i>	Enables/Disables the compressed (16-bit) extension.	on/off
<i>isa-d</i>	Enables/Disables the double-precision floating-point extension.	on/off
<i>isa-e</i>	Enables/Disables the RV32E base ISA (16 GPRs).	on/off
<i>isa-f</i>	Enables/Disables the single-precision floating-point extension.	on/off
<i>isa-g</i>	Enables/Disables additional standard extensions.	on/off
<i>isa-h</i>	Enables/Disables the Hypervisor mode.	on/off
<i>isa-i</i>	Enables/Disables the RV32I/64I/128I base ISA.	on/off
<i>isa-j</i>	Reserved	on/off
<i>isa-k</i>	Reserved	on/off

Option	Description	Valid Value
isa-l	Enables/Disables the decimal floating-point extension.	on/off
isa-m	Enables/Disables the integer multiply/divide extension.	on/off
isa-o	Reserved	on/off
isa-p	Enables/Disables the packed-SIMD extension.	on/off
isa-q	Enables/Disables the quad-precision floating-point extension.	on/off
isa-r	Reserved	on/off
isa-s	Enables/Disables the supervisor mode.	on/off
isa-t	Enables/Disables the transactional memory extension.	on/off
isa-u	Enables/Disables the user mode.	on/off
isa-v	Enables/Disables the vector extension.	on/off
isa-w	Reserved	on/off
isa-x	Enables/Disables the non-standard extensions.	on/off
isa-y	Reserved	on/off
isa-z	Reserved	on/off

For example, to enable the vector extension and hypervisor mode for the emulated processor, specify as follows in the `[andes-config "cpu"]` section.

```
isa-v = "on"  
isa-h = "on"
```



2.3.1.2 Misc. Configuration Register and Configuration 2 Register (`mmsc_cfg/`
`mmsc_cfg2`)

The following are options to configure the bit fields of the miscellaneous configuration register (`mmsc_cfg`) and configuration 2 register (`mmsc_cfg2`).

Table 9. Configurations for Misc. Configuration Register and Misc. Configuration 2 Register

Option	Description	Valid Value
<code>ecc</code>	Enables/Disables the parity/ECC support.	<code>on/off</code>
<code>isa-codense</code>	Enables/Disables the CodeDense extension.	<code>on/off</code>
<code>powerbrake</code>	Enables/Disables the PowerBrake (Performance Throttling) feature.	<code>on/off</code>
<code>stack-protection</code>	Enables/Disables the HW Stack protection and recording mechanism.	<code>on/off</code>
<code>isa-ace</code>	Enables/Disables the Andes Custom Extension.	<code>on/off</code>
<code>vectored-plic</code>	Enables/Disables the vectored PLIC mode.	<code>on/off</code>
<code>slave-port</code>	Enables/Disables the local memory slave/access port.	<code>on/off</code>
<code>pfm-nds</code>	Enables/Disables the Andes-enhanced performance monitoring feature.	<code>on/off</code>
<code>cctl</code>	Enables/Disables CSR features for CCTL operations.	<code>on/off</code>

Option	Description	Valid Value
<code>isa-efhw</code>	Enables/Disables the FLHW and FSHW instructions.	<code>on/off</code>
<code>cctl-version</code>	Specifies the version number of the supported CCTL command operation scheme.	Decimal
<code>ppi</code>	Enables/Disables the PPI region.	<code>on/off</code>
<code>isa-dsp</code>	Enables/Disables the DSP extension.	<code>on/off</code>
<code>ppma</code>	Enables/Disables the programmable PMA mechanism.	<code>on/off</code>
<code>msc-ext</code>	Specifies whether the CSR of the miscellaneous configuration 2 register (<code>mmsc_cfg2</code>) is present.	<code>on/off</code>
<code>isa-bf16cvt</code>	Enables/Disables the BFLOAT16 conversion extension.	<code>on/off</code>
<code>isa-v14</code>	Enables/Disables the vector INT4 load extension.	<code>on/off</code>
<code>crashsave</code>	Enables/Disables the crash debugging state save feature.	<code>on/off</code>
<code>veccfg</code>	Specifies whether the CSR of the vector configuration register (<code>mvec_cfg</code>) is present.	<code>on/off</code>
<code>isa-pp16</code>	Enables/Disables the support on 16-bit push/pop/popret instructions.	<code>on/off</code>

Option	Description	Valid Value
<code>isa-vsih</code>	Enables/Disables the vector small INT handling extension.	on/off
<code>mmsc-cfg-ccachemp-cfg</code>	Enables/Disables the CCACHE/MP information.	on/off
<code>mmsc-cfg-ccache</code>	Enables/Disables the cluster cache.	on/off
<code>codense-ver</code>	Specifies the version number of the Code Dense Extension.	0~3, indicating the opcode encoding for *EXEC.IT instructions as follows: <ul style="list-style-type: none"> • 0: The EXEC.IT encoding is “100-----0-----00”. • 1: The NEXEC.IT encoding is “1001-----00”. • 2~3: Reserved.
<code>isa-vdot</code>	Enables/Disables the vector dot product extension.	on/off
<code>isa-vpfb</code>	Enables/Disables the vector packed FP16 extension.	on/off
<code>stack-protection-only</code>	Enables/Disables the Hardware Stack Protection Only support (without recording).	on/off
<code>mmsc-cfg-rvarch</code>	Specifies whether the CSR of the RVA configuration register (<code>mrvarch_cfg</code>) is present.	on/off
<code>alt-fp-fmt</code>	Enables/Disables the support on the alternative floating point format.	on/off
<code>mmsc-cfg-rvarch2</code>	Specifies whether the CSR of the RVA configuration 2 register (<code>mrvarch_cfg2</code>) is present.	on/off

Option	Description	Valid Value
<code>msc-ext3</code>	Specifies whether the CSR of the misc. configuration register 3 (<code>mmsc_cfg3</code>) is present.	<code>on/off</code>

The following is an example to enable features for CCTL operation and specify the CCTL operation version in the [`andes-config "cpu"`] section.

```
cctl = "on"  
cctl-version = "1"
```



2.3.1.3 Misc. Configuration Register3 (`mmsc_cfg3`)

The following options configure the HVMCSR field of the miscellaneous configuration register3 (`mmsc_cfg3`).

Table 10. Configuration for Misc. Configuration Register3

Option	Description	Valid Value
<code>mmsc-cfg-hvmcsr</code>	Specifies whether the HVM base physical address and the HVM configurations are supported.	on/off

2.3.1.4 RISC-V Architecture (RVA) Configuration Register and Configuration 2 Register (`mrvarch_cfg` / `mrvarch_cfg2`)

The following are options to configure the `Zvqmac`/`Zvlsseg`/`MRVARCH_EXT3` fields of the RVA configuration register (`mrvarch_cfg`) and configuration 2 register (`mrvarch_cfg2`).

Table 11. Configurations for RVA Configuration Register and Configuration 2 Register

Option	Description	Valid Value
<code>isa-zvqmac</code>	Enables/Disables the <code>Zvqmac</code> (Vector Quad-Widening Integer Multiply-Add) support.	on/off
<code>isa-zvlsseg</code>	Enables/Disables the <code>Zvlsseg</code> extension.	on/off
<code>mrvarch-ext3</code>	Specifies whether the CSR of the RVA configuration 3 register (<code>mrvarch_cfg3</code>) is present.	on/off

For example, to enable the `Zvqmac` support for the emulated processor, define as follows in the `[andes-config "cpu"]` section of the configuration file.

```
isa-zvqmac = "on"
```

2.3.1.5 Vector Configuration Register (`mvec_cfg`)

The following options configure the bit fields of the vector configuration register (`vec_cfg`) for an Andes processor enabled with the RISC-V Vector ISA extension.

Table 12. Configurations for Vector Configuration Register

Option	Description	Valid Value	Note
<code>mvec_cfg-minor</code>	Specifies the minor version of the vector processor related to architecture spec and implementation versions.	Decimal	
<code>mvec_cfg-major</code>	Specifies the major version of the vector processor related to architecture spec and implementation versions.	Decimal	
<code>mvec_cfg-dw</code>	Specifies the width of the ALU data computation path.	0~7	Value meanings <ul style="list-style-type: none"> • 0: VLEN • 1: VLEN/2 • 2: VLEN/4 • 3: VLEN/8 • 4 ~ 7: Reserved
<code>mvec_cfg-mw</code>	Specifies the width of the memory data path.	0~7	Value meanings <ul style="list-style-type: none"> • 0: VLEN • 1: VLEN/2 • 2: VLEN/4 • 3: VLEN/8 • 4: 2*VLEN • 5: 4*VLEN • 6 to 7: Reserved
<code>mvec_cfg-misew</code>	Specifies the maximum integer SEW supported.	0~3	Value meanings <ul style="list-style-type: none"> • 0: Reserved • 1: 32 • 2: 64 • 3: Reserved

Option	Description	Valid Value	Note
<code>mvec-cfg-mfsew</code>	Specifies the maximum floating-point SEW supported.	0~3	Value meanings <ul style="list-style-type: none"> • 0: Vector floating-point instructions are not supported. • 1: 32 • 2: 64 • 3: Reserved

The following example gives some bit field definitions for the vector configuration register in the `[andes-config "cpu"]` section.

```
mvec-cfg-minor = "0"
mvec-cfg-major = "1"
mvec-cfg-dw = "0"
mvec-cfg-misew = "2"
mvec-cfg-mfsew = "2"
```

2.3.1.6 Machine Architecture ID Register (`marchid`) and Machine Implementation ID Register (`mimpid`)

The following options configure the machine architecture ID register (`marchid`) and the machine implementation ID register (`mimpid`).

Table 13. Configuration for Machine Architecture ID Register and Machine Implementation ID Register

Option	Description	Valid Value
<code>marchid</code>	Specifies the ID of the base microarchitecture.	An ID number greater than zero (>0)
<code>mimpid</code>	Specifies the version of the implementation.	A version number greater than zero (>0)

3. Known Issues and Limitations

- Not all Andes processors can perform code size optimization with the zcb extension. Specifically, processors implemented with the EXEC.IT instruction, such as A25 and NX45, are unable to decode with the zcb extension due to opcode conflicts. On the other hand, processors with the NEXEC.IT instruction, such as D23 or N225, can work well with the extension.

