Andes ICE Management Software (ICEman) User Manual

Document Number UM067-41

Date Issued 2025-09-09

Copyright © 2012–2025 Andes Technology Corporation. All rights reserved.

Copyright Notice

Copyright © 2012–2025 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
4.1	2025/09/09	1. Updated the document template to V19. 2. Added an option `--keep-halt`. (Chapter 1 and Section 2.1.29) 3. Added a monitor command to enable synchronized trace control on AMP targets that support core grouping. (Section 2.5.2) 4. Supported the AICE-T2 detection when diagnosing the local ICE device. (Appendix II-I)
4.0	2025/04/01	1. Updated the document template to V17. 2. Removed descriptions of ICEman options for V3 targets and AICE. Updated the ICEman help message. 3. Updated supported ICE editions by adding AICE-T2 and removing AICE, AICE-MCU, AICE-MINI, AICE2 and AICE 2-T. (Chapter 1) 4. Added a block diagram to illustrate the trace data flow and the interaction between trace components within the AndesCore NCETRACE200 trace subsystem. Added three trace commands "`monitor nds trace ram-mode [ram|smem|pib]`", "`monitor nds trace ram-size SIZE`" and "`monitor nds configure component COMP BUS ADDRESS`" for V5 targets with AndesCore NCETRACE200. Fixed privilege level errors for bit 1 and 2 of `INST-MODE`. (Section 2.5.2) 5. Added ICEman operations for crash debugging. (Appendix III)
3.9	2024/11/12	Replaced the Cygwin console screenshot with the MSYS2 console screenshot and updated the Target Manager view screenshots. (Appendix II-I)
3.8	2024/06/12	1. Fixed a typo --`target_cfg` to --`target-cfg` in Section 2.1.22. 2. Added monitor commands for triggering trace at a specific context and for using trace buffer in system memory. (Section 2.5.2)
3.7	2023/12/04	1. Listed monitor commands for V5 targets with AndesCore NCETRACE200 trace subsystem. (Section 2.5.2)
3.6	2023/07/26	1. Updated the document template to V15. 2. Added several V5-specific monitor commands. (Section 2.5.1))
3.5	2022/12/26	1. Replaced the option "`--no-CRST-detect`" with "`--no-reset-detect`" and supported the updated option on both V3 and V5 targets. (Chapter 1 and Section 2.1.1) 2. Added an option "`--no-halt-detect`". (Chapter 1 and Section

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

Rev.	Revision Date	Revised Content
		2.1.28) 3. Replaced "monitor nds no_CRST_detect" with "monitor nds no_reset_detect" and supported it on both V3 and V5 targets. Removed monitor commands that enable or disable icache or dcache. (Section 2.5.1)
3.4	2022/05/06	1. Removed custom timing script commands "set_trst" and "clear_trst" for V5 targets. (Section 2.2.4) 2. Changed the font of options in the headings to improve clarity.
3.3	2022/01/04	1. Updated the ICEman help message. (Chapter 1) 2. Updated the descriptions of the -f option. (Section 2.1.6) 3. Added that the "--l2c" option can also be used to enable GDB commands for L2 cache and noted the default base address of L2 cache controller on V5 targets. (Section 2.1.21) 4. Added an option "--detect-2wire". (Section 2.1.27) 5. Added that the script commands "set_trst" and "clear_trst" are not supported by AICE-MICRO R1.1. (Section 2.2.4) 6. Added several ICEman error messages to Appendix I. 7. Enhanced the AICE diagnostic report on AndeSight IDE. (Appendix II-I)
3.2	2021/04/09	1. Changed to specify aice_micro_sdp.cfg as the configuration file for the connection of AICE-MICRO with a target board of 2-wire debug interface. (Section 2.1.9)
3.1	2020/12/16	1. Updated the document template to V14. 2. Added that jtagkey_sdp.cfg must be specified for the connection of AICE-MICRO with a target board of 2-wire debug interface. (Section 2.1.9) 3. Updated the ICE diagnosis report in command line and the IDE UI to diagnose ICE (Appendix II-I)
3.0	2020/07/30	Added an ICEman option "--custom-aice-init" and descriptions of the initialization script to unlock the protection scheme on V5 targets of security architecture. (Chapter 1 and Section 2.3)

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

Rev.	Revision Date	Revised Content
2.9	2020/07/13	1. Added an ICEman option "`--rv[32|64]-bus-only`". (Chapter 1, and Section 2.1.26) 2. Reorganized Andes ICE-specific commands by the supported targets and added description for a number of monitor commands (Section 2.5).
2.8	2020/03/09	Added ICEman options "`--list-device`" and "`--device`". (Chapter 1, Section 2.1.24 and Section 2.1.25)
2.7	2019/11/11	1. Changed the document template to V13. 2. Added an AICE version, AICE-MICRO. (Chapter 1) 3. Added that both FTDI-based Adapter (AICE-MICRO) and AICE-MINI+ support extended TCK frequency range for V5 cores. (Chapter 1)
2.6	2019/03/28	1. Added ICEman options "`--l2c`", "`--smp`", "`--target-cfg`" and "`--halt-on-reset`". (Chapter 1, Section 2.1.21~2.1.23 and Section 2.4) 2. Added that `aice_sdp.cfg` must be specified for the connection of AICE-MINI+ with a target board of 2-wire debug interface. (Section 2.1.9) 3. Introduced the preparation for complex multicore debugging (Section 2.4) 4. Updated the AndeSight UIs to diagnose ICE (Appendix II-I)
2.5	2018/10/08	1. Added an AICE version, AICE-MINI+. (Chapter 1) 2. Added that AICE-MINI+ only supports JTAG clock setting option 10-15. (Chapter 1) 3. Added AICE-MINI+ to descriptions of script format for custom timing.
2.4	2018/07/20	1. Removed the note "For V3 Cores only" for options "`-H, --reset-hold`", "`-l, --custom-srst`", "`-L, --custom-trst`", "`-N, --custom-restart`" and "`-z, --ace-conf`" (Chapter 1 , Section 2.1.8, 2.1.20, and 2.2) 2. Added an ICEman option "`-I, --interface`" (Chapter 1 and Section 2.1.9) 3. Modified the help message and descriptions for "`-C, --`

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

Rev.	Revision Date	Revised Content
		check-times". (Chapter 1 and Section 2.1.4) 4. Removed the note about BSP v4.0 accepts only one coprocessor for a CPU core. (Section 2.1.20) 5. Removed v2/v3m from the supported argument for the option "-z, --target". (Chapter 1) 6. Added descriptions about the custom script format for V5 targets (Section 2.2.4) 7. Removed the note "For V3 core only" for the command "monitor nds mem_access [bus|cpu]" 8. Added "-z [v3|v5]" to the commands to enable the ICEman diagnosis feature and updated the diagnosis screenshot. (Appendix II-I)
2.3	2018/06/06	1. Changed the document template to V12. 2. Replaced "Olimex FTDI USB-TINY-H" with "FTDI". (Chapter 1)
2.2	2017/11/14	1. Replaced the note "For V3 Cores Only" with "For AICE only" for the option "-a, --reset-aice" (Chapter 1) 2. Removed the note "For V3 Cores only" for options "-c, --check-times" and "-x, --diagnosis"; revised the description of "-c" (Chapter 1 , Section 2.1.4 and 2.1.18) 3. Added an ICEman option "--use-sdm" (Chapter 1) 4. Added ICE error messages for V5 targets and added a target warning message. (Appendix I)
2.1	2017/09/25	1. Marked what ICEman options are specific for V3 targets (Chapter 1 and Section 2.1). 2. Noted that only two monitor commands "monitor nds boot_time TIME" and "monitor nds reset_time TIME", are supported on V5 targets. (Section 2.5) 3. Added Olimex FTDI USB-TINY-H to Andes ICEs. (Chapter 1) 4. Noted that the diagnosis function of ICEman works only on V3 targets. (Appendix II-I)
2.0	2017/03/13	1. Updated the help message for the option "-c" (Chapter 1) 2. Added an ICEman option "-f" (Chapter 1 and Section 2.1.6)
1.9	2016/11/22	1. Changed the document template to V11

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

Rev.	Revision Date	Revised Content
		2. Added two AICE versions, AICE2 and AICE2-T (Chapter 1) 3. Added an ICEman option "`-M, --edm-dimb`" 4. Added three script commands for custom timing ("`tck_scan`", "`scan_chain`", and "`write_ctrl 0 TCKDIV`") and two custom script examples
1.8	2016/10/07	1. Added three ICEman error messages "`SW/HW reset-and-hold failed.`" and "`Failed to initialize AICE box!`". (Appendix I) 2. Added two script commands for custom timing: "`delay DELAY_TIME`" and "`t_write_misc TARGET_ID MISC_ADDR MISC_DATA`".
1.7	2016/04/20	Added four breakpoint/watchpoint-related monitor commands
1.6	2015/07/27	1. Added that AICE-MINI can only support JTAG clock setting option 10-15 and the option "`-y`" or "`--idlm`" in the ICEman help message. (Chapter 1) 2. Added the description of the option "`-y`" or "`--idlm`".
1.5	2015/04/10	1. Merged two options "`-H`" and "`-X`" in the help message and revised their descriptions. (Chapter 1, Section 2.1.8 and 2.1.19) 2. Rephrased the description of "`-K`". 3. Noted that descriptions in this document apply to all AICE editions unless otherwise specified. (Chapter 1) 4. Added AICE-MINI to script format descriptions for custom timing. 5. Listed ICEman error and warning messages (Appendix I) 6. Added AICE troubleshooting guide. (Appendix II)
1.4	2015/01/12	1. Removed "`-B`", "`-j`", "`-J`" options and added "`-t`", "`-X`", "`-Z`" options (Chapter 1, Section 2.1.15, and 2.1.19) 2. Noted that "`-a`" is for AICE only. 3. Replaced the option "`-D, --unlimited-log`" with "`-D, --larger-logfile`" (Chapter 1, Section 2.1.5) 4. Changed the default boot time to 5000 milliseconds (Chapter 1, Section 2.1.16) 5. Changed the default $DBGER check times to 300 times (Chapter 1, Section 2.1.4)

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

Rev.	Revision Date	Revised Content
		6. Added a section for AICE(-MCU) specific commands (Section 2.5) 7. Revised the description of "-K".
1.3	2014/04/16	1. Added the description about the ICEman version to Chapter 1 and moved the help message from Chapter 2 to Chapter 1 too. 2. Modified the description of "-a," "-c" and "-j" (Section 2.1.4) 3. Added an option "-K". 4. Added options for custom timing (Section 2.2) 5. Added options to specify passcodes for Andes security processors.
1.2	2013/09/09	1. Added an option to diagnose connectivity problems (Section 2.1.18) 2. Added an option to assign coprocessor configuration files for CPU cores.
1.1	2012/11/02	Added an option "-k, --word-access-mem"
1.0	2012/9/13	Document Creation

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

Table of Contents

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

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

List of Figures

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

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	Underlined	12	Blue

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

1. Introduction

This document describes the option usages of ICE management software (ICEman). Unless otherwise specified, descriptions in this document apply to all Andes ICE editions, including AICE-T2, AICE-MICRO, AICE-MINI+ and FTDI.

Before working with ICEman, check its version first by issuing `$ ./ICEman -v`.

```
$ ./iceman -v
Andes ICEman (OpenOCD) 5.4.0-g67e2349 (2025-03-12-06:49)
Copyright (C) 2007-2024 Andes Technology Corporation
Open On-Chip Debugger 0.11.0+dev-gb8f647b (2025-03-12-06:47)
```

For a complete list of ICEman options, obtain it by issuing `$ ./ICEman -h`.

```
Usage:
ICEman --port start_port_number[:end_port_number] [--help]
-A, --no-reset-detect: No reset detection in debug session
-b, --bport:       Socket port number for burner connection
                   (default: 2354)
-c, --clock:       Specify JTAG clock setting
     Usage: -c num
         num should be the following:
         0: 30 MHz
         1: 15 MHz
         2: 7.5 MHz
         3: 3.75 MHz
         4: 1.875 MHz
         5: 909.091 KHz
         6: 461.538 KHz
         7: 232.558 KHz
         10: 10 MHz
         11: 6 MHz
         12: 3 MHz
         13: 1.5 MHz
         14: 750 KHz
         15: 375 KHz

      FTDI-based Adapter (AICE-MICRO) and AICE-MINI+ support extended TCK
      frequency range.
     The TCK frequency of AICE-MICRO ranges from 1KHz to 10MHz and that
      of AICE-MINI+ ranges from 1KHz to 5.6MHz.
```

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

```
        Usage: -c <clock range>KHz/MHz

-C, --check-times:    Second to check DTM
           (default: 3 seconds)
     Example:
        1. -C 100 to check 100 millisecond
        2. -C 100s or -C 100S to check 100 seconds

-D, --larger-logfile:  The maximum size of the log file is 1MBx2. The
size is increased to 512MBx2 with this option.
-f, --log-output:  output path for config and log files
-h, --help:    The usage is for ICEman
-H, --reset-hold:   Reset-and-hold while ICEman startup
-I, --interface:   Specify an interface config file in ice/interface.
-l, --custom-srst: Use custom script to do SRST
-L, --custom-trst: Use custom script to do TRST
-N, --custom-restart:  Use custom script to do RESET-HOLD
-o, --reset-time:  Reset time of reset-and-hold (milliseconds)
        (default: 1000 milliseconds)
-p, --port:    Socket port number for gdb connection
-s, --source:      Show commit ID of this version
-S, --stop-seq:    Specify the SOC device operation sequence while CPU
stop
-R, --resume-seq:  Specify the SOC device operation sequence before CPU
resume
     Usage: --stop-seq A1:D1[:M1],A2:D2[:M2]
            --resume-seq A3:D3[:M3],A2:rst
            A*:address, D*:data, [:M*]:optional mask, rst:restore
            A*,D*,[M*] should be hex as following example

     Example: --stop-seq 0x500000:0x80,0x600000:0x20
              --resume-seq 0x500000:0x80,0x600000:rst

-t, --tport:       Socket port number for Telnet connection
-T, --boot-time:   Boot time of target board (milliseconds)
        (default: 5000 milliseconds)
-v, --version:     Version of ICEman
-x, --diagnosis:   Diagnose connectivity issue
     Usage: --diagnosis[=address]
-X, --uncnd-reset-hold:   Unconditional Reset-and-hold while ICEman
startup (This implies -H)
-z, --ace-conf:    Specify ACE file on each core
     Usage: --ace-conf <core#id>=<ace_conf>[,<core#id>=<ace_conf>]*
```

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

```
        Example: --ace-conf core0=core0.aceconf,core1=core1.aceconf

    --l2c[=Base Address]:  Specify the base address of L2C and enable GDB
    commands for L2C.
    --target-cfg:      Specify the CPU configuration file for a complex
    multicore system
    --smp:             Enable SMP mode for multi-cores
    --halt-on-reset:   Enable/Disable halt-on-reset functionality
    --list-device:     List all connected devices
    --device <device-id>:   Connect selected device directly
    --custom-aice-init:    Use custom script to do initialization process
    --rv32-bus-only/--rv64-bus-only:  Read memory from system bus wihout
    checking CPU status
    --detect-2wire:    If JTAG scan chain interrogation (the 4-wire mode)
    fails, retry the connection with the SDP (2-wire) mode.
    --no-halt-detect:  No halt detection in debug session
    --keep-halt:  Halt the target at the start of examination and keep it
    halted
```

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

2. Detailed descriptions of ICEman options

2.1. General options

2.1.1. -A , --no-reset-detect

This option is to make ICEman skip the reset checking. It is used when the running program will reset the core but you don't want the GDB session to be closed by ICEman.

2.1.2. -b, --bport

This option is to specify a socket port number for IntelJ3/MXIC burner program to connect (default value is 2354).

2.1.3. -c, --clock

This option is to change the JTAG clock.

2.1.4. -C, --check-times

This option enables ICEman to check the DTM status for V5 cores by reading `$dmstatus`. It specifies the duration of the check, with a default value of 3 seconds.

For V5 cores, use a numeric argument to specify how many milliseconds that ICEman checks `$dmstatus` and a numeric argument suffixed with "`s`" or "`S`" to specify the duration in seconds. For example, "`-C 100`" enables ICEman to check `$dmstatus` for 100 milliseconds whereas "`-C 100s`" or "`-C 100S`" is for ICEman to check `$dmstatus` for 100 seconds.

2.1.5. -D, --larger-logfile

By default, ICEman uses a circular buffer to record debug information and output two debug log files of 1 MB at maximum when it's terminated. To store more debug information, you can use this option to increase the maximum size of each log file to 512 MB. Note that the performance of ICEman will be downgraded when this option is specified.

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

2.1.6. -f, --log-output

This option is to specify a save location for ICEman config files and logs. For command line users who don't have a write permission to the default folder of ICEman config files or logs (i.e., ANDESIGHT_ROOT/ice), such as those in a multi-user environment, they may encounter problems launching ICEman. To ensure a successful launch of ICEman, these users can use this command to specify another folder where they have full permissions for ICEman config files and logs to be saved in.

2.1.7. -h, --help

This option is to show all options supported by ICEman.

2.1.8. -H, --reset-hold

This option causes ICEman to perform the "reset and hold" operation when ICEman starts. The "reset and hold" operation is an AndesCore hardware mechanism that resets the target system and makes the target processor stop before executing the first instruction. This option is for debugging boot code problems where the target system crashes/hangs after running the first few instructions in the boot code. With this option, the processor is being stopped at the cleanly initialized state and the problems can be analyzed under full debugger control.

2.1.9. -I, --interface

This option allows you to configure the adapter interface by specifying a configuration file in the /ice/interface folder. To connect an AICE-MINI+ to a target board of 2-wire debug interface, the configuration file aice_sdp.cfg must be specified; to connect an AICE-MICRO to a target board of 2-wire debug interface, specify the configuration file aice_micro_sdp.cfg instead.

2.1.10. -o, --reset-time

This option is to specify the time (milliseconds) for which ICE will sleep to wait for the completion of the reset after reset-and-hold is issued. The default value is 1000 milliseconds.

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

2.1.11. -p, --port

This option is to specify a socket port number for GDB connection.

2.1.12. -s, --source

This option is to print the git commit ID of the ICEman in use.

2.1.13. -S, --stop-seq

If the target board needs some operations after the core is halted, use this option to add these operations to the sequence.

2.1.14. -R, --resume-seq

If the target board needs some operations before the core free runs, use this option to add these operations to the sequence. These operations usually are the reverse actions of --stop-seq.

2.1.15. -t, --tport

This option is to specify the socket port number for Telnet connection.

2.1.16. -T, --boot-time

This option is to specify the time (milliseconds) for which ICE will sleep to wait for the completion of the reset after SRST (System Reset) is issued. The default value is 5000 milliseconds.

2.1.17. -v, --version

This option is to show the version information of the ICEman in use.

2.1.18. -x, --diagnosis

When the debugger is not executed as expected, use this option to run a prerequisite check for the debugger. You can optionally specify a writable address for this option. If a writable

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

address is specified such as `-x0x45` or `--diagnosis=0x45`, ICEman will test the memory access.

2.1.19. `-X`, `--uncnd-reset-hold`

This option enables ICEman to perform the "reset-and-hold" operation when ICEman starts. The "reset-hold" operation is an AndesCore hardware mechanism that resets the target system and makes the target processor stop before executing the first instruction.

2.1.20. `-z`, `--ace-conf`

This option is to assign an ACE (Andes Custom Extension™) configuration file (`.aceconf`) for specific coprocessor. Its usage and example are like below:

Usage: `--ace-conf CORE_ID=ACE_CONF[,CORE_ID=ACE_CONF]*`

Example: `--ace-conf core0=core0.aceconf,core1=core1.aceconf`

For more details about ACE configuration files, see *Andes COPILOT Manual*.

2.1.21. `--l2c`

This option is to specify the base address of the L2 cache controller and enable GDB's monitor commands for L2 cache. If the address is not specified explicitly, it is default set to 0xe0500000 on Andes V5 targets.

2.1.22. `--smp` (for multicore systems of pure SMP cores)

This option is to enable SMP (Symmetric Multiprocessing) capability for V5 multiple cores so that they can be debugged as a grouped entity. It is used for multicore systems incorporating pure V5 cores that are capable of symmetric multiprocessing. For debugging on a complex multicore system, you need to use the option "`--target-cfg`" to specify a corresponding CPU configuration file. For more details on the specific ICEman option and the preparation for complex multicore debugging, see Section 2.4.

2.1.23. `--halt-on-reset`

This option is to enable a core to halt and enter debug mode right after it is reset.

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

2.1.24. --list-device

This option is to list all supported ICE devices on your PC network. The device list displays the device number, ID and description along with bus and port information for each ICE device.

The figure below gives an example of how to display the device list. With the IDs and descriptions shown in the device list, the device "0" or "1" is known as an AICE-MICRO or Corvette-FI, the device "3" is an AICE-MINI+ and the device "2" is an Andes ICE of another edition.

```
$ ./ICEman.exe  --list-devic
Andes ICEman v4.6.1 (OpenOCD) BUILD_ID: 2020012211
Burner listens on 2354
Telnet port: 4445
TCL port: 6666

List of Devices:
        #0 Bus 001 Port 002 Device 010: ID 0403:6010 AndeShape AICE-MICRO / FTDI USB device
        #1 Bus 001 Port 003 Device 007: ID 0403:6010 AndeShape AICE-MICRO / FTDI USB device
        #2 Bus 001 Port 004 Device 008: ID 1cfc:0000 AndeShape AICE/AICE-MCU/AICE-MINI/AICE2
        #3 Bus 001 Port 004 Device 009: ID 1cfc:0001 AndeShape AICE-MINI+
```

2.1.25. --device

This option is to specify an ICE device with its device number and have it connected to ICEman. To look up for the device number of a desired ICE device on your network, use "--list-device" to show the device list.

The following example is to select the AICE-MINI+ in the example of Section 2.1.24 and connect it to ICEman.

```
$ ./ICEman.exe -p 1111 -Z v5 --device 3
Andes ICEman v4.6.1 (OpenOCD) BUILD_ID: 2020012211
Burner listens on 2354
Telnet port: 4445
TCL port: 6666
Open On-Chip Debugger 0.10.0+dev-g066028a (2020-01-22-10:58)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Andes AICE-MINI+
JTAG frequency 10.000 MHz
There is 1 core in tap
The core #0 listens on 1111.
ICEman is ready to use.
```

2.1.26. --rv32-bus-only/--rv64-bus-only

When a CPU hangs, these options enable ICEman to access system bus and read memory contents through the bus without checking the CPU status. The options for RV32 and RV64 targets are different. To ensure the right operation on the system bus, it is required to select

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

the option that corresponds to your target (i.e., RV32 or RV64). Also, note that when this option is applied, the debugger can only be used to access the system memory and thus will be unable to perform other actions normally.

2.1.27. --detect-2wire

This option is to use the SDP mode (2 wires) for ICEman connection if the JDP mode (4 wires) fails.

2.1.28. --no-halt-detect

This option is to make ICEman skip the halt checking and not show the message "Unable to halt hart..." for a core that has been reset by the running program.

2.1.29. --keep-halt

This option enables a core to halt at the beginning of target examination and remain in the halted state.

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

2.2. Options for custom timing

When the default reset sequence timing of ICE does not fit your target system, you can customize your timing through ICEman options and a script file. In the following subsections, Section 2.2.1 to 2.2.3 list options to configure custom timing, and Section 2.2.4 introduces the format of custom scripts . For more information about the default reset timing, see the associated AICE user manuals.

2.2.1. `-l`, `--custom-srst`

If your target board needs custom timing for SRST (System Reset), use this option to specify a script for the reset timing.

2.2.2. `-L`, `--custom-trst`

If your target board needs custom timing for TRST (JTAG TAP Reset), use this option to specify a script for the reset timing.

2.2.3. `-N`, `--custom-restart`

If your target board needs custom timing for RESET-HOLD (Debug-on-Reset), use this option to specify a script for the reset timing.

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

2.2.4. Format of custom timing script

The scripts for custom timing must be plain ASCII files using the following commands:

Command	Operation
`set_srst DELAY_TIME`	Drive SRSTn to LOW
`clear_srst DELAY_TIME`	Drive SRSTn to HIGH
`set_dbgi DELAY_TIME`	Send a DBGI command
`clear_dbgi DELAY_TIME`	Remove the sent DBGI command
`delay DELAY_TIME`	Pause for `DELAY_TIME` millisecond(s) after executing this command

Custom script example: Pausing after executing commands

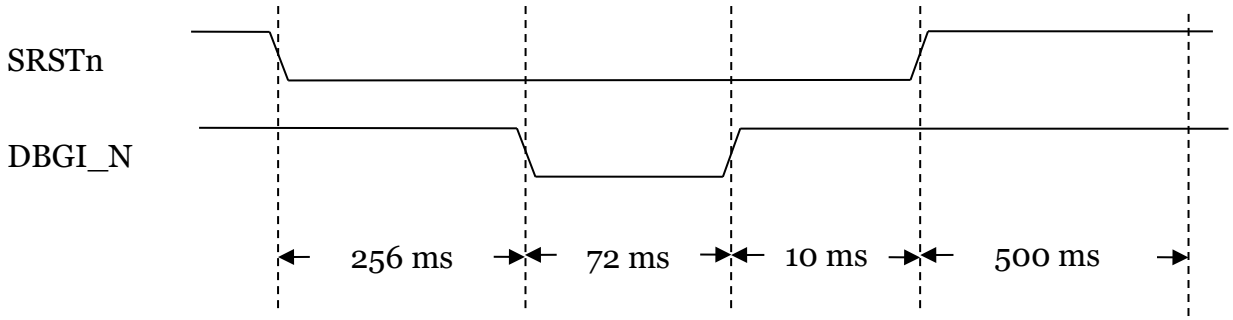
Script commands involved with interface signals "SRSTn", "DBGI_N" or "TMSC" carry a `DELAY_TIME` argument. The argument is used to specify the delay time in milliseconds after these commands are executed. Here is a script example (`example_custom.tcl`) using these commands:

```
set_srst 256     #Drive SRSTn to LOW and pause for 256 milliseconds
set_dbgi 72      #Send a DBGI command and pause for 72 milliseconds
clear_dbgi 10    #Remove the DBGI command and pause for 10 milliseconds
clear_srst 500   #Drive SRSTn to HIGH and pause for 500 milliseconds
```

With the script, you can specify the RESET-HOLD timing as follows:

```
--custom-restart example_custom.tcl
```

Then, after doing RESET-HOLD, AICE will generate the following waveform:



Likewise, you can customize the SRST timing and TRST timing using the options `--custom-srst` and `--custom-trst` respectively.

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

2.3. Options for Andes V5 cores of security architecture

The hardware debug security architecture (currently, NCEDBGLOCK100) blocks the JTAG signals from external debuggers. To unlock the protection scheme, you can use the ICEman option "`--custom-aice-init`" to specify an initialization script defined to drive certain patterns of debug interface signals.

2.3.1. `--custom-aice-init`

This ICEman option is to specify a script that defines an initialization process from AICE to your target board.

2.3.2. Format of custom initialization script to write patterns on target boards

The initialization script is to enable AICE to write certain signal patterns on target boards. You will need to use the following command to drive debug interface pins according to the patterns you specify:

```
ftdi write_pins NIBBLE
```

where **NIBBLE** is a string of up to 2044 numeric values that represent a sequence of signal patterns on the debug interfaces. Each value in the string refers to a state of pins for the serial debug port (2-wire) or JTAG debug port (5-wire). The following table lists the pattern values and the pin states they represent. Each pin state is realized by signal values on specific pins of debug ports. A signal value of 0 is to drive the signal LOW on a debug pin and a signal value of 1 is to drive it HIGH.

Pattern value in **NIBBLE**	Pins on Serial Debug Port (2-wire)		Pins on JTAG Debug Port (5-wire)		
	TMSC	TCKC	TDI	TMS	TCK
0	0	0	0	0	0
1	0	1	0	0	1
2	1	0	0	1	0
3	1	1	0	1	1
4	Undefined		1	0	0
5			1	0	1
6			1	1	0
7			1	1	1
Others			Undefined		

The script enables debug interface signals to perform the specified patterns accordingly. After all the specified patterns are carried out, the target board with security architecture is

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

initialized and the communication between the external debugger and processor debug interfaces is thus unlocked.

Custom script example:

Here is an initialization script example "`initial_script.tcl`" that uses the command "`write_pins`" to drive the debug interface signals TDI, TMS and TCK on a JTAG debug port.

```
#INFO: Code format - binary
#INFO: Total write_pins nibbles 47
ftdi write_pins 033223322332233201102332233201100110233201102 30
```

To execute the script, use the ICEman option "`--custom-aice-init`" to specify the script as follows:

```
--custom-aice-init initial_script.tcl
```

After the script is executed, AICE will generate the following waveform to indicate that the initialization is complete.

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

2.4. Options for complex multicore debugging

The ICEman option "`--target-cfg`" is to specify a CPU configuration file that helps OpenOCD (Open On-chip Debugger) to identify the JTAG implementation in a complex multicore system and enable multicore debugging.

For a multicore system of pure V3 cores, the debugging has been handled automatically in the background. For a multicore system of pure V5 SMP (Symmetric Multiprocessing) cores, just use the option "`--smp`" to enable the SMP capability for the cores. However, for a heterogeneous multicore system (i.e., incorporating both V3 and V5 cores) or a V5 multicore system containing both SMP and AMP (Asymmetric Multiprocessing) cores, you need to ensure that the JTAG implementation is configured properly, write a CPU configuration file from scratch and use the option "`--target-cfg`" to specify the file.

2.4.1. JTAG implementation in complex multicore systems

A complex multicore system needs to have a serial configuration of on-chip debug resources following the rules below:

1. Cores on the system interface with the debug tool through TAPs (Test Access Ports). Each TAP connects to a debug module for one or multiple V5 cores, an EDM (Embedded Debug Module) within a V3 core, or a SDM (System Debug Module) that communicates with several V3 cores through their EDMs.
2. TAPs are chained in a serial manner which the output (TDO) of the previous TAP is connected to the input (TDI) of the next TAP. TAP indexing starts with 0 from the one closest to the TDO pin of the debugger.
3. A GDB debug target can be created for a set of SMP cores or a single core without SMP support. Each GDB debug target has a dedicated GDB port and refers to a specific TAP.
4. While a TAP can be referred by one or multiple targets, it only allows cores of the same architecture. Cores of different architecture must communicate with the debug tool through separate TAPs.
5. A debug target of SMP cores must have a core assigned to the first hardware thread (hart 0) of its debug module.

The following illustrates the JTAG implementation in a complex system of eight cores. Four TAPs are used to communicate with the debugger in the configuration. TAP 0 is referred by a

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

GDB debug target of two V5 SMP cores, TAP 1 is referred by a GDB debug target of single V3 core, TAP 2 by two V3 debug targets via SDM, and TAP 3 by two debug targets – one comprising two V5 SMP cores and the other a single V5 core.

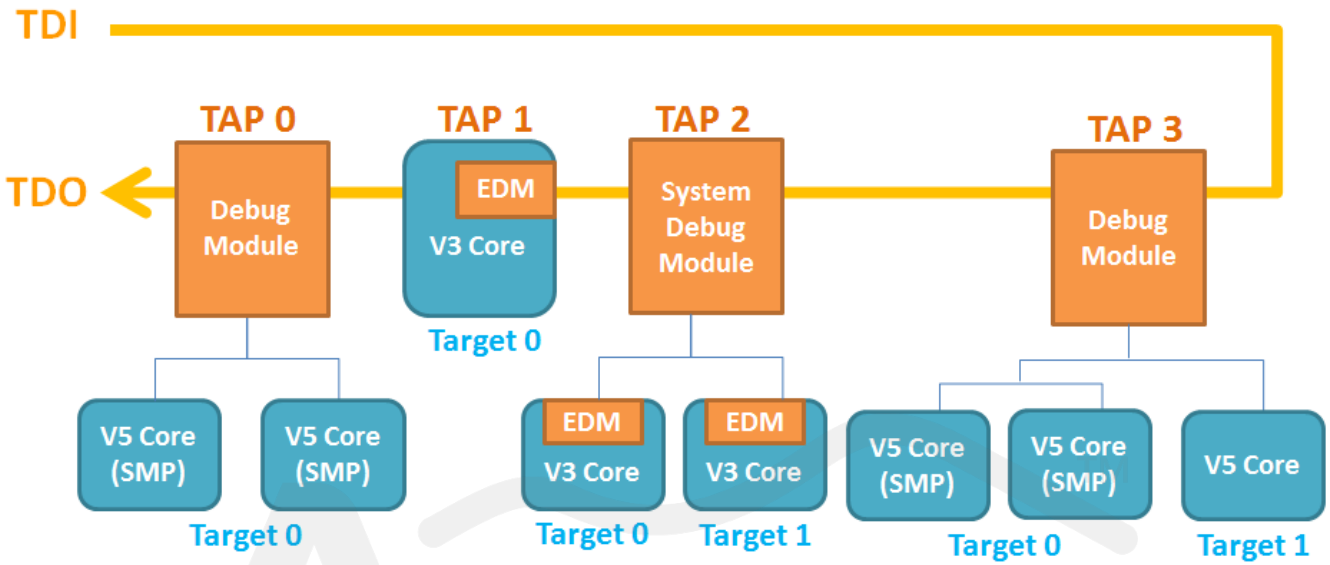


Figure 1. JTAG implementation in a complex multicore system

2.4.2. CPU configuration file for a complex multicore system

OpenOCD requires a CPU configuration file to commence debugging on a complex system having a mix of V3 and V5 cores or a V5 multicore system incorporating both SMP and AMP cores. The CPU configuration file must be a text file that uses a three-column table to describe the JTAG implementation in the multicore system. Its template is shown below.

```
                        Arch            core no.
  tapX_target_Y    [v3|v3_sdm|v5]         NUM
```

In the template, the first column uses the form "`tapX_target_Y`" to index TAPs on the JTAG scan chain and GDB debug targets that refer to each TAP. The second column specifies the architecture of the GDB debug target and if a SDM is involved. For a V3 core connecting to a TAP through SDM, make sure you specify `v3_sdm` in this column. The third specifies the number of cores each GDB debug target consists of. If a GDB debug target comprises more than one core (i.e., `NUM` > 1), it must come with SMP support.

The following example is the CPU configuration file "`user_target_tbl`" that defines the

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

multicore configuration in Figure 1.

```
                    Arch        core no.
tap0_target_0       v5           2
tap1_target_0       v3           1
tap2_target_0       v3_sdm       1
tap2_target_1       v3_sdm       1
tap3_target_0       v5           2
tap3_target_1       v5           1
```

To enable multicore debugging on the complex system in Figure 1, you can use the ICEman option "`--target-cfg`" to initiate OpenOCD as follows:

```
./ICEman.exe -p PORT_NUMBER -D --target-cfg user_target_tbl
```

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

2.5. Andes ICE-specific commands

After ICEman is launched, you can send a request to ICEman directly using GDB's "monitor" commands. This is because Andes ICEs have a proprietary monitor interface. With parameters specified, the "monitor" commands can request ICEman to return results. The following list the Andes ICE-specific commands by their supported targets.

2.5.1. Monitor commands supported by general V5 targets

- `monitor mdd ADDRESS [COUNT]`
- `monitor mdw ADDRESS [COUNT]`
- `monitor mdh ADDRESS [COUNT]`
- `monitor mdb ADDRESS [COUNT]`

  These commands display contents of the address `ADDRESS`, as 64-bit doublewords (`mdd`), 32-bit words (`mdw`), 16-bit halfwords (`mdh`), or 8-bit bytes (`mdb`). The optional parameter `COUNT` is to specify the number of units to be displayed.

- `monitor mwd ADDRESS DOUBLEWORD`
- `monitor mww ADDRESS WORD`
- `monitor mwh ADDRESS HALFWORD`
- `monitor mwb ADDRESS BYTE`

  These commands write the data of the `DOUBLEWORD` (64 bits), `WORD` (32 bits), `HALFWORD` (16 bits), or `BYTE` (8-bit) value, at the specified address `ADDRESS`.

- `monitor nds auto_break [on|off]`

  If software breakpoints can't be inserted on a target, use this command to automatically convert them into hardware breakpoints. The default value is "`on`".

- `monitor nds boot_time TIME`

  This command sets the waiting period (in milliseconds) after system resets.

- `monitor nds bp [ADDRESS LENGTH]`

  With no parameters, this command lists all hardware breakpoints. Otherwise, it sets a hardware breakpoint on code execution starting from the specified address for the specified length (bytes). Note that ICEman only takes the parameter `ADDRESS` as the

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

identifier. Thus, if this command is used several times for setting breakpoints from the same address for different lengths, only the first command will take effect.

- `monitor nds [cache|icache|dcache] invalidate`

  This command invalidates I&D cache, I-cache or D-cache.

- `monitor nds [icache|dcache] dump [all FILENAME|va ADDRESS]`

  This command dumps all entries from the I/D cache to a file or dump the I/D cache entry at the address **ADDRESS** to a console.

- `monitor nds mem_access [bus|cpu]`

  This command decides whether the memory access is through bus or CPU. Specify the parameter "cpu" to load/store memory through CPU or "bus" to use DMA for memory access. If there is no data cache, always use the bus mode.

- `monitor nds no_reset_detect [1|0]`

  When the parameter "1" is specified, no reset detection will be performed in a debug session.

- `monitor nds rbp ADDRESS`

  This command removes a hardware breakpoint from the specified address.

- `monitor nds reset_time TIME`

  This command sets the waiting period (in milliseconds) for the target to restart.

- `monitor nds rwp ADDRESS`

  This command removes the data watchpoint from the specified address.

- `monitor nds va [on|off]`

  This command converts virtual addresses to physical addresses when reading or writing memory.

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

- `monitor nds wp [ADDRESS LENGTH[(r|w|a) [VALUE [MASK]]]]`

  With no parameters, this command lists all active watchpoints. Otherwise, it sets a data watchpoint on data starting from the specified address for the specified length (bytes). The followings list the usages of the rest parameters:

  - `r` determines a read watchpoint.
  - `w` determines a write watchpoint.
  - `a` determines an access watchpoint.
  - `VALUE` determines if the watchpoint should be triggered. It can be masked first using the parameter "`MASK`".
  - `MASK` marks the fields that need to be ignored.

  NOTE

  To use breakpoint/watchpoint-related monitor commands listed below

  - `monitor nds bp [ADDRESS LENGTH]`
  - `monitor nds rbp ADDRESS`
  - `monitor nds wp [ADDRESS LENGTH [(r|w|a) [VALUE [MASK]]]]`
  - `monitor nds rwp ADDRESS`

  Please take note of the following:

  1. Each of these commands consumes a hardware non-simple breakpoint.
  2. Before using these monitor commands, remove all HW breakpoints and watchpoints to ensure sufficient HW resources.

- `monitor nds count_to_check_dm [TIME|COUNT]`

  This command checks the $DMI status for the specified duration (in seconds) or retry times. For example, "`monitor nds count_to_check_dm 10s`" is to enable the $DMI status to be checked for 10 seconds and "`monitor nds count_to_check_dm 10`" allows the $DMI status to be checked for 10 times.

- `monitor nds dmi_busy_delay_count COUNT`

  This command sets the number of idle cycles after each operation.

- `monitor nds stepie [on|off]`

  This command requests ICE to step into the interrupt handler during single-step execution.

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

- `monitor nds info`

  This command presents the target information detected by ICEman, such as the following.

```
[tap0_target_0] Found 8 triggers
hart.xlen              64
hart.trigger_count      8
target.memory.read_while_running8    1
target.memory.write_while_running8   1
target.memory.read_while_running16   1
target.memory.write_while_running16  1
target.memory.read_while_running32   1
target.memory.write_while_running32  1
target.memory.read_while_running64   1
target.memory.write_while_running64  1
target.memory.read_while_running128  1
target.memory.write_while_running128 1
dm.abits               7
dm.progbufsize         8
dm.sbversion           1
dm.sbasize            32
dm.sbaccess128         1
dm.sbaccess64          1
dm.sbaccess32          1
dm.sbaccess16          1
dm.sbaccess8           1
dm.authenticated       1
```

- `monitor nds repeat_read COUNT ADDRESS [size=4]`

  Read the value at the specified address repeatedly. The access size in memory region is 4 bytes by default.

- `monitor nds set_command_timeout_sec SECONDS`

  Specify the timeout duration (in seconds) for each command.

- `monitor nds set_reset_timeout_sec SECONDS`

  Specify the duration (in seconds) of the wall-clock timeout after the reset signal is deasserted.

- `monitor nds dmi_read DMI_ADDRESS`

  Execute a 32-bit DMI read operation at the specified address and return the corresponding value. Please execute the command with caution as it may disrupt the debugging.

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

- `monitor nds dmi_write DMI_ADDRESS VALUE`

  Execute a 32-bit DMI write operation of the specified value at the given address. Please execute the command with caution as it may disrupt the debugging.

- `monitor nds resume_order [normal|reversed]`

  Specify the sequence in which harts are resumed when the “hasel” setting that defines the currently selected harts is not available for your debug module. The normal order is from the hart with the lowest index to the one with the highest index, while the reversed order is from the highest hart index to the lowest. For more about the hasel setting, see RISC-V External Debug Support.

- `monitor nds set_ebreakm [on|off]`
- `monitor nds set_ebreaks [on|off]`
- `monitor nds set_ebreaku [on|off]`

  Enable or disable the ebreakm/ebreaks/ebreaku setting for the Debug Control and Status register (dcsr). These settings are enabled by default. When they are disabled, ebreak instructions will not be intercepted by ICEman in M-mode/S-mode/U-mode and software breakpoints will be affected. For more about ebreakm/ebreaks/ebreaku settings, see RISC-V External Debug Support.

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

2.5.2. Monitor commands supported by V5 targets with AndesCore NCETRACE200 trace subsystem

This section introduces commands for controlling Andes RISC-V trace components within the AndesCore NCETRACE200 trace subsystem, including the NCETRACE200 trace encoder, the NCETMUX200 trace multiplexer, and the NCETBUF200 trace buffer. Figure 2 below illustrates the debug and trace data flow within the trace subsystem and demonstrates the interaction of trace components. For more details about respective trace components, see TG RISC-V Nexus Trace and the data sheets for AndesCore NCETRACE20, NCETMUX200, and NCETBUF200.

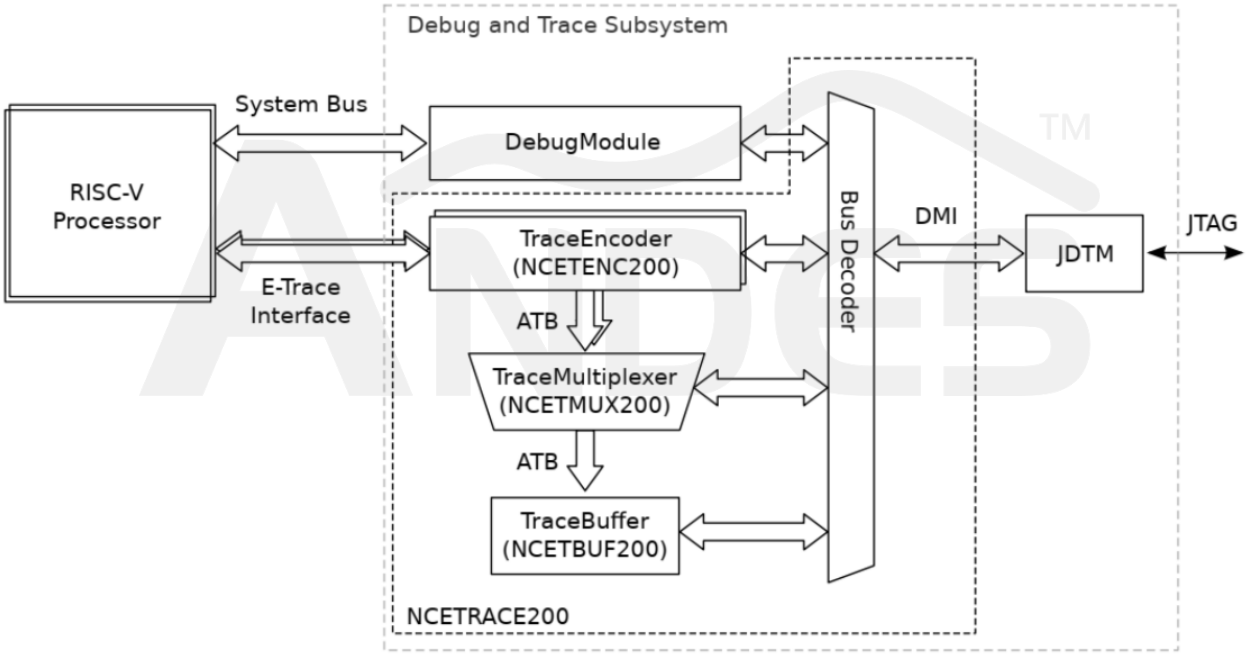


Figure 2. Debug and trace data flow within the NCETRACE200 trace subsystem

- `monitor nds trace [on|off]`

  When set to `on`, this command enables the trace encoder to record program execution from the current position until the trace buffer is full. When set to `off`, it disables the trace encoder.

- `monitor nds trace from`

  This command is equivalent to "`monitor nds trace on`", which enables the trace encoder to record program execution from the current position until the trace buffer is full.

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

- `monitor nds trace to`

  This command enables the trace encoder to record program execution from the current position until a breakpoint is encountered.

- `monitor nds trace dump-trace-file FILENAME`

  This command dumps recorded information from the trace buffer to the file `FILENAME`.

- `monitor nds trace sync-period PERIOD`

  This command specifies the maximum interval between synchronization messages. For more information, see the description of the register `trTeInstSyncMax` in *AndesCore NCETENC200 Data Sheet*.

- `monitor nds trace src-field [1|0]`

  When set to 1, this command includes the trace source field in each trace message to indicate its originating trace encoder.

- `monitor nds trace trace-hart ACTIVE_SRC`

  This command enables data transfer from activated trace sources to the trace buffer. With each bit indicating a specific trace source, `ACTIVE_SRC` designates the activated trace sources. For more information, see *AndesCore NCETMUX200 Data Sheet*.

- `monitor nds trace inst-mode MODE`

  This command specifies the main instruction trace generation mode. Currently, it only supports values 0, 3, and 6 (default), and a higher value implies a higher compression rate on trace data. Note that the `MODE` value must not be modified during the tracing process.

- `monitor nds trace timestamp [1|0]`

  When set to 1, this command enables the Timestamp Unit and includes timestamp information in all trace messages.

- `monitor nds trace match-inst INST-MODE`

  This command enables the recording of instructions at specified privilege levels. By default, all instructions are recorded regardless of privilege levels. However, you can use

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

INST-MODE to restrict the recording to specific privilege levels. The table below lists bits in INST-MODE and the corresponding privilege levels for the recording. For example, bit 0 represents the recording of U-mode instructions and bit 3 corresponds to the recording of M-mode instructions. For more information, see the description of the register trTeFilteriMatchInst in *AndesCore NCETENC200 Data Sheet*.

Bit	3	2	1	0
Privilege level	M-mode	Hypervisor mode	Supervisor mode	User mode

- monitor nds trace match-context SCONTEXT_MATCH

  This command enables the recording of instructions when the supervisor context register (SCONTEXT) is at a specific value. That is, it compares the SCONTEXT value with the specified SCONTEXT_MATCH value and triggers the recording of executed instructions when the two values match. For information about the context matching scheme, see the description of the register trTeFilteriMatchValueImpdef in *AndesCore NCETENC200 Data Sheet*.

- monitor nds trace smem-base ADDRESS

  This command enables the trace encoder to compress trace messages into fixed-width trace words (typically in bytes) and store them subsequently in an area of system memory allocated for trace usage at the specified address. Note that this feature is not supported by all the platforms. For further information, see the description of the register trRamControl in *AndesCore NCETBUF200 Data Sheet*.

- monitor nds trace smem-size SIZE

  This command specifies the size, in bytes, of the trace buffer in system memory that will be used to store the traced data. Note that not all platforms support this mode. For further information, see the description of the register trRamControl in *AndesCore NCETBUF200 Data Sheet*.

- monitor nds trace ram-mode MODE

  This command specifies the storage destination for the trace data. The available destination options for the parameter MODE include

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

- `ram`, which stores trace data in the SRAM configured in the AndesCore NCETBUF200 trace buffer. This option provides efficient storage with moderate capacity and performance. It is the default setting for the parameter `MODE` and is suitable for most general use cases.
- `smem`, which stores trace data in a designated area of system memory. The base address of this area must be specified using the command "`monitor nds trace smem-base ADDRESS`", as described earlier in this section.
- `pib`, which sends and stores trace data off-chip to external trace probes through I/O pins (i.e., the PIB sink). This option is only supported when the AICE-T2 is used. For further information about the storage destination, see the description of the register `trPibControl` in *AndesCore NCETBUF200 Data Sheet*.

NOTE

1. Not all the three storage destinations are available on Andes platforms. For details about platform-specific trace storage support, see *AndesCore NCETBUF200 Data Sheet*.
2. It is not allowed to change the trace data storage location while tracing a program.

- `monitor nds trace ram-size SIZE`

  This command specifies the buffer size in bytes for storing trace data in the destination specified by the command "`monitor nds trace ram-mode MODE`". If `smem` (system memory) is set as the storage destination, this command is equivalent to "`monitor nds trace smem-size SIZE`" described earlier.

- `monitor nds configure component COMP BUS ADDRESS`

  This command specifies the bus type for a specific trace component within the NCETRACE200 trace subsystem and sets its address on the selected bus. Its parameters include

  - `COMP` selects a trace component from the following
    - `ncetenc`, which indicates the trace encoder on Hart0. When specified, the bus type and address of trace encoders on other harts are determined accordingly.
    - `ncetmux`, which indicates the trace multiplexer for the trace encoder.
    - `ncetbuf`, which indicates the trace buffer.
  - `BUS` specifies the bus type to which the selected trace component is configured. The

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

available options for this parameter include

- `dmi` specifies the Debug Module Interface (DMI). Trace components on the DMI can only be accessed via ICEman through JTAG.
- `apb` specifies the system bus. Trace components on the system bus can be accessed by either ICEman or the CPU.

- `ADDRESS` specifies the address on the specified bus for the selected trace component.

[Example Usage]

Given the memory map for the AndesCore NCETRACE200 trace subsystem as follows,

Start Address	End Address	Description
0xC4290000	0xC429FFFF	Reserved
0xF2300000	0xF2300FFF	Harto Trace Encoder (4K)
0xF2301000	0xF2301FFF	Hart1 Trace Encoder (4K)
0xF2302000	0xF2302FFF	Hart2 Trace Encoder (4K)
0xF2303000	0xF2303FFF	Hart3 Trace Encoder (4K)
0xF2304000	0xF2304FFF	Hart4 Trace Encoder (4K)
0xF2305000	0xF2305FFF	Hart5 Trace Encoder (4K)
0xF2306000	0xF2306FFF	Hart6 Trace Encoder (4K)
0xF2307000	0xF2307FFF	Hart7 Trace Encoder (4K)
0xF2308000	0xF2308FFF	Reserved
0xF2309000	0xF2309FFF	Trace Buffer (8K)
0xF2310000	0xF2310FFF	Trace Multiplexer for Trace Encoder (4K)
0xF2311000	0xF2311FFF	Reserved

You can edit the configuration file `nds32_user.cfg` under `ANDESIGHT_ROOT/ice` and use this command like the following to configure trace components.

```
monitor nds configure component ncetenc apb 0xF2300000
monitor nds configure component ncetmux apb 0xF2310000
monitor nds configure component ncetbuf apb 0xF2309000
```

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

- `monitor nds set_group CORE_ID [halt|resume] GROUP_ID`

  This command is tailored for trace control on Asymmetric Multi-Processing (AMP) targets that support core grouping. It assigns an AMP core to or removes it from a specified group, ensuring that all cores in the group halt or resume together. By organizing AMP cores into halt or resume groups, this command enables synchronized execution control of the cores. It includes the following parameters:

  - `CORE_ID` specifies the target core ID, which must be an integer greater than or equal to 0.
  - `[halt|resume]` specifies a halt group or resume group.
  - `GROUP_ID` specifies the target group ID, which must be an integer greater than or equal to 1 for a valid group. When set to 0 or a negative value, the core is removed from its assigned halt/resume group.

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

Appendix

Appendix I. ICEman error & warning messages

The table below summarizes ICEman error messages along with their causes and solutions.

Error Message	Cause	Solution
`Can not open usb (vid=0x%x, pid=0x%x)`	The ICE is unplugged.	Connect the ICE to your target system.
`AICE ERROR! AICE is unplugged.`		
`TARGET ERROR! Target is disconnected with AICE.`	The target board is disconnected or power-off.	Connect your target board to the ICE or power it on.

`scan target error`
`No target specified`
`TARGET ERROR! Failed to identify AndesCore JTAG Manufacture ID in the JTAG scan chain. Failed to access EDM registers.`		

`There is no vid_pid in config files.`	There is no `vid_pid` value in the configuration file `nds32-aice.cfg`	Define vid & pid in `nds32-aice.cfg` (e.g., `aice vid_pid 0x1CFC 0x0000`)
`EDM access ERROR!`	EDM registers can't be accessed.	Restart the target board and check if the debug pins are pinmuxed or the CPU enters the standby mode.
`aice_issue_srst ERROR!`	The system reset feature may not be supported in your ICE.	Check if your ICE has appropriate firmware or pins to support this feature.
`aice_issue_restart ERROR!`		
`SW reset-and-hold failed.`		
`HW reset-and-hold failed.`		
`TARGET ERROR! Reaching the max interrupt stack level.`	The interruption stack system registers overflowed.	Check your target code and target state.
`TARGET ERROR! Insufficient security privilege to execute the debug operations.`	The target has insufficient security	Check for permission based on your DBGSPL

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

Error Message	Cause	Solution
	privilege for debugging.	(Debug Security Privilege Level) and use the correct passcode for debugging.
TARGET ERROR! Debug operations do not finish properly: 0x%08x 0x%08x 0x%08x 0x%08x.	The debug target failed to execute DIM (Debug Instruction Memory) programs.	Check your target state.
TARGET ERROR! The debug target failed to update the DTR register.	The debug target can't access the EDM_DTR (EDM Data Transfer Register) correctly.	Check your target code.
TARGET ERROR! AICE failed to write to the DTR register.		
TARGET ERROR! Unable to stop the debug target through DBGI.	The debug target can't enter the debug mode.	Check your target code.
Failed to initialize AICE box!	The ICE box can't be reset successfully.	Re-plug your ICE box.
reset-and-hold failed	The system reset feature may not be supported by your ICE.	Check if your ICE box has appropriate firmware or pins to support this feature.
TARGET ERROR! Debug operations do not finish properly.	The debug module of the target failed to execute debug operations.	Check your target state.
Unable to execute JTAG queue	The debug target failed to execute JTAG commands	Check your target state.
JTAG scan chain interrogation failed: all %s Check JTAG interface, timings, target power, etc.	The target board is disconnected or power-off.	Connect the target board to your ICE or power it on.
Failed to read from 0x%x	Data was not read successfully from DMI	Check your target state.

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

Error Message	Cause	Solution
`Failed to write to 0x%x`	Data was not written successfully to DMI	Check your target state.
`Unexpected hart status during reset.`	The current debug target is halted or unavailable.	Check your target state.
`unable to halt hart %d`	The debug target can't enter the debug mode.	Check your target state
`Failed to resume hart %d (for step?=%d)`	The debug target can't be resumed.	Check your target state
`<-- JTAG scan chain interrogation failed: all zeroes -->` `<-- Check JTAG interface, timings, target power, etc. -->`	The values on TDO are all zeros.	• Check the JTAG pin connections. • Make sure that RTL debugger pattern has passed the simulation test bench. • Try using SDP (2-wires)
`ERROR: No config file, unable to <read/write> <CFG_FILENAME>`	The config file can't be written or read.	• Make sure all config files are under `ANDESIGHT_ROOT/ice`. • Make sure you have the write permission to the folder where config files reside.
`ERROR: No config file, openocd.cfg.v5`		
`Failed [nop/read/write] at 0x%x; status=%d`	The JDTM transaction failed.	• Make sure that RTL debugger pattern has passed the simulation test bench. • Check if JDTM/PLDM works properly.
`Failed [read/write] (NOP) at 0x%x; value=0x%x, status=%d`		

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

Error Message	Cause	Solution
DMI operation didn't complete in %d seconds. The target was either really slow or broken. You could increase the timeout with riscv set_command_timeout_sec.	The DMI command has timed out.	• Increase delay cycles • Extend the waiting time.
Timed out after %ds waiting for the register field "busy" to go low (abstractcs=0x%x). Increase the timeout with riscv set_command_timeout_sec. Please set dmi_busy_retry_times > %d to increase delay cycles.		
OpenOCD only supports Debug Module version 2 (0.13), not %d (dmstatus=0x%x). This error might be caused by a JTAG signal issue. Try reducing the JTAG clock speed.	The version of the JDTM/PLDM is not supported.	• Check if JDTM/PLDM works properly. • Make sure that RTL debugger pattern has passed the simulation test bench.
Unsupported DTM version %d. (dtmcontrol=0x%x)		
Debug Module was not active. dmcontrol=0x%x	The PLDM didn't work properly.	• Check if PLDM works properly. • Make sure not to reset PLDM during the debug session.
Hart %d didn't complete a DMI read coming out of reset in %ds; Increase the timeout with riscv set_reset_timeout_sec.	The hart didn't run or halt after the reset signal is released.	• Make sure that the reset signal can be released. • Check the hart status after the reset signal is released.

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

Error Message	Cause	Solution
`Timed out after %ds waiting for the register field "sbbusy" to go low (sbcs=0x%x). Increase the timeout with riscv set_command_timeout_sec.`	System bus accesses failed.	• Check PLDM bus interface signals. • Check the bus status.
`<-- Unable to halt [%s] hart %d --> dmcontrol=0x00000001 dmstatus =0x00400ca2`	The hart kept running after the halt request was transmitted.	• Check the halt request signal. • Check the bus status.
`<-- Unable to halt [%s] hart %d --> dmcontrol=0x00000001 dmstatus =0x004030a2`	There was no available hart to be halted.	• Check if the reset signal is released. • Make sure the hart is available before enabling a GDB connection.
`unable to resume hart %d dmstatus =0x%08x`	The hart kept running and couldn't be resumed	Check the resume request signal.

In addition to error messages, ICEman may also produce the following warning messages, prompting you to examine your target code:

- `TARGET WARNING! The debug target has been reset.`
- `TARGET WARNING! The debug target exited the debug mode unexpectedly.`
- `TARGET WARNING! Exception is detected and suppressed.`
- `TARGET WARNING! Too many hardware breakpoints/watchpoints are inserted. The number of available hardware breakpoints/watchoint is N.`

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

Appendix II. ICE troubleshooting

Appendix II-I. Diagnosis

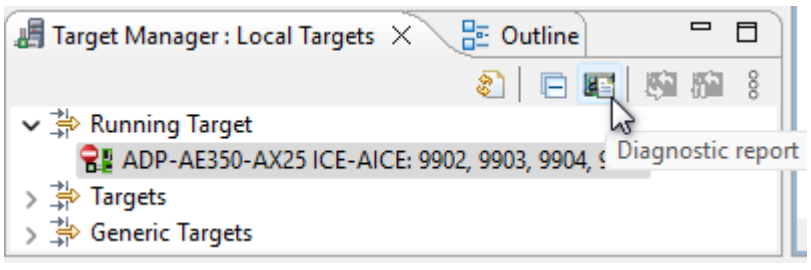
If you encounter problems constantly using an ICE device to debug a program on your target system, you may make use of ICEman to ascertain the state of your ICE.

In a command line terminal, change the directory to where your ICEman resides and issue "`./iceman.exe [-x|--diagnosis]`". The diagnosis process will be commenced right away. For details about the ICEman option `[-x|--diagnosis]`, please refer to Section 2.1.18. The following is an example to diagnose the AICE-MICRO for an Andes target board.

```
/d/andestech/andesight/ice
$ ./iceman.exe -x
Andes ICEman (OpenOCD) 5.4.0-g2e356e1 (2024-09-25-02:11)
Burner listens on 2354
Telnet port: 4444
TCL port: 6666
Open On-Chip Debugger 0.11.0+dev-g1c97df4 (2024-09-25-02:09)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Andes AICE-MICRO H/W R1.1
JTAG frequency 10.000 MHz
There are 4 cores in tap
target-0 = tap0_target_0 0x0 0x1
target-1 = tap0_target_0_1 0x1 0x1
target-2 = tap0_target_0_2 0x2 0x1
target-3 = tap0_target_0_3 0x3 0x1
The core #0 listens on 1111.
The core #1 listens on 1112.
The core #2 listens on 1113.
The core #3 listens on 1114.
ICEman is ready to use.
```

In AndeSight IDE, you can follow the steps below to diagnose your local ICE device:

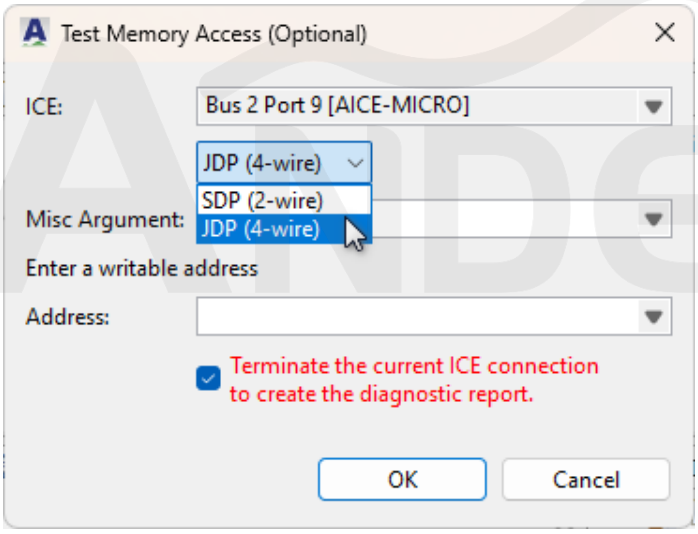
Step 1 In the **Target Manager** view, select the desired running target and click the Diagnostic Report button on the toolbar.



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

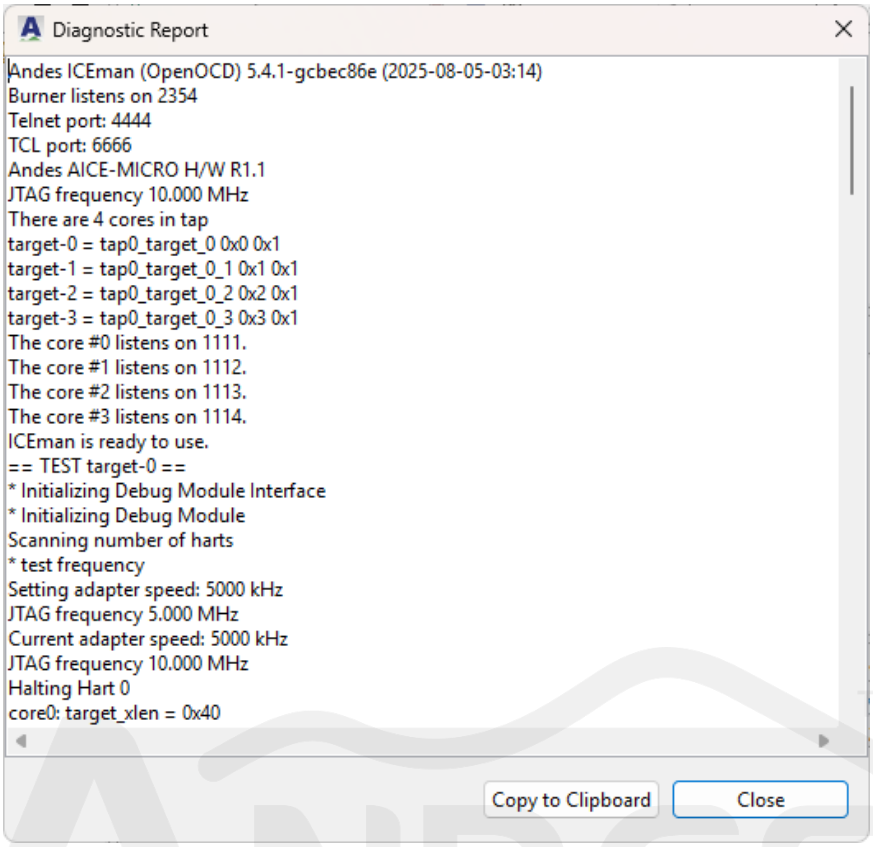
Step 2 The invoked dialog shows detected local ICE device(s) along with the device position(s) in the ICE combo box. If there are multiple local ICE devices available, select the one that you want to diagnose. For an AICE-MINI+, AICE-T2, or AICE-MICRO, ensure you also specify the target's debug interface – either "JDP (4-wire)" or "SDP (2-wires)" – from the combo box below.

Next, optionally enter connection arguments for your ICE device and input a writable address to test the memory access. If the selected ICE device is currently in use and occupied, be sure to select the option "Terminate the current ICE connection to create the diagnostic report" so that the report can be generated.



Step 3 The diagnostic report for the debugger will be generated and displayed in the dialog that pops up.

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



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

Appendix II–II. Resolution

- It should be a FPGA synthesis issue if a FPGA board is used and the diagnosis reports failure in following five tests –
  "check AICE versions ...",
  "check the detected JTAG frequency ...",
  "check changing the JTAG frequency ...",
  "check JTAG connectivity ..." and
  "check that JTAG domain is operational ...",

  In this case, please look up FPGA gated clock, JDP (5-wire) interface constraints and SDP (2-wire) interface constraints in corresponding datasheets and take note of the followings:
  1. All I/O pads should use the pad type with standard VREF, e.g., LVCMOS_33 for VREF 3.3V. (Must)
  2. The slew rate of tms must be "fast." (Must)
  3. The input/output delay of tms is adjustable if there is any unstable tms signal. (Optional)
  4. A design flow with the Syplify software is recommended.
  5. If you still fail to connect ICE after following above instructions, use the ICEman option "-c" to slow down the frequency (core_clk/bus_clk/edm_tck) and try again. As to the option values for specific clock frequencies, please refer to the ICEman help menu listed in Chapter 1.

- If the diagnosis reports failure in the following tests —
  "check that TRST resets the JTAG domain ...",
  "check that SRST does not resets JTAG domain ..." and
  "check reset-and-debug ...",
  please reference "System Reset Generator" and "Target System Reset" sections in relevant AICE user manuals to set the system reset circuit and timing.

Some error messages may also show in the diagnosis to indicate specific problems. Please refer to Appendix I for information about the messages and problem solutions.

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

Appendix III. ICEman operations for crash debugging

Certain Andes processors support crash debugging, allowing the retention of processor states prior to a reset. When these processors enter debug mode immediately after a reset for crash debugging,

- The register `dpc` points to the reset vector.
- The register `mepc` records the program counter before the reset.
- Register files retain their values before the reset.

To perform crash debugging with these processors, just manually terminate ICEman and restart it with the `-H` or `--reset-hold` option to hold the reset state.

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