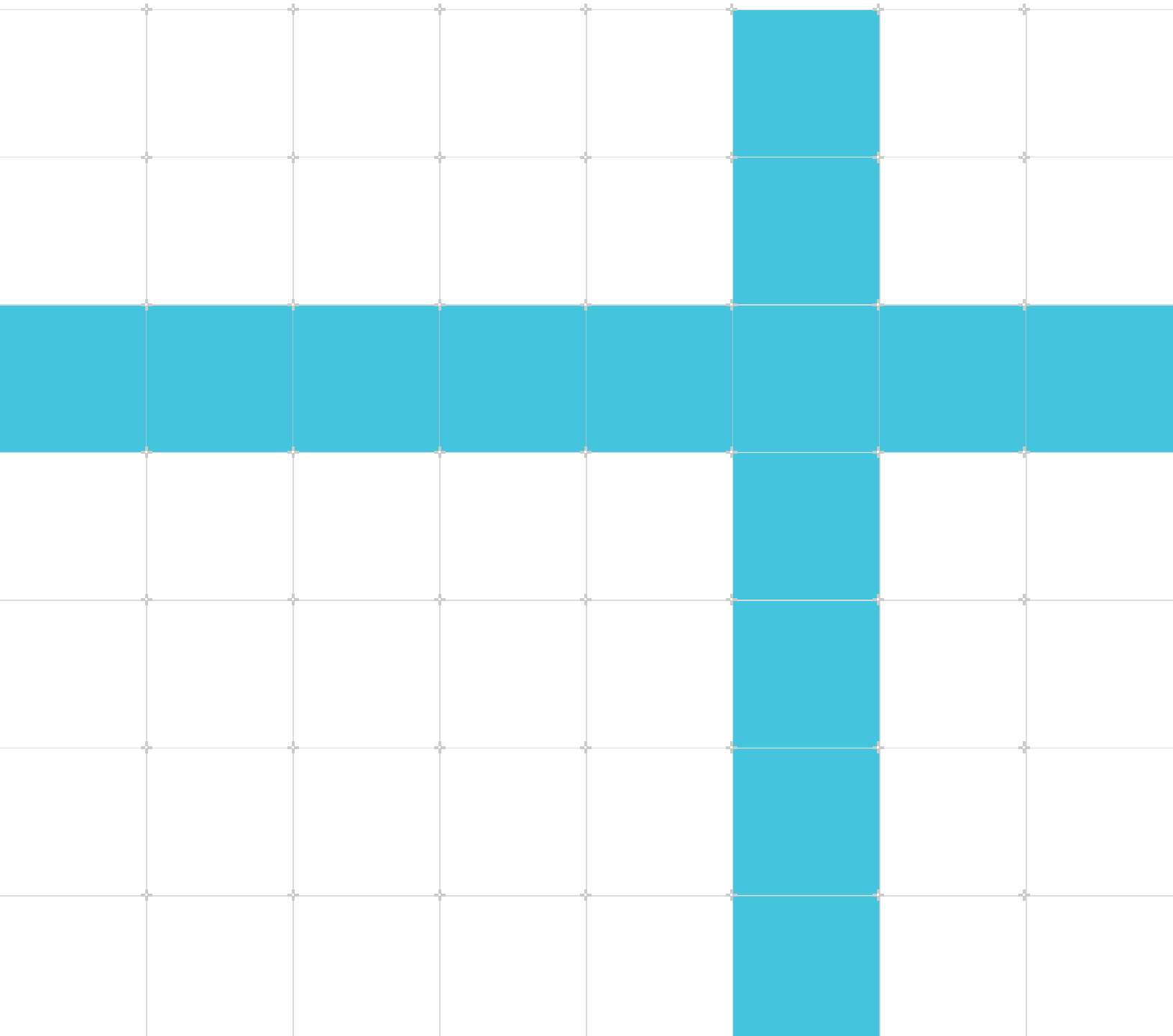
ASTFplugin

Version 1.0

FAQs

Issue 00
108074_0100_00_en



ASTFplugin
FAQs

Release information

Document history

Issue	Date	Confidentiality	Change
0100-00	13 September 2023	Non-Confidential	Initial release

Proprietary Notice

and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Introduction

1.1 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
Note	An important piece of information that needs your attention.

Convention	Use
	A useful tip that might make it easier, better or faster to perform a task.
	A reminder of something important that relates to the information you are reading.

1.2 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Fast Models Reference Guide	100964	Non-Confidential

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

1.3 Other information

See the Arm website for other relevant information.

- Arm® Developer.
- Arm® Documentation.
- Technical Support.
- Arm® Glossary.

2. How to trace a workload running in a multi-core Linux environment

I want to trace an application's workload running in a guest multi-core Linux environment but Linux OS migrates applications between cores. What should I do?

Answer

When using the `HLT` method of toggling trace with ToggleMTIPlugin, ASTF tracing in Fast Models is per core. So, you must enable the `enable_trace_special_hlt_imm16` and `trace_special_hlt_imm16` parameters of all the cores.

Related information

- How to use ToggleMTIPlugin

3. Architectural limitations of ASTFplugin

What are the architectural limitations of ASTFplugin in the Fast Models 11.23 release?

Answer

ASTF v0.11 supports up to Armv9.3-A. The plug-in is not guaranteed to work for architecture versions later than that.

4. Does ASTF trace generation have a maximum trace file size?

Answer: No, it runs until your disk space is full and is killed by your OS.

5. Minimum and maximum trace length

How much trace data should I capture?

Answer

A minimum trace log size of around 100M instructions is recommended to account for cache warming. A maximum of 10B instructions is recommended for efficient post-processing and analysis.

6. How to distinguish between multiple programs in trace logs

If I generate traces for multiple programs, how do I distinguish between them in the trace logs?

Answer

If the guest Linux kernel is configured with `'CONFIG_PID_IN_CONTEXTIDR'` enabled, PID information is included in the `context` section of the trace logs, for example:

```
5 context : CPU in EL0, non-secure, thread-mode PID: 29193
```

PID information is only recorded if there is a PID change on a core.

7. Can ASTFplugin trace process and thread IDs?

ASTFplugin supports recording PID/TID information through the CONTEXTIDR_EL1 register, if the guest supports it. However that requires an additional pass for full PID/TID information. Is there a way to collect PID/TID information that doesn't require a re-run?

Answer

It is possible to do this with some additional post-processing. ASTFplugin traces PIDs using the CONTEXTIDR_EL1 register and includes them in the trace, if they are available, in the first run. To fully match up the PIDs and TIDs in the trace, you then need to generate a PID-TID map from the OS and use the trpidannotate tool to amend the trace with the appropriate TID information.

Related information

- ASTF tools

8. How to distinguish between different clusters and cores

How does the plug-in distinguish between different clusters and cores?

Answer

The plug-in has no concept of the cluster and core topology of the model. It simply queries whether each component can execute code. If it can, the plug-in attaches itself to the trace sources of the component that it needs to generate the ASTF streams. If not, it ignores that component. The resulting file names that include the clusters and cores are generated by the model and accepted by the plug-in.

9. Why do files seem to be missing?

I sometimes find that files seem to be missing. For example, in a model with two CPUs, I want to record two blocks. Files `cpu0.0000.astf` and `cpu1.0000.astf` record the first block and `cpu0.0001.astf` records the second block. Why is `cpu1.0001.astf missing`?

Answer

If a CPU is inactive while the plug-in is recording, the plug-in omits generating the associated file. Otherwise, the file would only contain the ASTF header and nothing else. So if a file is missing, it might be because that CPU was not active while the plug-in was recording.

10. Running counter

Why does the running counter for my files sometimes start with 0000 and sometimes with 0001?

Answer

When you request the plug-in to stop recording, it increments the running counter to ensure the next block's ASTF files have a different filename to those of the current block.

By default, the plug-in starts recording after it has initialised. However, if you request the plug-in to stop recording immediately, it increments the running counter and produces no `cpu*.0000.astf` files, because none of the CPUs had the chance to execute code. If you then request the plug-in to continue recording, the next block's stream files will have the 0001 counter value in their names.

Related information

- Why do files seem to be missing?

11. Why do I see messages about missing trace sources

Why do I see messages like `trace source SVE_LOADS/SVE_STORES not detected -> omitting registration`?

Answer

The plug-in is not feature-aware and therefore does not know if a CPU supports SVE. As a result, it always tries to register `SVE_LOADS` and `SVE_STORES` trace sources. If a CPU supports SVE and this message appears, then see Why do I see messages about SVE_LOADS/SVE_STORES?. Otherwise, you can safely ignore them.

12. Why do I see messages about SVE_LOADS/SVE_STORES?

The message `trace source SVE_LOADS/SVE_STORES not detected -> omitting registration` appears when it should not. Why is this happening?

Answer

The most likely reason is that in the command line for launching the model, the SVE plug-in and the ASTF plug-in were set in the wrong order. To register those trace sources, you must load the SVE plug-in before the ASTF plug-in. For more details, see ASTFplugin usage notes in the *Fast Models Reference Guide*.