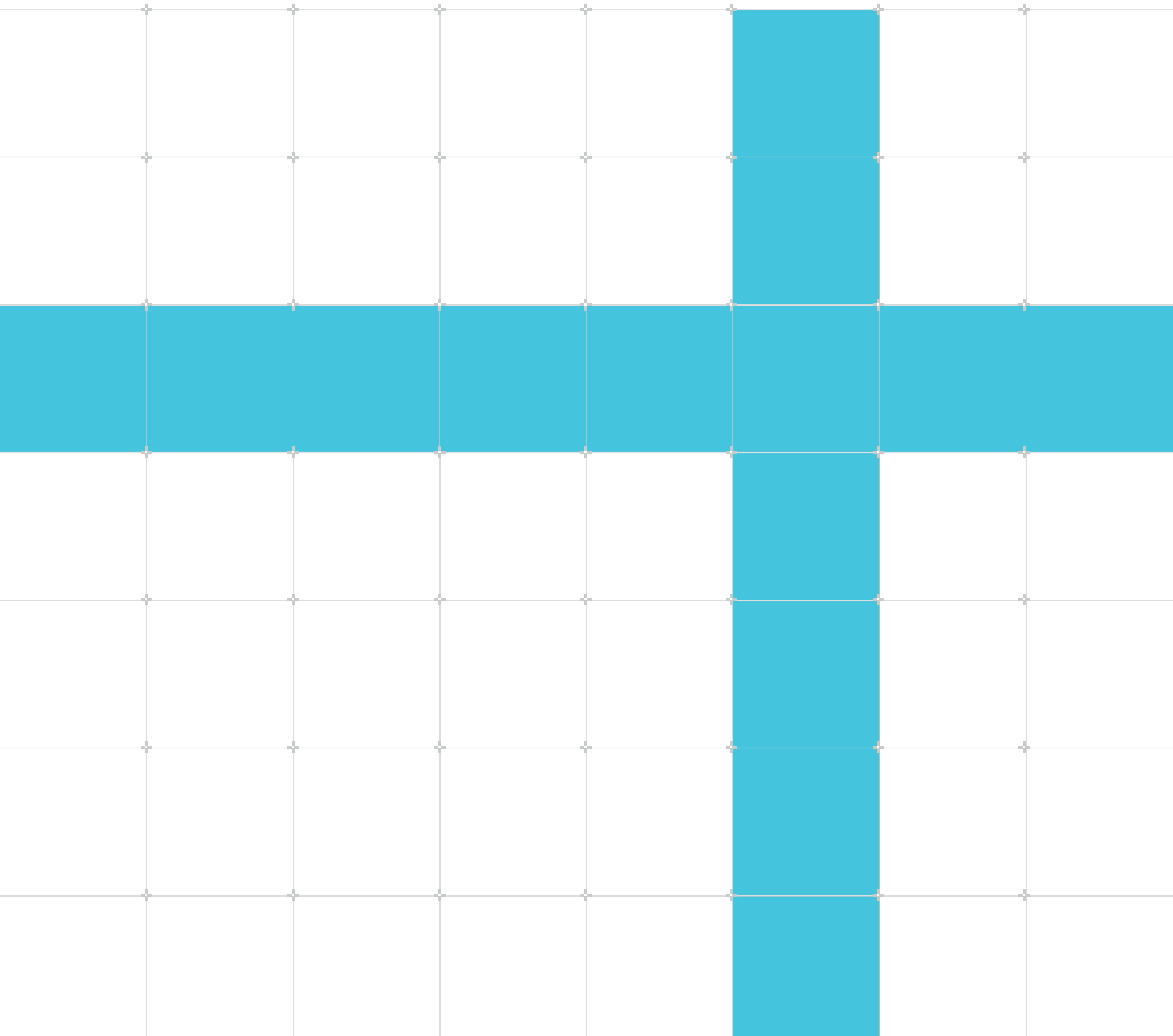
Troubleshooting DSTREAM-PT Trace

Version 1.0

Issue 02
102637_0100_02_en



Troubleshooting DSTREAM-PT Trace

Release information

Document history

Issue	Date	Confidentiality	Change
0100-02	9 August 2021	Non-Confidential	First release

Proprietary Notice

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. Overview

When using DSTREAM-PT with Arm Development Studio 2019.0 or later, you may find that:

- The trace buffer is not filling.
- No trace data is displayed in the Trace view.
- There is a <corrupted message> in the Trace view.
- There is some trace data displayed in the Trace view, but there is also a <corrupted message> entry as well.

Unlike other units in the DSTREAM family of products, DSTREAM-PT has a feature called a Real-time Trace Monitor (RTM). The RTM tries to calibrate the trace data lines to improve the chances of collecting reliable trace data. Very occasionally, the RTM is not able to collect enough trace data or is unable to get a lock onto the trace signals in order to do the calibration. When this occurs, you might not see the trace buffer fill, there might be no trace data displayed, or the trace data might be partially or entirely corrupted. This KBA is here to help you figure out if the RTM might be causing this behavior and tell you what to do if it is.

2. Troubleshooting steps

The RTM has an advanced logging feature, which when enabled, gives more information about the RTM's findings. If appropriate, the log information can be used to manually configure different aspects of the trace capture process.

The below steps go through:

1. Enabling the RTM advanced logging feature in the platform configuration.
2. Capturing and analyzing the RTM logging information.
3. Checking the trace signals for:
    - Stuck at '0' or '1' faults.
    - A stable trace clock frequency.
    - The correct trace clock alignment.
4. Potentially, setting the new trace data signal delay values.
5. Determining the results and any next steps.

3. Platform configuration

To enable the RTM advanced logging, the DSTREAM-PT configuration items need to be present and active in the platform configuration. How you work with and setup the configuration items is based on how the platform configuration is or was generated.

Below are the different options for the platform configuration you are working with.

Read one of the following sections that best fits the platform configuration you are using and follow the instructions provided:

- If you created a new platform configuration with the Arm Development Studio Hardware Connection Wizard or the Platform Configuration Editor (PCE).
- If you are using an existing platform configuration with a `.rcf` and/or `.sdf` file.
- If you are using an existing platform configuration with a `.rvc` file.

Configuration with Arm Development Studio Hardware Connection Wizard or PCE

The DSTREAM-PT configuration items will automatically by added to the platform configuration's `.sdf`. However, in Arm Development Studio 2019.0, the DSTREAM-PT configuration items are not enabled by default. To enable the DSTREAM-PT configuration items and RTM advanced logging feature, perform the following steps:

1. Open the platform configuration `.sdf` as a text file.
2. In Arm Development Studio, expand the ExtensionDB in the Project Explorer view, right-clicking on the `.sdf`, and selecting Open with > Text Editor.
3. Add HTML comments around the <trace_capture type="parallel"> configuration item block like below:

```
<!--
    <trace_capture type="parallel">
    ...
    </trace_capture>
    -->
```

4. Remove the HTML comments around the <trace_capture type="DSTREAM-PT"> configuration item block (that is, delete <!– and –> around the <trace_capture type="DSTREAM-PT"> section).
5. Set ENABLE_ADDITIONAL_RTM_LOGGING to 1, for example: <config_item name="ENABLE_ADDITIONAL_RTM_LOGGING>1</config_item>.
6. Click File > Save.
7. If you are editing the `.rcf` or `.sdf` outside Arm Development Studio, rebuild the configuration database in Arm Development Studio by clicking Window > Preferences > Arm DS > Configuration Database > Rebuild database. Click here for an `example.rcf` and Click here for an `example .sdf` with the ENABLE_ADDITIONAL_RTM_LOGGING configuration item set.


Note

If you wish to use the configuration items for the DSTREAM or DSTREAM-ST after making these changes to the .sdf, you will need to comment out the <trace_capture type="DSTREAM-PT"> configuration item block and un-comment the <trace_capture type="parallel"> configuration item block. Not performing this action might prevent the trace delay signals from being set correctly.

Using an existing platform configuration with a .rcf and/or .sdf file

Existing platform configurations might contain:

- Just a `.rcf` file.
- Just a `.sdf` file.
- `.rcf` and `.sdf` files.

Platform configurations with any of the above contents will allow the RTM advanced logging feature to be enabled.

You need to add the DSTREAM-PT configuration items to the `.rcf` or `.sdf`. If the platform configuration contains both a `.rcf` and a `.sdf`, the DSTREAM-PT configuration items need to be added to just the .rcf.

To add the DSTREAM-PT configuration items and enable RTM advanced logging feature, perform the following steps:

1. Open the platform configuration .rcf or .sdf as a text file. In Arm Development Studio, if the platform configuration is in the ExtensionDB, you can do this by expanding the ExtensionDB in the Project Explorer view, right-clicking on the `.rcf` or `.sdf`, and selecting Open with > Text Editor.
2. Add HTML comments around the <trace_capture type="parallel"> configuration item block like below:

```
<!--
    <trace_capture type="parallel">
    ...
    </trace_capture>
    -->
```

3. Right underneath the commented out <trace_capture type="parallel"> configuration item block, copy and paste all the DSTREAM-PT configuration items from the text file here.
4. Set ENABLE_ADDITIONAL_RTM_LOGGING to 1, for example: <config_item name="ENABLE_ADDITIONAL_RTM_LOGGING">1</config_item>.
5. Click File > Save.
6. If you are editing the `.rcf` or `.sdf` outside Arm Development Studio, rebuild the configuration database in Arm Development Studio by clicking Window > Preferences > Arm DS > Configuration Database > Rebuild database. Click here for an `example.rcf` and Click here for an `example.sdf` with the ENABLE_ADDITIONAL_RTM_LOGGING configuration item set.


Note

If you wish to use the configuration items for the DSTREAM or DSTREAM-ST after making these changes to the `.sdf` or `.rcf`, you will need to comment out the <trace_capture type="DSTREAM-PT"> configuration item block and un-comment the <trace_capture type="parallel"> configuration item block. Not performing this action might prevent the trace delay signals from being set correctly.

Using an existing platform configuration with a .rvc file

Platform configurations created with earlier Arm tools (like older DS-5 versions) might not contain a `.sdf` file, but will have a `.rvc` file. To add the DSTREAM-PT configuration items to the `.rvc` and enable RTM advanced logging feature, perform the following steps:

1. Open the platform configuration `.rvc` as a text file. In Arm Development Studio, you can do this by expanding the ExtensionDB in the Project Explorer view, right-clicking on the `.rvc`, and selecting Open with > Text Editor.
2. Add HTML comments around the <DataList Type = "Branch"> configuration item block like below:

```
<!--
   <DataList Type = "Branch">
    ...
    </DataList>
    -->
```

3. Add the DSTREAM-PT RTM configuration items from here after the commented out <DataList Type = "Branch"> configuration item block.
4. Set ENABLE_ADDITIONAL_RTM_LOGGING to 1, for example: <ENABLE_ADDITIONAL_RTM_LOGGING Type = "Int32">1</ENABLE_ADDITIONAL_RTM_LOGGING>.
5. Click File > Save.
6. If you are editing the `.rvc` outside Arm Development Studio, rebuild the configuration database in Arm Development Studio by clicking Window > Preferences > Arm DS > Configuration Database > Rebuild database. Click here for an `example .rvc` with the ENABLE_ADDITIONAL_RTM_LOGGING configuration item set.


Note

If you wish to use the configuration items for the DSTREAM or DSTREAM-ST after making these changes to the .rvc, you will need to comment out the added <DataList Type = "Branch"> configuration item block and un-comment the original <DataList Type = "Branch"> configuration item block. Not performing this action might prevent the trace delay signals from being set correctly.

4. RTM log analysis

Now the extra configuration items have been added for the DSTREAM-PT, we can now obtain the extra logging required for analyzing the output of the RTM.

1. Open a command prompt to the Arm Development Studio bin directory (<Arm Development Studio installation directory>).
2. In the command prompt, run:
   - For a DSTREAM-PT USB connection: `dbghw_log_client USB > <writable path>\log.txt`
   - For a DSTREAM-PT TCP connection: `dbghw_log_client TCP:<DSTREAM-PT IP address>><writable path>\log.txt` The above command will run the dgbhw_log_client utility and have it save any generated output to `log.txt`.

You may need administrator permissions to run the dbghw_log_client utility.

3. If you are currently connected to the target with Arm Development Studio, disconnect and then re-connect to the target.
4. Perform the trace capture again.
5. After trace has been captured, type Ctrl-C into the command prompt running the dbghw_log_client utility to stop the utility.
6. Open the `log.txt` file generated by the dbghw_log_client utility.

If the log.txt file only contains one line of text, RTM information will not be output, as trace capture has stopped before 1 second has elapsed, then the RTM was unable to generate logging information because there was not enough trace data generated by the board. You should go back to step 4 and perform the trace capture again for a longer period of time (i.e. more than 1 second) and until this message on longer appears.

7. Look for a RTM lock not achieved message.
   - If there is an RTM lock not achieved message, continue going through this tutorial.
   - If there is not an RTM lock not achieved message, go to Results.

Now three checks are performed to determine possible reasons for the RTM lock not achieved message.

5. Check 1

This chapter gives an instruction how to look for trace data signal stuck at '1' or stuck at '0' faults.

Look for trace data signal stuck at 1 or stuck at 0 faults

The RTM_P_EDGE and RTM_N_EDGE vectors in the RTM log indicate whether both a positive and negative edge was detected. Both of these vectors should have a single bit set for each active trace data signal to indicate that both a positive and negative edge was found. For example, if your trace port width is 16 bits then both RTM_P_EDGE and RTM_N_EDGE should equal `0x0000ffff` in the RTM log.

1. Check RTM_P_EDGE and RTM_N_EDGE from the RTM log.
    - If the values of the RTM_P_EDGE and RTM_N_EDGE vectors match the trace port width, go to Check 2.
    - If bit(s) are not set for RTM_P_EDGE or RTM_N_EDGE, then this might indicate that the corresponding trace data signal(s) are stuck at '1' or stuck at '0'. You will need to consult your board designer or check the trace data signals manually with an oscilloscope to see whether this is true before performing another trace capture attempt with the DSTREAM-PT.

6. Check 2

This chapter gives instructions how to look for a stable trace clock frequency.

Look for a stable trace clock frequency

1. Look for a Trace frequency stable message in the RTM log.
   - If Trace frequency stable: true, go to Check 3.
   - If Trace frequency stable: false, perform the following steps.
2. Save a copy of the `.sdf`, `.rcf`, or `.rvc` file before any modifications are made.
3. Open the platform configuration `.sdf`, `.rcf`, or `.rvc` as a text file.
4. Set RTM_FREQUENCY_OVERRIDE_ENABLE to 1, for example: <RTM_FREQUENCY_OVERRIDE_ENABLE Type = “Int32”>1</ RTM_FREQUENCY_OVERRIDE_ENABLE>.
5. Set RTM_FREQUENCY_OVERRIDE_VALUE to either the last RTM_MEAS_FREQ value shown in the RTM log or to the actual target trace port clock frequency.

If you are setting the RTM_FREQUENCY_OVERRIDE_VALUE to the actual trace port clock frequency, the RTM_FREQUENCY_OVERRIDE_VALUE format is the clock frequency in Hz divided by 100 and converted into hex. The below formula applies: RTM_FREQUENCY_OVERRIDE_VALUE = converted into hexadecimal (trace port clock frequency in Hz / 100)

6. Click File > Save.
7. Start a new dbghw_log_client session using a new log file as output (like `<writable path>\log1.txt`).
8. If you are editing the `.sdf`, `.rcf`, or `.rvc` outside Arm Development Studio, rebuild the configuration database in Arm Development Studio by clicking Window > Preferences > Arm DS > Configuration Database > Rebuild database.
9. If you are currently connected to the target with Arm Development Studio, disconnect and then re-connect to the target.
10. Perform the trace capture again.
11. Look for a RTM lock not achieved message in the new RTM log.

- If there is an RTM lock not achieved message, go to Check 3.
- If there is not a RTM lock not achieved message, go to Results.

7. Check 3

This chapter gives instructions how to look at the trace clock alignment.

Look at the trace clock alignment

The RTM_ALIGN_0 and RTM_ALIGN_90 vectors in the RTM log tell us whether the trace clock is aligned with the clock signal edge or whether it is aligned with the edge inverted 90 degrees respectively. The RTM assumes the trace clock is aligned to the edge by default.

To determine which alignment the RTM should use, count the number of consecutive bits that are set. The alignment with the most consecutive bits set is the alignment that the RTM should use. For example, if RTM_ALIGN_0 = `0x00000007` and RTM_ALIGN_90 = `0x019fffff`, the trace clock edge alignment should be inverted 90 degrees because RTM_ALIGN_90 has 21 consecutive bits set and RTM_ALIGN_0 only has 3 consecutive bits set.

1. Check RTM_ALIGN_0 and RTM_ALIGN_90 from the RTM log.
   - If RTM_ALIGN_0 has the most consecutive bits set or the RTM_ALIGN_0 and RTM_ALIGN_90 vectors have the same number of consecutive bits set, go to Manual delay setting.
   - If RTM_ALIGN_90 has the most consecutive bits set, continue following these instructions.
2. Save a copy of the `.sdf`, `.rcf`, or `.rvc` file before any modifications are made.
3. Open the platform configuration `.sdf`, `.rcf`, `.rvc` as a text file.
4. Set RTM_ALIGNMENT_OVERRIDE_ENABLE to 1, for example: <RTM_ALIGNMENT_OVERRIDE_ENABLE Type = “Int32”>1</ RTM_ALIGNMENT_OVERRIDE_ENABLE>.
5. Set RTM_ALIGNMENT_OVERRIDE_VALUE to 1, for example: <RTM_ALIGNMENT_OVERRIDE_VALUE Type = “Int32”>1</ RTM_ALIGNMENT_OVERRIDE_VALUE>.
6. Click File > Save.
7. Start a new dbghw_log_client session using a new log file as output (like `<writable path>\log2.txt`).
8. If you are editing the `.sdf`, `.rcf`, or `.rvc` outside Arm Development Studio, rebuild the configuration database in Arm Development Studio by clicking Window > Preferences > Arm DS > Configuration Database > Rebuild database.
9. If you are currently connected to the target with Arm Development Studio, disconnect and then re-connect to the target.
10. Perform the trace capture again.
11. Look for a RTM lock not achieved message in the new RTM log.

- If there is an RTM lock not achieved message, go to Manual delay setting.
- If there is not an RTM lock not achieved message, go to Results.

8. Manual delay setting

We will now try manually setting delays for the trace data signals based on the RTM log.

1. Look at the last set of RTM_DELAY_X entries displayed in the `log.txt` file (where X is a number from 1-32). :::Note You will set the DSTREAM-PT DELAY_TRACE_SIGNAL configuration items in the platform configuration's `.sdf`, `.rcf`, or `.rvc` file based on the values shown for RTM_DELAY_X. What value you use is based on the RTM_DELAY_X values in the log. :::
   - If the all the RTM_DELAY_X entries have a least significant byte of `0x00` or `0x1f`, you will set use a trace data signal delay value of `0x10`.
   - If a few of the RTM_DELAY_X entries have a least significant byte of `0x00 or 0x1f, but some of the least significant byte values are between 0x00 and 0x1f, you will use a trace data signal delay of one of the non-0x00` or non-`0x1f` values.

   The DELAY_TRACE_SIGNAL values are based in picoseconds at a 82 picosecond resolution, so you will need to convert the least significant byte value shown in RTM_DELAY_X using the below formula: DELAY_TRACE_SIGNAL_X = converted to decimal(RTM_DELAY_X bits[7:0]) * 82 For example, a least significant byte of `0x10` equals a trace data signal delay of 1312.
2. Save a copy of the `.sdf`, `.rcf`, or `.rvc` file before any modifications are made.
3. Open the platform configuration `.sdf`, `.rcf`, or `.rvc` as a text file.
4. Set a bit for every bit of the target's trace port size to RTM_INITIAL_DELAY_OVERRIDE_ENABLES configuration item. For example, if you have a 16 bit trace port, then you would set `<RTM_INITIAL_DELAY_OVERRIDE_ENABLES Type = "Int32">0xFFFF</RTM_INITIAL_DELAY_OVERRIDE_ENABLES>`. This will override all the delay times used for all the trace data signals.
5. Set the DELAY_TRACE_SIGNAL_X configuration items to the trace data delay value determined in step 1. For example, if you want to set a trace data delay value of 1312 ps to every trace data signal in a 16-bit trace port, you set `<DELAY_TRACE_SIGNAL_X Type = "Int32">1312</DELAY_TRACE_SIGNAL_X>` where X is 1-16.


Note

You only need to set trace data signal delays for the size of your trace port. For example, if you have a 4-bit trace port, you set delays for DELAY_TRACE_SIGNAL_1 - 4 and leave the rest unaltered.

6. Click File > Save.
7. Start a new dbghw_log_client session using a new log file as output (like `<writable path>\log3.txt`).
8. If you are editing the `.sdf`, `.rcf`, or `.rvc` outside Arm Development Studio, rebuild the configuration database in Arm Development Studio by clicking Window > Preferences > Arm DS > Configuration Database > Rebuild database.
9. If you are currently connected to the target with Arm Development Studio, disconnect and then re-connect to the target.

10. Perform the trace capture again.

You may need to run the capture for several seconds to ensure good output.

9. Results

If you have completed all the checks and steps above, you should now see valid, un-corrupted trace data in the Arm Development Studio Trace view.

If there is not an RTM lock not achieved message in the RTM advanced log, or if you get the same results in the Trace view after performing all the above contact your Arm tools support team because further investigation will be necessary. If you have a Support and Maintenance agreement with Arm you can contact their Technical Support team here.

The files you will need to provide to the support team are:

- The original platform configuration's `.sdf`, `.rcf`, or `.rvc` file.
- All the modified platform configuration `.sdf`, `.rcf`, or `.rvc` files.
- All the dbghw_log_client log files.

10. Example RTM Log

Below is an an example of an RTM log and comments explaining the different parts that will be used by this article (denoted by #).

```
\/_____RTM_____\/
Port width = 16

# The trace port width in bits being used by the RTM.  In this case, 16 bits.

RTM_STAT = 0x0000ad01
 ->        RTM initial value ready: true
 ->            Initial delay value: 1640 ps
 ->            First connect latch: false
 ->                       RTM lock: false
 -> Trace data half bit time value: 1722 ps

RTM_MEAS_FREQ = 0x80176ff4

# Used in Check 2. The trace clock frequency being used by the RTM.

 ->        Trace frequency: 153598800 Hz
 -> Trace frequency stable: true

# Used in Check 2.  Tells whether the trace clock frequency is stable.


RTM lock not achieved, so outputting alignments and edges:

# Message saying a RTM lock was not made.

RTM_ALIGN_0 = 0xfffffc80

# Used in Check 3.  In this case, there are 22 consecutive bits set.
RTM_ALIGN_90 = 0x011e7ffd

# Used in Check 3.  In this case, there are 13 consecutive bits set.

RTM_P_EDGE = 0x0000ffff

# Used in Check 1.  In this case, there are 16 bits set for a 16 bit trace port
 width.

RTM_N_EDGE = 0x0000ffff

# Used in Check 1.  In this case, there are 16 bits set for a 16 bit trace port
 width.


/\_____RTM_____/\
traced         info     (10:21:23.415): CTraceControlCmdProc::Process - command is
 262
traced         info     (10:21:23.416): TRC_CMD_STOP
traced         info     (10:21:23.416): Dstream2P32Store::Stop()
traced         warning  (10:21:23.420):
\/_____RTM_____\/
Trace capture has been stopped whilst waiting for the edge and data_eye_valids
 registers to become valid (equal to 0x0000ffff). The most recent values are:
RTM_P_EDGE = 0x0000ffff
RTM_N_EDGE = 0x0000ffff
RTM_DATA_EYE_VALIDS = 0x00008000
The data eye and delay registers will still be output anyway:
/\_____RTM_____/\
traced         info     (10:21:23.421):
```

```
\/_____RTM_____\/
Outputting data eye and delay registers:

RTM_DATA_EYE_VALIDS = 0x00008000
RTM_DATA_EYE_0 = 0x00000000
RTM_DATA_EYE_1 = 0x00000000
RTM_DATA_EYE_2 = 0x00000000
RTM_DATA_EYE_3 = 0x00000000
RTM_DATA_EYE_4 = 0x00000000
RTM_DATA_EYE_5 = 0x00000000
RTM_DATA_EYE_6 = 0x00000000
RTM_DATA_EYE_7 = 0x00000000
RTM_DATA_EYE_8 = 0x00000000
RTM_DATA_EYE_9 = 0x00000000
RTM_DATA_EYE_10 = 0x00000000
RTM_DATA_EYE_11 = 0x00000000
RTM_DATA_EYE_12 = 0x00000000
RTM_DATA_EYE_13 = 0x00000000
RTM_DATA_EYE_14 = 0x00000000
RTM_DATA_EYE_15 = 0x007fffff

RTM_DELAY_0 = 0x00001717

# Used when manually setting trace delays.  In this case, the trace delay is 1886

RTM_DELAY_1 = 0x00001818

# or can use this value which is 1968.

RTM_DELAY_2 = 0x00000000
RTM_DELAY_3 = 0x00000000
RTM_DELAY_4 = 0x00000000
RTM_DELAY_5 = 0x00001f1f
RTM_DELAY_6 = 0x00000000
RTM_DELAY_7 = 0x00001f1f
RTM_DELAY_8 = 0x00001f1f
RTM_DELAY_9 = 0x00001f1f
RTM_DELAY_10 = 0x00001f1f
RTM_DELAY_11 = 0x00001f1f
RTM_DELAY_12 = 0x00001f1f
RTM_DELAY_13 = 0x00001f1f
RTM_DELAY_14 = 0x00001f1f
RTM_DELAY_15 = 0x00000000
/\_____RTM_____/\
```