Cortex-M: Trigger on a Variable Application Note

Table of Contents

1 Introduction

This Application Note describes how to set a trigger on write to a variable with certain value and depict the content.

The following simple examples with source excerpt, which you can easily adapt to your environment, will show you how to:

- Use the DWT Trigger to mark the Trigger occurrence within ETM (Code execution) Trace stream
- Use another DWT Trigger to get the value of a variable and streaming it out via ITM, inserted in Trace
- Extend this trigger and record a certain value

Chapter 5 provides additional productivity tips.

1.1 ETM

ETM (Embedded Trace Macrocell - instruction and/or data trace) is supported on Cortex-M, Cortex-A/R. ETM architecture is available in different versions (e.g., ETMv4). Cortex-M ETM variants usually don't feature data trace and own comparators, while Cortex A/R ETM is usually fully featured.

1.2 DWT

DWT (Data Watchpoint and Trace - DWT hardware event trace) is supported on Cortex-M. It provides a low bandwidth focused data trace using comparators to detect memory accesses and then generate trace packets with information about memory accesses.

1.2.1 DWT Comparators

Each comparator can be configured for a specific address or/and for an address range (Address comparators). All comparators have a LSB masking register which implements an ignore mask. The size of the ignore mask is 0-31 bits, applied to address range matching. The range can be specified via *Ignore LSB bits* option.

Comparator 0 can compare against the clock cycle counter (CYCCNT). *Comparator 1* supports data comparison (Data comparator) and linking with another comparator. It can be set to two addresses or/and two ranges, and data value. In this case *Comparator 3* and/or *Comparator 2* are reserved.

A full DWT module contains four comparators that you can configure as:

- Hardware watchpoint,
- ETM trigger,
- PC sampler event trigger,
- Data address sampler event trigger.

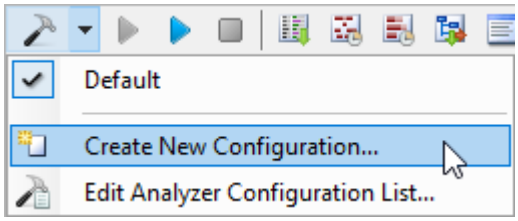
2 Trigger on a write to a Variable

First use case covers tracing and triggering on a write to a variable `dwTestWord` (16-bit, 32-bit) placed in a while-loop and updated once per loop.

➢ See productivity tip Data access sizes.

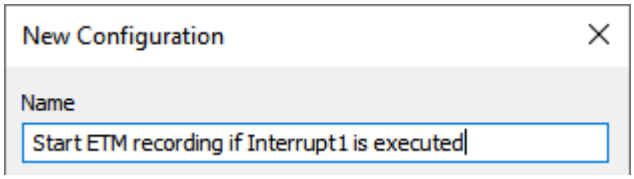
1 Create a new Trigger Configuration.

i. Open *View menu / Analyzer / Analyzer Configuration button*  .
ii. Select **Create New Configuration**.

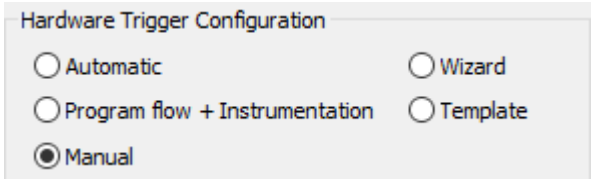


➢ See productivity tip Default Trigger.

iii. Give it a meaningful name.

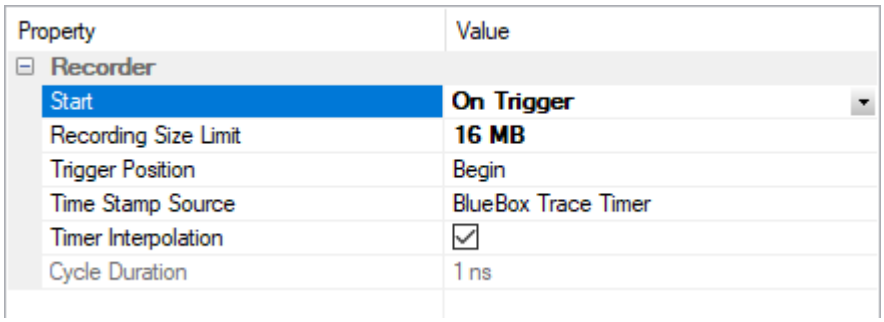


iv. Select *Manual* Hardware Trigger Configuration and confirm.



2 Select the *On Trigger* option.

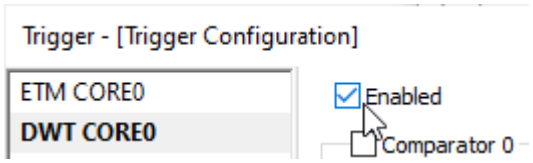
i. On the *Recorder* page set the sampling start point to *On Trigger*.
ii. Reduce Recording Size Limit to 16 MB. Note that 16 MB Recording Size Limit is set to save analyzing time.



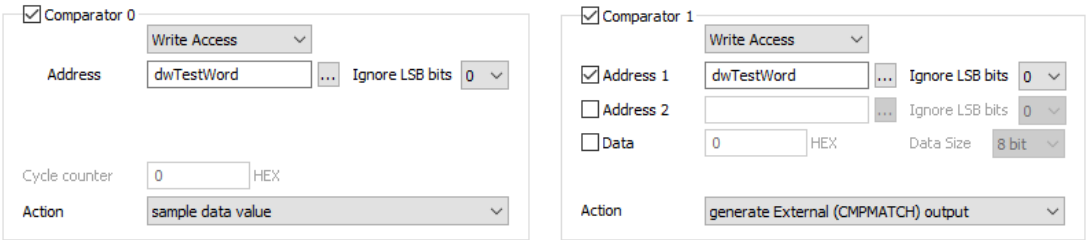
➢ See productivity tip Profile and limit the session duration.

3 Configure *Comparator 0* and *Comparator 1*.

i. Go to the *DWT CORE0* page and check the *Enabled* checkbox.



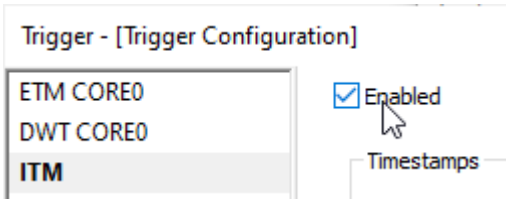
ii. Enable *Comparator 0* and *Comparator 1*.
iii. Select *Write Access* from the drop-down menu for both Comparators.
iv. Select addresses, e.g., `dwTestWord`.
v. Select Actions:
a. *Sample data value* for Comparator 0.
b. *Generate External (CMPMATCH) output* for Comparator 1.



➢ See productivity tip Search for variables via the Symbol Browser.

4 Enable ITM.

Go to the ITM page and check the *Enabled* checkbox.



➢ See productivity tip Configure unused DWT Comparators.

2.1.1 Results of the recording

CPU runs from RESET-entry until `main()` and is stopped on set breakpoint. The Trace recording is started and continued with execution from `main()`.

Analyzer Trigger marker  is placed in line 1.0. The *ETM Trigger* message itself is in the line 1.1. The value and action *write* you can find in the line 1.2.

-1.9	0000'0B04	00004770	ResetStrX_EXIT_	6.981960 ms
-1.10	0000'0B22	0000463B	sX1.m_pS = &sX2;	6.982248 ms
-1.11	0000'0B26	00004B04	++iCounter;	6.982285 ms
-1.12	0000'0B30	0000BF00	}	6.982380 ms
-1.13	0000'0B36	0000BD80	Type_Mixed_EXIT_	6.982437 ms
T 1.0	0000'122E	00004B40	dwTestWord++;	6.982456 ms
1.1	0000'0000	00000000	[ETM] Trigger	6.982544 ms
1.2	1FFE'001C	00000001	dwTestWord [ITM] Data Write	6.982584 ms
1.3	0000'123A	FC8DF7FF	Address_GlobalVariables();	6.982912 ms
1.4	0000'0B58	0000B580	{ Address_GlobalVariables	6.982920 ms
1.5	0000'0B5C	00004B05	++iCounter;	6.982937 ms

DWT/ITM acts independently from the ETM. MCU will send out ITM messages (i.e., if selected internal timestamps and variable values) constantly, regardless of the ETM part within MCU is set up, enabled or not. All the outputs are sometimes packed in one message and this is the reason for several lines like in the presented example. Therefore, some searching around the depicted trigger position is sometimes required.

1.1	0000'0AEC	0000683B	pS->m_c[0] = pS->m_c[1] = (char)iTo;	4.888824 ms
1.2	0000'0AFC	0000BF00	}	4.888845 ms
1.3	0000'0B04	00004770	ResetStrX_EXIT_	4.888856 ms
1.4	0000'0B22	0000463B	sX1.m_pS = &sX2;	4.889152 ms
1.5	0000'0B26	00004B04	++iCounter;	4.889204 ms
1.6	0000'0B30	0000BF00	}	4.889335 ms
1.7	0000'0B36	0000BD80	Type_Mixed_EXIT_	4.889413 ms
T 1.8	0000'122E	00004B40	dwTestWord++;	4.889440 ms
1.9	0000'0000	00000000	[ETM] Trigger	4.889480 ms
2.0	1FFE'001C	00000001	dwTestWord [ITM] Data Write	4.889520 ms
2.1	0000'123A	FC8DF7FF	Address_GlobalVariables();	4.889816 ms
2.2	0000'0B58	0000B580	{ Address_GlobalVariables	4.889827 ms
2.3	0000'0B5C	00004B05	++iCounter;	4.889849 ms

3 Trigger at a value of a Variable

You can extend *Trigger on a write to a variable* use case to trigger at certain value of a variable, e.g., `0xC01B` or `0x0111`. You only need to change one setting. Note that only the DWT Comparator 1 allows Data value configuration.

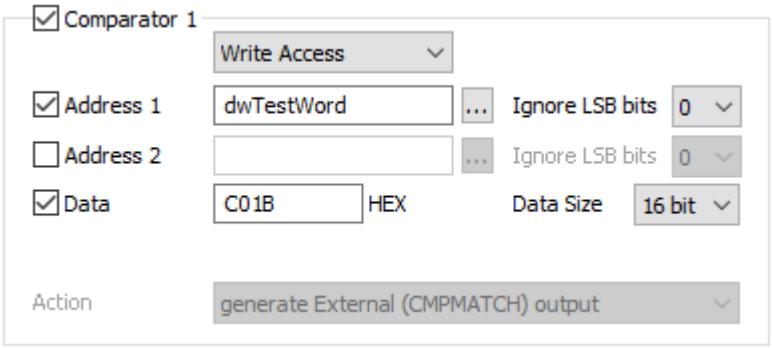
Also note that *Data* comparison can only lead to ETM Trigger. You cannot use it as a qualifier to see only this value.

1. Repeat the steps from the use case *Trigger on a write to a Variable*.

2. Set the Data value.

   i. Navigate to the DWT CORE0 page via *View menu / Analyzer / Analyzer Configuration button*  .

   ii. Enable *Data* and enter the HEX value.



➢ See productivity tip Configure Address 2.

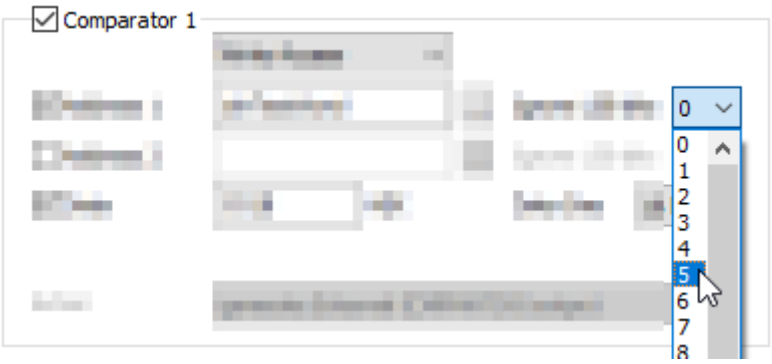
3.1.1 Results of the recording

Trigger was inserted in the trace stream when the configured variable was written with the configured value. In the previous example the Trigger was inserted on the first write to variable `dwTestWord`.

	-1.10	0000'0B18	0000463B	ResetStrX(&sX2, 2);	-162.013384 ms
	-1.11	0000'0AD0	0000B480	{ ResetStrX	-162.013318 ms
	-1.12	0000'0ADA	0000687B	pS->m_a = iTo;	-162.013236 ms
	-1.13	0000'0AE0	0000687B	pS->m_pS = 0;	-162.013186 ms
	-1.14	0000'0AE6	0000687B	pS->m_u.m_l = (long)iTo;	-162.013136 ms
	-1.15	0000'0AEC	0000683B	pS->m_c[0] = pS->m_c[1] = (char)iTo;	-162.013119 ms
	-1.16	0000'0AFC	0000BF00	}	-162.013072 ms
T	1.0	0000'0B04	00004770	ResetStrX_EXIT	-162.013048 ms
	1.1	0000'0B22	0000463B	sX1.m_pS = &sX2;	-162.012760 ms
	1.2	0000'0B26	00004B04	++iCounter;	-162.012715 ms
	1.3	0000'0B30	0000BF00	}	-162.012603 ms
	1.4	0000'0B36	0000BD80	Type_Mixed_EXIT_	-162.012535 ms
	1.5	0000'1258	00004B40	dwTestWord++;	-162.012512 ms
	1.6	0000'1264	FC78F7FF	Address_GlobalVariables();	-162.012471 ms
	1.7	0000'0000	00000000	[ETM] Trigger	-162.012464 ms
	1.8	1FFE'001C	0000C01B	dwTestWord [ITM] Data Write	-162.012384 ms
	3.0	0000'0B58	0000B580	{ Address_GlobalVariables	-162.012136 ms
	3.1	0000'0B5C	00004B05	++iCounter;	-162.012119 ms

4 Trigger on an Address Range

If it's a certain element of an Array, its address within could be used via option Ignore LSB (*Address*) bits.



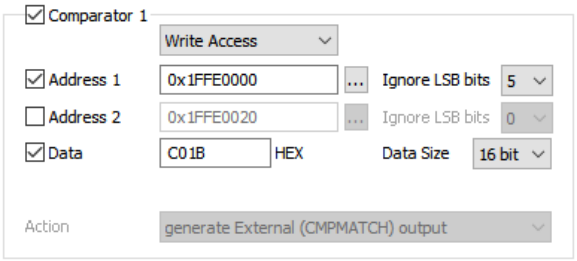
1. Repeat the steps from the use case *Trigger at a value of a Variable*.

2. Configure Ignore LBS bits.

   i. Navigate to the DWT CORE0 page via *View menu / Analyzer / Analyzer Configuration button*  .

   ii. Set the *Ignore LSB bits* option via the drop-down menu to e.g., number **5**.

   Read Address 1 (0x1FFE0000) yields string 0b00011111111111110000000000000**00000**. The red bits get ignored – any 0 or 1 there is a valid trigger. If you imagine these 5 bits to "1", you get 0x1FFE001F. In other words, every write access between 0x1FFE0000 and 0x1FFE001F would be a valid trigger. Because of Data comparator is set to 0xC01B (and correct 16-bit size for write), trigger position within result seems still correctly depicted.



4.1.1 Results of the recording

This setting for Comparator 1 yields a correct result - dwTestWord is starting from 0x1FFE'001C on.

T	0.0	0000'0AE6	0000687B	pS->m u.m 1 = (long)iTo;	-162.082376 ms
	0.1	0000'0AEC	0000683B	pS->m_c[0] = pS->m_c[1] = (char)iTo;	-162.082368 ms
	0.2	0000'0AFC	0000BF00	}	-162.082347 ms
	0.3	0000'0B04	00004770	ResetStrX_EXIT_	-162.082336 ms
	0.4	0000'0B22	0000463B	sX1.m_pS = &sX2;	-162.082040 ms
	0.5	0000'0B26	00004B04	++iCounter;	-162.081995 ms
	0.6	0000'0B30	0000BF00	}	-162.081883 ms
	0.7	0000'0B36	0000BD80	Type_Mixed_EXIT_	-162.081815 ms
	0.8	0000'1258	00004B40	dwTestWord++;	-162.081792 ms
	0.9	0000'1264	FC78F7FF	Address_GlobalVariables();	-162.081758 ms
	0.10	0000'0000	00000000	[ETM] Trigger	-162.081752 ms
C	0.11	1FFE'001C	0000C01B	dwTestWord [ITM] Data Write	-162.081664 ms
	2.0	0000'0B58	0000B580	{ Address_GlobalVariables	-162.081416 ms
	2.1	0000'0B5C	00004B05	++iCounter;	-162.081400 ms

With the *alternate* setting we get different results because of the complex trigger setting. Value 0x0111 was used as an additional requirement for the trigger. Like depicted the 2nd value could be the real trigger as 32-bit access, which contained our value as low word 0x5400111 within variable _edata (located at 0x1FFE'0004), which is within the configured address *range*. The *expected trigger* occurs about 8300 trace lines later within recording.

	-1.8	1FFE'001E	00000089	dwTestWord [ITM] Data Write	10.236448 ms
T	1.0	0000'11D8	00004B40	dwTestWord++;	10.236616 ms
	1.1	0000'11E4	FCB8F7FF	Address_GlobalVariables();	10.236731 ms
	1.2	0000'0B58	0000B580	{ Address_GlobalVariables	10.236750 ms
	1.3	0000'0B5C	00004B05	++iCounter;	10.236788 ms
	1.4	0000'0B66	FFE9F7FF	TestStatic();	10.236884 ms
	1.5	0000'0B3C	0000B480	{ TestStatic	10.236904 ms
	1.6	0000'0B40	00004B04	++staticI;	10.236912 ms
	1.7	0000'0B4A	0000BF00	}	10.236935 ms
1	1.8	0000'0000	00000000	[ETM] Trigger	10.236944 ms
	1.9	0000'0B50	00004770	TestStatic_EXIT_	10.237114 ms
2	2.0	1FFE'0004	00000111	_edata [ITM] Data Write	10.237200 ms
	2.1	1FFE'0004	00000112	_edata [ITM] Data Write	10.237360 ms

	Number	Address	Data	Content	Time
	8308.0	0000'0B04	00004770	ResetStrX_EXIT_	18.038464 ms
	8308.1	0000'0B22	0000463B	sX1.m_pS = &sX2;	18.038752 ms
	8308.2	0000'0B26	00004B04	++iCounter;	18.038797 ms
	8308.3	0000'0B30	0000BF00	}	18.038909 ms
	8308.4	0000'0B36	0000BD80	Type_Mixed_EXIT_	18.038977 ms
C	8308.5	1FFE'001E	00000111	dwTestWord [ITM] Data Write	18.039000 ms
	8310.0	0000'11D8	00004B40	dwTestWord++;	18.039168 ms

➢ See productivity tip Watch all accesses.

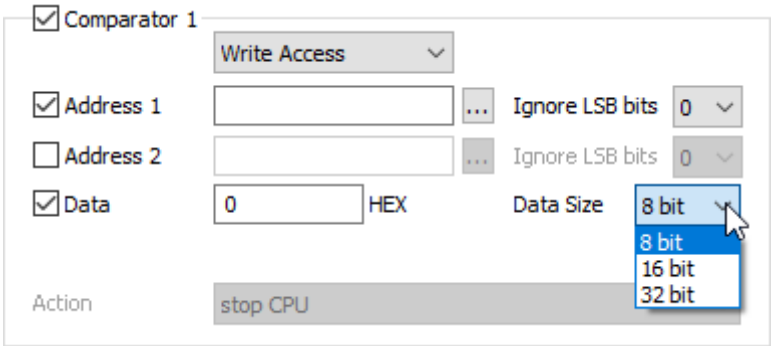
5 Productivity tips

5.1 Data access sizes

Note that Cortex-M can have 8, 16- or 32-bit Data access size, which you need to take into consideration to get a correct Trigger setup. This size could be MCU and Compiler declaration dependent and the Watch window shows you the real width according to the information in the Download file.

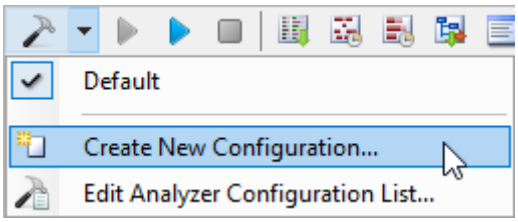
Name	Value	Type
dwTestWord	0x0000	unsigned short

To check whether a certain value occurs during application runtime, you can configure access breakpoint and see if the core gets stopped. Access breakpoint can be configured via *Debug / Hardware Breakpoints... / DWT* page.



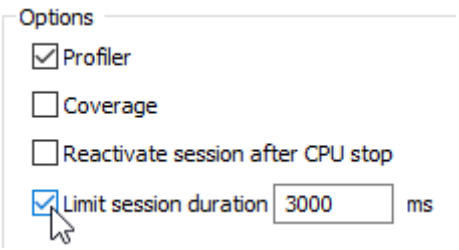
5.2 Default Trigger

Don't apply changes to the *Default* Trigger, because it is always a working reference that you should keep.



5.3 Profile and limit the session duration

Optionally you can check the Profiler via the *Analyzer Configuration* button  and limit the recording time via option *Limit session duration*. Whatever will be true first – *Recording Size Limit* or *Limit session duration* – will stop the recording.



5.4 Search for variables via the Symbol Browser

Use the ... button to open the Symbol Browser and select a variable.

5.5 Configure unused DWT Comparators

Two DWT Comparators are left in case you require additional variable watch.

5.6 Configure Address 2

You can use *Address 2* field for another address written with that value in parallel. If you enter data value in HEX, you have type here the syntax with “0x” prefix. To stay backwards compatible the entry in Data field is always HEX.

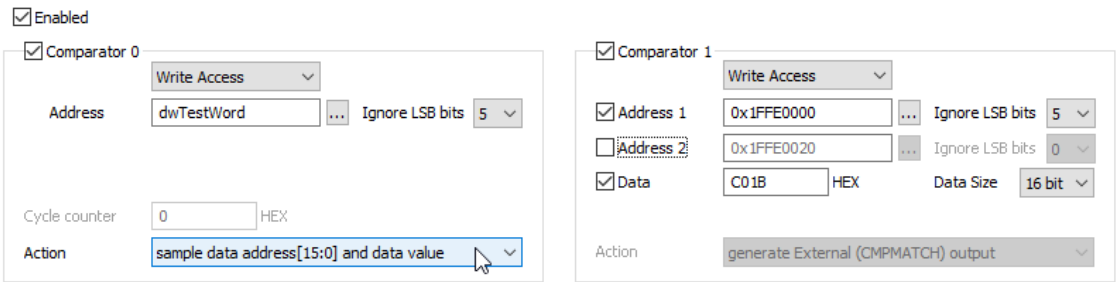
Note that Address 2 field is not meant as from address 1 to address 2, but like e.g., if certain Data get written onto Address 1 or Address 2, trigger is valid.

Make sure that your trigger address and value combination occur within selected or possible recording time.

5.7 Watch all accesses

To watch all accesses with the *Ignore LSB bits* option setting, you need to repeat this setup on Comparator 0.

To display the addresses beside values as well, change the Action option to sample data address [15:0] and data value. Note that this configuration could lead to overflow.



5.7.1 Results

	Number	Address	Data	Content	Time
	-1.13	0000'0AE0	000000687B	pS->m_pS = 0;	10.379892 ms
	-1.14	0000'0AE6	000000687B	pS->m_u.m_l = (long)iTo;	10.379944 ms
	-1.15	0000'0AEC	000000683B	pS->m_c[0] = pS->m_c[1] = (char)iTo;	10.379960 ms
	-1.16	0000'0AFC	00000BF00	}	10.380002 ms
T	1.0	0000'0B04	00000004770	ResetStrX_EXIT_	10.380024 ms
	1.1	0000'0B22	000000463B	sX1.m_pS = &sX2;	10.380320 ms
	1.2	0000'0B26	00000004B04	++iCounter;	10.380365 ms
	1.3	0000'0B30	00000BF00	}	10.380477 ms
	1.4	0000'0B36	00000BD80	Type_Mixed_EXIT_	10.380545 ms
	1.5	0000'1258	00000004B40	dwTestWord++;	10.380568 ms
	1.6	0000'1264	00FC78F7FF	Address_GlobalVariables();	10.380602 ms
	1.7	0000'0000	00000000000	[ETM] Trigger	10.380608 ms
C	3.0	1FFE'001C	000000C01B	dwTestWord [ITM] Data Write	10.380736 ms
	3.1	0000'0B58	000000B580	{ Address_GlobalVariables	10.380944 ms
	3.2	0000'0B5C	00000004B05	++iCounter;	10.380971 ms
	3.3	0000'0B66	00FFE9F7FF	TestStatic();	10.381041 ms
	3.4	0000'0B3C	000000B480	{ TestStatic	10.381054 ms
	3.5	0000'0B40	00000004B04	++staticI;	10.381082 ms
1	3.6	1FFE'0004	00025698035	_edata [ITM] Data Write	10.381152 ms
	5.0	0000'0B4A	00000BF00	}	10.381192 ms
	5.1	0000'0B50	00000004770	TestStatic_EXIT_	10.381378 ms
2	5.2	1FFE'0004	00025698036	_edata [ITM] Data Write	10.381440 ms
	7.0	0000'0B6A	00FFE7F7FF	TestStatic();	10.381480 ms

But if you change to *Ignore LSB bits*, you will get results like depicted in the picture below and in the 2nd example in chapter 4.

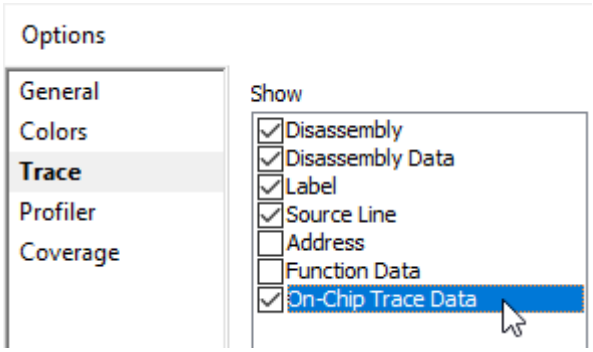
Trace

	Number	Address	Data	Content	Time
T	0.0	0000'0AE6	0000687B	pS->m_u.m_l = (long)iTo;	672.961816 ms
	0.1	0000'0AEC	0000683B	pS->m_c[0] = pS->m_c[1] = (char)iTo;	672.961824 ms
	0.2	0000'0AFC	0000BF00	}	672.961845 ms
	0.3	0000'0B04	00004770	ResetStrX_EXIT_	672.961856 ms
	0.4	0000'0B22	0000463B	sX1.m_pS = &sX2;	672.962144 ms
	0.5	0000'0B26	00004B04	++iCounter;	672.962197 ms
	0.6	0000'0B30	0000BF00	}	672.962332 ms
	0.7	0000'0B36	0000BD80	Type_Mixed_EXIT_	672.962413 ms
	0.8	1FFE'001E	0000600E	dwTestWord [ITM] Data Write	672.962440 ms
	2.0	0000'11D8	00004B40	dwTestWord++;	672.962608 ms
	2.1	0000'11E4	FCB8F7FF	Address_GlobalVariables();	672.962707 ms
	2.2	0000'0B58	0000B580	{ Address_GlobalVariables	672.962723 ms
	2.3	0000'0B5C	00004B05	++iCounter;	672.962756 ms
	2.4	0000'0B66	FFE9F7FF	TestStatic();	672.962839 ms
	2.5	0000'0B3C	0000B480	{ TestStatic	672.962856 ms
	2.6	0000'0B40	00004B04	++staticI;	672.962864 ms
	2.7	0000'0B4A	0000BF00	}	672.962887 ms
	2.8	0000'0000	00000000	[ETM] Trigger	672.962896 ms
	2.9	0000'0B50	00004770	TestStatic_EXIT_	672.963061 ms
C	4.0	1FFE'0004	01EAC01B	_edata [ITM] Data Write	672.963144 ms
	4.1	1FFE'0004	01EAC01C	_edata [ITM] Data Write	672.963312 ms

5.7.2 General note

More complex triggers and *Ignore LSB bits* setup could cause the MCU suppressed i.e., trigger information and other not essential information. To find what you are looking you can:

- Search within Trace results.
- Check option *On-Chip Trace Data* in *Trace window / Options / Trace* which will display origin messages sent from MCU within the Trace pane.



6 Technical support

6.1 More resources

Online Help ▸ winIDEA and testIDEA help	Knowledge Base ▸ Tips & tricks categorized by issue type and architecture	Tutorials ▸ From beginner to expert
Technical Notes ▸ How-tos for winIDEA functionalities with scripts	Application Notes ▸ How-to notes on advanced use cases	Webinars ▸ Technical webinars about iSYSTEM tools with use cases

6.2 Contact

Please visit https://www.isystem.com/contact.html for contact details.