XDS560 Trace

{{#switchcategory:MSP430=<McuHitboxHeader/>|C2000=<McuHitboxHeader/>|Stellaris=<McuHitboxHeader/>|TMS570=<McuHitboxHeader/>|MCU=<McuHitboxHeader/>|MAVRK=<MAVRKHitboxHeader/>|<HitboxHeader/>}}

Contents

What is XDS560 Trace?

The XDS560 Trace is the emulation technology that performs real-time gathering of all instructions being executed in a processor - also called *Core Trace* or *Instruction Trace*. This process is done through specialized hardware embedded into the device and requires additional JTAG pins to support the high data throughput.

> **Note:** this is different than *System Trace*, which is the emulation technology that monitors synchronization and timing between cores and on-chip peripherals. For these types of traces please check the System Trace page.

The data is gathered in real-time and the XDS560 Trace pod performs the tasks of data compression and buffering (its size is configurable). Once the data gathering is finished, it is decompressed and sent to the host PC running Code Composer Studio and therefore available for post processing and analysis by the developer.

What is XDS560 Trace useful for?

Find previously "invisible" complex, intermittent, context-sensitive real-time bugs

- Detect scheduling issues, intermittent glitches, false interrupts and more without stopping the processor
- XDS560 Trace can be access by using the Advanced Event Triggering capability which is integrated with Code Composer Studio through the Unified Breakpoint Manager

Fine tune code performance and cache optimization of complex switch intensive multi-channel applications

- Real-time code and event profiling
- Fast and accurate code analysis with profiling, cache view and code coverage
- Support available today on: C641x, DM64x, C6455, TNETV3020, TCI6487, TCI6488, TMS320C6474, ARM9/ARM11 (Embedded Trace Buffer only). Please check your datasheet for details.

There is a summary presentation here that describes how XDS560 Trace works, what it is useful for, and provides some examples. : Trace_Introduction-ext-01.pdf

For use in Code Composer Studio, see: Unified Breakpoint Manager

How large of a Trace Buffer should be used?

The XDS560 trace pod comes with 64MB of Buffer memory. However, you can configure trace to use any of the following values: 256Kb, 512kB, 1Mb, 2Mb, 4Mb, 8Mb, 16Mb, 32Mb, 64Mb.

Why not just use the largest buffer (64Mb)?

With a 64Mb buffer, you will capture the maximum amount of data. Keep in mind, though, this is 64Mb of compressed data. It is estimated that a 64Mb buffer could potentially hold 1.5 Billion PC Trace data samples, which could require up to an estimated 135 GBs of disk space for the Trace Display to decompress, with estimated decode time on the order of 12 hours. And then, once it's decompressed, there is still more processing to do to make the data useful.

The problem with the existing trace display is that it decodes the entire buffer and tries to display it. It must keep all samples around, and thus uses up gigabytes of disk space storing it all. There will be other decoding solutions that provide for simultaneous decoding and processing of the data that will alleviate the disk space requirement. However, now there is more burden on the processing application to gather the necessary data prior to discarding each piece of decoded data. This can have an impact on the decoding time.

The recommendation is to start with a small buffer and see if it gets you enough data. If it does not, gradually increase the size of the data and run again until you see desired results. Efficient use of AET triggers allows trace to capture only data that is interesting to your scenario and to eliminate data that is not applicable. Clever triggering can allow the user to reduce the size of the capture buffer.

Using the Embedded Trace Buffer (ETB)

Some devices feature the Embedded Trace Buffer. If you use the ETB you do not need the XDS560 Trace hardware, just a JTAG connection through an emulator to use Trace.

XDS560 Trace Pod Technical Requirements

The XDS560 Trace unit utilizes the 60-pin TI JTAG Connectors. The 60-Pin Emulator Header User's Guidedescribes the 60-pin connector and target board electrical requirements for XDS560 Trace and JTAG support. Also see your Emulator manufacturer's documentation for emulator specific information. Users may also want to read the "Common Trace Transmission Problems and Solutions" at tidoc:SPRAAK6.

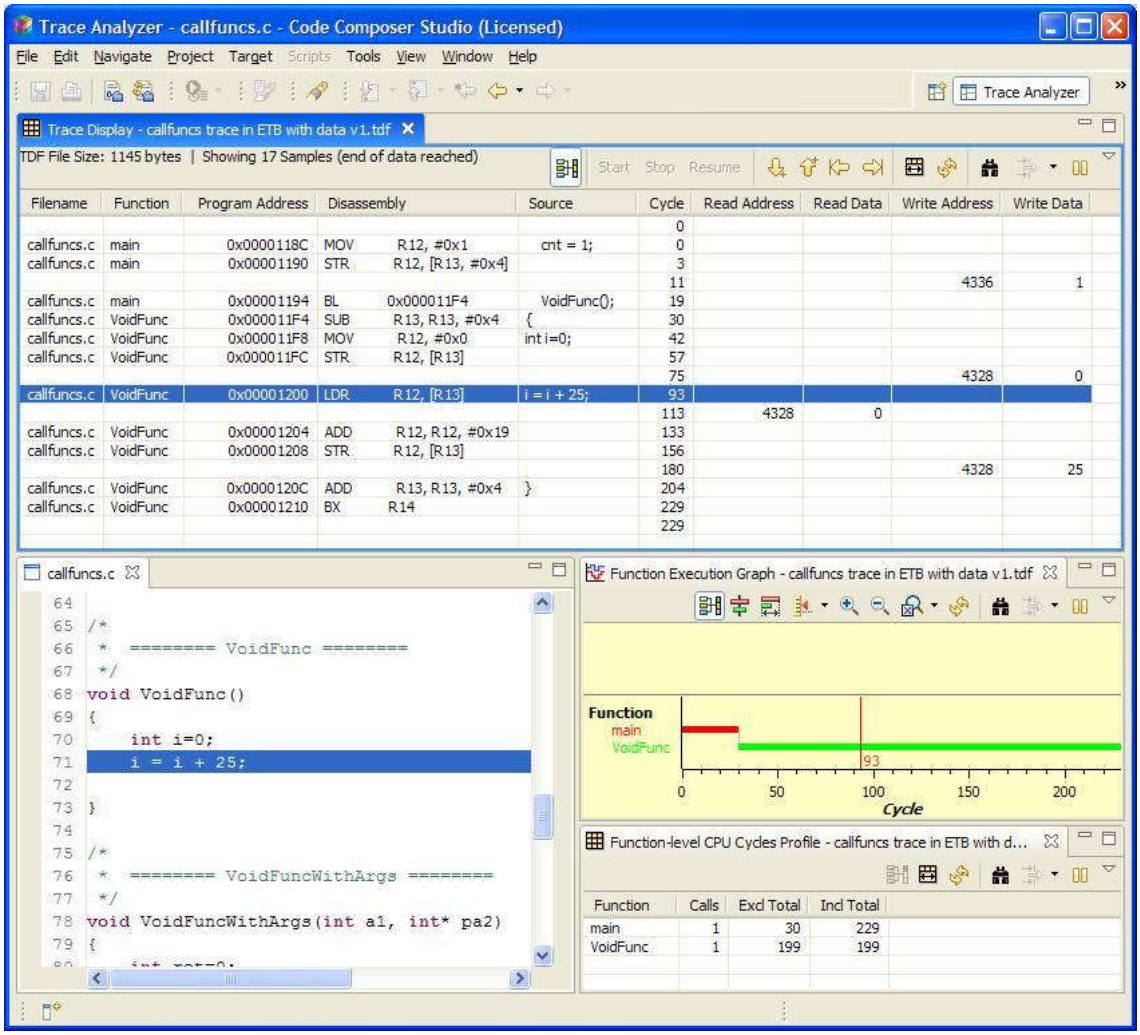
Debugging with Trace

Detailed documentation on the trace system, trace configuration and setting up trace jobs using Standard (PC, data and timing) and Event trace for debug can be found at Debugging with Trace.

Debugging with Trace provides:

- An introduction to trace
- How and when to use trace (includes Trace System Limitations discussion)
- Technical requirements for your board
- What's in the device to facilitate trace
- What's in CCS to facilitate trace
- Debug and profile examples

XDS560 Trace Display



- Trace Analyzer showing program and data tracing. Source code correlation and source file secondary display in lower window. There is also a function graph and a function level profile.

XDS560 Trace Software Versions

- The XDS560 Trace software is included with Code Composer Studio v4.2.

Conference Presentations

- 2007 TI Developers Conference (India) - 29-30 November 2007 (Bangalore)
  - Trace Overview and Use Cases

- Topics
  - AET/Trace Overview
  - Statistical Profiling
  - Interrupt Profiling
  - Thread Aware Profiling
  - Thread Aware Dynamic Function Call Graph

- 2007 TI Developers Conference (India) - 29-30 November 2007 (Bangalore)
  - Advanced Profiling Use Cases with XDS560 Trace
    - Topics
      - AET/Trace Overview
      - Statistical Profiling
      - Interrupt Profiling
      - Thread Aware Profiling
      - Thread Aware Dynamic Function Call Graph

Publications & Articles

- Trace Exposes the Toughest Real Time Bugs (http://www.embedded.com/design/testissue/201803561?_requestid=907033) (Sept 2007, Embedded Systems Design)
- XDS560 Trace press Release [1] (http://focus.ti.com/pr/docs/preldetail.tsp?sectionId=594&prelId=sc07053)
- Advanced Profiling with XDS560 Trace: SPRAAL8
- Common Trace Transmission Problems and Solutions SPRAAK6
- 60 pin trace header guide spru655 also see 60 Pin Trace Header
- JTAG Connectors

Related

- XDS560
- XDS560v2 System Trace
- XDS510
- XDS100
- AET Logic for the C6000 family
- Debugging With Trace
- Advanced Event Triggering

{{

1. switchcategory:MultiCore=
   - For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
   - For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **XDS560 Trace** here.

Keystone=
- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **XDS560 Trace** here.

C2000=*For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **XDS560 Trace** here.*

DaVinci=*For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article **XDS560 Trace** here.*

MSP430=*For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **XDS560 Trace** here.*

OMAP35x=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **XDS560 Trace** here.*

OMAPL1=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **XDS560 Trace** here.*

MAVRK=*For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article **XDS560 Trace** here.*

*For technical s please post yo questions at http://e2e.ti.cor Please post on comments abo article **XDS560 here.***

}}

Links

Amplifiers & Linear
Audio
Broadband RF/IF & Digital Radio
Clocks & Timers
Data Converters
DLP & MEMS
High-Reliability
Interface
Logic
Power Management
Processors
- ARM Processors
- Digital Signal Processors (DSP)
- Microcontrollers (MCU)
- OMAP Applications Processors

Switches & Multiplexers
Temperature Sensors & Control ICs
Wireless Connectivity

{{#switchcategory:MSP430=<McuHitboxFooter/>|C2000=<McuHitboxFooter/>|Stellaris=<McuHitboxFooter/>|TMS570=<McuHitboxFooter/>|MCU=<McuHitboxFooter/>|MAVRK=<MAVRKHitboxFooter/>|<HitboxFooter/>}}
