



System Trace Macrocell Packs Major Benefits for High-Performance SoC System Debug

WP-01229-1.0

White Paper

The CoreSight™ System Trace Macrocell (STM) is superior to an Instrumentation Trace Macrocell (ITM) in seven significant ways. From performance to quality of data, an STM is the more suitable debug macrocell for modern day SoC FPGAs that incorporate ARM® Cortex®-A9 application processors within high-performance FPGA fabrics. This paper explains these differences.

Introduction

When a system works as expected, debug tools and debug architectures can be big ‘don’t cares.’ But as soon as things go awry, the debug circuitry and software tools become crucial as engineers race to zero-in on root causes and performance inefficiencies.

When designing with SoC FPGAs, systems can get complex; in addition to the dual-core processor, SoC FPGA engineers have an array of custom logic to work with. Anyone who has debugged an embedded system knows that an increase in system complexity can propel the debug phase from minutes to hours to days to weeks. But that doesn’t have to be the case. With the right tools and debug architecture, an SoC FPGA-based system can be surprisingly simple to navigate.

This white paper contrasts specific debug circuitry implemented in two different ARM Cortex-A9 processor-based FPGA device families: ARM CoreSight STMs, which can be found in all Altera® SoCs, versus ARM CoreSight ITMs, which are included in Xilinx’s Zynq devices.

System Trace Macrocells

A System Trace Macrocell is a debug circuitry that processes trace streams originating from anywhere within the device.

“An STM provides non-intrusive and time-stamped software instrumentation of the kernel and user space, enabling software developers to gain more visibility on how their software execute in their software without altering the behavior of the system” (1)

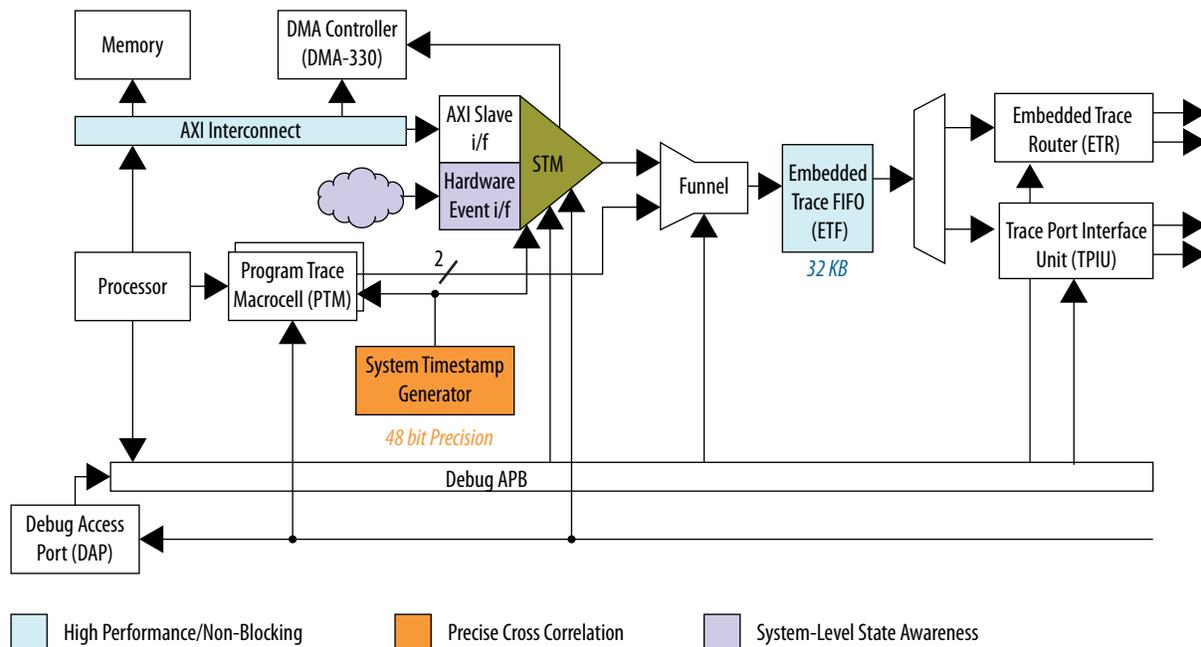


101 Innovation Drive
San Jose, CA 95134
www.altera.com

© 2014 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, ENPIRION, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera’s standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



Figure 1. Altera SoC STM Implementation Diagram



The Differences Between STM and ITM

At a high level, both STMs and ITMs process incoming debug trace from upstream CoreSight components, add timestamps, and send it on to the device Debug Access Port (DAP). The similarities, however, end there. In fact, these circuits are in totally different classes. STMs were designed specifically for high-speed multicore microprocessor-based systems much like the current class of SoC FPGAs. ITMs, on the other hand, were designed for microcontroller trace.

 Debug circuitry in Altera SoCs and Xilinx Zynq devices are drastically different. Architecture matters.

- All Altera SoCs contain CoreSight STMs, which were designed for ARM Cortex-A9 processor-based systems. STMs are in Cyclone® V SoC, Arria® V SoC, and Arria 10 SoC.
- Xilinx Zynq devices contain ITMs, which are a poor fit in an ARM Cortex-A9 processor-based system as they were designed for microcontrollers.

Differences between these debug architectures are significant. The remainder of this white paper explains how each difference might affect an embedded developer debugging a complex embedded system.

- **Intended use.** CoreSight STMs were designed specifically for ARM Cortex-A9-based systems. They run at the A9 processor speed and are capable of managing debug data from multiple cores. ITMs on the other hand were designed originally for microcontroller debug. They run at low speed, and get overloaded quickly.

- **Tool chain.** The SoC Embedded Design Suite (EDS) is included in every Altera SoC development kit. It contains ARM Development Studio 5 (DS-5™) Altera Edition Toolkit software. The DS-5 provides a rich and optimized debugging environment that makes full use of STM trace data. Conversely, the debugging tool included in the Xilinx SDK lacks the ability to access ITM data. ⁽²⁾ In order to access ITM trace, a developer would need to obtain additional tools from third-party vendors.
- **STMs do not drop data.** A STM can process all data received. Data comes in on a dedicated AXI bus. (A separate APB interface carries macrocell programming information). The STM can not only buffer incoming data, but signal back-pressure when needed. Conversely, ITM trace comes in on a shared APB. The ITM is unable to buffer much data, and cannot signal back-pressure. It can, however, arbitrate between incoming trace priority. ⁽³⁾ ITM configuration data and instrumentation trace share the same APB bus. When the ITM buffer is full, instrumentation trace is dropped. While the ITM does have a polling mechanism to help avoid data loss, it can be a burden for software. An STM, on the other hand, can receive trace from multiple sources effortlessly. ⁽⁴⁾ Upstream macrocells can simply write their data to the STM without negotiating with other sources. A STM can handle lots of bursty data from lots of sources and render all of it while ITMs can drop data. When you use trace, you need to be sure that the relevant information is captured. In a high-complexity, high-performance system, this is guaranteed only with STM.
- **STMs are high in performance (ITMs are slow).** A dedicated AXI™ slave, running at the SoC FPGA processor speed (400 MHz+), sends trace data to the STM. The STM is capable of receiving 4 KB bursts. The STM architecture is designed for high-performance systems. Conversely, anyone familiar with ARM architectures knows that APB is for low-bandwidth control signaling. Compounding the slow, shared nature of the APB architecture, it is worth noting that incoming data can only be received every other clock cycle. Furthermore, the ITM is restricted to writing a maximum of 8 bits at a time to the debug access port. Compare that with 32 bit output on the STM running at the Cortex-A9 processor clock speed. Between bus performance and transactional flexibility, STMs can run from 60X to 150X the rate of ITMs.
- **STMs are extensible and customizable.** Anyone wishing to build a CoreSight-compliant structure in the Altera SoC fabric can. Altera has included a linked list header in the DAP ROM Table that gives the STM visibility into trace data coming from the FPGA. ⁽⁵⁾ The DS-5 Altera Edition Toolkit renders all of the trace data these 'soft logic' CoreSight components produce. When using the Xilinx solution, users have to design their own method for transmitting custom debug data, and then manually correlate trace data across the processor subsystem and FPGA subsystem.
- **STM time-stamping correlates precisely and automatically.** All CoreSight compliant data structures such as Program Trace Macrocells (PTMs) and Embedded Trace Macrocells (ETMs) that have been built into Altera SoCs can send data with highly accurate 48 bit timestamps. After the STM processes time-stamped data, the outgoing packets retain all of the original precision. This allows for highly accurate cross-correlation of data across multiple processing units.

Conversely, ITM timestamps are 21 bit, but this is not all. There is no global timestamp input on the CoreSight ITM and as such there is no straightforward mechanism for correlating with other traces. This all makes for 'coarse grained' cross-correlation of data, and introduces uncertainty. When searching for root cause, uncertainty of when events happened can be maddening.

- **STMs are system-state aware.** This means that outgoing data can be tagged with information about what state the system was in. For example, these could include (but are not limited to) low-power state, memory transaction error correction code (ECC) flag or any 'system state' relevant to the application. A designer can program the STM to recognize up to 32 user-defined states. When investigating trace data at a process level, it can be immensely helpful to know that a previous memory access encountered an ECC error, or that the trace was generated during system 'wake up.' Unlike STMs, ITMs have no native capacity for correlating trace data to system state.

Key Advantages of CoreSight STM vs. ITM

Table 1 summarizes key advantages of Altera's CoreSight STM and the SoC EDS software versus the CoreSight ITM and the Xilinx Software Development Kit (SDK):

Table 1. Key Advantages of the CoreSight STM and the SoC EDS Software

	CoreSight STM and SoC EDS Advantages	CoreSight ITM and SDK Disadvantages
Intended Use	STMs are designed for debugging high-performance multicore Cortex-A9 processor-based systems.	Originally designed for debugging microcontroller-based systems.
Tool Chain	The DS-5 Altera Edition Toolkit is included in the SoC EDS as part of standard tool chain offering. No additional debug tools are required.	Developer must obtain 3 rd party debug tools.
Percentage of Trace Data Captured	STMs are non-blocking, have a dedicated AXI trace bus and can signal back-pressure. 100% of trace data is captured.	ITMs drop data. Incoming trace data is arbitrated on shared APB bus; ITMs have small data buffers and can drop data.
Trace Performance	High-performance dedicated AXI trace runs at embedded ARM processor speed and supports 4KB burst.	Low-performance shared APB trace runs at 66 MHz. Input is 32 bit (every other cycle). Output is 8 bit. No bursts.
Customizability	STMs can be made aware of CoreSight-compliant debug circuitry implemented in the FPGA. This 'custom' trace is added automatically to STM debug trace stream.	ITMs cannot incorporate trace from FPGA region. Developers must roll their own solution.
Precision	Highly precise timestamp (48 bit) = 'fine grained' and accurate cross correlation between events.	Low precision timestamp (21 bit) results in 'coarse grained' cross correlation. ⁽¹⁾
System-State Awareness	User defined 'state' awareness. Debug packets can be tagged with any number of custom 'states'. For example, 'low power mode.'	No system-level state awareness.

Note:

(1) 28.2.4 ITM www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf

Conclusion

CoreSight STMs combined with the DS-5 Altera Edition Toolkit provide powerful debugging resources. With these tools, an engineer can quickly understand what is going on within a complex FPGA based embedded system and shorten the debug phase.

Take a look under the hood for yourself. Compare CoreSight STMs and ITMs by reading the materials listed below, or dive right in and begin developing with a Cyclone V SoC or Arria V SoC Development Kit by ordering one today at buy.altera.com.

Further Information

SoC EDS and ARM DS-5 Altera Edition Toolkit

- ARM DS-5 Altera Edition Toolkit video
www.youtube.com/watch?v=HV6NHR6gLx0
- White Paper: *FPGA-Adaptive Software Debug and Performance Analysis*
www.altera.com/literature/wp/wp-01198-fpga-software-debug-soc.pdf
- ARM DS-5 Altera Edition Toolkit
www.altera.com/devices/processor/arm/cortex-a9/software/proc-arm-development-suite-5.html

CoreSight Architectural Details (STMs vs ITMs)

- About the System Trace Macrocell
infocenter.arm.com/help/topic/com.arm.doc.ddi0444b/CACEBJCA.html
- Hardware Brief: *Arria V SoC CoreSight Debug and Trace*
www.altera.com/literature/hb/arria-v/av_54007.pdf

References

1. arm.com/products/system-ip/debug-trace/trace-macrocells-etm/coresight-system-trace-macrocell.php
2. www.zedboard.org/content/zynq-coresight-debug
3. infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0439d/BIIFFGJD.html
4. 128 masters, each supporting 65,536 stimulus ports, enable significant scalability, with 16 stimulus ports per 4KB page. Stimulus ports are also known as channels.
5. infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0316d/Bjfheije.html

Acknowledgements

- Balatripura Chavali, Design Engineer, Silicon Systems Development
- Todd Koelling, Senior Marketing Manager, SoC Product Marketing, Altera Corporation
- Laura Reese, Senior Product Marketing Manager, SoC Product Marketing, Altera Corporation

Document Revision History

Table 2 shows the revision history for this document.

Table 2. Document Revision History

Date	Version	Changes
September 2014	1.0	Initial release.