SourcePoint™ with Intel® Trace Hub

SourcePoint ASSET InterTech's flagship software debugger has now added some power for Intel architecture tracing by adding support for Intel's Trace Hub. SourcePoint debugger now brings the power of the Intel Trace Hub and Intel Processor Trace (PT) to speed your insight to your toughest software problems.

SourcePoint is the debugger of choice for many Unified Extensible Firmware Interface (UEFI) developers including Intel and now with the addition of Trace Hub tracing code execution and finding that elusive bug is now but a trace away.

TraceHub is an implementation in silicon, of enabling Trace from various hardware and software sources. This technology employs a MIPI STPv2.1 protocol so that the trace data from these various sources can be encoded and streamed off the system to be captured and displayed in SourcePoint. This is first enhancement, outside of the Real Time Instruction Trace (RTIT) on Atom processors, which provides a meaningful trace without performance penalties for Intel Core and Xeon code developers.

Trace Configurat	race Configuration				
LBR BTS	Intel PT Intel PT Memory Trace Hub				
Masters to tra	Masters to trace				
None					
II (
🔘 List: 🛛 Ex	C List Example: 8, 32-255				
Trace source routing					
SW/FW:	Trace Hub Memory 🚽				
Intel PT:	Memory Direct 👻				
AET:	Emulator Direct 👻				
Trace buffer	Trace buffer				
🔘 Use BIOS	Use BIOS settings				
💿 Use Sourd	O Use SourcePoint settings				
Base addre	ess: 14000000P				
Length:	512k 🔹				
Timestamp ✓ Alignment packets Frequency: CTC16 ▼					
Master / Cha	Master / Channel definitions				
Filename: \BIOS\SkyLake\TraceHub\XML\SkIPorts.xml					
	OK Cancel Help				

Figure 1: Trace Hub Configuration

Trace Hub Setup

SourcePoint manages the TraceHub setup of selecting the trace stream from various sources. The software trace stream sources are Intel Architecture Application Cores, Firmware Cores, and Chipset Cores. The hardware trace streams are from Intel Processor Trace (PT), Architectural Event Trace, and others.

Figure 1 shows how to manage the trace from the various sources, configure filters and timestamp settings. Simple and straight forward.

Intel Processor Trace Setup

SourcePoint provides a means for intuitive configuration of selecting the desired data gathered from the different trace sources. One trace source is Intel PT, and Figure 2 shows how to configure processor trace setting to all cores or selected cores. Adding timestamps allows for easy correlation of data between traces. The final configuation item is to manage the Trace Hub

Trace Configurat	ion	×		
LBR BTS	Intel PT Intel PT Memory Trace Hub			
Processors to	trace	- I		
<u> N</u> one				
(O) (<u>A</u>)				
◯ <u>L</u> ist: Exa	ample: P0, P4-P7			
Share filter / timestamp settings				
Opply settings to all processors				
○ Apply <u>s</u> etti	ings to: P0 👻			
Filters		- I		
🔲 Range <u>1</u> :	Enter symbol or start-end			
🔲 Range <u>2</u> :	Enter symbol or start-end			
<u> </u>	User -			
C <u>B</u> 3:				
Timestamp				
<u>▼ I</u> SC				
<u>м</u> тс	Frequency:			
✓ Cycle accurate Threshold: 0 (fine)				
		-		
	OK Cancel Hel	p		

Figure 2: Intel Processor Trace Configuration



SourcePoint[™] with Intel[®] Trace Hub

buffer. The current trace buffer destination is target memory, Figure 3. This can be controlled by the BIOS or SourcePoint, depending on how and where you want the trace buffer controlled.

Trace Configuration				
LBR BTS Intel PT Intel PT Memory Trace Hub				
Trace buffer				
O Use processor settings				
O Use SourcePoint settings:				
Base address: 10000000P				
Length per core: 256k 🔹				
Trace capture mode				
Overwrite				
Append				
OK Cancel Help				

Figure 3: Intel Processor Trace Memory Configuration

Trace Display and Analysis

The data can be displayed in various forms. Raw trace, with or without timestamps, with assembly, source or both. Grabbing raw trace can be informative, but does not necessarily bring clarity to the problem. Often more information about the data collected needs to be added.

Displaying more information within the trace will provide for more clarity, but in some instances adds so much data the code seems to get lost.

SourcePoint has the ability to link the trace data to the specific code source providing clarity in understanding the code execution flow, whether it be from a processor core, chipset cores, the Software HUB or other hardware sources. Figure 4 is an example of Code view with Intel Processor Trace interlocked to provide detail with clarity.



Figure 4: Trace with Code Locked View



SourcePoint[™] with Intel[®] Trace Hub

System Trace Analysis

Another benefit of trace hub is a system view of code execution. By using other trace sources like, chip set and software hub, we can expand the time window so that a better view of system execution can be provided. In the previous example, the trace covered milliseconds of execution time. In Figure 5, the ME (chip set) and UEFI (software hub) trace could cover seconds of execution. This allows for a trace capture of a full boot sequence depending upon the target memory buffer size.

So the user, with a minimal amount of instrumentation, can trace progress of a varity of items such as hardware registers, from another piece of silicon in the target such as an FPGA, or trace FIFO, or messsage handles, or even message streams, and SourcePoint can provide clear, precise trace views all with time stamp correlation exposing the system effects on the code while simultaneously tracking all these forms of data. By assigning master and channel, via a editable XML file, the developer has total control over what data is trace on a given channel. The data can be color coded, providing even more clarity into code exectuion flow from the systems perspective. Multiple examples of XML for assigning or formatting are provided with the online help.

SourcePoint provides the developers the ability to flex their creativity and provide trace views into the system execution that match the needs of the developer.

SourcePoint for IA

SourcePoint offers a number of unique but highly intuitive features, making it an exceptional debugger for today's Intel Architecture projects.

A good debug solution offers the programmer or developer layers of debugging options, from the simple breakdown of code into ever smaller iterations to the ability to capture and analyze huge amounts of execution history in a single run. With the addition of Trace Hub the ability to trace code execution without negative side effects has finally been achieved and puts real muscle into trace for the Intel architectures thanks in part to SourcePoint.

F Trace Hub - SW/FW Trace (time aligned)		
STATE ADOR INSTRUCTION		TINESTARP
POSTCODE-(000000a7)		
PROGRESS CODE: V2070003 10		-560.020 MS (+04.114 US)
-12791 UEFT: DEEUG POSTCODE+<96 3		-552.181 ms (+2.612 ms)
-12772 UEFT: DEBUG POSTCODE=(00000036)		-552.134 ms (+47.271 us)
-12744 UEFI: DEBUG		-552.050 ms (+12.952 us)
-12677 UEFI: DEBUG		-538.913 ns (+2.226 ns)
-12244 UEFI:DEBUG	B63F02C7-A9C9-4472-A4C0-4D0AF365CC51 906	-430.111 ns (+1.803 ns)
Loading driver D22C15F5-81 -12215 UEFI:DEBUG	8B-4940-90EB-ABB377255643	-428.420 ms (+762.943 us)
InstallProtocolInterface:	5B1B31A1-9562-11D2-8E3F-009FC969723B 906	-428 102 as (+317 802 us)
Loading driver at 0x0008f6	af000 EntryPoint=0x0008f6af2fc	-120 012 (-11 061)
-12131 UEFI DEBUG		-427.141 ns (+875.883 us)
-12103 UEFI:DEBUG	BC62157E-3E33-4FEC-9920-2D3B36D750DF 906	-426.676 ms (+464.215 us)
PROGRESS CODE: V3040002 I0 -11186 UEFI: DEBUG		-107.532 ms (+1.352 ms)
PROGRESS CODE: V3040003 IO		-178 744 as (al 948 as)
InstallProtocolInterface:	DBC9FD21-FAD8-45B0-9E78-27148867CC93 0	
Unknown code = 0x80860002	004e0224	-5.272 MG (+245.440 GG)
-10339 ME:STATUS	00001, 0000000	-4.640 ms (+200.569 us)
Unknown code = 0x80860002 Arguments = 00000004, 160	0030240d 01310, 00000000	
-10302 ME:STATUS	0030240+	-4.634 as (+6.039 us)
Arguments = 00000004		1 524
Unknown code = 0x80860002	004e00de	-4.5/6 RS (+54./52 US)
Arguments = 00000001, 800 -10202 ME:STATUS	0000a, 00000001	-4.550 as (+26.014 us)
Unknown code = 0x80860002 Avguaents = 00000001 800	004e00df	
-10165 ME:STATUS	004-00-0	-4.343 ms (+207.353 us)
Arguments = 00000001, 080	10002, 00000001	
Unknown code = 0x80860002	004e0196	-4.340 MB (+2.334 US)
-10092 ME:STATUS	10042, 00000001	-4.338 ms (+2.751 us)
Unknown code = 0x80860002	004e00de	
· 🛛		
12791 Disassembly Configure	Display	

Figure 5: Trace Hub Trace



SourcePoint[™] with Intel®Trace Hub

For a demonstration of this exciting new technology contact your local sale SourcePoint representaive.

http://www.asset-intertech.com/company/contact

Trace Hub Benefits

System Trace visibilty

Not only code on IA but ME and SMM events Capture seconds of boot execution in trace

Shorten Time to insight

Better system visibility (macro) with Instruction level precision when needed

Decrease Root Cause analysis time 30%

See bug symptom and manifistation in a single trace or multiple views.

Find the toughest Asyncronous Bugs Quickly

See code execution complexities clearly





17791 Fitch Avenue, Irvine, CA 92614 Voice: 949-517-4150 outside the US Fax: 714-731-6344 E-mail: ai-info@asset-intertech.com. Web: www.asset-intertech.com. SourcePoint is a trademark of ASSET InterTech. Microsoft is a registered trademarks of Microsoft Corp.

> Product Brief Page 4

Copyright © 2015 ASSET InterTech, Inc.