presented by

UEFI Debug with Intel Architectural Event Trace

UEFI 2021 Virtual Plugfest

February 25, 2021

Presented by Alan Sguigna, ASSET InterTech, Inc.

Meet the Presenter

Alan Sguigna

Vice President, Sales & Customer Service,
ASSET InterTech, Inc.

Agenda

- Debug & Trace Described
- The Intel Trace Hub
- AET
- Other Trace
- Demo
- Call to Action

Debug vs Trace

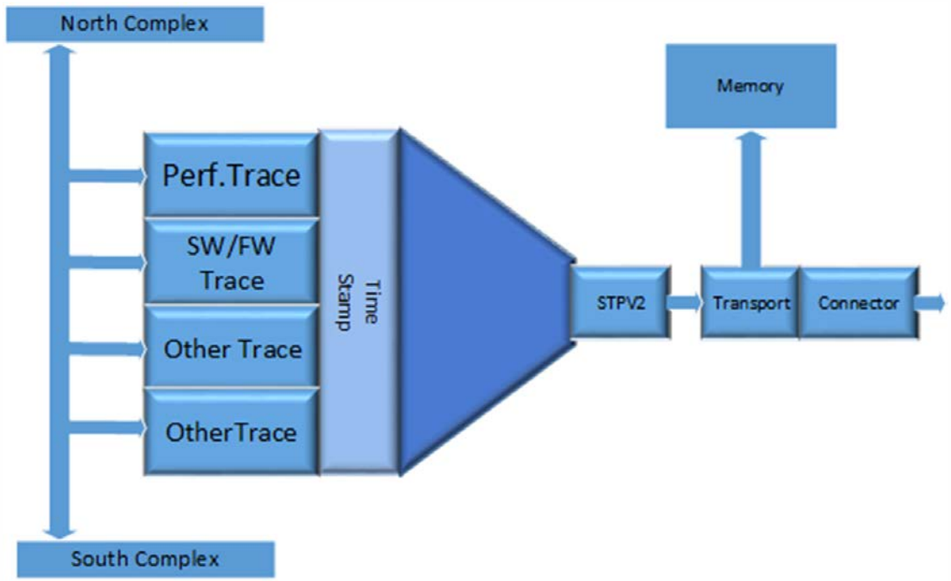
- Debug (Static)
  - Run-control (Break, Halt, Go)
  - Code Walking (stepping and running to a break)
- Trace (Dynamic)
  - Root causing more obscure (hard to find) bugs
  - These are the few bugs that really blow up the schedule
  - Finding these bugs is where Trace shines

“New” Intel Trace Features

- Instruction Trace (Intel Processor Trace)
- ***Event Trace (Intel Trace Hub)***

Intel Trace Hub

- Logic that comprises trace sources, a global hub with timestamp, trace destinations, and a trigger unit
- A sink for writes from cores and any other trace sources
- Acts as a PCI device, and aligned with industry standards
- Trace destinations include:
  - MTB (8kB, out of reset)
  - System Memory (after MRC)
  - ***Direct Connect Interface (out of reset, supports streaming trace)***



AET

Event Type	Event SubTypes	Description
HW/SW Interrupt	HW_INTR	HW interrupt trace
IRET	IRET	IRET trace
Exception	Exception	Exception, fault, trap trace
MSR	RDMSR, WRMSR	MSR trace
Power Management	POWER_ENTRY, POWER_EXIT	Power management
IO	PORT_IN, PORT_OUT, PORT_IN_ADDR	IO trace
SGX	AEX, EENTER, ERESUME, EEXIT	SGX trace
CODE_BP	CODE_BP	Code breakpoint trace
DATA_BP	DATA_BP	Data breakpoint
FIXED_INT	SMI, RSM, NMI	“Fixed” interrupt trace
SW_POWER	MONITOR/MWAIT	MONITOR/MWAIT trace
WBINVD	WBINVD_BEGIN, WBINVD_END	Write-back invalidate trace

AET Tips #1

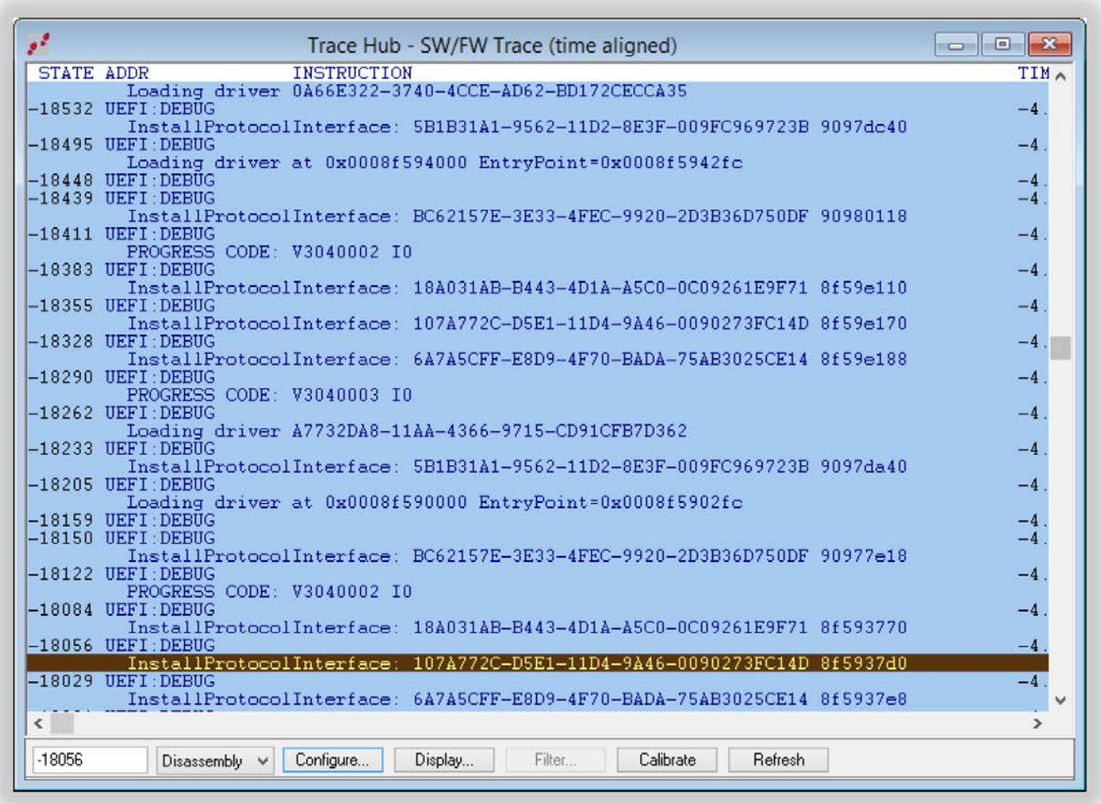
- ***Probe-mode (JTAG) needed to initialize AET – use outside of probe mode (i.e. BIOS, device driver) causes #GP.***
- AET is implemented in CPU microcode and does not modify the architectural behavior* of the processors – no need to instrument code!
  - ** Enabling CODE/DATA_BP changes the behavior of normal breakpoints – causes a trace event rather than a debug exception. Great for critical sections of code, concurrency issues, debugging memory accesses, etc.*
- This is event trace, not instruction trace - source code/ symbols not required (but it's great if you have them!)

AET Tips #2

- A Last Branch Record (LBR) instruction trace stack can be added to all event traces – a fast way to trace back ~ 160 instructions
  - *LBR uses MSRs to track from_address and to_address pairs, so operates out of reset – no need for system memory*
- Intel Processor Trace and AET can run concurrently
  - *IPT places trace data in system memory*
- On Ice Lake processors, both AET LBR tracing and Intel Processor Trace can be enabled at the same time
- AET using XDP access became available initially on Skylake Client and Server
- AET Streaming through DCI (USB) first became available on Ice Lake Client (not available on Purley or Whitley platforms)

Other Trace Hub Trace

- SW/FW Trace
  - Replacement for printf
  - Avoids backpressure from serial port
  - Great for “Heisenbugs”
- CSME (Management Engine)
- All timestamped and correlated*



SW/FW Trace timestamp correlation only available on later silicon.

Demo Configuration

SourcePoint debugger

“Special” USB cable

Intel DesignInTools

Ice Lake Client

Demo

Call to Action

- Take advantage of open source UEFI learning/ development opportunities
  - Debugging Intel Firmware using DCI & USB 3.0
  - Advanced Capabilities of Architectural Event Trace

Questions?

More Questions?

Following today's webinar, join the live, interactive WebEx Q&A for the opportunity to chat with the presenter

Visit this link to attend: https://bit.ly/38JIx9R
Meeting number (access code): 126 016 9253
Meeting password: UEFIForum (83343678 from phones and video systems)

Thanks for attending the UEFI 2021 Virtual Plugfest

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by