

*presented by*



# Beyond Printf – Real-Time UEFI Debugging

UEFI 2021 Virtual Plugfest

October 27, 2021

Alan Sguigna, ASSET InterTech, Inc.

# Meet the Presenter



Alan Sguigna

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

# Agenda

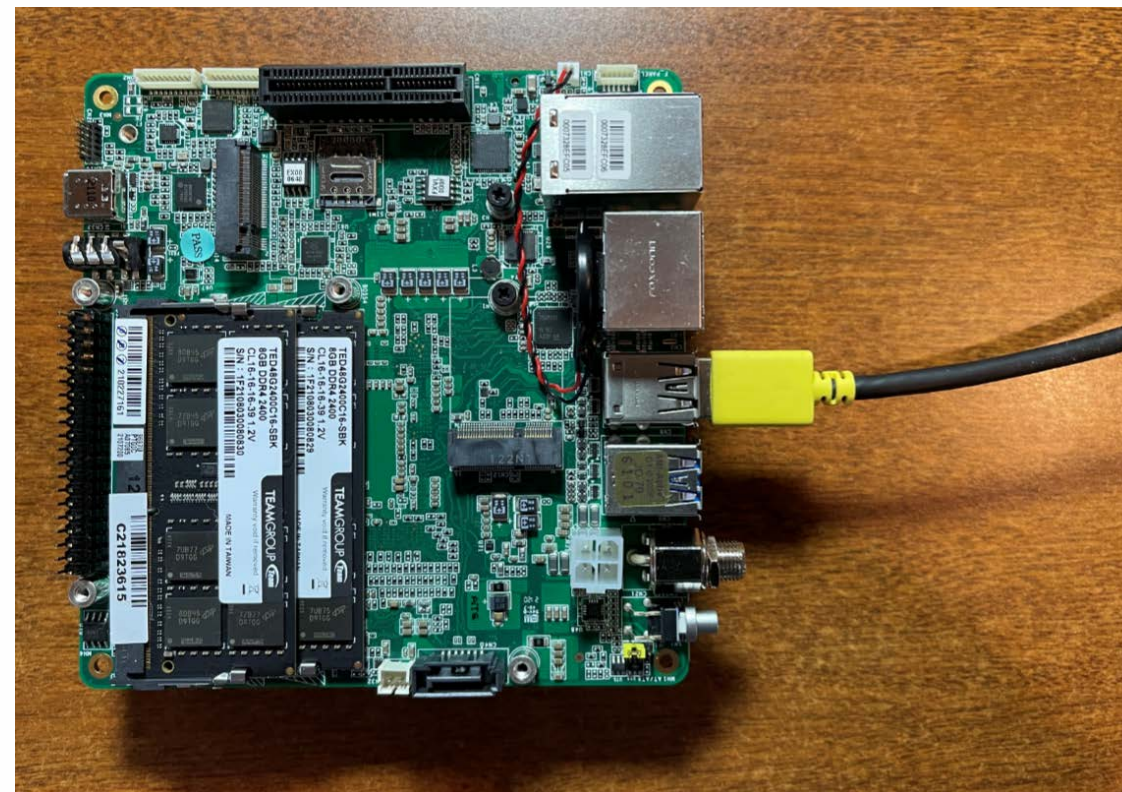


- Intel Trace Hub
  - Trace Sources
  - Trace Sinks
- Instruction Trace
- Setting up the Trace Hub
- Setting up At-Speed Printf
- Demo



# JTAG Access

- XDP – Open Chassis:  
“MinnowBoard”
- Direct Connect Interface (DCI) –  
Closed Chassis:  
“AAEON UP Xtreme  
i11”





# “New” Intel Trace Features

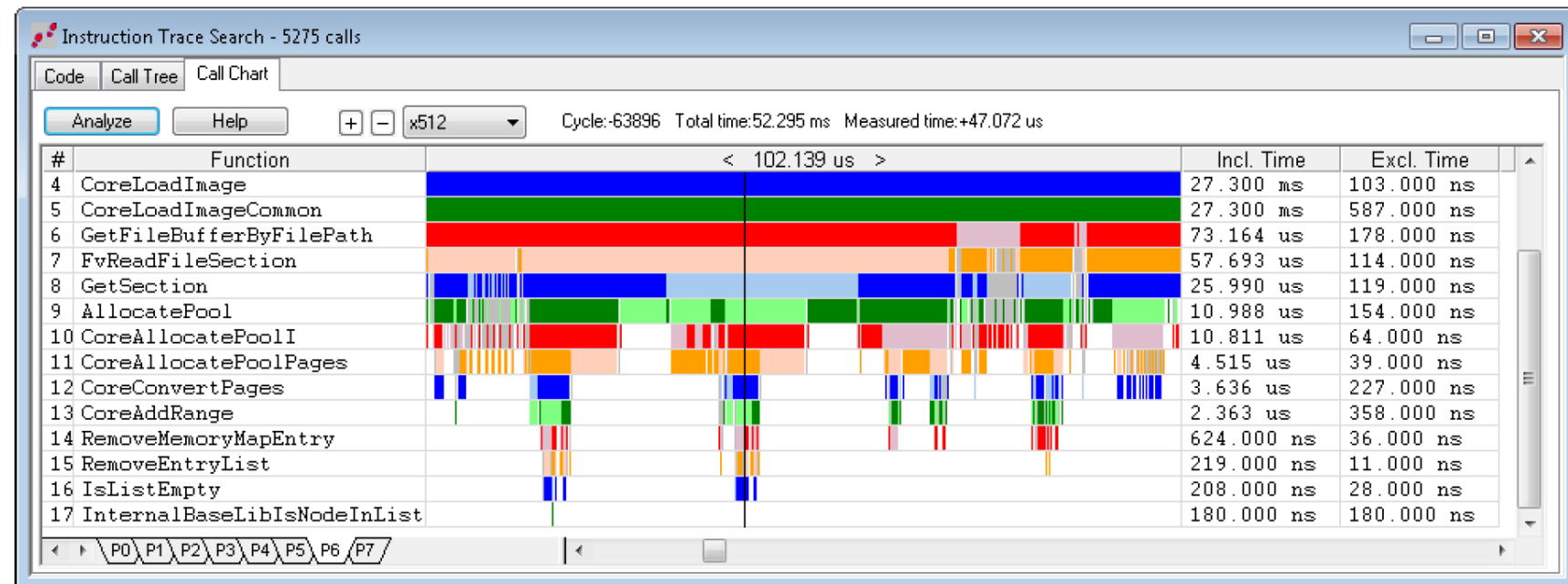
- Instruction Trace (Intel Processor Trace)
- Event Trace (Intel Trace Hub)

Between the two, provides for full system debug: testing the interaction of hardware and software as they produce complex system behaviors.



# Intel Processor Trace (IPT)

- Globally timestamped
- Highly compressed; no significant impact on execution speed
- Trace buffer in system memory

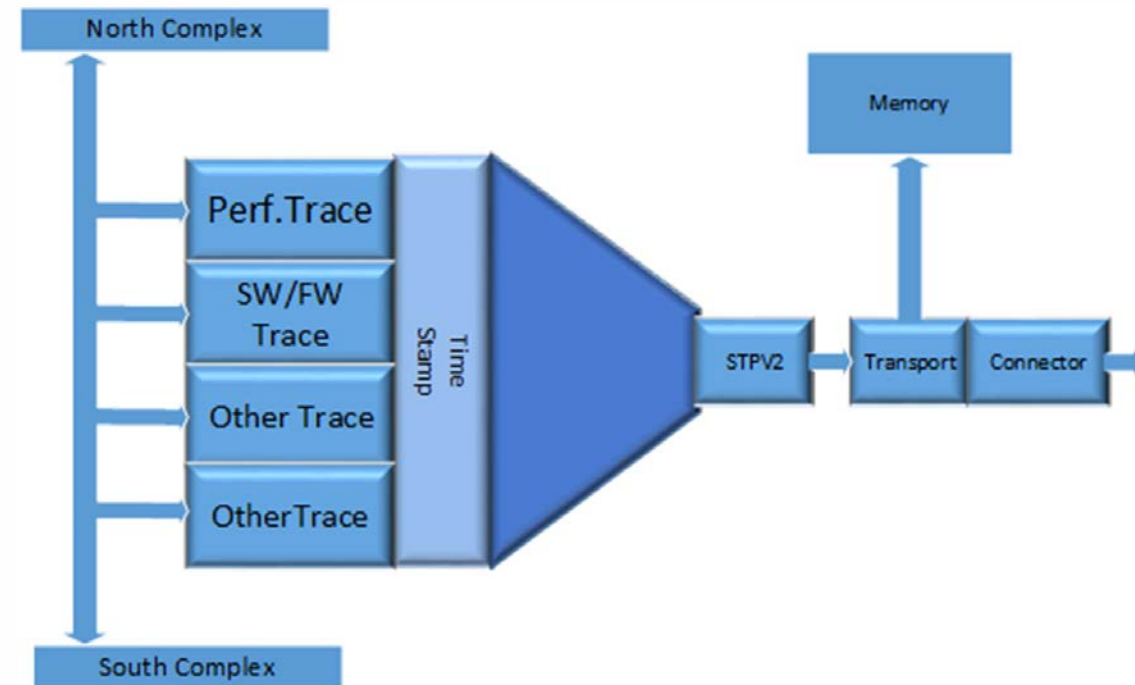




# Intel Trace Hub (ITH)



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



# ITH Sources: 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



# ITH Sources: SW/FW



## SW/FW Trace

- “At-Speed Printf” (ASPF)
  - Printf pointer goes to Trace Hub, and the PC host processes string
  - Speeds up debug build boot
  - Avoids backpressure from serial port and printf code execution
  - Great for “Heisenbugs”
- Timestamped, and can be correlated with IPT, AET and other run-control and trace data

The screenshot shows a window titled "Trace Hub - SW/FW Trace (time aligned)". The window contains a table with four columns: STATE, ADDR, INSTRUCTION, and TIM. The table lists various instructions and their addresses, including "Loading driver", "InstallProtocolInterface", and "PROGRESS CODE". The instruction at address -18056 is highlighted in yellow.

STATE	ADDR	INSTRUCTION	TIM
		Loading driver 0A66E322-3740-4CCE-AD62-BD172CECCA35	
-18532	UEFI:DEBUG		-4.
		InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-009FC969723B 9097dc40	
-18495	UEFI:DEBUG		-4.
		Loading driver at 0x0008f594000 EntryPoint=0x0008f5942fc	
-18448	UEFI:DEBUG		-4.
-18439	UEFI:DEBUG		-4.
		InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 90980118	
-18411	UEFI:DEBUG		-4.
		PROGRESS CODE: V3040002 I0	
-18383	UEFI:DEBUG		-4.
		InstallProtocolInterface: 18A031AB-B443-4D1A-A5C0-0C09261E9F71 8f59e110	
-18355	UEFI:DEBUG		-4.
		InstallProtocolInterface: 107A772C-D5E1-11D4-9A46-0090273FC14D 8f59e170	
-18328	UEFI:DEBUG		-4.
		InstallProtocolInterface: 6A7A5CFF-E8D9-4F70-BADA-75AB3025CE14 8f59e188	
-18290	UEFI:DEBUG		-4.
		PROGRESS CODE: V3040003 I0	
-18262	UEFI:DEBUG		-4.
		Loading driver A7732DA8-11AA-4366-9715-CD91CFB7D362	
-18233	UEFI:DEBUG		-4.
		InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-009FC969723B 9097da40	
-18205	UEFI:DEBUG		-4.
		Loading driver at 0x0008f590000 EntryPoint=0x0008f5902fc	
-18159	UEFI:DEBUG		-4.
-18150	UEFI:DEBUG		-4.
		InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 90977e18	
-18122	UEFI:DEBUG		-4.
		PROGRESS CODE: V3040002 I0	
-18084	UEFI:DEBUG		-4.
		InstallProtocolInterface: 18A031AB-B443-4D1A-A5C0-0C09261E9F71 8f593770	
-18056	UEFI:DEBUG		-4.
		InstallProtocolInterface: 107A772C-D5E1-11D4-9A46-0090273FC14D 8f5937d0	
-18029	UEFI:DEBUG		-4.
		InstallProtocolInterface: 6A7A5CFF-E8D9-4F70-BADA-75AB3025CE14 8f5937e8	



# ITH Sinks

- MTB (2kB – 8kB; available out of reset)
- System memory (post-MRC)
- **Direct Connect Interface (DCI)** – streaming out of reset
  - DbC3: USB Type A/C, S0 power state only
  - DbC2: USB Type A/C, S0ix debug, survives Sx power state transitions and warm/cold resets

# Setting up ITH and At-Speed Printf



- ITH is configured in BIOS
- Access it early before it gets “hidden”
- Build DEBUG printf “hooks” replaced with calls to system trace library
  - Typically DEBUG, RCPRINTF, and ASSERT\_EFI\_ERROR
- Use printf as you normally would

# Setup: Implementation Steps



Add ASSET System Trace Library to MdePkg.dsc :

```
MdePkg/Library/BaseDebugLibSystemTrace/BaseDebugLibSystemTrace.inf
```

and use ASSET-provided BaseDebugLibSystemTrace.inf and header and .c files. The Trace Library can now be included.

1. Modify within the PlatformPkg.dsc file:

```
DebugLib|MdePkg/Library/BaseDebugLibSerialPort/BaseDebugLibSerialPort.inf
```

to

```
DebugLib|MdePkg/Library/BaseDebugLibSystemTrace/BaseDebugLibSystemTrace.inf
```

OR

2. Use compiler option:

In build script, VAR\_BUILD\_FLAGS= .....-DASSET\_SYSTEM\_TRACE=TRUE

And for each module add conditional code:

```
!if $(ASSET_SYSTEM_TRACE) == TRUE
```

```
    DebugLib|MdePkg/Library/BaseDebugLibSystemTrace/BaseDebugLibSystemTrace.inf
```

```
!else
```

```
    DebugLib|MdePkg/Library/BaseDebugLibSerialPort/BaseDebugLibSerialPort.inf
```

```
!endif
```

**\MdePkg\Library\BaseDebugLibSerialPort  
\DebugLib.c**

```
VA_START(Marker, Format);  
AsciiVSPrint(Buffer, sizeof(Buffer), Format, Marker);  
VA_END(Marker);  
// Send print string to a Serial Port  
SerialPortWrite((UINT8*)Buffer, AsciiStrLen(Buffer));
```



**\MdePkg\Library\BaseDebugSystemTrace\  
DebugLib.c**

```
// Send print string to a Serial Trace Device  
VA_START(Marker, Format);  
SystemTraceWrite((UINT8*)Format, AsciiStrLen(Format),  
Marker);  
VA_END(Marker);
```

# Instruction Trace + Event Trace (DCI)



## Pre-MRC completion

- Instruction Trace: LBR
- Event Trace: Trace Hub out of reset (AET w/LBR, ME, ASPF, ...)

## Post-MRC completion

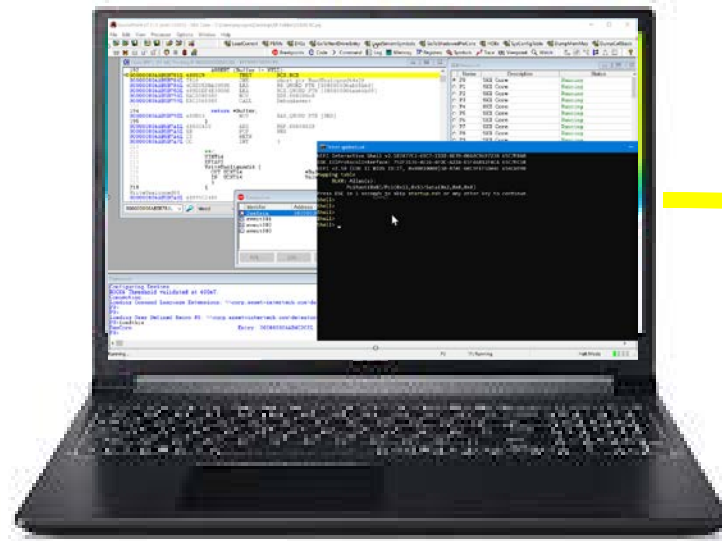
- Full Intel Processor Trace to System Memory
- Event Trace: Trace Hub out of reset (AET w/LBR, ME, ASPF, ...)



# Demo Configuration



## SourcePoint debugger



**“Special” USB cable**

**Intel DesignInTools,  
DataPro**



**Ice Lake Client**



# Demo



# Call to Action

- Take advantage of UEFI learning/ development opportunities
  - [Debugging Intel Firmware using DCI & USB 3.0](#)
  - [UEFI Debug with Intel Architectural Event Trace](#)
  - [ASSET blog](#)
- Access ASPF support files at [www.asset-intertech.com/sourcepoint-academy/at-speed-printf](#)



# Questions?

Thanks for attending the UEFI 2021 Virtual Plugfest

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

*presented by*

