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

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

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));

// Send print string to a Serial Trace Device
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