JTAG-based UEFI Debug and Trace

UEFI 2020 Virtual Plugfest
July 14, 2020
Presented by Alan Sguigna, ASSET InterTech, Inc.
Meet the Presenter

Alan Sguigna
Vice President, Sales & Customer Service
Member Company:
ASSET InterTech, Inc.

www.uefi.org
Agenda

• What is JTAG? Debug use case
• Access mechanisms (platform-dependent)
• Tools of the Trade: Run-control, Trace, scripting
• Examples/ Demo:
  – Run-control: halt, go, single-step, breakpoint
  – Trace: Last Branch Record (LBR), Branch Trace Store (BTS), Instruction Trace, Architectural Event Trace, ME trace
  – Intel CScripts
• Call to Action
What is JTAG?

- Celebrated its 30th Anniversary on February 15, 2020
- “Joint Test Access Group”
- IEEE 1149.1 and subsequent standards – ingrained within much of today’s commercial silicon
- Specifies a dedicated debug port with a serial communications interface
- Test Access Port implements a stateful protocol with test registers that connect with a chip’s system logic
- An “engine” within chips that drives embedded instrumentation for a plethora of applications
JTAG Applications

- **Test**: Boundary-Scan Test, JTAG-based functional test
- **Debug**: Run-control (Intel ITP, AMD HDT, Arm CoreSight)
- **Validation**: Hardware performance / conformance
- **Programming**: SVF, STAPL, JAM, and at-speed flash IP

www.uefi.org
Why is JTAG Useful for Debugging?

• “Bare-metal” debugging at the interface between the hardware and the software
• Essential for debug on wedged platforms
• Use same tools as used in silicon validation
Access Mechanisms (Intel)

• XDP (eXtended Debug Port)

• DbC/ DCI (Debug Class)

• BMC
Tools of the Trade

• Run-control

• Trace

• Scripting
Examples
Basic Run-Control – MinnowBoard
Intel Processor Trace – Apollo Lake
Intel AET – Skylake-SP
CScripts – Skylake-SP

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bkc</td>
<td>Best Known Configuration checker</td>
</tr>
<tr>
<td>coreinfo</td>
<td>Processor Core Information</td>
</tr>
<tr>
<td>edk2</td>
<td>UEFI Development Kit 2</td>
</tr>
<tr>
<td>ei</td>
<td>Error Injection</td>
</tr>
<tr>
<td>error</td>
<td>System Errors</td>
</tr>
<tr>
<td>mc</td>
<td>Memory Controller</td>
</tr>
<tr>
<td>pch</td>
<td>Platform Control Hub</td>
</tr>
<tr>
<td>pci</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>pcie</td>
<td>Peripheral Component Interconnect (PCIe) base class</td>
</tr>
<tr>
<td>pm</td>
<td>Power Management</td>
</tr>
<tr>
<td>ras</td>
<td>Reliability, Availability, Serviceability</td>
</tr>
<tr>
<td>skx</td>
<td>Model Specific Registers (MSR)</td>
</tr>
<tr>
<td>uncoreinfo</td>
<td>SKX uncore implementation</td>
</tr>
<tr>
<td>upi</td>
<td>UPI module implementation</td>
</tr>
</tbody>
</table>

www.uefi.org
Call to Action

• Take advantage of open source learning/development opportunities
  – The MinnowBoard Chronicles
  – Debugging Intel Firmware using DCI & USB 3.0
  – Intel Firmware site

www.uefi.org
Questions?
Thanks for attending the UEFI 2020 Virtual Plugfest

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