Agenda

- WinDbg
- SourcePoint
- Why combine the two?
  - Enhancing WinDbg with JTAG-based run-control and trace features
  - “Debugging the Undebuggable”
- Demo configuration
  - What you’ll see in the demo
- Demo
- Wrap-Up
WinDbg

- WinDbg is a kernel-mode and user-mode debugger that's included in Debugging Tools for Windows
- De facto standard for Windows system-level programming/debugging
- Very powerful: OS-aware (processes, threads, jobs, kernel symbols, etc.)
- Best-in-class UEFI debugger
- Support for x86: Intel (all CPUs) and AMD (EPYC)
- Source-level symbolic debugger, full run-control (stop, go, single-step, breakpoints, etc.)
- Supports Advanced (Conditional) Breakpoints on AMD
- Supports innovative Trace features on Intel
Microsoft released update to EXDI (Extended Debug Interface)

EXDI is an adaptation layer between a software debugger and a debugging target.

Extends WinDbg by adding support for hardware-based debuggers (i.e. JTAG-based)

WinDbg is the controller; SourcePoint is the worker

“Debugging the Undebuggable”
https://www.andrea-allievi.com/blog/debugging-the-undebuggable-part-1/

But on steroids!
Why combine SourcePoint with WinDbg?

Take advantage of powerful features of both applications:

**WinDbg**
- OS-aware
- Extensible
- Go-to tool for kernel debugging

**SourcePoint**
- Advanced breakpoint support (SMM entry/exit/data access, machine check, etc.)
- Intel Processor Trace
- Trace Hub
- Architectural Event Trace (AET)
- Execution Trace
- Real-time (nominal performance impact)
- Stored in system memory (post-MRC)
- Call Tree, Call Chart, all with Search capability
- Up to 2GB instruction trace
Intel Trace Hub

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

Event Trace, that complements Instruction Trace. Requires JTAG.

<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>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 trace</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>
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!)
▪ A Last Branch Record (LBR) instruction trace stack can be added to all event traces – a fast way to trace back ~ 300 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
  ▪ *Intel PT places trace data in system memory*
▪ On Ice Lake processors and later, both AET LBR tracing and Intel Processor Trace can be enabled at the same time
1. SourcePoint makes JTAG connection at reset vector with DCI.
2. COTS Tiger Lake target booted from reset vector to Windows desktop.
3. SourcePoint launches WinDbg, makes EXDI connection.
4. Symbols visible in both WinDbg & SourcePoint.
5. Demonstrate run-control, symbolic debug (stop, go, set breakpoint, single-step, etc.), Intel Processor Trace, AET, etc.

© 2023, ASSET InterTech, Inc.
Resources

- SourcePoint Academy: https://www.asset-intertech.com/resources/academy/sourcepoint-academy/
  - SourcePoint WinDbg Getting Started Guide
  - Getting Started Guide for the AAEON UP Xtreme i11
  - Videos, Online Help, Release Notes, etc.
Questions and Contact Information

alan.sguigna@asset-intertech.com
X DM @AlanSguigna
Real Insight from Code to Silicon