Intel DCI Secrets

HITBSecConf 2017 CommSec, Amsterdam
The 8th Annual HITB Security Conference in The Netherlands
April 10–14, 2017
• Maxim Goryachy  
  email: mgoryachy<at>ptsecurity<.>com  
  twitter: @h0t_max

• Mark Ermolov  
  email: mermolov<at>ptsecurity<.>com  
  twitter: @_markel___

• Dmitry Sklyarov  
  email: dsklyarov<at>ptsecurity<.>com
Our Previous Related Talks On the Subject

• *Tapping into the Core*, Chaos Computer Club (33C3)

• *Intel ME: The Way of the Static Analysis*, TROOPERS17
• X86 hardware debugging as a must
• Intel DCI overview
• Design for manufacturability, testability, and debuggability (DFx)
• DFx Abstraction Layer
• Intel DCI protocol
• Run-time activation
• Debugging early platform boot stages (SEC, PEI) is impossible with software debuggers.
• CPU is not really halted on software debuggers break.
• Debugging CPU-specific features:
  ✓ VMX-root operation
  ✓ SMM
  ✓ SGX
• Performance analyze without side effects.
• Anti-anti-debugging.
• Intel Direct Connect Interface (DCI) provides access to CPU/PCH JTAG via USB3.0
• Two transport types: Intel SVT CCA and USB 3.0 DbC
• **Software is available without NDA**
Intel DCI

DFx Abstraction Layer

Host
- Host-Software
- DAL
- Transport

Target
- ExI
- DFx
- Target Internal Devices

Hardware required:
- Intel® SVT Closed Chassis Adapter

BSSB Hosting DCI:
For lower power (Sx-State) & S0-State DFx access

USB3 Hosting DCI:
For S0-State DFx access and high performance operations
- DFx is private implementation of JTAG (1149.1 and 1149.7) by Intel.
- Inside PCH and CPU a lot of integrated devices coupled to DFx-chain.
- Access to DFx is performed by use of an Embedded DFx Interface (ExI).
- ExI is a bridge between DFx and external interfaces (such as USB).
- PCH has the special internal device DFX_AGGREGATOR controlling access to DFx.
• DAL stands for DFx Abstraction Layer, a software stack for DFx.
• DAL is heart of all recent Intel HW debugging/check tools (System Debugger, System Trace, Platform Debugging Toolkit).
• Supports wide range of platforms/CPUs.
• Supports multiple Intel hw probe types.
• **DAL is available without NDA.**
Overview of Intel DAL

- Written almost in C#.
- Has a GUI configuration tool.
- Python command line interface.
Interface of Intel DAL

- SCAN CHAIN 1
  - TAP
  - TAP
  - TAP
- SCAN CHAIN 2
  - TAP
  - TAP
  - TAP
- I2C
- OBS
- DMA

- CORE 1
  - THREAD 1
  - THREAD 2
- CORE 2
  - THREAD 1
  - THREAD 2
- BOX GC
- BOX EDC

- CHIPSET
- CHIPSET
- CHIPSET
- CHIPSET

Physical nodes
Logical nodes
What DAL Can Do

- Perform standard debugging stuff (run control, access to architectural state, memory).
- Handy JTAG IR/DR scan operation.
- Break/debug system events (reset vector, shutdown, SMM).
- Debug VMX (exit/launch, read/modify VMCS).
- Debug SGX (read/write enclave memory, EPCM, TCS, SSA).
- **Access PCIe bus and MCE banks without CPU halt (using uncore VCU).**
- Perform DMA in CPU independent way.
- Access to PCH internal devices (**IOSF SideBand**, Trace Hub, **PMC**).
Intel DCI Protocol (DCI Payload)

PACKETIZING JTAG ACROSS INDUSTRY STANDARD INTERFACES
Patent No.: US 9,015,542 B2
PACKETIZING JTAG ACROSS INDUSTRY STANDARD INTERFACES
Patent No.: US 9,015,542 B2
39.1 Overview

The PCH incorporates a wide variety of devices and functions. The registers within these devices are mainly accessed through the primary interface, such as PCI configuration space and DQPMI space. Some devices also have registers that are distributed within the PCH Private Configuration Space at individual endpoints (Target Port IDs) which are only accessible through the PCH Sideband Interface.

These PCH Private Configuration Space Registers can be addressed via SBREG_BAR or through PCI Index data pair programming.

Table 39-1. Private Configuration Space Register Target Port IDs (Sheet 1 of 2)

<table>
<thead>
<tr>
<th>PCH Device/Function Type</th>
<th>Target Port ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI0 Phy Configuration</td>
<td>0x80</td>
</tr>
<tr>
<td>General Purpose I/O (GPIOD) Community 1</td>
<td>0xA0</td>
</tr>
<tr>
<td>General Purpose I/O (GPIOD) Community 2</td>
<td>0xA1</td>
</tr>
<tr>
<td>General Purpose I/O (GPIOD) Community 3</td>
<td>0xA2</td>
</tr>
<tr>
<td>General Purpose I/O (GPIOD) Community 4</td>
<td>0xA3</td>
</tr>
<tr>
<td>DCI</td>
<td>0x88</td>
</tr>
</tbody>
</table>

34.3.1 DCI Control Register (ECTRL) Offset 4h

Access Method

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Default &amp; Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:5</td>
<td>Reserved.</td>
</tr>
<tr>
<td>4</td>
<td>1h</td>
</tr>
<tr>
<td>3:0</td>
<td>3h</td>
</tr>
</tbody>
</table>

Field Name (ID): Description

- 1h: Host DCI Enable (HDCIEN): 0 = Disable DCI
- 3:0: Reserved.
• CVE-2017-5684,5,6 – run-time dci activation

Summary:

Intel® NUC and Intel® Compute Stick systems based on 6th Gen Intel® Core™ processors do not have DCI debug capability properly locked for BIOS only access. This would allow an attacker with physical possession of the system to potentially enable DCI from outside the BIOS.
List of Checked Devices

- Gigabyte BRIX GB-BSi3HA-6100 (CPU + PCH)
- Gigabyte BRIX GB-BSi5HA-6300 (CPU + PCH)
- Lenovo t460s (CPU + PCH)
- ASUS Q170M-C (PCH)
- ASUS Z170-A (PCH)
Problem: Link between PCH JTAG and CPU is broken

In progress ...
Plans Ahead

- Turn-on JTAG for GT cores.
- Intel Silicon View and HOTHAM.
- Try to access IOSF internal buses.
- Win the battle to activate JTAG for Intel ME and ISH.
Hidden Hardware Random Generator from PCH?!
Thank you!

Q&A

www.ptsecurity.com
blog.ptsecurity.com
@PTsecurity_UK
github.com/ptresearch