Stupid PCIe Tricks

featuring

NSA Playset: PCIe

DEFCON 22
Joe FitzPatrick
Miles Crabill
This document is a preliminary revision. For the latest version, as well as information about tools released, visit http://securinghardware.com/nsa-playset
whoami

- Electrical Engineering education with focus on CS and Infosec
- 8 years doing security research, speed debug, and tool development for CPUs
- Hardware Pen Testing of CPUs
- Security training for functional validators worldwide

Joe FitzPatrick
@securelyfitz
joefitz@securinghardware.com
whoami

- Computer Science student at Lewis & Clark College
- Almost 3 years of experience in security research
- Little to no prior hardware hacking experience

Miles Crabill
@milescrabill
miles@milescrabill.com
What is PCIe?
PCle is PCI!
PCle is NOT PCI!

Foto tomada por Jorge González http://es.wikipedia.org

Photo by snikerdo http://en.wikipedia.org
Links and Lanes

Diagram: PCIe 2.1 specification
Hierarchy

Diagram: PCIe 2.1 specification
Switching and Routing

Diagram: PCIe 2.1 specification
Layers

Diagram: PCIe 2.1 specification
Accessing PCIe Space

Diagram: http://blog.codinghorror.com/dude-where's-my-4-gigabytes-of-ram/
1.1 PCI/PCI Express Configuration Space Memory Map

- Rest of Extended Configuration Space for PCI Express Parameters Capabilities
- PCI Express Extended Configuration Space
- PCI Express Extended Cap Structure (start at 100h)
- PCI Express Capability Structure (Basic Confic Req)
- PCI 2.x Compatible Configuration Header

- 4K/func/dev, 256MB per bus
- Flat memory mapped access
- Firmware indicates memory base
- First 256 bytes PCI compatible
- Do not assume CF8/CFC available for extended space access

# Configuration Space

Diagram: PCIe 2.1 specification
Configuration Space

Diagram: PCIe 2.1 specification
Configuration Space

Diagram: PCIe 2.1 specification
Configuration Space

Diagram: PCIe 2.1 specification
Configuration Space

Diagram: PCIe 2.1 specification
Enumeration

Diagram: PCIe 2.1 specification
Routing PCIe
The Step-By-Step, Complicated, Mandatory, Inflexible Rules of Routing PCIe:
The Step-By-Step, Complicated, Mandatory, Inflexible Rules of Routing PCIe:

1. route pairs adjacent and equal length
The Step-By-Step, Complicated, Mandatory, Inflexible Rules of Routing PCIe:

1. route pairs adjacent and equal length

… that’s mostly it
Routing PCIe

- System Board Traces: 12 Inches
- Add-in Card Traces: 3.5 inches
- Chip-to-Chip Routes: 15 inches

Follow these rules and your board might work. Break them and it might not.
Routing PCIe

Minimum PCIe:

● 2.5GHz TX
● 2.5GHz RX
● 100MHz Clock (optional)
Routing PCIe

Cross-section of a USB 3.0 cable. Image courtesy of USB Implementers Forum
Getting PCIe on Things Without It
Intel Galileo

Intel Galileo Front

Intel Galileo Back
```
root@clanton:~#
root@clanton:~# lspci -k
00:00.0 Class 0600: 8086:0958 intel_qrk_sb
00:14.0 Class 0805: 8086:08a7 sdhci-pci
00:14.1 Class 0700: 8086:0936 serial
00:14.2 Class 0c03: 8086:0939
00:14.3 Class 0c03: 8086:0939 ehci-pci
00:14.4 Class 0c03: 8086:093a ohci_hcd
00:14.5 Class 0700: 8086:0936 serial
00:14.6 Class 0200: 8086:0937 stmmaceth
00:14.7 Class 0200: 8086:0937
00:15.0 Class 0c80: 8086:0935
00:15.1 Class 0c80: 8086:0935
00:15.2 Class 0c80: 8086:0934
00:17.0 Class 0604: 8086:11c3 pcieport
00:17.1 Class 0604: 8086:11c4 pcieport
01:00.0 Class 0300: 1ade:11c2 nouveau
01:00.1 Class 0403: 10de:0e0b
root@clanton:~#```
Specifications:
Power Requirements: 100-240V, 50/60Hz
Drive Connections: SD x1, USB 2.0 x1
Network Connection: Gigabit Ethernet
Drive Formats: NTFS, FAT, HFS+, EXT2, EXT3
Web Browsers: Microsoft® Internet Explorer, Mozilla® Firefox, Apple® Safari, Google Chrome™
Operating Systems: Microsoft® Windows XP/7/8, Apple® Mac OS X 10.6.8 & above
Apps Available For: iPhone®, iPad®, Android™

What's Included:
Pogoplug
Power cable
Ethernet cable
User manual
Introducing SLOTSCREAMER
Xilinx Kintex-7 FPGA KC705 Evaluation Kit

Overview Hardware Tools & IP Docs & Designs

The Kintex-7 FPGA KC705 Evaluation Kit includes all the basic core designs including a targeted design featuring pre-verified reference designs and daughter cards.

What's Included
- KC705 Evaluation Board featuring a 2080-pin FPGA
- Targeted Reference Design featuring:
  - Including evaluation version of Vivado Design Suite
  - AMS 101 Evaluation Card
  - Full seat of Vivado Design Suite

Buy from Xilinx
Lead Time: 2 Weeks

Spartan-6 FPGA SP605 Evaluation Kit

Accelerate Your Designs – Right Out of the Box.

Product Information

The Spartan-6 FPGA SP605 Evaluation Kit delivers all the hardware, design tools, IP, and reference designs enabling you to hit the ground running. This kit provides a flexible environment for system design, reference design and examples on how to leverage features such as transceivers, PCI Express®, DVI, and/or DDR3. This kit includes a FMC (FPGA Mezzanine Card) connector for future scalability in applications and markets.

What's Included

Mouser Part #: 989-DK-START-4CGX15N
Manufacturer Part #: DK-START-4CGX15N
Manufacturer: Altera Corporation
Description: Programmable Logic IC Development Tools FPGA Starter Kit For EP4CGX15F14
Lifecycle: New At Mouser

Learn more about Altera Corporation DK-START-4CGX15N
- Page 292, Mouser Online Catalog
- Page 292, PDF Catalog Page
- Datasheet

Enter Quantity: [ ] Buy
Minimum: 1
Multiples: 1

Pricing (USD)
1: $495.00
8.6.3 **PCIOUT Endpoint**

PCIOUT is a Bulk endpoint that allows the USB Host to initiate Read and Write Requests to PCI Express Space, using the PCI Master Control Cursor registers. Packets sent to this endpoint consist of the format listed in Table 8-12.

There can be from 0 to 64 Payload DWords, requiring USB packet sizes from 8 to 264 bytes.

**Table 8-12. PCIOUT Packet Format**

<table>
<thead>
<tr>
<th>Byte Index</th>
<th>Register</th>
<th>Destination Register Bytes</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>[7:0]</td>
</tr>
<tr>
<td>1</td>
<td>PCIMSTCTL (USB Controller, offset 100h)</td>
<td>[15:8]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>[23:16]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>[31:24]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>[7:0]</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PCIMSTADDR (USB Controller, offset 104h)</td>
<td>[15:8]</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>[23:16]</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>[31:24]</td>
<td></td>
</tr>
<tr>
<td>8 through 11</td>
<td></td>
<td>Payload DW0 (LSB first; to PCIOUT FIFO)</td>
<td></td>
</tr>
<tr>
<td>12 through 15</td>
<td></td>
<td>Payload DW1 (LSB first; to PCIOUT FIFO)</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>And so forth</td>
<td></td>
</tr>
</tbody>
</table>
Register 15-57. 200h, 210h, 220h, 230h, 240h, 250h DEP_CFG Dedicated Endpoint Configuration for CSROUT, CSRIN, PCIOUT, PCIIN, STATIN, and RCIN (USB Controller)

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Description</th>
<th>Access</th>
<th>Serial EEPROM</th>
<th>Default</th>
</tr>
</thead>
</table>
| 3:0    | **Endpoint Number**  
Selects the endpoint number.                                               | RW     | Yes           | RCIN = Ch,  
CSROUT = Dh,  
CSRIN = Dh,  
PCIOUT = Eh,  
**PCIIN = Eh**,  
**STATIN = Fh** |
| 7:4    | **Reserved**                                                               | RsvdZ  | Yes           | 0h                                                                     |
| 8      | **Endpoint Type**  
0 = STATIN or RCIN endpoint becomes a BULK endpoint.  
1 = STATIN or RCIN endpoint becomes an INTERRUPT endpoint. Valid only for the STATIN or RCIN endpoint.  
All other endpoints are BULK. | RW     | Yes           | STATIN = 1,  
RCIN = 1,  
Others = 0 |
| 9      | **Reserved**                                                               | RsvdZ  | Yes           | 0                                                                      |
| 10     | **Endpoint Enable**  
1 = Enables this endpoint                                                  | RW     | Yes           | RCIN = 0 in Adapter mode,  
Others = 1 |
| 15:11  | **Service Interval**  
Determines the interrupt service interval for STATIN/RCIN endpoints in USB r3.0 mode. | RW     | Yes           | STATIN = 1,  
RCIN = 1,  
Others = 0 |
| 31:16  | **Reserved**                                                               | RsvdZ  | Yes           | 0000h                                                                  |
/* Explicitly disable the 6 dedicated endpoints */

tmp = 0x00;
for (i = 0; i < 4; i+=2, tmp++) {
    writel (tmp, &dev->dep[i].dep_cfg);
    writel (tmp, &dev->dep[i+1].dep_cfg);
}
writel (0x0f, &dev->dep[4].dep_cfg);
writel (0x0c, &dev->dep[5].dep_cfg);
PyUSB

About

PyUSB aims to provide easy USB access to the Python language.

The project is divided in two major versions: the stable 0.x and the under development 1.0. PyUSB 1.0 enhances the library in several ways:

- Support for libusb 0.1, libusb 1.0 and OpenUSB.
- Easy API to communicate with devices.
- Support for custom library backends.
- Isochronous transfer type support.
- 100% written in Python by ctypes.
- It runs on any Python version >= 2.4 (this includes Python 3).
Table 7-2. PCI Master Control Registers\(^a\)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Register</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>100h</td>
<td>PCIMSTCTRL</td>
<td>Specifies access type and direction (Read/Write)</td>
</tr>
<tr>
<td>104h</td>
<td>PCIMSTADDR</td>
<td>Contains the PCI Express address to be accessed</td>
</tr>
<tr>
<td>108h</td>
<td>PCIMSTDATA</td>
<td>Contains data to be written or data returned from a Read</td>
</tr>
</tbody>
</table>

\(a\). The PCI Master Control register set also includes one Status and one Message register.

Through the PCI Master Control registers, the 8051 or USB Host CPU can generate the following types of accesses into PCI Express space:

- Configuration Read
- Configuration Write
- Memory Read
- Memory Write
- I/O Read
- I/O Write
- PCI Express Messages
### Register 15-41. 100h PCIMSTCTL PCI Master Control (USB Controller)

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Description</th>
<th>Access</th>
<th>Serial EEPROM</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:0</td>
<td>PCI Express First Byte Enables</td>
<td>RW</td>
<td>Yes</td>
<td>0h</td>
</tr>
<tr>
<td></td>
<td>Determines the first Byte Enables of a PCI Express transaction. For 1-DWord transactions, it can be any value. For multiple DWord transactions, only contiguous Byte Enables are allowed, or the endpoint is halted. This field is used directly in the FBE field of the PCI Express Header.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:4</td>
<td>PCI Express Master Command Select</td>
<td>RW</td>
<td>Yes</td>
<td>00b</td>
</tr>
</tbody>
</table>
|        | When the USB 3380 performs PCI Express transactions initiated by the PCIOUT endpoint or 8051, determines the PCI Express Request type issued.  
**Note:** The Configuration Type (Type 0 or Type 1) is determined by the PCI Master Address format. |
| Value  | Read Command | Write Command |
| 00b    | Memory Read  | Memory Write  |
| 01b    | I/O Read     | I/O Write     |
| 10b    | Configuration Read | Configuration Write |
| 11b    | Reserved     | PCI Express Message |
| 6      | PCI Express Master Start | RW1S     | Yes           | 0       |
|        | Writing 1 causes a PCI Write or Read transaction to start. This bit is Cleared when the PCI transaction is complete. For Write operations, determines when to start another Write. For Read operations, determines when the PCIMSTDATA register (USB Controller, offset 108h) contains valid data. This bit is automatically Cleared when a UR or CA occurs. |
| 7      | PCI Express Master Read/Write | RW      | Yes           | 0       |
|        | 0 = PCI Write transaction is selected. 1 = PCI Read transaction is selected. For 8051 Writes to the PCI Express interface, this bit must be Cleared before the PCIMSTDATA register (USB Controller, offset 108h) is written. |
Attacking via PCIe
Demo, etc - WIP
Future Plans
Thunderbolt

Diagram: Apple Thunderbolt Device Driver Programming Guide
Thunderbolt

Photo credits: Chris Bergey via imgur.com
inception/funderbolt

http://www.breaknenter.org/projects/inception/
Questions?

Miles Crabill
@milescrabill
miles@milescrabill.com

Joe FitzPatrick
@securelyfitz
joefitz@securinghardware.com
http://www.securinghardware.com