PCI Bus

Understand a PCI Address

The PCI address is a series of numbers like 0000:04:00.0. The full form is
<domain>:<bus>:<device>.<function>.

The bus, device, and function numbers specify the device's location on the bus. We can break the device string "0000:04:00.0" down as follows:

- 0000 : PCI domain (each domain can contain up to 256 PCI buses)
- 04 : The bus number the device is attached to
- 00 : The device number
- .0 : PCI device function

Understand PCI Bus Structure

The 'lspci -tv' command shows a tree structure with limited verbosity.

```
# lspci -tv

-[0000:00]-+-00.0  Intel Corporation 8th Gen Core Processor Host Bridge/DRAM
Registers
           +-02.0  Intel Corporation CoffeeLake-S GT2 [UHD Graphics 630]
           +-08.0  Intel Corporation Xeon E3-1200 v5/v6 / E3-1500 v5 /
6th/7th/8th Gen Core Processor Gaussian Mixture Model
           +-12.0  Intel Corporation Cannon Lake PCH Thermal Controller
           +-14.0  Intel Corporation Cannon Lake PCH USB 3.1 xHCI Host
Controller
           +-14.2  Intel Corporation Cannon Lake PCH Shared SRAM
           +-16.0  Intel Corporation Cannon Lake PCH HECI Controller
           +-17.0  Intel Corporation Cannon Lake PCH SATA AHCI Controller
           +-1c.0-[01]----00.0  Realtek Semiconductor Co., Ltd.
RTL8111/8168/8211/8411 PCI Express Gigabit Ethernet Controller
           +-1d.0-[02]----00.0  Micron Technology Inc 2200S NVMe SSD
[Cassandra]
           +-1f.0  Intel Corporation H370 Chipset LPC/eSPI Controller
           +-1f.3  Intel Corporation Cannon Lake PCH cAVS
           +-1f.4  Intel Corporation Cannon Lake PCH SMBus Controller
           \-1f.5  Intel Corporation Cannon Lake PCH SPI Controller
```

The portion in brackets on the first line (***[0000:00]-+-00.0***) is the domain and bus that everything under this point in the tree is on. This is domain 0000 and bus 00, which is generally the root of the PCIe topology. The first couple of devices are directly on Bus 00.

However, the following devices are not directly on bus 0, but on Bus 1 and Bus 2 respectively:

- +-1c.0-[01]—-00.0 Realtek PCI Express Gigabit Ethernet Controller
- +-1d.0-[02]—-00.0 Micron Technology NVMe SSD

The Micron NVMe SSD is accessed through a device at 1d.0 on this bus, which thus has the (abbreviated) PCIe address of 00:1d.0. 'lspci -v' tells me that this is a PCIe bridge, as expected:

- 00:1d.0 PCI bridge: Intel Corporation Cannon Lake PCH PCI Express Root Port #9 (rev f0) (prog-if 00 [Normal decode])

Much like the []s of the root of the tree, the '[02]' bit after it in 'lspci -tv' means that all PCIe devices under this bridge are on bus 02, and there is only one of them, the NVMe drive, which will thus have the PCIe address 02:00.0:

- Unordered List Item02:00.0 Non-Volatile memory controller: Micron Technology Inc 2200S NVMe SSD

Query via sysfs

The Linux kernel represents PCI devices as pseudo-devices in the sysfs file system. Linux lists these devices in ***/sys/bus/pci/devices***.

```
# ls -al /sys/bus/pci/devices

total 0
drwxr-xr-x 2 root root 0 20 sep 14:25 .
drwxr-xr-x 5 root root 0 20 sep 13:53 ..
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:00.0 ->
../../../devices/pci0000:00/0000:00:00.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:02.0 ->
../../../devices/pci0000:00/0000:00:02.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:08.0 ->
../../../devices/pci0000:00/0000:00:08.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:12.0 ->
../../../devices/pci0000:00/0000:00:12.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:14.0 ->
../../../devices/pci0000:00/0000:00:14.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:14.2 ->
../../../devices/pci0000:00/0000:00:14.2
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:16.0 ->
../../../devices/pci0000:00/0000:00:16.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:17.0 ->
../../../devices/pci0000:00/0000:00:17.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:1c.0 ->
../../../devices/pci0000:00/0000:00:1c.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:1d.0 ->
../../../devices/pci0000:00/0000:00:1d.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:1f.0 ->
../../../devices/pci0000:00/0000:00:1f.0
```

```
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:1f.3 ->
../../../devices/pci0000:00/0000:00:1f.3
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:1f.4 ->
../../../devices/pci0000:00/0000:00:1f.4
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:00:1f.5 ->
../../../devices/pci0000:00/0000:00:1f.5
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:01:00.0 ->
../../../devices/pci0000:00/0000:00:1c.0/0000:01:00.0
lrwxrwxrwx 1 root root 0 20 sep 13:53 0000:02:00.0 ->
../../../devices/pci0000:00/0000:00:1d.0/0000:02:00.0
```

Query via lspci

```
# lspci -n

00:00.0 0600: 8086:3ec2 (rev 07)
00:02.0 0300: 8086:3e92
00:08.0 0880: 8086:1911
00:12.0 1180: 8086:a379 (rev 10)
00:14.0 0c03: 8086:a36d (rev 10)
00:14.2 0500: 8086:a36f (rev 10)
00:16.0 0780: 8086:a360 (rev 10)
00:17.0 0106: 8086:a352 (rev 10)
00:1c.0 0604: 8086:a33c (rev f0)
00:1d.0 0604: 8086:a330 (rev f0)
00:1f.0 0601: 8086:a304 (rev 10)
00:1f.3 0403: 8086:a348 (rev 10)
00:1f.4 0c05: 8086:a323 (rev 10)
00:1f.5 0c80: 8086:a324 (rev 10)
01:00.0 0200: 10ec:8168 (rev 15)
02:00.0 0108: 1344:5410 (rev 01)
```

We can break each line down like this:

Field 1 : 02:00.0 : bus number (02), device number (00) and function (0) Field 2 : 0108 : device class
Field 3 : 1344 : vendor ID Field 4 : 5410 : device ID

From:
https://wiki.oscardegroot.nl/ - **HomeWiki**

Permanent link:
https://wiki.oscardegroot.nl/doku.php?id=linux:system:pcibus:pcibus&rev=1758383782

Last update: **2025/09/20 15:56**