1. **PCI addressing: -**

Each PCI peripheral is identified by a **bus*number***, a **device*number***, and a **function number**.

A single system can hpst up to 256 buses. Each bus hosts up to 32 devices, and each device can have a maximum of 8 functions.

Since 256 buses are not sufficient for many large systems, Linux now supports PCI **address domains**. Each PCI domain can host up to 256 buses.

Example: -

$ cat /proc/bus/pci/devices | cut -f1

O/p –

0000

0001

….

00a0

Here, 0x00a0 can be decoded as 0000:00:14.0 when split into domain (16 bits), bus (8 bits), device (5 bits) and function (3 bits).

1. PCI ATU(Address Translation Unit): -

PCIe devices need to use PCIe addresses to send/receive packets over a PCIe link. The  
Address Translation Unit (ATU) within the PCIe module translates the device internal  
address into a PCIe address and vice versa. The PCIe address could be 32-bit or 64-bit  
(legacy EP may not support 64-bit addresses).  
For the outbound transaction, the outbound ATU translates the device internal address  
into a PCIe address. The data with a PCIe address is transferred over the PCIe link to  
the other device.  
For the inbound transaction, the Base Address Register (BAR) in the PCIe module  
accepts certain PCIe addresses and rejects the others. The data with an accepted PCIe  
address goes through the inbound ATU and is transferred to the device internal  
memory after address translation.

A diagram of a device

Description automatically generated with medium confidence

Reference: - [PCIe\_Docs\TI\_sprabk8.pdf](PCIe_Docs/TI_sprabk8.pdf) (Page 5)