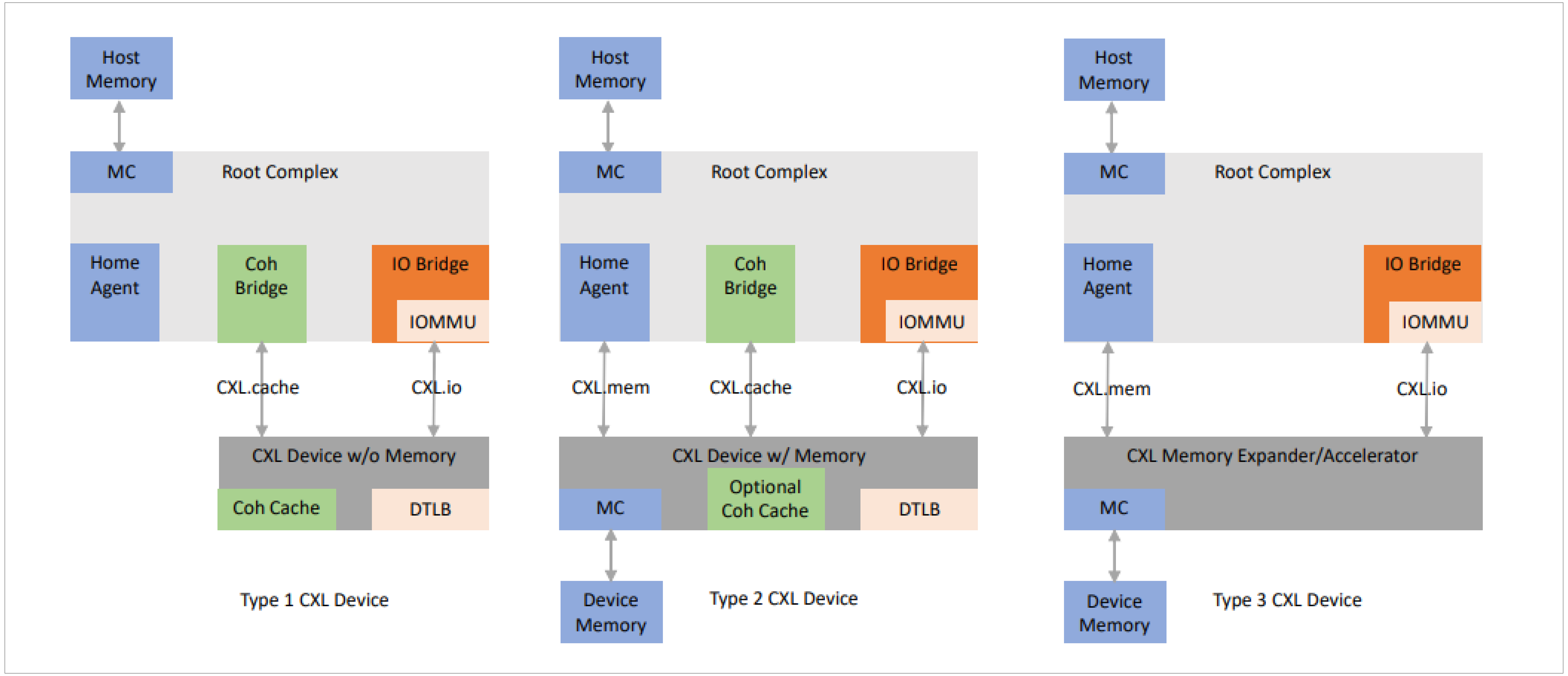
vfio-cxl: CXL Type 2 Device Passthrough With VFIO

Zhi Wang | KVM Forum 2024 / Sep 23, 2024

CXL Device Types



CXL 3.1 Specification - Figure 2-1 CXL Device Types

CXL Layering

Transaction Layer

- CXL.io utilizes the PCIe transaction layer.
- CXL.cachemem has their own transaction definition.

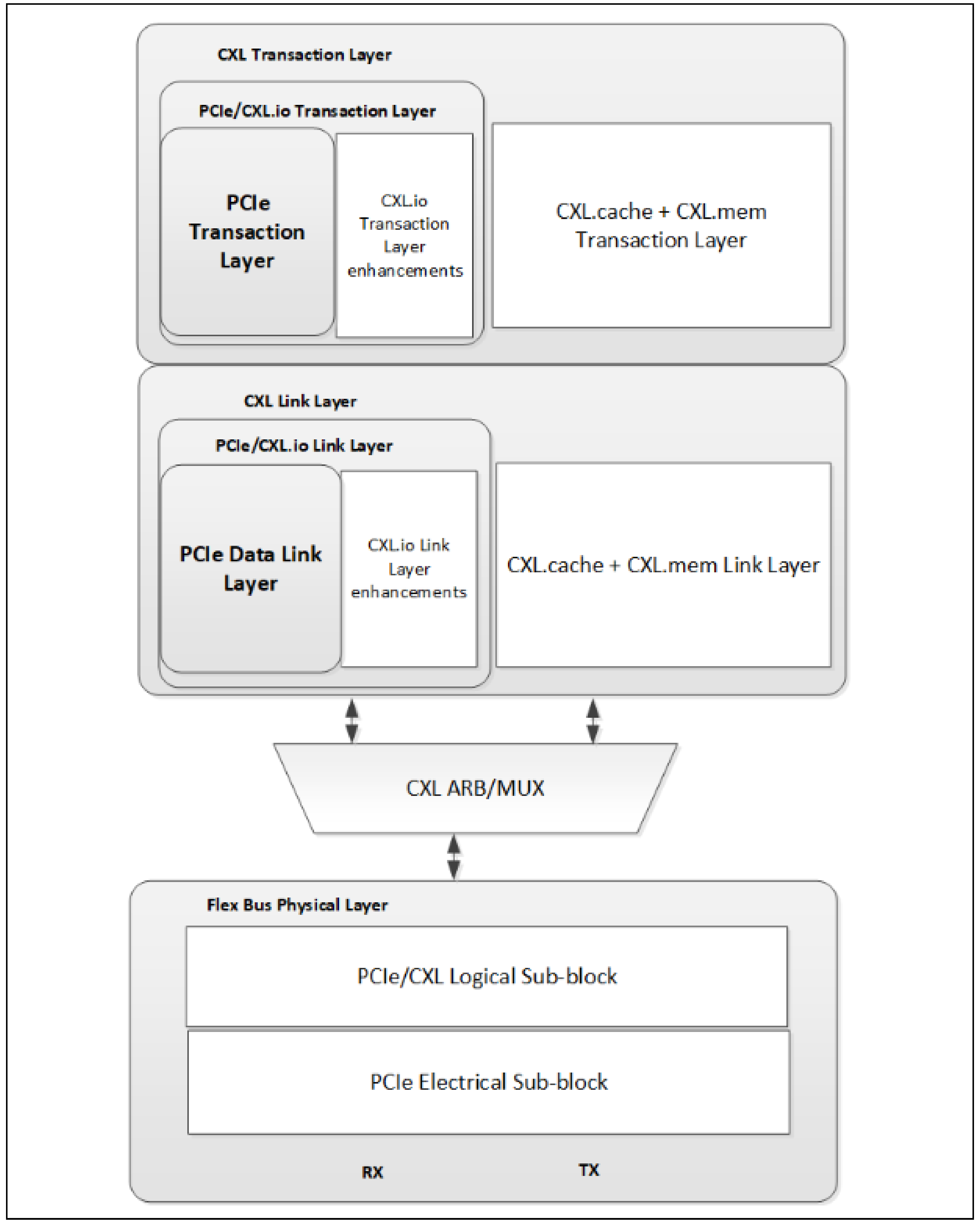
Link Layer

- CXL.io utilizes the PCIe data link layer.
- CXL.cachemem has their own encodings.

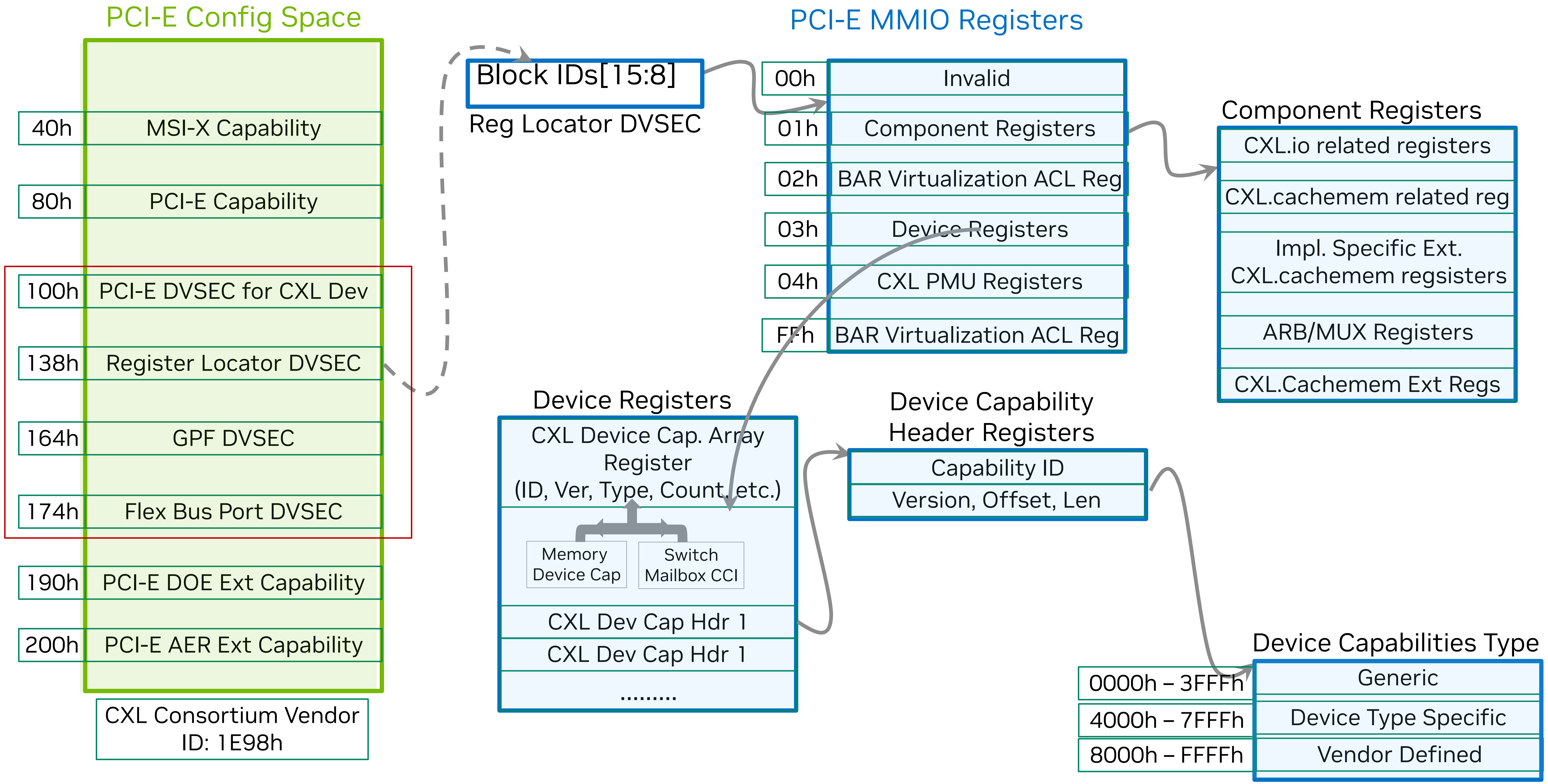
Physical Layer

- Flex Bus provides a point-to-point interconnect that can transmit native PCIe protocol or dynamic multi-protocol CXL to provide I/O, caching, and memory protocols over PCIe electricals.

Conceptual Diagram of Flex Bus Layering


CXL Transaction Layer
PCIe/CXL.io Transaction Layer
PCIe Transaction Layer
CXL.io Transaction Layer enhancements
CXL.cache + CXL.mem Transaction Layer
CXL Link Layer
PCIe/CXL.io Link Layer
PCIe Data Link Layer
CXL.io Link Layer enhancements
CXL.cache + CXL.mem Link Layer
CXL ARB/MUX
Flex Bus Physical Layer
PCIe/CXL Logical Sub-block
PCIe Electrical Sub-block
RX
TX

CXL Programming Interface



PCI-E Config Space

40h	MSI-X Capability
80h	PCI-E Capability
100h	PCI-E DVSEC for CXL Dev
138h	Register Locator DVSEC
164h	GPF DVSEC
174h	Flex Bus Port DVSEC
190h	PCI-E DOE Ext Capability
200h	PCI-E AER Ext Capability

CXL Consortium Vendor ID: 1E98h

Block IDs[15:8]

Reg Locator DVSEC

PCI-E MMIO Registers

00h	Invalid
01h	Component Registers
02h	BAR Virtualization ACL Reg
03h	Device Registers
04h	CXL PMU Registers
FFh	BAR Virtualization ACL Reg

Component Registers

- CXL.io related registers
- CXL.cachemem related reg
- Impl. Specific Ext. CXL.cachemem regsisters
- ARB/MUX Registers
- CXL.Cachemem Ext Regs

Device Registers

- CXL Device Cap. Array Register (ID, Ver, Type, Count, etc.)
- Memory Device Cap
- Switch Mailbox CCI
- CXL Dev Cap Hdr 1
- CXL Dev Cap Hdr 1
- ....

Device Capability Header Registers

- Capability ID
- Version, Offset, Len

Device Capabilities Type

0000h – 3FFFh	Generic
4000h – 7FFFh	Device Type Specific
8000h – FFFFh	Vendor Defined

More about CXL?

- CXL Specification

https://computeexpresslink.org/cxl-specification/

- Subscribe to CXL Mailing List

linux-cxl@vger.kernel.org

- CXL discord channel

https://discord.gg/6tNRp37B

- CXL Linux: Open Source Collaboration Sync – Monthly meeting

Why VFIO For CXL Type-2 Device?

“Device” passthrough is the requirement.

Programming models are compatible with each other

- The programming model of the CXL type-2 accelerator is more like operating a device.
- VFIO is the standard component that allows the guest OS to access the device directly via device pass-through.

Virtualization models are compatible with each other

- The virtualization model for the CXL type-2 accelerator follows the common I/O virtualization approach.
  - Assign the device into a virtual machine.
  - Partition, manage the HW resource, and expose it via virtual functions.
- VFIO supports device/VF passthrough and advanced features, e.g., live migration.

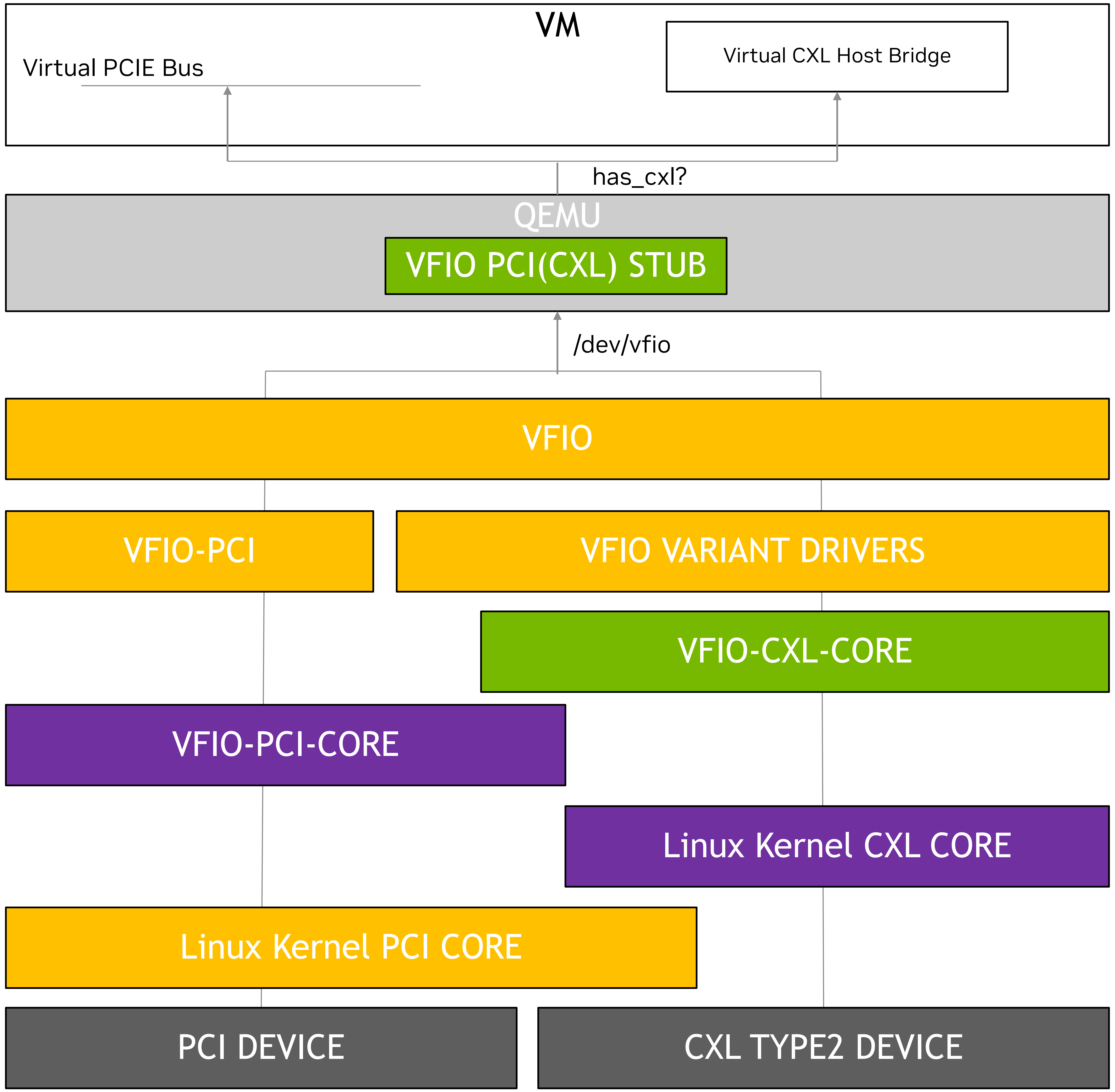
Building type-2 device passthrough on existing efforts

- CXL.io utilizes the PCIE transaction layer. Thus, the CXL device has a similar HW programming interface as the PCIE device.
- VFIO has the fundamental virtualization policies for managing the PCIE device interface, management ABIs for the user-space

VFIO-CXL Architecture

Virtualization Polices

- CXL Configuration Space Regs – Emulated
- CXL MMIO Regs – Mediated Passthrough



VM

Virtual PCIE Bus

Virtual CXL Host Bridge

has_cxl?

QEMU

VFIO PCI(CXL) STUB

/dev/vfio

VFIO

VFIO-PCI

VFIO VARIANT DRIVERS

VFIO-CXL-CORE

VFIO-PCI-CORE

Linux Kernel CXL CORE

Linux Kernel PCI CORE

PCI DEVICE

CXL TYPE2 DEVICE

- Enhance the QEMU VFIO PCI stub to support CXL devices

- Introduce a VFIO-CXL-CORE for the variant driver to handle the CXL device.

- Introduce the CXL aware to VFIO-PCI-CORE for special handling of CXL device

- Expose functions required by VFIO-CXL-CORE from Linux CXL core

🟩 New Components

🟨 Existing Components

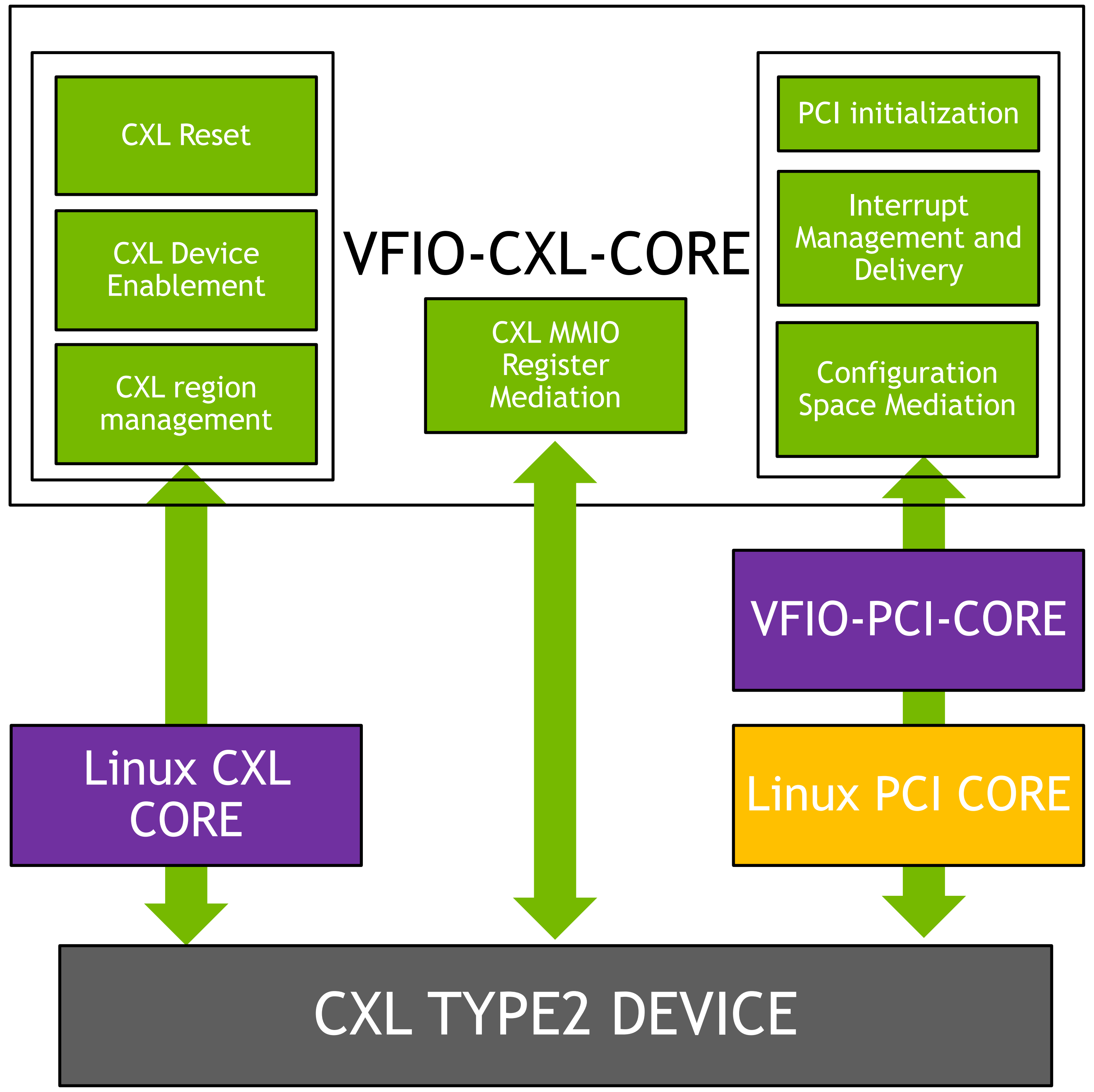
🟪 Changed Components

Introduce VFIO CXL CORE

The core functions for the VFIO variant driver

A core helper layer for the VFIO variant driver to manage the CXL type-2 device

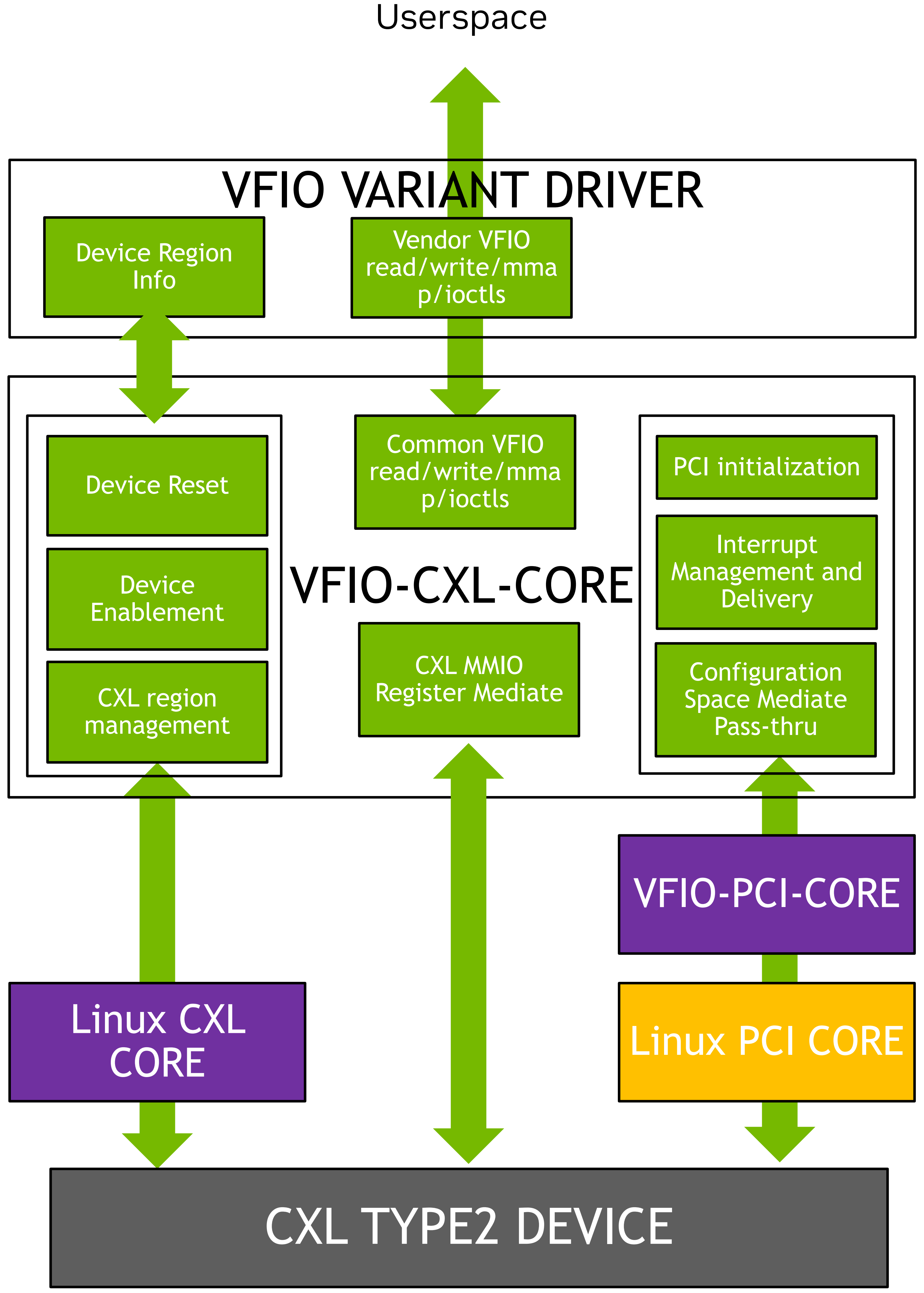
- CXL Reset
  - Reset the CXL type-2 device via the help of CXL core
- CXL Device Enablement
  - A type-2 device driver initialization flow to set up the device together with the kernel CXL core
- Interrupt Management And Delivery
  - Re-use VFIO PCI CORE interrupt handling routines
- CXL Region Management
  - Acquire the CXL region info from the CXL core
  - Expose the CXL region to userspace
  - Implement CXL region ops: read/write/mmap
- Configuration Space
  - Re-use VFIO PCI CORE manages the configuration space registers
- CXL MMIO register
  - Re-use VFIO PCI CORE manages the BAR MMIO registers
  - For CXL MMIO registers, emulate CXL HDM decoder registers



VFIO Variant Driver For CXL Type-2 Device

The VFIO variant driver is responsible for:

- Provide device info
  - Info needed by kernel CXL core to configure the device
    - DPA size
    - RAM size
    - Driver cap
  - Info for the user space
    - CXL region size
- Vendor-specific VFIO device ops (optional)
  - Vendor-specific CXL routines such as CXL reset and cache control.



Required VFIO PCI CORE changes

CXL Device Awareness

- Introduce CXL device awareness in VFIO PCI CORE

Interrupt Handling and Delivery

- Disable INTX interrupt support for CXL type-2 device

CXL MMIO BAR MMAP Probing

- VFIO CXL CORE handles the CXL MMIO BAR for CXL type-2 device
- VFIO CXL CORE emulates CXL HDM decoder registers

Device Reset

- VFIO CXL CORE handles the device reset for the CXL type-2 device
- VFIO CXL CORE performs CXL reset when QEMU requests DEVICE_RESET

Required QEMU changes

Building CXL type-2 device passthrough support on existing QEMU VFIO stub

QEMU VFIO Stub Changes

- Introduce CXL device awareness
- Emulate DOE with CDAT from the device
- CXL device setup
  - Acquire CXL device info from the VFIO CXL CORE
    - HDM decoder register block offset and BAR
    - HDM count
    - DPA size
  - Initialization of the CXL region exposed by the VFIO CXL CORE
- Virtual HDM decoder commit
  - Map the CXL region according to the GPA in the virtual HDM decoder registers

CXL Device Support Upstream Status

Kernel

- In-tree CXL type-3 device support
- Dan William's CXL type-2 device support (reviewed but not in-tree)
  - Remove the assumptions that a CXL device is a type 3 device
  - Refactor the region creation
  - DPA and HPA management
- Alejandro Lucero Palau(AMD) takes over Dan's patch and posted CXL type-2 device support v2
  - Patch series is under review
  - Patch series title: [PATCH v3 00/20] cxl: add Type2 device support

- QEMU
  - Support emulating CXL 2.0 topology and type 3 device
  - Qemu RFC patches to emulate the type-2 device
    - [PATCH RFC 0/5] hw/cxl: Type 2 Device RFC - Ira Weiny (kernel.org)

Required Linux CXL Core Changes

Changes for VFIO CXL CORE

- Query of device/register info
  - HDM count
  - Offset/size of the component register block
  - Offset/size of HDM decoder register block

Changes for CXL type-2 device support

- based on Alejandro Lucero Palau(AMD)’s patchset
- Remove the assumption of register groups for CXL type-2 device

The RFC Patchset

- Goal
  - Provide a code base to discuss vfio-cxl architecture
  - Provide an environment that developers can test with

- Status
  - Can test the RFC patches with nested virtualization with virtual device passthrough
    - QEMU host emulates a generic cxl-type 2 device
    - VFIO cores and variant driver are loaded in L1 with QEMU L1 with VFIO PCI stub support CXL
  - CXL type-2 device init sequence is based on Alejandro Lucero Palau(AMD)'s RFC patchset
    - Alejandro's patch set is still under review.

- Call for comments
  - Kernel CXL core changes
  - VFIO-PCI-CORE changes and VFIO-CXL-CORE APIs
  - UABI for QEMU
  - Device vendor's requirements
  - What should be included for PATCH V1

Checklist - What Is In The RFC Patchset

Patchset: https://lore.kernel.org/kvm/20240920223446.1908673-3-zhiw@nvidia.com/T/

Demo video: https://youtu.be/zlk_ecX9bxs?si=qJdGAtPH0KgtmRqf

- VFIO CXL CORE
  - CXL device enablement
  - CXL region management
  - Generic VFIO device ops read/write/mmap/ioctl
  - PCI initialization (based on VFIO PCI CORE)
  - Interrupt management delivery (based on VFIO PCI CORE)
  - HDM decoder emulation
- Example VFIO variant driver for the virtual cxl type-2 device
- VFIO PCI CORE changes
  - CXL device awareness and special handling
- QEMU
  - Introduce CXL device awareness
  - CXL device setup
  - Virtual HDM decoder commit

NVIDIA®