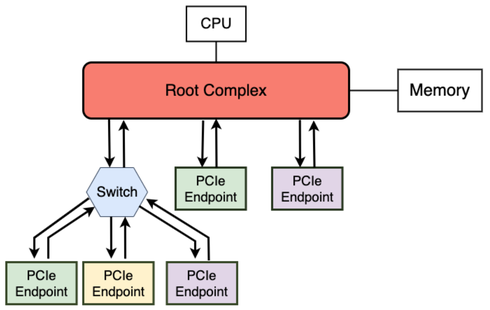
Exploring Industry Standard System Integration for HW/SW Co-design

Joonho Whangbo, Kevin Anderson, Coleman Hooper

Abstract Modeling of Complex Communication Protocols for High-Performance Hardware Simulation

Motivation

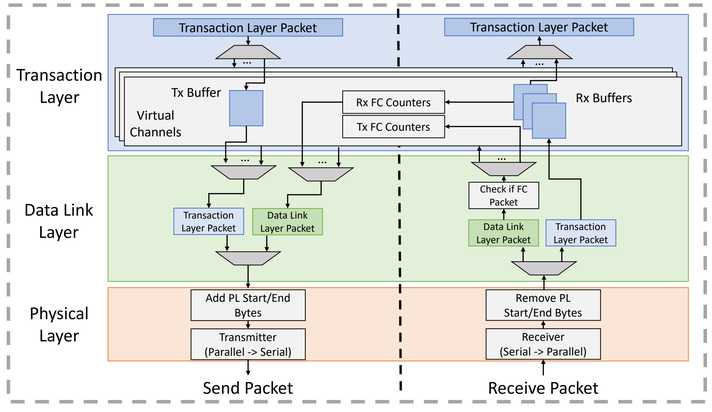
- PCIe is a P2P serial computer expansion bus standard widely adopted by industry
  - 3 Layer Stack: Transaction, Data Link, Physical
  - Common PCIe devices: Network Interface Controller (NICs), disk drives, graphics cards (GPUs), accelerators
- Existing models do not both provide both functional and timing modeling underneath a comprehensive framework for system design exploration



Objectives

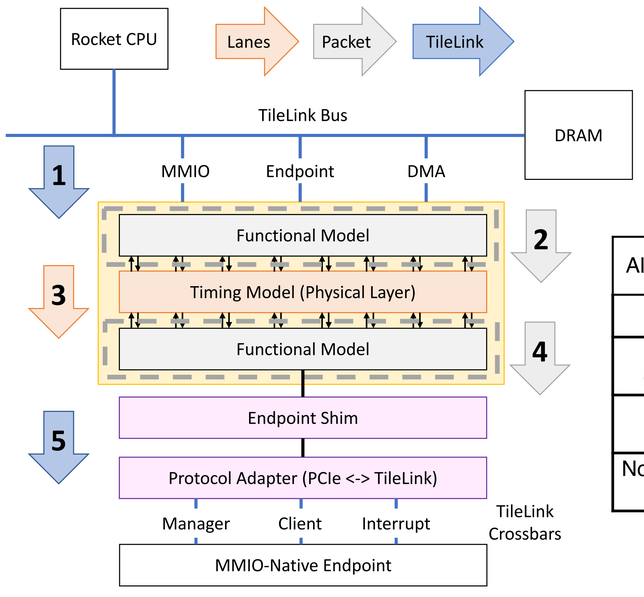
- Create a **synthesizable**, **performance validated** PCIe model for future academic research (accelerators, chiplets, etc)
- Integrate model within FireSim to enable cycle-exact execution of our model
- Create protocol adapters to allow users to test PCIe integration with existing accelerators
- Boot Linux with FireSim and exercise accelerator attached to model

Functional Model



System Architecture

- Rocket CPU - host
  - Configured using Chipyard
- TileLink-Based Communication
  - CPU-RC connections: MMIO, Endpoint, DMA
  - RC-Endpoint connections: Protocol Adapter and Xbars
- Root Complex Model
  - Model of 3-layer protocol stack
  - Timing model (latency)
- Endpoint
  - Contains PCIe functional model (protocol stack)
  - Translation layer (shim) for compatibility with non-PCIe devices (additional feature)
- Protocol Adapter (Shim)
  - Ease of use; plug-and-play capability
  - Connects MMIO-native devices to PCIe w/o modifying interface
  - Handles interrupts, converts them to PCIe TLPs



Arrows show how an MMIO write is converted to a packet, serialized, and then reconstructed to be sent to the endpoint

All TLP types	ECAM (limited)
MSI	Flow Control
8 Virtual Channels	Up to 16 Lanes
DMA	Address Routing
Non-fetchable Memory	ACK/NACK

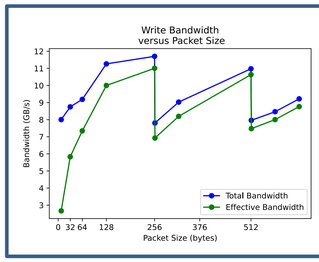
Table 1: Supported core functions in model

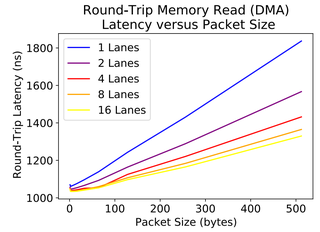
Hardware Infrastructure

Tool	Uses
FireSim	Cycle-exact simulation on hardware (FPGA)
Chipyard	SoC design, simulation, verification platform
Spike/QEMU	RISC-V ISA simulator for driver testing
FireMarshal	Workload generation tool (builds Linux distros)
TileLink	Interconnect protocol used within the SoC
Metasimulation	Software RTL simulation of FireSim

Table 2: Tools utilized for design, simulation, and verification

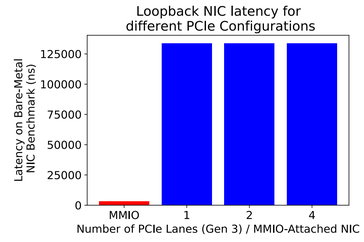
Model Validation





Case Study - NIC

- IceNIC (Chipyard peripheral)
  - NICs are commonly integrated over PCIe
  - Close-core NIC that typically sits over TileLink
  - Objective: analyze how performance characteristics change when it is attached over PCIe
  - Used protocol adapter (natively an MMIO device)



Linux Driver

- Bare-metal sufficient for simple, but limited tests
- Driver development in Spike, QEMU and FireMarshal
- Developing Linux driver to support core operations
  - Better integration into FireSim/Chipyard platform
  - Increased usability for complex devices
  - Leverage existing code

Future Work

- Project WIP:
  - Finish Linux Integration and run everything full-stack
  - Comparative analysis of the benefits of modelling flow control on bandwidth/latency
- Future Research:
  - CXL modeling
  - PCIe-to-RoCC shim
  - Improve Linux driver (supporting firmware)