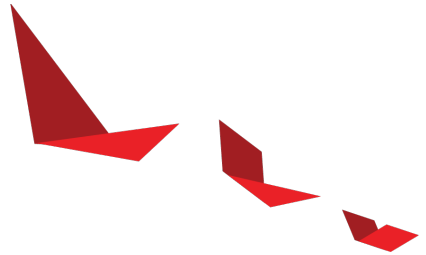
XILINX

QEMU Overview

Edgar E. Iglesias
Principal Software Engineer
April/19/2022

Agenda Title



- QEMU Overview
- Co-simulation / Hybrid
- Plans
- Demos

QEMU Overview

Modeling Technologies: Software Emulator (QEMU)

Virtual Platform for SW developers

▪ Open-Source	**Scalable distribution & cost model,** Popular in Open-Source community
▪ @Xilinx from 2009	MicroBlaze, Power PC, **ARM (Zynq, Zynq MPSoC, Versal)**, x86
▪ Transaction level	Fast but not cycle accurate, Linux boot **2sec to user-space, 2min to prompt**
▪ Debug & Profiling	GDB/XSDB, traces, code-coverage and error-injection (-ve testing)
▪ Co-simulation	SystemC/TLM-2, RTL and Hybrid
▪ Value	Shift left (early SW development), Cost, Speed and Flexibility

Users

▪ Internal	BootROM, System Software, SVT
▪ External	Petalinux, Vitis HW-Emulation, GitHub (Roll your own)

Success stories

▪ Zynq MPSoC	Xilinx a significant contributor to ARMv8, MicroBlaze, RegAPI, Reset, Clocks etc
▪ DARPA/POSH	Xilinx chosen for Open-Source QEMU Co-simulation efforts
▪ Customers/Vendors	X (PetaLinux), Y (Github) and Z (Vitis)

QEMU modes

System Emulation

- Emulation of full system (TCG)
  - Including Virtualization & Security, heterogenous cores, TZ, IOMMU's, XMPU's, PMU, PMC etc
  - DTB machines

Hypervisor KVM

- HW accelerated virtualization
- ZU+, Generic ARM virtual board
- x86 + PCIe co-simulation

Linux-user

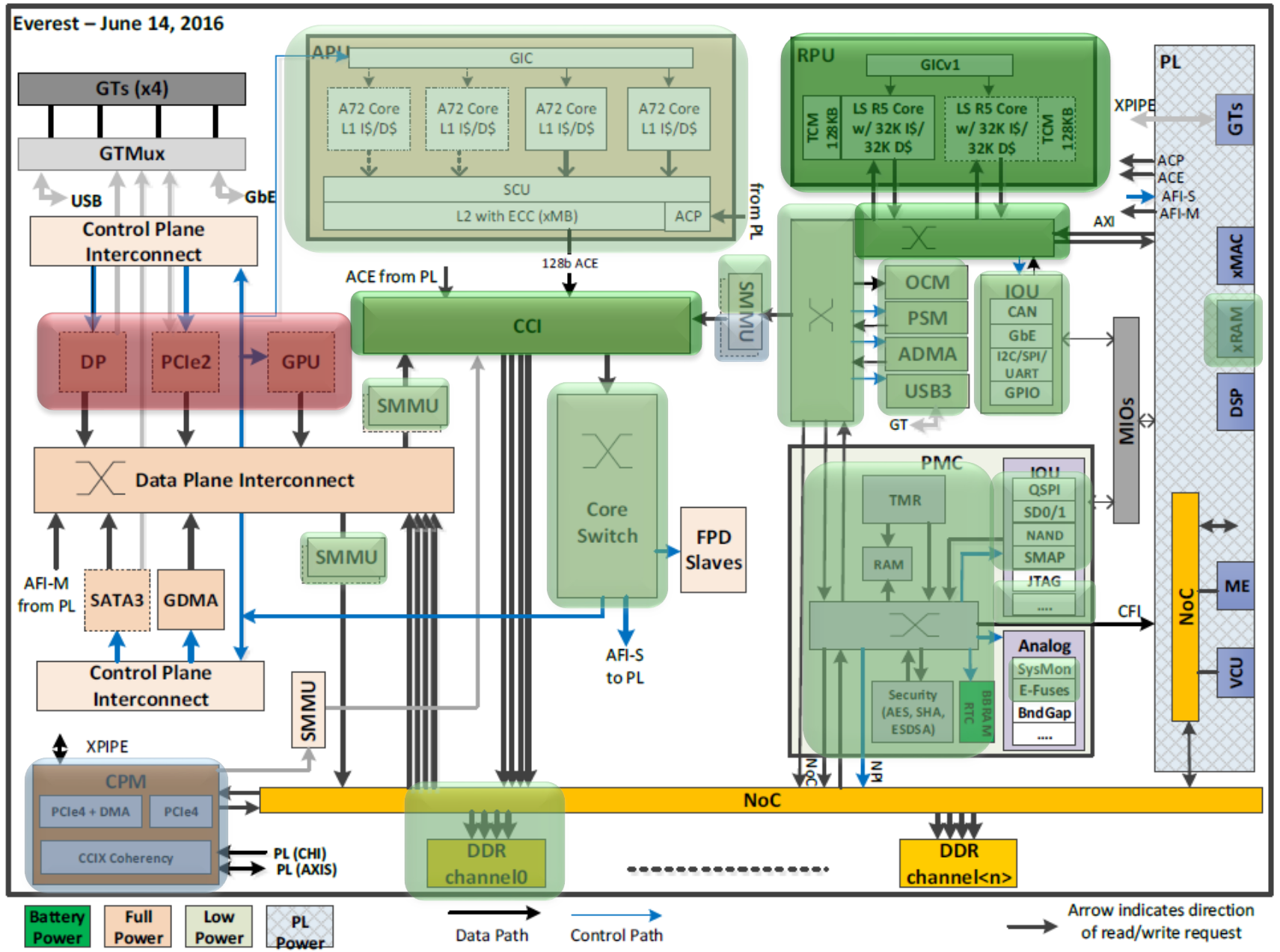
- Cross-run Linux binaries (e.g ARM Linux binary on x86 Linux host)
- GCC test-suite, Cross-compilation, Rootfs testing

Done or mostly done

In progress

Planned but not Started

Not Planned



XILINX.

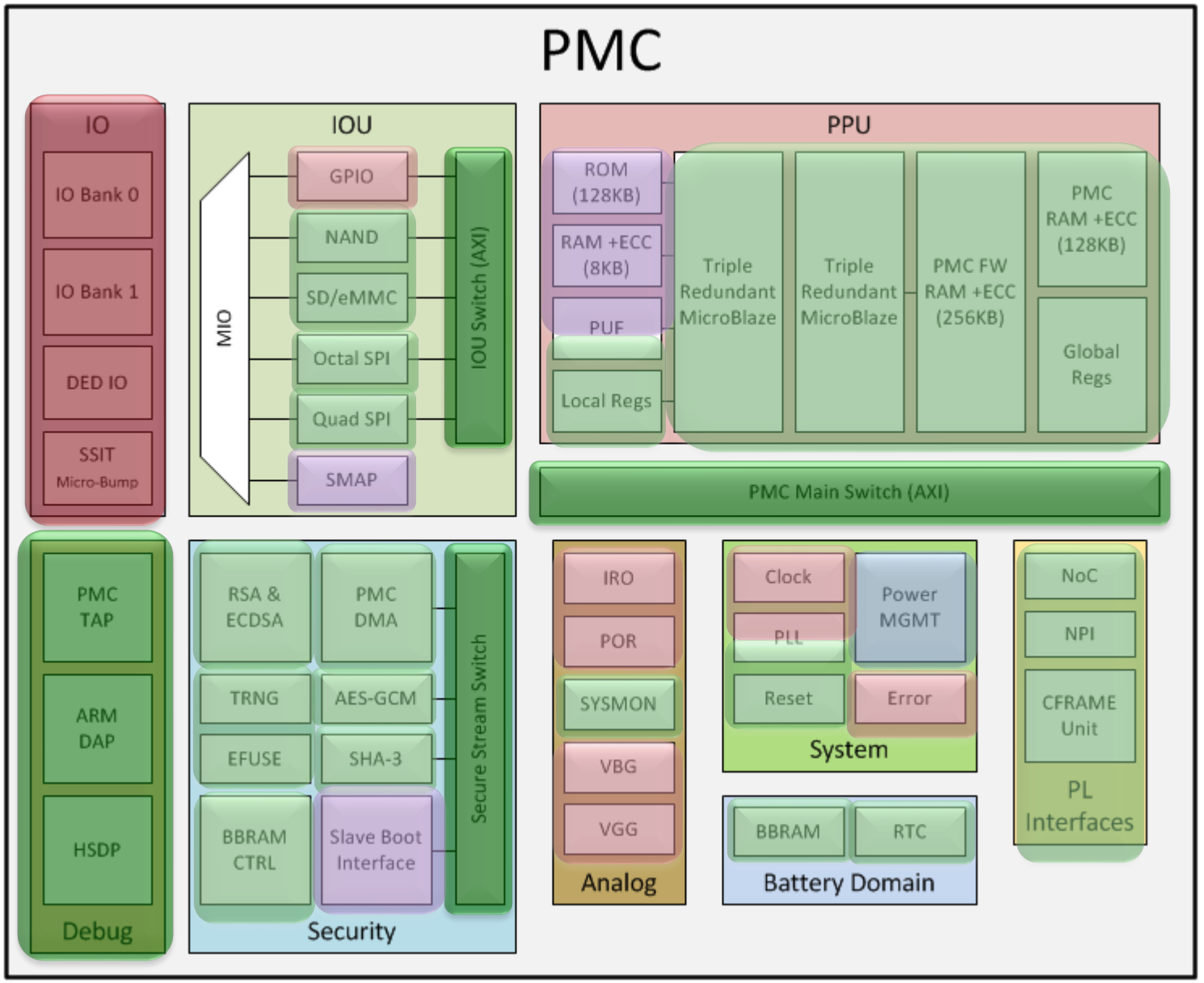
Done or mostly done

In progress

Planned but not Started

Not Planned

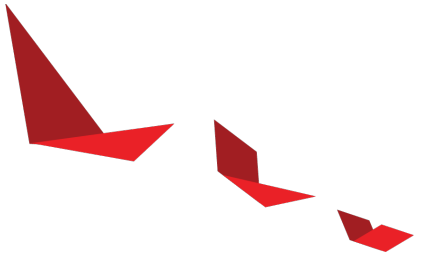
Internal Only



- PUF HW model internal only
- PUF Firmware model public
- SBI depends on BootROM (internal-only)

XILINX.

QEMU System Emulation



Boot

- Full boot through BootROM (internal only)
- FSBL, U-boot & Linux direct boots

Security

- CSU, PMC-Secure, crypto blocks, EL3/TZ
- Secure-boot depends on BootROM images (internal)

QEMU System Emulation

Debuggers

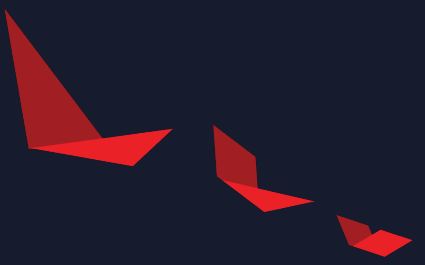
- GDB
- XSDB (PetaLinux, Vitis and Vivado)

Testing

- QTX - Pytest based test-suite – 500 cases
- TBM – Baremetal test and exploration environment
- PetaLinux 500+ testcases, Vitis 500+ co-simulation test-cases

Tools

- Etrace - tracing and code coverage
- Xregdb, irqmap-gen, memmap-gen – Machine and model skeleton/generation

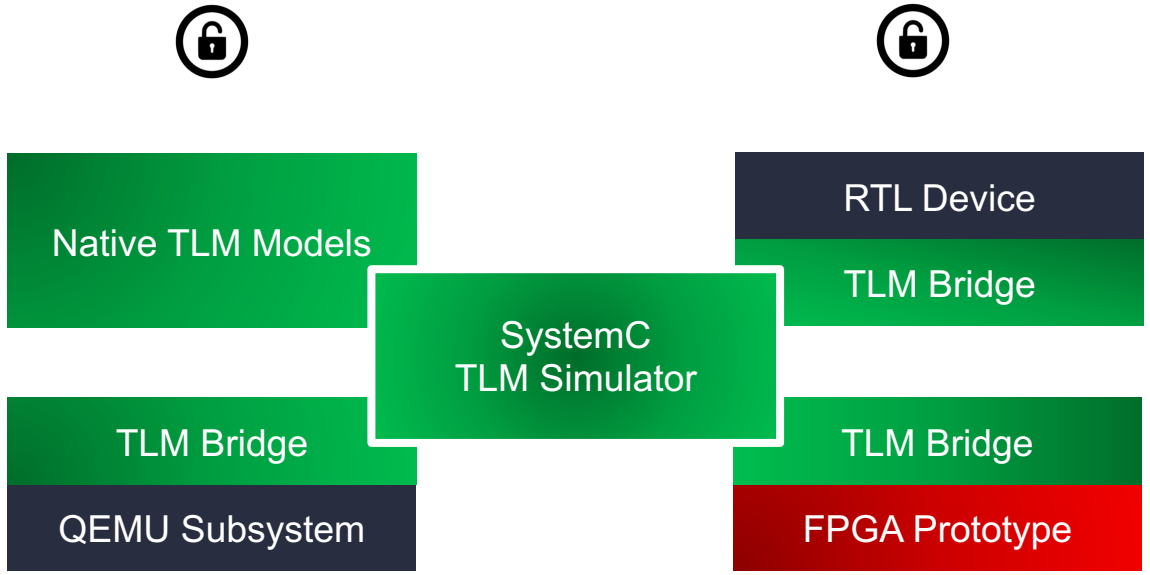


Co-simulation / Hybrid

Open Source Mixed Simulation Environment

LibSystemCTLM-SoC

https://github.com/Xilinx/libsystemctlm-soc



Phase 1a

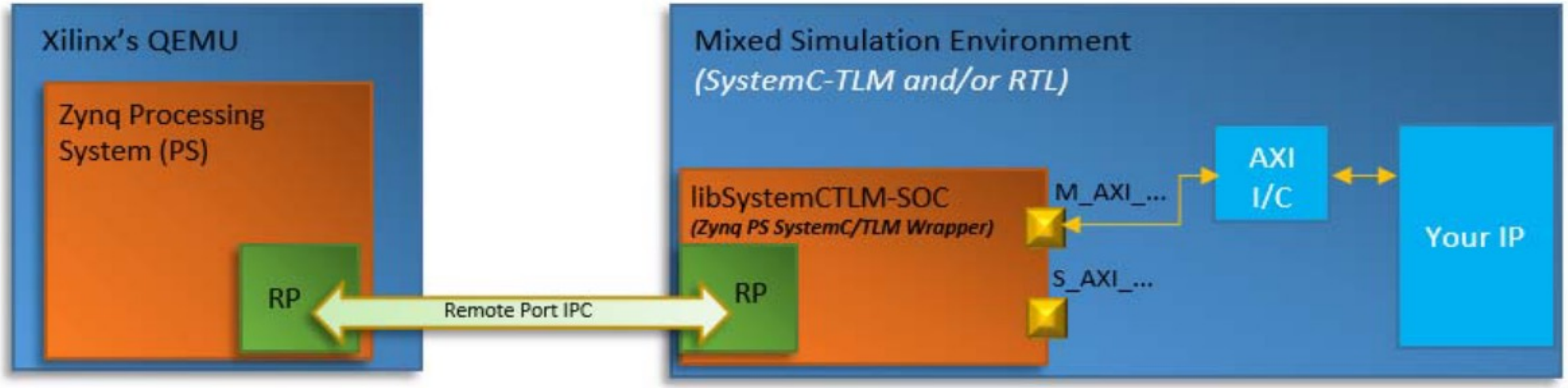
Phase 1b

Based on Open-Source Projects

QEMU accellera eclipse VERILATOR Kactus2 pytest GNU Debugger

XILINX

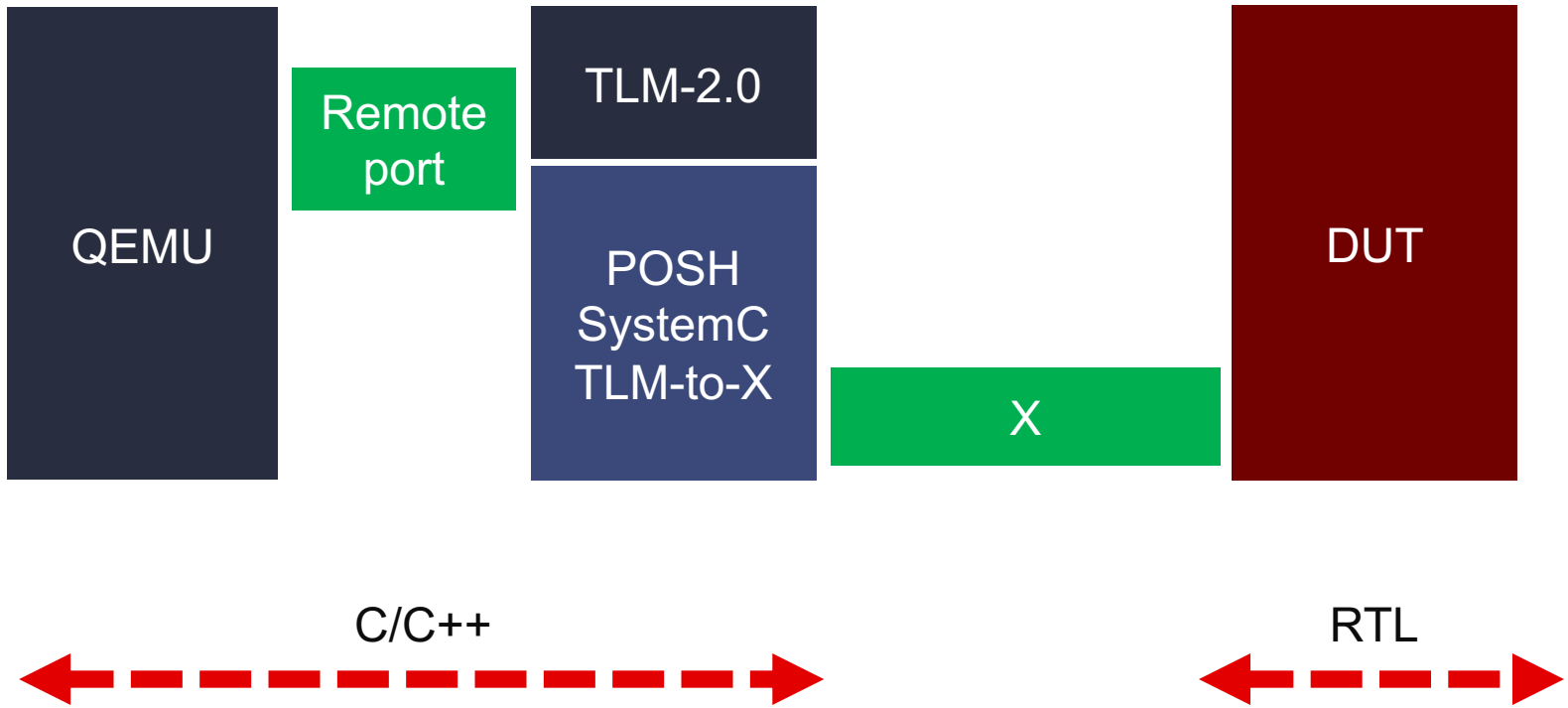
Xilinx Vivado/Vitis Co-sim



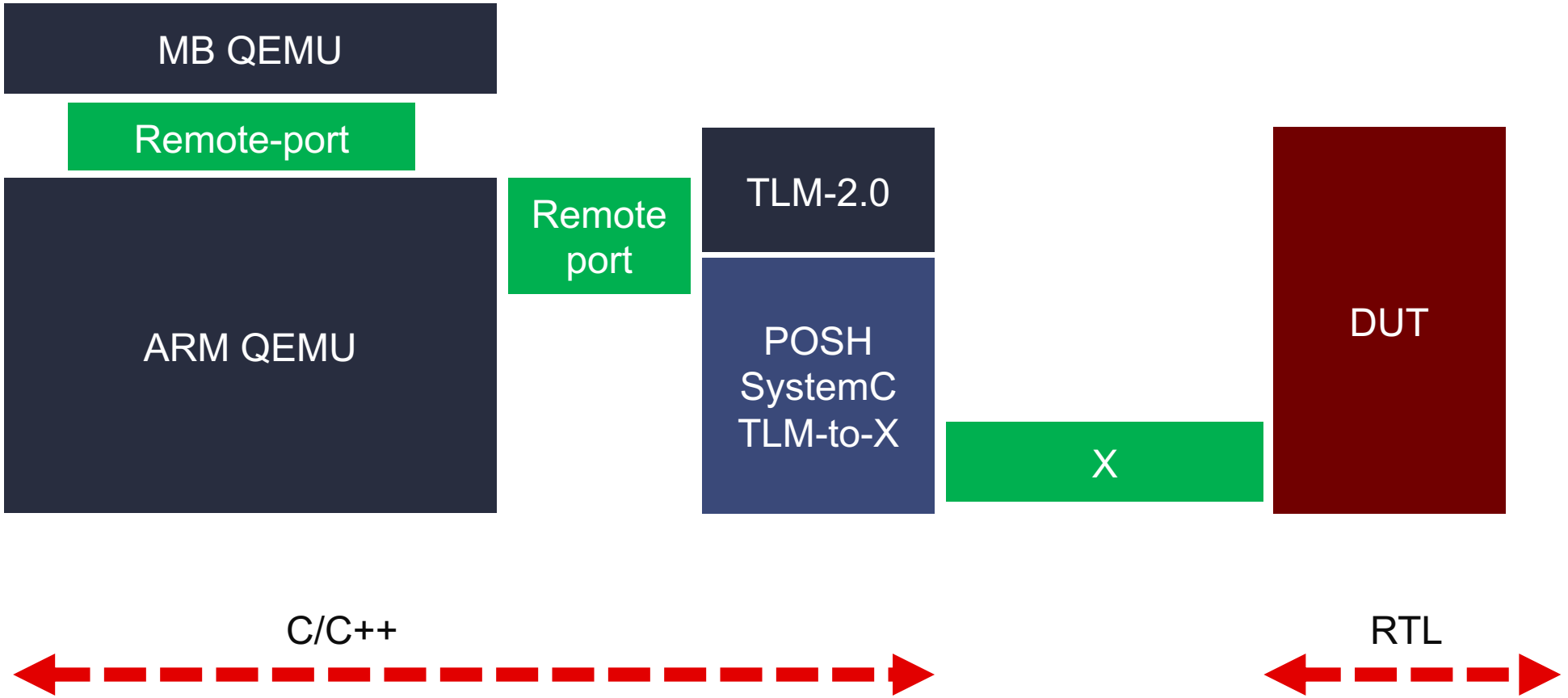
- Unique Process
- Xilinx's Remote Port
- Xilinx's Zynq PS **or** PS SystemC/TLM Wrapper
- Zynq PS –PL AXI Master/Slave Ports
- Your IP(s)

XILINX.

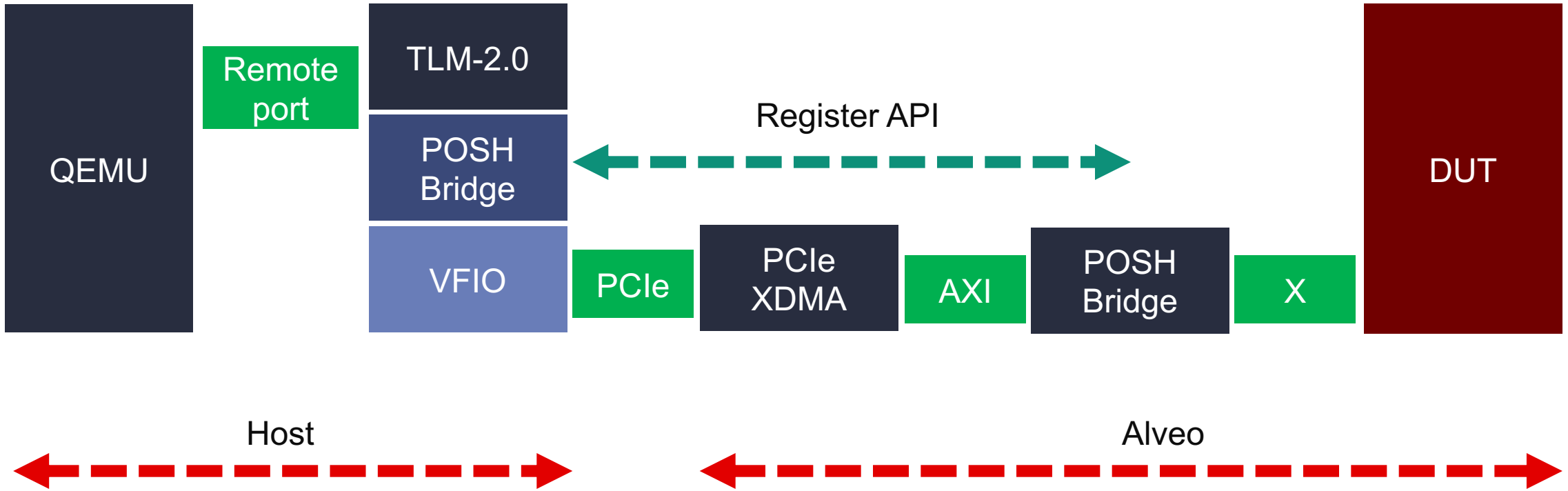
TLM2 bridges RTL simulation



QEMU heterogeneous

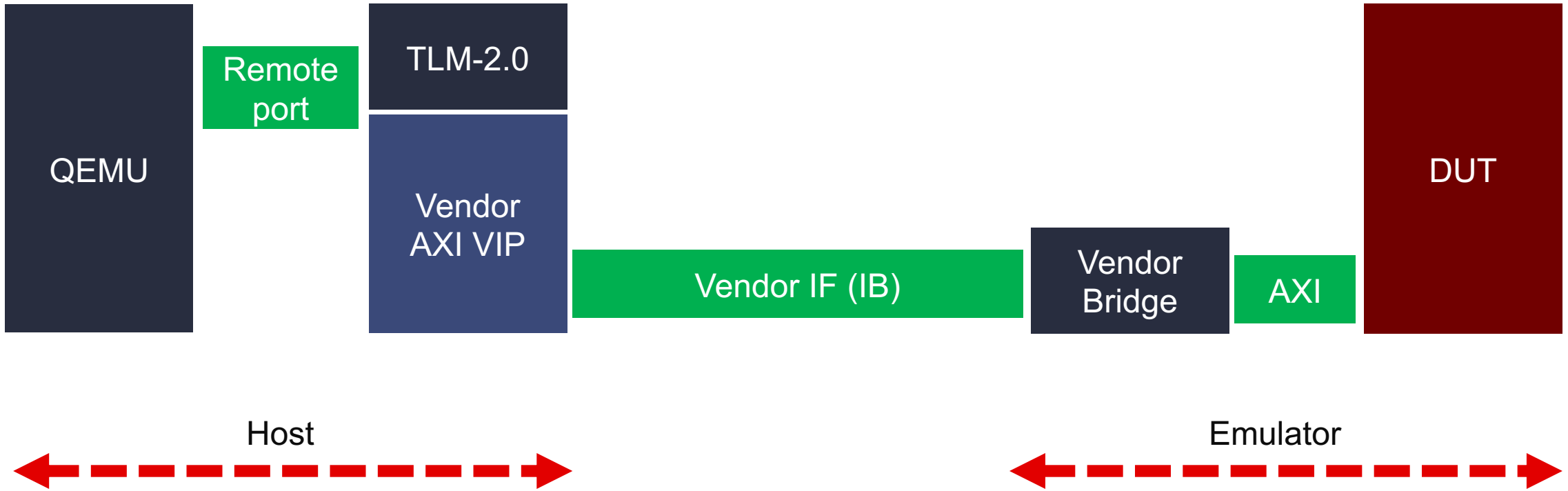


TLM2 bridges FPGA Prototyping



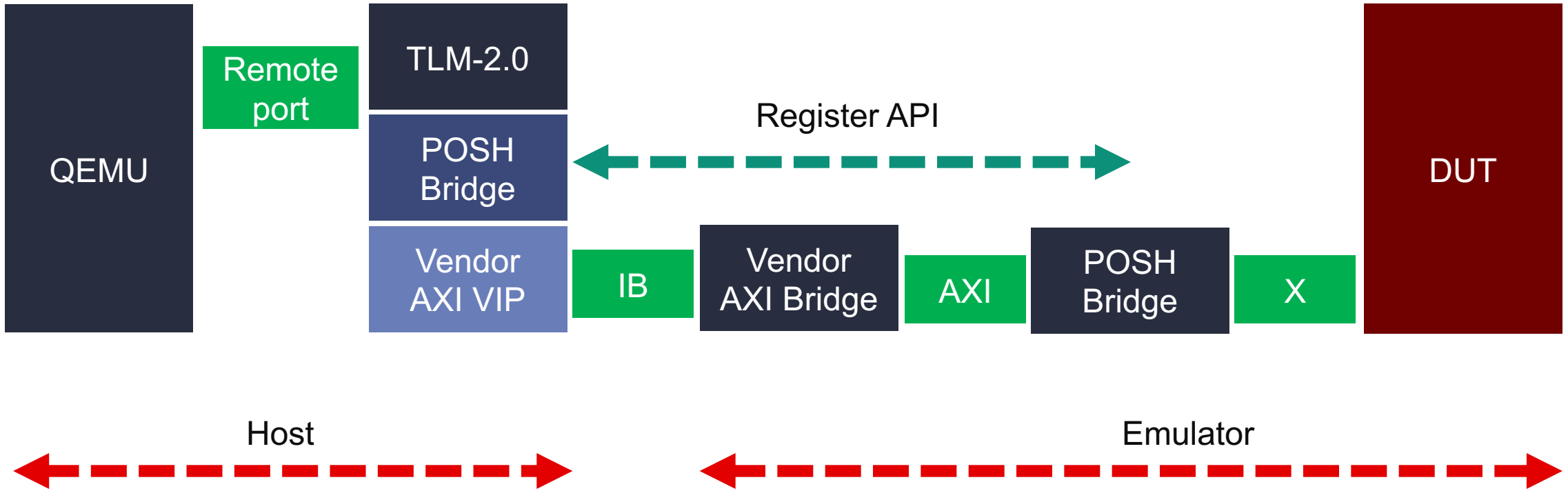
TLM2 bridges Emulator (Initial idea)

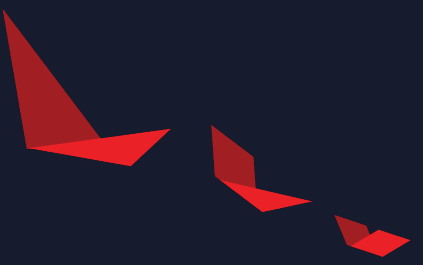
Pros: Fast, sort of “Vendor neutral”
Cons: Not debug-portable, Licensing AXI VIP



TLM2 bridges Emulator

Pros: Vendor neutral, debug-portable
Cons: Slow, Licensing AXI VIP





Plans around QEMU upstream

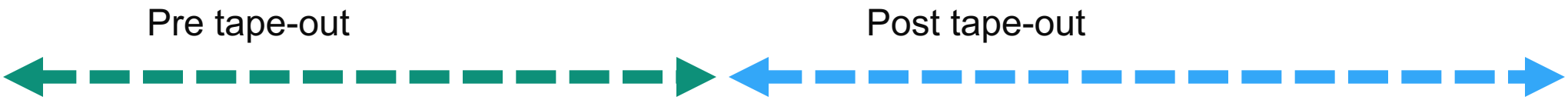
Upstream

Heterogenous CPUs Done

Heterogenous CPUs Needed

CPUs	MicroBlaze	ARMv8 (EL2, EL3)	PPC 440 BookE	MIPS (R32 SMP, SMT), CRIS	MTTCG, Per CPU AS	CPU Clusters, GDB Multi-process	Heterogenous CPU build and instance support (submitted)
Interrupt controllers	Xilinx Intc	GICv2 (Virt), ZynqMP IPI	PPC XInx Intc	EtraxFS PIC		Multiple GICs	
Peripherals	Xilinx UARTs, Cadence UART, DMA, Display	SPI, QSPI, Octal SPI, m25p80, NAND	SDHCI, eMMC, AHCI Sysbus	USB Sysbus, ZynqMP, Versal	Ethernet (MACb, Ethlite, AXI eth), EtraxFS, MDIO, PHYs	PMU, PMC. PSM devices	
Frameworks	Register API	Clock API	Reset API	Named GPIOs/Interrupts	Streaming API	Generic CPU reset & halt (GPIO based)	
Machines	Zynq, ZynqMP, Versal, PetaLinux MB, Virtex ML507 etc	Versal Cortex-A72 + Cortex-R5F CRL (Dynamic wake-up of RPU)	QMP driven machines (ongoing), DTB driven	ZynqMP, Versal Cortex-R5F GIC, IPI connections	PMU, PMC, PSM subsystems		

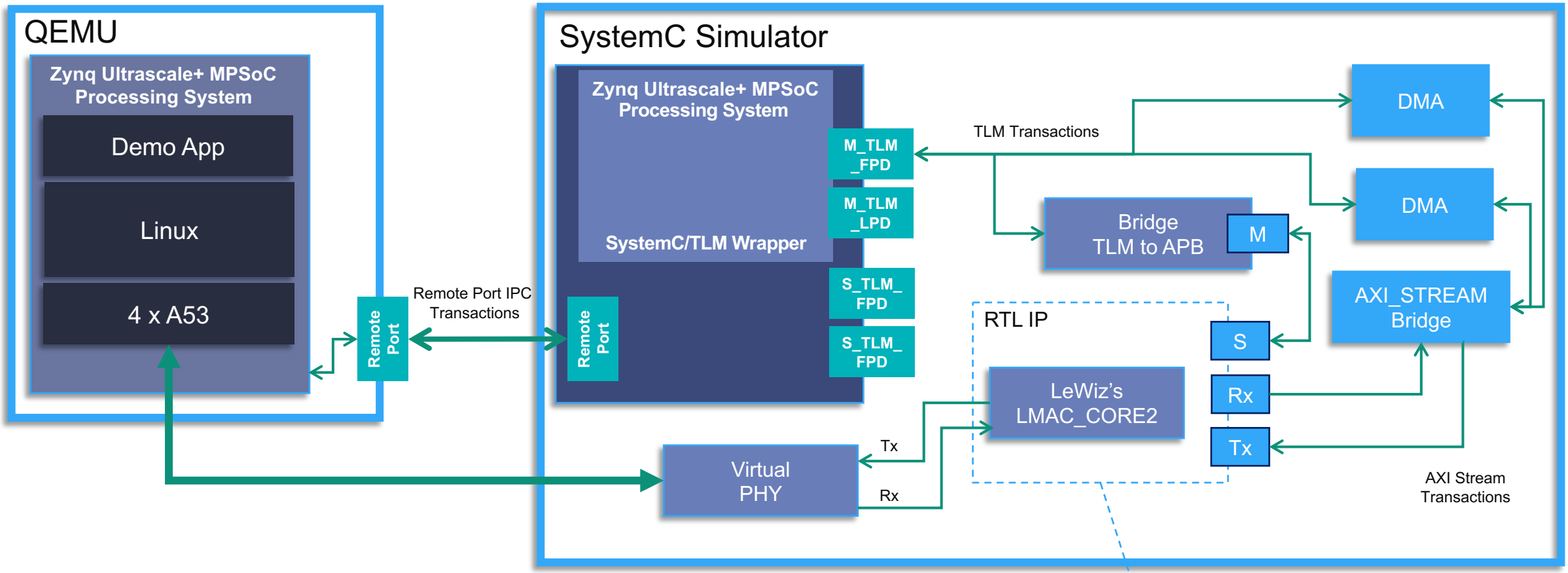
Programs



Pre-RTL	RTL Emnulation and FPGA Prototyping	Internal/Not public	Public
QEMU is the only platform SSW Bring-up BootROM development Arch exploration Prep for Emu and Proto Boot Firmware, TF-A, U-boot, Xen, Linux	Migration towards RTL QEMU still used due to productivity gains (turnaround speed) and due to RTL development lag CI loops SW/HW/QEMU More complex SSW setups	Prep for VnC Integrate into PetaLinux, Vitis, Vivado Prep for upstreaming	Open-Source community Early-Access programs Customer support Plans 2022.2: QEMU upstream: Versal eMMC Versal SSS and SHA Versal CAN-FD M25p80 Octal SPI offchip dev ZynqMP Display-Port fixes ZynqMP USB minor alignments ZCU102 i2c clock-devs ZCU102 DDR SODIMM SPD QMP machines (GreenSocs) Libsystemctlm-soc: PCIe core (TLP level) TLM Versal PCIe end-point model including PS/PMC

Demo

DARPA/POSH



Main Lessons Learnt

- Need Backend connections to the RTL/TLM to feed in real data
- Need for AXI Stream and APB Protocols
- Need to standardize on a particular set of protocols….AXI, Wishbone?

- Successfully identified bug in IP RTL during integration !

LeWiz Communications Ethernet MAC Core2 10G/5G/2.5G/1G

https://github.com/lewiz-support/LMAC_CORE2

Thank You