SAND2018-2230C

Configuration of the Xilinx® Zynq®-7000 All Programmable SoC Memory Resources for Loosely-Coupled Lockstep Applications

PRESENTED BY

Ryan Kral

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Setting the Stage

What is Loosely-coupled lockstep and why is it important?

What is the Zynq-7000 SoC?

Overview of the Transaction Checker Architecture

Loosely-Coupled Lockstep

Multiple processors execute identical code

All peripheral accesses are compared between processors before data enters or leaves the system

Loosely-coupled lockstep increases information assurance in a design

Information assurance is necessary for high consequence applications to ensure that only correct data is sent to only the intended target

- Violations of this rule will result in undefined behavior and can result in catastrophic failures

Tightly vs. Loosely-Coupled Lockstep in Computer Architecture

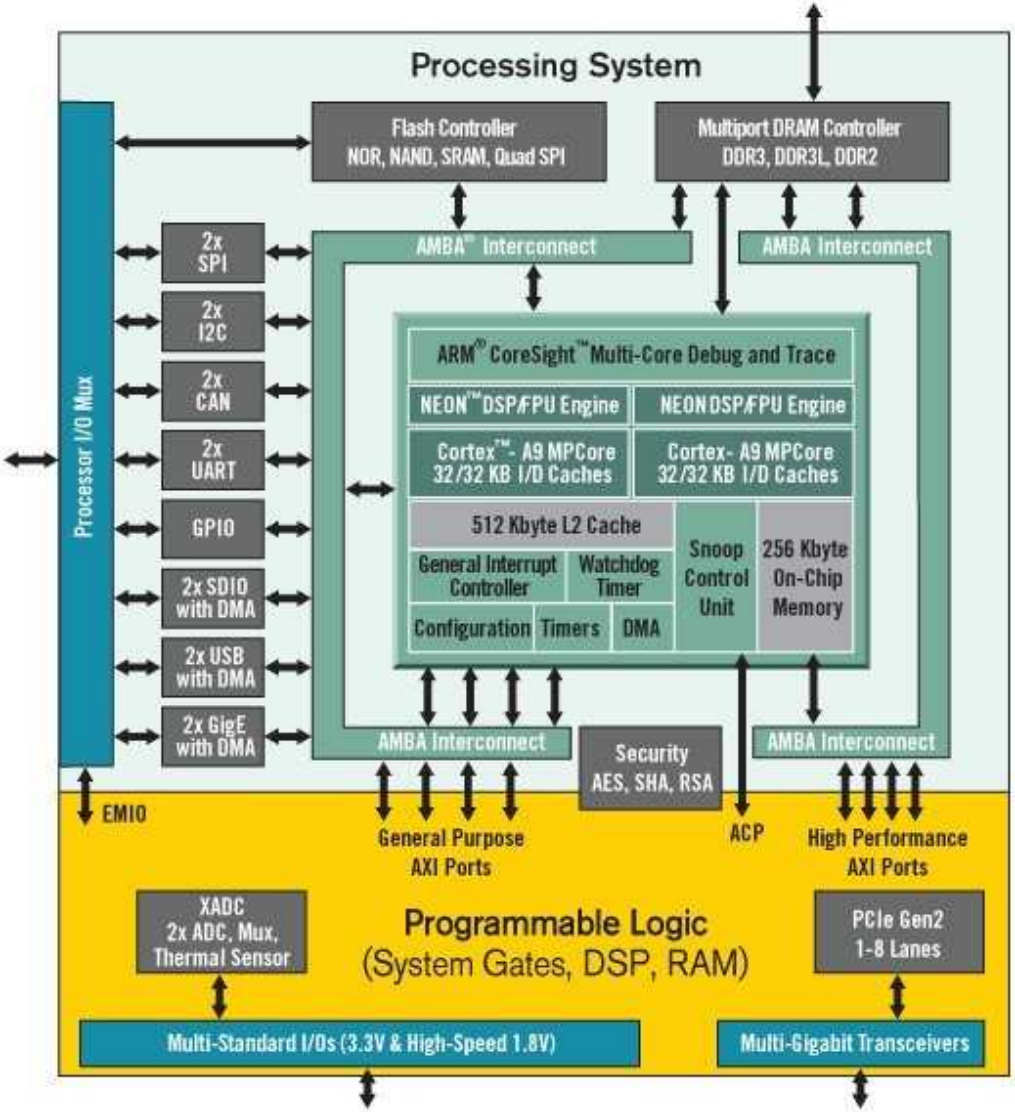
Tightly-Coupled

- Requires specialized hardware
- Each instruction is compared before execution
- Consistent behavior throughout

Loosely-Coupled

- Does not require specialized hardware
- Each peripheral access is compared before data enters or exits the system
- Consistency at key checkpoints

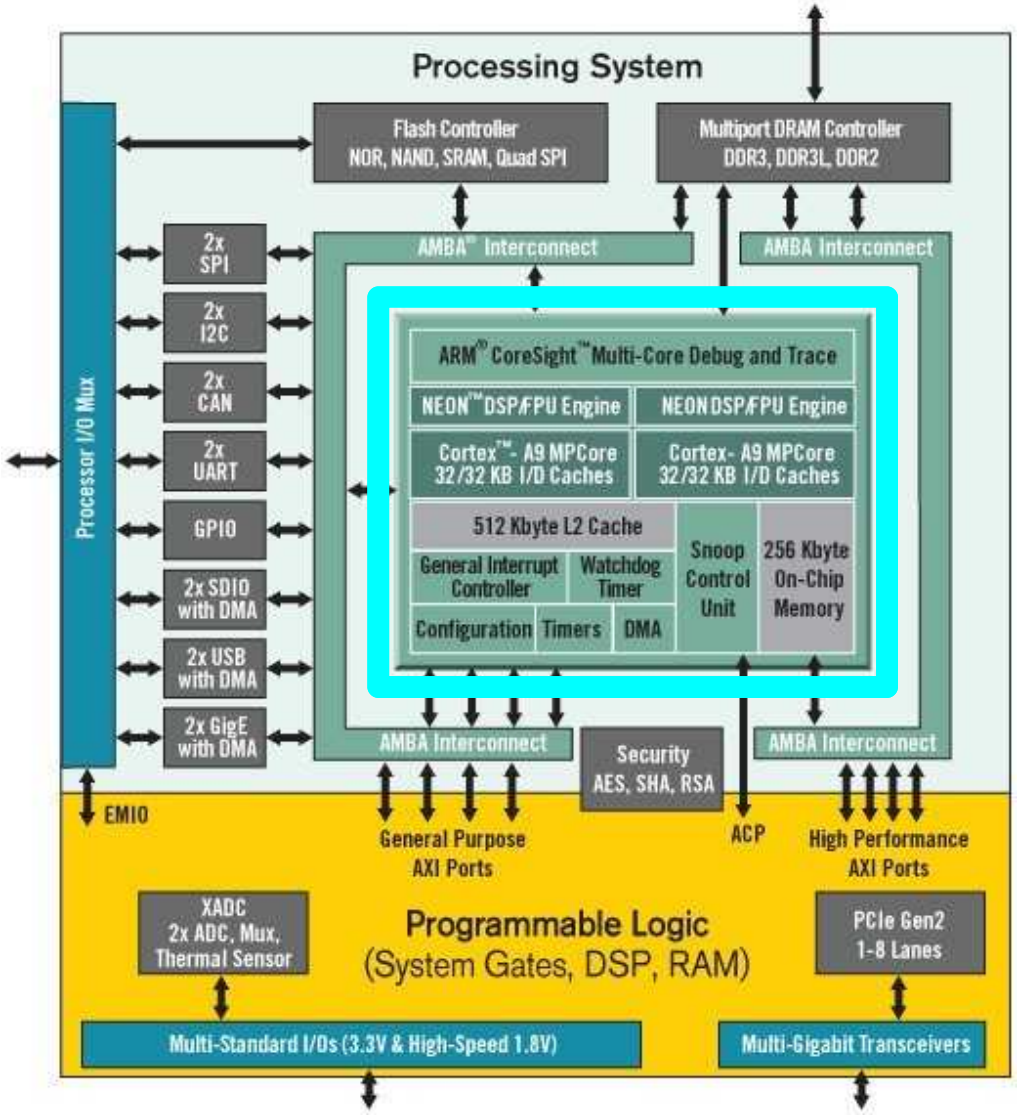
Xilinx Zynq-7000 SoC



Xilinx Zynq-7000 SoC

Dual-core ARM® Cortex™ A9 Processing System (PS)

- MMU for each core
- Snoop Control Unit
- Two on-chip memories

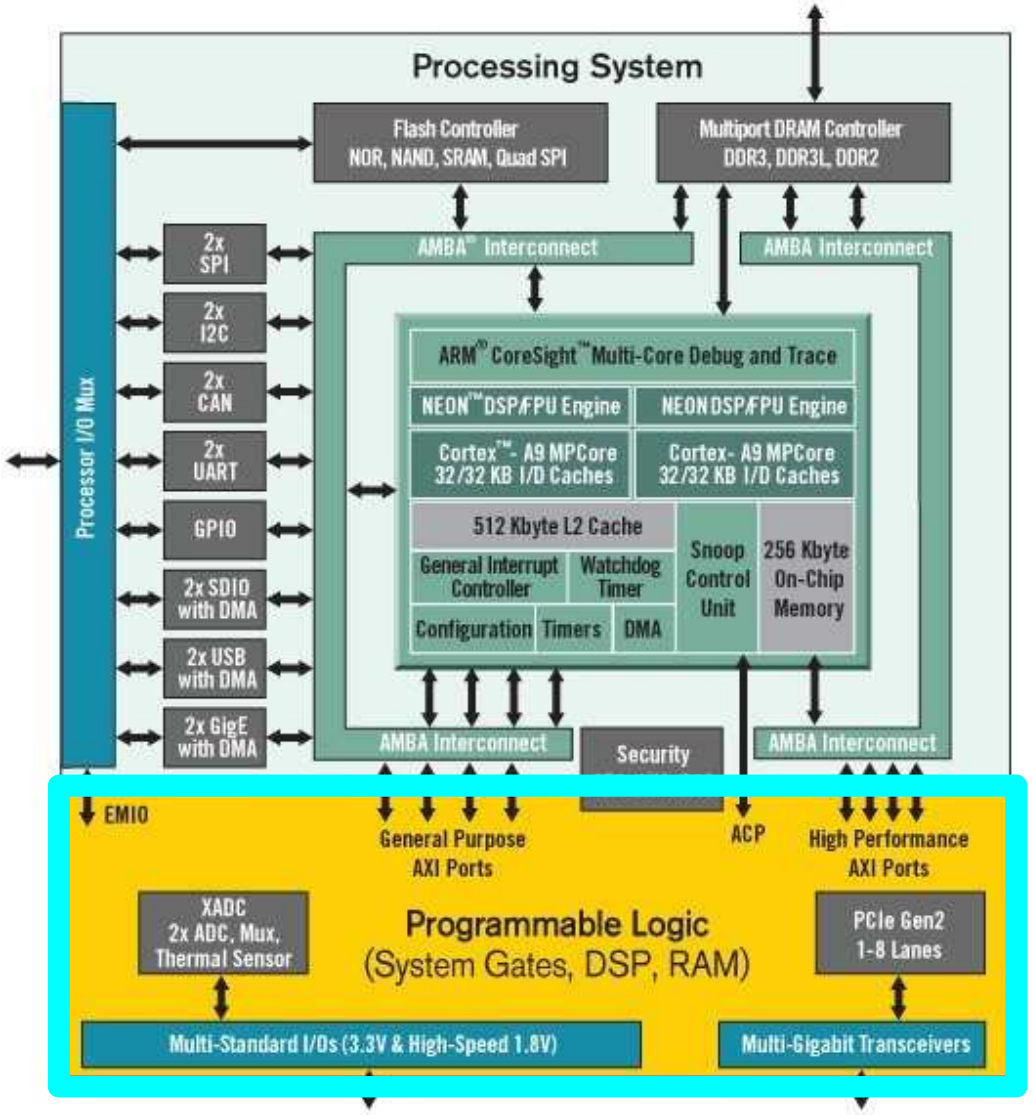


Xilinx Zynq-7000 SoC

Dual-core ARM® Cortex™ A9 Processing System (PS)

- MMU for each core
- Snoop Control Unit
- Two on-chip memories

Programmable Logic (PL)



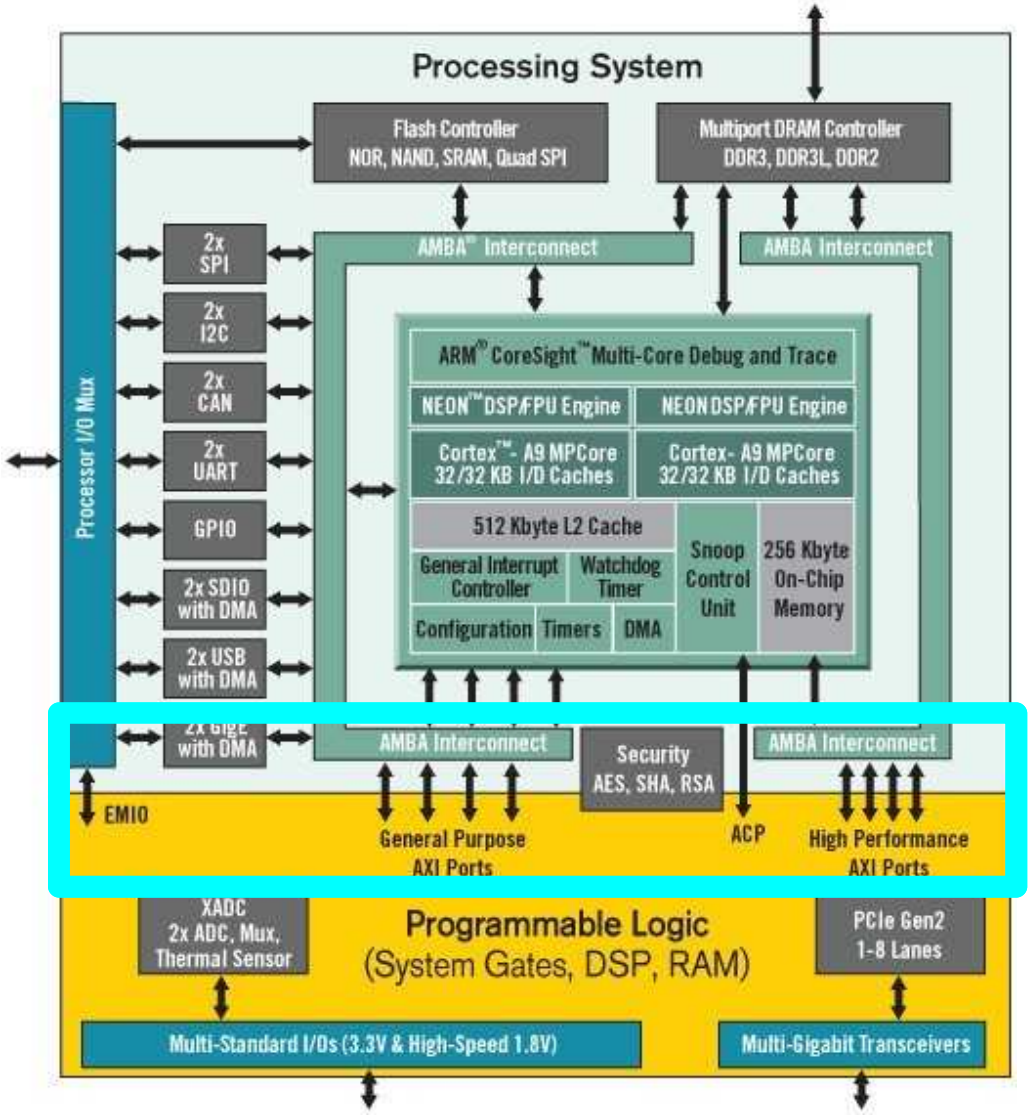
Xilinx Zynq-7000 SoC

Dual-core ARM® Cortex™ A9 Processing System (PS)

- MMU for each core
- Snoop Control Unit
- Two on-chip memories

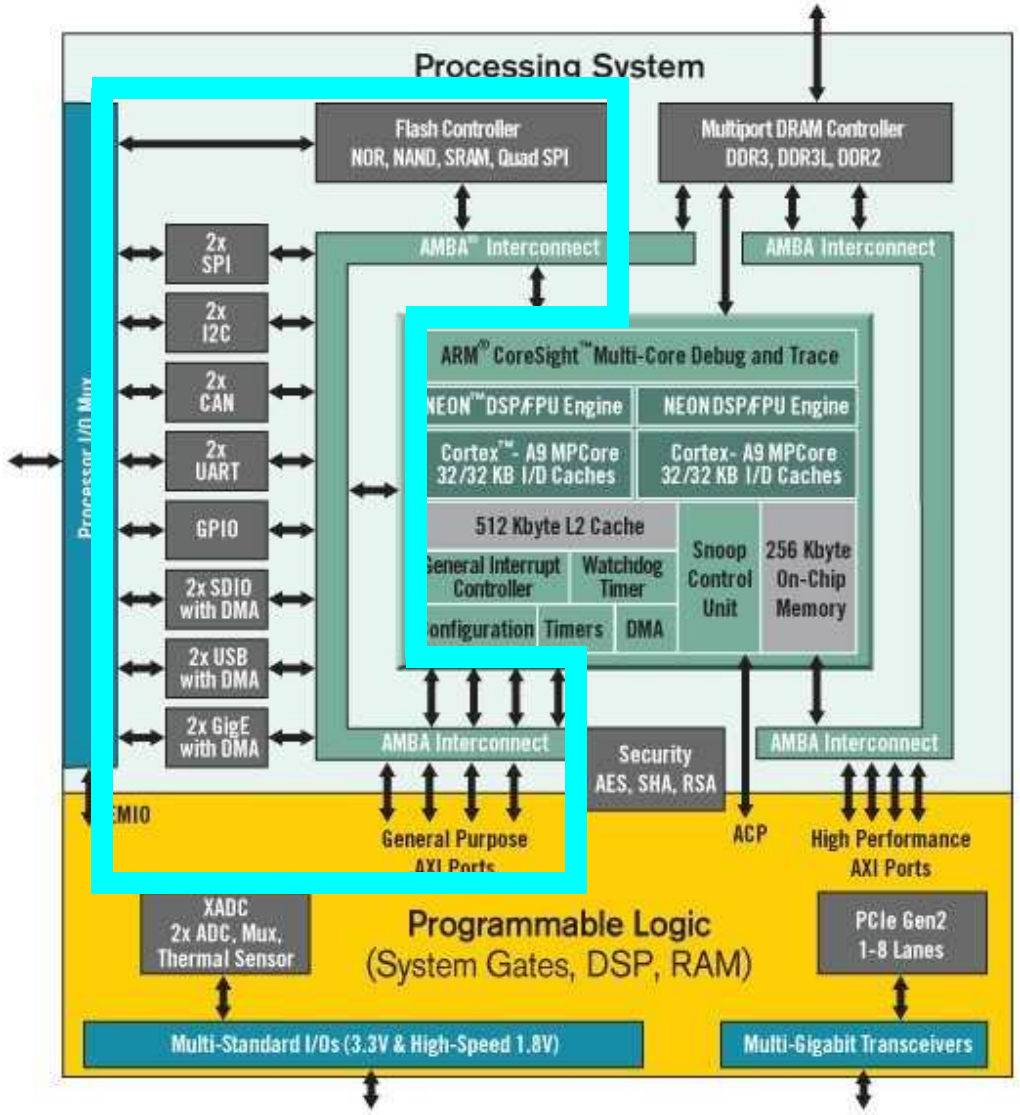
Programmable Logic (PL)

PS-PL AXI communication interfaces

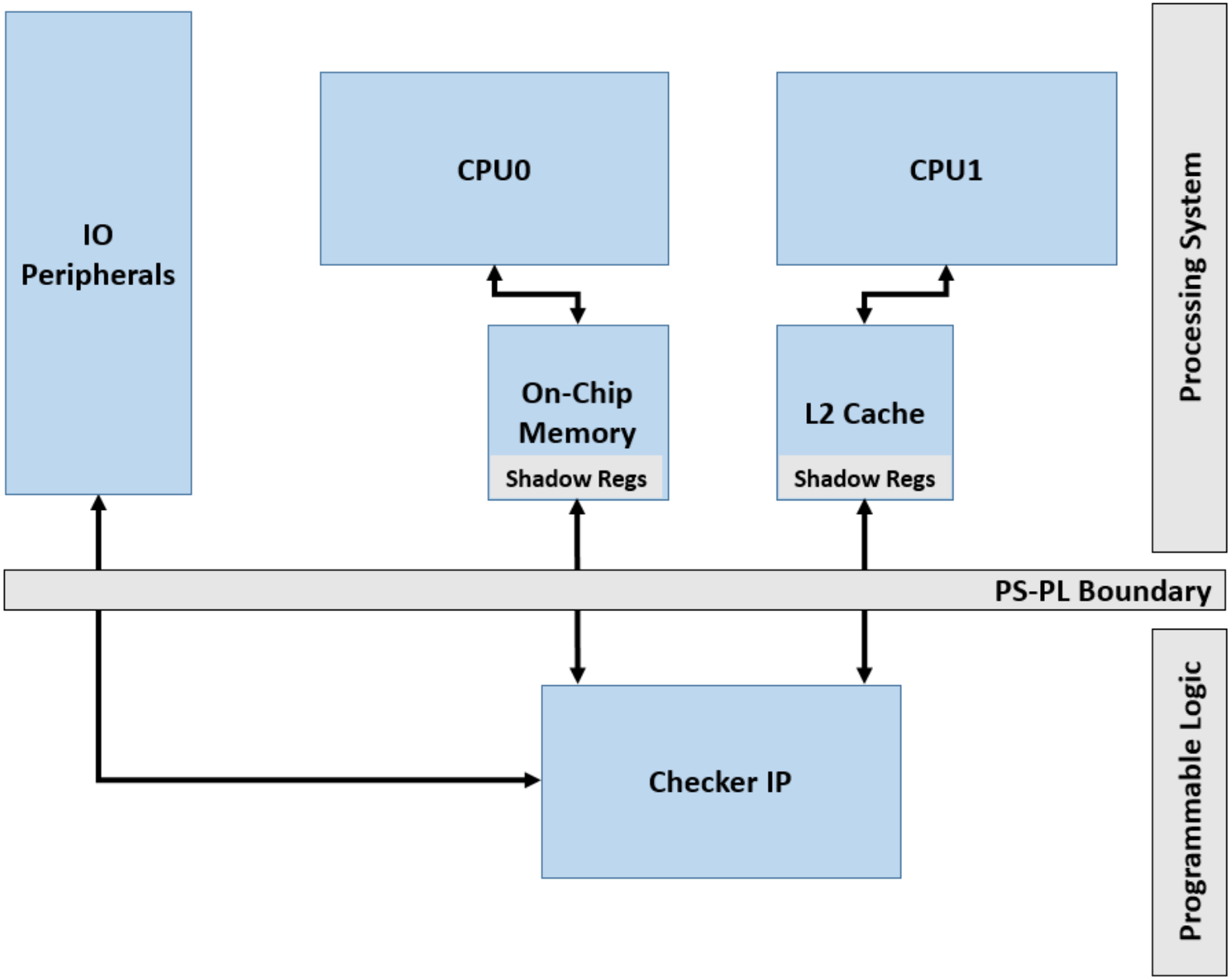


Xilinx Zynq-7000 SoC

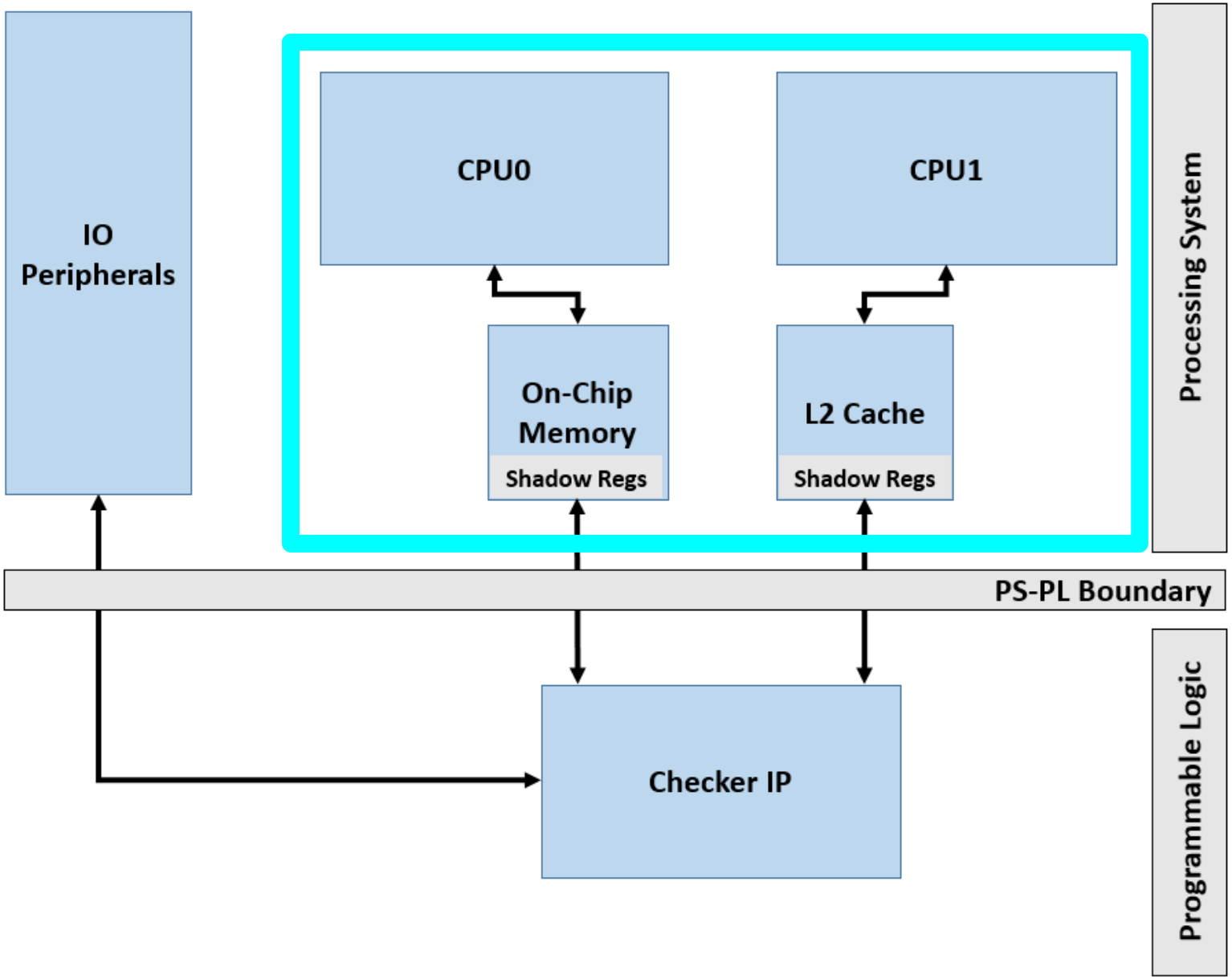
- Dual-core ARM® Cortex™ A9 Processing System (PS)
  - MMU for each core
  - Snoop Control Unit
  - Two on-chip memories
- Programmable Logic (PL)
- PS-PL AXI communication interfaces
- Input/Output Peripherals which are accessible from the CPUs or the PL



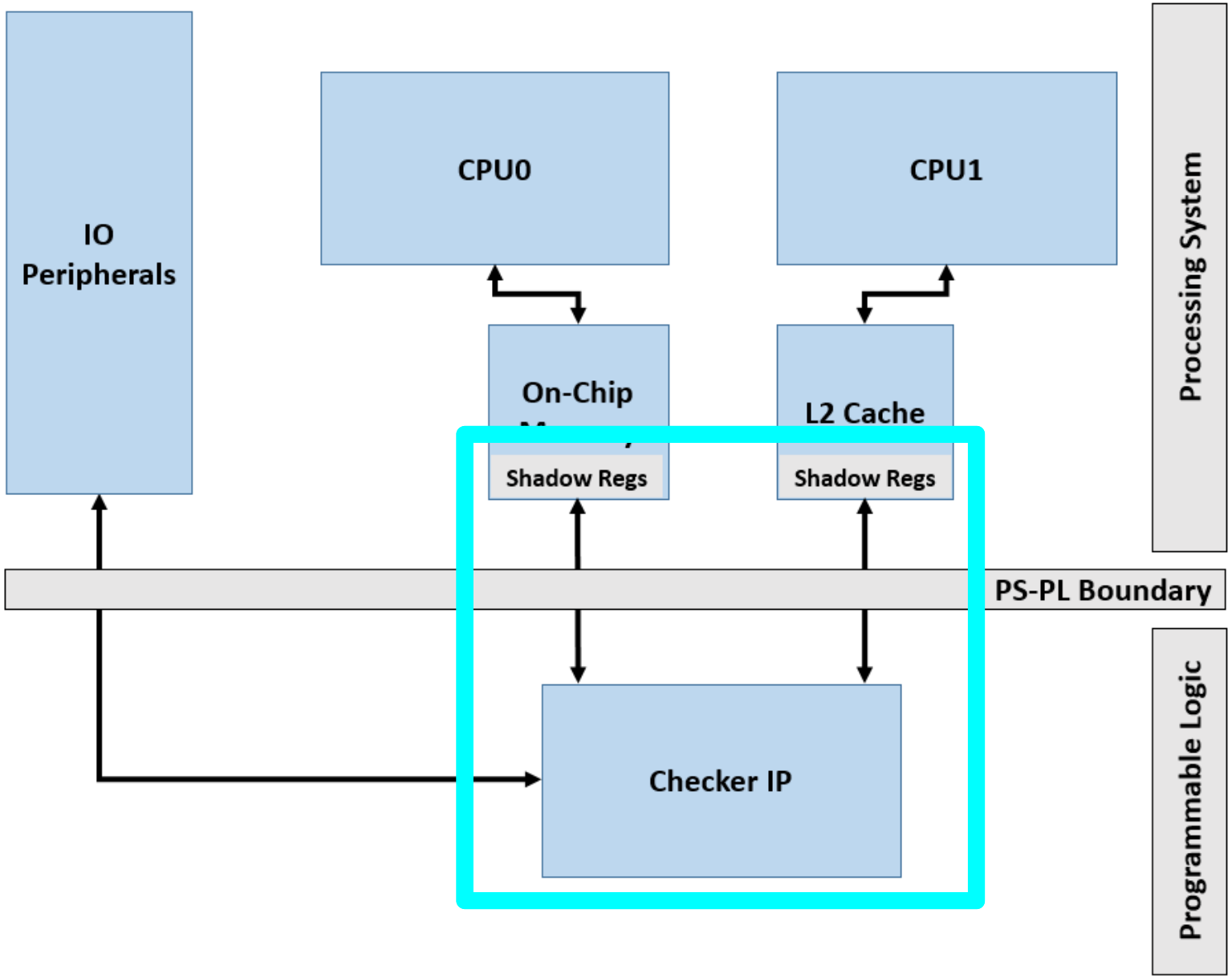
Transaction Checker Architecture Overview



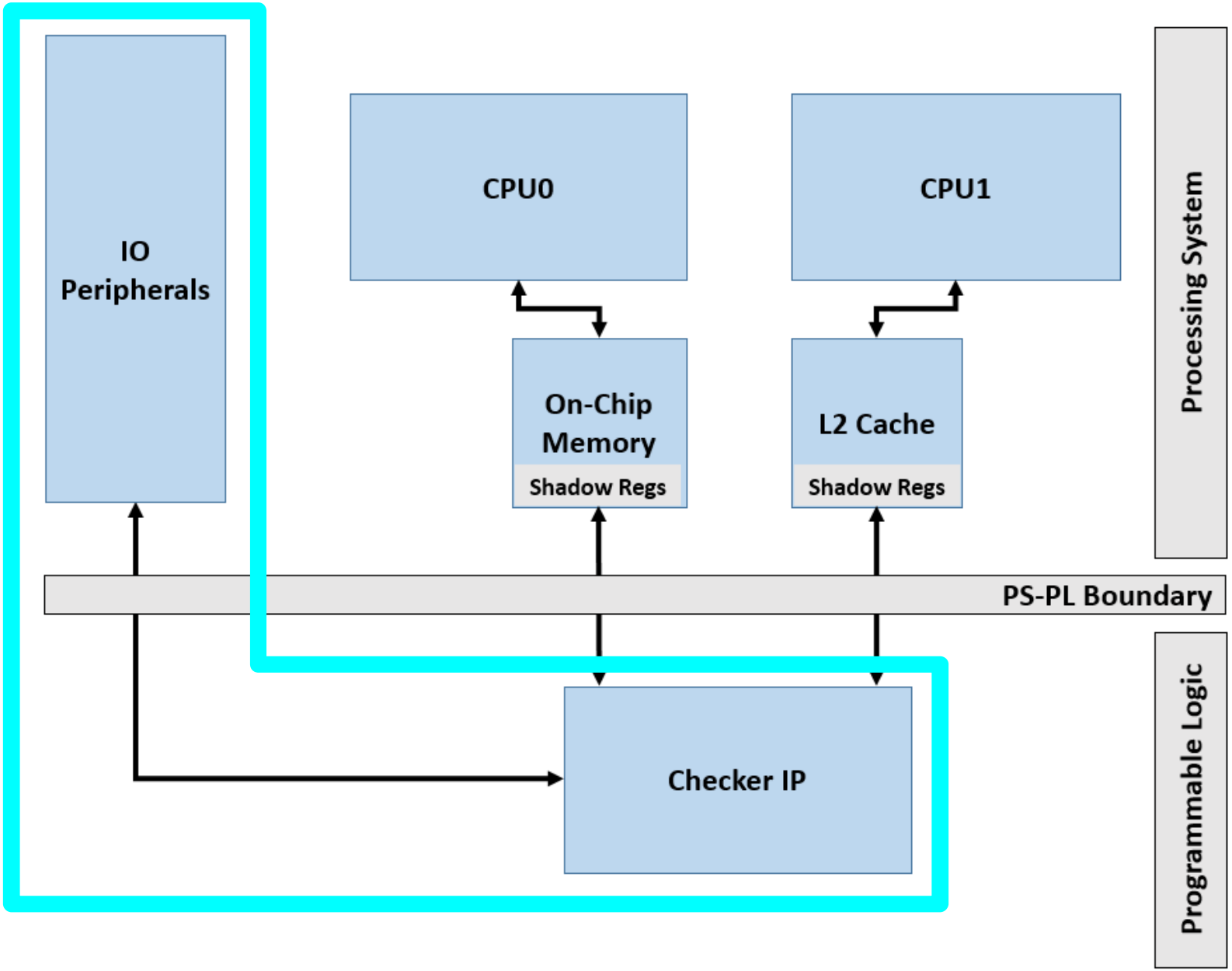
Transaction Checker Architecture Overview



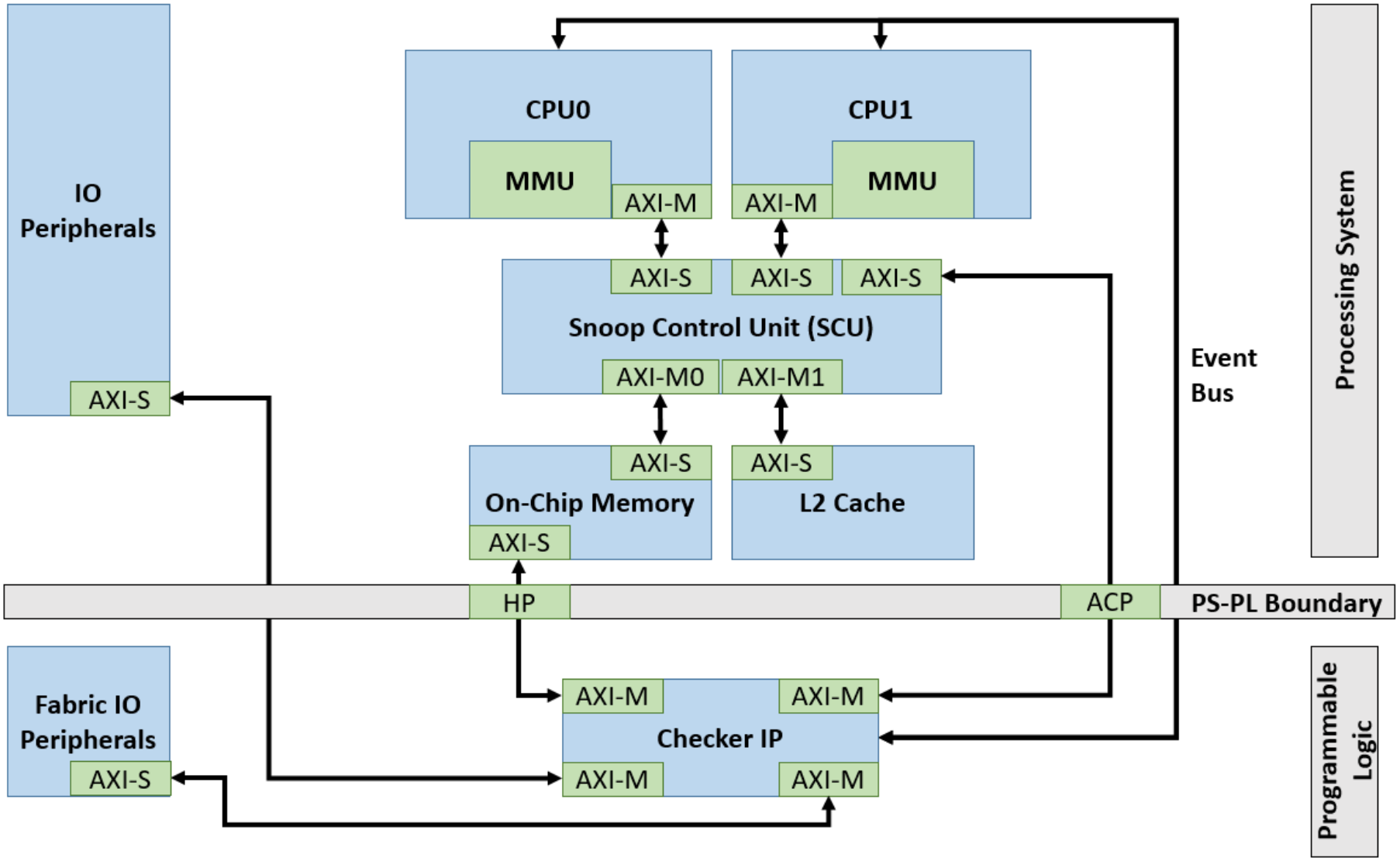
Transaction Checker Architecture Overview



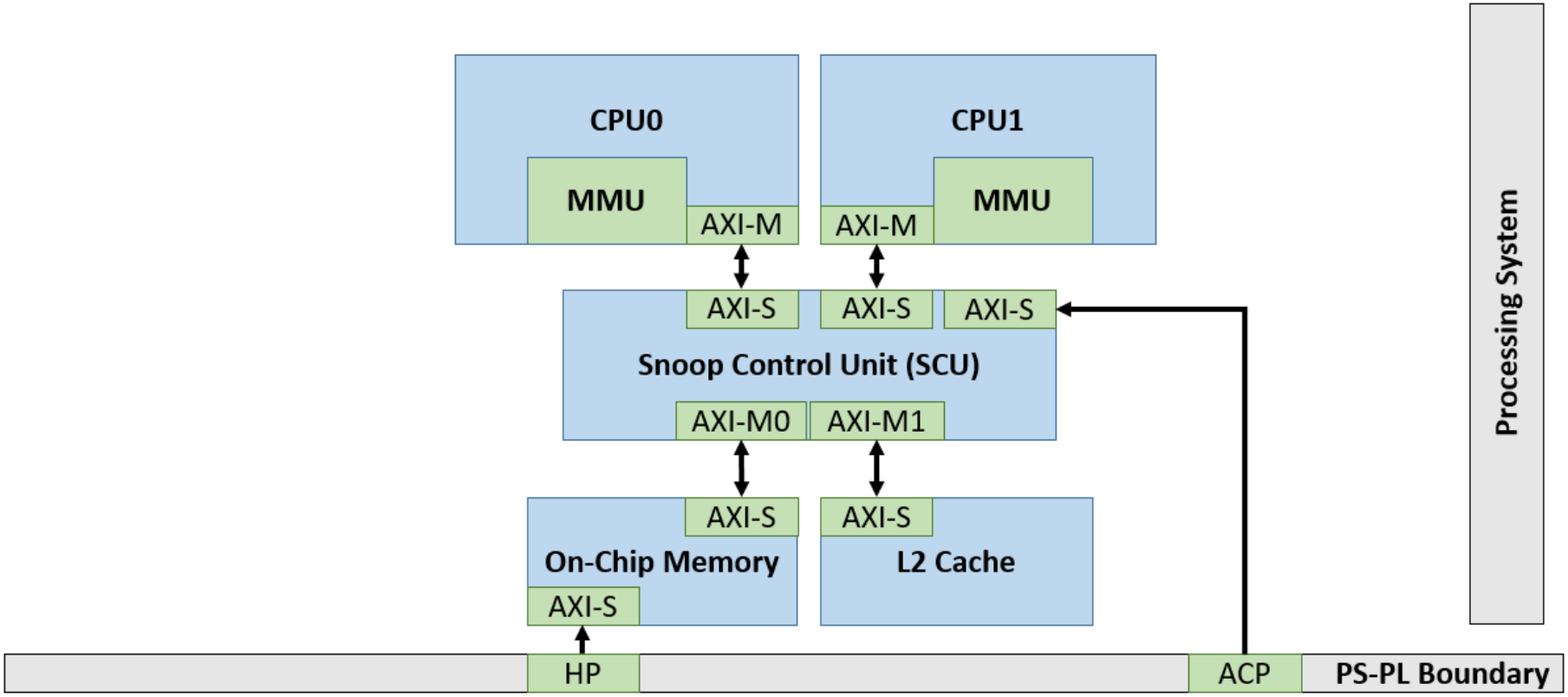
Transaction Checker Architecture Overview



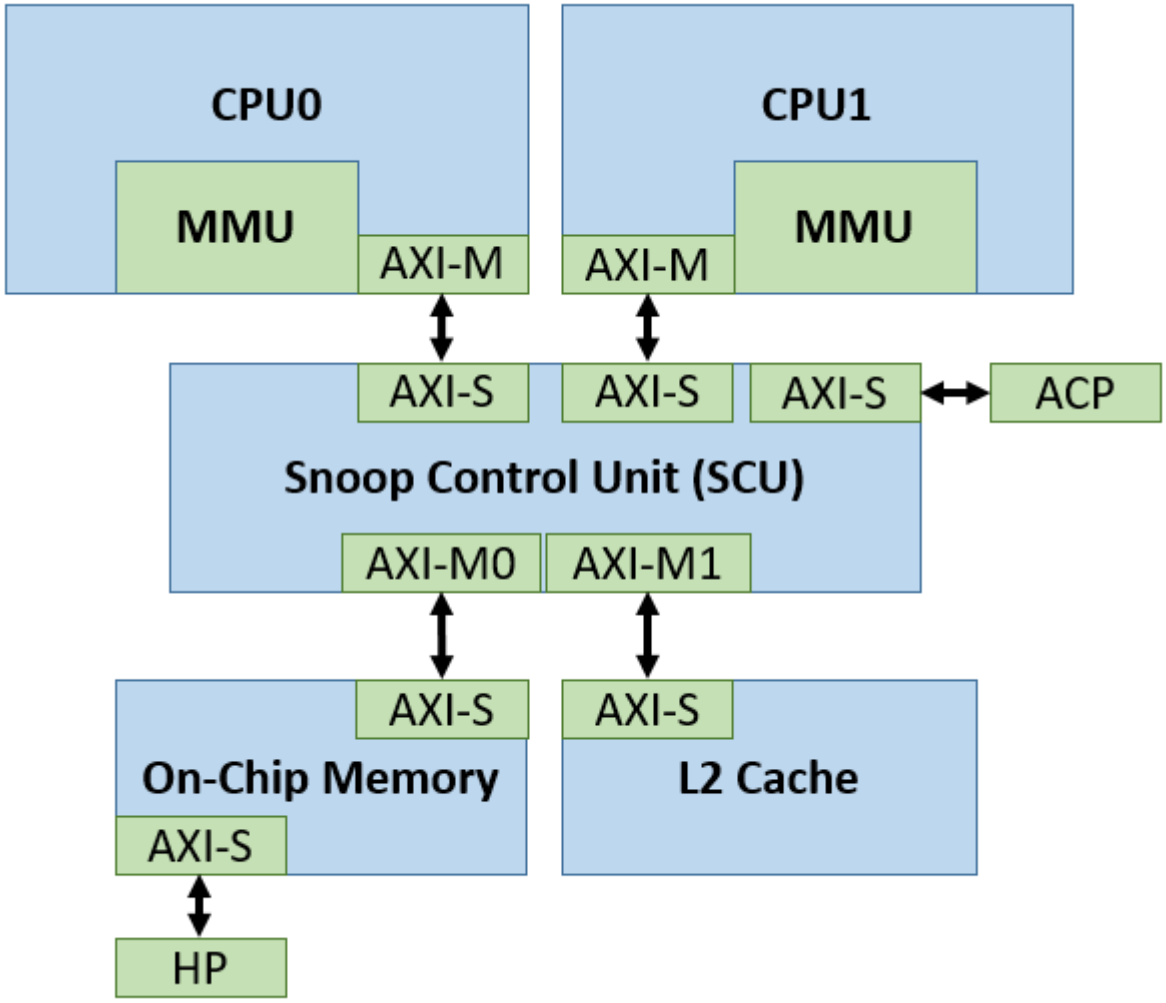
A Detailed Look



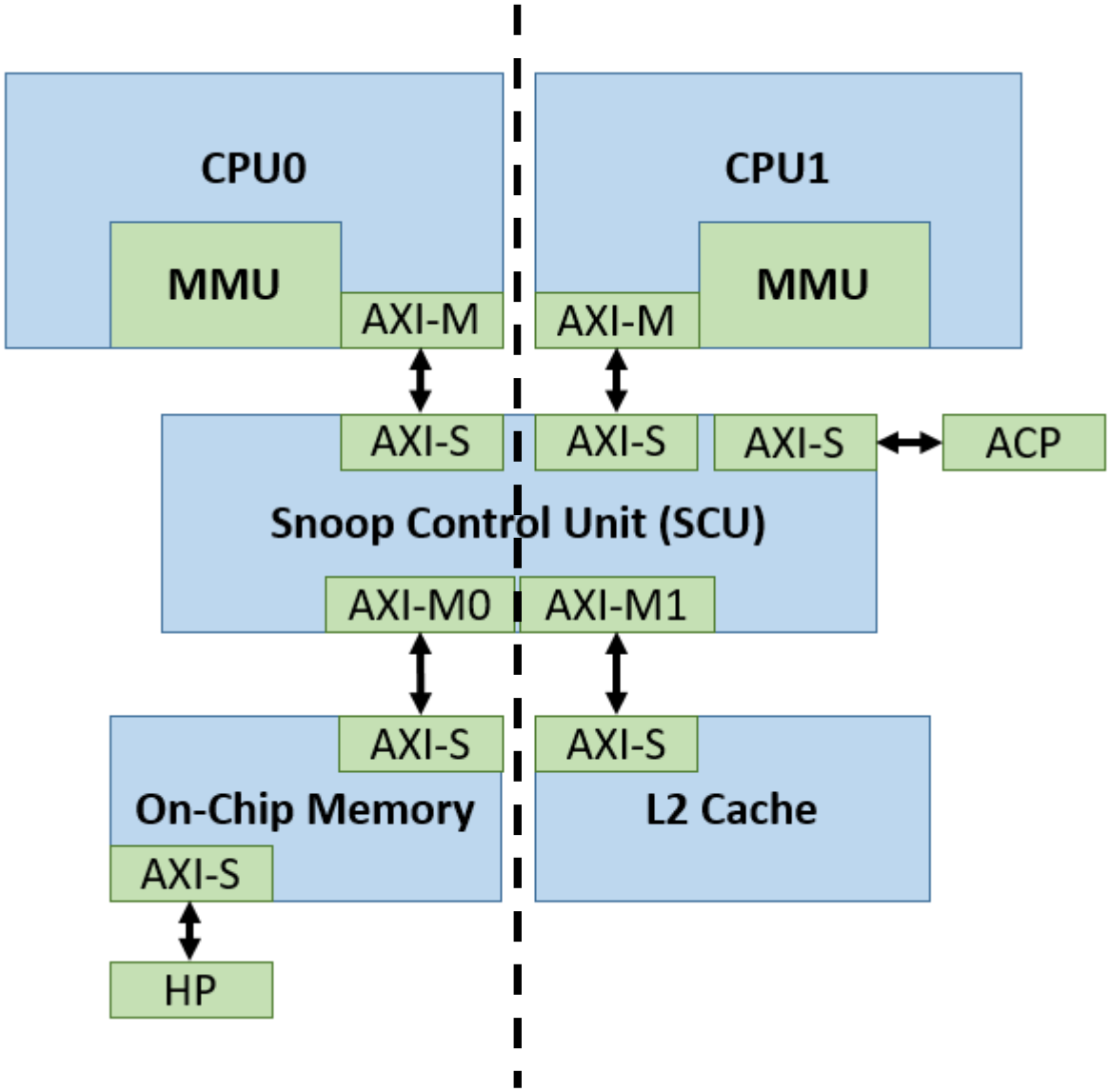
A Detailed Look



Focus on PS Memory Resource Configurations



Focus on PS Memory Resource Configurations



On-Chip Memory (OCM)

Contains 256 KB of RAM

- Used to store instructions and data for CPU 0

The size of OCM determines the maximum size of an application which can be run in loosely-coupled lockstep

All 256 KB of RAM must be configured to reside at contiguous addresses

L2 Cache

The two CPUs share a unified 512 KB, eight way, set associative cache

- Used to store instructions and data for CPU 1
- CPU 0's access is removed through MMU and SCU settings

The L2 cache is not directly addressable

- It is loaded by placing the application in cacheable memory which can then be preloaded into the cache

The L2 cache controller allows lockdown by way which is used to prevent the loaded entries from being evicted

Memory Management Unit (MMU)

Each CPU has its own MMU and pointer to an MMU table

MMU tables are used to store virtual to physical address translations and to configure access permissions for various memory regions

- Two CPUs can execute from identical virtual addresses which translate to different physical addresses

MMU table entries can be cached into Translation Lookaside Buffers (TLBs)

Each CPU is configured with only the TLB entries it needs to access its own on-chip memory

- All other memory region accesses will result in table walk errors as no full MMU tables are stored

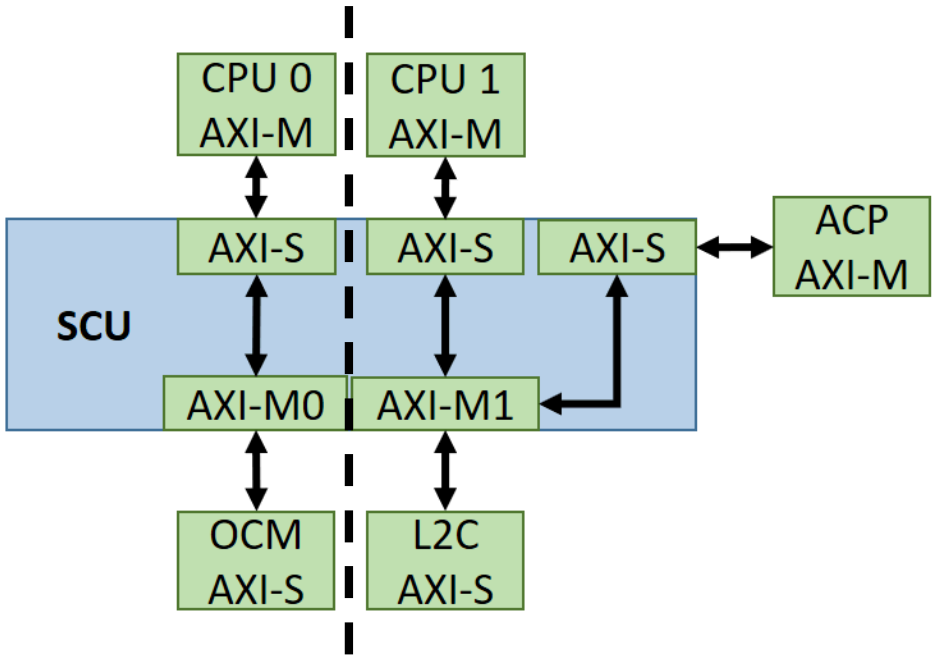
Snoop Control Unit (SCU)

Cache coherency through the SCU is disabled to separate the L1 caches from one another

The SCU performs address filtering by forwarding CPU accesses to either OCM or L2C based on physical addresses

- Combined with the MMU settings, this provides separation of memories

The ACP can access L2C



Boot Flow

Broken down into three stages:

- First Stage Boot Loader (FSBL)
- Second Stage Boot Loader (SSBL)
- Application Startup

Boot Flow - FSBL

Executes on CPU 0 from OCM

Remaps OCM to contiguous low addresses

Loads the FPGA bitstream from QSPI

Copies the Second Stage Boot Loader from QSPI to DDR

Boot Flow

	CPU 0	CPU 1
FSBL (OCM)	Remap OCM to low addresses	
	Load bitstream from QSPI	
	Copy SSBL from QSPI to DDR	
SSBL (DDR)		
App Startup (OCM/L2 Cache)		

Boot Flow - SSBL

Executes on CPU 0 from DDR

Copies the Application from QSPI to OCM

Copies the Application into a cacheable region of DDR

Loads and locks the Application into the L2 cache

Resets CPU 1

Boot Flow

	CPU 0	CPU 1
FSBL (OCM)	Remap OCM to low addresses	
	Load bitstream from QSPI	
	Copy SSBL from QSPI to DDR	
SSBL (DDR)	Copy application from QSPI to OCM	
	Copy application into DDR	
	Load and lock application into the L2 cache	
	Reset CPU 1	
App Startup (OCM/L2 Cache)		

Boot Flow – App Startup CPU 1

Executes from OCM following the reset

Configures the SCU for no cache coherency

Loads and locks CPU 1's MMU TLBs with the necessary virtual to physical address mappings

Enables the MMU

Switches Execution to the L2 cache

Resets CPU 0

Finalizes CPU 1's TLBs to remove OCM access

Jumps to the Application

Boot Flow

	CPU 0	CPU 1
FSBL (OCM)	Remap OCM to low addresses	
	Load bitstream from QSPI	
	Copy SSBL from QSPI to DDR	
SSBL (DDR)	Copy application from QSPI to OCM	
	Copy application into DDR	
	Load and lock application into the L2 cache	
	Reset CPU 1	
App Startup (OCM/L2 Cache)		Configure the SCU
		Load and lock CPU 1 MMU TLBs
		Enable MMU
		Switch execution to the L2 cache
		Reset CPU 0
		Finalize CPU 1 MMU TLBs
		Jump to application

Boot Flow – App Startup CPU 0

Executes from OCM following the reset

Loads and locks CPU 0's MMU TLBs with the necessary virtual to physical address mappings

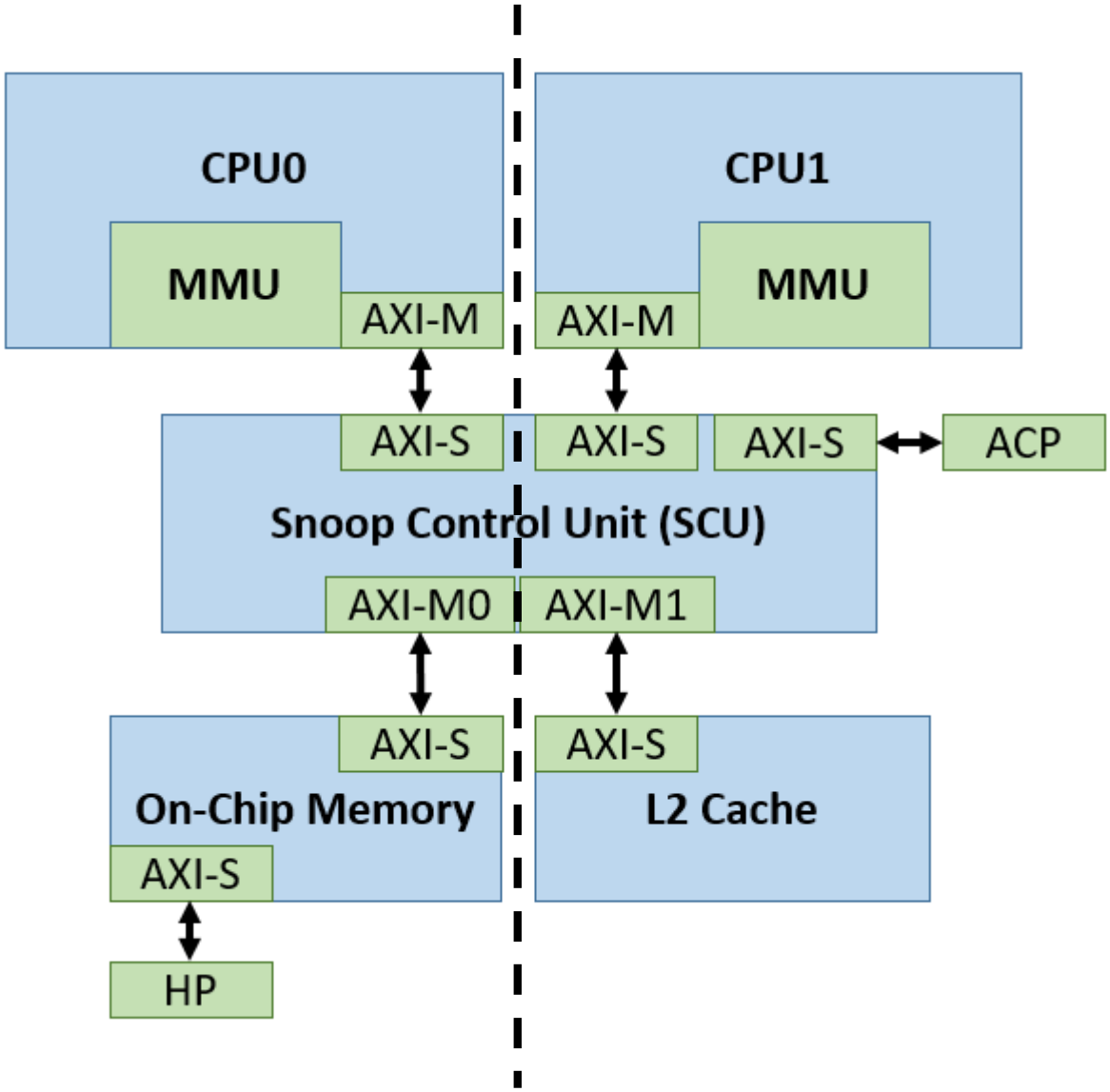
Enables the MMU

Jumps to the Application

Boot Flow

	CPU 0	CPU 1
FSBL (OCM)	Remap OCM to low addresses	
	Load bitstream from QSPI	
	Copy SSBL from QSPI to DDR	
SSBL (DDR)	Copy application from QSPI to OCM	
	Copy application into DDR	
	Load and lock application into the L2 cache	
	Reset CPU 1	
App Startup (OCM/L2 Cache)		Configure the SCU
		Load and lock CPU 1 MMU TLBs
		Enable MMU
		Switch execution to the L2 cache
		Reset CPU 0
	Load and lock CPU 0 MMU TLBs	Finalize CPU 1 MMU TLBs
	Enable MMU	Jump to application
	Jump to application	

PS Memory Resources Configured



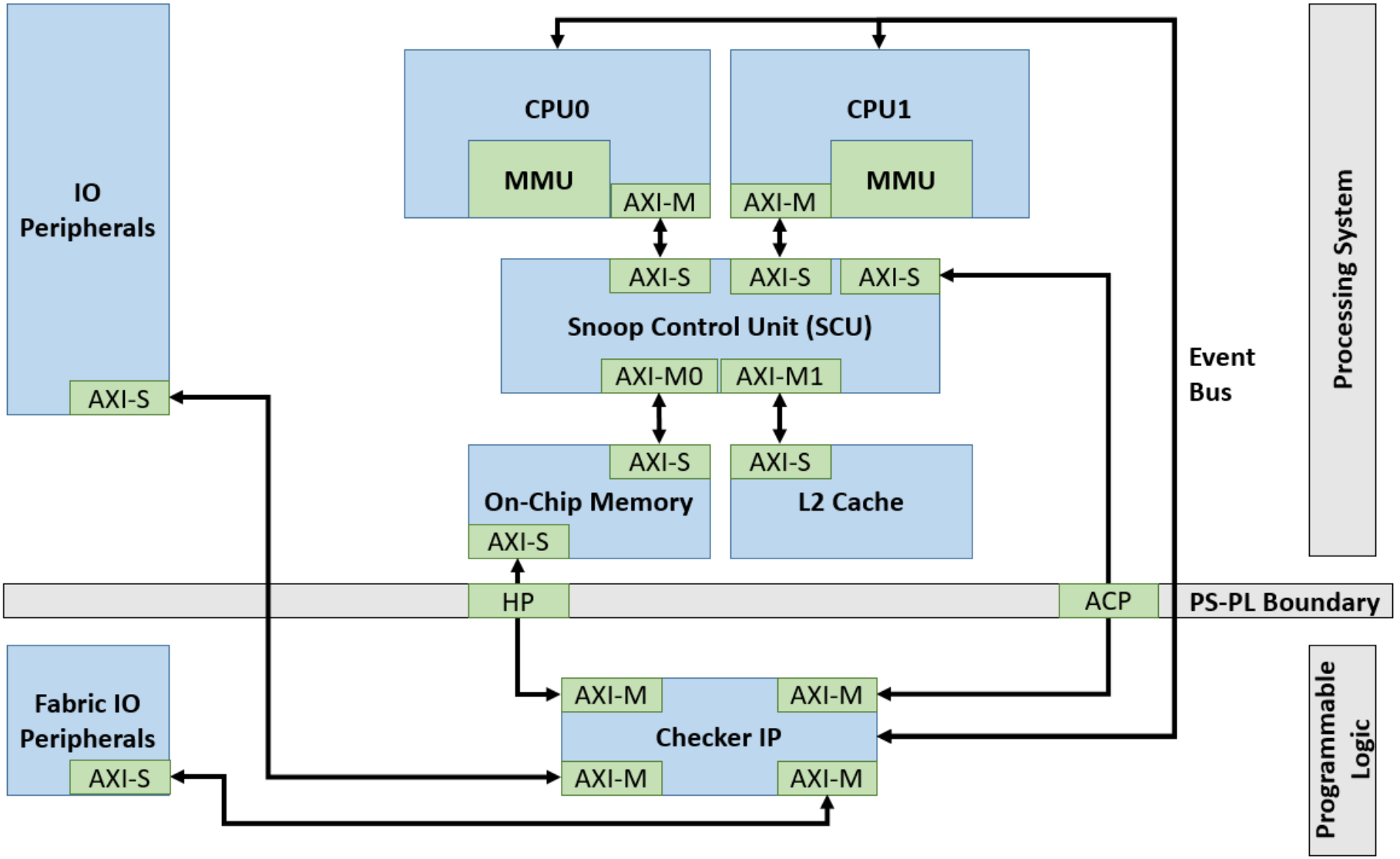
Summary

Using the Transaction Checker Architecture on the Zynq-7000 SoC will increase information assurance in a system

The Processing System memory resources can be configured to ensure separation between the memories used for CPU 0 and CPU 1

The boot process enables the Zynq to operate in a loosely-coupled lockstep manner while executing application software

Questions?


IO Peripherals
AXI-S
CPU0
MMU
AXI-M
CPU1
MMU
AXI-M
AXI-S
AXI-S
AXI-S
Snoop Control Unit (SCU)
AXI-M0
AXI-M1
Event Bus
Processing System
AXI-S
On-Chip Memory
AXI-S
AXI-S
L2 Cache
HP
ACP
PS-PL Boundary
Fabric IO Peripherals
AXI-S
AXI-M
AXI-M
Checker IP
AXI-M
AXI-M
Programmable Logic

Backup

Peripheral Transactions

Shadow Registers

- Dedicated, 80-byte memory banks for each CPU
- Control, address, data, and status for a peripheral access are specified in these memories

Checker IP Functions

1. Compare the dedicated shadow registers between the CPUs
2. Perform the peripheral accesses described by the CPUs
3. Interact with a Fabric Interrupt Controller (FIC)
4. Detect and report error conditions

Transaction Checker Architecture