FUNCTIONAL VERIFICATION OF NEXT GENERATION IC'S WITH NEXT GENERATION TOOLS

Applying Palladium XP Simulation Acceleration to an Existing Specman Testbench Framework

PMC-Sierra – Jeffrey Huang (Verification Engineer)
PMC-Sierra – Horace Chan (Verification Lead)
Cadence Design Systems – Dave Allen (Logic Verification Specialist)

OVERVIEW

- The Need for Acceleration
- The Tool: Palladium XP
- Infrastructure: The Specman Testbench
- Modifying the Testbench to support Palladium XP
  - Architectural Decisions
- Performance Achievement
- Optimizations and Future Work

SPECIAL THANKS

- Cadence
  - Chris Dietrich and the Palladium XP Solutions Deployment Team
  - Jacek Wuwer
- PMC-Sierra
  - Ted Wilson

THE NEED FOR ACCELERATION

WHO IS PMC-SIERRA?

PMC (NASDAQ:PMCS) is the semiconductor innovator transforming networks that connect, move and store digital content.

Building on a track record of technology leadership, the company is driving innovation across storage, optical and mobile networks.

PMC's highly integrated solutions increase performance and enable next generation services to accelerate the network transformation.

PMC CHIPS TODAY: BLEEDING EDGE DESIGN

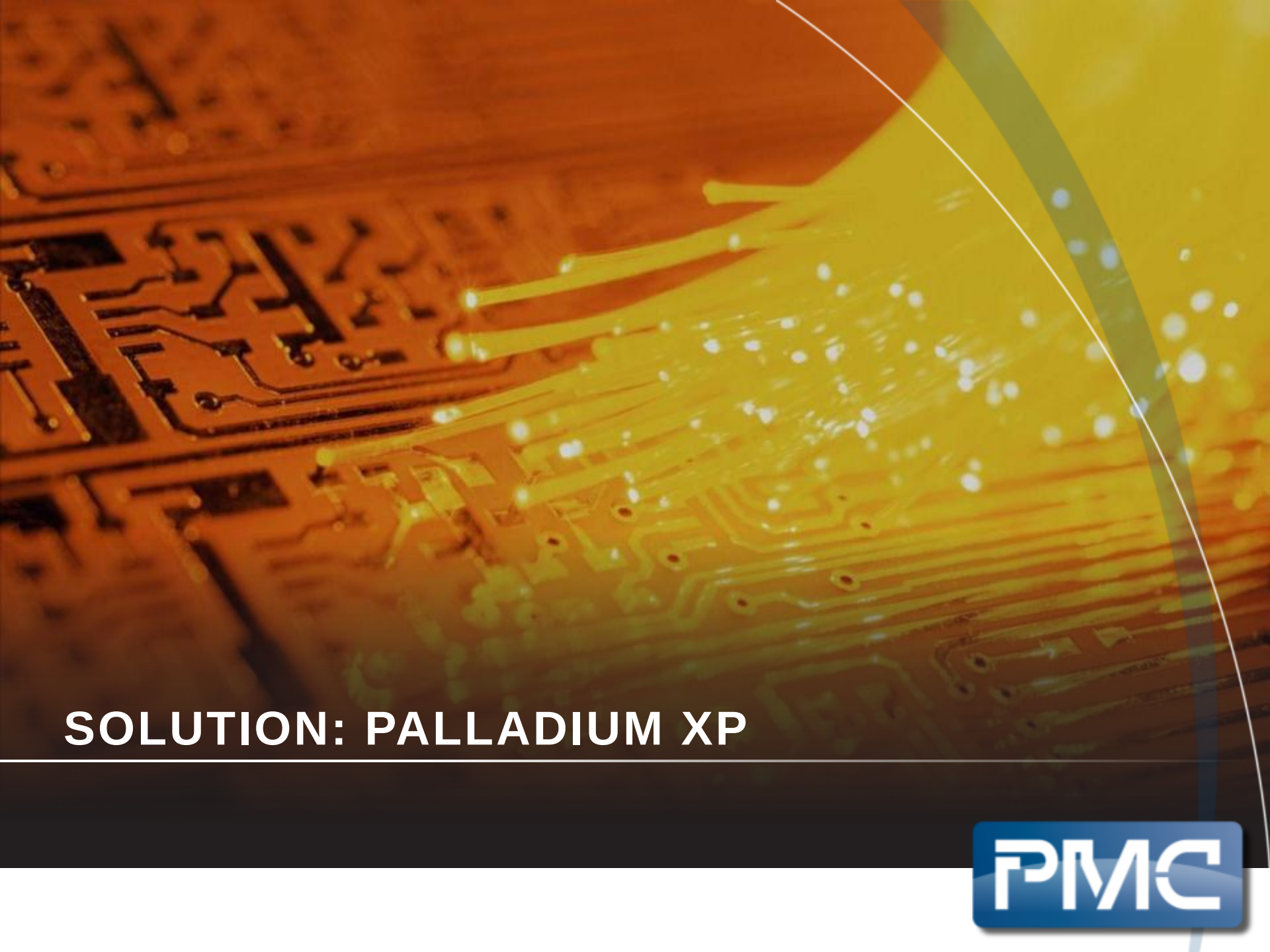
Application: High-Speed, Complex, Multi-protocol Routing

Massive, 100+ Million Gate SoC design.

- Architecture: 10 Subsystems
  - Some of the subsystems could be classified as stand-alone chips!
- 18-month Schedule

CHALLENGES WITH BLEEDING EDGE DESIGN

- Simulation Times :
  - It can take hours to send a single frame.
  - Running the full regression suite can take more than a week.
- Because of these run-times:
  - Interactive debug is difficult.
  - It can take a long time to reproduce bugs.
  - The team has to create simpler simulations to get reasonable simulation times
- Longer simulation times means greater time lag between the release of RTL and subsequent verification before the next RTL release.
- It's challenging to complete a comprehensive verification plan while meeting time-to-market demands.



SOLUTION: PALLADIUM XP

PALLADIUM XP™ VERIFICATION COMPUTING PLATFORM

A single system that can be used for Simulation Acceleration and In-Circuit Emulation.

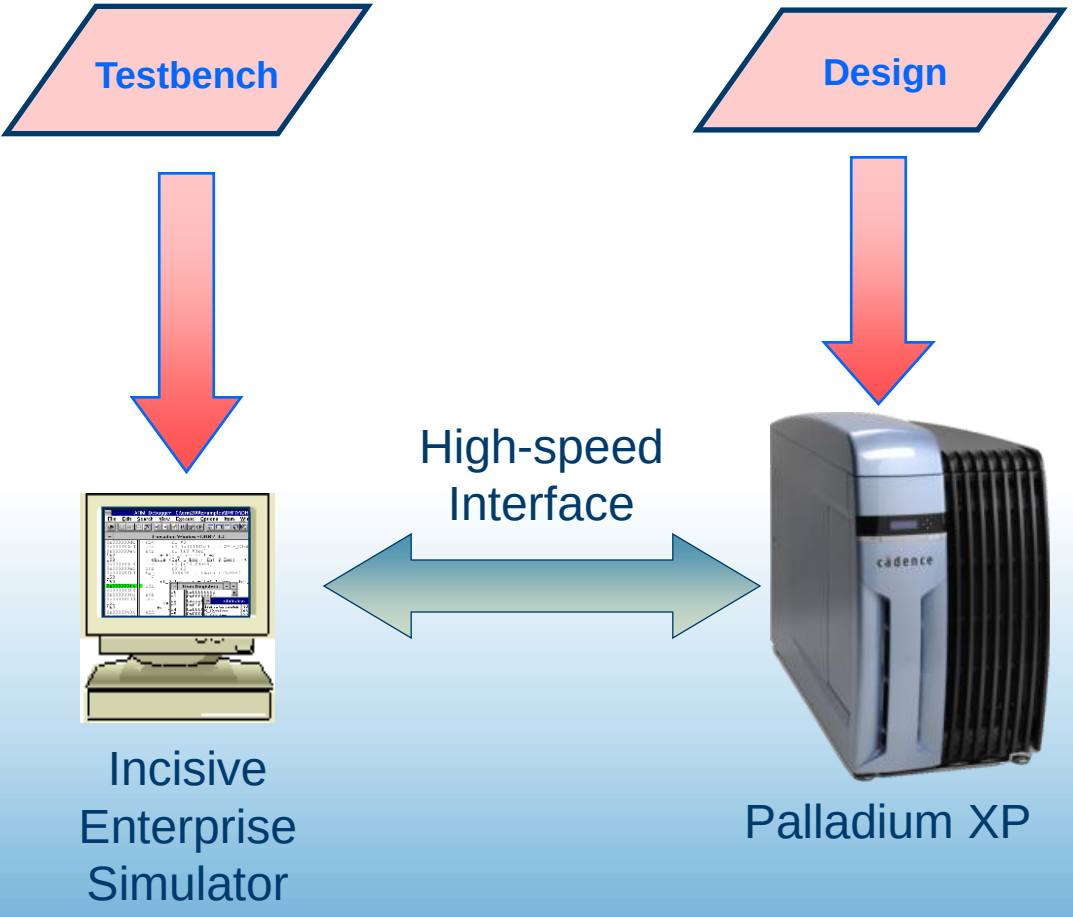
PMC's Installation:

- Up to 128 million gates capacity
- Up to 32 Gigabytes of on-board memory
- Up to 4,608 I/O pins

The system can be used simultaneously by up to 64 users, including 16 simulation acceleration sessions.

SIMULATION ACCELERATION

Dramatically increases throughput while maintaining simulation-like environment



- **Up to 4 MHz operation**
- **System-level debugging - with embedded software**
- **Control via the Incisive Enterprise Simulator (IES)**
- **Ideal for Block, Chip and early system-level validation**

TESTBENCH DESIGN

- Requirements:
  - Easy to move between simulation and simulation acceleration
  - Complex testbench resources available
    - Randomization
    - Scoreboarding
- Solution:
  - Unified Verification Methodology (UVM) structure
  - Specman *e* testbench
  - SystemVerilog DUT

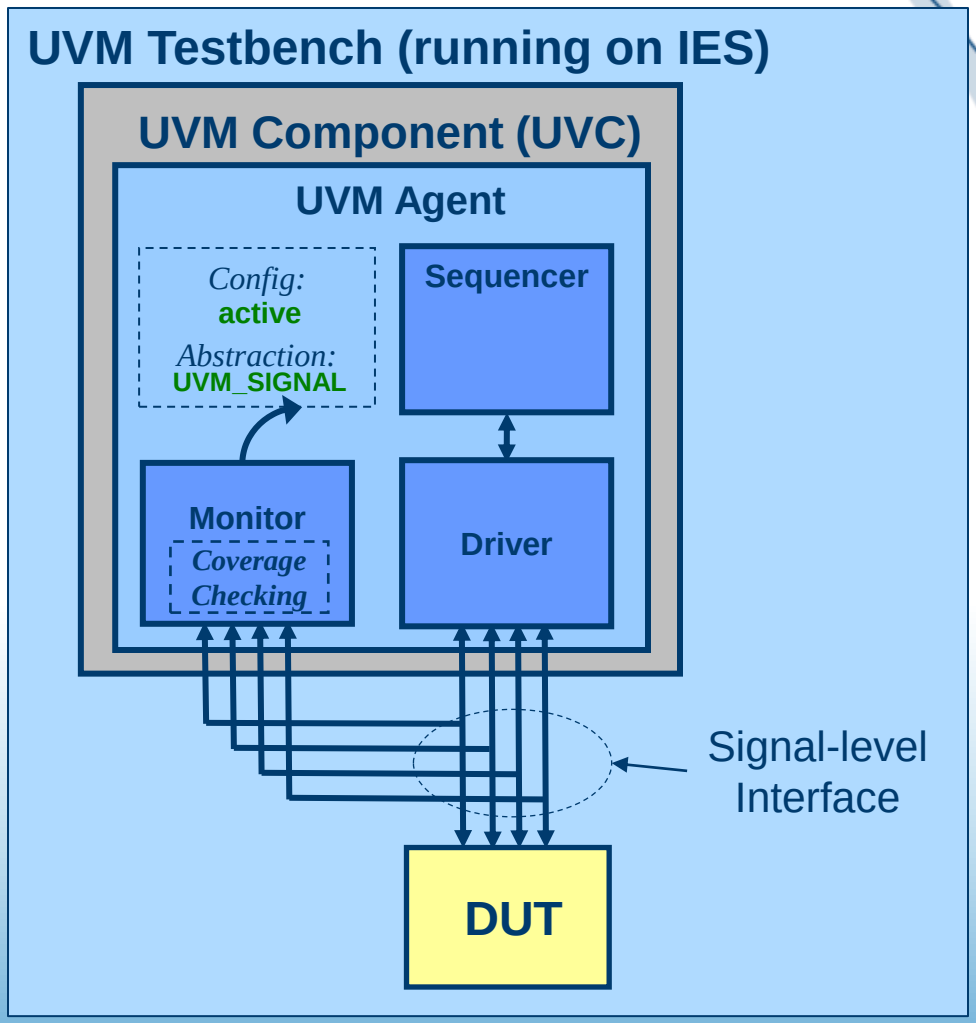
BASIC UVM STRUCTURE

The *UVM Agent* is a fundamental building block of UVM. It has:

- A *Sequencer* to generate stimulus for the interface.
- A *Driver* to convert each sequence to the signals needed by the interface.
- A *Monitor* to check the signal-level protocol and coverage.

One or more *UVM Agents* are instantiated in a *UVM Component (UVC)*.

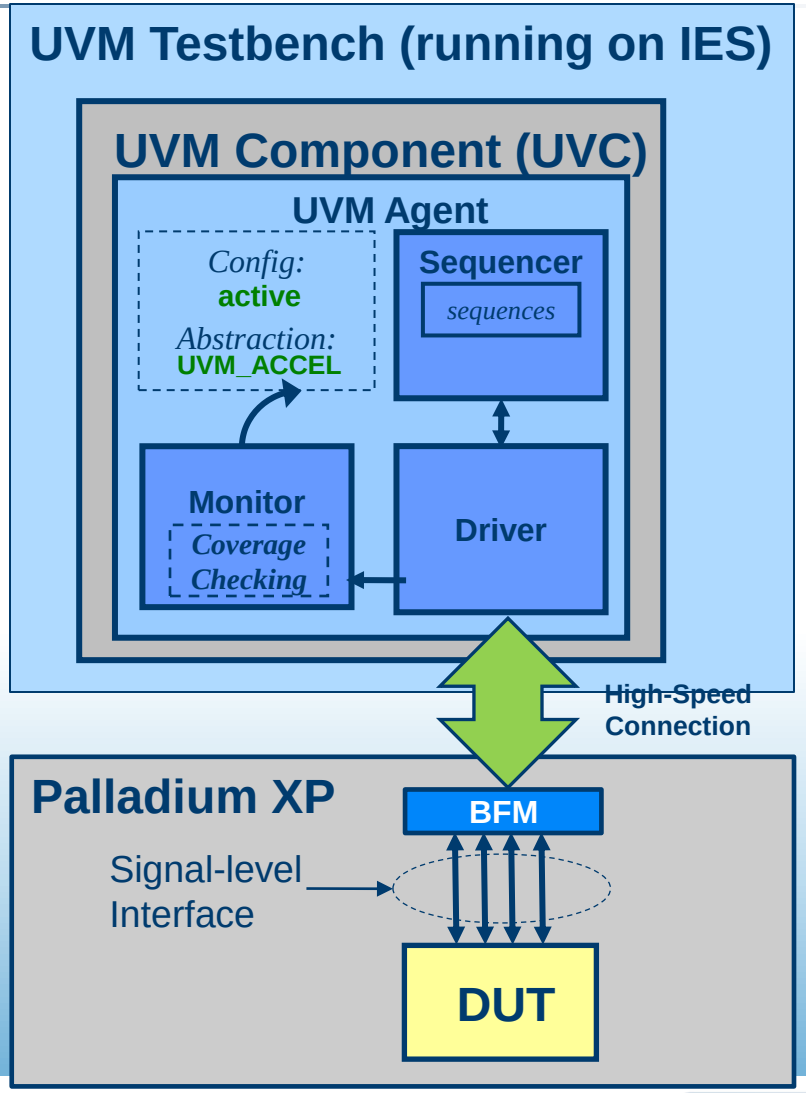
UVCs are instantiated in the *UVM Testbench*.



UVM STRUCTURE FOR SIMULATION ACCELERATION

Most of the environment in the simulator stays the same.

- The *Driver* takes sequences from the *Sequencer* and puts them into a High-Speed Connection to Palladium XP.
- In Palladium XP, a *BFM* takes the sequences from the High-Speed Connection and converts them to the signals needed by the interface.



WHY USE SPECMAN *e* FOR THE TESTBENCH?

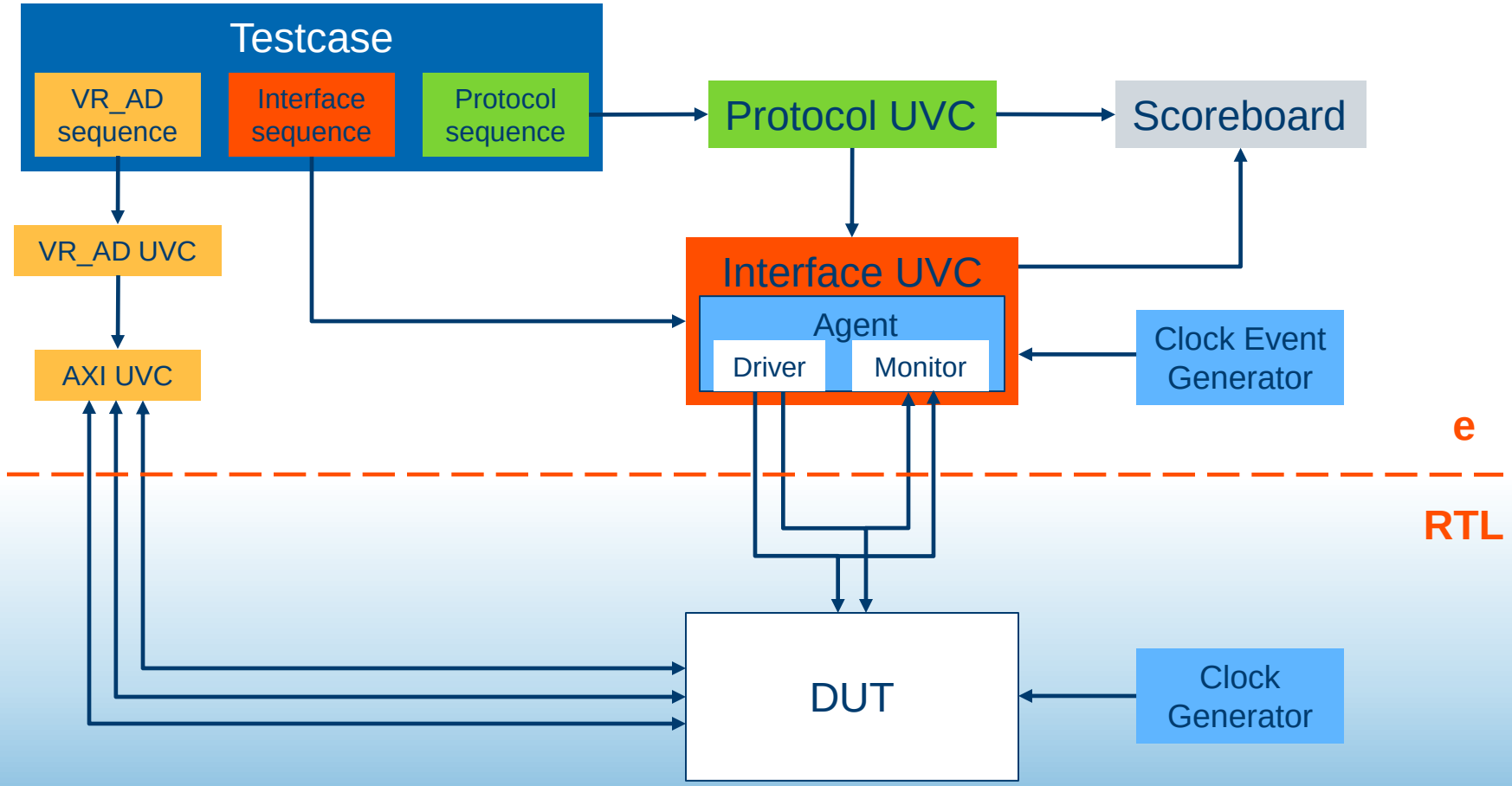
- *e* is an Aspect-Oriented Programming Language
  - Supports inheritance
  - *Supports extensions*
- Extension example:
  - An *Interface UVC* could be defined in a base *e* file.
  - A *UVM_SIGNAL* extension could add to the Interface UVC.
  - A *UVM_ACCEL* extension could add to the SAME Interface UVC.
  - Which extension is used depends on a variable in the *Config* unit.
- Result:
  - Easily switch from simulation to simulation acceleration.
  - Use the same pointers in the code, without the messy typecasting required by inheritance!

SPECMAN e EXTENSION EXAMPLE

```
extend UVM_ACCEL interface {
    connect_ports( ) is also {
        clk.disconnect( );
        resetn.disconnect( );

        do_bind(clk, empty);
        do_bind(resetn, empty);
    };
};
```

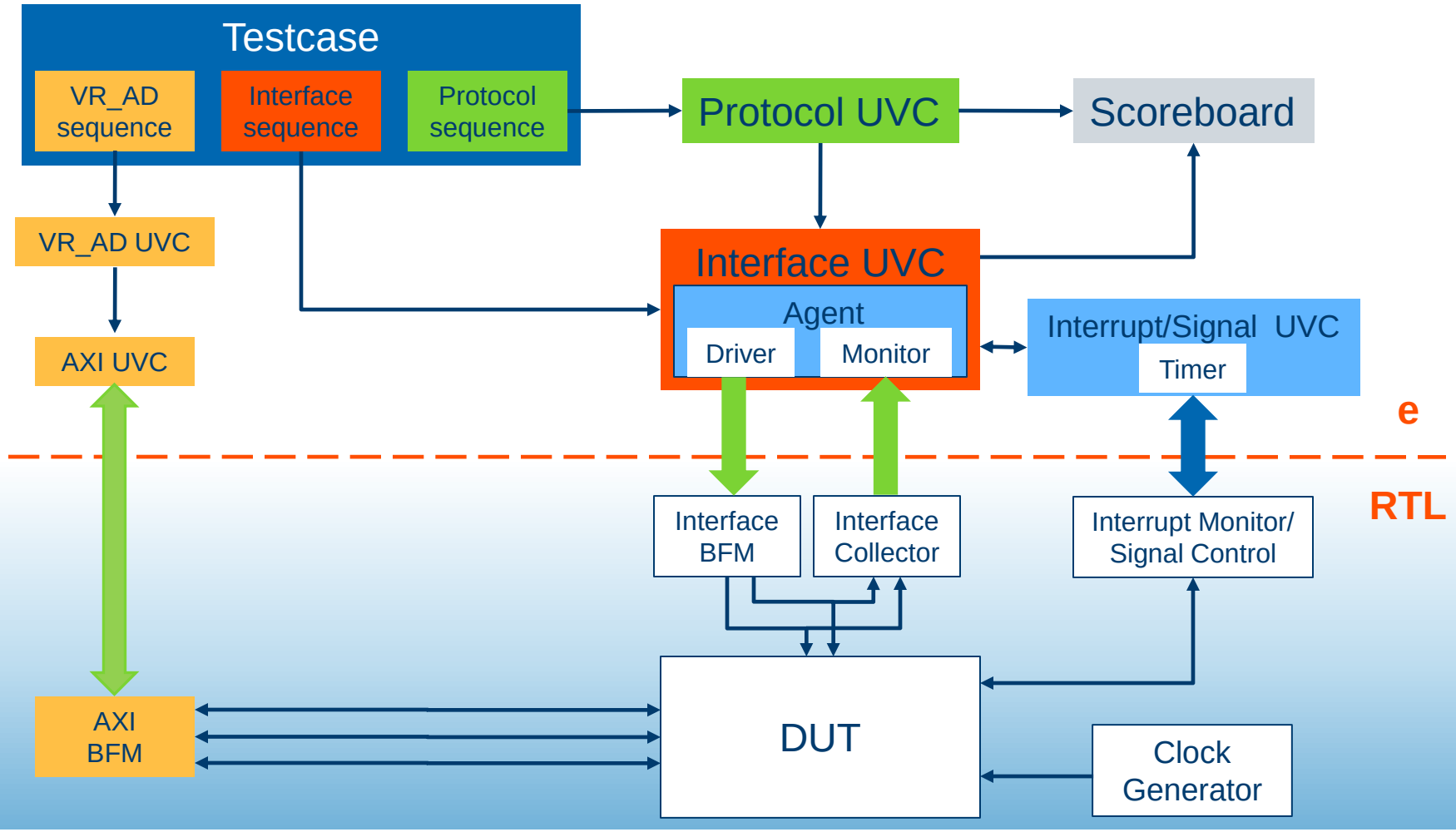
STRUCTURE FOR SIMULATION



CHALLENGES IN ADAPTING THE TESTBENCH TO SIM. ACCEL.

- To maximize performance, communication between the testbench and Palladium XP should be at the transaction level.
  - The testbench should not access individual signals in the DUT.
  - DUT clocks should not be used in the testbench.
  - Transactors had to be written provide interrupts to the software and signal-level control.
- Tool behavior differences
  - Some SystemVerilog constructs didn't compile into Palladium XP.
  - This required modifying some of the DUT RTL.

STRUCTURE FOR SIMULATION ACCELERATION



Signal → SCEMI ➡ SystemVerilog DPI ➡

PERFORMANCE ACHIEVEMENT

- Simulation: 1 hr
- Simulation Acceleration: 80 seconds
- Effectively, 40x speedup with a 26 Mgate DUT
- Results obtained with a suboptimal testbench

IDEAS FOR ADDITIONAL OPTIMIZATION

- The main performance bottle-neck is the testbench. To improve this:
  - Use the Specman *e* profiler to identify inefficient *e* code and optimize these blocks.
  - For some tests, pre-generate the traffic data, instead of generating the data on-the-fly in the testbench.
  - For some tests, move data generation from the testbench to the Palladium XP.
- The environment that IES runs in could also be improved:
  - Use faster workstations for running the testbench.
  - Use a dedicated file server for simulation acceleration to reduce file I/O.
- The DUT is configured by writing to registers through an AXI transactor.
  - Backdoor register access may speed up this process.
  - Or, use longer AXI burst transactions to send more data in a single operation.
- With these changes, the team believes that an additional 20x acceleration is easily achievable.

OTHER POSSIBLE OPTIMIZATIONS

- Compile the DUT to use the Palladium XP 1x clocking mode (instead of the default 2x clocking mode).
  - Uses more Palladium XP resources, but the DUT will run faster.
  - Currently, the DUT speed is not the critical path in simulation acceleration, so this may only make a small difference in simulation acceleration speeds.
- Optimize the UVC Hardware Collectors
  - Consolidate more data on the hardware side before sending to the testbench to minimize the number of times the data is transferred to the testbench.
  - This will add complexity to the collectors – complexity not needed in simulation.
  - The amount this would improve performance is unknown.

CONCLUSION / WRAP-UP

- Simulation Acceleration with a Specman Testbench is a viable solution for PMC.
- It's straight-forward to migrate from simulation to simulation acceleration in the PMC environment.
  - Major re-architecting of the testbench is not required.
  - Simulation and simulation acceleration can co-exist as part of an overall verification strategy.

NASDAQ: PMCS