

DAC Virtual Platform Workshop – July 29, 2009

---

# Integration of RTL Models into Virtual Platforms via Transaction-Based Emulation

THE **FASTEST** VERIFICATION

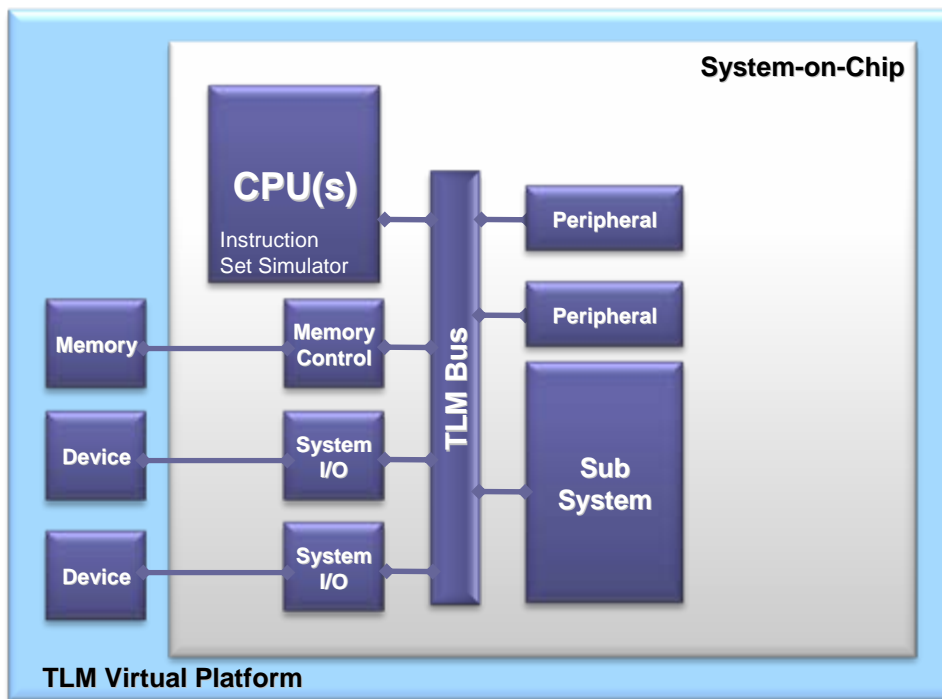


# Agenda

- **Introduction**
- **ZeBu Emulators**
- **Virtual Platform with Transaction-based Emulation**
- **Easy Transactor Integration**
- **ESL Partnerships**
- **Summary**

# Introduction

## Typical TLM Virtual Platforms

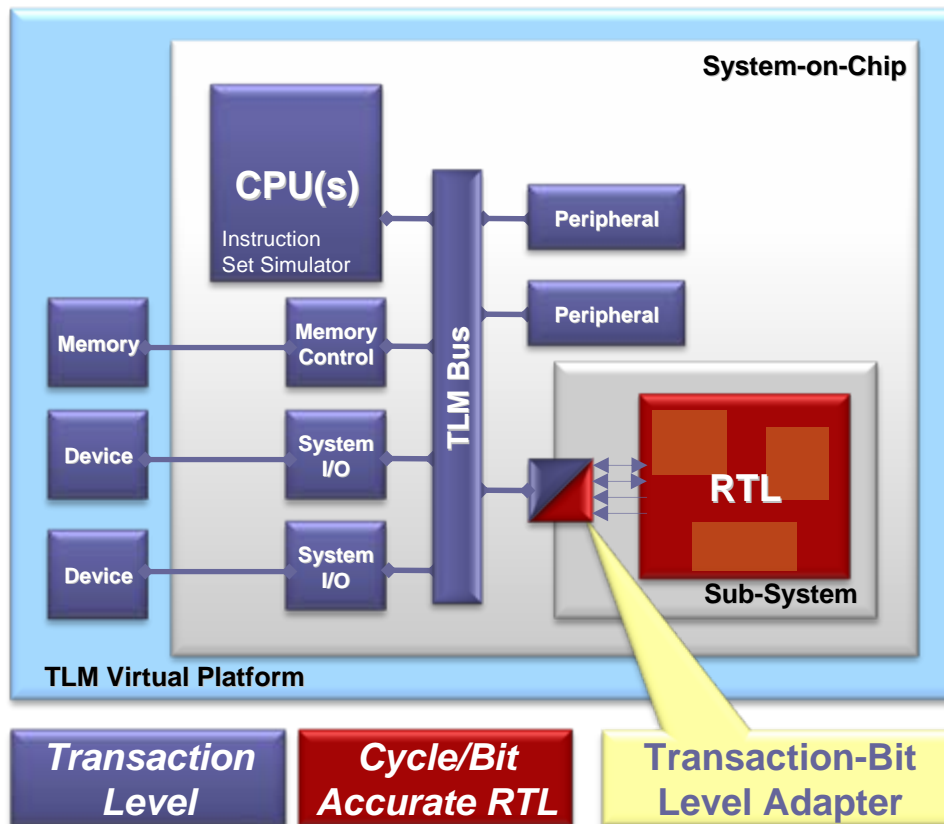


**Transaction Level**

- **Transaction-Level Modeling (TLM) used for the entire System**
  - OSCI SystemC-based standard
- **TLM systems provide very high performance simulation**
  - Great for early software development & architectural exploration
  - Various levels of timing accuracy:
    - Programmer's View
    - Timed or UnTimed
    - But no Cycle Accurate models

# Introduction

## TLM Virtual Platforms with an RTL Sub-System

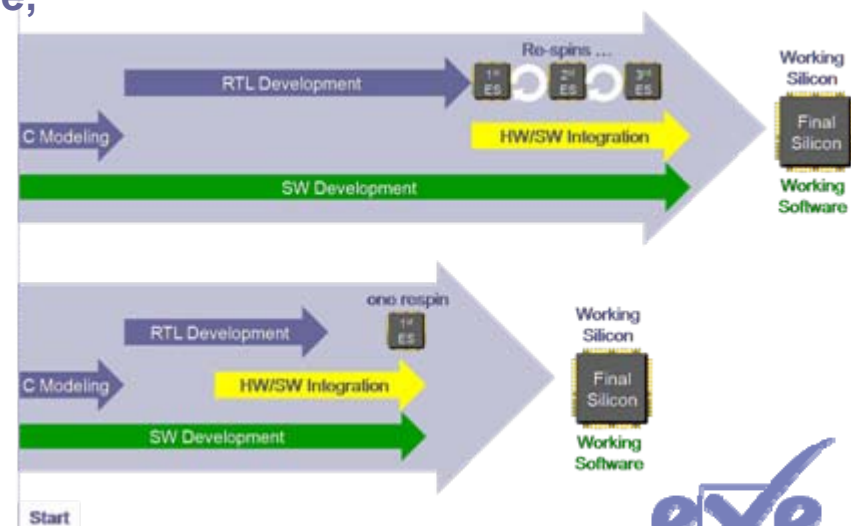


- **Hardware Sub-System is integrated with Virtual Platform**
  - Model available as RTL (SV/Verilog/VHDL)
  - Traditionally co-simulated with HDL simulation
- **Register-Transfer Level (RTL) code provides Cycle/Bit accurate model**
  - Requires adaption between TLM and Cycle accurate models : BFM

# Introduction

## Virtual Platform with an RTL Sub-System – Why?

- **Legacy RTL Sub-System: Early Architectural Exploration & SW Development**
  - Use RTL to provide accurate timing for architectural exploration and low level firmware development
  - Develop Application Software in a new VP without creating a TLM model for the Legacy RTL Sub-System
- **New RTL Sub-System: Early HW/SW Co-Verification**
  - Replace existing TLM blocks with new RTL, use the VP as the system integration platform, verifying both HW and SW
    - Using TLM as the golden reference, this provides more exhaustive testing of the RTL Sub-System
  - Turns some of the Post-Silicon debug into Pre-Silicon debug
- **Reduces overall product development time, and greatly improves pre-tapeout confidence**



# Introduction

## RTL Sub-Systems – Traditional Integration Techniques

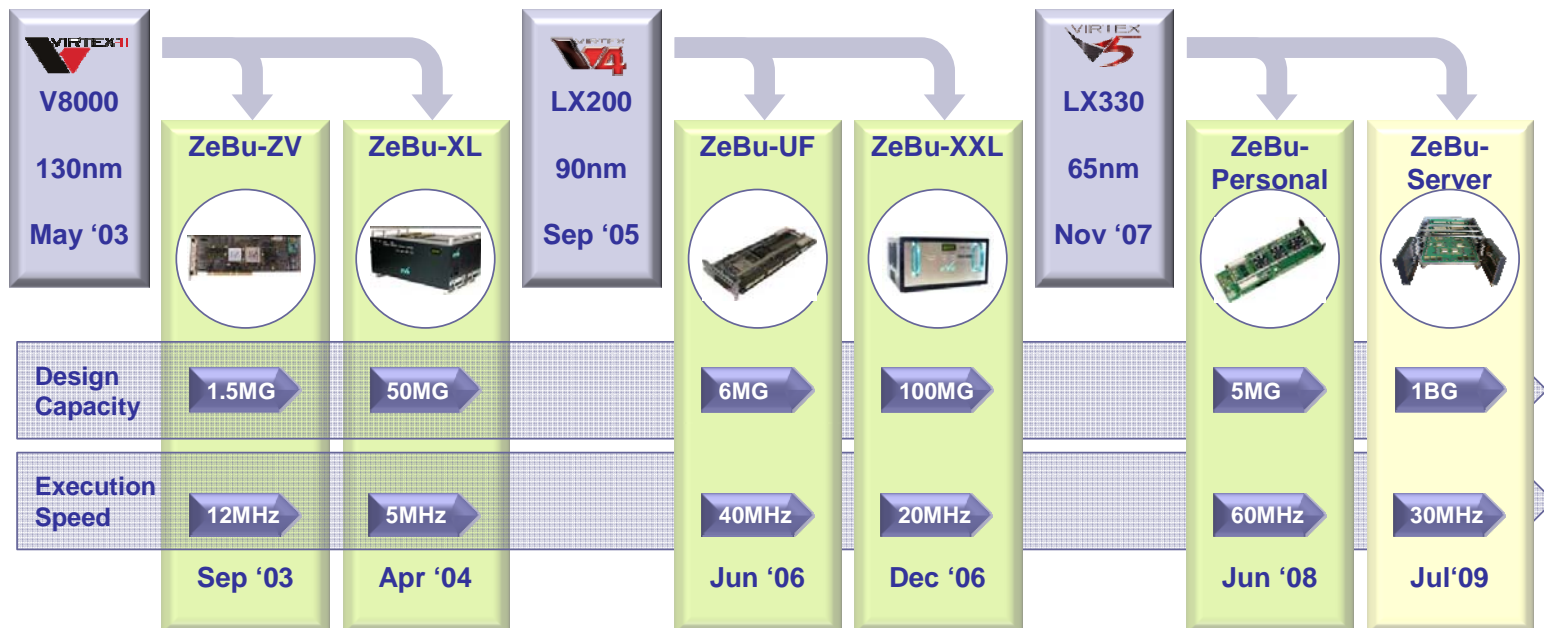
- **RTL Sub-Systems are traditionally co-simulated with VP using an HDL Simulator (e.g. VCS, ModelSim, NC-Sim)**
  - **Connectivity is usually cycle based (Verilog/VHDL), but may be transaction based as well (SystemVerilog/SystemC)**
- **Largest drawback is performance**
  - **HDL Simulators run orders of magnitude slower than a Transaction-Level Virtual Platform**
    - **Limits integration to only very small RTL Sub-Systems**
    - **Not practical for SW development**
- **Adaptation between the Transaction-level and the Cycle/Bit-level RTL abstraction levels is manual**
  - **Need a Cycle Accurate Model of the Bus, written by the user in SystemC/SystemVerilog/Verilog/VHDL**



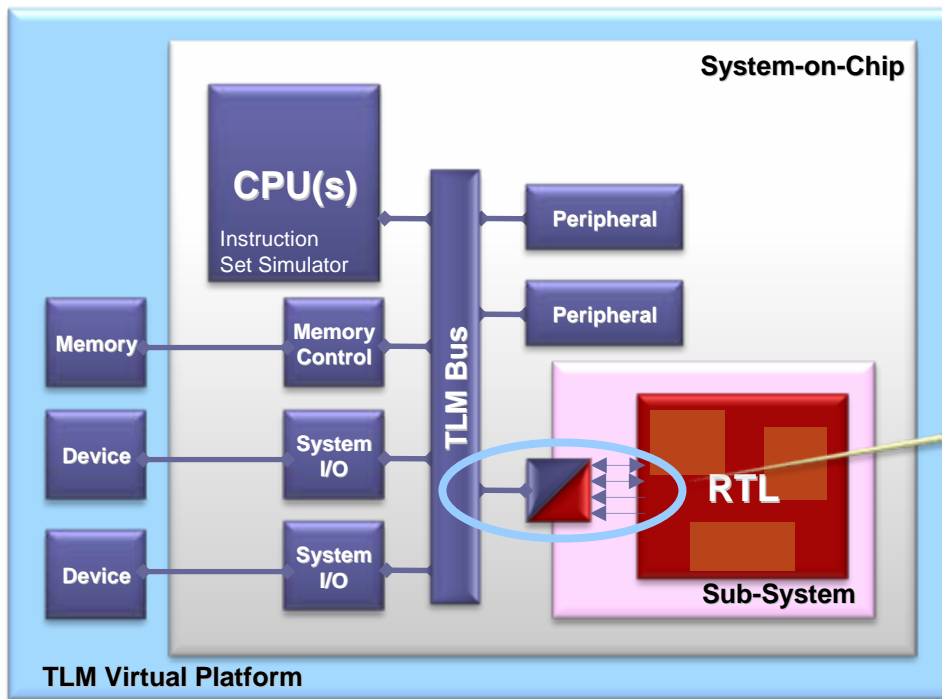
# EVE ZeBu Emulators

## A Family of Products Ideal for ESL integration

- *RTL Emulators execute a cycle-accurate model of the design in specialized hardware, while also providing debug capabilities similar to simulation, e.g. clock control, waveform dumping*
  - *EVE's ZeBu emulators use Xilinx FPGAs and external memories to model RTL*
- *FPGA-based ZeBu emulation can run at Multi-MHz speeds, and Millions of transactions/second*
  - *Largest portfolio of high performance Transactors*
  - *Typically between 1-10% of the real SoC speed*
  - *Fast enough to: Boot Linux, Develop Device Drivers, Run Application Software*
- *Two Product Lines: Desktop Based for Sub-Systems, and Stand-alone for Full-Chip Designs*



# TLM Virtual Platforms with Transaction-based Emulation



- Principles are the same as for RTL simulation, but with the Emulator executing the Sub-System

- Transaction to Cycle/Bit Level adapter is called a **Transactor**

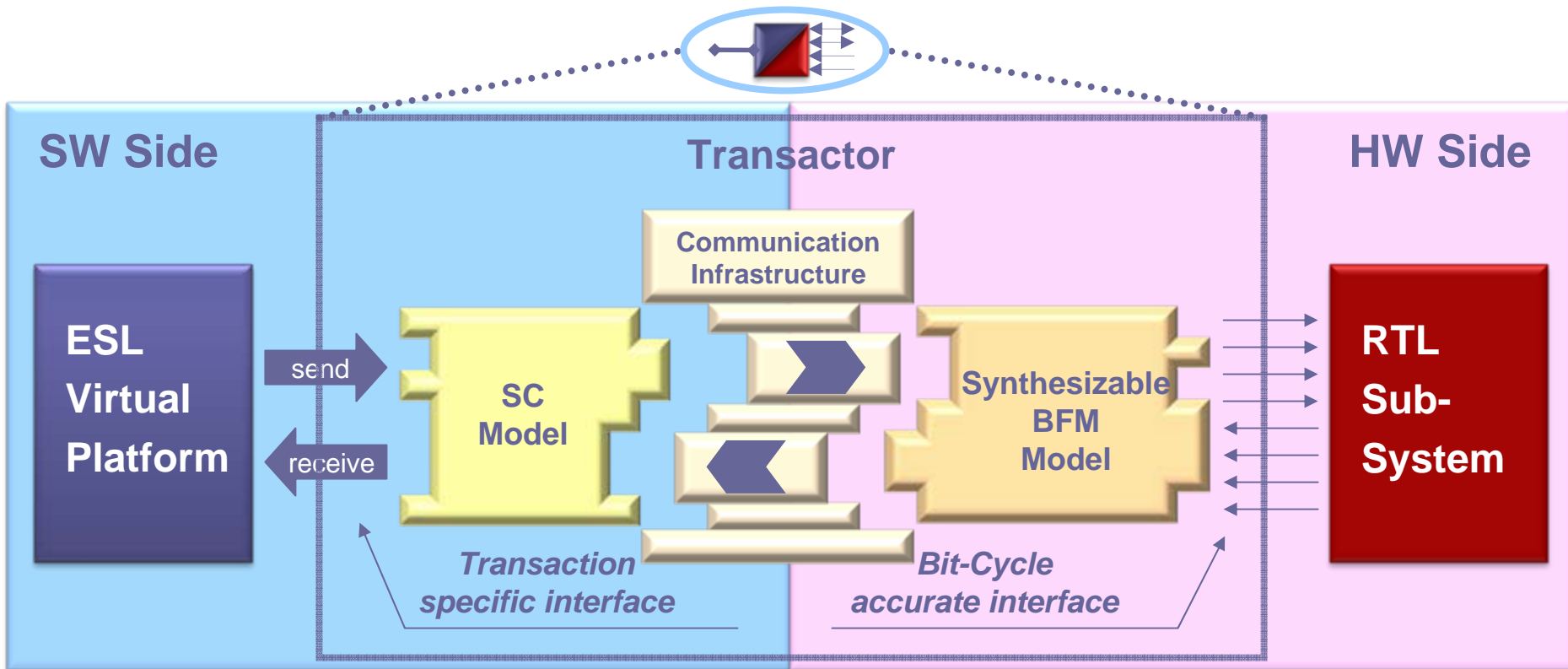
- All cycle/signal level conversion is performed in the Emulator BFM , preserving performance
- Transactor API provides transaction handling methods



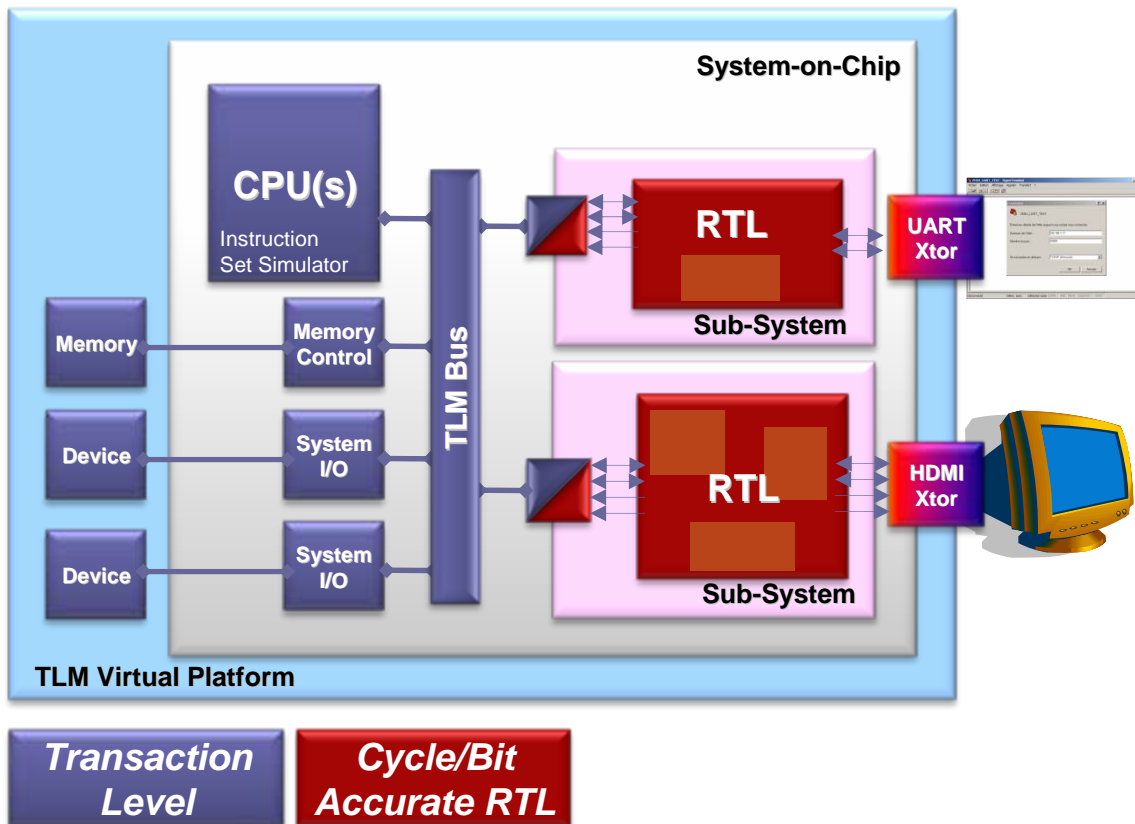
# Virtual Platforms with Transaction-based Emulation

## Anatomy of a Transactor

- Contains both a software (SystemC/C/C++ API) and synthesizable (SV/Verilog) Bus Functional Model (BFM) portion
- Communication between the two sides is implemented via EVE's high speed proprietary messaging system

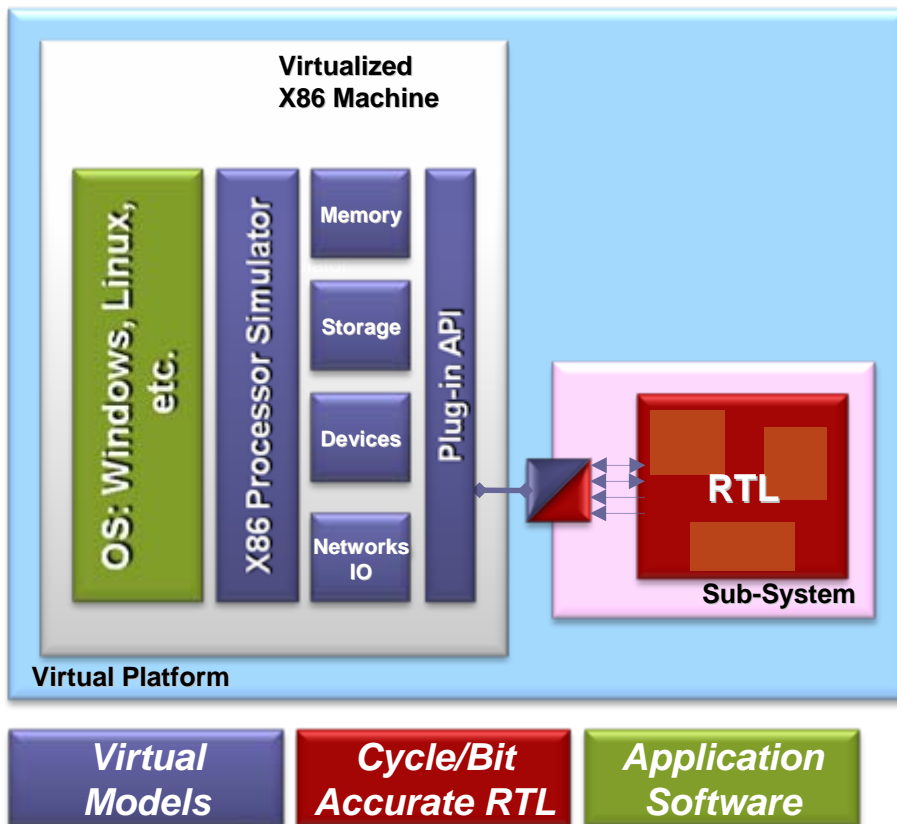


# Virtual Platforms with Transaction-based Emulation Extensions – External Interfaces to RTL Sub-Systems



- Transactors can also be used to connect external interfaces to the RTL Subsystems
- High level of accuracy : Pin accurate , protocol compliant
- High performance

# Virtual Platforms with Transaction-based Emulation Extensions – Non-TLM Integration



- **Some Virtual Platforms and Transactors might not be TLM**
  - e.g. SystemC but not TLM, or a completely Virtualized x86 machine
- **ZeBu emulator/transactors can still be integrated**
  - ESL environment can call transactor API functions
- **No change in performance**
  - All cycle-level activity still occurs inside the emulator

# Virtual Platforms with Transaction-based Emulation Advantages

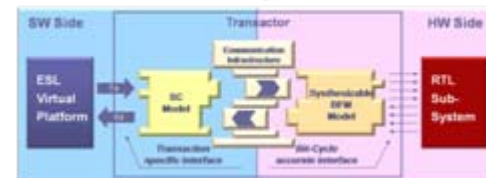
- **High Performance, even on large RTL blocks**
  - Typically 1-10% of real SoC speed, with 100% accuracy
  - Comparable performance to traditional Virtual Platforms
- **Easy Bring-Up**
  - Emulators feature “push-button” compilation, and do not require advanced synthesis or hardware experience
  - ZeBu supports communication at the transaction-level, which matches the level of abstraction used in the Virtual Platform
    - Integration can be TLM or non-TLM based
  - Test environment is easily scalable, including multiple peripherals implemented as RTL
- **Debug**
  - Emulators provide similar debug features to HDL simulation, e.g. waveform dumping, clock control
  - Emulators provide similar debug access as for the VP: Memory load/dump, Register read/write



# Easy Transactor Integration

## EVE Transactor IP

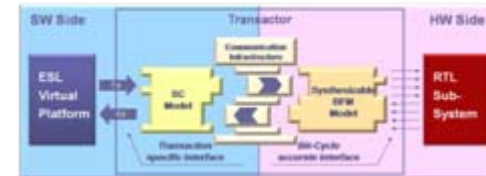
- **EVE's Vertical Solutions provides an extensive library of off-the-shelf transactors, modeling some of the most common standards**
  - PCI-Express, AXI, AHB/APB, USB, Ethernet, UART, JTAG, Digital Video, HDMI, MMC , etc...
  - Transactors available as Master/Slave, Host/Device, Endpoint/Root Complex
- **EVE Transactors are packaged with both the Synthesizable portion, as well as the API**
  - API functions are at the transaction level, maintaining high performance
  - Transactor API functions are directly callable from ESL environments
    - Supports both UnTimed and Timed execution



# Easy Transactor Integration

## Custom Transactors with ZEMI-3

- For proprietary internal bus modeling or checkers, custom transactors may be required
- ZEMI-3 is a transactor compiler that:
  - Generates RTL BFMs from simple behavioral code
    - Behavioral SystemVerilog code requires no HW or synthesis expertise
    - Easy to validate transactors using a SystemVerilog simulator
  - Automatically handles the low-level communication layer between HW (SystemVerilog) and SW (C/C++)
    - ZEMI-3 transactor functions are easily integrated into ESL SystemC environments
  - Automatically detects streaming applications that can be optimized for performance

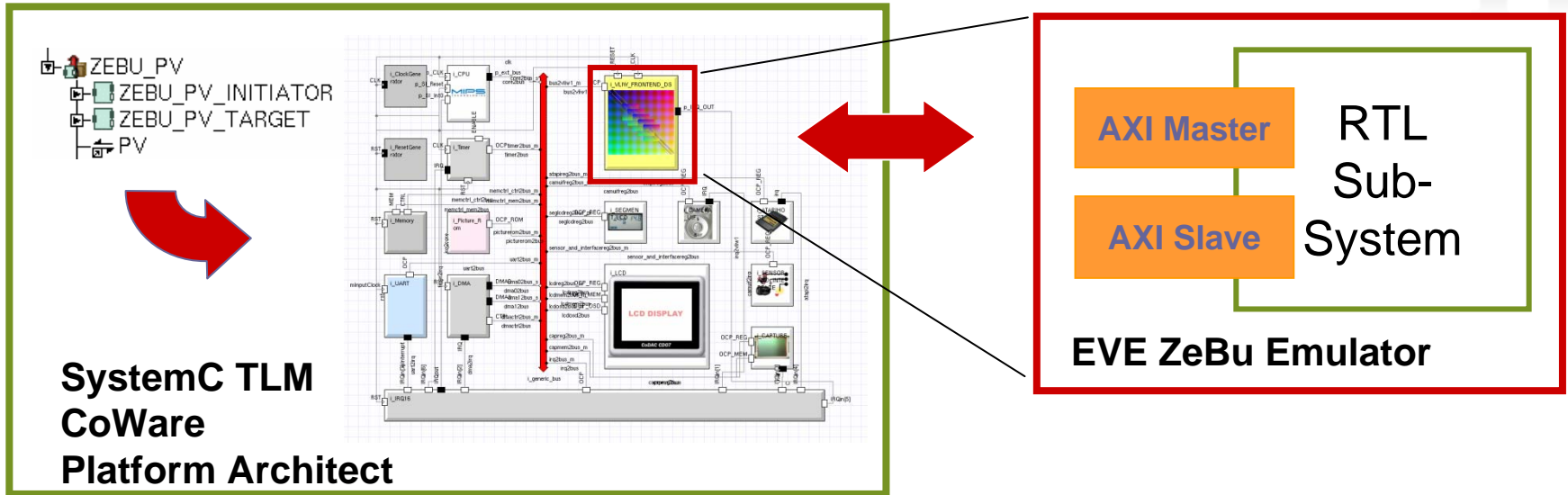


# ESL Partnerships



- To further ease the integration of Transaction-Based Emulation and Virtual Platforms, EVE has collaborated with a number of ESL providers, including:
  - CoWare
  - ARM
  - Synopsys
  - Virtutech
  - Imperas (OVP)
- With the flexible TLM and API based interfaces, integration with OpenSource ESL environment (e.g. OSCI, QEMU) are easy to implement as well

# ESL Partnerships CoWare and EVE



From CoWare:

- New Zebu-PV initiator and target adapters
- Platform Architect and Virtual Platform environment

From EVE:

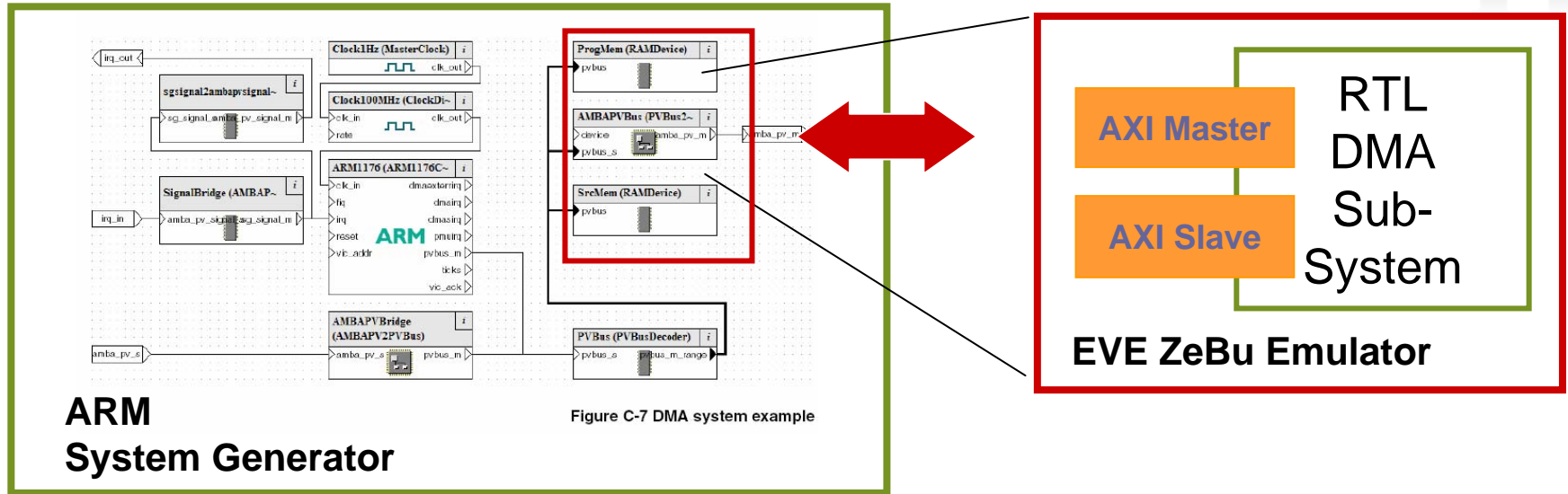
- New AXI master and slave synthesizable transactors
- ZeBu emulation environment

## Benchmark results for a Digital Still Camera Design with Emulated DSP Sub-system

- 3X faster than SystemC CA model; 300X faster than Co-simulation
- 1 picture per second



# ESL Partnerships ARM and EVE



## From ARM:

- Use PV modeling in the Virtual Platform
- FastModels and RealView environment

## From EVE:

- New AXI master and slave synthesizable transactors
- ZeBu emulation environment

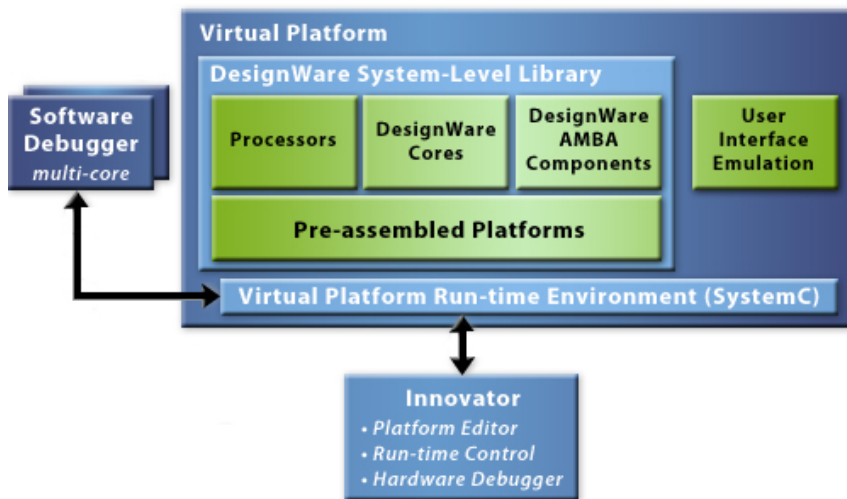
Run ARM SystemGenerator reference platforms with ZeBu emulated design :

- Based on the same PV\_Adapter EVE technology using TLM 2.0
- Dhystone and DMA examples
- Large set of ARM FastModels Ips for new CPUs



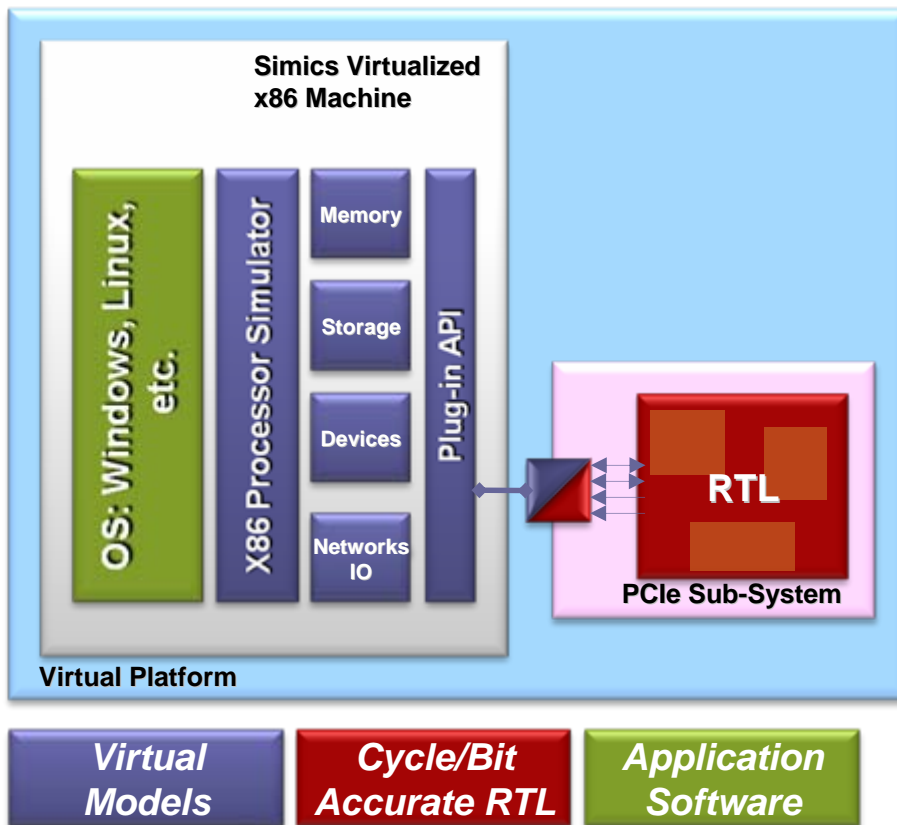
# ESL Partnerships

## Synopsys Innovator and EVE



- Synopsys' Innovator provides an integrated SystemC development environment for assembling Virtual Platforms
- SystemC Platform can call EVE transactor API functions
  - External interface can be modeled with Innovator (e.g. xterm interface), connected to an EVE UART transactor
  - UART transactor transmits/receives characters to/from the emulated design

# ESL Partnerships Virtutech and EVE



- **Virtutech® Simics® provides a virtual version of target hardware**
  - Includes x86 processors and peripherals
  - Virtual Platform behaves exactly the same as the physical target hardware, and runs the exact same binaries
- **Simics' Plug-in API can be used to call EVE transactor functions**
  - e.g. Simics can boot Windows XP, connect and enumerate an emulated PCI-Express device through EVE's PCI-Express transactor



# Summary

- **RTL Sub-Systems can be integrated into a Virtual Platform to reduce the overall product cycle**
  - Enables Early Architectural Exploration, SW Validation, HW/SW Co-verification with accurate timing
- **HDL Simulation is the traditional integration method, but is limited**
  - Huge performance degradation, manual creation of adapters
- **Transaction-based emulation with ZeBu provides cycle-accuracy, while executing fast enough for software development**
  - Multi-MHz performance on RTL, 1M+ transactions per second
- **EVE's Transactor IP and ZEMI-3Transactor compiler make it easy to plug-and-play or create adapters**
  - Keeps cycle-level activity in the emulator for performance
  - Supports TLM and API based integration
- **Collaboration with ESL tool providers ensures easy bring-up of the integrated environment**



# More Information

- Visit our DAC booth: #908
- Web: <http://www.eve-team.com>
- E-mail: [ronchoi@eve-team.com](mailto:ronchoi@eve-team.com)