Use of virtual platform for verification of an electronic system design

- Introducing ASTC.
- Introducing Virtual Platforms in V&V cycle.
- Hybrid prototyping.
ASTC Summary:

- Software products and engineering services company
  - Australia, USA, Japan and Europe locations
- 1994: Origins go back to 1994 as a Global EDA Center of Motorola
- 2005: Incorporated as an independent company
- 2011: Launch of VLAB Works business outlet

- Vendor of
  - ESL: System Simulation and Virtual Prototyping Software (VLAB Works)
  - HW: ASIC/IC Design Services
  - SW: Embedded SW Development

- Markets
  - Automotive
  - Control
  - Communications
  - Aerospace
  - Multimedia

- Customers:
  - Supply Chain for Embedded Electronics: OEM, ODM, Tier 1, Tier 2 (Semi and IP)

© ASTC 2015 - Confidential  www.vlabworks.com
The Virtual Platform Based V&V flow allows:
- V&V of the Production SW (exact binaries)
- Validation of HW Spec (VP model)
VLAB Virtual Prototyping: main use cases

System Modeling
- Model Development
- SOC to System Models
- Pre-Packaged Virtual Platform
- 3rd Party model integration

System Analysis
- HW-SW Performance Analysis
- Architecture Exploration

Test Executive
- Fault injection
- Multiple configurations
- Automated Regressions
- Test creation

HW/SW Verification
- Hardware validation
- HW/SW verification
- Early Software Development
- Software Debug
- Software Analysis
- HW/SW validation
Virtual Platform From User’s Perspective

Stimulus
- Software, scripts
- Test scenarios
- Dashboards

3rd Parties Connections
- Source code debuggers – GHS, GDB, ...
- Test environment – Vector CANoe, ...
- Co-simulation – HDL, RTL, AMS, FPGA
- Hardware (HiL, RealIO) – CAN, Ethernet, USB
- Plant models – Matlab/Simulink™

VLAB – Virtual Platform Creation and Simulation Environment
Virtual Platform (pre-packaged) – ARM, PPC, ...
Other Models – SystemC or Python models of other ECU components and ICs
Killer Features of Virtual Platforms for System-Level Debug & HW-SW Verification:

- Non-intrusiveness
- Controllability
- Repeatability
- Visibility
- Programmability
  - Fault Injection
  - Regression Test
  - Advance Instrumentation

Anything that can be observed can be subjected to a fault:
- at port/signal level
- at bus transaction level
- at register or memory level
- software failures
- etc.

Typically, a user would need to create a VLAB breakpoint and corresponding action for fault injection.
HW-SW Verification using VLAB Virtual Platforms: most common use cases.

Virtual Platform Based V&V

- SW/VP Validation
- SW Pre-Calibration
- SW Verification
- SW Test
- SW Debug
- SW Bring Up

System I/O Modeling
Test Harness creation

VLAB™ to connect & execute any of these Components:
- Python
- C++
- IPXact
- TLM
- SystemC
- SysC/AMS

Executable HW Specification

SW Driver & IP design verification

System Model
- HW Arch
- HW Spec

Virtual HW Platform (VP)
- RTL Design
- FPGA/Emulator Prototype

Real-IO or FPGA based Co-simulation: HW-SW verification with HIL

© ASTC 2015 - Confidential www.vlabworks.com
Example of HW - SW verification: RTL Co-simulation

**SW Engineer** developing drivers for peripherals, using VLAB virtual platform for software execution, debug, and co-simulation of the RTL IP developed by **HW engineer**.

**VLAB** – Virtual Platform Creation and Simulation Environment

Virtual Platform
- Including peripheral models

**RTL co-simulation Toolbox**: co-simulation with RTL simulator

**Trace32** for software development, debug, and compilation

**RTL simulator running CAN IP RTL code**
Hybrid Prototyping

- Combines the advantages of Virtual Prototyping and FPGA Prototyping (Emulation)

- Use cases
  - Accelerated RTL verification and early validation
  - HIL/SIL: interfacing with physical hardware and networks
  - Early software development

- Challenges
  - Overall performance
  - Cost effectiveness
  - Complexity of setup
  - Scalability
Hybrid Prototyping with VLAB

- Utilizes off the shelf FPGA boards, lower cost and easier scalability
- Uses PCI-Express FPGA interfaces for very low, sub-microsecond communication and synchronization latency
- FPGA capacity of 45K-1M logic cells (approx. 0.4 to 8 million useable ASIC gates) per board, up to four boards per host
- Bus and signal interfaces, time coupled or decoupled operation
Hybrid Prototyping Architecture

Host PC
- VLAB Virtual Platform
  - Core
  - Peripherals
  - Bus Model
  - RTL Interface Proxy Model
  - FPGA Interface API

FPGA Board
- FPGA
  - RTL IP (DUT)
  - Instrumentation Block
  - Clocks
  - Signal I/F
  - Bus I/F
  - Bus Bridge (Optional)
  - AXI Interconnect
  - PCI-e to AXI Bridge
  - Hardware PCI-e Block

PHY I/F (optional)

PCI-Express Link

ASTC IP
- Customer IP
- Xilinx AXI IP
Also known as quantum based synchronized co-simulation

- Both sides operate for a global time period called quantum
- During a quantum, emulation is asynchronous to the simulation
- Synchronization and data exchange at quantum boundaries
- Quantum should be sufficiently small to not miss important bus transactions and signal changes, e.g. interrupts
- DUT clocks are gated to achieve timing synchronization
Operating Modes: Decoupled Time

- Decoupled time co-simulation
  - Both VLAB and FPGA operate at speed
  - Interactions occur as they happen, with synchronization for data exchange
  - Best performance, but need to be careful with interface timing and host software induced communication jitter
- Practical for
  - Communication centric IP where buffering and flow control features allow time decoupling
  - Computation centric IP where timing is not critical
Dedicated FPGA blocks for effective hybrid prototyping:

**Master End Point Block**

- The Master End Point (EP) provides means to support bus master RTL IP on the FPGA
- Enables RTL IP to initiate transaction at any point
- Stores transactions until host can process
- Can accept next transaction as soon as host processes the current one

**Instrumentation Block**

- Instrumentation block is responsible for
  - Signal interface. Up to 1024 input and 1024 output signals from RTL IP can be connected and sampled
  - Clock control (for time coupled co-simulation). Up to 4 clocks can be controlled. Future upgrade to 8 clocks
- Interaction with the SystemC proxy module via register interface and FPGA API
Technical Specifications

- **PCI-E support:**
  - 1.x, 2.x, or 3.x (device specific)
  - x1 to x8 lanes

- **Typical synchronization and communication latency**
  - VLAB to FPGA: ~1uS
  - FPGA to VLAB: ~10uS

- **Bandwidth:** >1GB/s (x8 PCIe 2.0)

- **Host OS:** Linux, Windows

- **Clock control:** multiple DUT clocks supported, synchronized clock domains

- **FPGA design size, excluding DUT:** 15,000 logic cells

- **Quantum period:** variable, can be set for each individual quantum

- **AXI bus support:** 64- or 128 bit

- **FPGA devices supported:** Xilinx KC705, VC707, KCU105 and VCU108 (Q4 2015)