VPSim: Virtual Prototyping Simulator

- **VPSim Platform Overview**
  - VPSim includes a large library of component models: CPU models (RISC-V, ARM, ...), peripherals and memory models.
  - The main supported model provider in VPSim is the open source system emulator QEMU [2], which allows unthrottled simulation speed.
  - VPSim integrates QEMU by running its CPU and peripheral models within SystemC threads. These QEMU models are enriched with performance information [3].
  - VPSim stands out for its capacity to accommodate external subsystems through different standard and non-standard interfaces like SystemC, FMI, Python, and HW designs [4, 5].

- **Platform composition and simulation**
  - VPSim provides a user-friendly interface to compose and build virtual platforms using an in-house Domain Specific Language (DSL) based on Python.

RISC-V based platforms in VPSim

- **RISC-V single core platform**
  - The VPSim framework was designed to facilitate early-stage computer architecture design by providing support for SW/HW co-design [1]. It is a modular and highly configurable framework for:
    - Architectural exploration: by providing configurable models to evaluate the performance of different platform configurations.
    - Software design: by providing an enhanced user space to run, profile, and debug complete software stacks (e.g. BIOS, hypervisor, user space workloads) on the simulated platform.

- **VPSim includes a large library of component models**: CPU models (RISC-V, ARM, ...), peripherals and memory models.
- **The main supported model provider in VPSim is the open source system emulator QEMU** [2], which allows unthrottled simulation speed.
- **VPSim integrates QEMU by running its CPU and peripheral models within SystemC threads**. These QEMU models are enriched with performance information [3].
- **VPSim stands out for its capacity to accommodate external subsystems through different standard and non-standard interfaces like SystemC, FMI, Python, and HW designs** [4, 5].

**Platform composition and simulation**
- **VPSim provides a user-friendly interface to compose and build virtual platforms using an in-house Domain Specific Language (DSL) based on Python.**

**Describe**
- The user describes the targeted platform in Python script.

**Build**
- VPSim builds the platform, incorporating available software binaries.
- Set up the SystemC/TLM simulation kernel.

**Simulate**
- One VPSim binary can simulate infinitely.
- The user can control the simulation at a high level.
- Software debugging and profiling.

**VPSim Workflow**

**References**
[4] Further references and details can be found in the project documentation available online.

**Conclusion & Perspectives**
- VPSim presents a valuable solution for HW/SW designers to quickly evaluate and refine their designs while minimizing associated time and costs.
- It offers an extensive range of performance counters and statistics for profiling and benchmarking purposes while ensuring a commendable trade-off between precision and execution time.
- Continued improvements are being made to VPSim, introducing a host of new features that enhance its capabilities and yield superior results for evaluation and benchmarking purposes.

**ACKNOWLEDGMENT**
This work has been performed in the context of the European Processor Initiative (EPI) project, which has received funding from the European Union’s Horizon 2020 research and innovation program under Grant Agreement № 826647 and Specific Grant Agreement № 101036168 (EPI SGA2).