## Benefits of a Common Methodology for Emulation and Prototyping

Many design teams have used some form of hardware verification throughout their verification cycle for years now. Some engineering teams prefer to use emulation, some prefer to use prototyping, and some even use both. Why would engineering teams invest in both platforms?

Design teams typically use emulation for verification when RTL is less stable. Fast bring-up of multiple compiles per day as well as simulation-like debug are unparalleled in a processor-based emulation system because you have full control of compile, you do not need to worry about timing closure of FPGAs, and debug is not intrusive.

When engineers reach the time during a project when RTL becomes stable, speed becomes crucial for software development and hardware regressions. You no longer have to compile multiple times daily, so FPGA-based prototyping become the right platform to use here. And with a unified environment, not only can the bring-up be done in hours from a stable emulation model, but also when a defect is found at the hardware/software interface or in hardware during a regression, it can be reproduced in emulation using simulation-like debug.

In this workshop, learn why you should consider bridging emulation and prototyping into a continuous verification environment to speed up your verification throughput for early software validation and real world testing. This workshop will cover:

- Fast design bring up between platforms (e.g. common implementation flow, common look & feel UI)
- Advanced debug (e.g. FullVision engine, probes, memory force and release, etc.)
- Re-usable system-level interfaces (real-world testing)