Is Your System-Level Project Benefiting from Collaboration or Headed to Chaos?

Steve Brown

Steve Brown is Director of Marketing for Enterprise Verification Process Automation at Cadence Design Systems. He is a 20-year veteran of the EDA verification industry and has held various engineering and marketing positions at Cadence, Verisity, Synopsys, and Mentor Graphics. He specializes in solving engineering, management, and marketing challenges that arise when new technology and products enter the market. He earned BSEE and MSEE degrees from Oregon State University and has studied marketing strategy at Harvard, Stanford, Kellogg, and Wharton.

In order to parallelize project operations and meet aggressive schedules, system project teams designing both hardware and embedded software must address the need for much higher frequency of interactions. Without better forms of communication, automation enhancements, and verification engines that are powerful and flexible, attempts to parallelize flows will result in chaos and project paralysis.

Industry trends in electronics are resulting in design and verification schedules becoming more compact and complex. The use of SW to implement more functionality provides flexibility, but also compounds the difficulty in completing verification because of the need to do more HW/SW co-verification before silicon is available. The emergence of effective HW/SW co-verification solutions alleviates this pressure technologically, but it creates an unanticipated burden for many project teams: frequent daily interactions between HW and SW teams as they converge on closure independently and collectively.
In the world of software design, engineers typically try to steer clear of the complex hardware verification process. Historically they have presumed a stable silicon platform upon which to run and test their drivers and applications. In some cases this results in finding hardware bugs late. But the benefit is that most of the busy-work of designing the hardware has been completed. As more of the hardware and software is covered, the more the software team is exposed to the noisy, tumultuous process of reaching closure of the hardware. While in theory parallelizing hardware and software is an overall gain for the project, the team must address the significant increase in details that will be discovered and resolved while exposing the software team to the hardware development. Without an effective plan to handle the volume of issues, the project will stall and all parallelization benefits will be lost.

Communication Barriers Torn Down
When it comes to system-wide applications the “communication gap” is just not acceptable any longer. Hardware design and verification has matured to a point where we regularly see predictable schedule success when applying verification process automation. System verification projects must leverage these same approaches for HW/SW coverage and system-level closure. With more upfront planning between the hardware and software teams, communication and understanding of what the other is doing will bring huge benefits. The result of planning is documented intent, and measures of success using coverage data like code coverage, assertion coverage, and functional coverage, made visible across the team. Assumptions that each team needs to make based on the initial system specification will be regularly reviewed, check-points will be established continually checking for system-level bugs, and a broad-based agreement on how the system should behave to reach full system-level closure will be agreed upon. The metrics built into the plan will apply to the necessary check points of the hardware and embedded software and be tracked and managed together in a system-wide management solutions leading to prioritization and escalation of problems and priorities if needed.