

Low Power Design – we need a system perspective!

Written by Mike Bartley of [TVS](#), and Robin Kennedy and Jon Older of the [NMI](#)

The NMI recently organised a one day conference taking a software perspective on low power hardware design verification. While hardware designers are becoming increasingly adept at adding clever power saving features into their designs, it is not always clear how software engineers use them. This day was targeted at trying to bridge that gap between hardware and software engineers.

The delegate profile showed that the room was from a predominantly hardware perspective with 70% of attendees declaring themselves as hardware engineers, 20% as software engineers and 10% being the rare breed that straddles the two. Some major themes became clear during the day. The first was the need to accurately estimate power consumption early during the design cycle. The second was that low power is in conflict with other system design objectives such as high performance. Another theme was how to make it easy for software engineers to understand and to utilize the low power design features in hardware.

The first theme was clearly articulated by Paul Bailey of ST-Microelectronics. Paul works in “Home Entertainment and Displays” and is responsible for the tools methodology and flows at a division level with a particular focus on low power. Although these devices are not mobile, low power is increasingly an issue to meet thermal and regulatory constraints. Paul told of us a recent chip project that was cancelled during architectural analysis because of the predicted power consumption. Paul noted that the level of accuracy currently available was so variable that to be confident of coming in under power budget designs are being scaled back.

Both Alex Grove of Mentor and John Aynsley of Doulos considered the use of SystemC TLM models to perform this analysis but the conclusion seemed to be that this is an emerging technology. Nick Heaton of Cadence noted that even if we had the technology to make the analysis then identifying the software use cases was currently too complex.

Carl Culshaw joined the panel to consider low power from a systems engineering perspective. He is an Automotive Systems Engineer from Freescale Semiconductor and he clearly defined the biggest low power challenge as conflicting consumers’ demand. For example, demands for increased performance, better reliability, improved safety features and reduced running costs all have the potential to increase power consumption. Adding in new legislation for emerging standards only exacerbates the challenge!

Kerstin Eder and Steve Kerrison of Bristol University opened the third theme of how software engineers play their part in low power design. They have been looking at giving more control over energy consumption to software developers and how to automate low power optimizations at compile time. For example, one can reduce the accuracy of data representations and thus avoid some of the power hungry storage switching. In addition, they demonstrated that the complexity of energy modelling and optimisation grows significantly when considering multi-core and multi-threaded systems.

Barry Lock of Lauterbach demonstrated a system, initially developed for the mobile handset market, which gives immediate feedback on hardware power correlated with program flow from the trace port enabling software developers to immediately home in on areas of the code which can give the most improvement, for example minimising SDRAM access.

Others asked the key question of “how do software engineers access our complex low power hardware tricks?” and Carl Culshaw pointed out that engineers have been striving for a number of years to access hardware details through abstract software layers. Paul Bailey noted that ST had overcome this problem by defining a software API that allowed software engineers to access low power features via function calls. Similarly Carl Culshaw highlighted that Freescale have a similar scheme, although the ‘standardised’ approach that comes with automotive software architecture systems, such as AUTOSAR, frequently hides many of these low power features.

So, no clear answers but a clear articulation of the issues: accurate and early power estimation; balancing conflicting system design objectives; and making it easy for software engineers to access the low power design features in hardware. In what is fundamentally a systems issue, it seems there is a need to draw on expertise from the embedded software, hardware and verification communities to ensure there is cross-functional communication and bring a systems perspective.