

The Reality of System Design Today: Do Theory and Practice Meet?

Grant Martin

Cadence Design Systems, Berkeley California

gmartin@cadence.com

Abstract

ACSD aims to be a forum "for disseminating advanced research results on theory and practice of design of concurrent systems". In this talk, we will look at the reality of system design as practiced in industry today and what the needs as expressed and demonstrated by designers actually are. Most industrially successful applications of system design concepts and tools are in distinct design niches, and generic methods and tools have not yet been found which apply to a wide range of systems. We will look at these niche applications, especially those driven by System-on-Chip design, and identify the barriers to wider adoption of known results and methods. In addition, we will try to anticipate the nature of future architectures for highly-concurrent systems, and speculate on the system design needs that may be exposed by them.

1. Extended Summary

Colloquially, a 'system' is defined as 'that level of design which is one level up from the one you are currently working at'. And indeed, whereas the design methods, tools and flows for IC and PCB design have become established enough over the years, and evolved, to reflect common design 'ideologies' – i.e. a consensus on various standard and widespread design approaches – 'system-level design' has remained a rather arcane hodge-podge of idiosyncratic design flows and tools; much talked about, but not so widely practiced [1].

System-level design remains the hope of the EDA industry as one of the primary vehicles with which it can break out of the prison of RTL to GDSII design [2]. But this will not be some kind of generic system-level design capabilities; it will be the identification of some compelling design problems which are of particular importance, and value, to a specific design domain.

With the shrinking EDA focus in 2001-2002 – centred on digital IC implementation flows and the '90 nanometer' process node, it is clear that application of system-level design to SoC may be the most fruitful area to concentrate on. Therefore, we must analyse the most likely application domains for system-level design of SoC and determine the fit of current and future tools and flows.

There are several relevant system-level use models: algorithmic design and implementation; the modelling and design of SoCs and SoC platforms at the system level; and the creation of SoC 'system virtual prototype' models for use in development of embedded software (ESW). This last task takes place on at least two levels of speed-accuracy tradeoff: slower and more accurate models for Hardware-dependent Software (HdS); faster, less accurate models for ESW applications.

Despite great hopes over the last 7-8 years of the SoC 'revolution', the development and application of system-level design methods and tools has not made significant progress. However, we may be on the verge of some significant change, driven in part by the complexity of multi-processor SoCs, the emergence of networks-on-chip and communications modeling and tradeoff analysis; the growing importance of design space exploration (DSE) and the need for more accuracy in ESW validation. In addition, the rise of reconfigurable, highly programmable platforms incorporating networks of fixed processors and reconfigurable logic may open up new opportunities for behavioural synthesis, although offered as a compiler optimisation in an embedded SW development flow rather than to HW design engineers.

2. References

- [1] Grant Martin, "The Future of High-Level Modelling and System Level Design: Some Possible Methodology Scenarios", *Electronic Design Process Workshop*, Monterey, April 2002. Session 1.
- [2] Gabe Moretti, "System level design merits a closer look", *EDN Europe*, February 2002, pp. 22-28.