The 2005 International Technology Roadmap for Semiconductors (ITRS) chapter discusses a major system design driver - the power efficient SoC (SoC-PE) - and predicts its future evolution. The SoC-PE is a “power-efficient, consumer-driven, possibly wireless device with multimedia processing capabilities, based in part on the model created by the Japan Semiconductor Technology Roadmap Design Working Group.” As such, its basic architecture “will feature a highly parallel architecture, and consist of a main processor, a number of PEs (processing engines), peripherals, and memory.” A PE is “a processor customized for a specific function.” The ITRS predictions for the number of PEs in such an SoC ranges from 16 in 2005, 23 in 2006, 79 by 2010, 268 by 2015, to 878 in 2020.

Even if this represents a maximum complexity for a typical SoC-PE, one can see a rapid compounded growth in the number of processors. Already in 2006 we see many SoCs in many product categories with four or more processors, and announcements made over the last couple of years include examples with up to 192 processors or more.

We can conclude that a major evolution in SoC design methodology is happening: the rise of processor-centric design. Multiprocessor system on chip (MPSoC) has become common, and we also see a trend to multicore processor designs on the desktop and in large servers.

This new methodology places processors, especially application-specific instruction set processors (ASIP) at the center of complex SoCs, and seeks to use them in place of RTL hardware blocks wherever possible. Benefits include maximizing reuse, minimizing design time and effort and lowering product risk. But this new design style cannot be done with old methodologies and design tools.

There are several key issues in designing complex MPSoCs. First and foremost among them is the need for better programming models that allow designers to cope with the huge amount of concurrency in these designs, and to overcome their natural “fear of concurrency.” In addition, architecting an MPSoC, especially with a choice of many embedded processors, and configurable/extensible processors as a strong technology option, requires new approaches to design space exploration, mapping applications, controlling and debugging them, and finding the optimal
When it comes to programming models, and developing application code that explicitly exposes the relevant concurrency, we are still in an era of more art than science, and research than a single de facto industrial practice. Several decades of research in automatic recognition of parallelism or threading in source code has yielded disappointing results. In addition, the kind of parallelism or threading found in large scientific and server applications is often not the same as those found in deeply embedded applications. Determining the most appropriate kind of concurrency is part of the art of effective MPSoC design. Once that is found, a variety of programming library APIs are available to assist in mapping application code to a particular MPSoC architecture – but new models and new APIs are no doubt waiting to be discovered or developed.

Determining the right MPSoC architecture on which to map the applications is an area for which a growing number of academic research projects and commercial EDA and electronic system level (ESL) tools, and IP companies are offering capabilities. These range from models of major IP components - processors, memories, buses and other blocks – to models of complete platforms – to tools that allow the design space exploration of many different architectural and IP choices. When IP can be configured or matched closely to the end application, as, for example, configurable processors, an extra level of tool and model capability is important – often offered by the IP vendor. Once the design space has been suitably explored and an architecture and choice if IP established, fast simulation models are extremely useful to provide software and systems engineers a verification environment that provides sufficient accuracy for software verification under normal operating conditions, while greatly accelerating model speed.

Many of the ideas in this area build on roots dating back to the 1990s, but have had considerable technology development for several years. In addition, although the simpler systems of the mid to late 1990s may not have found advanced modeling and exploration essential, the complexities of today’s MPSoC systems are beginning to make use of ESL tools and abstract models much more essential for an effective design process.

The 43rd Design Automation Conference (DAC) 2006 has several interesting sessions dealing with MPSoC and processor-centric design and verification, including ones on processor and communication centric SoC design, MPSoC design methodologies and applications, advanced topics in processor and system verification, and a special session on MPSoC design tools. Some of the topics covered in these sessions include exploration of memory hierarchies, fault-tolerance in MPSoC, applications including biomedical, RFID and multimedia, networks-on-chip, and a number of specialized verification methods. The DAC exhibit floor features both several IP companies, and a host of new and old, large and small EDA companies with tools and models for MPSoC design, integration and verification. Many of the other topics at DAC will also be of interest to MPSoC designers. A visit to DAC will be worthwhile for everyone involved in MPSoC. For the full DAC event schedule and exhibitor listing, click here.

Grant Martin is chief scientist at Tensilica Inc. and a co-chair of the DAC 2006 Technical Program Committee, Design Methods.