



General-purpose versus application-specific processors

PRADIP BOSE

pbose@us.ibm.com

..... The world of special-purpose or application-specific processors has traditionally been quite different from that of general-purpose microprocessors. Sustained high performance across a broad suite of general-user applications is usually the key requirement in the design of general-purpose processor cores. These are therefore the basic processing engines of choice in desktop and server products. In the world of embedded-processor systems that are geared to solve a single application (or a limited *class* of applications) very efficiently, the processor core of choice has traditionally been quite different—one that uses a much simpler microarchitecture, well-matched to the characteristics of the targeted application(s).

Simplicity derives from precise knowledge of the specific application focus and from the constraints of cost and power consumption that are usual in such markets. For example, a DSP core that is part of a mobile phone product must necessarily be very low power, and the cost must be much less than that of a microprocessor that powers a desktop. Yet, for the very specific application that the DSP engine is designed to handle, the performance must be acceptably high. For its very specific application domain, the DSP core would usually exhibit much superior power-performance efficiency (measured in MIPS per watt) than a general-purpose microprocessor.

Design cost pressures also dictate a

much simpler design methodology in application-specific cores and systems, one that uses a lot more automation in terms of synthesis, placement, verification, and routing. Use of a technology-specific design cell library (instead of customized macros) typically implies lower performance targets than customized, high-performance, general-purpose cores built using similar CMOS technology. However, time to market, design cost, power, and frequency (performance) expectations have traditionally *all* been significantly lower in such application-specific processor markets (compared to general desktop and server domains). Hence, the two worlds have naturally remained distinct, with very different design cultures.

The relentless scaling of CMOS technology has fueled the growth in transistor densities and functionality in both worlds, with increased levels of on-chip integration. Embedded-processor systems have witnessed the integration of heterogeneous processing cores on single chips and modules to enable expanded functionality.

The underlying semiconductor technologies that fuel the two domains do not appear to be that different any more, and the complexity growth in the power-managed, high-performance general-processor is demanding the use of greater design automation and reusable macros (and even cores) in the new generation of chip design. Also, certain segments of the application-

specific embedded-processor systems are witnessing a huge increase in the demand for performance. The domain of set-top and stand-alone game systems is a specific and notable example.

Technological and application growth trends are thus causing a rethinking within the processor design community in general. Does the industry need (or, perhaps more importantly, can the basic economics of chip design sustain) two very different processor design cultures in the near future? Can the high-performance, general-purpose world build its processing engines out of commodity cores, using a design methodology that is largely borrowed from the embedded, SoC world? Will the trend towards multicore, lower-frequency designs in the general-purpose computing world hasten the merging of the technologies, design methodologies, and tools in the two worlds? In summary, is the industry headed toward an era of unification, where designers integrate heterogeneous cores, storage, and communication elements onto a single chip to build the basic processing engine within a system, irrespective of the target market?

Even if such unification, driven by technology and design complexity constraints, is not on the immediate horizon, it is clear that the world of application-specific, commoditized processor designs will become more pervasive and will influence everyone's daily lives in an ever-increasing manner.