MultiCore Designs
By Hem Hingarh, VP, Synapse Design

Over many years and based upon Moore's Law, transistor counts have doubled approximately every 24 months as features increase and semiconductor dies grow. This has led to performance increases of 1000x over 20 years with microarchitecture advances and faster transistors.

With the technology, developers obtained performance increases through processor upgrades, clock frequency boosts and microarchitecture improvements. However, processor designs ran into power and performance limits of single core processors, focus shifted to multicore designs.

There are three basic classes of multicore architectures: homogeneous multicore (with same ISA) with shared memory (PCs & Servers); heterogeneous multicore (different ISA) with mix of shared and non-shared memory; and homogeneous multicore (same ISA) with non-shared memory.

With the shift to multicore, responsibility for performance gains has moved into software and away from technology scaling. New supporting software libraries, development tools and technologies allow skilled software engineers to continue to improve performance.

Many companies are adopting an incremental approach to enable multicore processors based on their business needs.

Trends

While the PC market is slowing, the focus has shifted to embedded multicore processors in mobile, automotive and communication markets with some of the cores implemented as accelerators for various algorithms.

Hardware trends have led to scalable on-chip multiprocessors, lots of inter-processor communication, fast responsive programmable input-output channels and various mixed signal blocks. A variety of algorithms and techniques requiring predictable execution (flat

memory hierarchy; predictable caches) and multithreading with low synchronization overhead.

Additionally, a low-power/energy-efficient multicore market has evolved to support the “Internet of Things,” including wearables with support for computing in clouds.

Some of applications require high instantaneous performance (vision, automotive requires embedded intelligence). They need to be real-time, secure, highly reliable and energy/power efficient, available in a small form factor and easy to use. For the large consumer market where lower cost is fundamental, we must rethink the architecture and build SOCs like memories with many repeated regular blocks with behavior defined by software.

For improved energy efficiency, asynchronous techniques are being used at block levels. However, overall SOC performance may be limited by data movement more than by processing engines requiring scalable and predictable on chip interconnect.

On Chip Interconnect

Until the 1980s, computing has been expensive and communication cheap but with continuous technology scaling this has changed. The high-density of the components in the SoC complicate the design and implementation of a shared bus architecture. In today’s System-on-Chip (SoC) that include as many as 300 -500 different IP blocks, fast communication between different IPs is implemented using a well-structured design approach called Network-on-Chip (NoC).

A design methodology is needed in order to make efficient use of all on chip the resources, with programming models and predictable behavior. The basic performance parameters of NoC are latency, bandwidth and jitter. The basic cost factors are power consumption and area usage.

Designing energy efficient methodologies for various NoC domains, such as the routing algorithms, buffered and buffer-less router architectures, fault tolerance, switching techniques, voltage islands, and voltage-frequency scaling significantly affects the NoC

performance. Therefore, the optimization of routing algorithms for the NoC is a key concern in enhancing the NoC performance and in order to minimize the energy consumption.

EDA Tools and Trends

For the embedded consumer market where lower cost is key volume driver, we will need various innovations occurring in multiple areas such as process technology, chip architectures and software, and the need for improvement in SOC implementation is a must.

EDA innovations are key part of the equation. We need:

- Improvement in design productivity (design cycle reduction for design, verification, layout and silicon validation), to support multi-billion transistor designs.
- Reduction in power consumption (fast on/off mixed signal designs, leakage power verification etc)
- Seamless hardware/software co-design (block modeling, virtual prototyping, compiler performance validation etc)
- System level integration

To see improvement in areas listed above requires improvement in EDA tools and methodology.