POINT OF VIEW

TEXAS INSTRUMENTS

When we come to a fork in the road…

By
Ray Simar,
TI Fellow

About the author

Ray Simar is currently responsible for enhancing Texas Instruments digital signal processing solutions by developing advanced architectures for diverse applications. Simar is also a TI Fellow for his pioneering work on DSP technology.

We must change not just what we do but how we do things as well. It is only by clearly understanding the state we are in and how we got here that we can begin to formulate a secure path toward the future.

Legacy processors have run their course. As layers of instructions and hardware pile up, locked into the architecture by the golden handcuffs of backward compatibility and bloated by numerous extensions, these architectures are becoming increasingly complex and difficult to verify.

The inability of future process technology nodes to deliver performance improvement through clock-rate increases will necessitate parallelism enabled by higher degrees of integration made possible by shrinking feature sizes, the use of stacked die and the eventual use of photonics. Designs and architectures that ignore the impact of the physics of interconnect will become laden with power and area consuming crossbars and large synchronous clock networks that will eventually fail.

The emergence of a single multiprocessor programming standard is unlikely and unnecessary. Experience and research have shown that any number of common programming languages, with the addition of a few key multiprocessing constructs and a good performance analysis/program debugging environment, can form the basis for allowing users to effectively program systems built from large numbers of processors.

It is important to realize that there will be no silver bullet to solve the problem of automatically partitioning user code. Instead, effort should focus on tools that help the programmer understand the performance and power implications due to their partitioning of the application and support the debugging of this partitioned application.

As power continues to become a greater priority, development tools must support optimization for power just as they optimize for performance and code size. With a power profiling simulator customers could, for the first time, make appropriate tradeoffs in terms of power and performance as they write their applications. This foundation also enables power-based compiler optimizations and emulation analysis.