Graphical System Design

Rahman Jamal
National Instruments Germany

In today’s economy, products have to reach the market faster and faster, forcing a compression of the design cycle. At the same time, silicon gate costs continue to decline as densities increase. So, heterogeneous devices with multiple processors and FPGAs are becoming more common, resulting in design with greater complexity and longer development cycles.

However, the traditional tools have only delivered on incremental improvements over the last decade. For years, designers have hoped that approaching product design from a more software-oriented perspective would improve productivity. But any improvement so far has been modest at best, due mostly to increased computer performance. Traditional software development tools haven't improved productivity at anywhere near the pace that design complexity is increasing, let alone enough to shorten the design cycle.

Software development based on a textual sequential programming language may meet the requirements of a single small processor system, but in general it is weak in exploiting phenomena such as parallelism. Sequential paradigms try to overcome this limitation by adding threads for parallelism. This however greatly compromises the semantics of the language and dramatically increases complexity leading to a poor model of implementation for highly parallel devices such as FPGAs. The well-known Von Neumann model, which even today serves as the foundation of sequential programming languages, simply is inappropriate for parallel, distributed, heterogeneous devices.

A big jump in productivity improvement can only be achieved through an implementation on a higher level of abstraction. This level must naturally represent parallelism including timing and synchronization. Moreover it must be equally appropriate for implementation on processors and FPGAs in systems where they are tightly coupled as well as in a distributed application.

One way of addressing these needs is the use of graphical paradigms. Experience shows how much the graphical user interface has made computers more accessible and users more productive. The same benefits are available to designers using graphical design methodologies. Graphical dataflow programming lets engineers and domain experts rapidly develop and iterate designs, reducing the time from idea to prototype. Graphical programming also is a highly interactive and responsive process that facilitates greater exploration, leading to a more optimized design.

Ideally, one would work at a higher level of abstraction for the entire design. But it may be necessary to gain access to finer-grained details for the critical portions of the design that require the highest performance. Graphical programming is well suited to represent system components at different levels of abstraction.

Dataflow represents a very powerful model of computation. Like any text-based programming language, it is Turing-complete, yet being more flexible and inherently parallel. Furthermore

it represents a natural model for distributed systems. With the addition of a timed-loop structure, it can elegantly depict distributed multirate systems.

Graphical tools typically include a rich library of user interface components for displaying-data in many forms. Being able to probe and modify design parameters interactively through the user interface while the model runs is a very effective way to develop intuition about the design and explore alternatives quickly.

The unmatched combination of an interactive front-panel (user interface) and graphically structured dataflow diagrams (the actual control program) has led to an exponential growth in productivity. The continued refinement of graphical design tools, supported by computers with high-speed graphics engines, will be how design engineers achieve the high productivity levels required to reduce time-to-market with more complex designs in our highly competitive world leading to a dramatic decrease in overall design costs.

Rahman Jamal is Technical & Marketing Director National Instruments Germany.