Understanding a Key Trend in Embedded System Design

Combining Multiple Processing Elements in Heterogeneous Architectures Enables High-Performance Embedded Systems

As embedded applications grow in complexity, hardware architectures and embedded system design tools must evolve to address increasingly demanding requirements as well as minimize design time. Historically, many embedded systems have featured a single CPU, so system designers have relied on CPU clock speed improvements, the shift to multicore computing, and other innovations to achieve the processing throughput required by complex applications. However, more and more system designers are migrating to computing architectures featuring multiple distinct processing elements. Heterogeneous architectures, to provide a more optimal balance between throughput, latency, flexibility, cost, and other factors. This balances the next generation of embedded system designs.

To illustrate some of the benefits that heterogeneous computing architectures can provide, consider an architecture composed of a CPU, an FPGA, and I/O. FPGAs are ideally suited to handle parallel computations such as signal processing operations on a large number of parallel data channels. Additionally, since FPGAs implement computations directly in hardware, they provide a low-latency path for tasks like custom triggering and high-speed, closed-loop control. Third, incorporating FPGAs into computing architectures also improves the flexibility of embedded systems, making them easier to upgrade than systems with fixed logic and enabling them to adapt to changing I/O requirements. Coupling a CPU and an FPGA in the same heterogeneous architecture means that system designers do not need to choose between these FPGA advantages and the corresponding strengths of a CPU. Additionally, a heterogeneous architecture can be more optimal than attempting to adapt a single-element solution to a problem that the element is not well suited for. For example, a single FPGA might handle a parallel task requiring low latency equally as well as a large number of CPUs.

Heterogeneous Computing Architecture

Combining a microprocessor and an FPGA in a heterogeneous computing architecture makes it possible for embedded system designers to use the strengths of each processing element and more optimally meet complex application requirements.

While embedded system designs featuring multiple processing elements have many advantages, they raise some challenges when it comes to software development. Large design teams are often a necessity due to the specialized architectures of individual processing elements and the fragmented set of tools and expertise required to program them. For example, FPGA programming commonly requires knowledge of VHDL programming—a skill that can require a significant training investment, larger staff, or costly outsourcing. Additionally, developing the software stack to support a heterogeneous architecture is a considerable undertaking that involves driver integration, board support, middleware for inter-element communication, I/O interface logic, and more. System designers can address these challenges with an integrated hardware and software platform composed of a standard heterogeneous architecture, interchangeable I/O, and high-level system design software. Building on knowledge of the underlying hardware, high-level design tools abstract both the system architecture and I/O during the development process, improving productivity and reducing the need for system designers to manage low-level implementation details. When developing embedded systems based on heterogeneous architectures, high-level system design tools are capable of abstracting the architectures of individual computing elements such as FPGAs, and providing a unified programming model that embedded system designers can use as they take advantage of the capabilities of different elements. Furthermore, abstraction in high-level design software aids in the concise description of functional behavior and facilitates code reuse despite changes in hardware or communication interfaces.

The LabVIEW Reconfigurable I/O (RIO) Architecture

Off-the-shelf embedded system platforms based on heterogeneous architectures are available today, and can eliminate the need to design custom hardware. One example is the NI LabVIEW RIO architecture, which combines LabVIEW system design software and NI RIO hardware (based on processors, FPGAs, and modular I/O) and is available in a variety of form factors and performance levels spanning from board-level NI Single-Board RIO to industrially packaged NI CompactRIO and PXI. A broad ecosystem of I/O modules including analog and digital measurements, industrial bus connectivity, and more helps engineers use these platforms across a range of applications from power electronics control to medical imaging.

LabVIEW makes it possible to program CPUs and FPGAs on heterogeneous NI hardware using a consistent graphical programming model. In addition, LabVIEW abstracts system timing, I/O access, and inter-element communication based on knowledge of the underlying architecture. The result is that these heterogeneous architectures are both more accessible and cost-effective to use for embedded system designs. Domain experts without knowledge in specialized implementation tools can play a larger part in the implementation of their ideas, and system designers can get to market faster with a highly productive platform-based design approach.

Heterogeneous Computing in Action

Heterogeneous architectures in embedded system design are already impacting many industries. One example comes from Thales UK, a transportation solutions company, which designed an automated signaling system with several virtual test trains based on LabVIEW and CompactRIO. The goal of the signaling project was to boost system capacity by 33 percent (the equivalent of an extra 5,000 passengers per hour and cut journey times by 22 percent.)
Engineers at Thales UK simulate a train on London’s underground rail network with CompactRIO and LabVIEW.

To mimic an actual passenger train and test railway conditions, the team at Thales UK elected to use a combination of an FPGA and embedded processors to provide high-speed control, data acquisition, and data analysis in a single system. Using off-the-shelf CompactRIO hardware and LabVIEW, the company implemented a complete control and monitoring system capable of simulating signals to imitate a real-world train, measuring test data, and logging information for later review. Not only was Thales UK able to innovate on traditional testing methods, the company also reported a reduction in test time from days to hours. And tests using the new virtual test trains require about half the manpower to operate.

Design teams such as the group at Thales UK are using heterogeneous computing architectures to deliver innovative, high-performance embedded systems. Aided by high-level system design tools and off-the-shelf hardware, small teams and domain experts can take advantage of these architectures without building custom hardware or learning multiple, fragmented design tools as has traditionally been required to program various processing elements. Ultimately, as heterogeneous architectures continue to play a growing role in embedded system design, the organizations and teams using them will benefit from the capability, performance, and flexibility they provide.

Carlos Pazos
Embedded Software Product Marketing Engineer

This article first appeared in the Q1 2013 issue of Instrumentation Newsletter.

Legal
This material is protected under the copyright laws of the U.S. and other countries and any uses not in conformity with the copyright laws are prohibited, including but not limited to reproduction, downloading, duplication, adaptation and transmission or broadcast by any media, devices or processes.