

A Reconfigurable Computing Architecture Utilizing a Switch Fabric Network

P. Rudolph¹, E Bentley¹, W. Turri¹, K. Hill²

¹Systran Federal Corporation, Dayton OH

²AFRL/IFTA, WPAFB, OH

Introduction

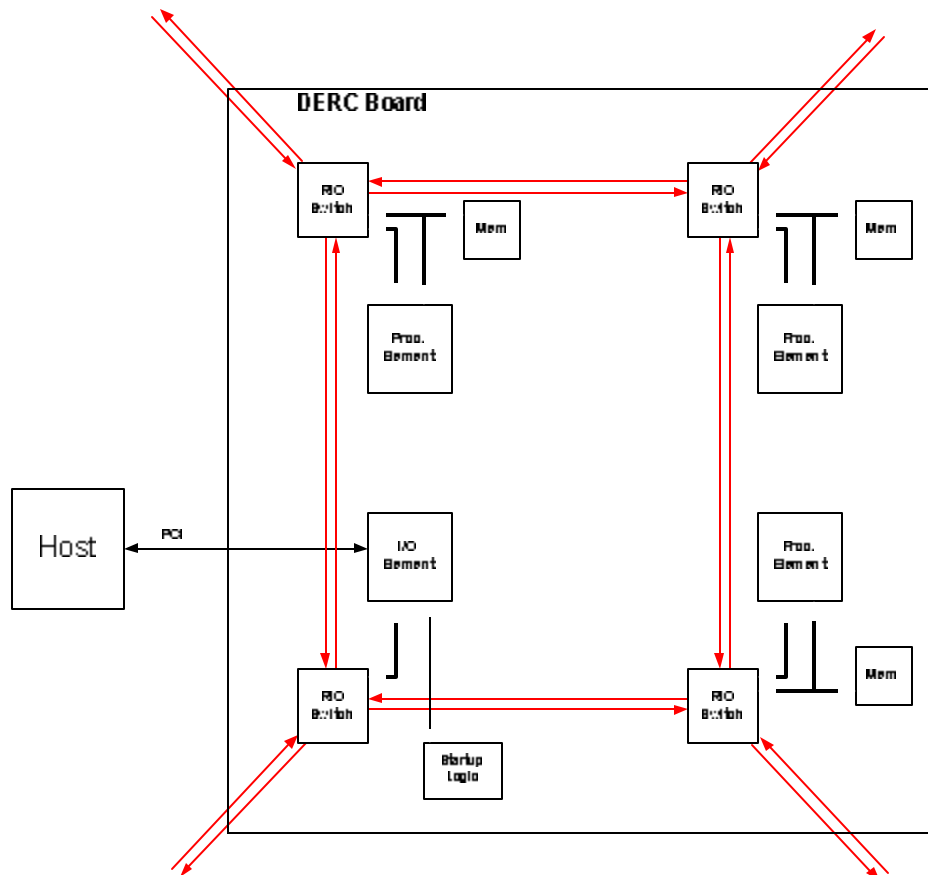
This paper describes a novel hardware architecture that combines Xilinx FPGAs with a RapidIO switch fabric network to create a highly flexible, general-purpose reconfigurable computing platform. This architecture provides a variety of benefits, including high-speed communication between processing elements, multi-board scalability, the ability to support multiple applications, and a basis for fault tolerance.

This architecture was developed as part of the Development Environment for Reconfigurable Computing (DERC) project, a Phase II SBIR contract. The project also includes the development of software and logic libraries and software drivers to support the creation of custom reconfigurable computing applications. This work is being funded by the Air Force Research Laboratory Information Directorate, Embedded Information Systems Engineering Group.

References

- RapidIO Interconnect Specification, Rev. 1.2
- RapidIO Technical White Paper, Ver. 3.

Architecture



The DERC board consists of multiple nodes, which are connected with each other via a RapidIO switch fabric. Nodes are devoted to either I/O operations (I/O nodes) or data processing (Processing Nodes). Figure 1 shows the architecture of the prototype board, which uses the PCI form factor. As this figure indicates, this board includes four nodes: three Processing Nodes and one I/O node. The paired arrows (in red) represent RapidIO links.

Every node includes a RapidIO switch, also implemented using Xilinx FPGAs. Each Processing Node also includes a Processing Element, consisting of a Virtex-II FPGA (the prototype uses XC2V3000 parts) and a Memory Element, consisting of 128 MB of SDRAM (expandable in future versions). The Processing Element contains the application developer's custom logic. In the prototype, the Processing Elements are XC2V3000 FPGAs, but the board can accept larger FPGAs, up to a XC2V10000. The I/O node includes an I/O Element (also an FPGA), which acts as a bridge to any non-RapidIO interface (for the prototype, this is the PCI interface).

Via the RapidIO switch fabric network, the Processing Elements can exchange data with each other, with the Memory Elements, with external devices, and with the system host. The DERC architecture is highly scalable; multiple DERC boards can be connected together using the switch fabric network, and can be treated as a single processing resource by the microprocessor host.

Benefits

The benefits of this architecture include the following:

- **Performance** – The use of advanced FPGAs allows implementation of large and complex user logic, the large Memory Elements allow large data sets to be processed, and the high speed of the RapidIO network (1 GByte/sec *per link* minimum) will reduce communication bottlenecks between nodes.
- **Scalability** – Multiple boards can be linked together to create a large-scale reconfigurable computing resource. The RapidIO network will also allow direct connections to other hardware components (I/O devices, mass storage, etc.).
- **Multiple Applications** – Because each node can operate independently, this architecture can easily support multiple applications operating on the same hardware platform. Several applications may also share a single node, or one application may utilize several nodes.
- **Fault Tolerance** – The mesh topology of this RapidIO implementation means that there are multiple potential paths between any two nodes. Therefore, it would often be possible to re-route data around a damaged portion of the network. Further, since all processing nodes are identical and reconfigurable, it would be possible to shift the functionality of a damaged or corrupted node to an undamaged one.

Results & Conclusions

This section will describe the implementation of a sample user application, and will include measures of performance of the application and architecture.