Flexible Architecture for FPGAs in Embedded Systems

## A small device simplifies FPGA development in cPCI systems.

NASA's Jet Propulsion Laboratory, Pasadena, California

Commonly, field-programmable gate arrays (FPGAs) being developed in cPCI embedded systems include the bus interface in the FPGA. This complicates the development because the interface is complicated and requires a lot of development time and FPGA resources. In addition, flight qualification requires a substantial amount of time be devoted to just this interface.

Another complication of putting the cPCI interface into the FPGA being developed is that configuration information loaded into the device by the cPCI microprocessor is lost when a new bit file is loaded, requiring cumbersome operations to return the system to an operational state.

Finally, SRAM-based FPGAs are typically programmed via specialized cables and software, with programming files being loaded either directly into the FPGA, or into PROM devices. This can be cumbersome when doing FPGA development in an embedded environment, and does not have an easy path to flight. Currently, FPGAs used in space applications are usually programmed via multiple space-qualified PROM devices that are physically large and require extra circuitry (typically including
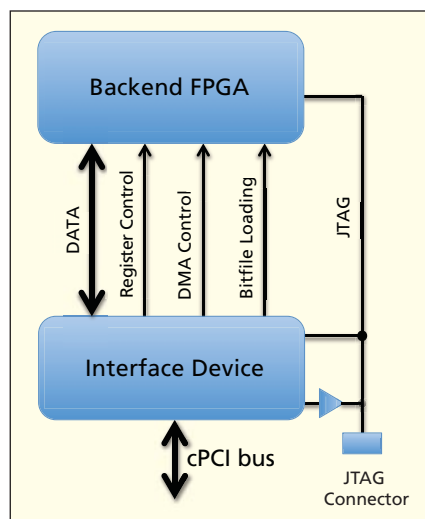
The **cPCI Interface** is a common interface between the cPCI bus and the backend FPGA. It is implemented as a separate interface device on the cPCI bus.

a separate one-time programmable FPGA) to enable them to be used for this application.

This technology adds a cPCI interface device with a simple, flexible, high-performance backend interface supporting multiple backend FPGAs. It includes a mechanism for programming the FPGAs

directly via the microprocessor in the embedded system, eliminating specialized hardware, software, and PROM devices and their associated circuitry. It has a direct path to flight, and no extra hardware and minimal software are required to support reprogramming in flight. The device added is currently a small FPGA, but an advantage of this technology is that the design of the device does not change, regardless of the application in which it is being used. This means that it needs to be qualified for flight only once, and is suitable for one-time programmable devices or an application specific integrated circuit (ASIC). An application programming interface (API) further reduces the development time needed to use the interface device in a system.