

FPGA Modules with an Integrated Processor Drive Real-Time Applications

Engineers developing DSP and high speed logic applications are now well-aware that FPGAs can help them create an integrated, sophisticated solution. The availability of commercial off-the-shelf (COTS) [FPGA boards](#) can make these solutions viable and do so in reduced development times. Today, with systems architected to perform extremely time-critical tasks on a COTS FPGA module, the host CPU is often relegated to managing the flow of processed data to and from the FPGA module across a PCI-X bus, PCIe, Serial RapidIO, or other data interface. This data transfer is usually necessary because some processing or data storage activities are shared between the FPGA module and host CPU. Must these activities be shared? Could more processing and management of the data be performed in one place? If the application has high-speed requirements and more of the slower data management or calculation tasks can be lifted from the host CPU, then a COTS FPGA module with an integrated processor might be the solution.

FPGAs, by design, are massively parallel processing devices. Processes such as image processing, where a large amount of data has to be manipulated and filtered simultaneously, can take advantage of this parallel processing capability. The language developed to process this information, VHDL, is well-suited to create the parallel processes necessary. On the other hand, CPUs are designed to process complex, but sequential, configurable data manipulations. Mixed resolution matrix processes used to rotate an image are one example best handled by a CPU. Coding for the CPU process is usually written in one of the popular high level languages (HLL) such as C++. When both the FPGA and the processor appeared to have a natural fit within the application, the norm was to marry the two devices in an attempt to optimize the application's data processing. The problem that arose from this approach was the constant and time-consuming task of passing sizable chunks of data between the FPGA and processor. This problem can be minimized, or even eliminated, if the two components can be united in a single, integrated design

So, how does a design engineer gain access to the computational and data management conveniences afforded by a processor within an FPGA? Two methods have evolved:

1. When sufficient logic cells are available in the FPGA device to accommodate a soft CPU core (e.g. Xilinx® MicroBlaze™ or PicoBlaze™), it is possible to load the IP core, a minimum kernel of the desired RTOS, and application code in the HLL of choice. The application logic executes within the fabric of the FPGA device, and information is exchanged between the HLL-coded application and the FPGA/DSP logic components through FIFO, block RAM allocations, or other mechanisms. Since FPGAs are such high-performance engines, as long as overhead and interface complexity are kept low, they will have limited effect on overall performance. Implementing a soft-core processor is a popular approach often used on Acromag's PMC-LX60 and PMC-VLX modules, or on the PMC-VSX95 when high DSP block usage is required.
2. For applications where a PowerPC is a pre-requisite or time is of the essence, using an FPGA with an embedded processor such as the Xilinx XC5VFX70T may prove to be the optimal solution. In this case, Acromag's new PMC-VFX70 – with a hard-core PPC440 processor tightly integrated into the FPGA – becomes the module of choice. In this implementation, it is only necessary to load the minimum RTOS kernel and the HLL application code. Logic cells are not occupied by the PPC440 implementation. The PPC440 functions independently, waiting only to exchange commands and information with the FPGA through mechanisms provided by Xilinx or devised by the system architect.

There are many benefits to deploying a processor within an FPGA module. First, there is the ability to capture, process, and manage information on the same device (the FPGA module) and in the language implementation best-suited to the task. Typically, VHDL is used for the FPGA and perhaps MATLAB® for the DSP components. The alternative is to use an HLL atop an RTOS for data computation, exchange, and management.

A second benefit is that an FPGA with embedded processor offloads many RTOS tasks which are normally executed on the system's host CPU. Moving compute bound data analysis or data management tasks off the host CPU and performing these operations on the processor-equipped FPGA module reduces the host CPU load. This move permits other CPU tasks to run faster and/or more frequently. In some situations, by shifting tasks off the host CPU and onto the FPGA module it may be possible, even desirable, to downsize the host CPU.

Additionally, by moving more of the host CPU tasks to the FPGA module, there is a tremendous reduction in backplane bus traffic. This reduction significantly increases overall system performance. Since more processing is executed directly on the FPGA module, less information requires transfer from the data acquisition module (the PMC FPGA module) to the host CPU for additional processing and vice versa.

The benefits of embedding a processor on an FPGA module have proven substantial in many time-critical applications. We are aware of several recent applications which have migrated to a single COTS FPGA module solution. With careful planning, perhaps there is an application for an FPGA module with integrated processor in your future?

For additional assistance contact your local [Acromag representative](#) or the factory for answers to your COTS FPGA applications.

Acromag is an ISO 9001 and AS9100 certified manufacturer.