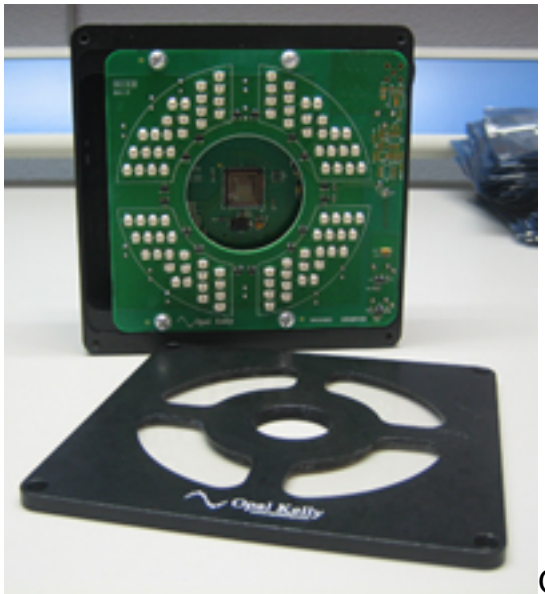
Bridging the Worlds of Hardware and Software with USB-based FPGA

Jake Janovetz, President, Opal Kelly

Field-Programmable Gate Arrays (FPGA) are the most flexible and capable devices available to interface to a large variety of system devices such as memories, data acquisition (ADC/DAC), image sensors, displays, and so on. Often, the goal is to interface these system devices to a software infrastructure running on a workstation platform using a Windows, Linux or Mac operating system.

The ubiquitous PC, with its myriad of software development environments, comfortable presence, and low-cost, should be the ultimate microprocessor development platform but it has generally been ill-suited to the intimate hardware development scene. This is largely due to the layers of operating system between the application software and the device pins a developer wishes to "wiggle." Development engineers in commercial design and academic research have found a way to bring these two worlds together fortuitously with USB-based FPGA modules, designed to bridge the gap between hardware and software worlds.



Off-the-shelf modules, with software interface and API, are available to fully abstract the FPGA/PC interface. Using these modules makes this challenge of the system a "non-issue" for designers, so that they can

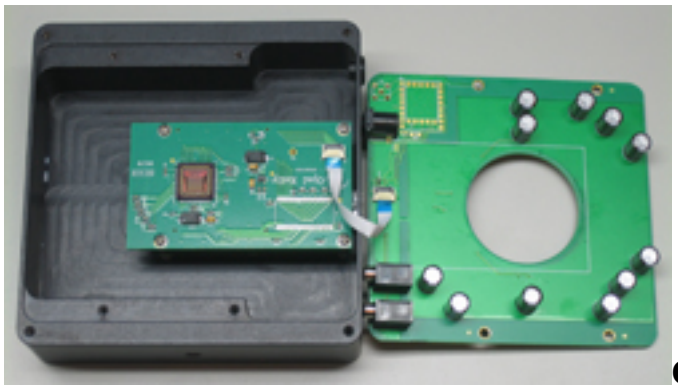
concentrate on the FPGA's HDL or the PC's software. This abstraction gives a standard PC ready access to the hardware realm. With the particulars of the FPGA/microprocessor interface out of the picture, designers are also able to migrate aspects of the design across this thin barrier to the more appropriate component: the microprocessor programmed in C and other traditional, high-level languages, or the FPGA programmed using HDLs with a more intimate interface to external hardware.

Mating IC Digital Data Output to Data Processing Software

A typical problem facing a system engineer is to mate a high-speed integrated circuit outputting digital data on physical pins to a specialized suite of software being used to process the data. This problem can be generalized to both a production environment, where a final product is being delivered, or to the prototyping workbench where designs need to be tested. In both cases, engineers are challenged with perpetual time-to-market demands and need to focus their efforts on core development (or their specific product development) rather than the PC-to-device interconnect. For the purposes of this discussion, let's consider the specific case of a machine-vision application being deployed at a few hundred locations on a busy factory floor.

The engineer has a high-speed CMOS image sensor capable of capturing frames based on an external hardware trigger from the manufacturing line. These images must then be conveyed to software that is used to perform a number of transformations on the image and, ultimately, make a quality judgment as feedback to the manufacturing line. Due to the low volume and time constraints, the image processing is best performed on a PC rather than the FPGA and may be done in a very high-level environment such as Matlab or LabView. On the other hand, volumes are high enough to justify some custom hardware rather than relying on expensive off-the-shelf systems.

Typically the hardware engineer would need to develop the entire link from the image sensor to the PC's USB, including programming the firmware on a USB microcontroller and writing drivers on the PC platform to allow her user application to communicate with the device. Finally, a protocol would need to be developed to transfer data from the FPGA to the PC application in a way that makes sense for both the hardware and software. All of this requires additional research and development, distracting the hardware engineer from the expertise she already has and the project at hand.



Off-the-shelf USB-based FPGA Module

and Software API
The off-the-shelf Opal Kelly USB-based FPGA module with FrontPanel Software API can integrate the custom hardware for image capture to the commodity microprocessor (PC) at a very low cost and with minimal development effort required. The FPGA-based reconfigurable hardware and a PC-based software programming interface require no USB expertise to complete the link. Both regimes are allowed to focus where their strengths lie. The FPGA's reconfigurable hardware is ideally-suited to the task of talking with the image sensor and the PC's rich development environment is perfect for image analysis with minimal engineering.

With the FPGA-USB module, the effort of interfacing the FPGA to the PC's microprocessor becomes a non-issue, since this interface has been abstracted by the manufacturer. The FPGA designer gets a handful of small HDL modules to plug into her design at key points where control signals or bulk data are to be transferred. The PC software engineer gets a complete application programmer's interface (API) so he is essentially making the FPGA and added hardware an extension of his software application.

Reduces Cost and Time of Engineering Development
The benefits for a company and its engineers are clear. Engineering resources can focus their efforts on the problems that are unique to the company (the specific manufacturing line requirements, in this case) rather than problems that have a general solution and might better be solved by off-the-shelf components (the USB-connected module and associated PC drivers).

This focus dramatically reduces the cost and time of engineering development for a particular problem. Additionally, the use of appropriate off-the-shelf components accelerate product development and clears engineering resources for other opportunities. One Opal Kelly customer estimates the Opal Kelly 'out-of-the-box' FPGA USB 2.0 module allowed them to focus on designing their core product and saved them over $100,000 while accelerating the time-to-market for its commercial product by 12 months.

Machine vision and its application to a manufacturing line was the example used above, but FPGA hardware and microprocessors have joint application in nearly every vertical market imaginable. Military applications often make use of microprocessor-based processing or networking combined with highly-specialized data acquisition. The nature of

parallelism in common cryptography and bioinformatics algorithms means that PC-based systems can benefit significantly from FPGA-based co-processing. And test equipment and many other industrial / scientific / medical applications require advanced machinery to be controlled or observed through a human interface, often a PC.

FPGAs and microprocessors are two great technologies that complement each other extremely well. The most common microprocessor "box," the PC (using Linux, Windows, or Mac OS), however, doesn't typically come with an FPGA available for engineering work. One of the most convenient and lowest-cost ways to marry these two technologies and reap the benefits of both is with an off-the-shelf USB-based FPGA module.

For more information, please visit www.opalkelly.com [1]

Source URL (retrieved on *12/17/2014 - 8:36pm*):
http://www.ecnmag.com/articles/2009/09/bridging-worlds-hardware-and-software-usb-based-fpga

Links:
[1] http://www.opalkelly.com/