

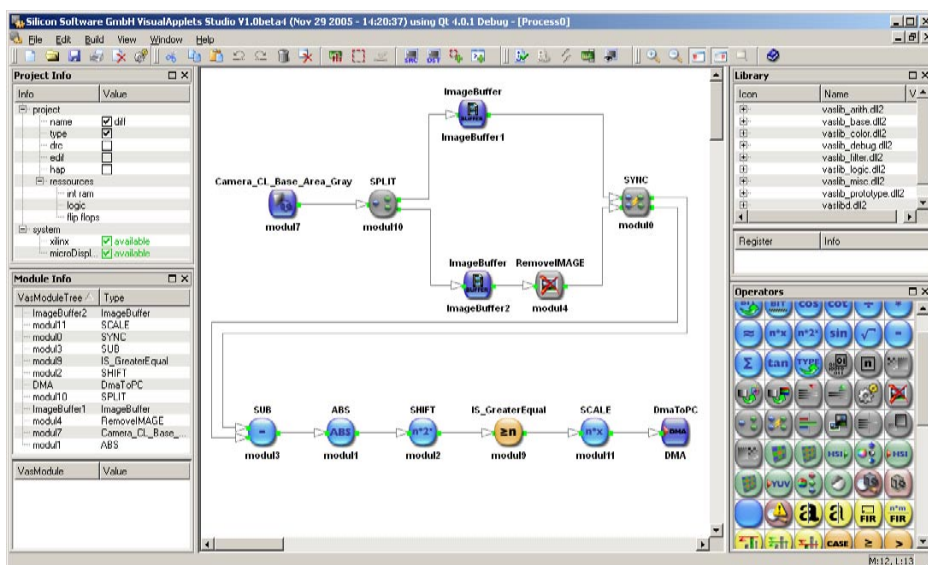
VISUALAPPLETS MAKE FPGA PROGRAMMING A SNAP

Although most every camera and frame grabber uses field-programmable gate arrays (FPGAs) to perform camera setup and embedded image-processing functions such as Bayer color conversion, the ability to program these devices is often hidden from the user. From a system integrator's perspective, however, leveraging the power of such devices can be very beneficial, since dataflow-like algorithms can often run at speeds hundreds of times faster than on any host PC.

One of the drawbacks of using these devices for image processing and machine vision has been the need for the developer to understand high-level languages such as VHDL, which, for nonprogrammers, make using the devices both expensive and slow. Of course, there have been companies that have attempted to create high-level dataflow-like languages that speed the development of FPGA based systems. One of the first of these was Databuc, which introduced its Visual CHIP Studio more than two years ago (see *Vision Systems Design*, September 2004, p. 87).

At the recent Vision show (Stuttgart, Germany; November 2005), SILICON SOFTWARE (Mannheim, Germany; www.silicon-software.com) introduced a graph-ically oriented interface called VISUALAPPLETS that lets developers program machine-vision and image-processing functions on the company's Xilinx (San Jose, CA, USA; www.xilinx.com) Spartan IIE-based microEnable III PCI CameraLink frame grabber. "VISUALAPPLETS is a tool for hardware programming of FPGAs based on graphical dataflows," says Klaus-Henning Noffz, managing director of SILICON SOFTWARE (see figure). "These dataflows are arranged using a combination of operators and filter modules and are compiled to a loadable hardware applet." These libraries contain operators for pixel manipulation, logical operators for classification tasks, and more complex modules for color processing and image compression.

Using these libraries, a number of image-processing functions such as look-up tables, thresholding, binarization, and counter functions can be intuitively placed on-screen in a pipelined fashion. Without any additional programming, the finished program is converted to functional blocks within the microEnable III's FPGA. "No programming is necessary to control the synchronization or timing of the dataflow within the FPGA," says Noffz. "The developer simply controls the complexity



Building an FPGA-based machine-vision system can now be done with a high-level graphical programming language called VISUALAPPLETS (top). In this example, the image from a single Camera Link camera is split and sequential images compared (right). The sum of absolute differences of these sequential images is displayed.



of the processing by allocating different processing resources or functions from the library."

SILICON SOFTWARE's synthesis and Xilinx's place and route software transparently convert the hardware design into an FPGA layout and, once installed, the system developer's program is automatically integrated, configured, and executed. After this process is complete, a hardware applet that describes the image-processing functions performed in the dataflow diagram is created. This can then be loaded into the company's microDisplay viewer and camera-configuration software.

At Vision 2005, SILICON SOFTWARE demonstrated the power of the software performing a sum of absolute difference (SAD) algorithm to compute image motion. Interestingly, this is the same algorithm used by Focus Robotics (Hudson, NH, USA; www.focusrobotics.com) to compute depth perception from two independent cameras (see *Vision Systems Design*, August 2005, p. 23).

In the dataflow architecture developed by SILICON SOFTWARE for motion analysis, data from a single Camera Link

camera is split into two adjacent buffers. Every other frame from the second image buffer is then removed and the two data paths synchronized so that each pixel in both images is correctly registered. After synchronization, the two images are subtracted and the result of the absolute value determined. This," says Noffz, "will result in an image that visually shows the motion between the image frames." The result is then shifted to the PC and displayed as a new image.

"To develop such an image-processing function using VISUALAPPLETS would take an integrator approximately five minutes," says Noffz, "greatly reducing FPGA development time over using traditional VHDL programming methods." With a price of around 5500 euros for both the microEnable III and the VISUALAPPLETS software, the software currently runs under Windows 2000/XP; a device driver from Windows XP64 is currently under development. —AW (Andy Wilson, VSD)

Vision Systems Design

January 2006

www.vision-systems.com