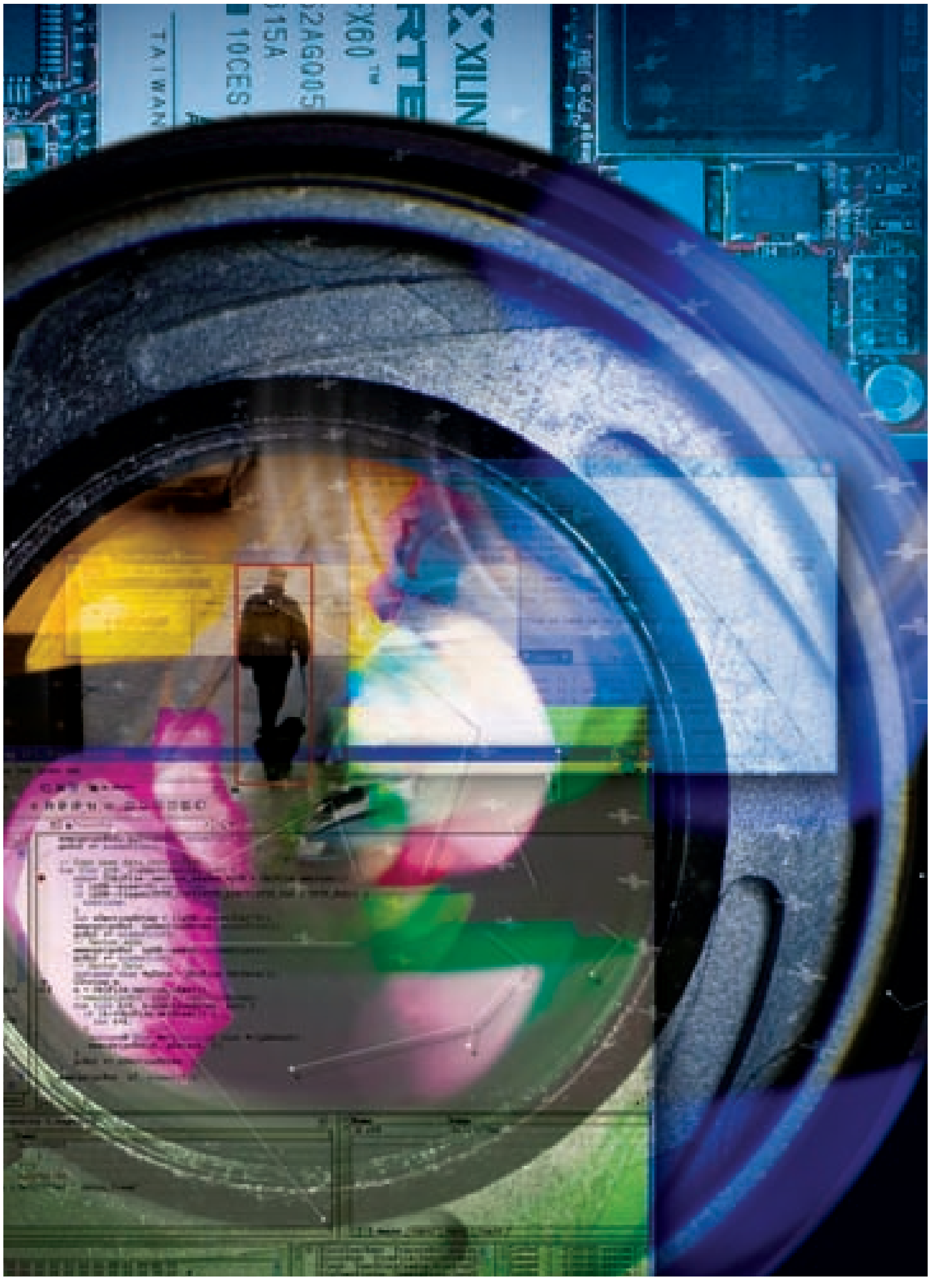
Develop Imaging Systems with the Right Mix of Technology

You can harness the combined strengths of DSPs and FPGAs to create a powerful imaging system.



by Justin Wheatley
Systems Manager
Sundance Multiprocessor Technology Limited
justin.wheatley@sundance.com

Imaging applications demand quite a range of processes to be performed in a single application. Although you can use multiple DSPs to provide more power and design nearly any function in a large enough FPGA, this is not usually the easiest, cheapest, or most efficient approach. High-level imaging algorithms like object recognition are best suited to the agility of a modern image processing DSP, while FFTs, edge enhancement, or MPEG-4 encoding would be better left to the vast parallel power of a large high-speed FPGA.

The obvious answer is to mix the two technologies (as discussed in the article, "The Benefits of FPGA Coprocessing," from the Third Quarter 2006 issue of the *Xcell Journal*), but DSP and FPGA designs are quite disparate disciplines, involving very different techniques, skills, tools, and usually people. These differences create obstacles to a fluid co-design process. Thus, the idea of using both technologies together can prove rather unpalatable to a specialist in one of the two fields – and even more so to an expert of neither. Integrating the hardware itself also presents a significant amount of work that you could avoid if you stuck with just one technology.

In this article, I'll review the SMT339 from Sundance, a powerful, off-the-shelf development system for imaging solutions created to help design teams realize competent imaging applications in a small, flexible, and highly expandable unit. A typical imaging function, such as a tracking system that highlights any person moving in a video image, suitably illustrates the features of the SMT339.

Performance Awaits Your Command

Figure 1 illustrates the hardware. The Texas Instruments Module (TIM) form-factor SMT339 is mounted on a carrier that might be PCI, PCI-X, VME, CompactPCI, or even stand-alone. The carrier provides a home and power to the module and usually an interface to a host computer.

A variety of analog video formats can be connected to the system through its video decoder. Alternatively, a range of digital video sources may be connected to the appropriate I/O daughtercard. The input sources are routed to the FPGA, which is a 60,000-logic-cell Xilinx® Virtex™-4 FX60 device. The data can be preprocessed here before passing to the DSP.

The FPGA has two independent 8-MB banks of ZBTRAM, which are ideal for frame stores for the various pre- and post-processing functions performed on the FPGA.

The DSP is a Texas Instruments TMS320DM642 video digital signal processor. You can program it using Texas Instruments's Code Composer Studio, which is fully software-compatible with the C64x DSP range. Clocked at 720 MHz, it is capable of 5.76 GMACs per second and is linked to the FPGA by three bidirectional video ports, with more ports for control and other data. All of the illustrated interconnections are already implemented in the FPGA, leaving you free to implement the specific functions required for your application. The DSP also comes with 128 MB of fast SDRAM. This 64-bit-wide memory holds the image buffers for the DSP and program/data space.

There is an output from the FPGA to a video encoder so that a live image (raw, part, or fully processed) can be displayed directly on a monitor – which is particularly useful when developing algorithms. Digital output may be obtained from the appropriate I/O daughtercard attached to the Sundance LVDS bus (SLB) interface or from high-speed serial links: connectivity to and from other modules is provided by 16 Xilinx RocketIO™ channels in the form of four Sundance rocket serial links (RSLs), giving the system significant expandability.

A Tough Act – To Follow

Figure 2 shows how the image data flows through the hardware resources when configured for our person-tracking imaging system.

Video is captured by the video decoder and fed to the FPGA. The decoder can handle a variety of different input formats from PAL/NTSC CVBS or separate YC. Alternatively, you can input digital video through a daughtercard digital interface module.

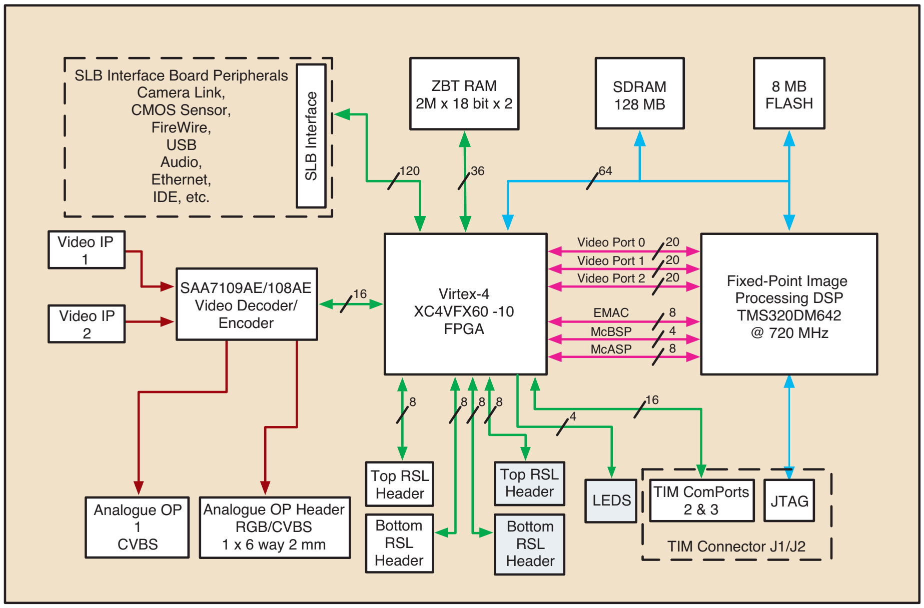


Figure 1 – Hardware diagram of the imaging module (SMT339)

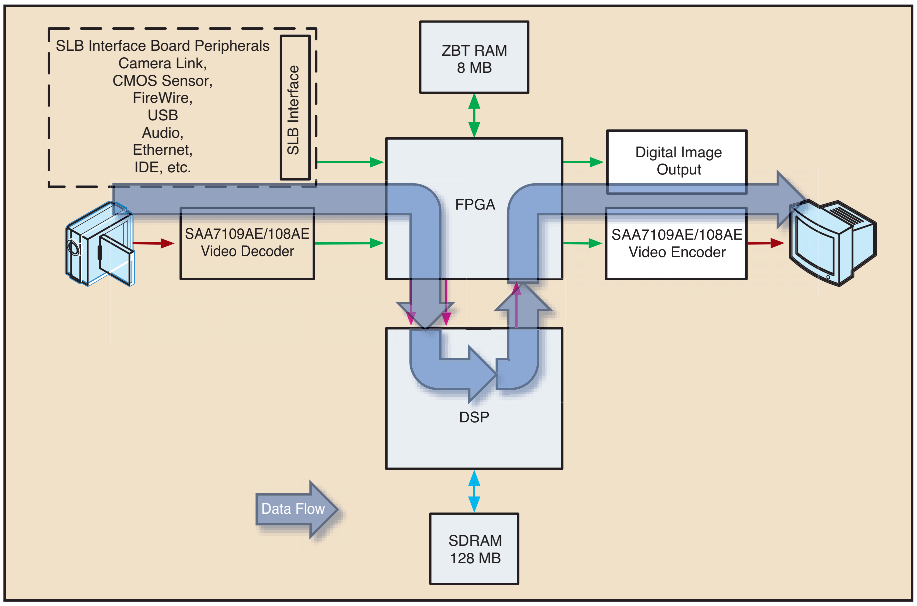


Figure 2 – Person-tracking system data flow diagram

The FPGA pre-processes the incoming video stream using one of the ZBTRAM banks as an image buffer. A slow-moving reference image is maintained by iterating each pixel by one bit per frame towards its value on the current image. The current image is compared with the reference image to calculate a difference image.

A great feature of this system, with its large and powerful FPGA, is that it is very easy to experiment with different pre-processors – like spatial filters, edge-enhancement filters, or histogram functions – and see the results immediately on the system's live output.

The calculated difference image, together with the current image, is passed to the DSP for analysis. The DSP performs a detection algorithm on the difference image, extracting the silhouette and looking for the characteristic shape of a person in the moving parts of the image. Various properties of a person detected in the field of view can then be calculated from this, such as:

- The coordinates of the corners of a region of interest around the person
- The center point (calculated as the mid-point of the region's corners)
- Their center of gravity (based on the center of the area of their silhouette)
- The coordinates of their entire outline (possibly returned in a binary overlay image of the scene)

The FPGA then post-processes this information, recombining it with the source image. For example, you could overlay an outline around the person, change their color, or perform other manipulations.

The results of the calculations are then available on a live video output (as shown in Figure 3), but they may also be output through the daughtercard or through other interfaces for transfer to, logging, or display on digital video systems.

The Software Supporting Role

This all sounds great in principle, but any DSP engineer is sure to wonder what sorcery would be required to realize such a cooperative combination in practice. Fortunately, the SMT339 (as with Sundance's many other modules) is supported by a comprehensive environment of development tools conceived to address this very problem.

Besides the many VHDL cores and DSP library modules available, a coherent software model is necessary to simplify the overall application development for these mixed-processor systems. Diamond from 3L Limited provides this model by describing multiprocessor systems as a number of independent tasks that communicate over channels. Whether these tasks are executing on DSPs or FPGAs, Diamond manages the interconnections and programming so that you can concentrate on their application.

Diamond uses the FPGAs on Sundance DSP TIMs by automatically adding the engineer's tasks to the standard Sundance firmware. These tasks are created in VHDL or tools such as Xilinx System Generator, or they can be standard netlist files, allowing you to bring proprietary cores into the system and mix them with standard and user-developed cores. Diamond automatically adds the logic to allow the tasks to communicate with other tasks; it then builds an FPGA bitstream using the standard Xilinx tools.

This technique creates a hardware independence that allows you to make major changes to the underlying hardware without having to change any code. It is now possible to expand the hardware by adding DSPs or FPGAs as required, without so much as a recompilation. Repositioning a task on a different DSP reduces to only a tiny change in a single text file. Thus, when more processing power becomes necessary, you can add it at will, and none of the existing development is made redundant. Even if a task needs to be moved from a DSP to an FPGA for acceleration, the surrounding functions are not affected.

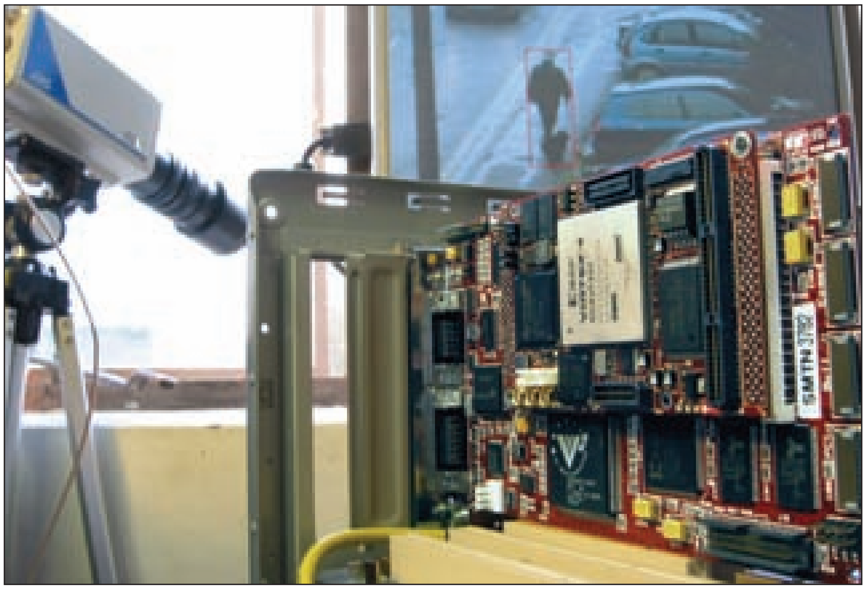


Figure 3 – The Sundance SMT339 in a simple system

Conclusion

The subtle art of combining the respective strengths of FPGAs and DSPs can be simplified by starting with a scalable system whose delicate integration is already complete. Supported by a comprehensive software environment, such complex hardware can become both adaptable and accessible, allowing its collective power to be finely tuned to the desired application. After the solution is developed, with such a modular approach, you can then create a production system efficiently from off-the-shelf modules.

For more information on creating FPGA and DSP systems, please contact *XL1@sundance.com* or visit *www.sundance.com.*