

When requirements outrun an architecture

By Ron Wilson , [EE Times](#)

June 03, 2004 (11:44 AM EDT)

URL: <http://www.eet.com/article/showArticle.jhtml?articleId=21401243>

Design house eInfochips Ltd. does a significant level of contract design work, ranging from conventional design services to verification and functional intellectual property, at both the board and chip levels. That gives the company, based in Ahmedabad, India, a wide selection of resources when making implementation decisions.

In the design services business, implementation decisions are usually made by the client, but an exception to this for eInfochips occurs when the company provides manufacturing-ready IP to its clients. Here, the IP may be implemented as a combination of ASIC, FPGA and standard-product solutions.

A case in point is the company's Keyboard/Video/Mouse (KVM) IP. The design targets remote server control: The KVM chip or board attaches to the keyboard, display and mouse connectors on the server and then to Ethernet. That lets a remote PC or workstation take control of the server over the Internet just as if the user were sitting in front of the server.

The KVM IP must replicate the PS-2 interface that the server expects to see from the keyboard and mouse, and it must absorb the video stream from the server's graphics engine. The outbound signals from the server must be compressed and transmitted as Internet Protocol packets, and incoming data from the remote user's keyboard and mouse must be received from the Internet, decoded and fed into the PS-2 interface. All of this must happen smoothly.

In 2001, eInfochips developed such a design at the board level for a client. "The decision about how to implement the IP was left to us," said Upendra Patel, chief technical officer. At the time, there were feasibility studies of lossy and lossless compression algorithms that would handle an XGA-resolution display at 10 frames/second-roughly a 95-MHz video stream. Key criteria included a low cost for compression hardware-the client wanted a bill of materials below \$400-and the ability to decompress in software alone on the remote terminal. And there was a significant Java component to the processing.

The 2001 design was successful using off-the shelf DSP parts for compression and control. But this year the client came back, and things had changed.

The new video requirement was for 1,280 x 1,024-pixel resolution at 35 frames/second, or roughly 150 MHz. And competitive pressure had forced the client's bill-of-materials target below \$100.

Patel said the architecture team quickly determined that the previous DSP would not handle the video rate for the current design. That left the team with the alternatives of an ASIC development, use of FPGAs or a search for a standard IC to do the job.

The first place to look was at existing video compression application-specific standard products. The most powerful video processing ASSPs available might have been able to handle the 150-MHz data rate, but they packed far too much stuff, having been designed for other applications. Nor

was an ASIC design feasible for the volumes the client was anticipating. KVM cards are a rapidly growing market, but not a huge one.

That appeared to leave FPGAs. But a quick calculation showed that an FPGA large enough and fast enough to meet the requirements would crush the cost targets. So eInfochips looked inward.

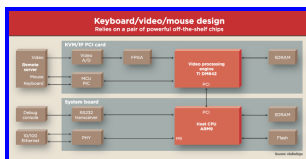
"We keep a large collection of vendor development tools for microprocessors and DSPs in our lab," Patel said. "That allows us to quickly put together a software prototype of an application on a number of different possible engines and compare them."

Those investigations found a solution. Texas Instruments Inc. had C64x-family DSP chips with video inputs-and sufficient processing power for the higher compression load. The chips also had a 10/100-Mbit Ethernet media access controller, suggesting that a single large chip might solve the problem. But further investigation showed the TI part would not be able to handle the Internet connection, Patel said.

A second search located an integrated off-the-shelf chip that, while again not really an ASSP, filled in the other missing blocks. A Netsilicon chip provided an ARM9 processor and more hardware that could handle both the control functions and the Java issues. The chip included a hardware-based, higher-level Ethernet controller and MAC, for dealing with the intense Ethernet traffic of the compressed video.

But there was not enough bandwidth left over in either the DSP or the ARM9 to handle the bit-flipping details of the PS-2 interface or the vital front-end processing of the video. Those tasks, and the PS-2 interface implementation, were handled in small FPGAs that were economical enough for the application.

Thus, the implementation evolved by a process of elimination. The most critical blocks were assigned to large off-the-shelf chips. Smaller tasks were implemented in inexpensive small FPGAs or software. By staying flexible, Patel's team met its cost target without resorting to an ASIC or using big FPGAs or ASSPs.



[See related chart](#)

