Squeezing a low-cost multiprocessor platform out of 130 nm

Michael Santarini - April 26, 2007

AT A GLANCE

LSI implemented the 2 million-gate Zevio 1020 in 130 nm instead of 90 nm. LSI designed a new 16-bit graphics processor and memory controller and modified the AHB. The actual design took nine months, excluding spec work. LSI derivatives will take only six months to produce.

Sometimes designing an SOC (system on chip) on the latest and greatest process technology is not what it takes to make an impact on the cost-conscious consumer-electronics market. That’s a lesson LSI Logic engineers took to heart in designing LSI’s Zevio 1020 multimedia-application-processor platform.

In December 2004, educational-electronics company VTech and IP (intellectual-property) developer Koto commissioned the company to create a multiprocessor SOC to run VTech’s VFlash “edutainment system” (Figure 1). Traditionally an ASIC vendor, LSI Logic has over the last few years been transitioning to selling standard products. So, when VTech commissioned LSI to create an SOC, LSI’s management decided to turn what would have traditionally been an ASIC design into a general multimedia-processor platform.

Shinya Fujimoto, architect of the Zevio project, says that his design group at LSI had to work within several constraints: Create a multiprocessor platform that was generally modular, so LSI engineers could swap out blocks to quickly create derivative products; implement the SOC on a mature, 130-nm, low-power process technology to keep costs reasonably low and still hit performance and power goals; and, finally, finish the initial SOC platform in nine months.

“Our background was consumer ASICs,” says Fujimoto. “Our group did chips in the PlayStation and PlayStation 2 and some of the iPod designs, and, during the development of those [products], we noticed that we were spending a lot of time redefining the noncritical part of the chip…That’s why we decided to develop this architecture.”

Fujimoto says that the first step in defining the Zevio architecture involved meeting with the VTech and Koto system architects. “We try to get as much feedback and even complaints from the customer to define where the potential bottlenecks are going to be in the design process,” he says.

The Zevio, Fujimoto claims, is not a typical application processor (Figure 2). “It’s what I call a heterogeneous multiprocessor,” he says. “It has multiple processors that are not the same but run in parallel, and each of the processors [is] dedicated to certain tasks they are good at.”
The group determined that the SOC would incorporate an ARM9 processor for generic processing. The LSI team would create a custom graphics processor and audio- and memory-controller cores, and the design would incorporate an LSI Logic ZSP DSP core to do more immediate decoding and codec-type applications. “The key for us was to have multiple processors that run independently”—meaning, according to Fujimoto, that each becomes its own master and completes its operations without CPU intervention. “We had to make sure that they could all run efficiently without causing a bottleneck on the bus or the memory.”

Implementing the design on a mature, low-power, 130-nm process (from an undisclosed partner in Taiwan) instead of a 90- or 65-nm process helped keep chip costs down, stabilize power management, and avoid the use of DFM (design-for-manufacturing) tools, all to speed the design process. The chip targeted applications for systems that cost less than $100, says Fujimoto. “Anyone can make a huge chip, but some people, especially video-game vendors who will remain nameless, are struggling now because of increased chip costs going into consumer products.”

Fujimoto notes that many companies are too quick to jump to the latest and greatest new process when they could have accomplished more with a more mature and stable process, such as 130 nm. “Some people left the 130-nm segment just because it is considered lower end, but we felt we could create a high-end design through smart engineering. Customers hear stories from competitors about high-performance this and that—all the flashy terms—but in the end, it comes down to [getting] the best bang for your buck.” He maintains that the maturity of the 130-nm process, its cost, and its performance made the best fit.

LSI wanted Zevio to be a reusable platform, not just an ASIC, so a key component was creating unique cores that you could combine with a range of LSI or even third-party vendor cores. However, a major component of the architecture specification was a new graphics-processor-core design. LSI worked with IP-vendor Koto to develop Zevio’s 3-D-graphics-processing core. Fujimoto notes that several of the Koto engineers previously worked on the Nintendo GameBoy design team. LSI had little experience in this area and so welcomed Koto’s experience, says Fujimoto. “[The engineers] provided a lot of insight into the features software developers need and helped us transfer a lot of that know-how into hardware.”

Koto’s team did most of the specification work and verification on the core, and LSI designers handled the design and RTL implementation of the core. The 16-bit, 3-D-graphics core requires only 300,000 gates and consumes 20 mW running at 75 MHz, which allows it to draw 1.5 million polygons per second.

Fujimoto pointed out that another potential bottleneck was the memory controller and its interface to the system memory. Although, as chip designers, it would have been easier for them to work with a 32-bit bus interface, run it fast, and obtain the desired performance, LSI had to figure out how to get the same performance from a 16-bit interface memory, to reap the cost benefits.

The team had to design from scratch an efficient controller core that would use a 16-bit interface and allow for efficient arbitration (Figure 3). “What we figured out when we looked at a 16-bit interface was that we could get a nice timing window where we [could] issue many commands to the SDRAM,” says Fujimoto, who notes that the controller can fetch two words for every bus-clock cycle. “We’re patenting that technology. It allows us to efficiently use these timing slots to open up multiple banks.” Fujimoto views this development as the key innovation in the architecture.

The team wrote RTL for the controller and then ran proof-of-concept simulations. “We saw a lot of improvement over typical designs of memory controllers,” he says. “At that stage, we proved that we could get good performance on a 16-bit interface.”
Fujimoto says that next the group had to look at the related bus-protocol bottleneck. The group decided to use the popular ARM AMBA (Advanced Microcontroller Bus Architecture) AHD (AMBA high-speed-bus) protocol but had to figure out ways to overcome some of AHB’s inefficiencies. “We had to look at its inability to do burst writes to random addresses and its inability to specify large enough burst writes,” he says. To work around these issues, the group wrote its own extensions to AHB, which helped double the memory controller efficiency.

After working out the larger issues of the specification phase, the design group, which ranged at times from 10 to 15 design and verification engineers, began work on the RTL design. Fujimoto broke his team into subteams. “One team focused on the memory controller, one on the graphics core, another on the audio core, and another team on the overall integration,” he says.

Fujimoto says that LSI’s traditional methodology is to have each subteam design and then verify the individual blocks and then verify the blocks with other cores in the system using simulators. But for the first time, his team this time employed an FPGA-prototyping system to run verification and hardware/software validation on the graphics core, memory controller, and an audio processor LSI developed.

“There would have been too many corner cases to validate this design just in a simulation environment,” says Fujimoto. “This [instance marked] the first time our group used an FPGA-prototyping system. In retrospect, we should have done a bit more verification on the individual cores earlier in the design.” Fujimoto recalls that, once the FPGA system was working, things progressed quickly: “When we ran into problems, we could see them on the LCD screen hooked up to the FPGA-prototyping board. We could simply stop the debugger and look at the internal registers where the error occurred.”

The team also ended up chasing down bugs that turned out to be glitches in the FPGA-programming software, not the design. “We didn’t realize we were tackling bugs that were caused by the tool vendor,” says Fujimoto. “We used the FIFO controller provided by the FPGA vendor, [which] also had a problem.”

Fujimoto notes that the graphics team created a C model of the graphics core during the spec phase. “We would have used the C model for the verification, but it was too much work to update the C model as well as the RTL,” says Fujimoto. That C-model methodology works if you have a fixed specification, he notes, when you can use the C-model output to verify RTL output. In LSI’s case, however, the spec wasn’t 100% complete when the team started RTL: “We dropped the C model once our FPGA system was up and running,” he says. The group also started preliminary driver development once it stabilized the design on the FPGA-based prototyping system.

The prototyping board served as the basis for a validation board that LSI now offers customers wishing to develop other systems with Zevio SOCs. After the group had stabilized the RTL, the team ran a trial synthesis and a trial layout, filling in portions of the preliminary layout with “blank gates” to get a rough idea of the area and die size. Fujimoto says that his group uses Synopsys for synthesis and Magma for place and route.

Fujimoto says that the trial layout was especially important for determining the correct placement of the memory blocks. The design incorporated 240 kbytes of SDRAM. Therefore, the Zevio layout team had to work with the RTL team to break up the blocks to ensure that the various cores could efficiently access the memory and not take up too much room in the layout. “We had this big memory, but we allowed both the graphics engine and the DSP to access the same memory,” he says. “At the functional register, we assigned which core gets access to the memory and what it will access, so we had to predefine certain segments for the DSP and graphics engine.” To achieve this
goal, the layout and RTL group went through several iterations to determine the optimum layout of the memory. In the end, the entire Zevio SOC consisted of 2 million gates.

To help keep the design low power, the group used a multivoltage-threshold, 130-nm library. “The idea was to synthesize the entire design with all low-leakage, low-performance gates and then identify the bottlenecks in the timing and the critical path and convert those paths into high-speed gates,” says Fujimoto. Following that process allowed the team to achieve the right mix of low-leakage and high-performance transistors.

Physical verification and tapeout of the design went fairly smoothly, says Fujimoto. “The prototypes [silicon] came, and, three days later, all our demos were running on the real system,” says Fujimoto. “[The customers] were able to go home a week early ... Going with a mature technology along with the validation and prep work we did early in the process really paid off.”

The fast board “breakup” also meant that software teams at VTech could quickly move into product development and ultimately allowed VTech to introduce its system to the market in time for Christmas 2006.

But VFlash isn’t the only application of the Zevio product line. In fact, Fujimoto says that, because the design group made the SOC modular, LSI teams can tailor it for other customer applications. “We designed the platform with the goal [of creating] derivatives of the platform in only six months,” says Fujimoto.

He notes that the modular platform allows users to fairly easily swap out the ARM core for a MIPS core because LSI has licenses for both. Users can also swap out several of the peripheral cores. LSI is working on adding USB support to the system. Fujimoto says that next-generation Zevio platforms will also likely incorporate DDR instead of SDRAM. Although Koto created the original operating system for the platform, LSI is expanding the number of operating systems the platform supports and is now working on Linux OS and Windows CE.

Fujimoto says that his group has finished another project on Zevio for an unnamed customer. He says that the silicon is in production but the customer has not yet introduced the product.

LSI began the Zevio specification process in December 2004, started the design at the end of 2005, and went into production nine months later in September 2006.

The Zevio project is yet another example that success in the consumer market is not always equal to implementing the fastest SOC in the latest and greatest process technology. With a lot of planning at the architecture stage and a bit of creative engineering along the way, the LSI team created a relatively powerful, cost-effective platform that helped VTech hit its market window and gave LSI a versatile tool to help other customers do the same. It will be interesting to see how long the Zevio remains a viable platform for LSI and how many derivatives the company can spin for other customers.
You can reach Senior Editor Michael Santarini at 1-408-345-4424 and michael.santarini@reedbusiness.com.