AT&T

Ensuring a First-Pass Success for a Next-Generation RAID ASIC

Pressure is a standard component of every project undertaken by the Wichita, Kansas-based Global Information Solutions Logic Design Group at AT&T. But for a team of design and test engineers working there in the summer of 1993, the pressure was particularly intense. That's when they began designing a chip due for a mid-1994 release and destined to play a key role in an important next-generation product. That's also when they faced a number of risk factors that bore heavily on the success of the chip and of the product itself.

The chip was one of the most complex the engineers had ever designed—a quad-flatpack ASIC with a size of 39,000 equivalent-NAND gates. Moreover, its performance and reliability would be critical to the workings of the much-awaited new product: a SCSI-to-SCSI disk-array controller —the latest addition to the group's industry leading RAID product line. Acting as the interface to the board's internal PCI bus, the chip would provide vital DMA functionality as well as high-speed RAID parity assist between bus and memory.

Adding more pressure still, this would be the first time the engineers had designed such an architecture using the PCI bus. They had no choice, because only the PCI bus could provide the high internal bandwidth vital for SCSI-to-SCSI and SCSI-to-memory transfers.

> "The ability to perform such thorough testing using the PCI Bus Interface Model gave the engineers the confidence they needed to get the job done under great pressure."
>
> John Kloeppner
> Senior Principal Design Engineer
> AT&T Global Information Solutions Logic Design Group

The project also represented the first time that many of the engineers would be using VHDL for both high-level ASIC design and testing. As if all this weren't enough, the engineers also would be working with a brand-new version of EDA software.

The engineers felt the pressure keenly, particularly when it came time for design verification and testing. "It was extremely important for this chip to be a first-pass success," says Tom Langford, consulting test engineer and a member of the design/testing team. "That's the only way we had a chance of meeting the scheduled release date."

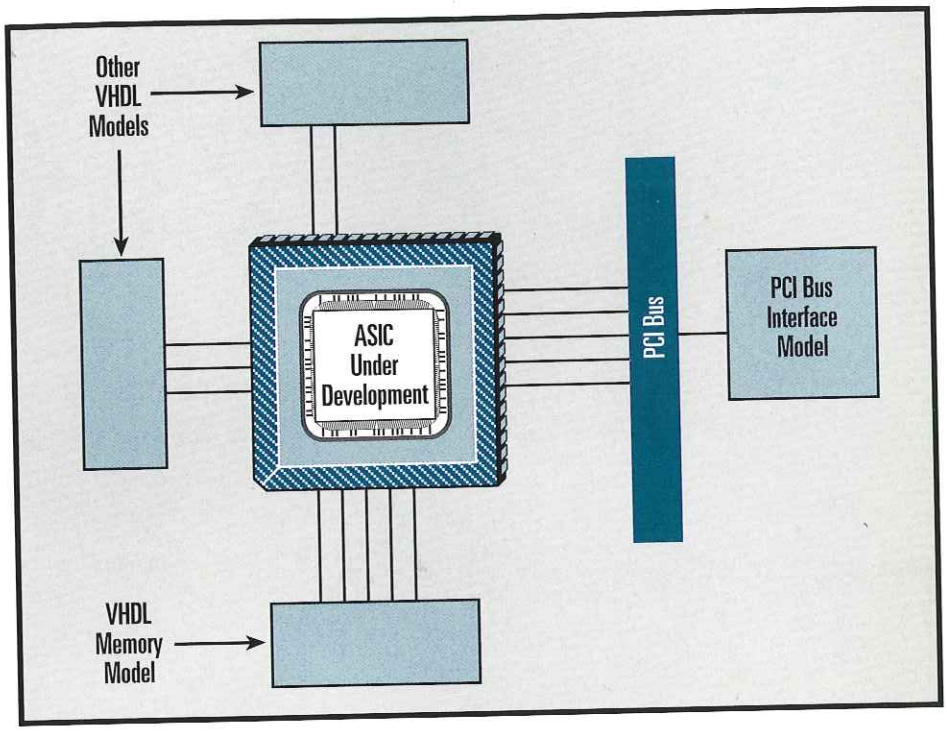
MORE THAN 250 TESTS

Considering the complexity of the chip, the hundreds of separate tests it would require, and the time pressures ahead, the design team decided it needed to use a PCI bus modeling tool. The engineers were concerned, however, about devoting resources to building such a tool which would require implementing more than 100 pages of specifications. They also were concerned about the risks of building their own tool for such a critical product development effort, so they decided to buy one instead. The PCI Bus Interface Model from the Logic Modeling Group of Synopsys, a provider of simulation models the team had used successfully in the past, was selected. "It just made sense

to purchase a tool that already was built and verified, and came from a company we had worked with before," Langford explains.

Simulation began in the fall of 1993. With the help of the automatic protocol-generation capabilities of the PCI Bus Interface Model, the engineers constructed and performed 50 production tests and an additional 200-plus functional tests. Because the tool was easy to use and learn, it was the 'workhorse' tool used for design verification and testing. The tool filled a role in the engineers' larger strategy of "pins-out" ASIC verification. "By using the PCI Bus Interface Model along with internally written VHDL models for other components on the RAID board, we were able to greatly simplify test vector generation and design verification," explains John Kloeppner, senior principal design engineer and a member of the design/test team.

On April 1994 the ASIC taped out on schedule. The first parts were received for integration into the RAID board and proved to be compatible with other PCI interface chips on the board. These included several NCR 800 Series PCI-to-SCSI controllers and the Intel 82420 (Saturn) PCI chip set, which interfaced between the PCI bus and Intel 80486 CPU. A first pass success!



▲ **"Pins-out" verification environment using the PCI model and custom developed VHDL models.**

BUILT-IN TIMING, MACRO LANGUAGE STREAMLINE WORK

As Langford and Kloeppner both observed, the PCI Bus Interface Model played a major role in the success of the RAID ASIC, particularly by freeing engineers from having to develop force files or to generate macros from scratch. "Thanks to the timing and macro language built into the PCI Bus Interface Model, we were able to avoid doing any of that sort of work, which saved us a lot of time and effort," Kloeppner says.

For Langford, the user-friendliness of the tool's macro language was critical. "Because it was so easy to use, we were able to create, modify, and manipulate data patterns a lot faster than we had ever done before," he says. On other projects, for example, engineers ordinarily would spend a full day creating a pattern from scratch, but on the RAID ASIC project they performed the same kind of task in only half an hour. They also were able to streamline the modification required for reuse of existing patterns — cutting the time to perform this task by 25 to 30 percent.

THE CONFIDENCE TO GET THE JOB DONE

The PCI Bus Interface Model was critical to the overall design verification effort, especially at crunch time – tape out. Before the tape could be released, an internal requirement of single-stuck-at (SSA) fault coverage had to be met. It was discovered that during fault simulation several functions had not been thoroughly verified. One of these functions was the lock-function. "With the PCI model we could quickly modify an existing debugged pattern and verify the functionality in a minimum amount of time," explained Langford. This is just one example late in the design cycle where the PCI Model helped keep the chip on schedule.

Another advantage of using the PCI Bus Interface Model was its support for timing variation, particularly when it came to creating worst-case simulation. "Without the tool we would have had to add timing to each of our stimulus commands," Kloeppner points out. "As it was, we could change one timing value at a central location and the tool would take care of varying the delay between signals. This gave us the ability to pump data across the PCI bus efficiently and stay on schedule."

In the end, says Kloeppner, the ability to perform such thorough testing gave the engineers the confidence they needed to get the job done under great pressure. "All in all, we were delighted," he reports. "We met our schedule, there were no problems with the PCI bus, and the chip was a first-pass success."

```
- repeat with memory parity checking enabled
- SET UP APC CONTROL
- disable parity check mif option reg
        config_write ("010000c0",1,'0',0,"00100000",0,0,false,false,false);
        config_read ("010000c0",1,'0',0,"00100000",0,0,false,false,false);
- write and read back apc options
        config_write ("01000090",1,'0',0,"3e000000",0,0,false,false,false);
        config_read ("01000090",1,'0',0,"3e000000",0,0,false,false,false);
        mem_write ("a8000080",1,'0',0,"00000010",0,0,false,false,false,
        linear);
        mem_write ("a8000084",1,'0',0,"00000000",0,0,false,false,false,
        linear);
        mem_write ("a8000088",1,'0',0,"00000020",0,0,false,false,false,
        linear);
        mem_write ("a800008c",1,'0',0,"00000040",0,0,false,false,false,
        linear);
        mem_write ("a8000090",1,'0',0,"00000060",0,0,false,false,false,
        linear);
- start
        config_write ("01000094",1,'0',0,"11230000",0,0,false,false,false);
-read back apc command and status
        config_read ("010000a4",1,'0',0,"00000000",0,0,false,false,false);
        idle (1);
-sleep until interrupt occurs
        sleep (0);
        idle (1)
```

▲

Actual bus model control file used to generate test patterns.

LOGIC MODELING

LOGIC MODELING GROUP

19500 N.W. Gibbs Drive
P.O. Box 310
Beaverton, OR 97075
Tel.: (503) 690-6900
Fax: (503) 690-6906
Tech. Product Support:
1-800-445-1888
e-mail: modelinfo@lmc.com

EUROPE

Stefan-George-Ring 24
D-81929 Munich, Germany
Tel.: 49 89 99 39 12 30
Fax: 49 89 99 39 12 32

JAPAN

Dai Tokyo Kasai Shunjuku Bldg. 6F
3-25-3 Yoyogi Shubuya-ku
Tokyo 151, Japan
Tel.: 81-3-5351-5300
Fax: 81-3-5351-5330

ASIA-PACIFIC

No. 31-02
The Concourse
300 Beach Road
Singapore 0719
Tel.: 65-296-7433
Fax: 65-297-1272

SS214 12/94 WSS 7.5k