

STMicroelectronics Engineers Find Mixed Analog-Digital Verification Success with OVM and Mentor Graphics Questa ADMS Simulator

by Alberto Allara, Matteo Barbati, and Fabio Brognara, STMicroelectronics

DESIGN CHALLENGE

- Verify IP for a R/W channel to be integrated into a STMicroelectronics hard disk component (an SoC)
- Build a verification process from reusable steps

SOLUTION

- Focus on integration into the larger SoC, not just on the IP
- Make maximum use of standards, start with OVM
- Start writing verification components as soon as design work in RTL is underway
- Use Mentor Graphics Questa ADMS tool for analog accuracy and easy integration into digital verification flow

By now it's a cliché to speak of the rise of digital technology. Follow technology coverage in the media for any length of time and it doesn't take long to note the tacit assumption that nearly anything with an on/off switch will eventually communicate with the world at-large exclusively in strings of 0s and 1s. Of course, as long as the electronics industry is built on harnessing the laws of physics, the importance of analog signals will never go away. Nature speaks in waveforms, not regimented bitstreams. So the challenge, and one that must be repeatedly solved by those building ever more complex semiconductor devices, is how to verify what's happening at the analog-digital interface.

One recent solution comes from a trio of engineers at STMicroelectronics in Milan, Italy working on verifying an important bit of intellectual property (IP) for an ST hard disk component. The team's approach was to combine OVM methodology, Mentor Graphics Questa ADMS simulator and lots of collaboration among engineers at both companies.

A PRIMER ON HARD DRIVES

In general hard drives offer a case study in handling the analog-digital hand off. Consider a stream of binary data to be written to a drive. Those 1s and 0s in the bitstream must be encoded and then output as an analog waveform,

a conversion handled by the ST IP. The waveform is imprinted on the magnetic regions of the drive's spinning disk, or platter, thus storing the binary stream.

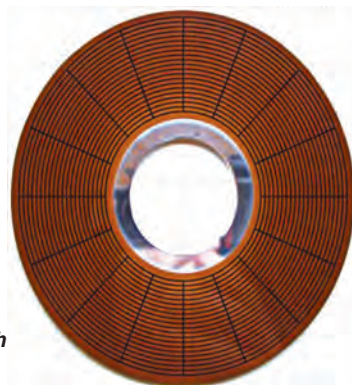
To retrieve the data from the drive, the process more or less runs backward. The head of the actuator, the writing and reading element, moves over the disk where the data are stored. The pattern of magnetization on the disk changes the current in the head, and this change can be represented as a waveform. Next comes sampling of this waveform by the ST device and finally a new stream of binary data, with all the algorithms in place to check data integrity and apply data corrections, to be fed back to that emerging digital future we mentioned earlier.

Allara and his colleagues were working to verify IP for a read/write channel to be integrated into the larger ST SoC. Perhaps the biggest challenge was how to meaningfully work across analog and digital domains. Historically, engineers specialize and develop a set of skills relevant to just one domain. Digital verification engineers eschew graphics and spend most of their time writing and compiling vast amounts of code.

Figure 1. Typical hard disk drive; ST's component converts analog waveforms into digital bitstreams and vice versa — critical to reading from and writing to the drive.



Figure 2. At their core, hard disk drives are analog devices, storing data magnetically on tracks — concentric circles placed on the surface of each hard disk platter.



By contrast, analog verification engineers look warily at code as most of their work is done via graphical interfaces. Though references to “mixed-mode simulation” abound, Allara believes the phrase generally refers to implementing an analog design digitally, not truly working across domains.

“The bottom line is that this is complex IP,” says Allara, who earned his master’s degree from Politecnico di Milano in 1994 and has been working in the industry since then. “And through the years, the complexity of the digital portion of all of our analog-digital devices has constantly increased, forcing us to look for leading edge verification technology.”

Another challenge was how to approach the verification process in such a way that at least some of the steps and tasks could be subsequently reused. A major issue, one faced by many verification engineers, is how to avoid starting from scratch with each new project. Yes, a custom approach may give a team a chance to demonstrate its technical prowess. However, one-off verification is also undeniably a huge sunk cost, particularly as verification complexity skyrockets. Recall that verification complexity increases at some multiple of the rate of increase of the number of gates, a troubling maxim given that gate counts already number in the hundreds of millions or more.

BEFORE CODING, THINK FIRST

The first step for the ST team had nothing to do with coding or even formal planning. Rather, they worked to begin thinking differently about the verification process. For example, rather than focus exclusively on the read/write IP, they instead chose to consider how this IP could be integrated into the SoC. The team realized that given the nuanced relationships among the various building blocks that make up the SoC, there is more to verification than just looking at the various signals expected to pass through the IP they were assigned to verify.

Allara and his colleagues also were determined to make maximum use of standards by leaning heavily on Open Verification Methodology, or OVM. (As an aside, it’s worth noting that since this ST project was done, Accellera ratified version 1.0 of Universal Verification Methodology, or UVM,

a direct derivative of OVM; see <http://bit.ly/eK4oUI> (PDF). Allara, who with his nearly 20 years of experience counts as a seasoned pro in IP and SoC verification, says the rise of standards is the biggest change he’s experienced in how he approaches his work. It wasn’t long ago that engineers wrestled with HDL and other description languages to build and verify RTL designs. Companies cobbled together their own techniques — in fact this state of affairs still exists today — using everything from C code to SPICE simulation.

Standards came into play immediately when the actual verification work began. A second ST team working on the processor’s analog front end used VHDL AMS to build a model of the analog domain which was provided to Allara and his colleagues, Fabio Brognara and Matteo Barbati. The model allowed Allara to close the loop with the RTL describing the digital channel front end and was relatively easy to integrate with other languages, particularly VHDL AMS.

Allara says his approach was to start writing verification components as soon as his ST designer colleagues started developing the digital design in RTL. Accordingly, the first phase in Allara’s verification effort was to verify the digital design in greater and greater detail as more of the RTL took shape. Later, when the RTL was finished and Allara had received the model of the analog front end, his team moved to true mixed mode simulations, while reusing much of their verification infrastructure.

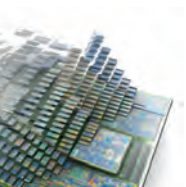
The OVM methodology requires specifying verification components for each interface of the device. These components are coordinated via a multi-channel sequencer operating at a higher, more abstracted layer of the OVM environment. The components developed by Allara’s team allowed them to program the registers of the read/write IP. For the “read” portion of the IP, they created a component that generated a model stream of bits similar to that which might be read from a hard drive’s spinning disc. Another verification component extracted the information after the stream was processed to compare the output to the expected result. The team developed a similar set of components for the “write” portion of the IP.

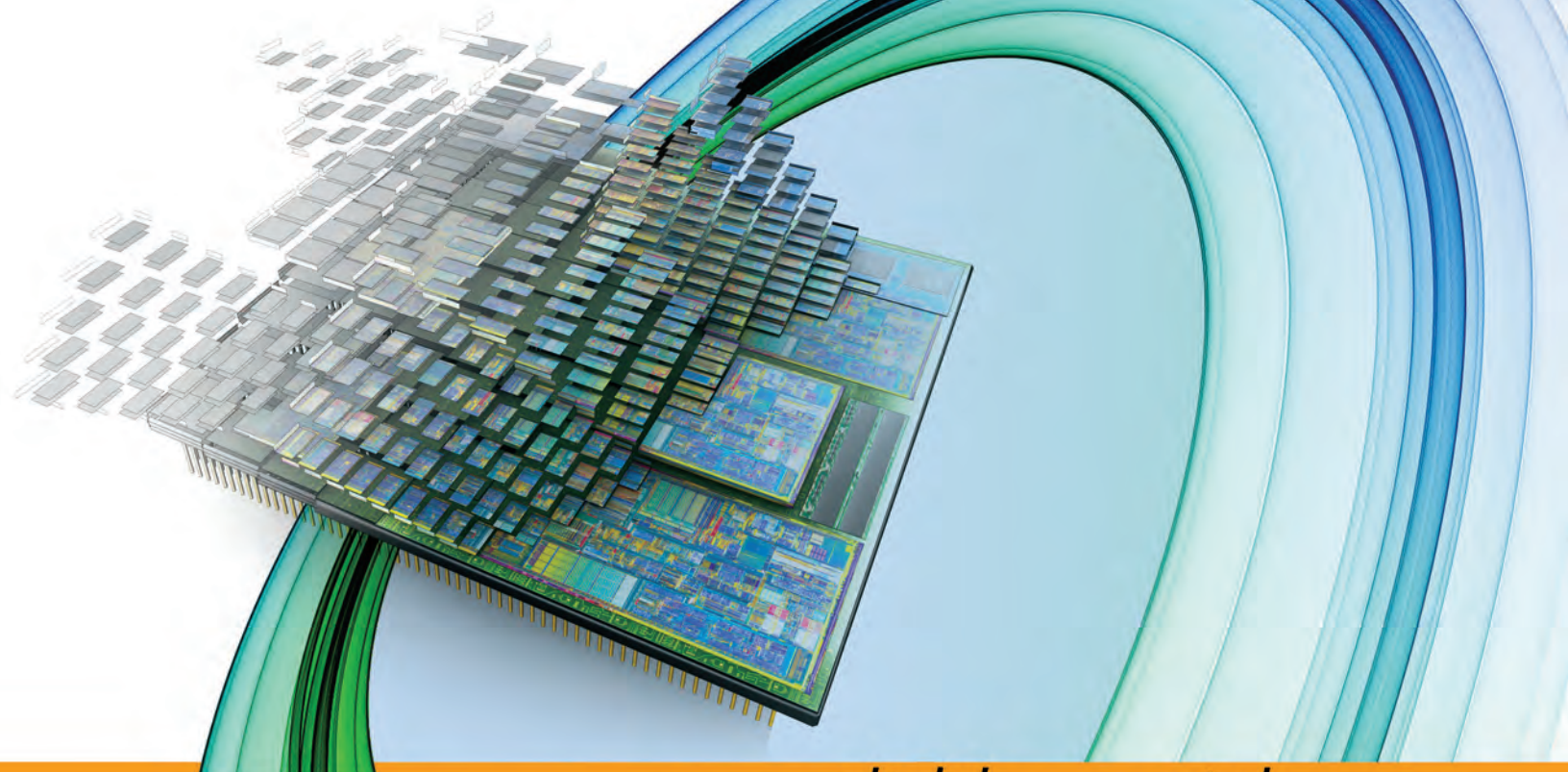
The simulated bitstream was produced by a pattern generator written in C code and embedded in a verification component. To reuse their environment for mixed mode simulation, the team created a layer that could convert the bitstream into a phased series of various input signals that roughly approximated the expected input to the IP's analog front end. This simulated conversion from analog to digital signal was done via VHDL AMS.

Using the methodology they built, the team was able to find bugs in the analog domain that the ST analog designers themselves missed. The reason, Allara says, is that his team's approach was able to feed the analog front end a pattern very similar to one that might actually be read from a disc. By contrast, in analog-only verification, the simulated patterns are often simple and fairly symmetrical, two characteristics that don't describe the complex, sprawling and often asymmetrical waveforms produced by the magnetic fields.

FINDING BUGS IS THE FINISH LINE OF VERIFICATION

In the end, the goal of verification at the analog-digital interface is to make a discovery, or more precisely, to find problems with the design before it's built and those problems are found by customers. Any methodology that seems to ratify designs as mostly okay is more apt to be evidence of faulty verification than proof of a good design, which is why Allara's success in finding bugs that others had overlooked is perhaps the best evidence of his team's success.





verification HORIZONS

Editor: Tom Fitzpatrick
Program Manager: Rebecca Granquist

Wilsonville Worldwide Headquarters
8005 SW Boeckman Rd.
Wilsonville, OR 97070-7777
Phone: 503-685-7000

To subscribe visit:
www.mentor.com/horizons

To view our blog visit:
VERIFICATIONHORIZONSBLOG.COM

Mentor
Graphics[®]

www.mentor.com