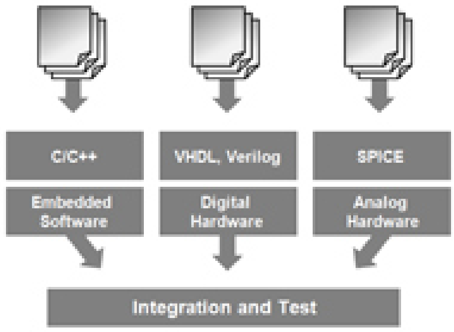
Playing Nicely Together in Wireless Design

By Mike Woodward, Communications Industry Marketing Manager, The MathWorks

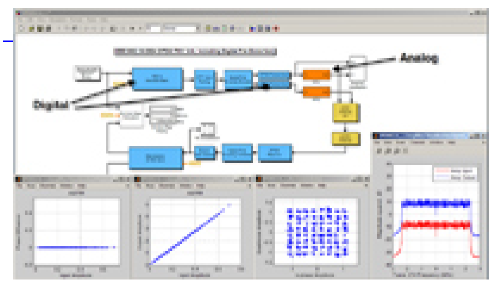


[1]

The use of silo-based methods in wireless systems design often introduces problems in the development process. Using system-level models solves many of these problems, and results in significant design process improvements. Companies are using Model-Based Design to unite different development teams (e.g., analog and digital) and different design stages (e.g., design and verification) to great benefit. In fact, the use of system-level models can bring verification right to the start of the design process where errors are cheapest to correct. In this article, we show how development teams are achieving these goals.

At times, creating a modern wireless communication system feels like conducting a discordant orchestra. The different parts of the orchestra (analog, digital, network) all play their separate parts in isolation, and it's taking more and more rehearsal time (verification) to bring everything together to produce the final piece. Heightened competition, the specter of commoditization, and recent staff layoffs are all creating a pressing need for more efficient design processes. So is it possible to enable different development teams to play together better? Can we make design process improvements without abandoning our existing and trusted toolchain? Is it possible to cut down on rehearsal time and get to market faster?

The Problem



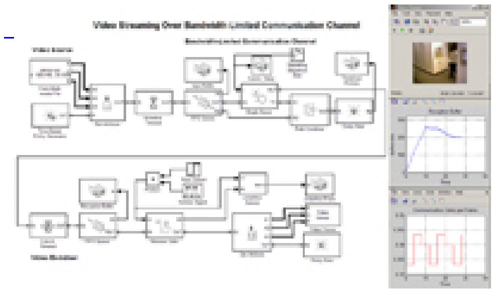
[2]

The silo development methodology (see Figure 1) causes system development problems because we develop components in isolation. As Alberto Sangiovanni-Vincentelli has written, "The resulting lack of an overall understanding of the interplay of the subsystems and of the difficulties encountered in integrating very complex parts causes system integration to become a nightmare" (1).

This silo approach makes design tradeoffs extremely difficult. For example, we may wish to use a cheaper power amplifier and correct for nonlinearities using digital methods (e.g., digital predistortion), something difficult to do with many design tools.

Often, design tools are targeted at a single design domain (e.g., analog design) and don't cosimulate well with tools for other design domains. This makes interaction between engineering teams more difficult than it needs to be. Silo development also introduces verification inefficiencies. It pushes integration testing toward the end of the design process, when bugs are more expensive to fix. Adding to this is the tendency of engineering teams to write test harnesses from scratch, instead of using trusted models created earlier in the design process. By design, the silo development approach is fragmented. This fragmented process increases project risk and makes it difficult to produce globally optimized designs. Can we link different parts of the design process and get to market faster? Can we do this in an evolutionary way?

Linking Design Domains

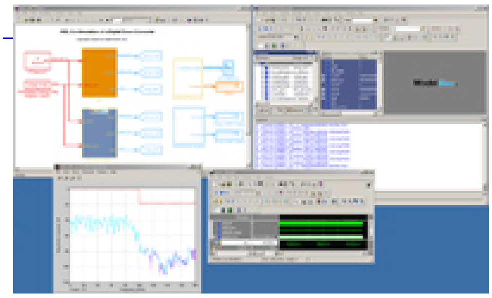


[3]

To link different design disciplines, we need to be able to simulate and design different design domains in the same model in the same model execution run. For example, we may wish to simulate in one model a system that has digital, analog, network, and MEMS subsystems. This requires a simulation platform that can simulate different types of systems at the same time. Fortunately, such system-level design platforms exist (2). Figure 2 shows one such platform modeling a digital predistortion system including both analog and digital elements, and Figure 3 shows the same platform used to model a network and digital system. These system-level design platforms carefully combine different simulation types and tools, allowing the user to build one system model combining the behaviors of all subsystems.

Using multidomain platforms rapidly enables us to find out whether a system composed of different subsystems will work and, of course, to optimize across design domains. This system model can act as a golden reference for the next stages of the design flow.

Linking Different Tools



[4]

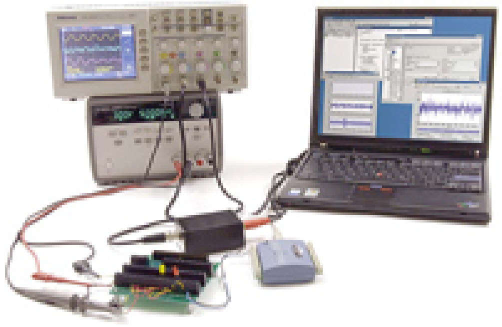
click to enlarge
Figure 4. Cosimulation with Simulink and ModelSim.

Linking different domains offers advantages, but such system-level models can't replace detailed implementation work using dedicated analog or digital tools. Unfortunately, analog and digital design tools often don't interface well with one another, and often the only communication is via offline file sharing. This makes it difficult to test analog-digital behavior before a physical prototype is available.

However, if we already have a functioning multidomain system-level model, could we use this model to help? For example, can we use our system-level model as a test harness for our detailed analog design? To reuse our system model, we need our system-level design platform to have cosimulation links to implementation tools. These cosimulation links should be run-time links enabling us to examine the dynamic behavior of systems (e.g., the analog-digital or digital-mechanical interface). These cosimulation links require different tool vendors to work together to bridge the gap between design tools. Fortunately, this is happening; Figure 4 shows two such tools cosimulating, in this case Simulink from The MathWorks and ModelSim from Mentor Graphics (3). Here, the tools are exchanging data at each simulation time step, enabling simulation of the dynamic behavior of the analog-digital system.

This kind of cosimulation offers three benefits. First, it enables the system-level model to be reused as a test bench during the implementation phase of the project. Second, the system model acts as a common simulation platform between different disciplines, enabling collaboration via a common model all can understand and use. Third, it enables us to benefit from a more integrated development approach while still using existing tools, reducing adoption risk.

Linking Verification

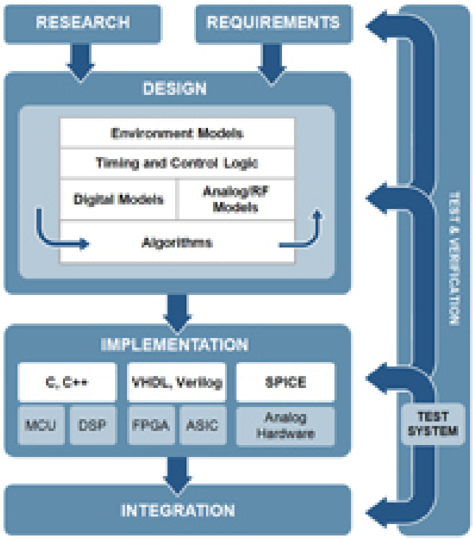


The most expensive time to find errors in any wireless design is right at the end of the development process, which is unfortunately where we find most errors. We want to get bad news as soon as possible, so we would like to verify as much of our system as early as we can in the design process.

We can reuse system models as test harnesses when we implement and verify our designs. This approach moves integration testing much earlier in the design process and allows us to find integration errors before we have committed to silicon. This early verification has yielded large savings in several projects (4). In fact, by using a multidomain simulation we are able to rule out unworkable designs at the start of the project. Effectively we are bringing verification to project inception.

The last step in the wireless development process is the physical prototype, for example the RF front end for a base station. We examine the prototype's behavior using test equipment. There is an opportunity for us to reuse the system model to generate test vectors or to provide custom data analysis (for example, calculating standard specific measurement parameters such as ACLR for HSDPA (5)). Although modern test equipment generates standard specific waveforms and does have some standard specific measurement options, wireless developers will often have their own parameters and quality metrics to measure. Here, we can extend the system model to take those measurements and to do custom calculations and analytics. To do this the system modeling tool must have links to test equipment , and many system modeling tools have these links. Figure 5 shows one such tool communicating directly with a device-under -test via test equipment.

Playing Nicely Together



[5]

The design flow we are advocating, shown in Figure 6, starts with a system-level design and integrates with existing flows to offer efficiency improvements without the risk of revolutionary change. We reuse the system model as much as possible throughout the design process, extracting the maximum value from it (6). At the early stages of the project, the model unifies different design domains and enables us to make design tradeoffs—so we find out very early on if the design will work. This brings verification to the start of the design process. We reuse the system model with cosimulation as a test harness during implementation and verification, allying the system model with implementation tools. This links the different design teams via a common platform that enables earlier and more detailed testing. Finally, we reuse the system model as a test harness and golden reference for verification, comparing the physical wireless prototype with the system model golden reference. This saves on test harness development costs.

Conclusions

System-level design brings more harmony to the wireless design process, enabling different design disciplines to play together more easily, thereby cutting down on project development time and risk. As the recession squeezes resources and we have to do more with less, these kinds of more integrated design flows will give those who adopt them a competitive advantage.

Bibliography

1. Quo Vadis, SLD? Reasoning about the trends and challenges of system level design. Sangiovanni-Vincentelli, A. 3, New York : Proc IEEE, 2007, Vol. 95.

2. Transmissions systems prototyping based on Stateflow/Simulink models. N. Papandreou, M. Varsamou, Th. Antonakopoulos. s.l. : Proceedings of the 15th IEEE International Workshop on Rapid System Prototyping, 2004.

3. Designing digital downconverters. B. Ogilvie, P. Pacheco. s.l. : RF Design, 2006, Vol. May.

4. IBS, Inc. Return on investment in Sumulink for electronic system design. 2005.

5. 3GPP. Terminal conformance specification, Radio transmission and reception (FDD). Valbonne, France : 3GPP. 34.121.

6. ASIC Design and Verification in an FPGA Environment. Dejan Markovic, Chen Chang, Brian Richards, Hayden So, Borivoje Nikolic, Robert W. Brodersen. s.l. : IEEE 2007 Custom Intergrated Circuits Confer, 2007.