

Concurrent Design of Adaptive High-Frequency Circuits

Dr. Michael C. Heimlich
Director, Microwave Marketing Segment
mike@appwave.com

Joel Kirshman
Manager for System Simulation
joel@appwave.com

Applied Wave Research, Inc.
1960 East Grand Avenue
Suite 430
El Segundo, CA 90245
www.appwave.com

Introduction

Adaptive circuitry is a popular avenue for extending the reliability, longevity, or parametric performance of many mission-critical systems. The design of today's adaptive circuits, when at high frequency and digitally controlled, requires consideration of design domains whose tools and methodologies have traditionally been partitioned, separate, and distinct throughout the design process, and have only been integrated at the final phases of development. This paper presents and discusses a new, fully integrated, concurrent design approach for adaptive circuits in communication systems. It describes how this new design flow concurrently brings together and enables real-time trade-offs among system, circuit, physical (layout), and software issues, and demonstrates that desired performance can be achieved more quickly and easily by simultaneously considering the disparate domains as opposed to the time-consuming traditional incremental methodology.

Components containing adaptive circuits offer much in the way of potential advantages to the systems utilizing them. A historically popular example for commercial communication systems is the use of predistorted power amplifiers (PPAs), whether by digital or other means, whereby a PA of more moderate capability is able to outperform those with greater linearity.

Today's adaptive circuits are even more challenging because of the technologies involved. The higher frequencies require that circuit design itself consider distributed effects, so much so that designs might best be layout-driven rather than designed in a traditional "top-down," schematic-based flow. Complex modulated waveforms, while not obsolescing multi-tone, frequency-based design, do not clearly correlate to linear and nonlinear frequency-based simulations; they require other techniques, such as shooting methods or envelope simulation. Finally, the adaptive component is more often than not a software-based design, adding yet another design domain, which if not included early in

the design process can cause design iterations due to “bugs.” Optimizing overall circuit performance can be difficult or impossible when the interactions among all of these design areas can not be considered simultaneously.

In this paper a simple example (Figure 1) of real-time concurrent design across these disparate design domains is shown. By way of a simple high-frequency amplifier, digitally-controlled attenuator, power sensor, and adaptive control algorithm, the example begins with a simple attenuator, sensor, and algorithm design that incrementally and concurrently improves the design. The need to create a more capable attenuator at frequency and power in order to couple to both the amplifier and the algorithm is demonstrated, as well as the fact that this design challenge is best-handled when considered concurrently, rather than serially.

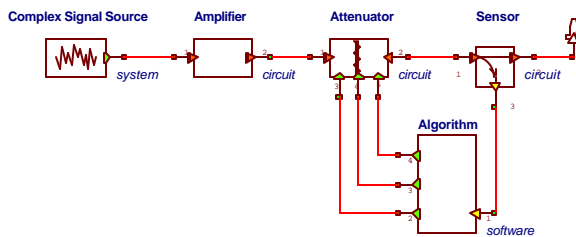


Figure 1 - Example Design - component shown in bold, portioning shown in italics

Serial Design Flow

A serial design flow is shown in Figure 2. The design proceeds in a top-down manner where the design is partitioned based on parametric budgeting into smaller and smaller pieces until the individual design components, specified by the budgeted parametric definitions, are realized and only then integrated. At this point, the second, debugging phase proceeds to identify interactions among the design domains and parts are redesigned for new parametric performance with the understanding that such performance minimizes or moves the interactions to a state where system performance is acceptable.

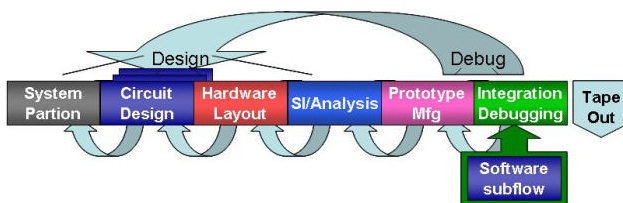


Figure 2 - Serial Design Flow – hardware, with software, design flow and iterations

The typical serial design flow is a consequence of the difficulties in moving data accurately and quickly among domains. Inherent in each step in the flow is a software EDA or CAD tool with its own database capturing the portion of the design represented by that step with the assumption that the whole database of the design is represented by the sum of the parts. This impedes or totally inhibits concurrency because at the heart of concurrency is the notion of interaction among disparate design domains.

Concurrent Design Flow

A concurrent design flow is shown in Figure 3. In concurrent design, the intention is to bring the debugging phase up into the earlier top-down phase, and with it, the software subflow. The debugging phase is enabled by considering software simulations, or so-called “virtual prototypes” of the individual hardware blocks and their integration. This provides at least the possibility during the design phase for the removal or mitigation of design problems normally uncovered only during the debugging phase. The enabling feature of the concurrent flow is a unified data model, versus the distributed data model in the serial flow depicted in Figure 2.

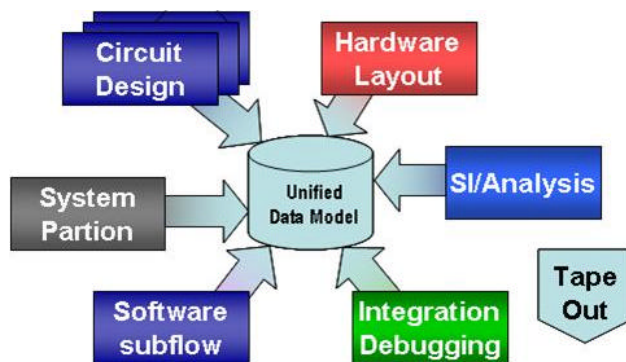


Figure 3 - Concurrent Design Flow – hardware/software co-design with virtual integration

Conclusion

The challenge for high-performance product design teams today is that design phases are traditionally isolated by separate EDA design and analysis environments, incompatible databases and use tools, and models that are not designed for the gigahertz frequencies at which next-generation electronics products are operating. The separate EDA environments and databases prevent designers from analyzing and optimizing circuit performance early in the design cycle, where it is most critical.

A new, highly integrated co-chip/package/module EDA solution is needed that is developed specifically to address the complex cross-domain issues inherent in the design of next-generation high-performance/high-frequency products. The software must be designed around a single, object-oriented database that is inherently synchronized with schematic, simulation, and layout data, enabling easy integration of all the design phases within one environment. The core technology must be a unified data model that is open

and flexible and enables interactive tradeoffs between system requirements and circuit implementation throughout the design cycle, enabling engineers to get their designs right the first time and shortening time-to-market.