The Life After SPICE

Ken Kundert

January **30, 2011**

This is an article I was invited to write by Larry Nagel for IEEE Solid-State Circuits Magazine for a special issue that commemorated the 40th anniversary of SPICE, which was published in spring 2011.

I graduated amongst the last wave of students that really focused on circuit simulation at Berkeley. Those students included Tom Quarles, who wrote SPICE3 as a way of improving the underlying architecture of SPICE, and Jacob White, who explored new numerical algorithms such as waveform relaxation as a way of accelerating traditional SPICE transient analysis. These students, like most of the ones that preceded them, accepted the basic capabilities of SPICE and were working to improve them by making them faster, more accurate, and more robust. My work was a bit different; I focused on teaching the old dog new tricks. I tried to develop new analyses specifically tailored for designers of high frequency circuits. In particular, I worked to develop methods for efficiently computing the steady-state response of a circuit. This work culminated in the development of a harmonic balance simulator that eventually formed the basis of the commercial microwave simulator from Hewlett Packard (now Agilent). After graduation I went in another direction. I took the harmonic balance simulator I wrote as a student to Cadence and converted it to a general purpose SPICE-like simulator. At the time HSPICE was the dominant commercial simulator for integrated circuits, and some of us at Cadence had aspirations of building a new simulator to replace it. I took the lessons I learned about simulator architecture from Tom Quarles and simulator algorithms from Jacob White and produced a new simulator named Spectre that was several times faster than HSPICE, as well as being more accurate and more robust (actually, Jacob and I working together completed the first version in two weeks). Essentially, I took everything that had been learned since the development of SPICE2 and put it into Spectre. It was not enough.

During this phase of my career I learned how difficult it is to replace an existing SPICE simulator. Few people seem to have learned this lesson as it seems like every year a new company pops up trying to do just that. SPICE certainly appears to be a tempting target. One can find serious faults in any SPICE simulator you try. Ironically, the faults are one of the things that make it hard to replace. Ask anyone who has ever tried to write an HSPICE parser. However, while backward compatibility is a daunting and often underestimated challenge, there is another aspect of these faults that conspire against those that attempt to dethrone the king. SPICE is used by analog designers, and analog designers take great pride in their ability to get the job done regardless of the obstacles. Designers who have used SPICE for years develop a kind of folk wisdom on how to overcome the problems they encounter. For example, when I worked at Tektronix as an intern I was told to randomly shuffle the order of the statements in my input deck if I had trouble with convergence.[1]

Designers take great pride in the knowledge that they accumulate at great personal cost over the span of their career. As such, they are loath to give up their simulator, regardless of how badly it has treated them. This is a form of Stockholm syndrome.

All this convinced me that I would need something big to get people to switch to Spectre. A speed-up of 2-3× would not do it, even when combined with better convergence and accuracy. These things were incremental. To get someone to switch we would need a “must have” feature that was clearly unique. After pondering this challenge for a while we decided to add behavioral modeling to Spectre. It was not completely unique as Analogy had been selling a simulator that supported behavioral modeling for years.

1. As odd as this sounds, it did occasionally work. And it seemed more palatable than the advice I received several years later from Larry Nagel, the creator of SPICE. He told me that if SPICE did not converge on a circuit, the problem was generally in the circuit itself and not the simulator. The circuit was probably not well designed and should be thrown out and re-designed.

However, Analogy's simulator, Saber, was not a SPICE simulator, and Analogy's language, Mast, was clearly a first-generation language and seemed rather cumbersome. Unfortunately, because of the personalities involved (mine included), the development was very contentious. However, we did develop a nice little language that we called Spectre-HDL and released it to the world.

During this phase of my career I learned how difficult it is to establish a modeling language, especially with analog designers, who seemed to like pictures more than text. As we were releasing Spectre-HDL, a new language under development as a standard was getting a lot of attention: VHDL-AMS. It would be years before VHDL-AMS would be available and it would require a full mixed-signal simulator, so it really was not a suitable mate for SPICE, but the idea of a standard language appealed to us. After all, clearly the ability to write your own model to include into your SPICE simulations was very useful; maybe people would warm up to the idea if we provided the ability in a standard language. That way they would be more comfortable learning the language because it would likely be around for a very long time and be used in many simulators. VHDL-AMS was not suitable, because it did not have an analog-only subset, and the developers of that language were not inclined to add one. So we came up with the idea of Verilog-A/MS: two closely related languages. Verilog-A would be an analog-only language that would be suitable for pairing with a SPICE simulator, and Verilog-AMS, would be a full mixed-signal language.

During this phase of my career I learned how difficult it is to establish a standard (and I thought developing Spectre-HDL had been contentious and difficult). We persevered and eventually released Verilog-A as both a standard and as a product. Unfortunately, it did not bring the success we had hoped for. Spectre still trailed badly in the market. Verilog-A was very useful, but acceptance would be slow, and by the time it was widely accepted by designers our competitors would also provide it.

So we looked around again for some compelling differentiator. The obvious choice was Verilog-AMS; after all it was new, and even more powerful than Verilog-A. But we were learning; and now we knew the meaning of the phrase “missionary sale”. We could see that the acceptance rate for Verilog-AMS would be even slower than that for Verilog-A. And frankly, I was tired, and could use a break. I was nostalgic for a simpler time. Perhaps I could go back to my thesis work, and add RF simulation to Spectre. I thought RF would be a very unique and nifty feature, and would be fun to develop, but it never occurred to me that it might be either popular or important. Maybe I could defer the quest for a killer feature until I had a chance to unwind.

At this time Jim Hogan was head of marketing at Cadence. I proposed the idea to him, and he asked one of his engineers to do the requisite due diligence on the market. That particular engineer had little experience in the analog market, let alone the RF market, a fact he freely admitted, but he did his best. At the time RF was not really a recognized market, and so he searched for a proxy. Things did not look good when he said that the closest thing he could find was the GaAs IC market, which, he pointed out, was both small and shrinking. At that point Jim looked over to me for some kind of response. The best I could manage was an encouraging shrug. So Jim looked back to his engineer, and then back to me, and said (I am paraphrasing here): “Oh what the heck, let's do it.” I look back to that meeting and still have no idea what Jim was thinking, but it was exactly the right decision as it finally gave Spectre a unique and compelling differentiator.

A key factor in the success of Spectre-RF was Jacob White, who, it turns out, will work for food. Jacob was a professor at MIT at the time (and still is for that matter), and looked for opportunities to come to California to satisfy his craving for good Mexican food, which was hard to find in Cambridge. I had done harmonic balance as a graduate student and so was familiar with the microwave simulators of the day, which were based on the simulator I wrote while at Berkeley. Those simulators had a tremendous head start in the market, and by this time I was understandably reluctant to take on the market leader head on. In addition, I knew that harmonic balance had inherent difficulties with convergence, and it would take a considerable amount of time to tame them. So I proposed the idea of trying shooting methods. Now, shooting methods were known to be an N^3 algorithm, meaning that as you double the size of the problem it becomes 8 times more expensive to solve. However Jacob had been having good success accelerating the parasitic extraction problem using Krylov methods, and was excited about applying those methods to the RF problem.

After many long days at Cadence and late nights at Rosita's, a local Mexican restaurant in nearby Alviso, Jacob was successful in implementing a prototype matrix implicit Krylov solver for shooting methods. Applying Krylov methods converted shooting methods from an order-N^3 algorithm to an order-N algorithm. This meant that we could handle much larger circuits than comparable simulators, which were generally limited to 10-30 transistors. Now it became clear that with our capacity and speed, we would be able to take on the established harmonic balance simulators and win even had we chosen to go with harmonic balance ourselves. However, going with shooting methods gave us several other important differentiators. Convergence was easy. We were able to release the product without having to develop any convergence aids, which are notoriously difficult to get right. Also, shooting methods handled sharply discontinuous waveforms much better than harmonic balance, which would allow us to eventually apply our new RF algorithms to analog circuits such as switched-capacitor filters and even to digital circuits.

We recruited Ricardo Telichevsky, one of Jacob's former students, along with Dan Feng to write most of the code for Spectre-RF. Meanwhile, Jacob led the effort to improve and extend the algorithms while I worked on directing development so that we solved real RF problems. Dan contributed a key insight that allowed us to use the gmres form of the Krylov methods, which provided better efficiency and robustness, and we all worked together to create the linear time-varying analyses, PAC and PNoise. Finally, Eric Filseth joined the team to market the product. He would come up with the most outlandish sales projections. By this time I had years of experience trying to crack the simulation market, and I knew that his projections were naive; growth would be slow, painfully slow. Once again, I was wrong about the market, but this time in a good way. Eric's projections turned out to be spot on. The wireless market was suddenly hot with the explosive growth of cell phones and wireless LAN.

Spectre-RF gained substantial credibility as a result of two Berkeley circuit designers. When Bill Mack presented his design of a transceiver at ISSCC, he was so complimentary of Spectre-RF, which he used to simulate the design, that he was scolded by the moderator. It was unusual for designers to say anything nice about their tools, and so when a well-respected designer did so in his presentation at ISSCC, people took notice. Then, Asad Abidi, a professor at UCLA, and his student Jacob Rael, used Spectre-RF to really explore and understand the fundamental mechanisms behind oscillator phase noise. Their work resulted in new way of building oscillators that eliminated an important source of flicker-noise from the close-in phase noise spectrum of the oscillator. This work was so successful that Spectre-RF was quickly adopted by many of the students at UCLA, and they in turn carried it into the many new design companies that were being formed at that time in Southern California. Normally fresh graduates do not have much say in which simulator a company uses, but in this case the companies were new and did not have established simulators and it was easy to justify the purchase of Spectre-RF because it did things no other simulator of the day could do.

Cadence counted revenue for Spectre-RF separately from that from Spectre. During this time revenue for both products grew rapidly; but of course, Spectre's revenue was always considerably larger than Spectre-RF's. There were two reasons for this. First, Spectre-RF was an option on Spectre, and every Spectre-RF sale was also a Spectre sale. Second, at the end of the day most of the simulations performed by analog design groups were still done with the traditional SPICE analyses. However, nobody wanted to use two different simulators, one for RF and one for analog, and Spectre was the only simulator available that could do both. So people bought Spectre because they needed Spectre-RF — Spectre-RF drove the sales of Spectre. In fact, it was so successful in doing so that Spectre soon eclipsed HSPICE to become the dominant circuit simulator for IC design. Oddly, the management team at the time failed to recognize that Spectre-RF was the reason why Spectre revenue had grown so substantially. What they saw in Spectre-RF was a product that was very difficult to support and that brought in relatively little revenue. As such they tried several times to kill the product, only to be forced to reverse course when customers revolted. Those managers are no longer at Cadence.

Today, both Verilog-A and RF simulation are very heavily used. In fact, both are considered so essential to the success of a SPICE simulator that it is fair to say that the concept of SPICE has expanded so that it now includes behavioral modeling with Verilog-A and RF simulation (either harmonic balance, shooting methods, or increasingly, both). I am proud of the role my friends and I played in bringing both to analog designers. The approach we took was very much aligned with, and inspired by, Don Pederson, Larry Nagel, and all of the people involved with SPICE and circuit simulation at Berkeley. In particular we followed the path they blazed by combining scientific curiosity, theoretical rigorousness, and a pragmatic focus on solving the real problems of circuit designers.

Today I am working on establishing Verilog-AMS. The simulators are available but adoption is slow; slower than I ever imaged it would be. However, Henry Chang and I believe that we have found the perfect application. Henry, another alumnus of the Berkeley CAD group, and I formed Designer's Guide Consulting. Our mission is to develop and promote the idea of analog functional verification based on building models and testbenches in Verilog-AMS. During this phase of my career I am learning how difficult it is to establish a new methodology. Once again I have misjudged the market, but I believe, just like Verilog-A and Spectre-RF, analog verification and Verilog-AMS will eventually become essential tools for analog designers.