Gerhard Klimeck
Network for Computational Nanotechnology

nanoHUB.org

Electrical Engineering Community
The user-to-user forum is for everyone, from design engineers to hobbyists, to discuss technology, products, designs and more. Join the discussions that match your interest or offer your expertise to others.

Join the discussion now at: www.digikey.com/techxchange

It’s all about connections.

The industry’s broadest product selection available for immediate delivery
# TABLE OF CONTENTS

## Gerhard Klimeck
**Director, Network for Computational Nanotechnology**
Interview with Gerhard Klimeck - Professor of ECE at Purdue University

## nanoHUB.org - Online Simulation and More
**By Gerhard Klimeck**
An introduction to nanoHUB, a global nanotechnology user facility.

## Featured Products

## Trading Off Performance and Code Space
**By Dave Lacey with XMOS**
How to optimize performance within embedded systems’ memory limits.

## S Parameter Causality Correction: A Dissenting View
**By Michael Steinberger with SISOFT**
Steinberger offers alternatives to S parameter causality correction.

## RTZ - Return to Zero Comic
How did you first become involved with nanotechnology?

After my third year of college, in 1988, I came to the U.S. as an exchange student from Germany and went to Purdue. In the spring of 1989 I took a class with Professor Supriyo Datta about electron flow in ultra-small structures where quantum mechanics are important. I realized that at some point in my career the down-scaling of devices will ultimately stop when the number of atoms in material layers becomes countable at the nanometer scale. This field was called quantum electronics or quantum transport—it was very exciting and challenging. I finished my Ph.D. in January 1994 and joined the first industrial research group with the label “Nanoelectronics” at Texas Instruments. There, we built the first industrial nanoelectronic modeling tool called NEMO.

What are your favorite hardware tools that you use?

- MacBook Pro
- Purdue Community Clusters, over 20,000 computing computational cores available
- Oak Ridge Supercomputer, Jaguar, over 225,000 cores
- Bandstructure Lab, Quantum Dot Lab, and RTDnegf for education and research

What are your favorite software tools that you use?

- NEMO – Nanoelectronic Modeling Toolkit
- nanoHUB/HUBzero software
- Adobe Illustrator
- Bandstructure Lab, Quantum Dot Lab, and RTDnegf for education and research

What is on your bookshelf?

Books about semiconductor physics, quantum mechanics, nanoelectronics, software development, programming languages—C, C++, Python, Tcl, and MATLAB.
Do you have any tricks up your sleeve?

How to structure public presentations of any length:

- Spend 1/3 of the time motivating the problem. Everyone in the audience should think that this is a critical problem to solve.

- 1/3 of time, show beautiful intuitive solutions which show insight and knowledge. There should be no technical details, but top-level insight. Everyone in the audience should feel embarrassed for not already working on the problem, because the solution is apparently so easy.

- 1/6 of the time, show technical details and why you are the expert in the room. Show that these problems in reality are very hard to solve, and you have the technical expertise to solve them. Do not show all the details for all the problems you solved, just pick one of the many. Leave all other details for backup slides that you can pull up in case you do get questions at the end of your presentation. Respond to questions like this: “Great question! Let me go to my backup slides and give you some detail.”

- 1/6 of the time, summarize the high level of your work and show your plans for the future. The audience should feel that “we should give him the money,” or “the job,” and cetera.

The piece of the presentation that in my opinion should be about 1/6 of the time usually takes on 95 percent of presentations and derails the true intent of the presentation, which is engagement of the audience. Most audiences are neither interested nor qualified to understand the technical details. Audiences want to hear about relevance and impact.

Following a nanoHUB presentation overviewing these concepts can be found at http://nanohub.org/resources/7615.

I most enjoyed transforming nanoHUB from a web form-based portal to a fully interactive simulation facility that serves over 10,000 users annually, with over 350,000 simulations.

What has been your favorite project?
The creation of nanoHUB.org as a global nanotechnology user facility. I most enjoyed transforming nanoHUB from web form-based portal to a fully interactive simulation facility that serves over 10,000 users annually, with over 350,000 simulations. Over 180,000 users come to nanoHUB to view lectures and courses on nanotechnology.

How did the nanoHUB project come about?

In about 1995, Professor Mark Lundstrom wanted to share a Unix-based simulation tool he built by his theory group with an experimentalist without rewriting it for a different computer. The idea to share this tool via web pages was conceived and the Purdue University Networking Computing Hub (PUNCH) was created, even before standard web servers were available. The technical development was performed by Nirav Karpedia under the supervision of Professors Jose Fortes and Mark Lundstrom. In 1998, the PUNCH system was serving about 1,000 users with about 30 simulation tools for research and education. That is also when the name nanoHUB was coined.

In 2002 the Network for Computational Nanotechnology was created and I joined as a technical director. In 2005 the web forms-based simulation tools were replaced by fully interactive simulation engines with friendly user interfaces. Michael McLennan was the core nanoHUB architect and creator of Rappture for the rapid development of user interfaces and data management. A completely new delivery system for fully interactive simulations was built by Rick Kennell.

Do you have any note-worthy engineering experiences?

Development of the first industrial nanoelectronic modeling tool (NEMO) that enabled quantum device simulation. This was done from 1994 to 1998 at the Central Research Laboratory of Texan...
Instruments in Dallas. At TI I also co-authored two U.S. patents on tunneling-based memory.

At Purdue, I have co-authored 34 nanoHUB tools that have now served over 20,000 users worldwide.

**What are you currently working on?**
Within my research group at Purdue, we are developing NEMO5—a generalized 3D, 2D, and 1D quantum transport simulation engine. Within nanoHUB.org, I’m studying the behavior of users to help support them better.

**What direction do you see your business heading in the next few years?**
My research will continue to support the downscaling and optimization of nanoelectronic transistors, plus the coupling of electronic devices to photons (optoelectronics, photovoltaics) and phonons (thermoelectrics).

I hope nanoHUB.org will grow further and manage data of simulation usage and also experimental data.

We are now creating ManufacturingHUB.org to support small manufacturing companies with modeling and simulation. I see that as a critical support for the regrowth of the U.S. manufacturing base.

**What challenges do you foresee in our industry?**
I see a substantial and ever-increasing shortage of U.S. citizens going to graduate school to study fundamental engineering. Even in other developed countries, one can see the same trend. This includes Germany, Japan, Korea, and even India. It will be a challenge to attract the brightest minds to this.

---

**Automotive, Medical, Telecom, POS**

**LCD for Any Application**

**Microtips Technology**

From design to service, Microtips offers a variety of competitively priced Liquid Crystal Display modules which includes standard character and graphic monochrome, passive and active color displays with white LED as well as custom LCD modules and complete OEM services.

For your own design needs please contact

**Microtips Technology:**
www.microtipsusa.com
1.888.499.8477
mtusainfo@microtipsusa.com
Avago Technologies

**Moonstone™ 3x1 Watt RGB High Power LED**

gives you a reliable, energy efficient, long life operation product for ease of design in any type of color mixing application.

**Moonstone™ Tri-color Power LED light source from Avago Technologies: ASMT-MT00**

Avago’s Moonstone™ Tri-color Power LED light sources are high performance energy efficient devices which can handle high thermal and high driving current. The exposed pad design has excellent heat transfer from the package to the mother board.

The advanced 3x1Watt RGB Moonstone™ package delivers typical 108 lumens at 350mA.

The robust package design makes it an excellent choice for color mixing applications.

**Applications**

- Architectural Lighting
- Decorative & Mood Lighting
- Retail Lighting
- Garden lighting
- Show Lighting

For more information or to request a sample please go to: www.avagotech.com/led

---

Your Imagination, Our Innovation
Sense · Illuminate · Connect
nanoHUB.org is funded by the National Science Foundation and supports the National Nanotechnology Initiative with a highly successful cyber-community for theory, modeling, and simulation now serving more than 181,000 researchers, educators, students, and professionals annually. In the past 12 months nanoHUB users performed over 388,000 nanotechnology simulations using more than 200 different simulation programs. nanoHUB.org is the world’s largest nanotechnology user facility.

nanoHUB.org hosts over 2,500 resources to help users learn about nanotechnology, including online presentations, full courses, learning modules, podcasts, animations, and other teaching materials. Most importantly, nanoHUB offers simulation tools that can be run directly from a web browser, allowing users to not only learn about, but also simulate nanotechnology devices. In addition, nanoHUB provides a collaboration environment via workspaces, online meetings, and group environments.

Resources come from 789 contributors in the nanoscience community, and are used around the world. Most of our users come from academic institutions and use nanoHUB as part of their research and educational activities, but we also have users from national labs and from industry. Many of our applications are devoted to nanoelectronics ranging from semiconductor device models to nanowire simulations, but we also have content focused on nanomechanics, nanophotonics, and nanobio.

The nanoelectronics simulation tools available on nanoHUB address quantum dots, resonant tunneling diodes, carbon nanotubes, PN-junctions, MOS capacitors, MOSFETs, nanowires, ultra-thin-body MOSFETs, finFETs, and other devices. The nanoHUB simulation facility is interactive, allowing users to set up a numerical experiment, view results, and easily compare different simulation runs and ask “What if?” questions. Computationally, the tools range from sophisticated industrial device simulation engines to simple MATLAB scripts that explore concepts. The user is not tasked with the setup of complicated input decks; rather, the tool capabilities are exposed through a graphical user interface created using our Rappture technology.

Figure 1: (a) nanoHUB.org simulation users. (b) nanoHUB cumulative simulation users and annualized “And More” users who view seminars, tutorials, and classes.
There are currently 215 simulation tools deployed on nanoHUB.org, with more being deployed regularly. We encourage the tool authors to supplement the tools by additional learning materials such as first time user guides and homework or project assignments that can be used in the classroom or for self-learning. Curated collections of tools and learning materials have been developed to function as a “one-stop shop” for a given topic area. An example of this tool-powered curriculum is the ABACUS package for semiconductor device education.

Researchers use nanoHUB simulation tools to explore concepts and assist in the selection of experiments to conduct. Some of the tools are open-source and the source code can be downloaded for inspection, self-installation, and modification. nanoHUB resources have accumulated over 700 citations in the scholarly research literature, and analysis of these citations clearly shows use of nanoHUB resources by computational researchers, by experimentalists working in the lab, and even educators. Due, in part, to the relative ease with which a tool developer can deploy a tool on nanoHUB, we see programs that may once have been utilized by only a small, local research group disseminated to a global community. Far more than a website, nanoHUB is a science gateway helping to connect a social network of scientists.

Figure 2: Historical monthly and cumulative nanoHUB.org user numbers. (a) Simulation users and (b) total users including the nanoHUB “and more” content consistent of seminars, tutorials and courses.

Figure 3: Collage of nanoHUB applications: Quantum Dot Lab, CNTbands, Bandstructure Lab. Users can interactively set up experiments and explore data without software installation.
8.5 GHz 4 Port VNA
The latest generation network analyzer from Rohde & Schwarz now also comes with four test ports and a second internal generator. Users who need to characterize multiport DUTs, mixers and amplifiers will benefit from the extremely wide dynamic range, short measurement times and exceptionally easy operation. The new four-port R&S ZNB models cover the frequency ranges from 9 kHz to 4.5 GHz or 8.5 GHz. Rohde & Schwarz has designed the powerful instruments for demanding applications in the production and development of RF components with multiple ports.

Two internal signal sources and a frequency-converting mode enable comprehensive measurements on mixers or amplifiers. Using mixed-mode S-parameter measurements, the R&S ZNB fully characterizes even balanced DUTs such as SAW filters used in mobile phones. For more information, please click here.

Precision Current-Sense Amplifier
Touchstone Semiconductor, a developer of high-performance analog integrated circuit solutions, today announced the immediate availability of two new drop-in replacements for its Maxim alternate-source products. These new current-sense amplifiers are pin-compatible, specification-identical and functionally identical to Maxim Integrated Products' analog ICs. Touchstone now offers seven analog ICs as part of its Maxim alternate-source product family. Touchstone’s alternate-source parts can be used in conjunction with Maxim products to ensure a constant supply of product so companies can meet shipping deadlines. All Touchstone alternate-source Maxim ICs are in stock and available to ship immediately. For more information, please click here.

TI Simple Switcher Nano Modules
Texas Instruments Incorporated today introduced four new SIMPLE SWITCHER® power management integrated circuits for space-constrained point-of-load designs in industrial, communications and automotive applications. The National 1-A LMZ10501 and 650-mA LMZ10500 nano modules with on-chip inductor and 2-A LMR24220 and 1-A LMR24210 nano regulators feature high performance and tiny, nano packaging as small as 7.5-mm2. Used in conjunction with the WEBENCH® online design tool, the products simplify and speed the design process. In a typical space-constrained design, TI’s new nano power modules, which support an input voltage range of 2.7 V to 5.5 V, can be designed alongside the nano power regulators to step down from an intermediate voltage rail to the point of load, providing a complete power management solution in applications with space and height limitations. For more information, please click here.
Trading Off Performance and Code Space

Dave Lacey
Technical Director of Software Tools

Usually embedded systems programmers and boxers do not draw many comparisons. However, in one aspect they do. Boxers need to fight within a particular weight limit—so their challenge is to maximize their performance (build muscle) within their specified allowance. Embedded systems are similar; they only have a limited amount of memory and you need to maximize performance within those limits.

Often, code optimizations that improve performance also reduce memory size (by reducing the amount of code that needs to be stored). With risk of overstretcing a rather tortuous analogy, perhaps this is similar to boxers shedding fat. However, some optimizations require a trade-off—it can go faster or it can be smaller. Making this decision is hard and depends on the application we are writing, but it is worth being aware of what optimizations fall into this category. This article covers some of the major space vs. speed trade-off optimizations we are likely to come across.

Tables vs. Calculation

Suppose we want to calculate \( \sin(x/256) \) for an unsigned 8-bit value \( x \) (i.e., the input represents values in the range 0.0 to 1.0). One method would be to calculate a polynomial approximation of the function. This is likely to be in the order of 5-20 instructions, depending on the architecture.

Figure 1: Space vs. Speed Trade-Off
Another option would be to have a lookup table. The size of this table will depend on the required output precision, but say we wanted 16 bits of output, then the table would be $256 \times 16$ bits = 512 bytes. This could reduce the calculation to 1 or 2 instructions but at the expense of memory.

In the above case it may be worth the extra memory if the speed is required for those sin calculations (remember – profile first, optimize later). However, in some sense this case is fortunate in that the input is only 8-bits. If the input were 16-bits then you would require 128KB of memory (more than on many microcontrollers).

The size of a lookup table goes up linearly with the number of bits in the output and exponentially with the number of bits in the input. So the number of bits in the input is the key factor.

For some algorithms, it is also possible to have hybrid techniques where we do both a lookup and some calculation. In these cases we can sometimes trade off the number of bits we use for lookup against the amount of calculation we do afterwards. An example of this is using a lookup to get an initial estimate of a function (e.g., $1/x$) and then using iterative refinement to get an accurate solution.

### Loop Unrolling

Loop unrolling is the transformation of a loop to a new loop whose body executes several iterations of the original loop. Consider the following code:

```c
for (int i=0;i<10;i++) {
  a[i] = b[i];
}
```

**Figure 2**

It can be rewritten as a loop that iterates only five times but each iteration does two operations:

```c
for (int i=0;i<10;i+=2) {
  a[i] = b[i];
  a[i+1] = b[i+1];
}
```

**Figure 3**

You could also fully unroll the loop to get this code:

```c
a[0] = b[0];
a[1] = b[1];
a[2] = b[2];
a[3] = b[3];
a[4] = b[4];
a[5] = b[5];
a[6] = b[6];
a[7] = b[7];
a[8] = b[8];
a[9] = b[9];
```

**Figure 4**

Why is this a good thing? There are a couple reasons why an unrolled loop (and particularly a fully unrolled loop) may go faster:

- Unrolling a loop can reduce the overhead of exit checks since there are fewer iterations. Furthermore, there are fewer branch instructions which may have an overhead depending on the architecture. In the case of a fully unrolled loop there are no exit checks at all.

- Unrolling a loop can give rise to other optimizations a compiler can perform (for example in the fully unrolled version above, all the array offsets are constants, which is something the compiler may be able to exploit).

Generally, compilers can do the unrolling for you. In the XMOS compiler you can control the unrolling of loops in XC programs by using pragmas. The following example will cause the compiler to unroll the loop 10 times:

```c
#pragma loop unroll 10
for (int i=0;i<10;i++) {
  a[i] = b[i];
}
```

**Figure 5**
The following is also equivalent:

```
#pragma loop unroll
for (int i=0;i<10;i++) {
    a[i] = b[i];
}
```

**Figure 6**

The unroll pragma with an argument will cause the compiler to fully unroll the loop if it can. This is particularly useful if the number of times a loop iterates varies with a #define but you always want to fully unroll the loop. Following is an example:

```
#pragma loop unroll
for (int i=0;i<NUM_ELEMENTS;i++) {
    a[i] = b[i];
}
```

**Figure 7**

### Function Inlining

Function inlining replaces a function call with the body of the function definition. As an example, consider the following two functions:

```
int f(x) { return (x+5); }
int g(x) { return (x + f(x)); }
```

**Figure 8**

The definition of g could be replaced with

```
int g(x) { return (x + (x+5)); }
```

**Figure 9**

by taking the body of f and replicating it inside g.

This method might cause a code size increase if the function being inlined is used more than once since the body of the function can be much larger than the call. The advantages of inlining a function are as follows:

- The register usage may be better.
- It can enable further optimizations.

Of these three, it is arguable that in most cases the third is the most important. Sometimes inlining is seen as a magic wand that can automatically make your code go faster, but often it can be marginal unless fitting the callee’s function code into the caller will allow a compiler to optimize more. Where it is often a big win is when the function being inlined is very small (in this case better register usage is likely).

C and similar languages (such as the XMOS derivative XC) have the inline keyword that can be added to function definitions. This function qualifier acts as a hint to the compiler that you would like this function to be inlined. It is just a hint though; compilers can inline functions without the keyword and can choose not to inline functions that have the keyword.

### Summary

This article shows three of the major examples of code optimizations that increase code size: using lookup tables, loop unrolling, and function inlining. Deciding when to trade off performance against code-size is one of the tricky parts of code development but it is important to be aware of when you can do it, how you can do it, and what the benefits are.

### About the Author

Dr. David Lacey works as Technical Director of Software Tools at XMOS Ltd. With over ten years of research and development in programming tools and compilation technology, he now works on the development tools for XMOS devices. As well as tools development, he has worked on application development for parallel and embedded microprocessors including work in areas such as math libraries, networking, financial simulation, and audio processing. ■
80V, 500mA, 3-Phase MOSFET Driver

HIP4086, HIP4086A

The HIP4086 and HIP4086A (referred to as the HIP4086/A) are three phase N-Channel MOSFET drivers. Both parts are specifically targeted for PWM motor control. These drivers have flexible input protocol for driving every possible switch combination. The user can even override the shoot-through protection for switched reluctance applications.

The HIP4086/A have a wide range of programmable dead times (0.5ms to 4.5ms) which makes them very suitable for the low frequencies (up to 100kHz) typically used for motor drives.

The only difference between the HIP4086 and the HIP4086A is that the HIP4086A has the built-in charge pumps disabled. This is useful in applications that require very quiet EMI performance (the charge pumps operate at 10MHz). The advantage of the HIP4086 is that the built-in charge pumps allow indefinitely long on times for the high-side drivers.

To insure that the high-side driver boot capacitors are fully charged prior to turning on, a programmable bootstrap refresh pulse is activated when VDD is first applied. When active, the refresh pulse turns on all three of the low-side bridge FETs while holding off the three high-side bridge FETs to charge the high-side boot capacitors. After the refresh pulse clears, normal operation begins.

Another useful feature of the HIP4086/A is the programmable undervoltage set point. The set point range varies from 6.6V to 8.5V.

Features

- Independently drives 6 N-Channel MOSFETs in three phase bridge configuration
- Bootstrap supply max voltage up to 95VDC with bias supply from 7V to 15V
- 1.25A peak turn-off current
- User programmable dead time (0.5µs to 4.5µs)
- Bootstrap and optional charge pump maintain the high-side driver bias voltage.
- Programmable bootstrap refresh time
- Drives 1000pF load with typical rise time of 20ns and Fall Time of 10ns
- Programmable undervoltage set point

Applications

- Brushless Motors (BLDC)
- 3-phase AC motors
- Switched reluctance motor drives
- Battery powered vehicles
- Battery powered tools

Related Literature

AN9642 "HIP4086 3-Phase Bridge Driver Configurations and Applications"

"HIP4086EVAL Evaluation Board Application Note" (Coming Soon)
With All Due Respect...

The causality of S parameter data is a mysterious subject to many, and so automatically correcting the causality of S parameter data seems like magic. This article suggests that while measures of causality can be useful in detecting problems, such problems should be solved by correcting their root cause rather than by making a superficial change using an automated procedure.

The basic requirement is that the response from a system or circuit must occur after the stimulus has been applied and not before. Failure to satisfy this requirement can cause errors ranging from minor glitches in time domain simulations to completely unstable SPICE simulations. Thus, engineers want to detect and correct causality errors in their S parameter data.

Those who have Penetrated the Mystery talk of the Kramers-Kronig relation and the ability that gives them to calculate the imaginary part of the data from the real part, and vice versa. To be sure, the Kramers-Kronig relation is a nice piece of math, and it does have its uses. If you really want to understand it, I recommend Colin Warwick’s excellent tutorial on the subject [1].

Unfortunately, the Kramers-Kronig relation doesn’t offer much engineering insight, and so the whole subject becomes the province of experts.

There are people I have the utmost respect for who have used causality correction, although I don’t know the exact nature of the errors they were correcting. There are people I respect who offer products which include causality correction; and in fact my own product, SiSoft’s Quantum Channel Designer™, includes causality correction because our customers asked for it.

Nonetheless, I have yet to see an application in which causality correction was more than a cosmetic fix to a larger problem, and I have yet to see an instance in which causality correction provided insight into the underlying problem. I therefore believe that causality...
correction isn’t a very good idea, and I’m going to offer alternatives that I think are more effective, especially in that they result in better engineering.

There are also S parameter quality measures that Yuriy Shlepnev has introduced [2], and one of those measures is for causality. While I believe these measures are well engineered and can be of some use, they don’t always flag problems, even in the most seriously flawed S parameter files. I will present an example, explain why the causality measure for this pathologically flawed data isn’t particularly alarming, and show how this example suggests other options.

I will answer these questions in the order 2, 4, 3, 1.

Is this data correct? No.

What makes this data erroneous? Following Eric Bogatin’s suggestion, let’s restate the question as: “This data isn’t what was expected. What was it in the data itself or our understanding of the system that caused the data to not be what we expected?” To make a long story short, the answer is shown in the phase plot below (Figure 2).

An Instructive Pathological Example

The pulse response above (Figure 1) is for an S parameter file that came in a bug report from a customer.

This impulse response is clearly not that of a well designed, properly analyzed channel; and yet the causality measure for this S parameter file is 96.83%, which is generally considered to be respectable. This raises a few questions:

1. Why is the causality measure for this file as good as it is?
2. Is this data correct?
3. If this data isn’t correct, how can one fix it?
4. If this data isn’t correct, what makes it erroneous?

One would expect the phase to be a relatively smooth function of frequency, and yet this phase seems to have some sort of stair step behavior to it. It’s not at all clear what the customer did to generate this data, but it’s not believable.

How can one fix this data? As described in [3], causal correction would only eliminate the spurious response on the right hand side of the pulse response shown above. The pulse response on the left hand side would remain, and that doesn’t look correct. Thus, causal correction is not a solution in this case.

The fix comes from understanding what went wrong in the first place. It’s evident from the phase plot that every other frequency point is more or less a duplicate of the point that came before it. The solution in this case is therefore to eliminate every other frequency point in
the S parameter file. The pulse response before and after the fix is shown in Figure 3. It appears that the fix was appropriate.

This type of fix is the norm rather than the exception. While causality correction could have been used to disguise the problem, the true solution had to come from a deeper understanding of the data. This is not something a tool can do for you automatically.

Why was the causality measure for this case as good as it was? That’s because the response had some delay to it to begin with. Thus, the erroneous response that occurred before the main response was delayed enough to still occur after time zero, and therefore by definition it wasn’t anti-causal. Only a small portion of the total response was actually anti-causal.

This example offers several valuable lessons:

1. Causality is a time domain problem. Look at the data in the time domain. For example, at least some VNA’s (Vector Network Analyzers) can display the data in the time domain. Also, EDA tools such as SiSoft’s Quantum Channel Designer™ convert from the frequency domain to the time domain. Check the tools and equipment you have on hand. Chances are that you can display the data in the time domain.

2. Combine circuit elements. If the data for an electrically short component such as a via has some modest portion of its response occurring before time zero, combine the data for that model with the data for an electrically longer component such as a PC board trace. That is, use frequency domain techniques to concatenate the S parameters before putting them into a time domain simulator such as SPICE. In a sense, the short component will borrow some delay from the longer component and the response for the two together will be causal and in all other respects well behaved. As with frequency domain to time domain conversion, there are many EDA tools that can combine S parameters in the frequency domain, SiSoft’s Quantum Channel Designer™ being one of them.

3. If there is a causality problem, fix the root cause of the problem.

References


About the Author

Michael Steinberger, PhD, has over 30 years experience in the design and analysis of very high-speed electronic circuits. Dr. Steinberger began his career at Hughes Aircraft, designing microwave circuits. He then moved to Bell Labs, where he designed microwave systems that helped AT&T move from analog to digital long-distance transmission. He was instrumental in the development of high-speed digital backplanes used throughout Lucent’s transmission product line. Prior to joining SiSoft, Dr. Steinberger led a group of over 20 design engineers at Cray, Inc. responsible for SerDes design, high-speed channel analysis, PCB design, and custom RAM design.
Join Today

www.eeweb.com/register
Hmm, this circuit's not working. Let's do the finger test...

Just touch the top of each part with your finger...

Yeeoww!...until you find the hot one...

This one? This one? This one? How about this one? What did you do??

First page of the manual. Before use, turn unit on using the on switch...

You mean things are still printed on paper?

Hold it right there, I need to do a screen capture.

Hurry up chip! I can't sit here all day! Err! I give up, this is ridiculous!

Say cheese!

Three years of work, all down to this moment... I'm flipping the switch in three... two... one...

I can't believe it!

Oh, sorry guys, I thought everyone left for the night...
RETURN TO ZERO

EVERYONE WRITE DOWN YOUR SALARY AND PLACE IT IN THE HAT...
THEN WE'LL MIX THEM UP AND SHARE THE RESULTS.

$15K
$65,000
$90,000

I ALWAYS THROW IN THAT HUM-DRUM TO MAKE THE GUYS JEALOUS OF EACH OTHER...

RETURN TO ZERO

Contact Us For Advertising Opportunities

1.800.574.2791
advertising@eeweb.com

www.eeweb.com/advertising