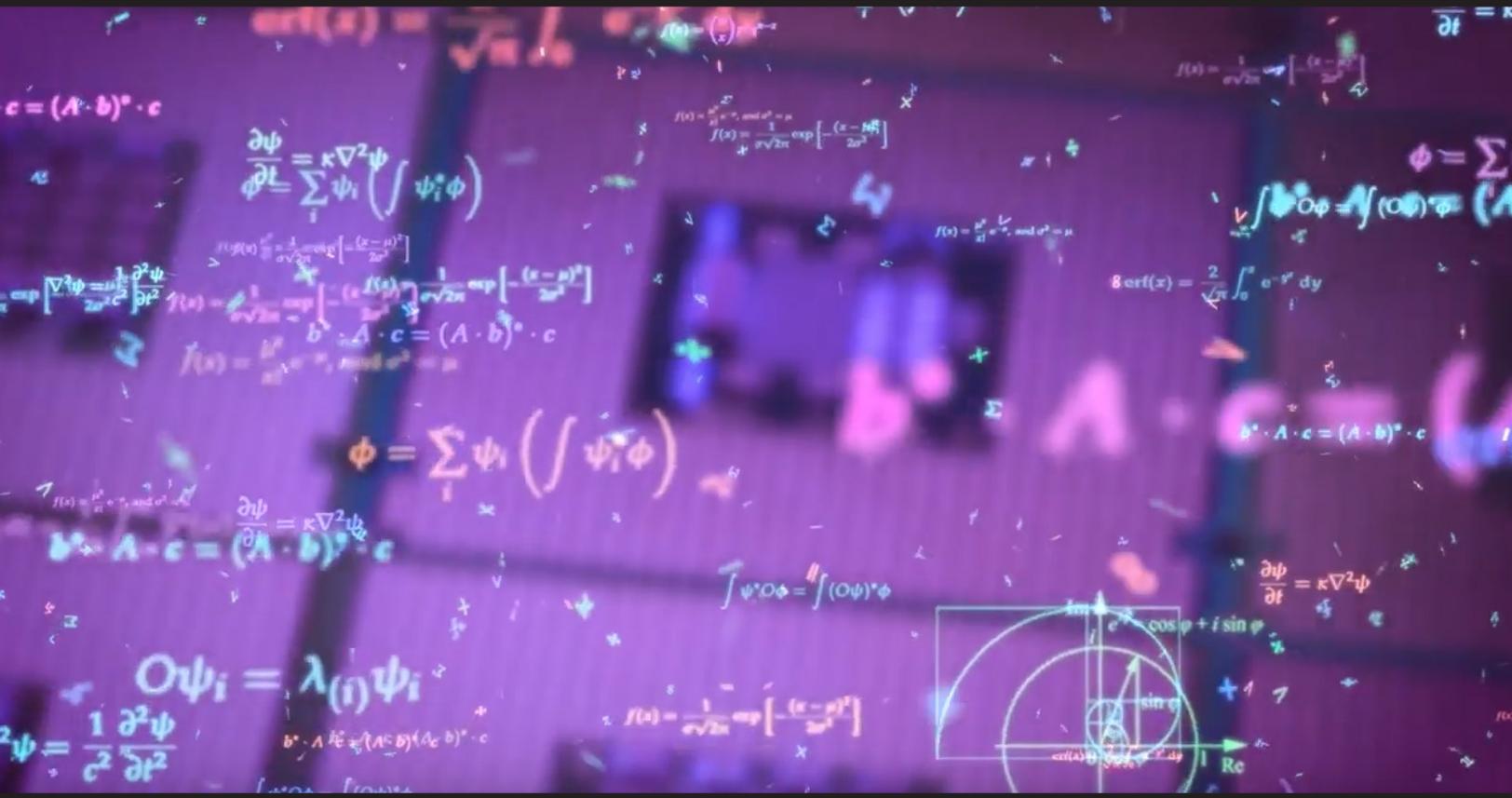


I-Connect007

GOOD FOR THE INDUSTRY



Cadence's Celsius:

“Don't End up Holding the Hot Potato!”

by Clive “Max” Maxfield

Cadence's Celsius: "Don't End up Holding the Hot Potato!"

by Clive "Max" Maxfield

I was just thinking about the party game Hot Potato, which involves players gathering in a circle and quickly tossing a small object such as a beanbag or even a real potato to each other while music is being played. The player who is holding the "hot potato" when the music stops is eliminated, after which the music resumes and the game recommences with the remaining players until only the winner remains standing.

As an aside, although the origins of the hot potato game are not clear, there is reference to a similar game in Sidney Oldall Addy's *Glossary of Sheffield Words* from 1888. This is particularly poignant for me because I was "born and bred in Sheffield," as they say. Amazingly enough, this book is still in print 132 years after it was first published. We can only wonder what Sidney would have thought about that, and I can only hope that my own books survive the test of time in such a robust manner.

The reason for my reminiscing on this game is that it reminds me of the thermal issues involved in designing an electronic product—anything from an integrated circuit (IC) to a PCB to a full-up system. In today's increasingly competitive marketplace, accurate thermal analysis must be performed, and any potential issues have to be identified and addressed as early as possible in the design cycle, otherwise the system will run into problems, market win-

dows will be missed, and someone will be left holding the hot potato. Trust me, you do not want to be that someone.

Happily, you don't have to be the poor soul to whom everyone else is casting aspersions (not me, you understand, because my throwing arm isn't what it used to be). The reason for my joy is that I was recently introduced to the Celsius Thermal Solver from Cadence, and this bodacious beauty can solve all of our hot potato problems (which isn't something I expected to hear myself saying when I woke up this morning).

As another aside, most people think Anders Celsius and Daniel Gabriel Fahrenheit invented the temperature scales bearing the names we know and love today. Most people would be wrong! For example, as I discussed in my *What the FAQ are Celsius and Fahrenheit?* column^[1], Anders initially started off with 0°C representing the boiling point of water, while 100°C represented the freezing point of water. It wasn't until a year after his death that other users decided to swap them over. If you are interested in this sort of thing, you might also wish to peruse and ponder my *What the FAQ are Kelvin, Rankine et al?* column^[2], which, amongst other things, provides a technique of telling the temperature by timing the chirps of common field crickets. But we digress...

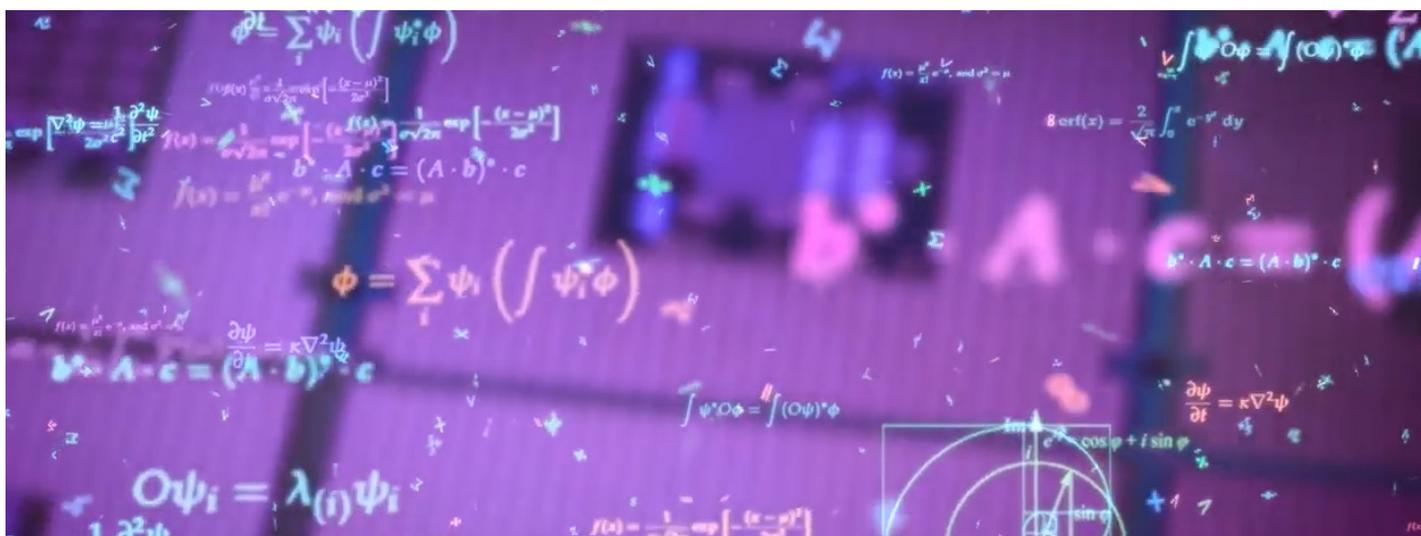


Figure 1: The complex interactions of electrical-thermal co-simulation require that components in the system be accurately modeled and analyzed in more detail than ever before.

If you know me at all, you will be aware that I can't see one of today's state-of-the-art tools like the Celsius Thermal Solver without taking a trip down memory lane to cogitate and ruminate over.

The Way Things Were

In my previous column, [Cadence's Clarity: 'I Can See Clearly Now...'](#), I discussed my first position after leaving university, which was at International Computers Limited (ICL) as a member of a team designing central processing units (CPUs) for mainframe computers.

I now appreciate that I was extremely lucky because ICL embraced the concept of a mentorship program. The idea was that each new engineer was assigned to work with an older, more experienced (mentor) engineer. My mentor was my team leader, Dave Potts. At that time, I thought Dave was much older than me. He was definitely much wiser, but, looking back, he was probably only in his mid-20s.

Dave was a wonderful instructor. The reason I'm where I am today (an internationally renowned technology superstar to hear my dear old mother tell the tale) is largely due to the way in which he guided me in my early design efforts. For example, my first task was to design an application-specific integrated circuit (ASIC) to implement a barrel shifter-rotator function that was capable of shifting-rotating 128-bit words by anything from 1 to 128 bits in a single clock cycle.

These ASICs, which were implemented at the 5-micron process node (that's three orders of magnitude larger than today's latest and greatest process node), were incredibly limited by today's standards. For example, each contained only 200 equivalent logic gates. This means that we needed to use a bunch of my ASICs to implement the shifter-rotator. Meanwhile, other members of the team were busily beaver away designing their own ASICs to implement their portions of the system.

If Dave had presented me with the entire shifter-rotator specification in one go, my brains would have leaked out of my ears and I would have had a nervous breakdown. Instead, he broke things down into a series of simpler tasks and built things up layer-by-layer, thereby guiding me to success and instilling a possibly mistaken sense of confidence and self-worth in me.

This was back in the mists of time we used to call 1980. The team of which I was a part was working on a mainframe computer called the S4L. Dave was a veteran from the previous 2900-series of machines, which were them-

selves predated by the 1900-series. I wasn't involved in the thermal side of things, but some of the older engineers told me that the 1900-series had employed a chilled-water forced-air cooling system. They also told me about an unfortunate incident in the early days of the program when the cooling system had passed the dew-point threshold causing moisture to precipitate out of the air. They described what ensued as being "like a tropical rainstorm inside the CPU cabinet," which resulted in the contents of the cabinet ending up as a pile of rust.

I must admit that many of the nitty-gritty details from those days of yore have faded from my memory. However, such are the wonders of modern technology that, even though I haven't seen Dave for 30 years, I tracked him down via LinkedIn and we enjoyed a transatlantic video call a couple of hours ago as I pen these words.

After reminiscing a little about the old days, I started to ask questions about some of the things I'd forgotten, like the sizes of the circuit boards and power supplies and suchlike. Dave told me that the power supply unit (PSU) for the 2980 computer he'd worked on, the predecessor to the S4L, had involved a motor generator that generated three phases at 415 volts, and that was located some distance away in a back room because of the noise it created. The three phases from the generator were chopped up by thyristors and smoothed by trays of capacitors (the PSU cabinet holding the thyristors and capacitors was 6 x 6 x 2 feet in size). The resulting supply was 5.2 volts at 1,800 amps. I just had to read those numbers again to make sure I wasn't dreaming. Amazingly enough, there was also a rheostat on the side of the PSU cabinet that could be used to adjust the voltage by $\pm 10\%$.

I'm not sure as to the PSU for the S4L because Dave and I both left ICL before the project was completed, but Dave says that it would have probably ended up looking similar to that for the 2980. In the case of the S4L, the ASICs we were designing were about 1-inch square, and each one dissipated between five and ten watts. The circuit boards were humongous by today's standards. We're talking about 2 x 3 feet in size and about 1/4" thick, with multiple power and ground planes and numerous signal layers. It's scary to think that these boards were plugged into even bigger backplane boards.

The S4L was also designed to use forced-air cooling. Our ASICs were implemented using emitter-coupled logic (ECL). Dave says we didn't have "hot spots" per se because everything ran hot. Each ASIC had its own heat sink with spines to dissipate the heat. The only way to

Cadence's Celsius: "Don't End up Holding the Hot Potato!"

probe one of the circuit boards when the computer was in operation would have been to use an extender card, which would have resulted in the board in question being out of the cooling airflow. The team leaders feared that the chips on the board on the extender would run so hot as to melt the solder and drop off the board. Although this would have removed them from the thermal equation and mitigated the thermal problem, it goes without saying that the performance of the computer and the quality of its calculations would have suffered.

As Rachel Caine, the American writer of science fiction, fantasy, mystery, suspense, and horror novels, said in one of her books, “God, it was hot! Forget about frying an egg on the sidewalk; this kind of heat would fry an egg inside the chicken.” I dare not think what she would have said about the S4L. Thankfully, as I mentioned earlier, Dave and I had left the project before it reached this stage.

Dave and I closed our conversation with me asking, “How were we planning on performing thermal analysis on the S4L?” In answer, Dave simply stuck the tip of his index finger in his mouth to wet it and then held it in the air in a traditional “Let’s see which way the wind is blowing” gesture. The sad thing is that I wasn’t surprised.

Celsius Lets Us Turn the Temperature Down

I must be feeling in a literary mood at the moment because I’m reminded of a line in the book *Glass Sword* by American author Victoria Aveyard which goes, “The thing with heat is, no matter how cold you are, no matter how much you need warmth, it always, eventually, becomes too much.” I cannot but help feel this is apposite to our discussions here.

As I said in my recent article, “As every electronic design engineer on the planet knows (unless they’ve been living under a rock), Cadence has state-of-the-art design and analysis tools for every portion of the design flow—chip design, package design, and board/module design. What is less well known is that, for the past few years, Cadence has been ramping up its capabilities in system-level simulation, analysis, and verification space (where no one can hear you scream).”

I also noted that, “One underlying problem is that many of today’s software analysis tools were conceived and developed in the era of single-core computing, which means they simply don’t scale well, even if they are run on systems that have multiple cores with multiple threads running on each core. To address this problem, the boffins at Cadence started with a clean slate and created a distributed multiprocessing technology that was designed from the ground up to take full advantage of multiple cores—both central processing units (CPUs) and graphical processing units (GPUs).”

All of which leads us to the Cadence Celsius Thermal Solver, which takes full advantage of Cadence’s distributed multiprocessing technology to deliver virtually unlimited capacity and 10X the speed of legacy 3D finite element analysis (FEA) solvers while maintaining gold-standard accuracy.

In addition to heat sources, it’s also necessary to model the conduction, convection, and radiation of heat throughout the system. In turn, this means it’s necessary to model air flow inside and outside the system while understanding that the dynamics of this flow, and its capacity to remove heat, will be modified by the thermal envi-

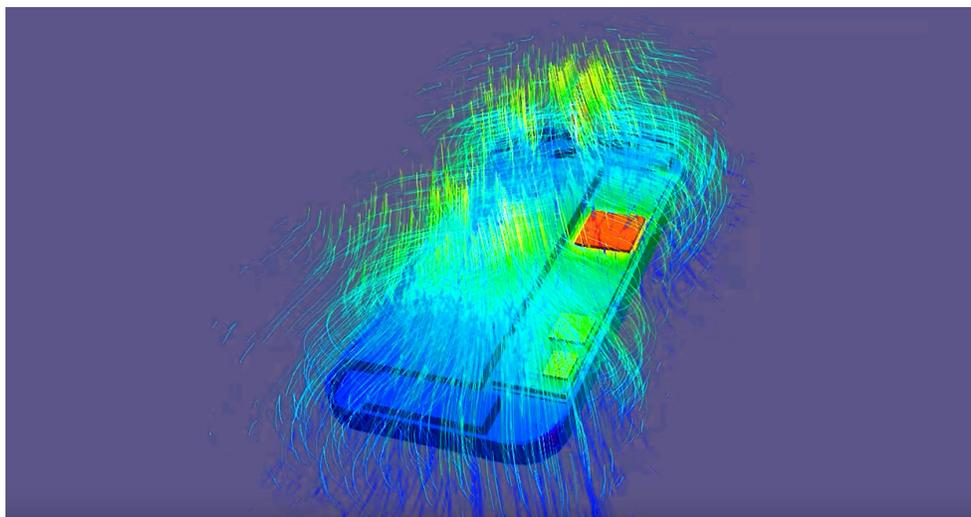


Figure 2: The Celsius Thermal Solver combines finite element analysis and computational fluid dynamics algorithms to perform both steady state and dynamic analysis.

ronments it encounters. Thus, the Celsius Thermal Solver combines its finite element analysis of solid structures with computational fluid dynamics (CFD) analysis for the air flowing around and through the system, thereby providing complete system thermal analysis in a single tool.

The Celsius Thermal Solver is a true 3D solution that can be used to perform both static and transient 3D thermal analysis on ICs, 3D-ICs (that is, chips composed of multiple dice stacked on top of each other), IC packages, system-in-package (SiP) components (which involve multiple IC and/or 3D-IC dice mounted on a common substrate), modules, and PCBs. Furthermore, Celsius can also be used to analyze enclosures, including chassis, racks, and entire systems.

This is quite a feat when you think about it because it involves the ability to work over 10 orders of magnitude, from nanometers (10⁻⁹) to meters (100). There are several factors that make this work, including the fact that Celsius employs an automatic adaptive mesh, which means it adds or removes mesh elements as necessary to provide the required level of accuracy. That is, the automatic adaptive mesh will refine the mesh density in the regions having large temperature gradients so as to obtain more accurate results, while relaxing the mesh density in those regions where the temperature gradients do not vary so dramatically.

There's also the fact that Celsius scales in an almost linear fashion using an "elastic compute architecture." What this means is that, if you have only a single core, a simulation will take a certain amount of time. With two cores, the same simulation will take half the time. Where are all these cores coming from? Well, you can use your own on-premises distributed computing solution if you wish. Alternatively, by making their tools cloud-friendly, the folks at Cadence have provided the option for essentially unlimited scaling, which means essentially unlimited capacity.

Something else that must be considered is the classic "chicken or egg" problem (i.e., which came first?). In this case, the electrical and thermal aspects of the system are interrelated and interact with each other; temperature affects electrical resistance, and every element of electrical resistance introduces an additional thermal source. We can't nail down a good thermal analysis until we've locked down the electrical design, and we can't tie the electrical design down until we've performed our thermal analysis. The solution is to perform electrical-thermal co-simulation; that

is, to perform electro-magnetic analysis using Clarity in conjunction with thermal analysis using Celsius.

Celsius and Clarity can both import mechanical structures from all major MCAD tools (thereby facilitating the modeling of enclosures). Also, both tools can easily read design data from all standard chip, IC package, and PCB platforms. Furthermore, Celsius and Clarity both offer unparalleled integration with Cadence's own tools, such as Virtuoso Layout, SiP Layout, Allegro PCB Designer, and Voltus static and transient power profiling.

In the olden days, a traditional scenario was to work all the way through the development of a system, have the end in sight, discover that something was running too hot, get together with the thermal analysis team, track down the problem, and return to the drawing board with much gnashing of teeth and rending of garb.

Today, the ability to use tools like the Celsius Thermal Solver in conjunction with the Clarity 3D (FEM) Solver and the Clarity 3D Transient Solver early in the design cycle—alongside the IC, package, PCB, and enclosure tools—means that none of us are going to be left standing alone holding the hot potato. Instead, we can break out the party hats and perform our happy dance.

References

1. Max's Cool Beans blog, [What the FAQ are Celsius and Fahrenheit?](#)
2. Max's Cool Beans blog, [What the FAQ are Kelvin, Rankine et al?](#)



Clive "Max" Maxfield is the founder of Maxfield High-Tech Consulting and the author of a variety of books, including "Bebop to the Boolean Boogie." He has been at the forefront of EDA for over 30 years. To check out Max's blog "Cool Beans," [click here](#).



Note: This article originally appeared online December 17, 2020 at PCBDesign007.com. It is reprinted here with permission in its entirety by BR Publishing, Inc. dba I-Connect007.com.