

Technology Brief 11 Circuit Simulation Software

In Chapters 2 and 3 we examined all of the common methods used for analyzing linear electric circuits. In practice, these are used for designing and analyzing the many building blocks that make up larger circuits, or for obtaining approximate solutions for how more complex circuits function. In Technology Brief 1, we noted that **very large scale integrated circuits** (VLSI) have experienced exponential scaling for almost 50 years, so some of today's electrical networks may include as many as 100 billion transistors! The standard circuit analysis methods available to us are accurate and applicable, but it takes a great deal of computer automation to apply them to a 100 billion-transistor network. The **Multisim** circuit analysis software provides an excellent start towards modeling the behavior of complex circuits. Accordingly, Multisim will be the first of two computer-based tools we will explore in this Technology Brief. Whereas Multisim is an excellent tool, it treats a circuit as a 2-D configuration, which does not account for thermal effects associated with heat generation by the circuit elements, nor possible capacitive or inductive cross-coupling of voltages between elements (through the air or insulator medium between them). To account for these effects, we need to use a sophisticated **3-D computer simulation tool**. This is the subject of the second part of this Technology Brief.

Multisim Software

(1) Using Simulation Tools to Calculate and Understand

Engineers use **electronic design automation** (EDA) tools, such as Multisim, to understand the function of a circuit and calculate its response. Consider the simple example shown in **Fig. TF11-1(a)**, and let us assume we need to determine what voltage V_T would be measured by the voltmeter shown in the circuit. In this case, because the circuit is very simple, we can analyze it by hand or we can implement it and solve it by Multisim (**Fig. TF11-1(b)**). But if the circuit has more than five nodes, the by-hand approach becomes tedious, and the Multisim option becomes far more practical.

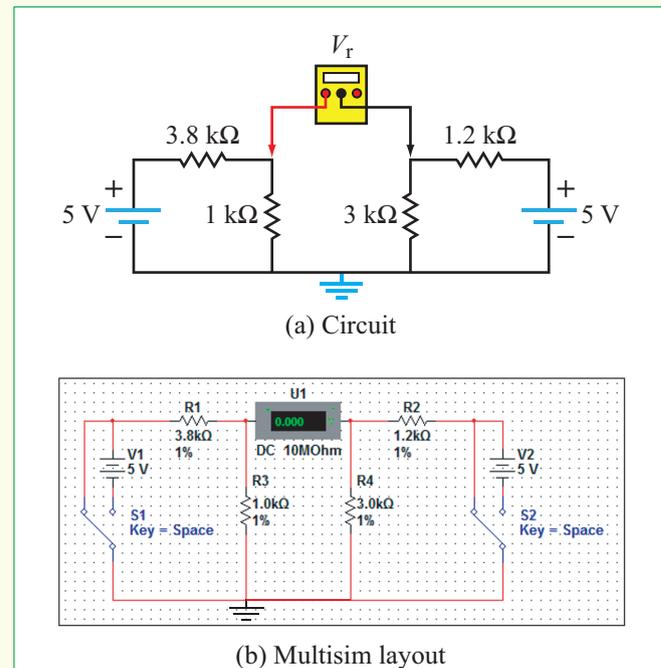


Figure TF11-1: Two-source circuit and Multisim representation using switches to switch one or both voltage sources on or off.

(2) Using Simulation Tools to Lay Out a Circuit

Once a circuit has been designed, we can either build it on a protoboard or, alternatively, we can have a circuit board built for it and then solder the parts to the board to create the circuit. **Printed circuit board** (PCB) layout tools help us plan the circuit layout and routing architecture, which often are multiple layers deep, as in the circuit of **Fig. TF11-2**.

When using silicon chips, for example, these designs involve hundreds, millions, or trillions of components arranged in one or more layers, and carrying thousands of simultaneous signals throughout the circuit, all acting together to obtain the desired voltage and/or current output of the circuit. Classic EDA tools (such as Multisim) begin with a **graphical user interface** (GUI) that allows users to specify what type of circuit elements (sources, resistors, switches, etc.) are needed and how they are connected together. Circuits made up of several elements can often be grouped or bundled together and stored in **libraries** for later reuse. Often, libraries of complex

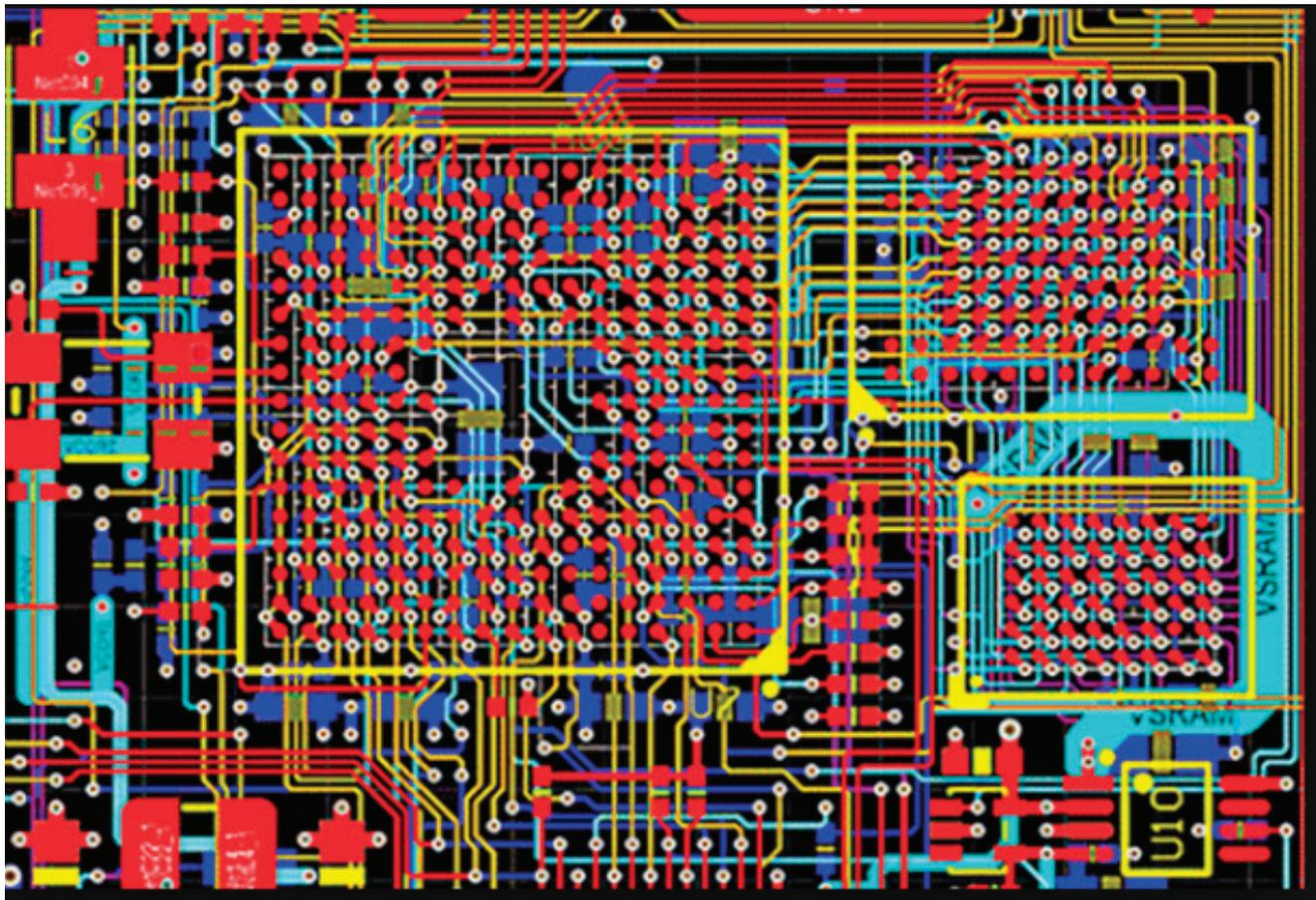


Figure TF11-2: Multilayer PCB layout, with each layer assigned a different color. Holes and solder pads are planned for each chip and component attached to the board, and multilayer routing built into the circuit board connects them all together. (Courtesy of ZYPEX Inc.)

circuits (such as the core of a computer processor) are shared or purchased to reduce engineering design time. For circuits whose design can be expressed as either logical rules or a desired logical function—primarily digital circuits—modern software tools transform circuit design into an exercise in writing code. In essence, programs can be written in **hardware description languages** (HDL), which define the structure and/or operation of digital circuits. The program is then executed and a circuit description suitable for manufacture, or instantiation into a field-programmable gate array (FPGA), is synthesized. Programming in HDLs is similar to assembly language or C coding, although major

differences exist. Most modern complex digital circuits are designed, simulated, and synthesized with the aid of HDL tools.

Once the elements and their connections are defined, they are then modeled with either more or less detail (by specifying tolerance levels or other relevant parameters) depending on the level of accuracy needed. Simulation results are only as good as the circuit model and input parameters, so this is a very important consideration when using EDA software. The more detailed the model, the more accurate the results can be expected to be, but also the longer it takes the simulation to run. Consider, for example, the ideal and the more realistic models

for voltage and current sources listed in [Table 1-5](#). The realistic source models are certainly more accurate than the ideal models, but even the “realistic” models are approximate, because they neglect nonlinearities, stray capacitance and inductance, and potential feedback loops within the sources. For many applications, the ideal model is sufficient, for others the first-order realistic model (including a resistor) is sufficient, but for others, a more detailed nonlinear model is required. How do you, the engineer, know what model to use? The intuition and knowledge gained from working with the common circuit analysis tools from Chapters 2 and 3 help you determine when you may or may not need a more realistic model. Often, we will first try a simplified model, and then one that is slightly more realistic. If there is minimal change, we do not go on to a more complex model, but if there is substantial change, we may try more and more realistic models (each requiring more time and memory for the software to run), until the result converges and we are satisfied that we have modeled the real system at hand.

Now let's consider VLSI circuits involving trillions of transistors. Even with relatively simple models of the transistor (such as the BJT in Section 3-9 or MOSFET in Section 4-11), there are still more unknowns than we generally care to wait for the computer to solve. In this case, two simplifications are essential. First, we must break the circuit down into **functional blocks**, so we can design each block individually and cascade or connect the blocks together. We have already seen simple examples of doing this using the Thévenin equivalent circuit technique. Thévenin is also used this way in much larger circuits, including VLSI designs. Second, we must simplify the models we use for each circuit element. Fortunately (or perhaps necessarily!) the largest circuits electrical engineers design are digital circuits, for which we can use the simplest models of all. We can assume that all voltages are either **high/on** (digital 1) or **low/off** (digital 0). This flexibility in the voltages allows us to use much simpler models. The transistor, for example, can be modeled as just a switch (on or off), or just as a resistor that is switched in or out of the circuit. Assuming all voltages are either on or off is the simplest assumption. We also can model them as on/off or in transition between on and off. The transition (which is actually a **bouncy switch**) can be modeled as a linear slope from low to high or high to low. The length of this slope is the **rise time** of the transition, and the faster the rise time, the faster the circuit can send data.

3-D Modeling Tools

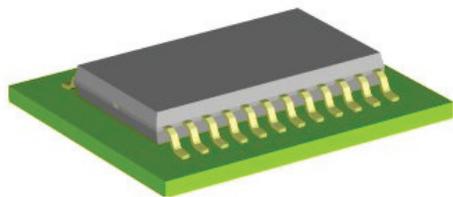
Model-based EDA tools define how a circuit is supposed to function electrically, but sometimes effects not included in the models come into play to make the circuit malfunction. Two of these that are particularly relevant are associated with thermal problems and coupling problems. We know that resistors and other devices are designed with specific power ratings. The power rating is related to the size and material the resistors are made of and their ability to withstand the heat generated by current moving through them. If we start pushing all of the elements of the circuit to their maximum capability, their interactions (hot chips next to other hot chips) may make the most vulnerable of these parts fail. But how do we determine which parts are the most vulnerable, and what solution can we offer to mitigate the heat problem? 3-D simulation tools help us to identify these potential problems or (all too often) diagnose them when they occur. The 3-D simulation process starts with the physical model of a given part, such as the high-speed IC package shown in [Fig. TF11-3\(a\)](#). The spatial distributions of electrical voltage and current are then modeled for part or all of the package, as shown in [Fig. TF11-3\(b\)](#). The current density at a given location is representative of what the temperature will be at that location. If overheating were to occur, it would most likely occur at the points with the highest current. More detailed thermal modeling can include the effects of heat sinks, fans, and other cooling effects. The voltage is used to calculate coupling between nearby electrical signals (such as two adjacent legs of this package).

Another interesting circuit simulation is shown in [Fig. TF11-4](#), which displays the amount of power radiated by a crescent antenna.

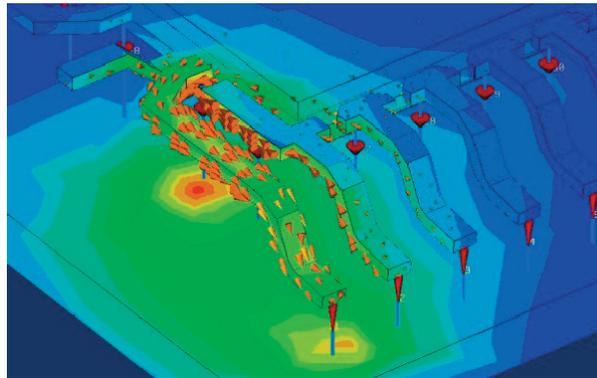
So WHY Should You Learn the Circuit Analysis Methods Introduced in This Book?

Having learned how to apply the various circuit analysis tools covered in this book thus far, you may wonder why you need to learn so many different methods when they all can give you the same result. And now that you have read this Technology Brief and seen that you can use a computer to analyze circuits, you may wonder why you need to learn these analytical methods at all!

While it is true that automated tools are essential for testing circuits used in practical applications, it is equally true that the success of the design process is highly coupled to one's understanding of the fundamental



(a) Physical package



(b) Current density contour

Figure TF11-3: High-speed IC package and contour and vector plot of the current density flowing through it at 5 GHz. The brighter/redder colors show higher current density (A/m^2) (which also results in higher temperature) than the darker/bluer colors. The arrows show the direction in which the current is flowing, and the size of the arrow is also proportional to the magnitude of the current density. (Courtesy: CST MICROWAVE STUDIO[®] IC Package Simulation.)

concepts in circuit analysis and design. Designing a new circuit to address a specified application is a creative endeavor that relies on one's past experience and fluency in circuit behavior and performance. Once an initial circuit

configuration has been developed, computer simulation tools are then used to fine-tune the design and optimize the circuit performance.

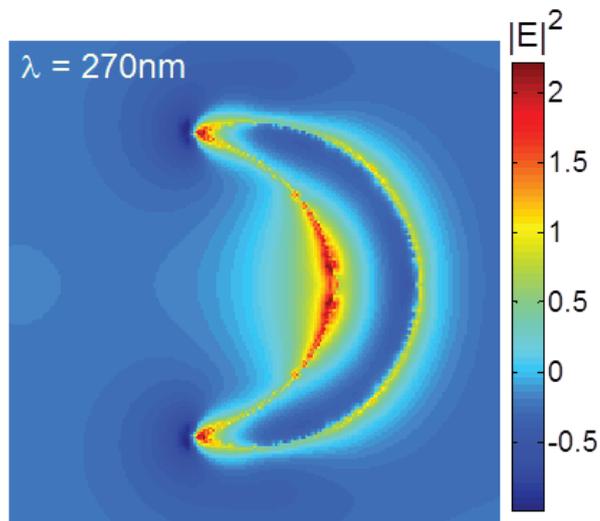


Figure TF11-4: This 3D electromagnetic simulation was used to evaluate the fields (in this case the square of the electric field, which is proportional to power) in the nanocrescent antenna shown in Technology Brief 1. We can see the strong fields at the tips (because charge congregates there), and also in the center. (Credit: Miguel Rodriguez.)