Practical Aspects of Modeling Complex Analog Behavior in Modern Circuit Simulators

Marek Mierzwinski

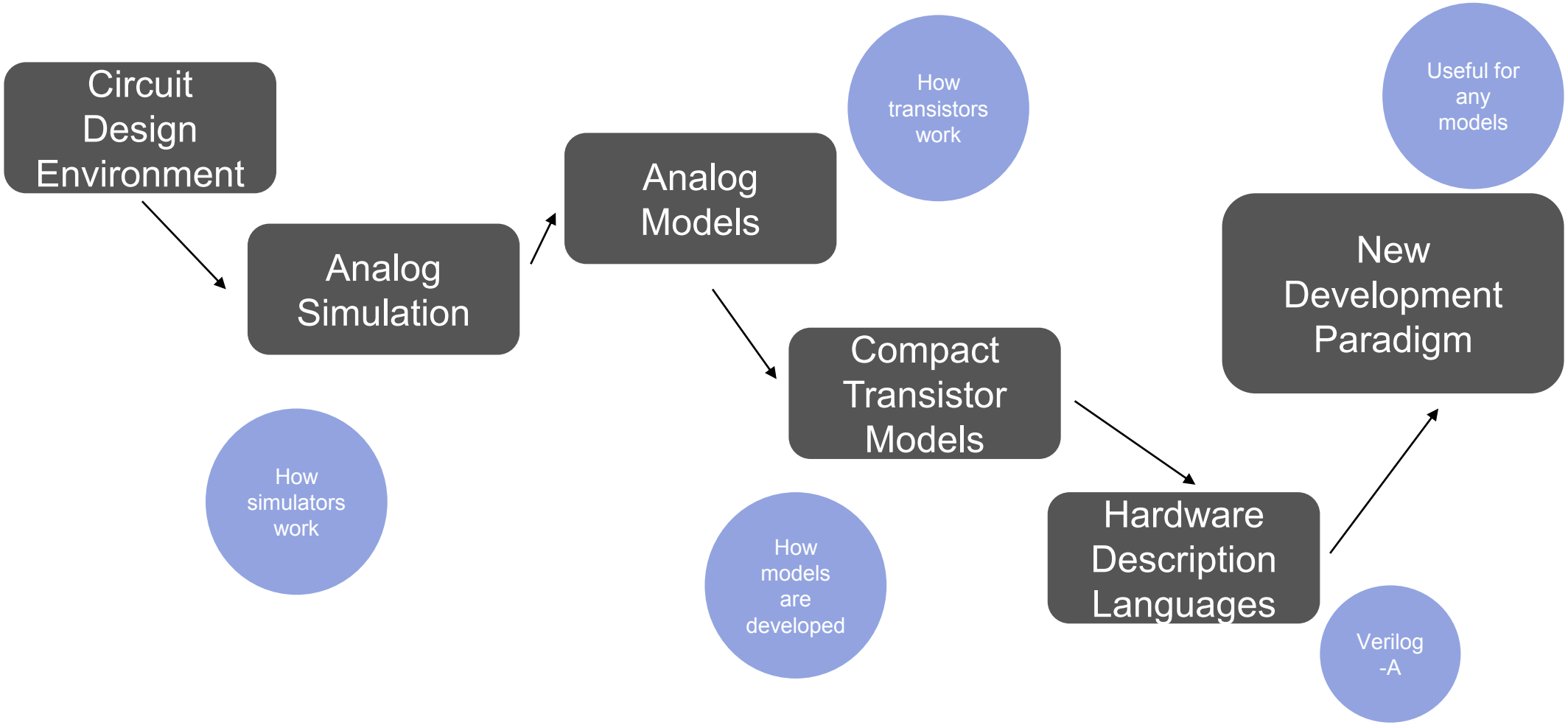
KEYSIGHT TECHNOLOGIES

What I hope you'll learn

- How the overall circuit design process works
  - But how it's similar to most other engineering endeavors
- How simulators work
  - And the models are the critical piece
- Getting a model into a simulator has become very simple
  - New programming languages have made it possible
  - New models can be implemented by individuals
- A bit about Keysight Technologies

Agenda

Or at least a guide



KEYSIGHT TECHNOLOGIES

A little background on Keysight Technologies

KEYSIGHT TECHNOLOGIES

History

- The original test and measurement divisions of Hewlett-Packard
- HP spun off Agilent Technologies in 2000
- Agilent Technologies spun off Keysight Technologies in 2014
- Corporate headquarters in Santa Rosa

KEYSIGHT TECHNOLOGIES

Around the World, We're Ready to Help You Change It

150 locations

Conducting business in more than 100 countries

12,200 employees

ASIC design center and proprietary fabrication facility

Technology centers for MMICs, optical components and microelectronic packaging

Over 1,000 patents

>4,000 products

R&D centers in 13 countries around the world

40 service hubs

KEYSIGHT TECHNOLOGIES

New Keysight R&D Center at Georgia Tech

- Keysight has opened a new Atlanta-based Software Development Center.
- Will house >200 software developers & engineers over next 5 years.
- > 50 new engineers hired.

KEYSIGHT TECHNOLOGIES

Circuit Design Environment

KEYSIGHT TECHNOLOGIES

The Problem

Even something as basic as an amplifier has multi-technology design challenges

Complete front-to-back GaAs/Silicon MMIC design

Multiple IC's on different fabrication technologies

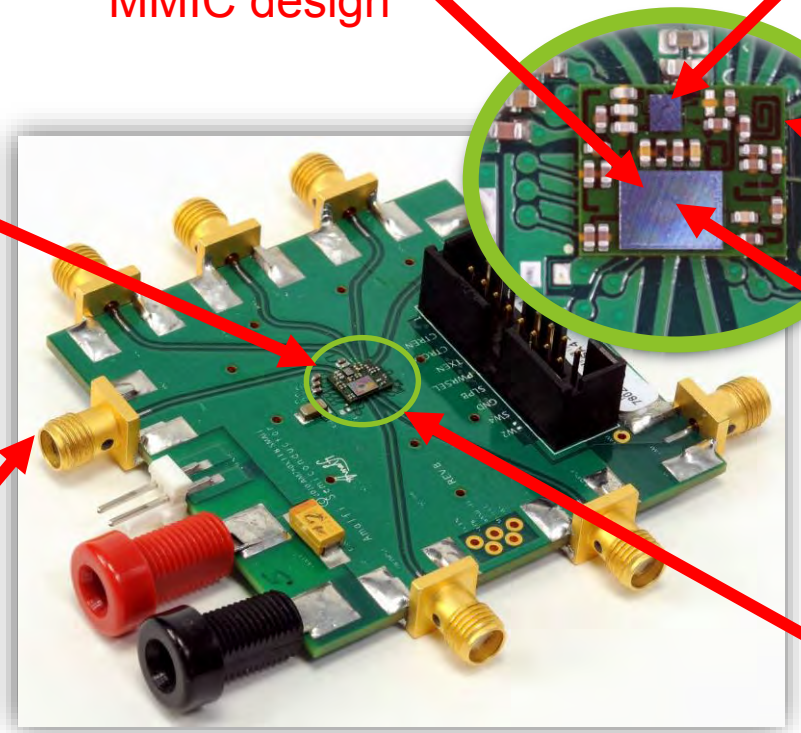
Integrated full 3D FEM to model package, solder bumps, bond wires

Design & EM simulation of entire laminate

Electro-Thermal MMIC inter

3D EM Components

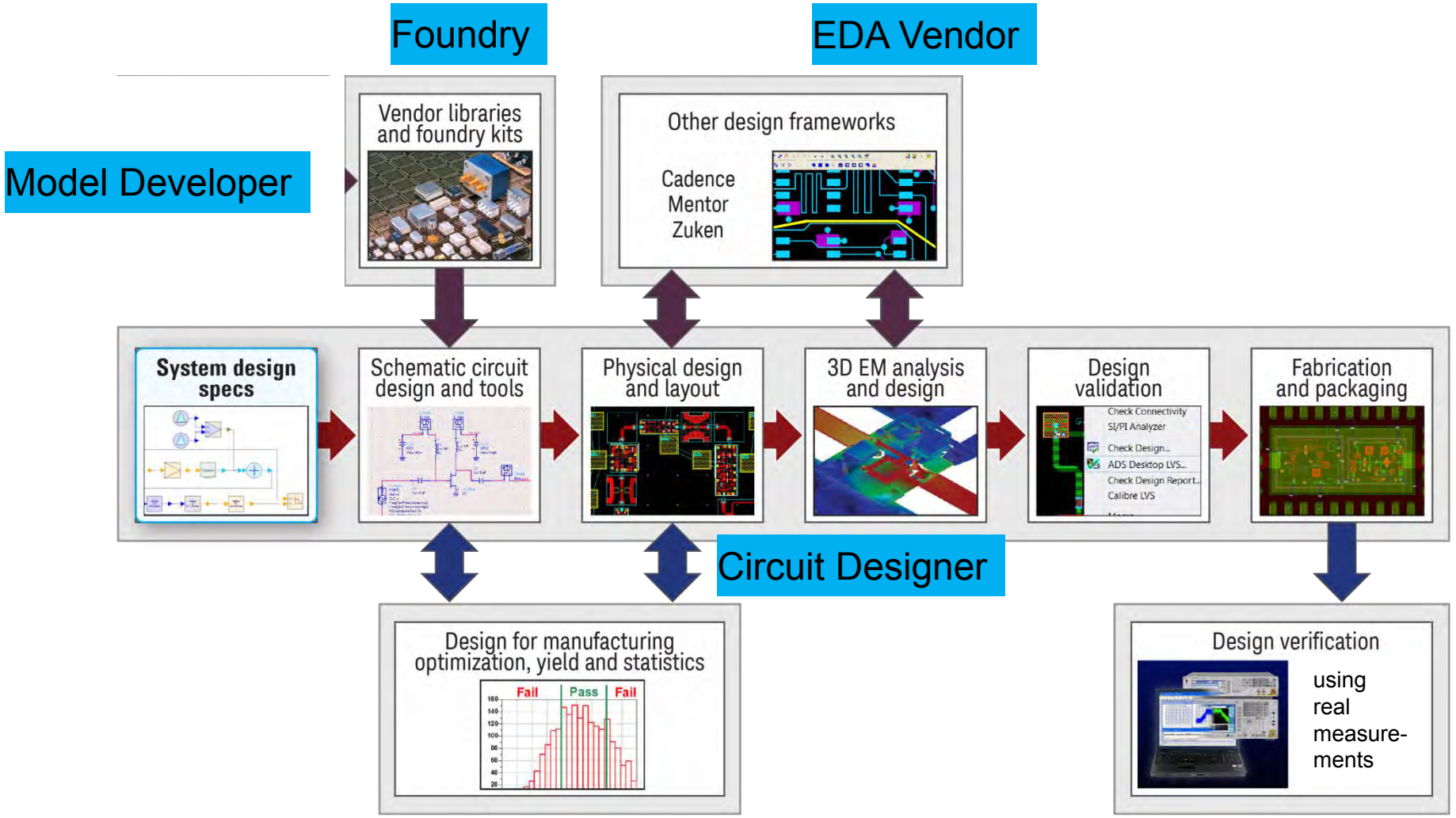
Design, layout and verify IC, laminate, and PCB in a single environment



RF Power Amplifier Module

KEYSIGHT TECHNOLOGIES

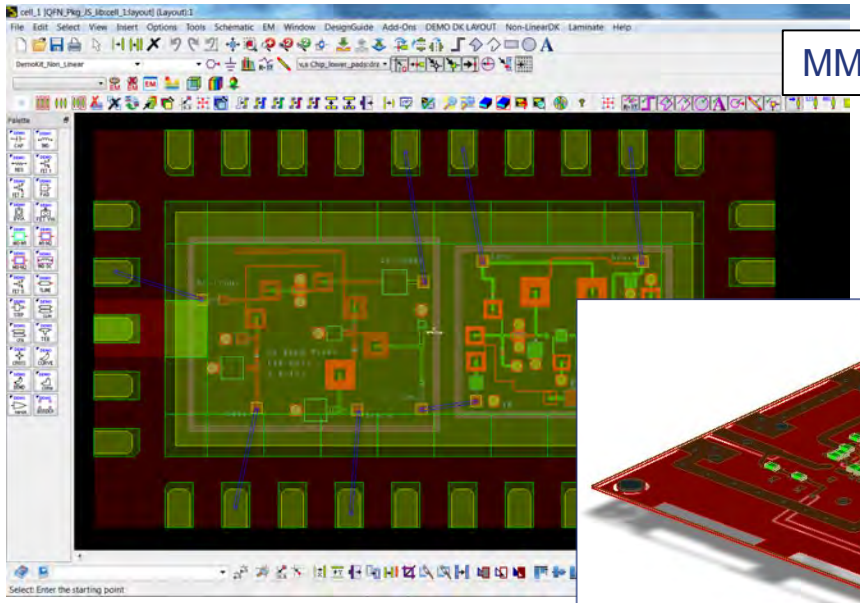
RF and Microwave Design Flow



KEYSIGHT TECHNOLOGIES

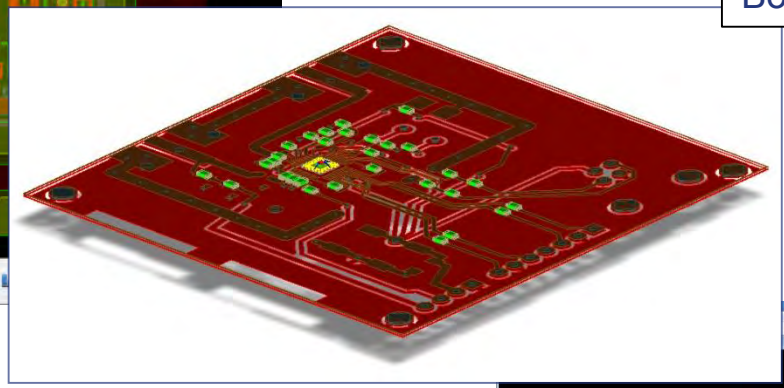
Each step of the flow can be complex

Layout and visualization

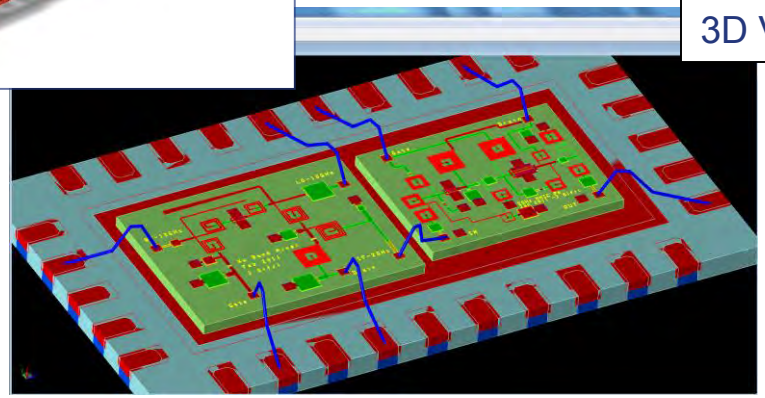


MMIC/Module

Board

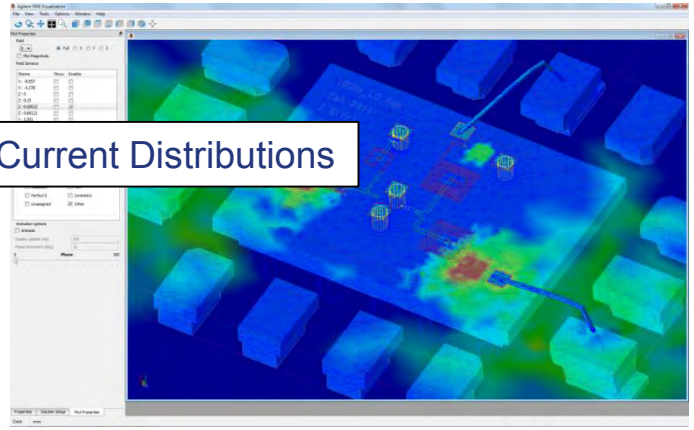


3D View

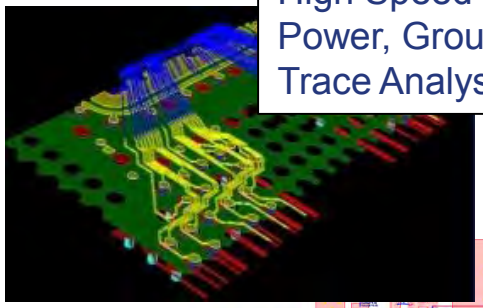


Metal traces become circuits on their own at high frequency

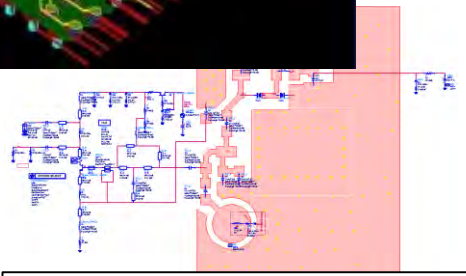
Use 3D Planar simulator and Full 3D Finite Element Method simulator



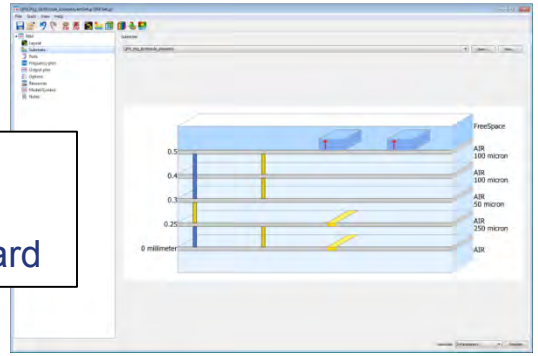
Field and Current Distributions



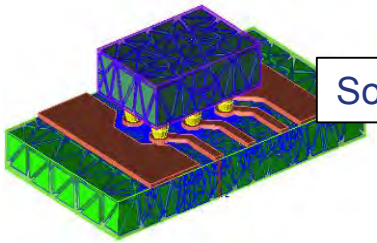
High Speed Digital Power, Ground, and Trace Analysis.



Co-Simulation Between Circuit and EM



Multiple Technologies
- Multi-Chip Modules
- Integrated IC-Package-Laminate-Board

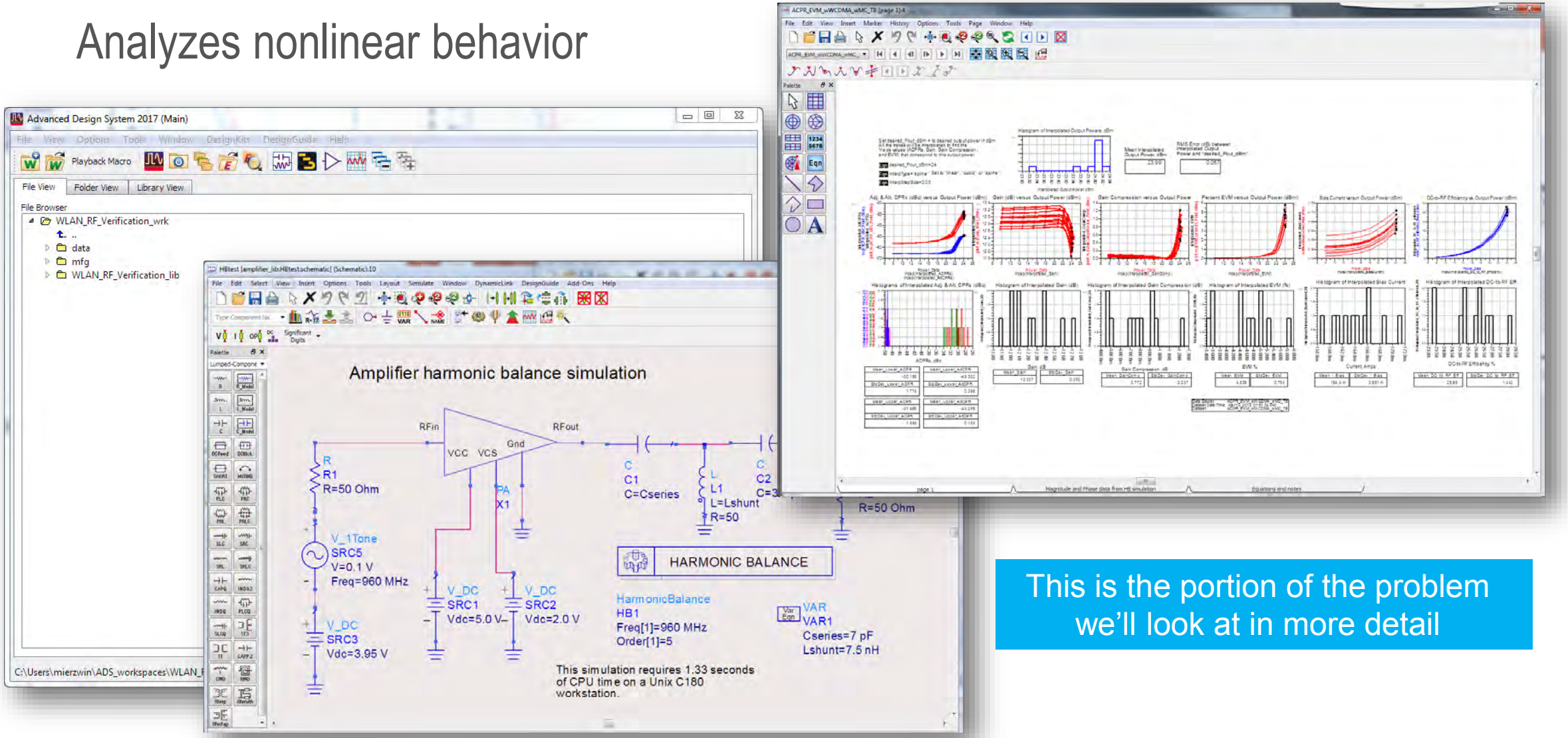


Solder Balls and Spirals

KEYSIGHT TECHNOLOGIES

The Circuit Simulator

Analyzes nonlinear behavior



This is the portion of the problem we'll look at in more detail

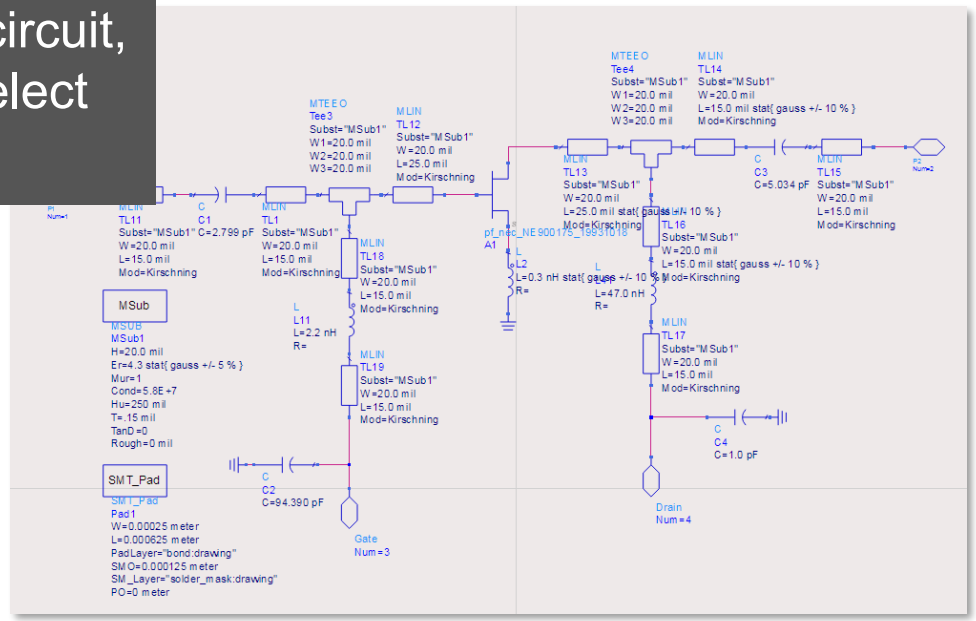
KEYSIGHT TECHNOLOGIES

Basic simulation flow

1

Start with some collection of information that describes the circuit, add sources and select an analysis to run

Schematic



Netlist

```
global 0
parameters _sw_fngeo=0

model testmodel_p r3
+ sw_et = _sw_et
+ sw_fngeo = _sw_fngeo
+ sw_mmgeo = _sw_mmgeo
+ subversion = 0
+ revision = 0
+ level =  1003
+ scale = 1.0
+ tmin = -100.0
+ tmax = 500.0

X1 (net1 net2 net3) testmodel_p
+ m =5
+ l = 0.05e-6

VD ( 0 net1_noise ) vsource type=sine sinedc=VD ampl=1
VG ( 0 net2_r ) vsource dc=VG type=dc
VS ( 0 net3_r ) vsource dc=VS type=dc

R1  net1 net1_noise resistor r=1 isnoisy=no
R2 (net2_r net2) resistor r=0.1 isnoisy=no
R3 (net3_r net3) resistor r=0.1 isnoisy=no
```

KEYSIGHT TECHNOLOGIES

Basic simulation flow

2

Simulator “parses” the information and creates an internal database of elements and values

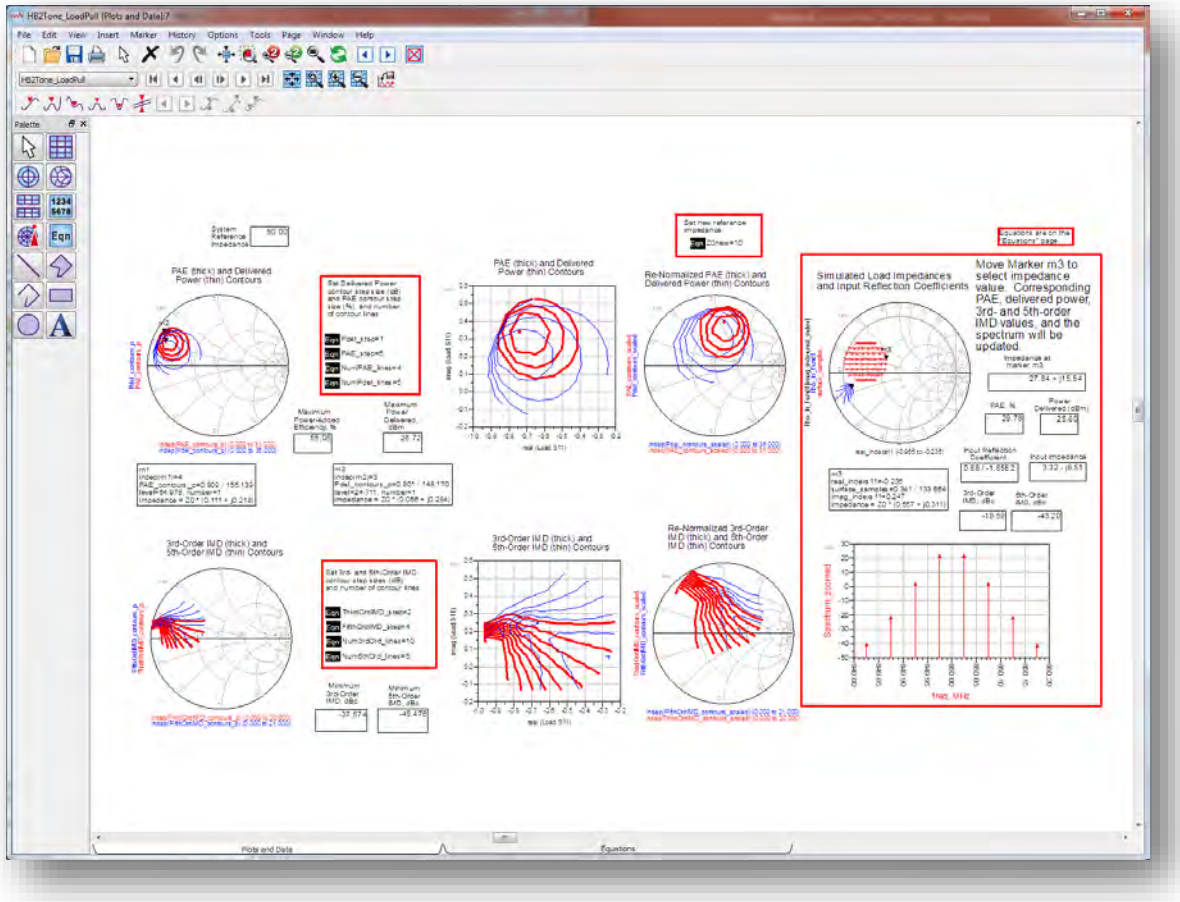
KEYSIGHT TECHNOLOGIES

Basic simulation flow

3

Simulator creates and solves the matrix of circuit equations, storing the results in a data file.

Post processing algorithms provide a convenient way to view the results of the data file.



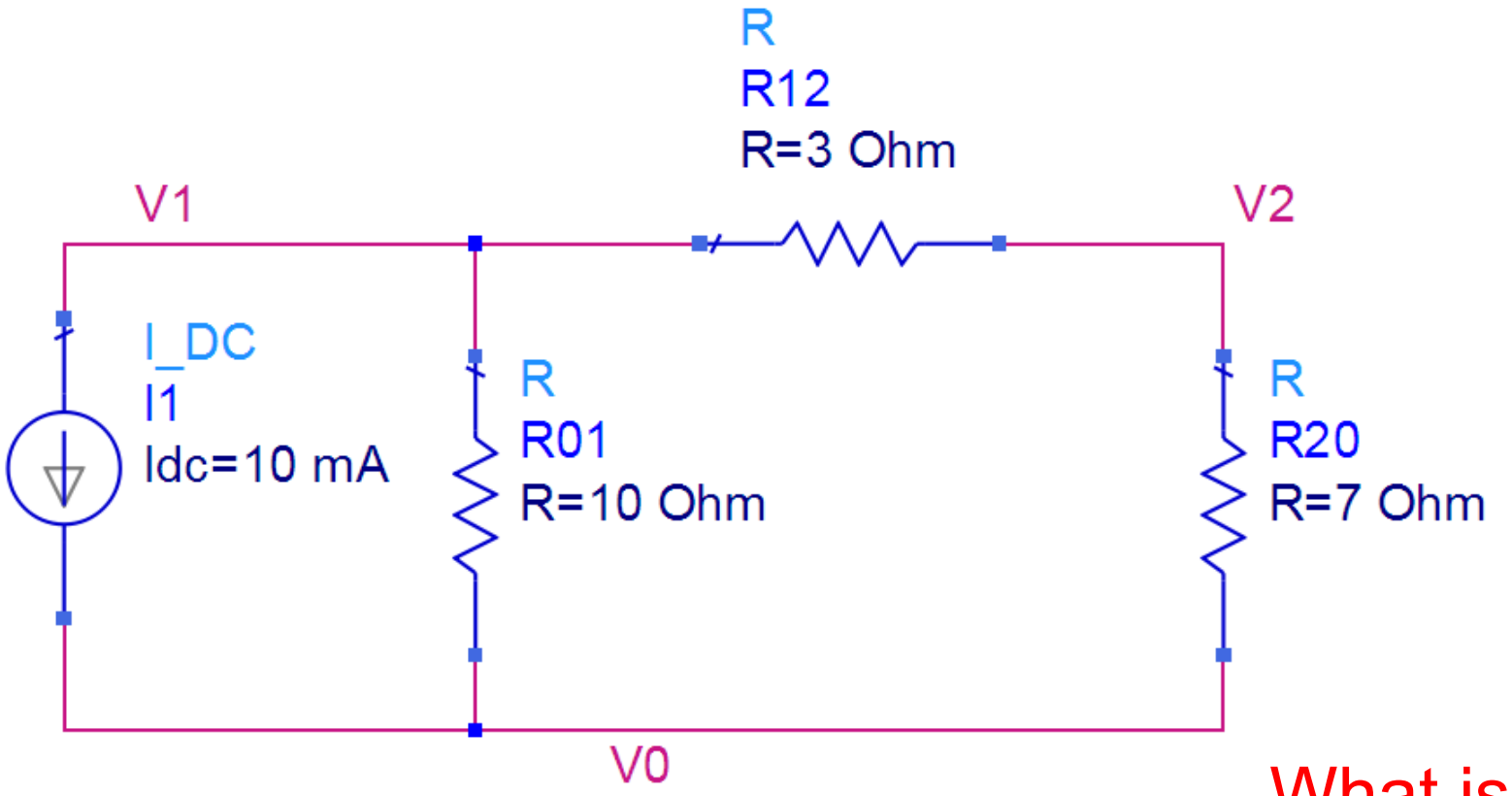
KEYSIGHT TECHNOLOGIES

Circuit Simulation

Let’s now at a simple example of a circuit, but follow the steps in solving a bit more closely

A simple circuit simulation example

Start with a current source and some resistors



What is V1 and V2?

Everything you always wanted to know about SPICE – Colin Warwick, Keysight

KEYSIGHT TECHNOLOGIES

Nodal Analysis

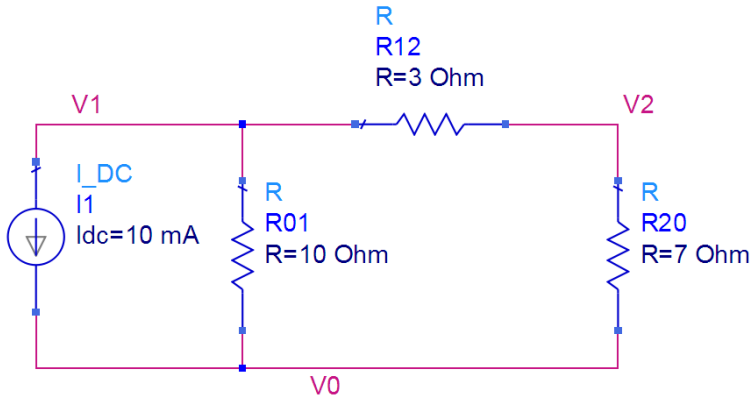
1

Apply Kirchoff's Current Law to the nodes

$$G_{01}(V_1 - V_0) - G_{20}(V_0 - V_2) - I_1 = 0$$

$$G_{12}(V_2 - V_1) - G_{01}(V_1 - V_0) + I_1 = 0$$

$$G_{20}(V_0 - V_2) - G_{12}(V_2 - V_1) = 0$$



The resistor's constitutive relation: I = V/R = G* V

KEYSIGHT TECHNOLOGIES

Rearrange to matrix form

2

Arrange into the matrix form of

$$\mathrm{G} \times V = I$$

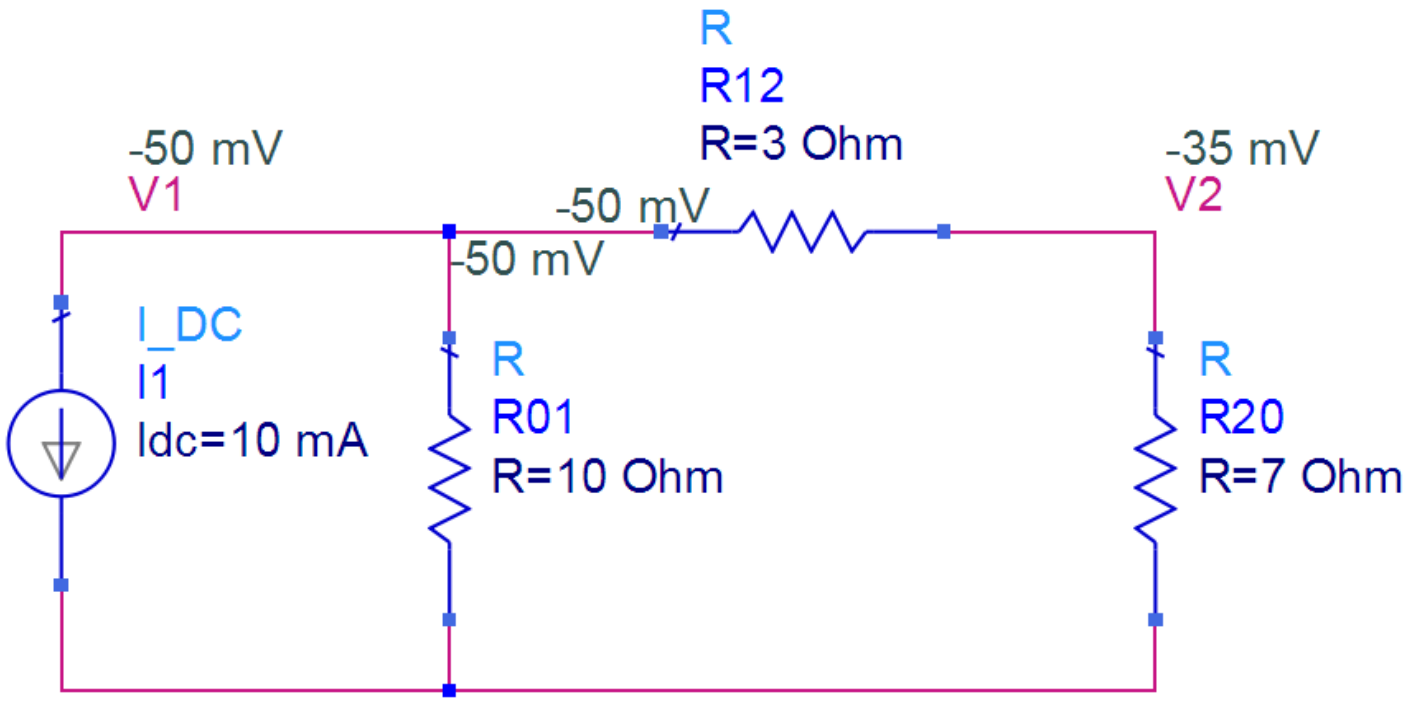
$$\begin{bmatrix} (G_{01}+G_{20}) & -G_{01} & -G_{20} \\ -G_{01} & (G_{12}+G_{01}) & -G_{12} \\ -G_{20} & -G_{12} & (G_{20}+G_{12}) \end{bmatrix} \times \begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} -I_1 \\ I_1 \\ 0 \end{bmatrix}$$

Solve the matrix

3

Invert the matrix, solve for

$$\mathrm{V} = G^{-1} I$$



V0 assigned to ground

Some complicating factors

- Voltage sources don’t work in simple nodal analysis
- Multi-terminal devices get loaded into the matrix differently
- Some elements are best described by their frequency dependence rather than voltage-current relations
- Users want DC, time/frequency-dependent, noise, small-signal, large signal, and other types of analyses in order to understand how the circuit works
- Nonlinear elements eliminate straight-forward solutions, as we’ll see next

Another simple circuit

But much more difficult to solve

Replacing the resistor network with a resistor and a simple (ideal) diode makes the solution much harder to determine.

The current through the diode is:

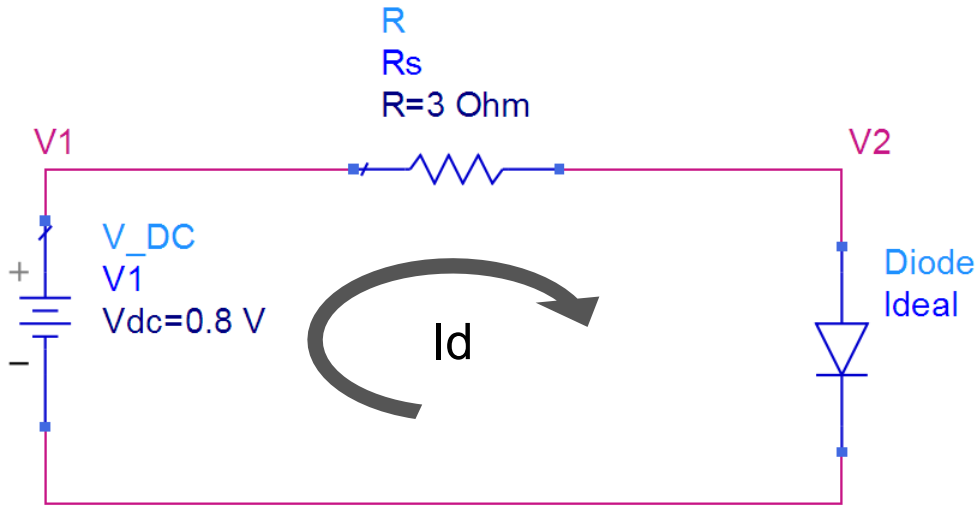
$$I_d = C\left(e^{\frac{V_2}{k}} - 1\right)$$

So, substituting and rearranging using Kirchoff's Voltage Law:

$$R_s C\left(e^{\frac{V_2}{k}} - 1\right) + V_2 - V_1 = 0$$

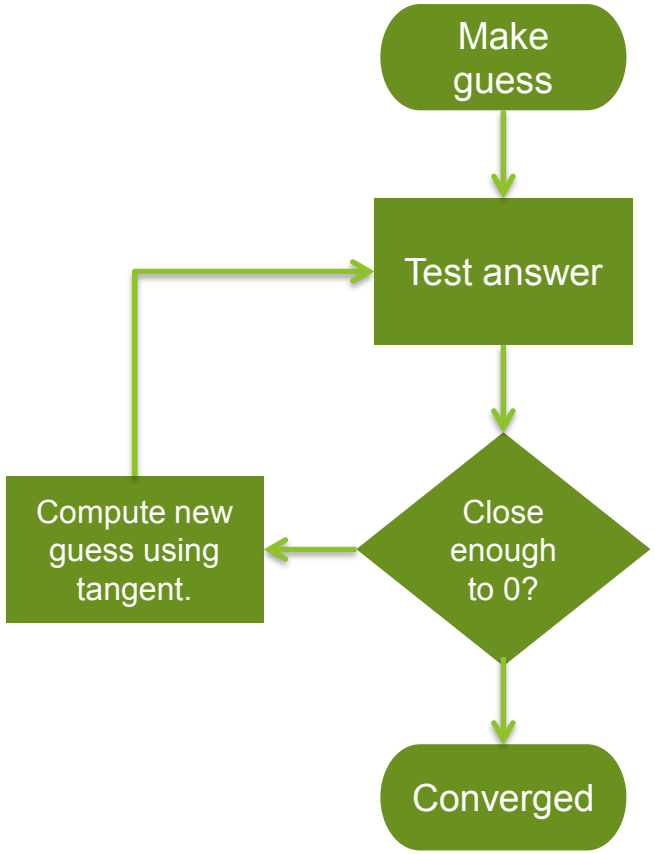
A transcendental equation of the form:

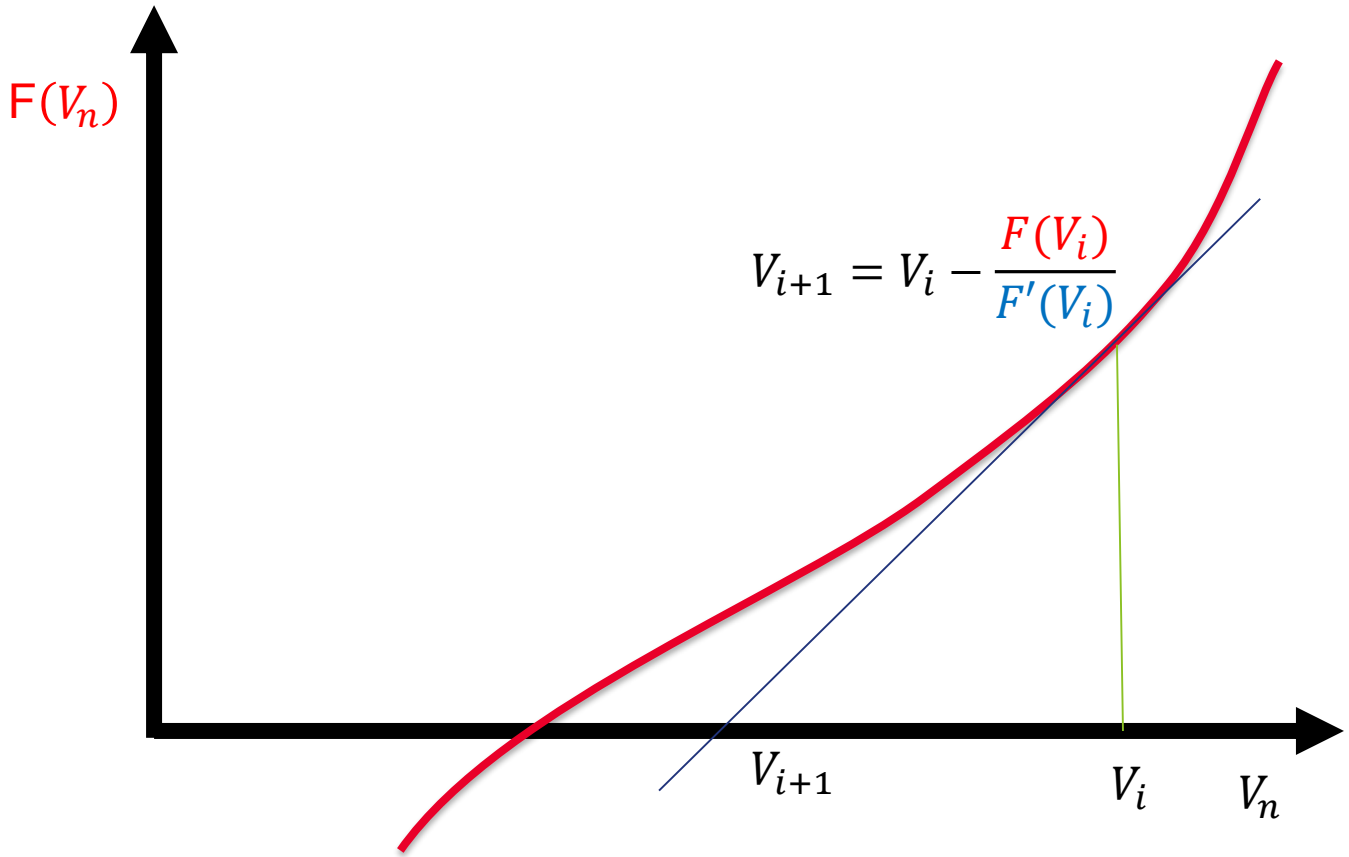
$$F(V_n) = 0$$



Nonlinear Circuit Simulation

Solves equations iteratively using Newton-Raphson technique





KEYSIGHT TECHNOLOGIES

Basic Simulator Equation

The simulator solves the equation

$$v_N^{(k+1)} - v_N^{(k)} = -J^{-1}F\left(v_N^{(k)}\right)$$

Where $v_N^{(k)}$ is the vector of node voltages on the kth iteration

And J is the 'Jacobian' matrix, the first order partial derivatives of the function

Every component must be described not just by its constitutive relations, but by its Jacobian entry.

The models have the information about how the circuit works, the rest of the job is just solving the matrix equation.

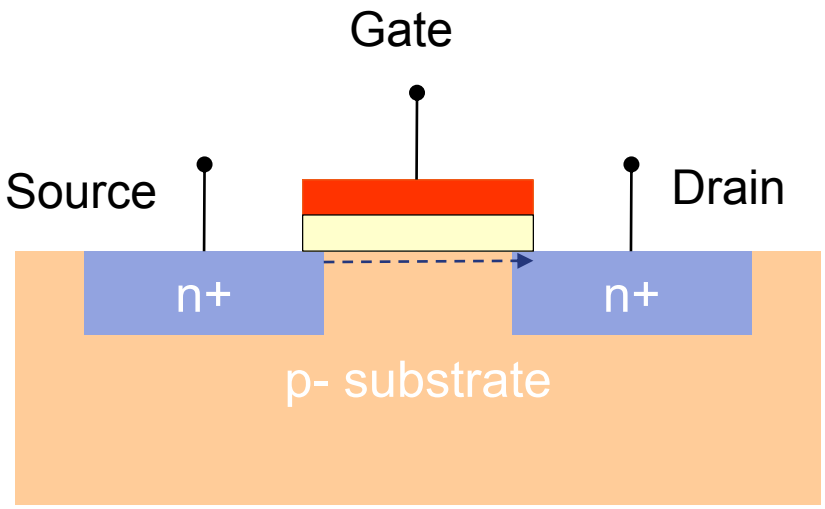
UC Berkeley's SPICE was the first program to do this and today's simulators share many design ideas.

Analog Models

Let’s now look in detail at the transistor models that the circuit simulator uses

Modeling the MOSFET

How can we get the answers the simulator needs?

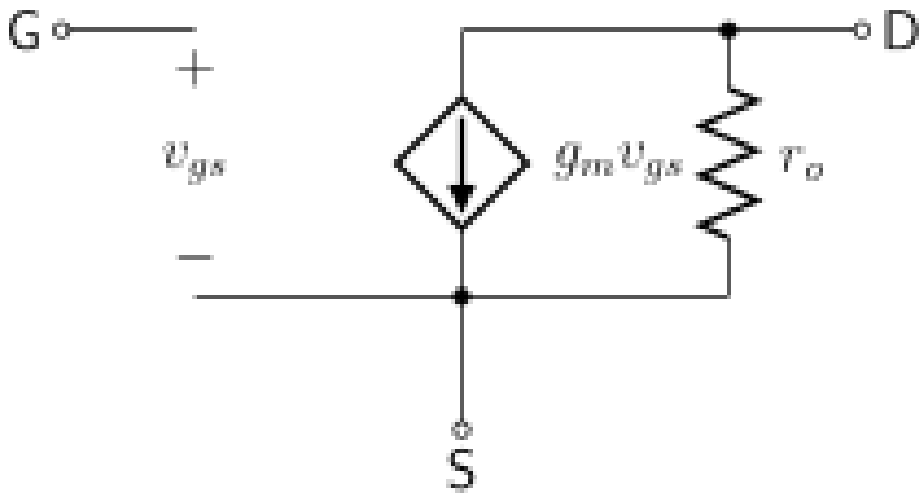


1. What current flows if we apply biases to each node?
2. How fast can the current change if we change the bias quickly?
3. How much noise is generated?
4. What if I change the size of the transistor
5. What if I change the materials?

Macro Models

e.g., The Hybrid Pi

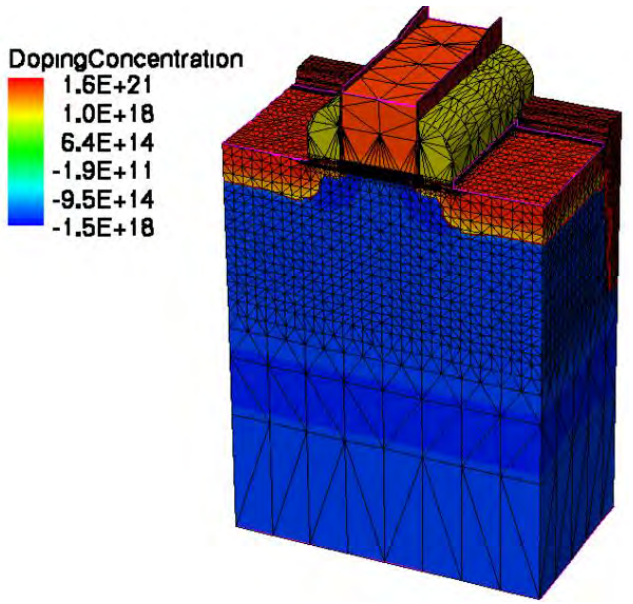
- Quick and easy and portable since just consists of standard components
- Nonlinear behavior can sometimes be implemented if the simulator supports it
- Messy when used beyond simple devices
  - Parameters…



Physics Based Models

Solve the Fundamental Equations

- Commercial and research programs allow the user to define the material and geometric values of a device, then apply biases or other stimuli
- Simulator will apply varying levels of physics to the problem
  - E-M / charge
  - Thermal
  - Quantum mechanics
  - Statistical physics
- Produces output such as potentials, currents and charges, light, etc.



https://en.wikipedia.org/wiki/File:SemiProcSimRslt.png

KEYSIGHT TECHNOLOGIES

Problem solved?

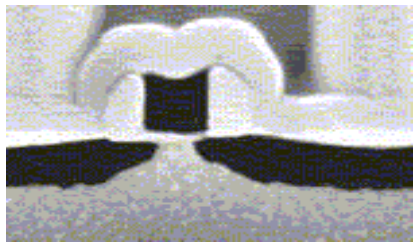
KEYSIGHT TECHNOLOGIES

Reality

Transistors do not come out of the process ‘as-drawn’

A cross section of a simple CMOS FET device shows that geometries are hard to define, hard to control, and hard to measure

So physics based modeling can’t give you the correct answer because it doesn’t have the correct question!



Credit: Texas Instruments

So, what to do?

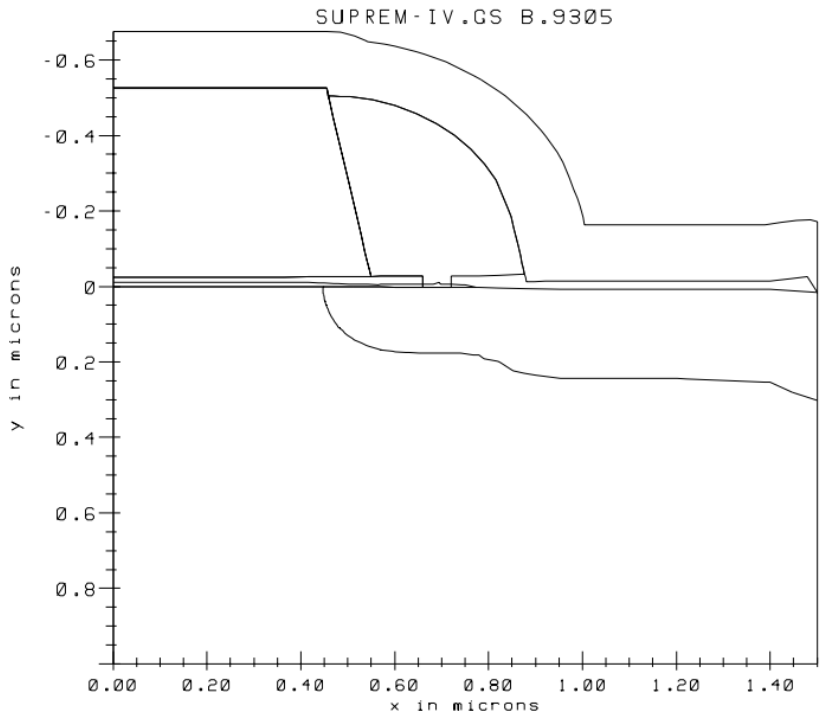
Simulate the semiconductor process

Feeds the results to the Physics Model

Process simulators exist, Stanford's SUPREM was one of the early ones

Process simulators take the equipment parameters as input, calculate the doping profiles, dielectric dimensions, etc.

They take complex phenomenon such as stress-induced diffusion, oxidation processes, etc. into account



KEYSIGHT TECHNOLOGIES

Problem solved now?

KEYSIGHT TECHNOLOGIES

Accurate Physics

Doesn't Mean Correct Answer

It just pushes the problem of calibration back another level

The fundament modeling issue

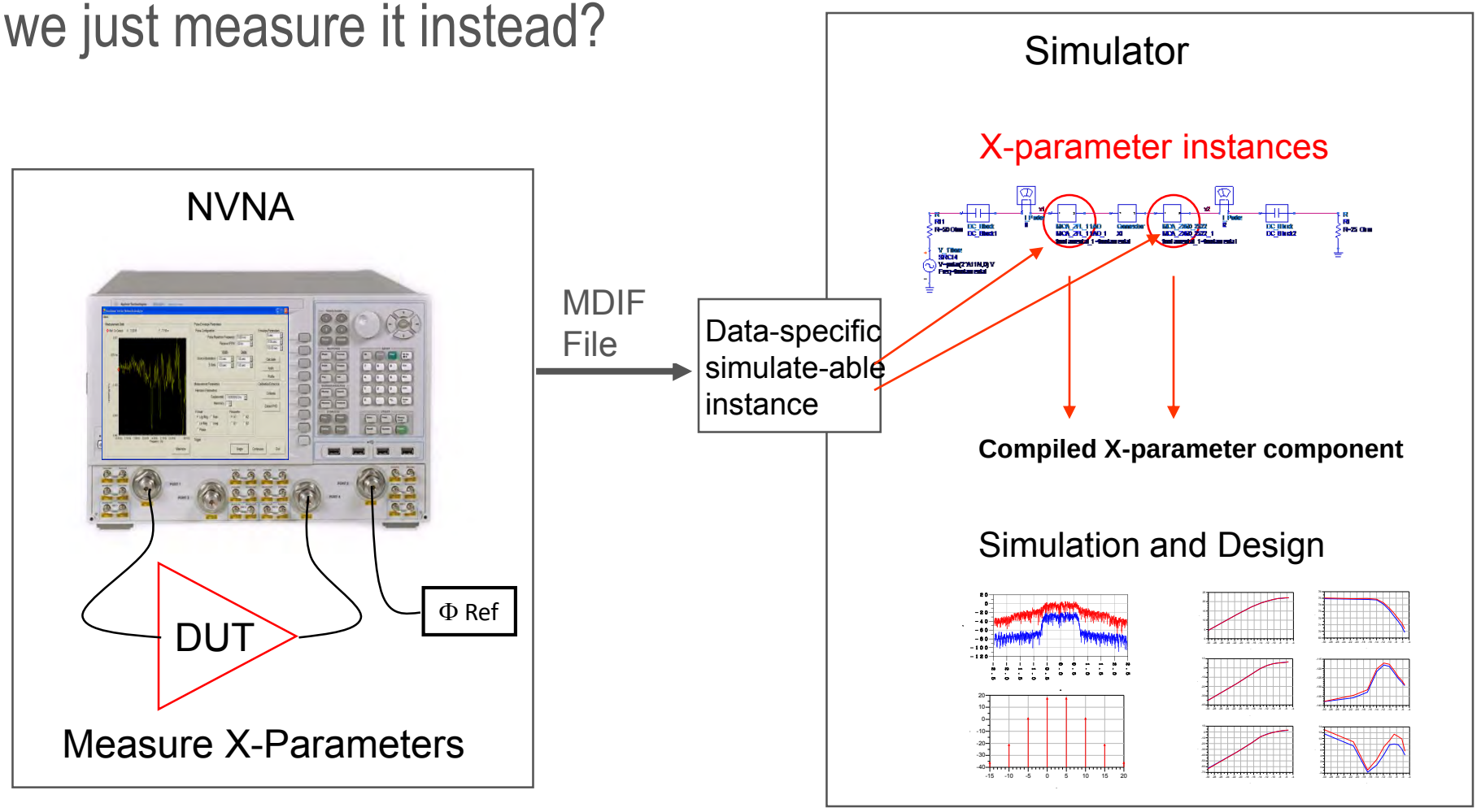
Why can’t we simulate everything?

We don’t have all the information

Even with the information we already have, it’s too much for our computers

Do We Even Need a Model?

Can we just measure it instead?



KEYSIGHT TECHNOLOGIES

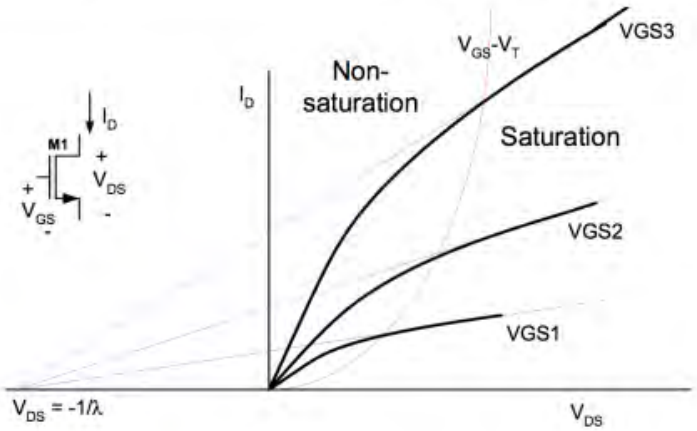
Compact Transistor Models

A set of equations and related parameters that describe the behavior

- Calculate equations from physics but simplify the math
- Keep the physics that's important
- Parameterize it so it can be applied to a wide variety of device types in that family
- Use empirical relations where you can see an effect but don't understand the cause

a) N-channel MOSFET

Cut Off	$V_{GS} \le V_T$	$I_{DS} = 0$
Linear	$V_{GS} > V_T,\ V_{DS} \le V_{GS} - V_T$	$I_{DS} = \mu_n C_{ox} \frac{W}{L}\left[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}\right](1 + \lambda V_{DS})$
Saturation	$V_{GS} > V_T,\ V_{DS} > V_{GS} - V_T$	$I_{DS} = \frac{1}{2}\mu_n C_{ox} \frac{W}{L}(V_{GS} - V_T)^2(1 + \lambda V_{DS})$

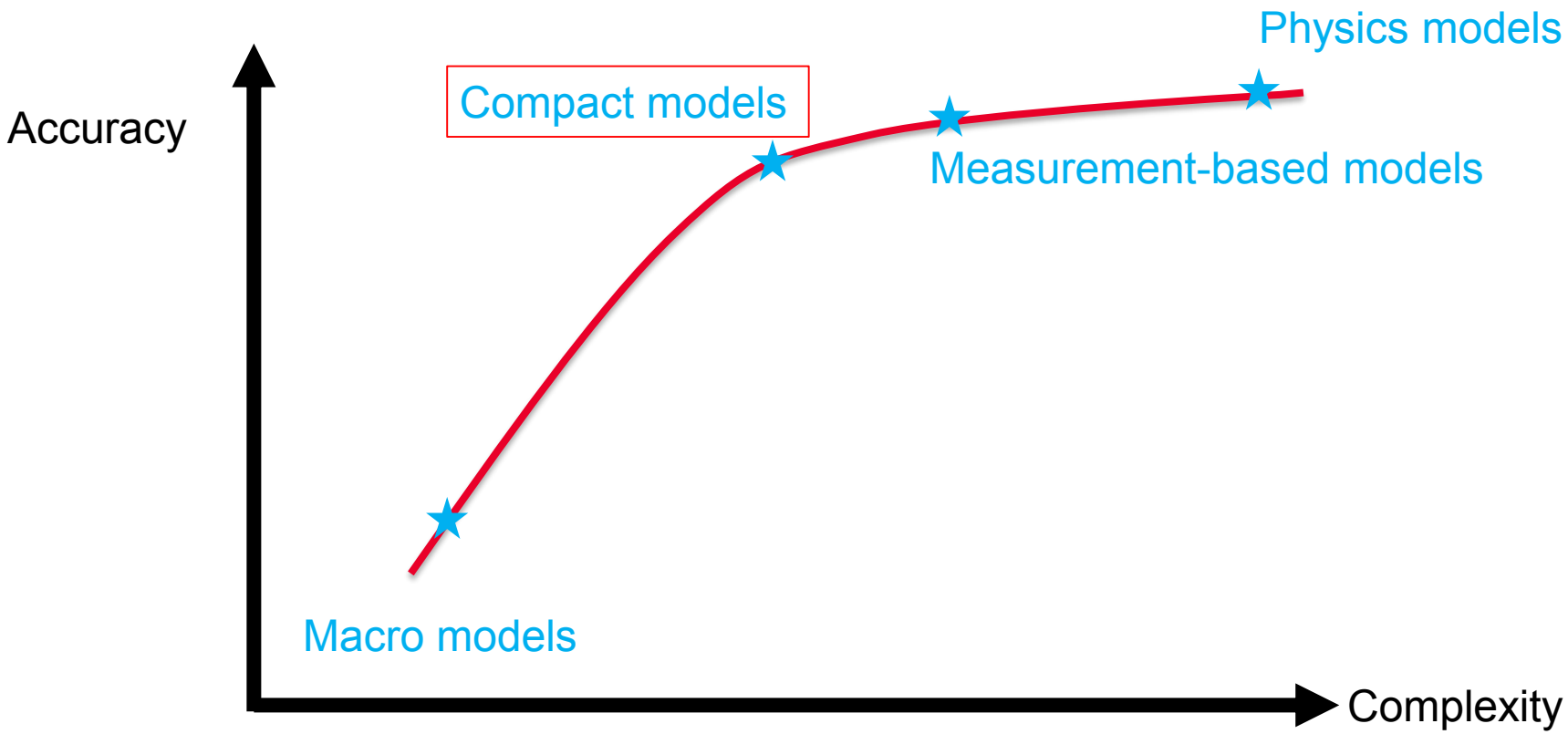


https://inst.eecs.berkeley.edu/~ee105/fa05/handouts/discussions/Discussion5.pdf

KEYSIGHT TECHNOLOGIES

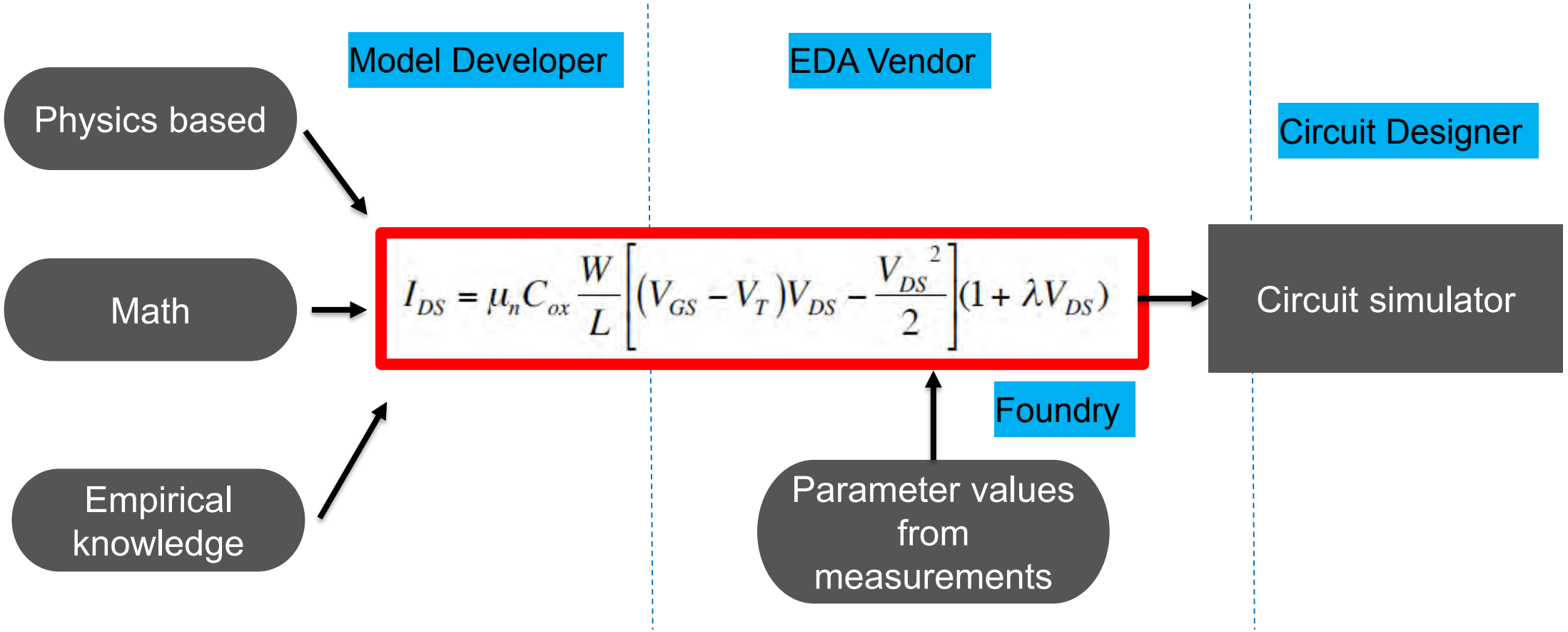
Choosing the level of model complexity

An optimization problem



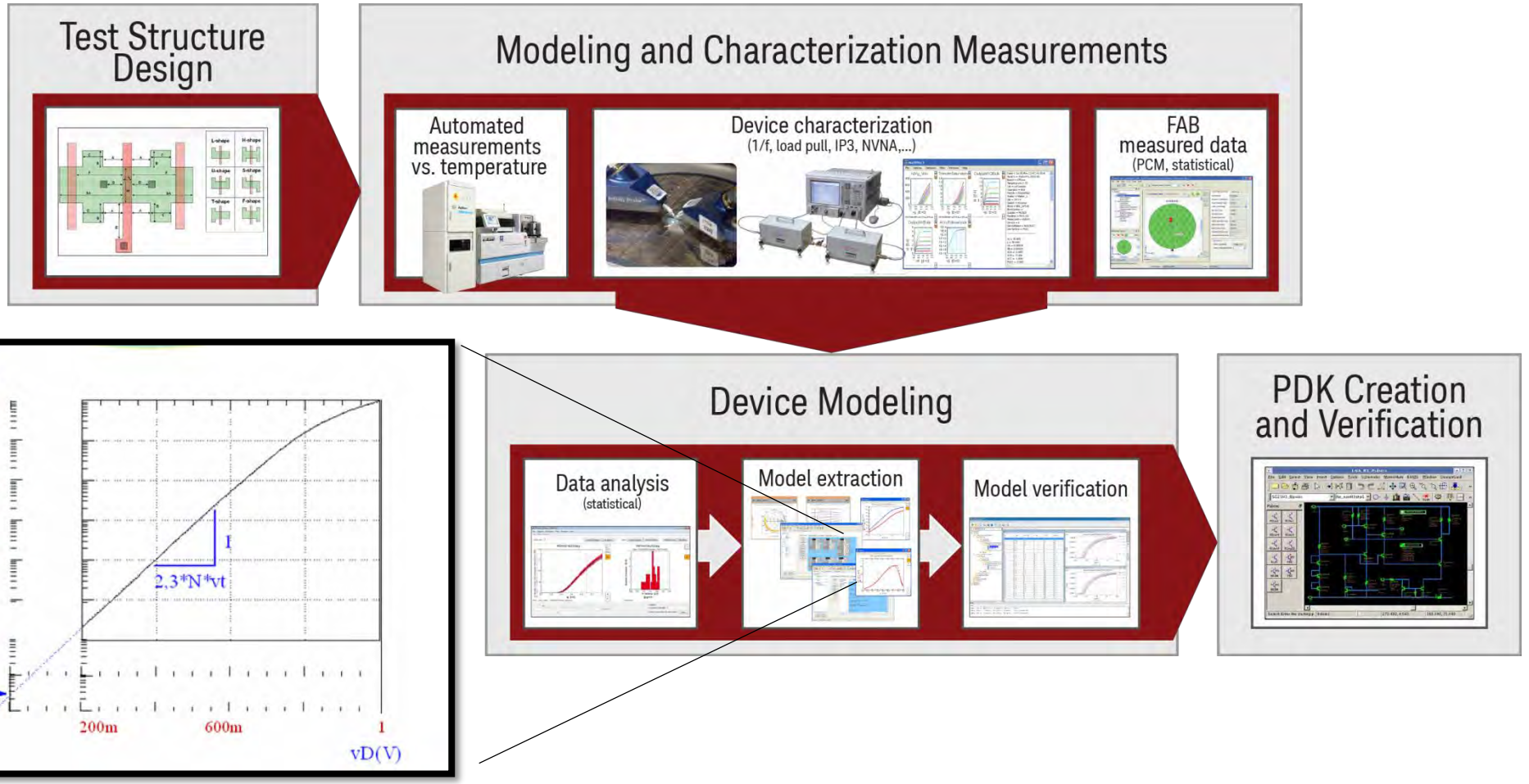
KEYSIGHT TECHNOLOGIES

Compact Transistor Models



KEYSIGHT TECHNOLOGIES

Device Modeling: Compact Model Parameter Extraction



KEYSIGHT TECHNOLOGIES

Compact Model Implementation

Compact models are the best way to describe nonlinear behavior, like transistors. How are they actually implemented in the simulator?

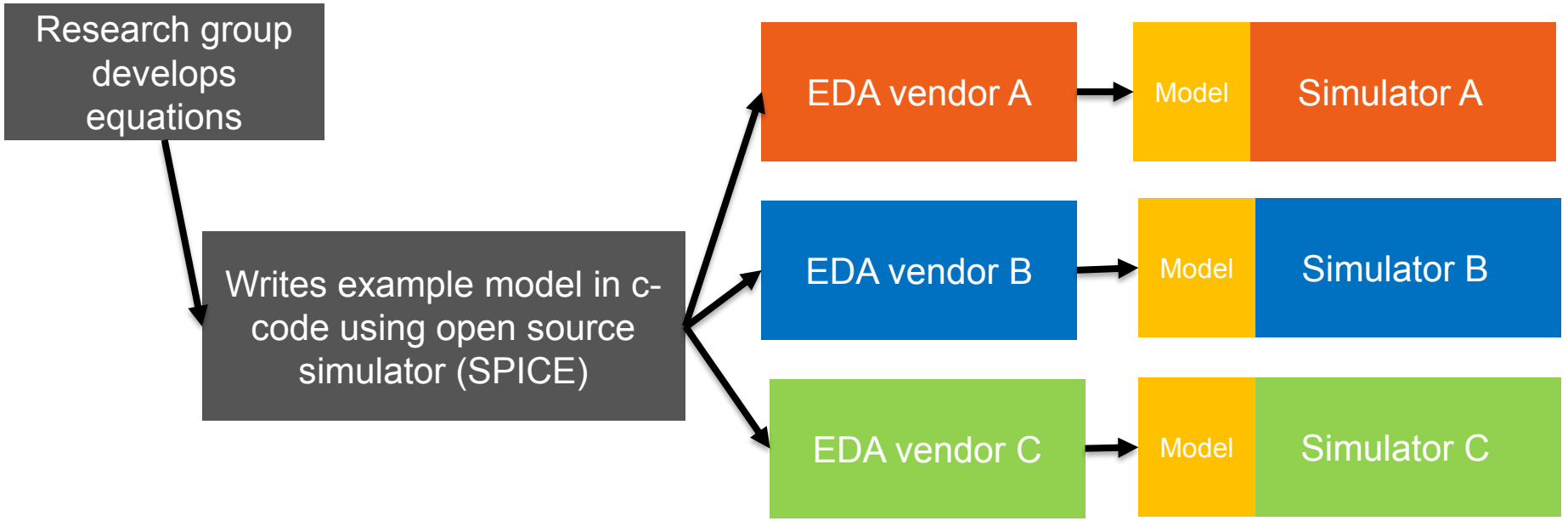
Compact Transistor Model Implementation

Lots of work

- Models need to:
  - compute matrix stamp
  - Implement multiple entry points for different analysis types
  - Handle housekeeping of parameter input and range checking
  - Compute derivatives for the Jacobian entry
- Keep up with simulator changes as new features or analyses are added
- Historically, this was done by writing subprograms that the would be compiled into the simulator program

Model Implementation

Historically



This step takes a few months, even years.

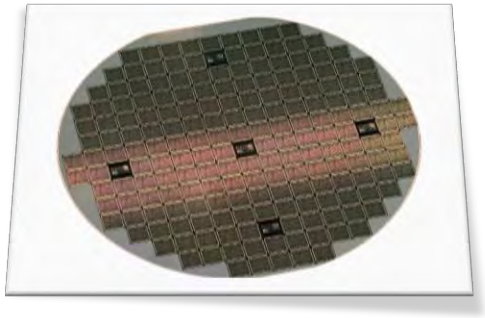
SPICE Implementation of the Resistor

A simple linear component

- To implement the resistor model, SPICE uses 30,000 lines of code!
- Each simulator vendor would take the c-code and convert to their own programming language
- Typically the original code would have bugs (typically in the derivative equations)
  - Some vendors would quietly fix the bug
  - Users not always happy to have a different answer

```
2046 Sep  4  1992 res.c
4762 Jun 17  1991 resask.c
 908 Apr  1  1991 resdel.c
 813 Apr  1  1991 resdest.c
1089 Apr  1  1991 resload.c
1185 Apr  1  1991 resmask.c
1045 Apr  1  1991 resmdel.c
1505 Apr  1  1991 respar.c
4480 Mar 21  1993 resnoise.c
1185 Apr  1  1991 resparam.c
1132 Apr  1  1991 respzld.c
1857 Apr  1  1991 ressacl.c
1397 Apr  1  1991 ressetup.c
1399 Apr  1  1991 ressload.c
1302 Apr  1  1991 ressprt.c
 986 Apr  1  1991 ressset.c
2590 Apr  1  1991 restemp.c
```

What do Foundries want?



- Want accurate, robust, and tested models.
- Want their end users to use any simulation too.
- Want a wide range of models, not one model that you have to fit your device to.
- Want to easily modify the underlying equations for special needs.

What do EDA (Electronic Design Automation) Vendors want?

- Want to have as many models in their simulator as possible
- Want their models to work just like everywhere else
- Don’t want to have to change anything

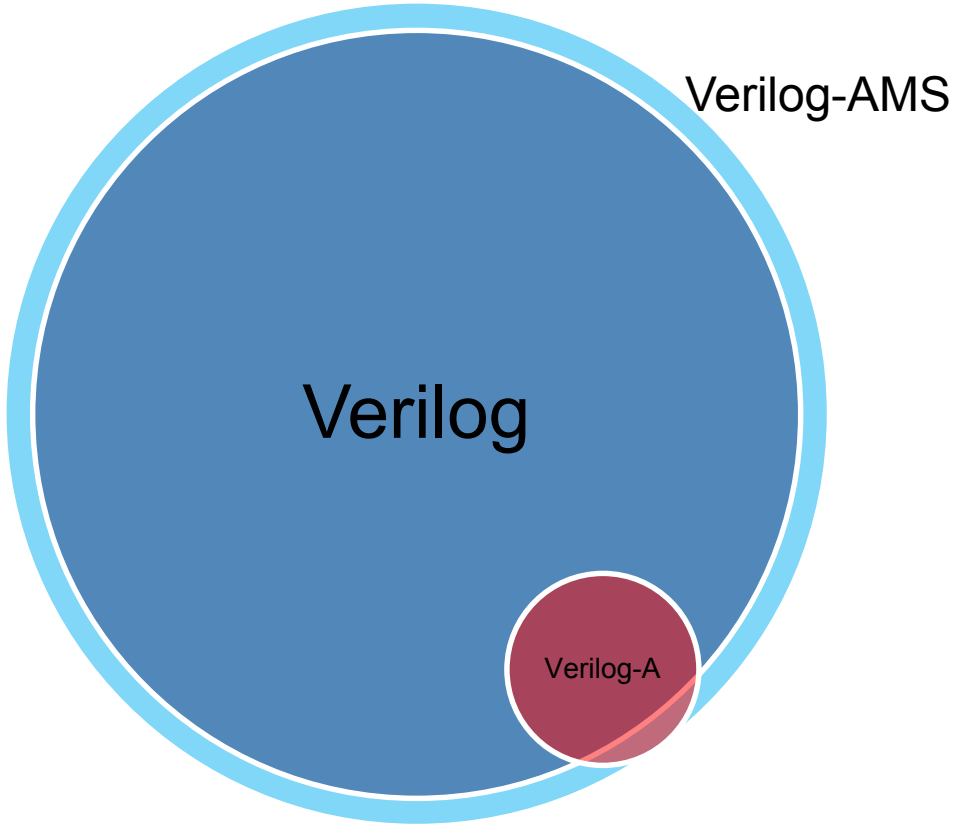
The Solution

A new programming language made for device model developers

KEYSIGHT TECHNOLOGIES

Verilog-A

Hardware **D**escription **L**anguage



Digital circuits are ‘programmed’ with Verilog

Verilog-A is a subset of this language – simulators can choose to just implement this small part rather than the full language

KEYSIGHT TECHNOLOGIES

Features of Verilog-A

– Natural programming language uses potential and flows, can be learned in hours

- Models do not have to be electrical

  - Thermal
  - Mechanical
  - Rotational
  - User defined

– Models can ‘instantiate’ other models

  - It can even be the netlist

– Works same in all simulators

– Code execution is a bit slower, but that will improve and should surpass manually-coded models

The Resistor in Verilog-A

~10 lines of code!

```
`include "disciplines.vams"
`include "constants.vams"

module resistor(p,n);
    electrical p,n;

    parameter real R=1 from (0:inf];
    real g;
    analog begin
       g = 1/R;
       I(p,n) <+ V(p,n)*g +
              white_noise(4 * `P_K *
              $temperature/ R,"thermal");
    end
endmodule
```

Include files

Ports

Parameter(s)

Variable(s)

Behavior

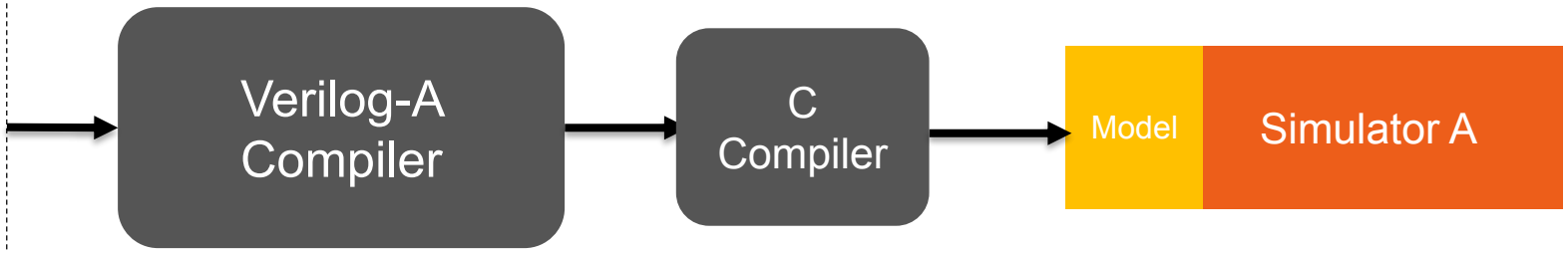
Declaration block

My_resistor.va

KEYSIGHT TECHNOLOGIES

Verilog-A in the Simulator

```
`include "disciplines.vams"
`include "constants.vams"

module resistor(p,n);
    electrical p,n;

    parameter real R=1 from
(0:inf];
    real g;
    analog begin
       g = 1/R;
       I(p,n) <+ V(p,n)*g +

                  white_noise(4 *
`P_K *

                  $temperature/
R,"thermal");
    end
endmodule
```



A separate program, called the Verilog-A compiler, takes the model and converts the parameters and equations into the information that the individual simulator needs.

- This includes calculating all the derivatives
- All the parameters and their limits
- How to load into the matrix (the stamp)

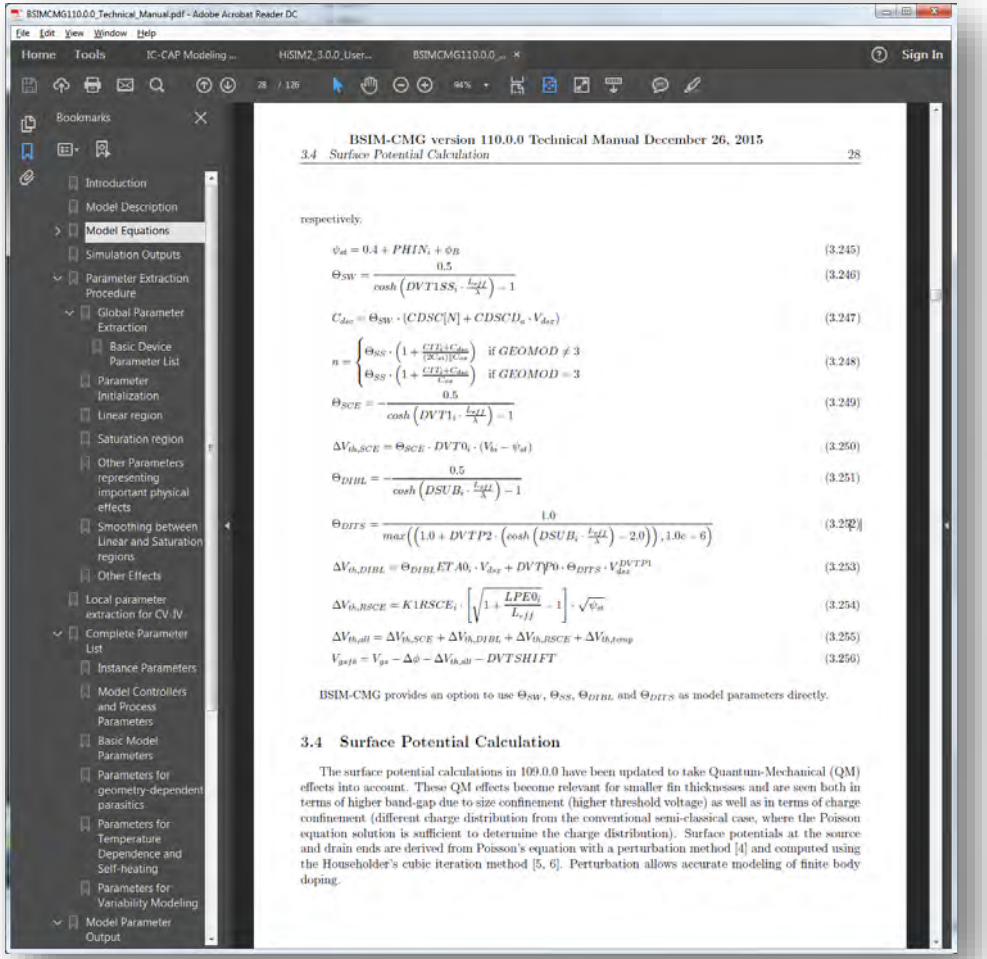
This step takes a few seconds.

KEYSIGHT TECHNOLOGIES

State of the Art Compact MOSFET Model

BSIM-CMG (Berkeley Short channel Insulated gate Model - Common Multiple Gate)

- Release document contains about 70 pages of equations
- These are just the constitutive relations – no derivatives provided!
- 500 parameters
- This is typical for a MOSFET model



KEYSIGHT TECHNOLOGIES

BSIM-CMG Verilog-A Code

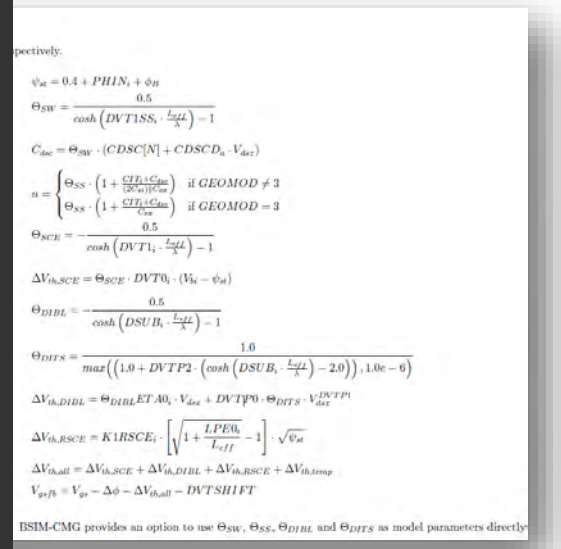
The source code is the reference and the documentation

Although not yet as well-documented as the reference manuals, the source code is readable.

It is also correct-by-definition. The model's behavior is defined by the source code.

This section shows the equations related to those displayed on the previous slide.

```
if (!$param_given(THETADIBL)) begin
   tmp  =  DSUB_i * Leff / scl + 1.0e-6;
   if (tmp < 40.0) begin
      Theta_DIBL  =  0.5 / (cosh(tmp) - 1.0);
   end else begin
      Theta_DIBL  =  exp(-tmp);
   end
end else begin
   Theta_DIBL  =  THETADIBL;
end

Theta_RSCE  =  sqrt(1.0 + LPE0_i / Leff) - 1.0;

tmp  =  DSUB_i * Leff / scl + 1.0e-6;
if (tmp < 40.0) begin
   T0  =  1.0 / max((1.0 + DVTP2 * (cosh(tmp) - 2.0)), 1.0e-6);
end else begin
   T0  =  exp(-tmp) / max((exp(-tmp) + DVTP2), 1.0e-6);
end

Theta_DITS  =  T0;
nbody       =  NBODY_i;
qbs         =  `q * nbody * Ach / Cins;

// Gate Current
if (TYPE == `ntype) begin
   Aechvb  =  4.97232e-7;  // NMOS
   Bechvb  =  7.45669e11;  // NMOS
end else begin
   Aechvb  =  3.42537e-7;  // PMOS
   Bechvb  =  1.16645e12;  // PMOS
end

T0  =  TOXG * TOXG;
T1  =  TOXG * POXEDGE_i;
T2  =  T1 * T1;
Toxratio      =  lexp(NTOX_i * lln(TOXREF / TOXG)) / T0;
```



pectively.

$$\psi_{st} = 0.4 + PHIN_i + \phi_H$$

$$\Theta_{SW} = \frac{0.5}{\cosh\left(DVT1SS_i \cdot \frac{L_{eff}}{\lambda}\right) - 1}$$

$$C_{dsc} = \Theta_{SW} \cdot (CDSC[N] + CDSCD_a \cdot V_{dsz})$$

$$n = \begin{cases} \Theta_{SS} \cdot \left(1 + \frac{CIT_i + C_{dsc}}{(2C_{ox})(C_{ox}} \right) & \text{if } GEOMOD \neq 3 \\ \Theta_{SS} \cdot \left(1 + \frac{CIT_i + C_{dsc}}{C_{ox}} \right) & \text{if } GEOMOD = 3 \end{cases}$$

$$\Theta_{SCE} = \frac{0.5}{\cosh\left(DVT1_i \cdot \frac{L_{eff}}{\lambda}\right) - 1}$$

$$\Delta V_{th,SCE} = \Theta_{SCE} \cdot DVT0_i \cdot (V_{bi} - \psi_{st})$$

$$\Theta_{DIBL} = \frac{0.5}{\cosh\left(DSUB_i \cdot \frac{L_{eff}}{\lambda}\right) - 1}$$

$$\Theta_{DITS} = \frac{1.0}{max\left(\left(1.0 + DVTP2 \cdot \left(\cosh\left(DSUB_i \cdot \frac{L_{eff}}{\lambda}\right) - 2.0\right)\right), 1.0e-6\right)}$$

$$\Delta V_{th,DIBL} = \Theta_{DIBL} ETA0_i \cdot V_{dsz} + DVTP0 \cdot \Theta_{DITS} \cdot V_{dsz}^{DVTP1}$$

$$\Delta V_{th,RSCE} = K1RSCE_i \cdot \left[\sqrt{1 + \frac{LPE0_i}{L_{eff}}} - 1\right] \cdot \sqrt{\psi_{st}}$$

$$\Delta V_{th,all} = \Delta V_{th,SCE} + \Delta V_{th,DIBL} + \Delta V_{th,RSCE} + \Delta V_{th,temp}$$

$$V_{gsfb} = V_{gs} - \Delta\phi - \Delta V_{th,all} - DVTSHIFT$$

BSIM-CMG provides an option to use Θ_{SW}, Θ_{SS}, Θ_{DIBL} and Θ_{DITS} as model parameters directly

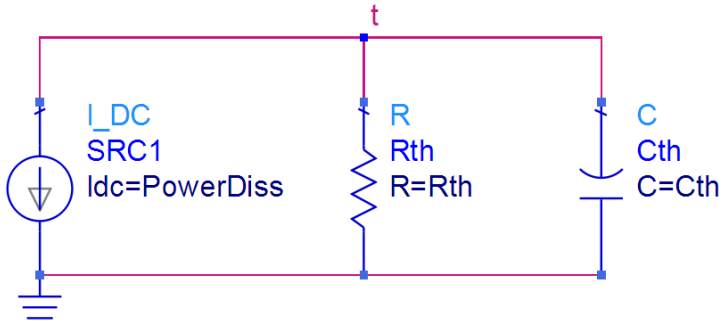
KEYSIGHT TECHNOLOGIES

Modeling Self-heating in Devices

An illustration of the simplicity and power of Verilog-A models

– Any device dissipates some power and this raises the device temperature, which changes the device characteristics – much more important as devices shrink

– Modeling self-heating is conceptually very simple:



– The voltage on node t is proportional to the temperature rise, so just add that to the device temperature

– And then add all the partial derivatives with respect to temperature!

Modeling Self-heating in Devices

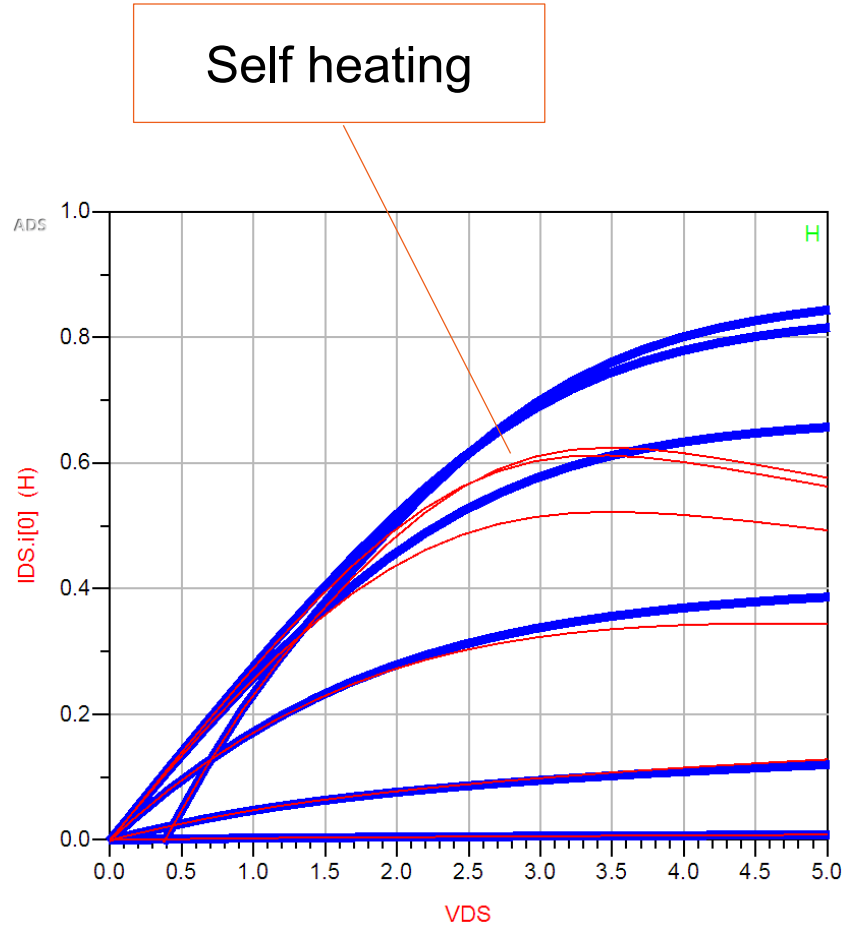
The Verilog-A solution

Self-heating is simple to add to any Verilog-A model:

```
T = $temperature + Temp(t);

Pwr(t) <+ - Id * Vd;
Pwr(t) <+ Temp(t) / Rth_T + ddt(Cth * Temp(t));
```

The circuit can even be described in terms of thermal characteristics.

And all the partial derivatives with respect to temperature are automatically calculated!



KEYSIGHT TECHNOLOGIES

Compact Model Coalition

The CMC is an international standards body formed to qualify and describe compact models in a way that

1. all simulators would access them in the same way
2. all simulators would return the same results when used in a simulator
3. New requirement: all models must be written in Verilog-A

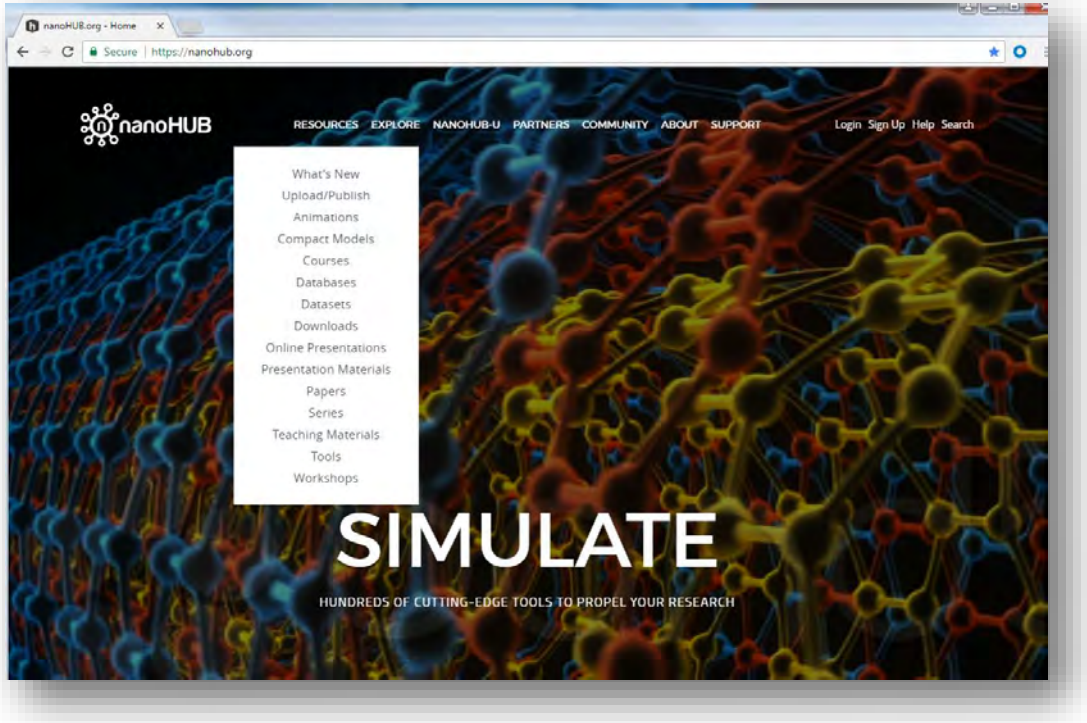
KEYSIGHT TECHNOLOGIES

nanoHUB.org

A NSF funded site for model collaboration

“nanoHUB.org is the premier place for computational nanotechnology research, education, and collaboration.

These resources help users learn about our simulation tools and about nanotechnology in general. ”



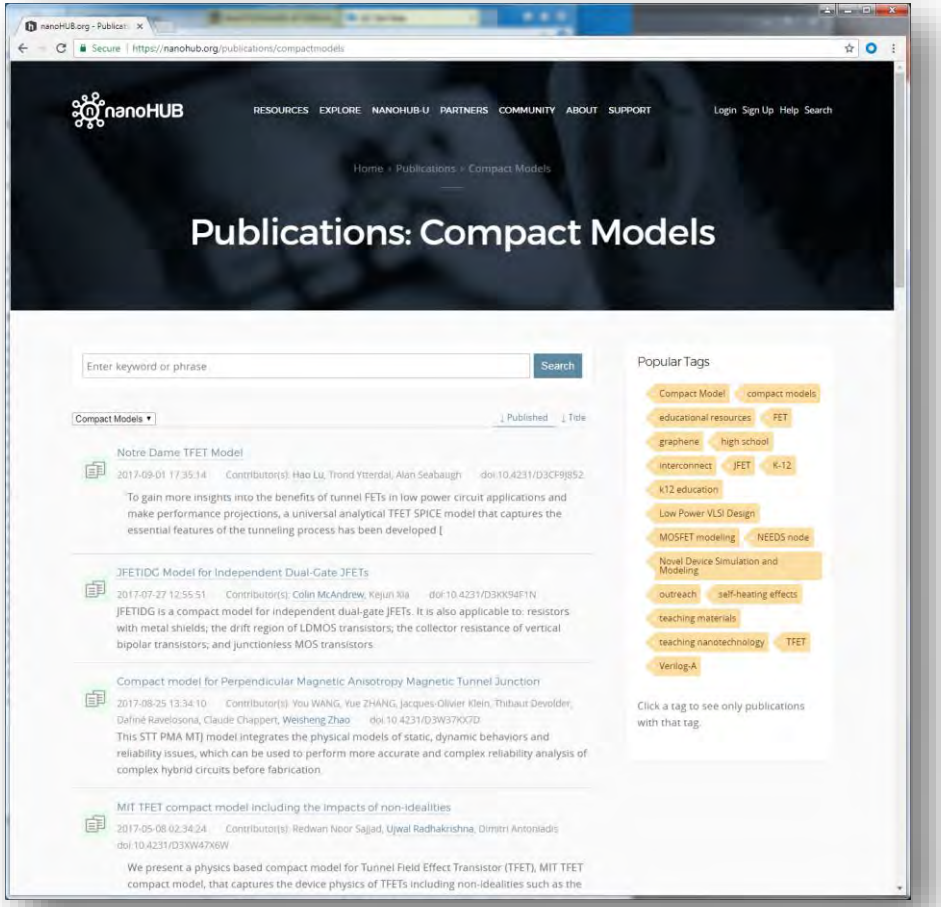
KEYSIGHT TECHNOLOGIES

nanoHUB.org

An active compact model development site

Users can

- post their models,
- provide feedback on other models,
- create tools for helping others create models
- access teaching material to understand devices

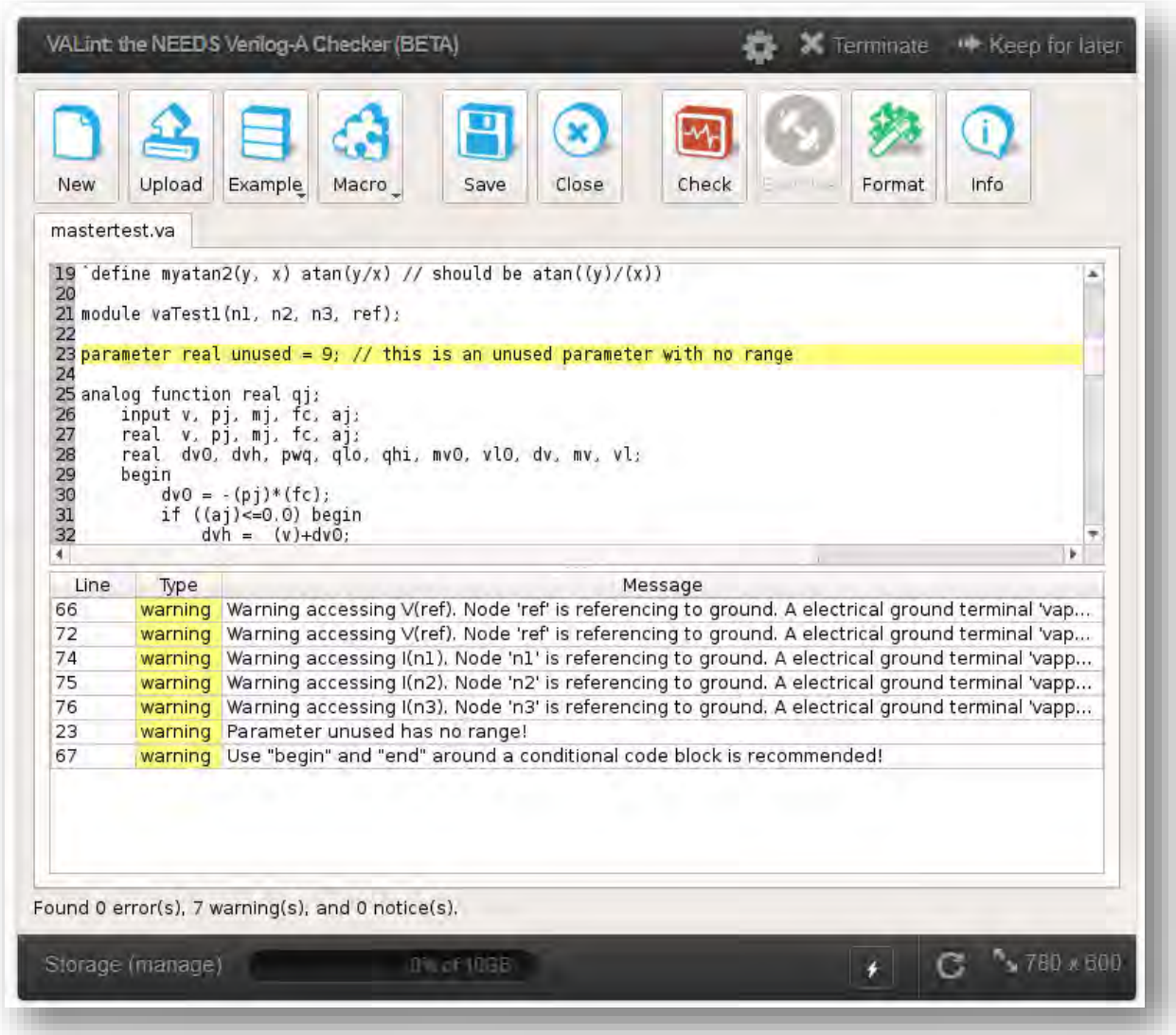


KEYSIGHT TECHNOLOGIES

NanoHub

Tools to help model developers

- VALint looks through Verilog-A code and checks for problems that a compact model might encounter
- Verilog-A is a general language for any analog behavior so it has more features than a compact model developer would (or should) use



KEYSIGHT TECHNOLOGIES

Compact Model Coalition

In the early 2000's there was

1

standard compact model, the BSIM4 model developed by UC Berkeley.

Now there are

> 20

standard compact models, from universities including

- UC Berkeley (3 separate families)
- Auburn
- University of Hiroshima (4 separate families)
- EPFL (France)
- Leti (France)
- UC San Diego
- MIT
- IIT – Kampur (India)
- Chalmers (Sweden)

KEYSIGHT TECHNOLOGIES

Summary

KEYSIGHT TECHNOLOGIES

What I Hope You've Learned

- The circuit design flow is complex and software tools play a critical part in developing today's products
- Analog circuit simulators work on the basic principle of solving Kirchoff's laws to create the equations, then using an iterative solving procedure based on Newton-Raphson to find the solution.
- To participate, nonlinear models must supply their constitutive relations but also their partial derivatives, plus a lot of other information
- Verilog-A provides a convenient way to quickly implement models – changing the paradigm from one model fits all devices to a model for each device
- Analog models can be developed by individuals now and a new infrastructure to support that is being created

Thank you for your attention!

KEYSIGHT TECHNOLOGIES

References

- SPICE: https://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/
- How to (and how not to) write a compact model in Verilog-A - Geoffrey J. Coram
- http://www.designers-guide.org/

HOW TO (AND HOW NOT TO) WRITE A COMPACT MODEL IN VERILOG-A

Geoffrey J. Coram
Analog Devices, Inc.
804 Woburn St., Wilmington, MA 01887

ABSTRACT

Verilog-A was recently enhanced to provide greater support for compact modeling. In order for Verilog-A to become the standard language for compact model development and implementation, two more steps are necessary: compact model developers must become familiar with the language, and simulators must run compact models written in Verilog-A almost as quickly and reliably as those hand-coded in C. This paper addresses both of these steps: it provides a quick introduction to writing compact models in Verilog-A and, by indicating the sorts of techniques that compact model writers may use, helps simulator vendors understand the sorts of optimizations that are expected from their Verilog-A interfaces.

1. INTRODUCTION

The standard language of compact modeling has been C since about 1985, when Spice was re-written from FORTRAN (Spice2) into C (Spice3). Most recent compact models have been written in C, although some still use FORTRAN. The Compact Model Council [1] has preferred C code in the past, but it encourages the release of Verilog-A source code for its next-generation MOSFET model standardization effort. All the developers of candidate models for the CMC effort have indicated the intent to write Verilog-A implementations of their models.

The Verilog-AMS hardware description language [2], and specifically its analog-only subset called Verilog-A, was recently enhanced to provide greater support for compact modeling by the release of the Verilog-AMS Language Reference Manual (LRM) version 2.2. This version of the LRM was developed over the course of the past two years with involvement from compact model developers, vendors of commercial simulators, and others, and is expected to be approved by the Accellera [3] board of directors in September 2004. Many of the language extensions in the new LRM were among those proposed by Lemaire [4].

Even before the addition of these compact modeling extensions, Verilog-A was an excellent language for compact model development and a dramatic improvement over C. The advantages are listed in Section 2. MATLAB is often used for compact model development, due to its ease of use and powerful data manipulation and plotting routines. However, MATLAB models cannot be used in circuit simulators. Verilog-A is almost as easy to use as MATLAB, and it can be used directly in circuit simulators – as well as in parameter extraction software, which provides methods for handling measured data. Verilog-A support, compliant with version 2.1 of the LRM, is available in a number of commercial simulators and parameter extraction tools, as well as in proprietary simulators of several semiconductor companies. Support for version 2.2 of the standard will happen soon, because most of the features are already available for built-in Spice models and for proprietary C interfaces to the simulators. The Verilog-A interface simply needs to connect the new syntax to existing functions in the C interfaces.

This paper is specifically intended for compact model developers who want to learn how to write compact models in Verilog-A. Section 5 provides a quick overview of the most useful elements of the language, which are scattered through the Language Reference Manual. Unlike the LRM, reference [5] is a user's guide, but intended more for circuit designers writing high-level behavioral models than for compact model developers. This paper focuses only on the analog features of Verilog-A and does not address mixed-signal modeling. Section 4 highlights a few tricks that might be useful, while Section 5 notes areas of particular concern for compact modeling.

This paper has a second intended audience: the programmers developing the Verilog-A interfaces for circuit simulators. Compact models written in Verilog-A should run reasonably fast, and simulator developers must invest some effort to make this happen. It might be acceptable for a new, automatically-compiled model to run slower than an older hand-coded model, if the features of the new model are compelling. For example, circuit designers accepted the BSIM3 MOSFET model for its improvements, even though it ran slower than the MOS level 3 model. And yet, it is not necessarily the case that Verilog-A models will run more slowly. Circuit simulators are not written in assembly language; modern C compilers are very good at automatically generating the assembly-level code. A good Verilog-A compiler could well produce more efficient code than a person would by hand, particularly if the person concentrates on developing the correct equations for the model. Section 6

0-7803-8615-9/04/$20.00 ©2004 IEEE

- http://signal-integrity.blogs.keysight.com/2009/circuit-simulation-part-one-spice-turns-thirty-six/ Colin Warwick
- IC-CAP Modeling Handbook, Franz Sischka

Jobs at Keysight

A few words about what skills are best for working at Keysight

KEYSIGHT TECHNOLOGIES

Core Skills for EDA Software Engineers

- Electronics
  - Circuits and systems
  - Semiconductor physics
- Software
  - Programming
  - Math and algorithms
  - User interface and usability
- Curiosity, creativity, and community

Careers at Keysight

Learn more and apply at jobs.keysight.com

KEYSIGHT TECHNOLOGIES

Everywhere the Electronic Signal Goes, Keysight is There

To help design, test, manufacture and optimize

- Multi-standard smartphones, tablets, IoT devices
- Networks and cloud environments
- Connected cars
- Clean energy
- Semiconductors
- Aerospace, Defense, and Government
- General electronics

KEYSIGHT TECHNOLOGIES