

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

From Electrical Measurement Data To Compact Device Models

SAND2019-9190C



Karthik V. Aadithya
Org. 1355



Paul A. Kuberry
Org. 1442



Eric R. Keiter
Org. 1355



Pavel B. Bochev
Org. 1442



Photo

Alan A. Mar
Org. 5445



Ting Mei
Org. 1355



Biliana S. Paskaleva
Org. 1356



Kenneth M. Leeson
Org. 1356



Machine Learning and Deep Learning Conference 2019



Overview of this talk

Part 1.

Circuit design basics.

Overview of this talk

Part 1.

Circuit design basics.

Part 2.

Compact models: what they are, why they're needed.

Overview of this talk

Part 1.

Circuit design basics.

Part 2.

Compact models: what they are, why they're needed.

Part 3.

Developing compact models is hard.

Overview of this talk

Part 1.

Circuit design basics.

Part 2.

Compact models: what they are, why they're needed.

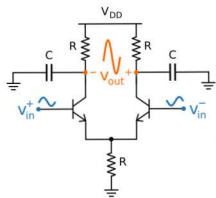
Part 3.

Developing compact models is hard.

Part 4.

Our solution: use machine learning to create compact models.

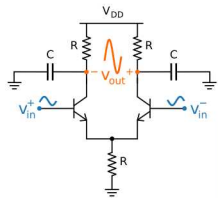
Lego-style circuit design



Problem

Designing circuits
(e.g., amplifiers)
that meet Sandia's
stringent specs

Lego-style circuit design

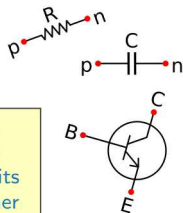


Problem

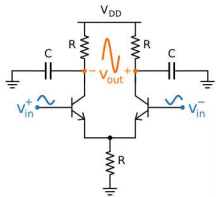
Designing circuits
(e.g., amplifiers)
that meet Sandia's
stringent specs

Design philosophy

Build complex circuits
by connecting together
simpler devices
(e.g., diodes, BJTs)

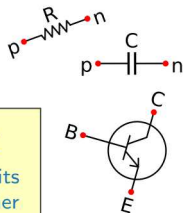


Lego-style circuit design

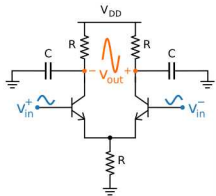


Problem
Designing circuits
(e.g., amplifiers)
that meet Sandia's
stringent specs

Design philosophy
Build complex circuits
by connecting together
simpler devices
(e.g., diodes, BJTs)



Lego-style circuit design

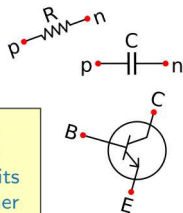


Problem

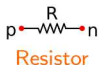
Designing circuits (e.g., amplifiers) that meet Sandia's stringent specs

Design philosophy

Build complex circuits by connecting together simpler devices (e.g., diodes, BJTs)

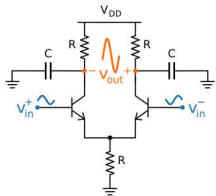


Each device performs a specific function



Impediment. Limit current in a circuit branch.

Lego-style circuit design

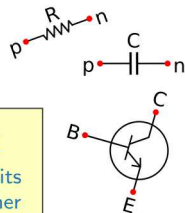


Problem

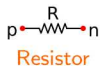
Designing circuits (e.g., amplifiers) that meet Sandia's stringent specs

Design philosophy

Build complex circuits by connecting together simpler devices (e.g., diodes, BJTs)



Each device performs a specific function

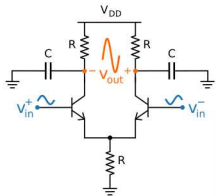


Impediment. Limit current in a circuit branch.



Reservoir. Store charge, convert it into current later.

Lego-style circuit design

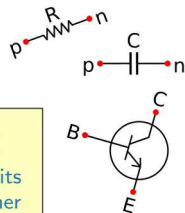


Problem

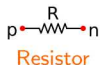
Designing circuits (e.g., amplifiers) that meet Sandia's stringent specs

Design philosophy

Build complex circuits by connecting together simpler devices (e.g., diodes, BJTs)



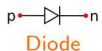
Each device performs a specific function



Impediment. Limit current in a circuit branch.

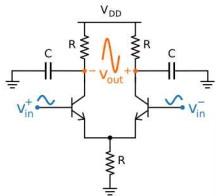


Reservoir. Store charge, convert it into current later.



Gatekeeper. Allow current one way, block it the other way.

Lego-style circuit design

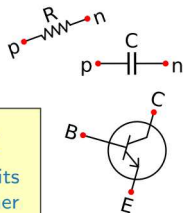


Problem

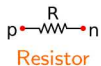
Designing circuits (e.g., amplifiers) that meet Sandia's stringent specs

Design philosophy

Build complex circuits by connecting together simpler devices (e.g., diodes, BJTs)



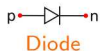
Each device performs a specific function



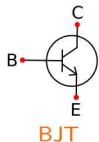
Impediment. Limit current in a circuit branch.



Reservoir. Store charge, convert it into current later.



Gatekeeper. Allow current one way, block it the other way.

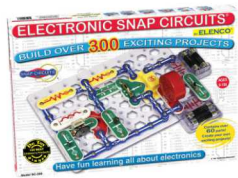


Switch. When B is +ve, allow current between C and E; disallow when B is -ve.

Circuit design illustration: Snap Circuits



My cousin



His toy

Circuit design illustration: Snap Circuits



My cousin



Devices in Snap Circuits kit



His toy

Circuit design illustration: Snap Circuits



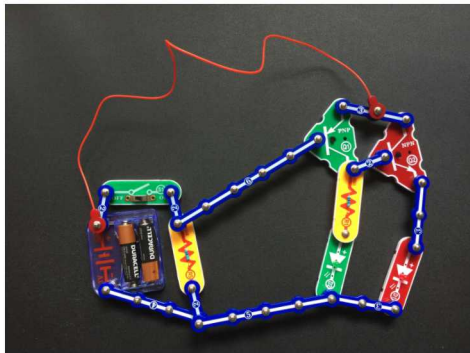
My cousin



Devices in Snap Circuits kit



His toy



Movie: Circuit 1

Circuit design illustration: Snap Circuits



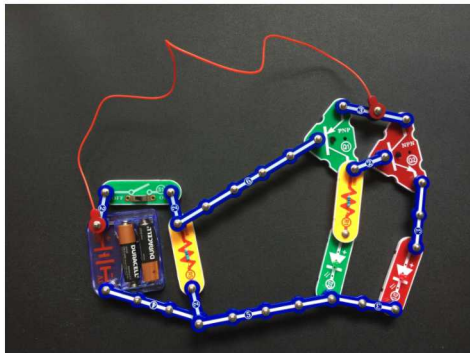
My cousin



His toy



Devices in Snap Circuits kit



Movie: Circuit 1 → Circuit 2

Modern circuit design = Heavily computer-aided

- Modern circuits are huge, cannot be built by hand



9.6B transistors, 14nm



2B transistors, 20nm

Modern circuit design = Heavily computer-aided

- Modern circuits are huge, cannot be built by hand



9.6B transistors, 14nm



2B transistors, 20nm

- Complex designs
 - Lots of inputs, outputs, and internal states
 - Digital, analog, and mixed-signal functionality
 - Wide range of operating environments

Modern circuit design = Heavily computer-aided

- Modern circuits are huge, cannot be built by hand



9.6B transistors, 14nm



2B transistors, 20nm

- Complex designs
 - Lots of inputs, outputs, and internal states
 - Digital, analog, and mixed-signal functionality
 - Wide range of operating environments
- EDA/CAD software tools used for every step in design cycle

Modern circuit design = Heavily computer-aided

- Modern circuits are huge, cannot be built by hand



9.6B transistors, 14nm



2B transistors, 20nm

- Complex designs
 - Lots of inputs, outputs, and internal states
 - Digital, analog, and mixed-signal functionality
 - Wide range of operating environments
- EDA/CAD software tools used for every step in design cycle
- Extensive circuit simulation to validate, test designs pre-fab



OrCAD PSpice

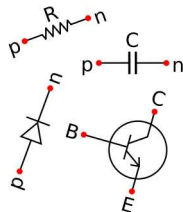


Spyce MAPP

Circuit simulators need to be told how devices behave

Precise mathematical relationships between voltages, currents in each device

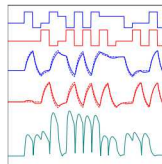
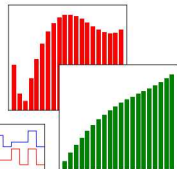
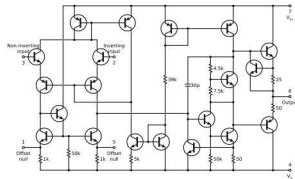
Devices Library



Circuit designer

Circuit

Circuit simulator



Quantitative predictions

Circuit simulators need to be told how devices behave

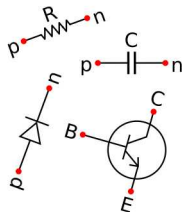


Compact model developer

Precise mathematical relationships between voltages, currents in each device

Compact models

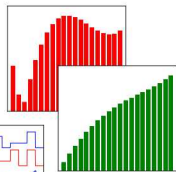
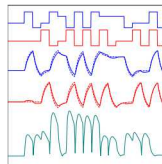
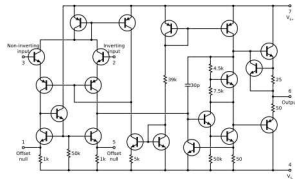
Devices Library



Circuit designer

Circuit

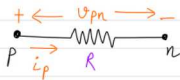
Circuit simulator



Quantitative predictions

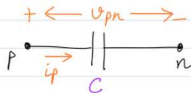
Compact models: some simple examples

Resistor



$$i_p = \frac{V_{pn}}{R}$$

Capacitor

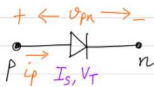


$$i_p = \frac{d}{dt} C V_{pn}$$

General Form

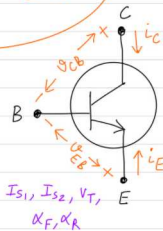
$$\frac{d}{dt} \vec{z} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) + \vec{f} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) = \vec{0}$$

Diode



$$i_p = I_s \left(e^{V_{pn}/V_T} - 1 \right)$$

BJT



$$i_1 = I_{s1} \left(e^{-V_{EB}/V_T} - 1 \right)$$

$$i_2 = I_{s2} \left(e^{-V_{CB}/V_T} - 1 \right)$$

$$i_E = \alpha_R i_2 - i_1$$

$$i_C = \alpha_F i_1 - i_2$$

“Real” compact models: the BSIM family

```
[E1] Modeling Body...
3320 Inv_L = 1.0e-4 / (Leff1);
3321 Inv_WFN = 2.0 / WFN;
3322 Inv_LWFN = 1.0e-4 / (Leff1 * WFN);
3323
3324 // Steady State Equation For UTEC Parameters
3325 NBS0F_1 = NBS0F + Inv_L * LWBS0F + Inv_WFN * NBS0F + Inv_LWFN * PABS0F;
3326
3327 IF (ABS(NBS0F - 6.0) begin
3328     NBS0F_1 = NBS0F_1 * (1.0 + ABS(NBS0F * Inv_L * (1.0 - WFN / ABS(NBS0F)))));
3329 end
3330
3331 // Model Parameters For Verified FinFET Compact Model by Joan Suarez 18/200
3332 case (VERIFIED)
3333     0: begin // Double Gate
3334         IF (Iparam_given(TFIN_TOP)) { Iparam_given(TFIN_BSD2) begin
3335             ueff_UFCM = 2.0 * WFN;
3336             Cinv = ueff_UFCM * EPSBOX * EPISB / EST;
3337             Rch = WFN * TFIS;
3338             r0 = (2.0 * Cinv / (ueff_UFCM * ueff_UFCM * epostub / Rch));
3339             Gdep_wt_Cinv = -'g * NBS0F_1 * Rch / Cinv;
3340         end else begin
3341             ueff_UFCM = 2.0 * uopt(WFN * WFN * TFIN_TOP - TFIN_BSD2) * (TFIN_TOP -
3342             Cinv = ueff_UFCM * EPSBOX * EPISB / EST;
3343             Rch = WFN * (TFIN_TOP + TFIN_BSD2) / 2.0;
3344             r0 = (2.0 * Cinv / (ueff_UFCM * ueff_UFCM * epostub / Rch));
3345             Gdep_wt_Cinv = -'g * NBS0F_1 * Rch / Cinv;
3346         end
3347     end
3348     1: begin // Triple Gate
3349         IF (Iparam_given(TFIN_TOP)) { Iparam_given(TFIN_BSD2) begin
3350             ueff_UFCM = 2.0 * WFN * TFIS;
3351             Cinv = ueff_UFCM * EPSBOX * EPISB / EST;
3352             Rch = WFN * TFIS;
3353             r0 = (2.0 * Cinv / (ueff_UFCM * ueff_UFCM * epostub / Rch));
3354             Gdep_wt_Cinv = -'g * NBS0F_1 * Rch / Cinv;
3355         end else begin
3356             ueff_UFCM = 2.0 * uopt(WFN * WFN * (TFIN_TOP - TFIN_BSD2) * (TFIN_TOP -
3357             Cinv = ueff_UFCM * EPSBOX * EPISB / EST;
3358             Rch = WFN * (TFIN_TOP + TFIN_BSD2) / 2.0;
3359             r0 = (2.0 * Cinv / (ueff_UFCM * ueff_UFCM * epostub / Rch));
3360             Gdep_wt_Cinv = -'g * NBS0F_1 * Rch / Cinv;
3361         end
3362     end
3363     2: begin // Quadruple Gate
3364         IF (Iparam_given(TFIN_TOP)) { Iparam_given(TFIN_BSD2) begin
3365             ueff_UFCM = 2.0 * WFN * 2.0 * TFIS;
3366             Cinv = ueff_UFCM * EPSBOX * EPISB / EST;
3367             Rch = WFN * TFIS;
3368             r0 = (2.0 * Cinv / (ueff_UFCM * ueff_UFCM * epostub / Rch));
3369             Gdep_wt_Cinv = -'g * NBS0F_1 * Rch / Cinv;
3370         end else begin
3371             ueff_UFCM = 2.0 * WFN * 2.0 * TFIS;
3372             Cinv = ueff_UFCM * EPSBOX * EPISB / EST;
3373             Rch = WFN * TFIS;
3374             r0 = (2.0 * Cinv / (ueff_UFCM * ueff_UFCM * epostub / Rch));
3375             Gdep_wt_Cinv = -'g * NBS0F_1 * Rch / Cinv;
3376         end
3377     end
3378 end else begin
3379     1.0e-1 320
```

5,444 lines of code

"Real" compact models: the BSIM family

```
[E] Modeling Body
3322 Inv_L = 1.0e-4 / (Leff1);
3323 Inv_WFin = 2.0 / WFin;
3324 Inv_WFin = 1.0e-4 / (Leff1 * WFin);
3325
3326 // Body Biasing Equation For UFB Parameters
3327 NBS0F_1 = NBS0F + Inv_L * LBS0F + Inv_WFin * NBS0F + Inv_WFin * PABS0F;
3328
3329 if (ABS0F1 != 0.0) begin
3330   NBS0F_1 = NBS0F_1 * (1.0 + ABS0F1/WFin * (1+(1.0 - WFin/ABS0F1^2)));
3331 end
3332
3333 // Model Parameters for Verified FinFET Compact Model by Jean Duriez 14/20
3334 case (CFIN0F)
3335 0: begin // Double Gate
3336   if (IsParam_given(TFIN_TOP)) { IsParam_given(TFIN_BOS1) begin
3337     wuff_UFCB = 2.0 * WFin;
3338     Cfin = wuff_UFCB * EPSIBK + EPISB + EST;
3339     Rch = WFin * TFIX;
3340     r0 = (2.0 * Cfin) / (wuff_UFCB * wuff_UFCB * epsub0 / Rch);
3341     Qcap_ox_Cfin = - q * NBS0F_1 * Rch / Cfin;
3342   end else begin
3343     wuff_UFCB = 2.0 * wuff_UFCB * WFin * (TFIN_TOP - TFIN_BOS1) * (TFIN_TOP
3344     Cfin = wuff_UFCB * EPSIBK + EPISB + EST;
3345     Rch = WFin * (TFIN_TOP + TFIN_BOS1) / 2.0;
3346     r0 = (2.0 * Cfin) / (wuff_UFCB * wuff_UFCB * epsub0 / Rch);
3347     Qcap_ox_Cfin = - q * NBS0F_1 * Rch / Cfin;
3348   end
3349 1: begin // Triple Gate
3350   if (IsParam_given(TFIN_TOP)) { IsParam_given(TFIN_BOS1) begin
3351     wuff_UFCB = 2.0 * WFin + TFIX;
3352     Cfin = wuff_UFCB * EPSIBK + EPISB + EST;
3353     Rch = WFin * TFIX;
3354     r0 = (2.0 * Cfin) / (wuff_UFCB * wuff_UFCB * epsub0 / Rch);
3355     Qcap_ox_Cfin = - q * NBS0F_1 * Rch / Cfin;
3356   end else begin
3357     wuff_UFCB = 2.0 * wuff_UFCB * WFin * (TFIN_TOP - TFIN_BOS1) * (TFIN_TOP +
3358     Cfin = wuff_UFCB * EPSIBK + EPISB + EST;
3359     Rch = WFin * (TFIN_TOP + TFIN_BOS1) / 2.0;
3360     r0 = (2.0 * Cfin) / (wuff_UFCB * wuff_UFCB * epsub0 / Rch);
3361     Qcap_ox_Cfin = - q * NBS0F_1 * Rch / Cfin;
3362   end
3363 2: begin // Quadruple Gate
3364   if (IsParam_given(TFIN_TOP)) { IsParam_given(TFIN_BOS1) begin
3365     wuff_UFCB = 2.0 * WFin + 2.0 * TFIX;
3366     Cfin = wuff_UFCB * EPSIBK + EPISB + EST;
3367     Rch = WFin * TFIX;
3368     r0 = (2.0 * Cfin) / (wuff_UFCB * wuff_UFCB * epsub0 / Rch);
3369     Qcap_ox_Cfin = - q * NBS0F_1 * Rch / Cfin;
3370   end else begin
3371     wuff_UFCB = 2.0 * wuff_UFCB * WFin * (TFIN_TOP - TFIN_BOS1) * (TFIN_TOP +
3372     Cfin = wuff_UFCB * EPSIBK + EPISB + EST;
3373     Rch = WFin * (TFIN_TOP + TFIN_BOS1) / 2.0;
3374     r0 = (2.0 * Cfin) / (wuff_UFCB * wuff_UFCB * epsub0 / Rch);
3375     Qcap_ox_Cfin = - q * NBS0F_1 * Rch / Cfin;
3376   end
3377   end else begin
3378     1.0e-1 324
```

5,444 lines of code

3.1.6 NFIN scaling equations

$$FINDL_N = FFINL_N \left[1 + \frac{FINDL_N}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{FINDL_N} \right) \right] \quad (3.76)$$

$$I_{D,OFF} = I_{D,OFF,NOM} \cdot FFINCEP_{L,off} \quad (3.77)$$

$$I_{D,OFF,N} = I_{D,OFF} \left[1 + \frac{I_{D,OFF}}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{I_{D,OFF}} \right) \right] \quad (3.78)$$

$$I_{D,ON} = I_{D,ON,NOM} \cdot FFINCEP_{L,on} \quad (3.79)$$

$$CINDEN = CINDEN \left[1 + \frac{CINDEN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{CINDEN} \right) \right] \quad (3.80)$$

$$CINDEN = CINDEN \left[1 + \frac{CINDEN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{CINDEN} \right) \right] \quad (3.81)$$

$$CINDEN = CINDEN \left[1 + \frac{CINDEN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{CINDEN} \right) \right] \quad (3.82)$$

$$XBS0FN = XBS0FN \left[1 + \frac{XBS0FN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{XBS0FN} \right) \right] \quad (3.83)$$

$$VFBATN = VFBATN \left[1 + \frac{VFBATN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{VFBATN} \right) \right] \quad (3.84)$$

$$VFBATN = VFBATN \left[1 + \frac{VFBATN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{VFBATN} \right) \right] \quad (3.85)$$

$$VFBATN = VFBATN \left[1 + \frac{VFBATN}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{VFBATN} \right) \right] \quad (3.86)$$

$$I_{D,N} = I_{D,N} \left[1 + \frac{I_{D,N}}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{I_{D,N}} \right) \right] \quad (3.87)$$

$$I_{D,N} = I_{D,N} \left[1 + \frac{I_{D,N}}{FFINL_N} + \ln \left(1 + \frac{FFINL_N}{I_{D,N}} \right) \right] \quad (3.88)$$

3.1.7 Length scaling equations

$$FINDL_N = FFINL_N \cdot FFINL_{L,off} \quad (3.89)$$

$$I_{D,N} = \left[\frac{I_{D,N}}{I_{D,N} - I_{D,N}^{OFF}} \right] \cdot I_{D,N} \quad (3.90)$$

$$\left[\frac{I_{D,N}}{I_{D,N} - I_{D,N}^{OFF}} \right] \cdot I_{D,N} = \left[\frac{I_{D,N}}{I_{D,N} - I_{D,N}^{OFF}} \right] \cdot I_{D,N} \quad (3.91)$$

$$\left[\frac{I_{D,N}}{I_{D,N} - I_{D,N}^{OFF}} \right] \cdot I_{D,N} = \left[\frac{I_{D,N}}{I_{D,N} - I_{D,N}^{OFF}} \right] \cdot I_{D,N} \quad (3.92)$$

"Real" compact models: the BSIM family

```
[E1] BSIM3v2 Body
3320 Inv_L = 1.0e-4 / (Leff1);
3321 Inv_WFin = 2.0 / WFin;
3324 Inv_LMFin = 1.0e-4 / (Leff1 * WFin);
3325
3326 // Study Biasing Equation For UICN Parameters
3327 MDSF1_1 = MDSF1 + Inv_L * LAMBDA + Inv_WFin * MDSF1 + Inv_LMFin * PMSD1;
3328
3329 IF (ABS(MDSF1) >= 6.0) begin
3330 MDSF1_1 = MDSF1_1 * (1.0 + ABS(MDSF1) * ((1.0 + WFin/ABS(MDSF1))))
3331 end
3332
3333 // Model Parameters for Verified FinFET Compact Model by Juan Suarez (14/20)
3334 case (USEMOS)
3335 0: begin // Double Gate
3336 IF (ISparam_given(TFIN_TOP)) { ISparam_given(TFIN_BOSZ)} begin
3337 ueff_UICN = 2.0 * WFin;
3338 Cinv = ueff_UICN * EPSIHS * EPISB / EST;
3339 Rch = WFin * TFIS;
3340 r0 = (2.0 * Cinv / (ueff_UICN * ueff_UICN * epostb / Rch));
3341 Qinv_uicn_Cinv = - q * MDSF1_1 * Rch / Cinv;
3342 and uicn begin
3343 uicn_UICN = 2.0 * epostb * WFin * WFin * (TFIN_TOP - TFIN_BOSZ) * (TFIN_TOP
3344 Cinv = ueff_UICN * EPSIHS * EPISB / EST;
3345 Rch = WFin * (TFIN_TOP + TFIN_BOSZ) / 2.0;
3346 r0 = (2.0 * Cinv / (ueff_UICN * ueff_UICN * epostb / Rch));
3347 Qinv_uicn_Cinv = - q * MDSF1_1 * Rch / Cinv;
3348 end
3349 1: begin // Triple Gate
3350 IF (ISparam_given(TFIN_TOP)) { ISparam_given(TFIN_BOSZ)} begin
3351 ueff_UICN = 2.0 * WFin * TFIS;
3352 Cinv = ueff_UICN * EPSIHS * EPISB / EST;
3353 Rch = WFin * TFIS;
3354 r0 = (2.0 * Cinv / (ueff_UICN * ueff_UICN * epostb / Rch));
3355 Qinv_uicn_Cinv = - q * MDSF1_1 * Rch / Cinv;
3356 and uicn begin
3357 ueff_UICN = 2.0 * epostb * WFin * (TFIN_TOP - TFIN_BOSZ
3358 Cinv = ueff_UICN * EPSIHS * EPISB / EST;
3359 Rch = WFin * (TFIN_TOP + TFIN_BOSZ) / 2.0;
3360 r0 = (2.0 * Cinv / (ueff_UICN * ueff_UICN * epostb / Rch));
3361 Qinv_uicn_Cinv = - q * MDSF1_1 * Rch / Cinv;
3362 end
3363 2: begin // Quadruple Gate
3364 IF (ISparam_given(TFIN_TOP)) { ISparam_given(TFIN_BOSZ)} begin
3365 ueff_UICN = 2.0 * WFin * TFIS;
3366 Cinv = ueff_UICN * EPSIHS * EPISB / EST;
3367 Rch = WFin * TFIS;
3368 r0 = (2.0 * Cinv / (ueff_UICN * ueff_UICN * epostb / Rch));
3369 Qinv_uicn_Cinv = - q * MDSF1_1 * Rch / Cinv;
3370 end uicn begin
3371 1.0e-1 320
```

5,444 lines of code

3.3.6 NFET scaling equations

$$V_{SAT1} = V_{SAT2} N_1 (1 - \beta_T (T - T_{NOM})) \quad (3.100)$$
$$V_{SAT2} = V_{SAT2} N_2 (1 - \beta_T (T - T_{NOM})) \quad (3.101)$$
$$V_{SAT3} = V_{SAT3} N_3 (1 - \beta_T (T - T_{NOM})) \quad (3.102)$$
$$I_{D0} = \exp\left(\frac{V_{SAT1} - V_{SAT2} - V_{SAT3} - V_{DSAT}}{N_{AS}}\right) \quad (3.103)$$
$$I_{D1} = I_{D0} \cdot I_{D1} \quad (3.104)$$
$$I_{D2} = I_{D0} \cdot I_{D2} \quad (3.105)$$
$$I_{D3} = I_{D0} \cdot I_{D3} \quad (3.106)$$
$$I_{D4} = I_{D0} \cdot I_{D4} \quad (3.107)$$
$$I_{D5} = I_{D0} \cdot I_{D5} \quad (3.108)$$
$$I_{D6} = I_{D0} \cdot I_{D6} \quad (3.109)$$
$$I_{D7} = I_{D0} \cdot I_{D7} \quad (3.110)$$
$$I_{D8} = I_{D0} \cdot I_{D8} \quad (3.111)$$
$$I_{D9} = I_{D0} \cdot I_{D9} \quad (3.112)$$
$$I_{D10} = I_{D0} \cdot I_{D10} \quad (3.113)$$
$$I_{D11} = I_{D0} \cdot I_{D11} \quad (3.114)$$

3.3.7

3.3.8

3.3.9

3.3.10

3.3.11

3.3.12

3.3.13

3.3.14

3.3.15

3.3.16

3.3.17

3.3.18

3.3.19

3.3.20

3.3.21

3.3.22

3.3.23

3.3.24

3.3.25

3.3.26

3.3.27

3.3.28

3.3.29

3.3.30

3.3.31

3.3.32

3.3.33

3.3.34

3.3.35

3.3.36

3.3.37

3.3.38

3.3.39

3.3.40

3.3.41

3.3.42

3.3.43

3.3.44

3.3.45

3.3.46

3.3.47

3.3.48

3.3.49

3.3.50

3.3.51

3.3.52

3.3.53

3.3.54

3.3.55

3.3.56

3.3.57

3.3.58

3.3.59

3.3.60

3.3.61

3.3.62

3.3.63

3.3.64

3.3.65

3.3.66

3.3.67

3.3.68

3.3.69

3.3.70

3.3.71

3.3.72

3.3.73

3.3.74

3.3.75

3.3.76

3.3.77

3.3.78

3.3.79

3.3.80

3.3.81

3.3.82

3.3.83

3.3.84

3.3.85

3.3.86

3.3.87

3.3.88

3.3.89

3.3.90

3.3.91

3.3.92

3.3.93

3.3.94

3.3.95

3.3.96

3.3.97

3.3.98

3.3.99

3.3.100

3.3.101

3.3.102

3.3.103

3.3.104

3.3.105

3.3.106

3.3.107

3.3.108

3.3.109

3.3.110

3.3.111

3.3.112

3.3.113

3.3.114

3.3.115

3.3.116

3.3.117

3.3.118

3.3.119

3.3.120

3.3.121

3.3.122

3.3.123

3.3.124

3.3.125

3.3.126

3.3.127

3.3.128

3.3.129

3.3.130

3.3.131

3.3.132

3.3.133

3.3.134

3.3.135

3.3.136

3.3.137

3.3.138

3.3.139

3.3.140

3.3.141

3.3.142

3.3.143

3.3.144

3.3.145

3.3.146

3.3.147

3.3.148

3.3.149

3.3.150

3.3.151

3.3.152

3.3.153

3.3.154

3.3.155

3.3.156

3.3.157

3.3.158

3.3.159

3.3.160

3.3.161

3.3.162

3.3.163

3.3.164

3.3.165

3.3.166

3.3.167

3.3.168

3.3.169

3.3.170

3.3.171

3.3.172

3.3.173

3.3.174

3.3.175

3.3.176

3.3.177

3.3.178

3.3.179

3.3.180

3.3.181

3.3.182

3.3.183

3.3.184

3.3.185

3.3.186

3.3.187

3.3.188

3.3.189

3.3.190

3.3.191

3.3.192

3.3.193

3.3.194

3.3.195

3.3.196

3.3.197

3.3.198

3.3.199

3.3.200

3.3.201

3.3.202

3.3.203

3.3.204

3.3.205

3.3.206

3.3.207

3.3.208

3.3.209

3.3.210

3.3.211

3.3.212

3.3.213

3.3.214

3.3.215

3.3.216

3.3.217

3.3.218

3.3.219

3.3.220

3.3.221

3.3.222

3.3.223

3.3.224

3.3.225

3.3.226

3.3.227

3.3.228

3.3.229

3.3.230

3.3.231

3.3.232

3.3.233

3.3.234

3.3.235

3.3.236

3.3.237

3.3.238

3.3.239

3.3.240

3.3.241

3.3.242

3.3.243

3.3.244

3.3.245

3.3.246

3.3.247

3.3.248

3.3.249

3.3.250

3.3.251

3.3.252

3.3.253

3.3.254

3.3.255

3.3.256

3.3.257

3.3.258

3.3.259

3.3.260

3.3.261

3.3.262

3.3.263

3.3.264

3.3.265

3.3.266

3.3.267

3.3.268

3.3.269

3.3.270

3.3.271

3.3.272

3.3.273

3.3.274

3.3.275

3.3.276

3.3.277

3.3.278

3.3.279

3.3.280

3.3.281

3.3.282

3.3.283

3.3.284

3.3.285

3.3.286

3.3.287

3.3.288

3.3.289

3.3.290

3.3.291

3.3.292

3.3.293

3.3.294

3.3.295

3.3.296

3.3.297

3.3.298

3.3.299

3.3.300

3.3.301

3.3.302

3.3.303

3.3.304

3.3.305

3.3.306

3.3.307

3.3.308

3.3.309

3.3.310

3.3.311

3.3.312

3.3.313

3.3.314

3.3.315

3.3.316

3.3.317

3.3.318

3.3.319

3.3.320

3.3.321

3.3.322

3.3.323

3.3.324

3.3.325

3.3.326

3.3.327

3.3.328

3.3.329

3.3.330

3.3.331

3.3.332

3.3.333

3.3.334

3.3.335

3.3.336

3.3.337

3.3.338

3.3.339

3.3.340

3.3.341

3.3.342

3.3.343

3.3.344

3.3.345

3.3.346

3.3.347

3.3.348

3.3.349

3.3.350

3.3.351

3.3.352

3.3.353

3.3.354

3.3.355

3.3.356

3.3.357

3.3.358

3.3.359

3.3.360

3.3.361

3.3.362

3.3.363

3.3.364

3.3.365

3.3.366

3.3.367

3.3.368

3.3.369

3.3.370

3.3.371

3.3.372

3.3.373

3.3.374

3.3.375

3.3.376

3.3.377

3.3.378

3.3.379

3.3.380

3.3.381

3.3.382

3.3.383

3.3.384

3.3.385

3.3.386

3.3.387

3.3.388

3.3.389

3.3.390

3.3.391

3.3.392

3.3.393

3.3.394

3.3.395

3.3.396

3.3.397

3.3.398

3.3.399

3.3.400

3.3.401

3.3.402

3.3.403

3.3.404

3.3.405

3.3.406

3.3.407

3.3.408

3.3.409

3.3.410

3.3.411

3.3.412

3.3.413

3.3.414

3.3.415

3.3.416

3.3.417

3.3.418

3.3.419

3.3.420

3.3.421

3.3.422

3.3.423

3.3.424

3.3.425

3.3.426

3.3.427

3.3.428

3.3.429

3.3.430

3.3.431

3.3.432

3.3.433

3.3.434

3.3.435

3.3.436

3.3.437

3.3.438

3.3.439

3.3.440

3.3.441

3.3.442

3.3.443

3.3.444

3.3.445

3.3.446

3.3.447

3.3.448

3.3.449

3.3.450

3.3.451

3.3.452

3.3.453

3.3.454

3.3.455

3.3.456

3.3.457

3.3.458

3.3.459

3.3.460

3.3.461

3.3.462

3.3.463

3.3.464

3.3.465

3.3.466

3.3.467

3.3.468

3.3.469

3.3.470

3.3.471

3.3.472

3.3.473

3.3.474

3.3.475

3.3.476

3.3.477

3.3.478

3.3.479

3.3.480

3.3.481

3.3.482

3.3.483

3.3.484

3.3.485

3.3.486

3.3.487

3.3.488

3.3.489

3.3.490

3.3.491

3.3.492

3.3.493

3.3.494

3.3.495

3.3.496

3.3.497

3.3.498

3.3.499

3.3.500

3.3.501

3.3.502

3.3.503

3.3.504

3.3.505

3.3.506

3.3.507

3.3.508

3.3.509

3.3.510

3.3.511

3.3.512

3.3.513

3.3.514

3.3.515

3.3.516

3.3.517

3.3.518

3.3.519

3.3.520

3.3.521

3.3.522

3.3.523

3.3.524

3.3.525

3.3.526

3.3.527

3.3.528

3.3.529

3.3.530

3.3.531

3.3.532

3.3.533

3.3.534

3.3.535

3.3.536

3.3.537

3.3.538

3.3.539

3.3.540

3.3.541

3.3.542

3.3.543

3.3.544

3.3.545

3.3.546

3.3.547

3.3.548

3.3.549

3.3.550

3.3.551

3.3.552

3.3.553

3.3.554

3.3.555

3.3.556

3.3.557

3.3.558

3.3.559

3.3.560

3.3.561

3.3.562

3.3.563

3.3.564

3.3.565

3.3.566

3.3.567

3.3.568

3.3.569

3.3.570

3.3.571

3.3.572

3.3.573

3.3.574

3.3.575

3.3.576

3.3.577

3.3.578

3.3.579

3.3.580

3.3.581

3.3.582

3.3.583

3.3.584

3.3.585

3.3.586

3.3.587

3.3.588

3.3.589

3.3.590

3.3.591

3.3.592

3.3.593

3.3.594

3.3.595

3.3.596

3.3.597

3.3.598

3.3.599

3.3.600

3.3.601

3.3.602

3.3.603

3.3.604

3.3.605

3.3.606

3.3.607

3.3.608

3.3.609

3.3.610

3.3.611

3.3.612

3.3.613

3.3.614

3.3.615

3.3.616

3.3.617

3.3.618

3.3.619

3.3.620

3.3.621

3.3.622

3.3.623

3.3.624

3.3.625

3.3.626

3.3.627

3.3.628

3.3.629

3.3.630

3.3.631

3.3.632

3.3.633

3.3.634

3.3.635

3.3.636

3.3.637

3.3.638

3.3.639

3.3.640

3.3.641

3.3.642

3.3.643

3.3.644

3.3.645

3.3.646

3.3.647

3.3.648

3.3.649

3.3.650

3.3.651

3.3.652

3.3.653

3.3.654

3.3.655

3.3.656

3.3.657

3.3.658

3.3.659

3.3.660

3.3.661

3.3.662

3.3.663

3.3.664

3.3.665

3.3.666

3.3.667

3.3.668

3.3.669

3.3.670

3.3.671

3.3.672

3.3.673

3.3.674

3.3.675

3.3.676

3.3.677

3.3.678

3.3.679

3.3.680

3.3.681

3.3.682

3.3.683

3.3.684

3.3.685

3.3.686

3.3.687

3.3.688

3.3.689

3.3.690

3.3.691

3.3.692

3.3.693

3.3.694

3.3.695

3.3.696

3.3.697

3.3.698

3.3.699

3.3.700

3.3.701

3.3.702

3.3.703

3.3.704

3.3.705

3.3.706

3.3.707

3.3.708

3.3.709

3.3.710

3.3.711

3.3.712

3.3.713

3.3.714

3.3.715

3.3.716

3.3.717

3.3.718

3.3.719

3.3.720

3.3.721

3.3.722

3.3.723

3.3.724

3.3.725

3.3.726

3.3.727

3.3.728

3.3.729

3.3.730

3.3.731

3.3.732

3.3.733

3.3.734

3.3.735

3.3.736

3.3.737

3.3.738

3.3.739

3.3.740

3.3.741

3.3.742

3.3.743

3.3.744

3.3.745

3.3.746

3.3.747

3.3.748

3.3.749

3.3.750

3.3.751

3.3.752

3.3.753

3.3.754

3.3.755

3.3.756

3.3.757

3.3.758

3.3.759

3.3.760

3.3.761

3.3.762

3.3.763

3.3.764

3.3.765

3.3.766

3.3.767

3.3.768

3.3.769

3.3.770

3.3.771

3.3.772

3.3.773

3.3.774

3.3.775

3.3.776

3.3.777

3.3.778

3.3.779

3.3.780

3.3.781

3.3.782

3.3.783

3.3.784

3.3.785

3.3.786

3.3.787

3.3.788

3.3.789

3.3.790

3.3.791

3.3.792

3.3.793

3.3.794

3.3.795

3.3.796

3.3.797

3.3.798

3.3.799

3.3.800

3.3.801

3.3.802

3.3.803

3.3.804

3.3.805

3.3.806

3.3.807

3.3.808

3.3.809

3.3.810

3.3.811

3.3.812

3.3.813

3.3.814

3.3.815

3.3.816

3.3.817

3.3.818

3.3.819

3.3.820

3.3.821

3.3.822

3.3.823

3.3.824

3.3.825

3.3.826

3.3.827

3.3.828

3.3.829

3.3.830

3.3.831

3.3.832

3.3.833

3.3.834

3.3.835

3.3.836

3.3.837

3.3.838

3.3.839

3.3.840

3.3.841

3.3.842

3.3.843

3.3.844

3.3.845

3.3.846

3.3.847

3.3.848

3.3.849

3.3.850

3.3.851

3.3.852

3.3.853

3.3.854

3.3.855

3.3.856

3.3.857

3.3.858

3.3.859

3.3.860

3.3.861

3.3.862

3.3.863

3.3.864

3.3.865

3.3.866

3.3.867

3.3.868

3.3.869

3.3.870

3.3.871

3.3.872

3.3.873

3.3.874

3.3.875

3.3.876

3.3.877

3.3.878

3.3.879

3.3.880

3.3.881

3.3.882

3.3.883

3.3.884

3.3.885

3.3.886

3.3.887

3.3.888

3.3.889

3.3.890

3.3.891

3.3.892

3.3.

"Real" compact models: the BSIM family

```

[1] BSIM3v3 Body
332  Inv_L = 1.0e-4 / (L*EFF);
333  Inv_WFN = 2.0 / WFN;
334  Inv_L*WFN = 1.0e-4 / (L*EFF * WFN);
335
336 // Body Biasing Equation For UFBT Parameters
337  BODYF_1 = -NDDOT * Inv_L * L*NDOT + Inv_WFN * NDDOT + Inv_L*WFN * P*NDOT;
338
339 IF (ABS(BODYF_1) > 6.0) begin
340  BODYF_1 = BODYF_1 * (1.0 + ABS(BODYF_1) * (1+(1.0 - WFN/ABS(BODYF_1))))
341 and
342 end
343
344 // Model Parameters for Verified FinFET Compact Model by Juan Duarte (4/2008)
345 case (LEVEL)
346 0: begin // Double Gate
347   IF (ISparam_given(TF2N_TOP)) { ISparam_given(TF2N_BDSE)} begin
348     ueff_UFBT = 2.0 * WFN;
349     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
350     Rth = WFN * TF2N;
351     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
352     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
353   and else begin
354     ueff_UFBT = 2.0 * epostub * WFN * TF2N * (TF2N_TDP - TF2N_TOP);
355     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
356     Rth = WFN * (TF2N_TOP + TF2N_BDSE) / 2.0;
357     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
358     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
359   and
360   end
361 1: begin // Triple Gate
362   IF (ISparam_given(TF2N_TOP)) { ISparam_given(TF2N_BDSE)} begin
363     ueff_UFBT = 2.0 * WFN * TF2N;
364     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
365     Rth = WFN * TF2N;
366     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
367     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
368   and else begin
369     ueff_UFBT = 2.0 * epostub * WFN * (TF2N_TOP - TF2N_BDSE) * (TF2N_TOP +
370     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
371     Rth = WFN * TF2N;
372     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
373     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
374   and
375   end
376 2: begin // Quadruple Gate
377   IF (ISparam_given(TF2N_TOP)) { ISparam_given(TF2N_BDSE)} begin
378     ueff_UFBT = 2.0 * WFN * TF2N;
379     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
380     Rth = WFN * TF2N;
381     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
382     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
383   and else begin
384     ueff_UFBT = 2.0 * epostub * WFN * TF2N * (TF2N_TDP - TF2N_TOP);
385     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
386     Rth = WFN * (TF2N_TOP + TF2N_BDSE) / 2.0;
387     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
388     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
389   and
390   end
391 3: begin // Quadruple Gate
392   IF (ISparam_given(TF2N_TOP)) { ISparam_given(TF2N_BDSE)} begin
393     ueff_UFBT = 2.0 * WFN * TF2N;
394     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
395     Rth = WFN * TF2N;
396     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
397     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
398   and else begin
399     ueff_UFBT = 2.0 * epostub * WFN * TF2N * (TF2N_TDP - TF2N_TOP);
400     Cinv = ueff_UFBT * EPSSUB * EPIS / EST;
401     Rth = WFN * (TF2N_TOP + TF2N_BDSE) / 2.0;
402     r0 = (2.0 * Cinv / (ueff_UFBT + ueff_UFBT * epostub / Rth));
403     Qsig_ox_Cinv = -q * NDDOT_1 * Rth / Cinv;
404   and
405   end
406 end
407
408 1.0e-12 320
    
```

5,444 lines of code

3.1.6 NFET scaling equations

$$V_{SAT}(I) = V_{SAT}(N) (1 - \beta_T (I - I_{DSTAT})) \quad (3.100)$$

$$V_{SAT}(I) = V_{SAT}(N) (1 - \beta_T (I - I_{DSTAT})) \quad (3.101)$$

$$V_{SAT}(I) = V_{SAT}(N) (1 - \beta_T (I - I_{DSTAT})) \quad (3.102)$$

$$F_{th} = \exp\left(\frac{V_{SAT}(I) - V_{SAT}(N)}{k_B T}\right) \quad (3.103)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.104)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.105)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.106)$$

$$F_{th} = \exp\left(\frac{V_{SAT}(I) - V_{SAT}(N)}{k_B T}\right) \quad (3.107)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.108)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.109)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.110)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.111)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.112)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.113)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.114)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.115)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.116)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.117)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.118)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.119)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.120)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.121)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.122)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.123)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.124)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.125)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.126)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.127)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.128)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.129)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.130)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.131)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.132)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.133)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.134)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.135)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.136)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.137)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.138)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.139)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.140)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.141)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.142)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.143)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.144)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.145)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.146)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.147)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.148)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.149)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.150)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.151)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.152)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.153)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.154)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.155)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.156)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.157)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.158)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.159)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.160)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.161)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.162)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.163)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.164)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.165)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.166)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.167)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.168)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.169)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.170)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.171)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.172)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.173)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.174)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.175)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.176)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.177)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.178)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.179)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.180)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.181)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.182)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.183)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.184)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.185)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.186)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.187)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.188)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.189)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.190)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.191)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.192)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.193)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.194)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.195)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.196)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.197)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.198)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.199)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.200)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.201)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.202)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.203)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.204)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.205)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.206)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.207)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.208)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.209)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.210)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.211)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.212)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.213)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.214)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.215)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.216)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.217)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.218)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.219)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.220)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.221)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.222)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.223)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.224)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.225)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.226)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.227)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.228)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.229)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.230)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.231)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.232)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.233)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.234)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.235)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.236)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.237)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.238)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.239)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.240)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.241)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.242)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.243)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.244)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.245)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.246)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.247)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.248)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.249)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.250)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.251)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.252)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.253)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.254)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.255)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.256)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.257)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.258)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.259)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.260)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.261)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.262)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.263)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.264)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.265)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.266)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.267)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.268)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.269)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.270)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.271)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.272)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.273)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.274)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.275)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.276)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.277)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.278)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.279)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.280)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.281)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.282)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.283)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.284)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.285)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.286)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.287)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.288)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.289)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.290)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.291)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.292)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.293)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.294)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.295)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.296)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.297)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.298)$$

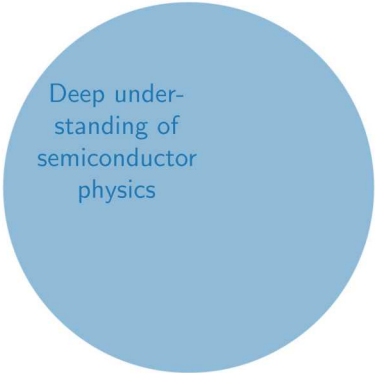
$$A_{eff}(I) = F_{th} F_{th} \quad (3.299)$$

$$A_{eff}(I) = F_{th} F_{th} \quad (3.300)$$


Pages and pages of equations (644 in all)

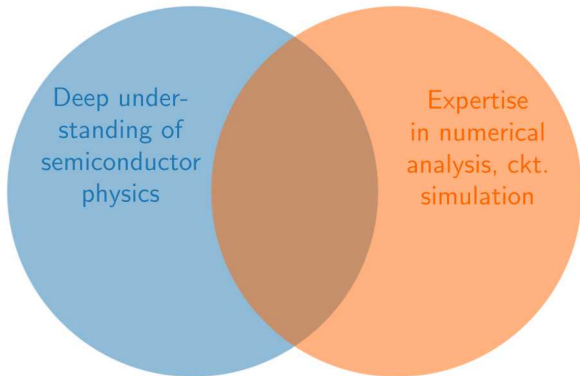
Several PhD students over 20+ years; this is just the current group.

Compact model development: what it takes

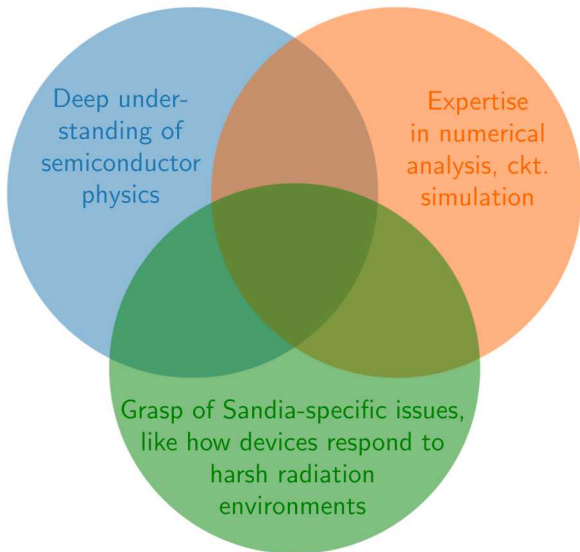


Deep understanding of
semiconductor
physics

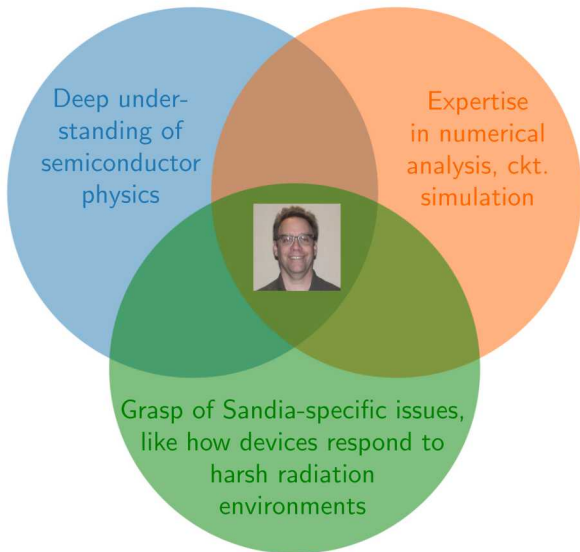
Compact model development: what it takes



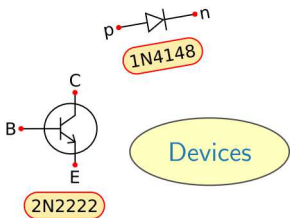
Compact model development: what it takes



Compact model development: what it takes



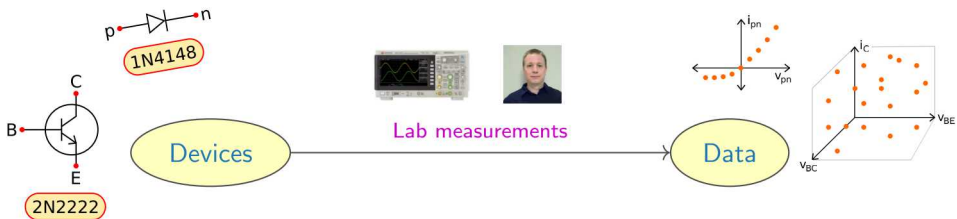
Our approach: Data-driven compact models (1/2)



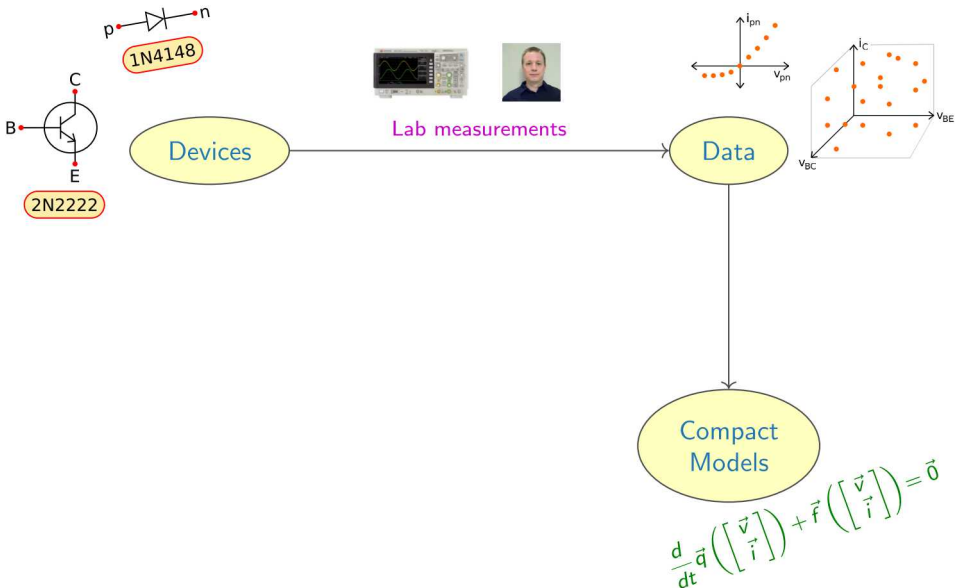
Our approach: Data-driven compact models (1/2)



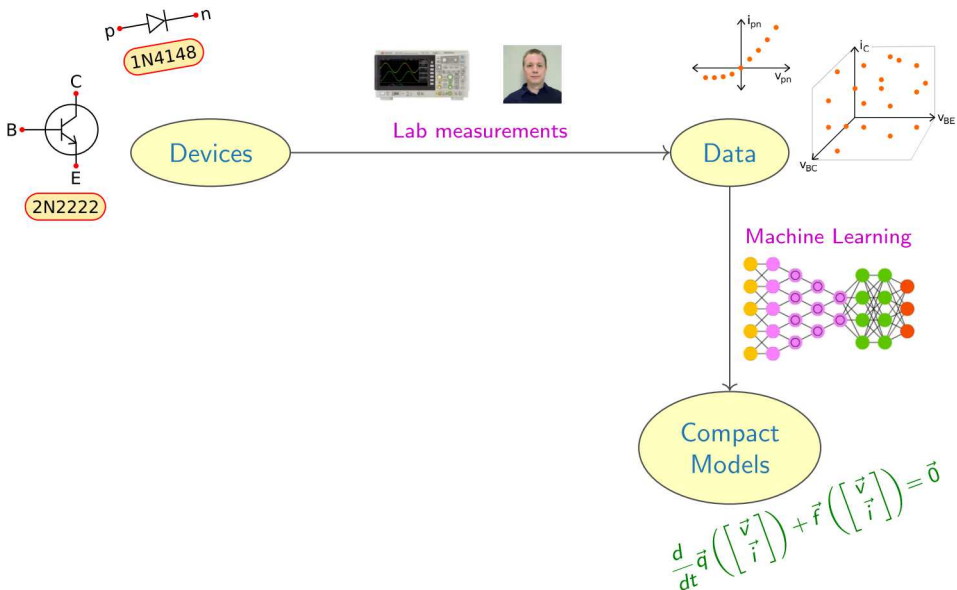
Our approach: Data-driven compact models (1/2)



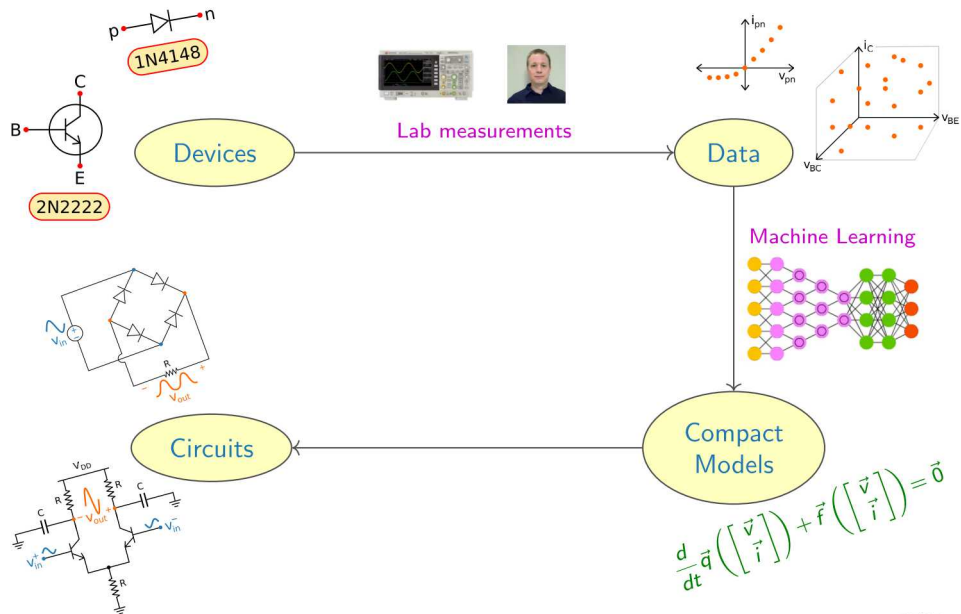
Our approach: Data-driven compact models (1/2)



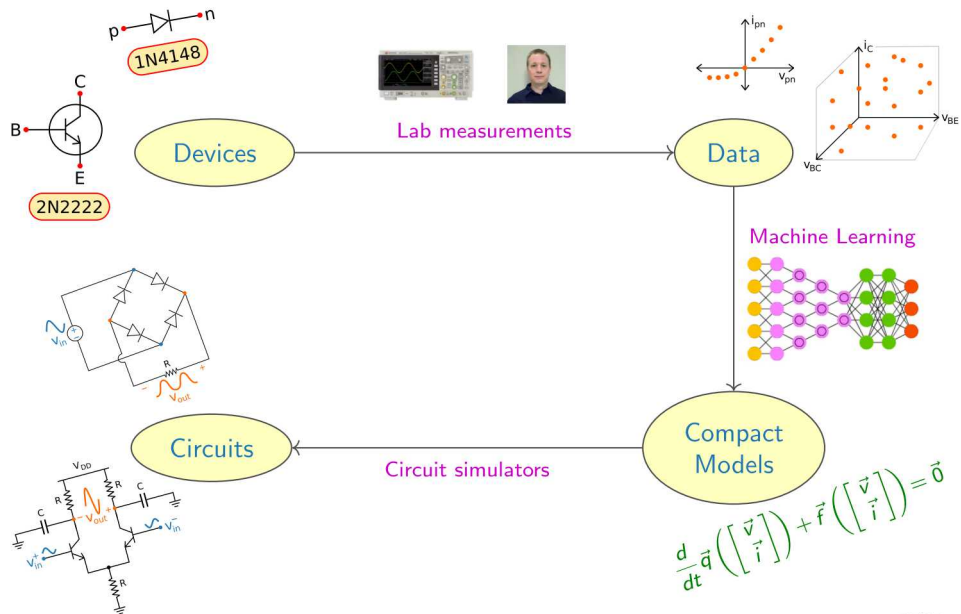
Our approach: Data-driven compact models (1/2)



Our approach: Data-driven compact models (1/2)



Our approach: Data-driven compact models (1/2)



Our approach: Data-driven compact models (2/2)

Task

Techniques we use

Software we use

Machine
Learning

Compact Model
Development

Circuit
Simulation

Our approach: Data-driven compact models (2/2)

Task

Techniques we use

Software we use

Machine
Learning





Backpropagation, deep
learning, neural networks

TensorFlow 







Compact Model
Development

Circuit
Simulation







Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
Compact Model Development		
Circuit Simulation		








Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley <small>UNIVERSITY OF CALIFORNIA</small>
Compact Model Development		
Circuit Simulation		











Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley <small>UNIVERSITY OF CALIFORNIA</small>
Compact Model Development	Differential-algebraic equation systems	
Circuit Simulation		











Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley <small>UNIVERSITY OF CALIFORNIA</small>
Compact Model Development	Differential-algebraic equation systems	ModSpec  Berkeley <small>UNIVERSITY OF CALIFORNIA</small>
Circuit Simulation		





















Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley UNIVERSITY OF CALIFORNIA
Compact Model Development	Differential-algebraic equation systems	ModSpec  Berkeley UNIVERSITY OF CALIFORNIA
		Spyce   
Circuit Simulation		






















Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley UNIVERSITY OF CALIFORNIA
Compact Model Development	Differential-algebraic equation systems	ModSpec  Berkeley UNIVERSITY OF CALIFORNIA
		Spyce   
Circuit Simulation	Newton-Raphson, limiting, linear multi-step methods like forward and backward Euler, small-signal circuit analysis, etc.	


























Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
Compact Model Development	Multi-dimensional splines	STEAM  Berkeley UNIVERSITY OF CALIFORNIA
	Differential-algebraic equation systems	ModSpec  Berkeley UNIVERSITY OF CALIFORNIA
Circuit Simulation	Newton-Raphson, limiting, linear multi-step methods like forward and backward Euler, small-signal circuit analysis, etc.	Spyce   
		Xyce          

Our approach: Data-driven compact models (2/2)

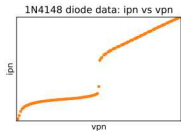
Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley UNIVERSITY OF CALIFORNIA
Compact Model Development	Differential-algebraic equation systems	ModSpec  Berkeley UNIVERSITY OF CALIFORNIA
		Spyce   
Circuit Simulation	Newton-Raphson, limiting, linear multi-step methods like forward and backward Euler, small-signal circuit analysis, etc.	         
		MAPP  Berkeley UNIVERSITY OF CALIFORNIA

Our approach: Data-driven compact models (2/2)

Task	Techniques we use	Software we use
Machine Learning	Backpropagation, deep learning, neural networks	TensorFlow 
	GMLS	Compadre    
	Multi-dimensional splines	STEAM  Berkeley UNIVERSITY OF CALIFORNIA
Compact Model Development	Differential-algebraic equation systems	ModSpec  Berkeley UNIVERSITY OF CALIFORNIA
		Spyce   
Circuit Simulation	Newton-Raphson, limiting, linear multi-step methods like forward and backward Euler, small-signal circuit analysis, etc.	  
		       
		MAPP  Berkeley UNIVERSITY OF CALIFORNIA
		Spyce   

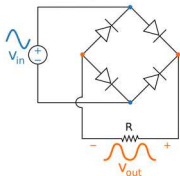
Results (1/2)

1N4148



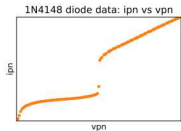
- We designed a full-wave rectifier circuit using data-driven compact models for a 1N4148 diode.

Data-driven
Compact Model

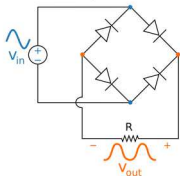


Results (1/2)

1N4148

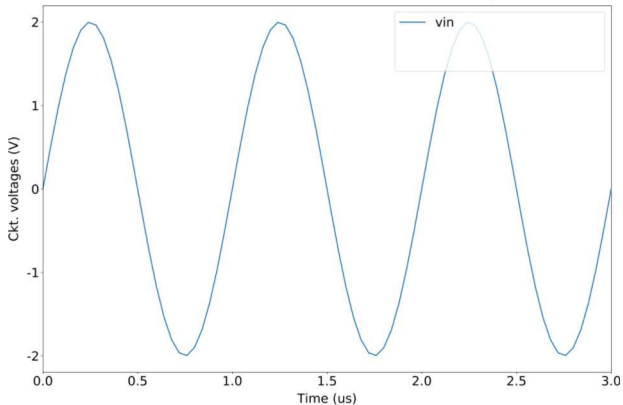


Data-driven
Compact Model



- We designed a full-wave rectifier circuit using data-driven compact models for a 1N4148 diode.

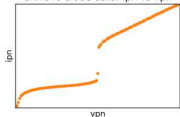
Full-wave rectifier: TRAN simulations with neural network, GMLS diodes



Results (1/2)

1N4148

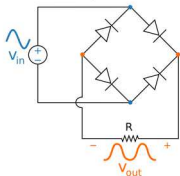
1N4148 diode data: ipn vs vpn



Neural networks via TensorFlow

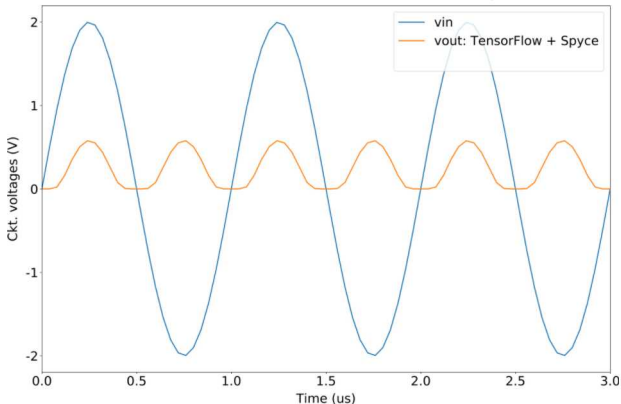
Data-driven Compact Model

Spice



- We designed a full-wave rectifier circuit using data-driven compact models for a 1N4148 diode.

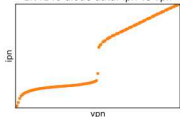
Full-wave rectifier: TRAN simulations with neural network, GMLS diodes



Results (1/2)

1N4148

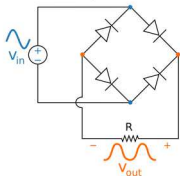
1N4148 diode data: ipn vs vpn



Neural networks via TensorFlow
GMLS via Compadre

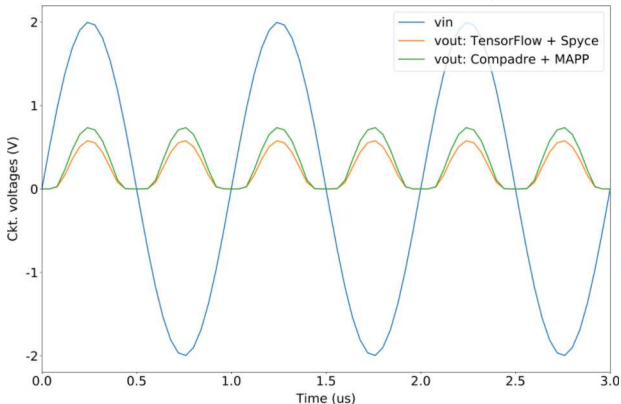
Data-driven Compact Model

Spyce MAPP



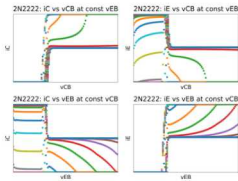
- We designed a full-wave rectifier circuit using data-driven compact models for a 1N4148 diode.

Full-wave rectifier: TRAN simulations with neural network, GMLS diodes



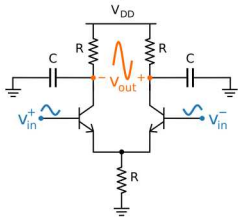
Results (2/2)

2N2222



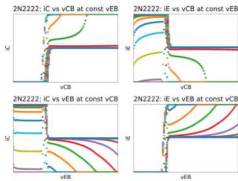
- We designed a differential-pair amplifier using a data-driven compact model for a 2N2222 BJT.

Data-driven
Compact Model



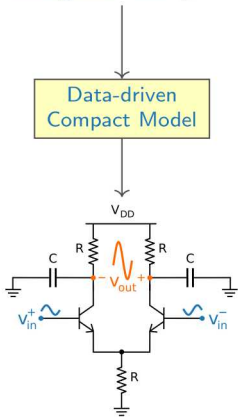
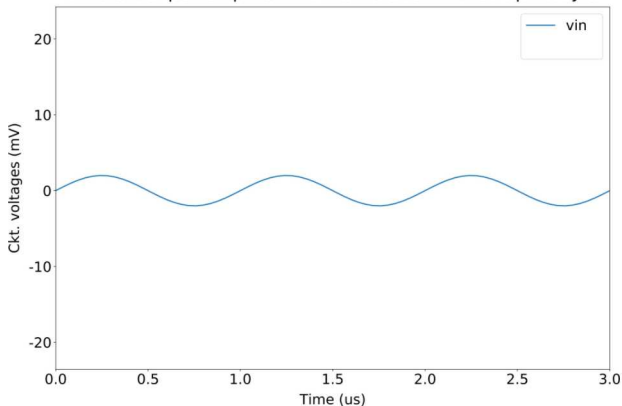
Results (2/2)

2N2222



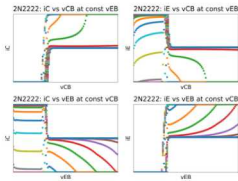
- We designed a differential-pair amplifier using a data-driven compact model for a 2N2222 BJT.

Differential-pair amplifier: TRAN simulation with 2D-spline BJTs



Results (2/2)

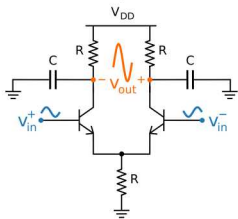
2N2222



2D splines
via SciPy

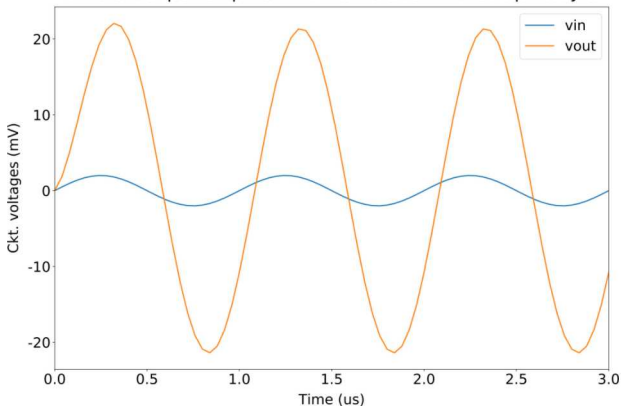
Data-driven
Compact Model

Spysce



- We designed a differential-pair amplifier using a data-driven compact model for a 2N2222 BJT.

Differential-pair amplifier: TRAN simulation with 2D-spline BJTs



Challenges

- Learning \vec{q} is harder than learning \vec{f}

$$\text{Compact model: } \frac{d}{dt} \vec{q} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) + \vec{f} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) = \vec{0}$$

- We have techniques to learn \vec{f} from DC data
- We are working on how to learn \vec{q} from small-signal impedance data

Challenges

- Learning \vec{q} is harder than learning \vec{f}

$$\text{Compact model: } \frac{d}{dt} \vec{q} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) + \vec{f} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) = \vec{0}$$

- We have techniques to learn \vec{f} from DC data
- We are working on how to learn \vec{q} from small-signal impedance data
- Newton-Raphson often fails with data-driven compact models
 - Idea 1. Construct models with better monotonicity
 - Idea 2. Limit oscillations with Predictor/Corrector Newton-Raphson
 - Idea 3. Extrapolation heuristics for GMLS

Challenges

- Learning \vec{q} is harder than learning \vec{f}

$$\text{Compact model: } \frac{d}{dt} \vec{q} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) + \vec{f} \left(\begin{bmatrix} \vec{v} \\ \vec{i} \end{bmatrix} \right) = \vec{0}$$

- We have techniques to learn \vec{f} from DC data
- We are working on how to learn \vec{q} from small-signal impedance data
- Newton-Raphson often fails with data-driven compact models
 - Idea 1. Construct models with better monotonicity
 - Idea 2. Limit oscillations with Predictor/Corrector Newton-Raphson
 - Idea 3. Extrapolation heuristics for GMLS
- The compact model form above may not be general enough
 - Cannot handle “dead delays”, transmission lines
 - We may need to generalize it further

Questions