Berkeley MAPP and VAPP
(Model and Algorithm Prototyping Platform)
(Verilog-A Parser and Processor)

Tianshi Wang, A. Gokcen Mahmutoglu, Karthik Aadithya*,
Archit Gupta and Jaijeet Roychowdhury

EECS Department, University of California, Berkeley
*Sandia National Laboratories
Compact Model Development

Compact Model Equations

ModSpec format
MATLAB

Test immediately (standalone)

Run Small Circuits in MAPP

Verilog-A

Is it the model?
Is it the simulator?
The equations?

Commercial Simulators

Open-source Simulators

Problems?

Compact Model Equations

ModSpec format
MATLAB

Test immediately (standalone)

Run Small Circuits in MAPP

Verilog-A

Is it the model?
Is it the simulator?
The equations?

Commercial Simulators

Open-source Simulators

Problems?

DC/AC/TRAN in MATLAB

modelformat itself eliminates some common modelling mistakes

→ smoothing
→ define custom functions/derivatives
→ custom init/limiting
→ gmin

→ model doesn't evaluate
→ overflow/domain errors
→ DC conv. failure
→ transient timestep too small
→ unphysical results
→ voltage/current blows up

Problems?

→ smoothing
→ define custom functions/derivatives
→ custom init/limiting
→ gmin

→ format itself eliminates some common modelling mistakes

→ model doesn't evaluate
→ overflow/domain errors
→ DC conv. failure
→ transient timestep too small
→ unphysical results
→ voltage/current blows up

Problems?

→ smoothing
→ define custom functions/derivatives
→ custom init/limiting
→ gmin

→ format itself eliminates some common modelling mistakes

→ model doesn't evaluate
→ overflow/domain errors
→ DC conv. failure
→ transient timestep too small
→ unphysical results
→ voltage/current blows up

Problems?
Compact Model Development

- Verilog-A
  - Is the model?
  - Is it a simulator?
  - The equations?

VAPP

ModSpec format
MATLAB

- Basic debugging
- Test immediately (standalone)

ModSpec C++

- Run small circuits in MAPP

MAPP

Compact Model Equations

- smoothing
- define custom functions
- custom init/limiting
- gmin

Commercial Simulators

- DC/AC/TRAN in MATLAB
- homotopy sensitivity
- PSS

Open-source Simulators

- Problems?
- Unphysical results
- Voltage/current blows up
- Transient timestep too small
- Overflow/domain errors
- DC conv. failure
- Model doesn't evaluate

T. Wang, UC Berkeley
What’s ModSpec: a glimpse

```matlab
function MOD = diodeCapacitor_ModSpec_wrapper()
% ModSpec description of an ideal diode in parallel with a capacitor
MOD = ee_model();
MOD = add_to_ee_model(MOD, 'external_nodes', {'p', 'n'});
MOD = add_to_ee_model(MOD, 'explicit Outs', {'ipn'});
MOD = add_to_ee_model(MOD, 'parms', {'C', 2e-12, 'Is', 1e-12, 'VT', 0.025});
MOD = add_to_ee_model(MOD, 'f', @f);
MOD = add_to_ee_model(MOD, 'q', @q);
end

function out = f(S)
  v2struct(S);
  out = Is*(exp(vpn/VT)-1);
end

function out = q(S)
  v2struct(S);
  out = C*vpn;
end
```

MOD. terminals
MOD. parms
MOD. explicit_ outs
MOD. f: function handle
MOD. q: function handle
...

\[
\vec{z} = \frac{d}{dt} \vec{q}_e(\vec{x}, \vec{y}) + \vec{f}_e(\vec{x}, \vec{y}, \vec{u})
\]

\[
\vec{0} = \frac{d}{dt} \vec{q}_i(\vec{x}, \vec{y}) + \vec{f}_i(\vec{x}, \vec{y}, \vec{u})
\]

Differential Algebraic Equations
What’s ModSpec: a glimpse

- Executable & debuggable standalone
- Easy to examine/write by hand
- General: any device in any physical domain
- Easily & directly usable by any simulator

MOD. terminals
MOD. parms
MOD. explicit_outs
MOD. f: function handle
MOD. q: function handle
...

- Supports every analysis DC/AC/tr/PSS

Mathematically well defined, modular

\[
\dot{z} = \frac{d}{dt} \bar{q}_e(\bar{x}, \bar{y}) + \bar{f}_e(\bar{x}, \bar{y}, \bar{u})
\]

\[
\bar{0} = \frac{d}{dt} \bar{q}_i(\bar{x}, \bar{y}) + \bar{f}_i(\bar{x}, \bar{y}, \bar{u})
\]

Differential Algebraic Equations
ModSpec: Model Debugging Example

MVS: “notch” in IDS at exactly VDS = zero

MVS: dIDS/dVDS

Verilog-A

Compact Model Equations

ModSpec format MATLAB

VAPP

Test immediately (standalone)

Basic debug

T. Wang, UC Berkeley
What's ModSpec: a glimpse

Executable & debuggable standalone

Easy to examine/write by hand

General: any device in any physical domain

Easily & directly usable by any simulator

Supports every analysis DC/AC/tr/PSS

Mathematically well defined, modular

\[
\vec{z} = \frac{d}{dt} \vec{q}_e(\vec{x}, \vec{y}) + \vec{f}_e(\vec{x}, \vec{y}, \vec{u})
\]

\[
\vec{0} = \frac{d}{dt} \vec{q}_i(\vec{x}, \vec{y}) + \vec{f}_i(\vec{x}, \vec{y}, \vec{u})
\]

Differential Algebraic Equations

MOD.terms
MOD.parms
MOD.explicit_outs
MOD.f: function handle
MOD.q: function handle
...
MAPP: Compact Model Prototyping

(a) BSIM6.1.0
default: L=10μm, W=10μm

(b) PSP Level 103 v3.0
default: L=10μm, W=10μm

(c) MVS v1.0.1
default: L=80nm, W=1μm

(d) MOS11 v2
default: L=1μm, W=1μm
What’s ModSpec: a glimpse

Executable & debuggable standalone

Easy to examine/write by hand

General: any device in any physical domain

Easily & directly usable by any simulator

Supports every analysis DC/AC/tr/PSS

Mathematically well defined, modular

MOD.terminals
MOD.parms
MOD.explicit_outs
MOD.f: function handle
MOD.q: function handle

\[ \dot{z} = \frac{d}{dt} q_e(\vec{x}, \vec{y}) + f_e(\vec{x}, \vec{y}, \vec{u}) \]

\[ \dot{0} = \frac{d}{dt} q_i(\vec{x}, \vec{y}) + f_i(\vec{x}, \vec{y}, \vec{u}) \]

Differential Algebraic Equations
ModSpec: Multiphysics Support

Optical
Network Interface Layer
- Electric fields, polarizations, modes, wavelengths, wave continuity, ...

Electrical
Network Interface Layer
- Node voltages, branch currents, KCL, KVL

MOD.terminals
MOD.parms
MOD.explicit_outs
MOD.f: function handle
MOD.q: function handle
...

\[
\dot{z} = \frac{d}{dt} \bar{q}_e(\bar{x}, \bar{y}) + \bar{f}_e(\bar{x}, \bar{y}, \bar{u})
\]

\[
\dot{0} = \frac{d}{dt} \bar{q}_i(\bar{x}, \bar{y}) + \bar{f}_i(\bar{x}, \bar{y}, \bar{u})
\]

Differential Algebraic Equations
Multiphysics Systems

**potential/flow systems:**

- **kinematic NIL:**
  - "flow": force
  - "potential": position

- **magnetic NIL:**
  - "flow": magnetic flux
  - "potential": magnetomotive force

- **thermal NIL:**
  - "flow": power flow
  - "potential": temperature

**Spintronic systems:**

- vectorized spin currents
- vectorized spin voltages

---

Chemical reaction networks

- *rates and concentrations*
- "KCLs" at nodes have $d/dt$ terms

---

Kerem Yunus Camsari; Samiran Ganguly; Supriyo Datta (2013), "Modular Spintronics Library," [https://nanohub.org/resources/17831](https://nanohub.org/resources/17831).

---

T. Wang, UC Berkeley
What’s ModSpec: a glimpse

Executable & debuggable standalone

Easy to examine/write by hand

General: any device in any physical domain

Easily & directly usable by any simulator

Supports every analysis DC/AC/tr/PSS

Mathematically well defined, modular

\[ \dot{z} = \frac{d}{dt} \bar{q}_e(\vec{x}, \vec{y}) + \vec{f}_e(\vec{x}, \vec{y}, \vec{u}) \]

\[ \dot{0} = \frac{d}{dt} \bar{q}_i(\vec{x}, \vec{y}) + \vec{f}_i(\vec{x}, \vec{y}, \vec{u}) \]

Differential Algebraic Equations
Glimpse: ModSpec Model in Xyce

1 *** Test-bench for generating dc response of an inverter
2
3 *** Create sub-circuit for the inverter
4 .subckt inverter Vin Vout Vvdd Vgnd
5
6 yModSpec_Device X1 Vvdd Vin Vout Vvdd /MVSmod:;type=-1 W=1.0e-4
7 Lgdr=32e-7 dLg=8e-7 Cg=2.57e-6 beta=1.6 alpha=3.5 Tj=300
8 Cif=1.38e-12 Cof=1.47e-12 phib=1.2 gamma=0.1 mc=0.2
9 CTM_select=1 Rs=100 Rd=100 n0=1.68 nd=0.1 vxo=7542204
10 mu=165 Vt0=0.5535 delta=0.15
11
12 yModSpec_Device X0 Vout Vin Vgnd Vgnd /MVSmod:;type=1 W=1.0e-4
13 Lgdr=32e-7 dLg=9e-7 Cg=2.57e-6 beta=1.8 alpha=3.5 Tj=300
14 Cif=1.38e-12 Cof=1.47e-12 phib=1.2 gamma=0.1 mc=0.2
15 CTM_select=1 Rs=100 Rd=100 n0=1.68 nd=0.1 vxo=1.2e7
16 mu=200 Vt0=0.4 delta=0.15
17
18 .model MVSmod MODSPEC_DEVICE SONAME=MVS_ModSpec_Element.so
19 .ends
20
21 *** circuit layout
22 Vsup sup 0 1
23 Vin in 0 0
24 Vsource source 0 0
25 X2 in out sup 0 inverter
26
27 *** simulation
28 .dc Vin 0 1 0.01
29 .print dc V(in) V(out)
30
31 .end
32 *** END
33 .end

Xyce netlist for inverter
(using MVS ModSpec/C++ model)

ModSpec Model
(C++ API)

compile standalone

.so libraries
(dynamically loadable)

Xyce-ModSpec Interface

model supported in Xyce

Updates in the last year

- limiting correction
- composite parameters
- works in Xyce 6.5
What’s ModSpec: a glimpse

Executable &
debuggable
standalone

Easy to
examine/write
by hand

General:
any device in
any physical
domain

Easily &
directly
usable by
any simulator

Supports
every analysis
DC/AC/tr/PSS

Mathematically
well defined,
modular

MOD. terminals
MOD. parms
MOD. explicit_outs
MOD. f: function handle
MOD. q: function handle

\[
\begin{align*}
\ddot{z} &= \frac{d}{dt} \overrightarrow{q_e}(\overrightarrow{x}, \overrightarrow{y}) + \overrightarrow{f_e}(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{u}) \\
\ddot{0} &= \frac{d}{dt} \overrightarrow{q_i}(\overrightarrow{x}, \overrightarrow{y}) + \overrightarrow{f_i}(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{u})
\end{align*}
\]

Differential Algebraic Equations
STEAM: Fast, Accurate Table-Based Models

- Compact model using only tabulated i-v, q-v data?
  - previous table-based attempts: important details unclear, poor accuracy, low speedup
  - our goal: can we speed up existing compact models?

- Our approach: STEAM
  - tabulate ModSpec functions fe, fi, qe, qi (one time cost)
  - device eval: multi-dimensional cubic spline interpolation

- Initial results
  - 150x eval speedup for BSIM3 (6-15x tran/DC)
  - relative error as low as you like: eg, $10^{-4}$
    - but memory requirements grow with accuracy

Replace with “lookup” tables

Implementation details: multi-dimensional splines, passive extrapolation

Differential Algebraic Equations
BSIM3 Inverter: STEAM vs Original

DC sweep

Transient

DC sweep: error

Transient: error
What’s ModSpec: a glimpse

Executable & debuggable standalone
Easy to examine/write by hand
General: any device in any physical domain
Easily & directly usable by any simulator
Supports every analysis DC/AC/tr/PSS
Mathematically well defined, modular

MOD. terminals
MOD. parms
MOD. explicit_outs
MOD. f: function handle
MOD. q: function handle

\[ \dot{z} = \frac{d}{dt} \overrightarrow{q_e}(\overrightarrow{x}, \overrightarrow{y}) + \overrightarrow{f_e}(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{u}) \]

\[ \dot{0} = \frac{d}{dt} \overrightarrow{q_i}(\overrightarrow{x}, \overrightarrow{y}) + \overrightarrow{f_i}(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{u}) \]

Differential Algebraic Equations
Compact Model Development

**Compact Model Equations**

- Verilog-A
- DC/AC/TRAN in MATLAB
- Homotopy sensitivity
- PSS

**ModSpec format**

- MATLAB
- C++
-format itself eliminates some common modelling mistakes

**VAPP**

- Test immediately
  
  - (standalone)

**Run Small Circuits in MAPP**

- Problems?
  - model doesn't evaluate
  - overflow/domain errors
  - DC conv. failure
  - transient timestep too small
  - unphysical results
  - voltage/current blows up

**Problems?**

- Is it the model?
- Is it the simulator?
- The equations?

- Basic debug:
  - smoothing
  - define custom functions/derivatives
  - custom init/limiting
  - gmin
VAPP: New Graph Based Core

Algorithm:
- Construct a spanning tree (ST)
- Designate branches in the ST as independent voltages
- Remaining branches are independent currents
- Construct loop and cutset matrices
- Express dependent quantities in terms of independent ones

How do we know that \{vn1i1, vi1i2\} are internal unknowns?
And \{vnci1, vnci2\} dependent voltages?
VAPP: What Is Still Lacking?

• Node collapse:

```plaintext
if (rdsmod == 0)
begin
    V(source, sourcep) <+ 0;
    V(drainp, drain)   <+ 0;
end
else
begin
    I(drain, drainp)   <+ type * gdtot * vded;
    I(source, sourcep) <+ type * gstot * vses;
end
```

Changes the number of unknowns.

• Separate networks (graphs) for different disciplines. E.g., thermal, magnetic,...
  Important for self heating.

• Support for noise functions in MAPP. E.g., white_noise, flicker_noise
Compact Model Development

Compact Model Equations

ModSpec format MATLAB

Run Small Circuits in MAPP

ModSpec C++

Test immediately (standalone)

Verbose debugging

DC/AC/TRAN in MATLAB

ModSpec format itself eliminates some common modelling mistakes

Problems?

Is it the model? Is it the simulator? The equations?

VAPP

Is it the model? Is it the compiler? The inputs?

Verilog-A

Commercial Simulators

Open-source Simulators

Problems?

➔ model doesn't evaluate
➔ overflow/domain errors
➔ DC conv. failure
➔ transient timestep too small
➔ unphysical results
➔ voltage/current blows up

➔ smoothing
➔ define custom functions
➔ custom init/limiting
➔ gmin

T. Wang, UC Berkeley
Memristive Devices & Applications

**devices**

UMich, Stanford, HP, HRL Labs, Micron, Crossbar, Samsung, ...

**applications**

- nonvolatile memories
- FPAAs
- neuromorphic circuits
- oscillators

**Compact Models**

- Linear/nonlinear ion drift models: Biolek (2009), Joglekar (2009), Prodromakis (2011), etc.
- UMich RRAM model (2011)
- TEAM model (2012)
- Simmons tunneling barrier model (2013)
- Yakopcic model (2013)
- Stanford/ASU RRAM model (2014)
- Knowm “probabilistic” model (2015)

**Verilog-A problems**

- idt(), $bound_step,
- $abstime, @initial_step,
- $rdist_normal, ...

**not one works in DC**
Challenges in Memristor Modelling

- **hysteresis**
  - internal state variable

- **model internal unks in Verilog-A**
  - use potentials/flows

- **upper/lower bounds of internal unks**
  - filament length, tunneling tap size
  - clipping functions

- **smoothness, continuity, finite precision issues, ...**
  - use smooth functions, safe functions
  - GMIN
  - scaling of unks/eqns
  - SPICE-compatible limiting function (the only smooth one)
How to Model Hysteresis Properly

**Template:**

\[
\text{ipn} = f_1(\text{vpn}, s)
\]

\[
\frac{d}{dt}s = f_2(\text{vpn}, s)
\]

**ModSpec:**

\[
\text{ipn} = \frac{dq_e(\text{vpn}, s)}{dt} + f_e(\text{vpn}, s)
\]

\[
0 = \frac{dq_i(\text{vpn}, s)}{dt} + f_i(\text{vpn}, s)
\]

![Circuit Diagram](image1.png)

![Graphs](image2.png)
How to Model Hysteresis Properly

**homotopy**

**top**

**side**

T. Wang, UC Berkeley
Memristor Models

\[ \frac{d}{dt}s = f_2(vpn, s) \]
\[ ipn = f_1(vpn, s) \]

Available \( f_2: \)

1. linear ion drift
   \[ f_2 = \mu_v \cdot R_{on} \cdot f_1(vpn, s) \]

2. nonlinear ion drift
   \[ f_2 = a \cdot vpn^m \]

3. Simmons tunnelling barrier
   \[ f_2 = \begin{cases} c_{off} \cdot \sinh\left(\frac{i}{i_{off}}\right) \cdot \exp(-\exp\left(\frac{s-a_{off}}{w_c} - \frac{i}{b}\right) - \frac{s}{w_c}), & \text{if } i \geq 0 \\ c_{on} \cdot \sinh\left(\frac{i}{i_{on}}\right) \cdot \exp(-\exp\left(\frac{a_{on} - s}{w_c} + \frac{i}{b}\right) - \frac{s}{w_c}), & \text{otherwise,} \end{cases} \]

4. TEAM model

5. Yakopcic model

6. Stanford/ASU
   \[ f_2 = -v_0 \cdot \exp\left(-\frac{E_a}{V_T}\right) \cdot \sinh\left(\frac{VPN \cdot \gamma \cdot a_0}{t_{ox} \cdot V_T}\right) \]

Available \( f_1: \)

1. \[ f_1 = (R_{on} \cdot s + R_{off} \cdot (1 - s))^{-1} \cdot vpn \]

2. \[ f_1 = \frac{1}{R_{on}} \cdot e^{-\lambda \cdot (1 - s)} \cdot vpn \]

3. \[ f_1 = s^n \cdot \beta \cdot \sinh(\alpha \cdot vpn) + \chi \cdot (\exp(\gamma \cdot t) - 1) \]

4. \[ f_1 = \begin{cases} A_1 \cdot s \cdot \sinh(B \cdot vpn), & \text{if } vpn \geq 0 \\ A_2 \cdot s \cdot \sinh(B \cdot vpn), & \text{otherwise.} \end{cases} \]

5. \[ f_1 = I_0 \cdot e^{-\frac{Gap}{g_0}} \cdot \sinh\left(\frac{VPN}{V_0}\right) \]
   \[ Gap = s \cdot \text{minGap} + (1 - s) \cdot \text{maxGap}. \]

- set up boundary
- fix \( f_2 \) flat regions
- smooth, safe funcs, scaling, etc.
Memristor Models

A collection of 30 models:

- all smooth, all well posed
- not just RRAM, but general memristive devices
- not just bipolar, but unipolar
- not just DC, AC, TRAN, but homotopy, PSS, ...

PSS using HB
ESD Snapback Model


\[ I_{\text{on}} = G_{\text{on}} \cdot (V - V_H) \]

\[ I_{\text{off}} = I_s \cdot (1 - e^{-V/\phi_T}) \cdot \sqrt{1 + \frac{\max(V,0)}{V_D}} \]

\[ I = s \cdot I_{\text{on}} + I_{\text{off}} \]

\[ \frac{d}{dt} s = f(V, s) \]

T. Wang, UC Berkeley
ESD Snapback Model

forward/backward
DC sweeps

homotopy

transient
voltage sweeps

"impact ionization doesn't happen instantaneously (although all compact models assume that it does)" — C.C. McAndrew

Human Body Mode (HBM) test

T. Wang, UC Berkeley
Compact Model Development

- Compact Model Equations
  - ModSpec format
    - MATLAB
  - ModSpec C++
  - Run Small Circuits in MAPP

VAPP

- Problems?
  - Is the model?
  - Is it the simulator?
  - The equations?

Commercial Simulators

Open-source Simulators

- DC/AC/TRAN in MATLAB
  - homotopy sensitivity PSS

- Verilog-A

- format itself eliminates some common modelling mistakes

- Problems?
  - model doesn't evaluate
  - overflow/domain errors
  - conv. failure
  - transient timestep too small
  - unphysical results
  - voltage/current blows up

Basic debug

Test immediately (standalone)

Run Small Circuits in MAPP

T. Wang, UC Berkeley