NEEDS Annual Review May 18-19, 2016

NEEDS Infrastructure/Tools: Berkeley MAPP and VAPP
(Model and Algorithm Prototyping Platform)
(Verilog-A Parser and Processor)

Jaijeet Roychowdhury
EECS Department
University of California at Berkeley
MAPP/VAPP: Contributors

Core Contributors:
1. Jaijeet Roychowdhury, P.I.
2. Tianshi Wang (4th yr PhD student)
3. Gokcen Mahmutoglu (post-doc)
4. Archit Gupta (1st yr PhD student)
5. Karthik Aadithya (graduated Jan '16)
6. Xufeng Wang (Purdue)

Special thanks to:

Eric Keiter
Sandia Labs

Colin McAndrew
Freescale/NXP

Geoffrey Coram
Analog Devices

Peter Bermel
Purdue
NEEDS Tools for Device Model Development

NEEDS compatible Verilog-A

Compact Model Equations

Write in MATLAB (ModSpec format)

Test immediately (standalone)

Run Small Circuits in MAPP

Verilog-A Parser and Processor

DC/AC/TRAN/etc in MATLAB

• format itself eliminates some common modelling mistakes

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

Problems? No

Yes

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

code/facilities for inspection and debugging
Model Development Flow (2)

Step 2: 

- NEEDS-compatible Verilog-A model
  - Already tested in MAPP → High probability of success
  - Use model in Commercial Simulators
    - High probability of success via ModSpec C++ API support
  - Model supported in Open-source Simulators (Xyce)
    - Speed near native implementation (compiled C++ code)
    - Via ModSpec C++ API support (easy: example provided for Xyce)
  - Compile standalone .so libraries (dynamically loadable)
    - Fast/efficient
    - Model supported in Open-source Simulators (Xyce)
  - Compile standalone
    - So libraries (dynamically loadable)
  - Standalone proof of model's convergence/speed performance
    - Confirm model with DC/AC/TRAN in C++ MAPP
  - "Simulation ready" model deployed
    - "Simulation ready" model deployed
    - Compile standalone
    - So libraries (dynamically loadable)
    - Model supported in Open-source Simulators (Xyce)
      - Speed near native implementation (compiled C++ code)
      - Via ModSpec C++ API support (easy: example provided for Xyce)
VAPP (Verilog-A Parser and Processor)

- **Verilog-A to ModSpec translator: total rewrite**
  » for flexibility, modularity, cleanliness, extensibility
  » fully-fledged parser/preprocessor/AST processor written from scratch in MATLAB
  » generates derivative code (10-20x faster than automatic differentiation in MAPP)

- **Working on:** MVS 1, MVS 2, MVS GaN HEMT, BSIM3, BSIM4, BSIM6, R3*, Purdue FEFET
  » PSP, VBIC, MEXTRAM, HICUM, HISIM, etc. in progress

- **NEEDS Simulation-Ready Modelling Guide draft**
  » updates in progress; to be released by Aug 31, 2016
VAPP: Examples, Modularity

BSIM3 ring osc: transient
BSIM3 VA translated by VAPP

(fixed) Purdue FEFET: homotopy
FEFET VA translated by VAPP

VAPP code: modular internal structure

Create IRT

Modify IRT over multiple passes

Intermediate representation tree (IRT)

VAPP: Examples, Modularity

BSIM3 ring osc: transient
BSIM3 VA translated by VAPP

(fixed) Purdue FEFET: homotopy
FEFET VA translated by VAPP

VAPP code: modular internal structure

Create IRT

Modify IRT over multiple passes

Intermediate representation tree (IRT)
VAPP: used by VAlint

- **VAlint** (UI tool by Xufeng Wang) now uses VAPP as its syntax checking engine
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?
    – w negligible loss of accuracy?

• 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
BSIM3 Inverter: STEAM vs Original

DC sweep

Transient

DC sweep: error

Transient: error
STEAM: Speedup, Accuracy

Raw Device Eval Speedup

Resulting Simulation Speedup

Worst-Case Relative Accuracy: Raw Device Eval

Why is MVS accuracy so poor?
Model Issues affect STEAM Accuracy

MVS: $\frac{dI_D}{dV_D}$

"notch" at $V_D = -1.2$

BSIM3: $\frac{dI_D}{dV_D}$

$V_D$
ModSpec: Debugging Model Issues

- finding such physics issues
  - made possible by ModSpec
    - standalone eval of internal device functions

\[
\frac{d}{dt} q_e \left( x(t), y(t) \right) + f_e \left( x(t), y(t) \right) = \ddot{z},
\]

\[
\frac{d}{dt} q_i \left( x(t), y(t) \right) + f_i \left( x(t), y(t) \right) = \vec{0}.
\]
Memristive Device (incl. RRAM) modelling

● Previous memristive/RRAM device models: ill-posedness + implementation issues
  » unphysical predictions, DC response problems, ...

● Well-posed modelling methodology for generic memristive devices (including RRAM)
  » uses only continuous/smooth functions
  » length/gap always within physical limits
  » well-defined, physical, DC operating points
  » proper implementations

● Existing RRAM + memristor models fixed
  » more detail: Tianshi's poster and afternoon talk
Other Progress Highlights ('15-'16)

- **MAPP (Model and Algorithm Prototyping Platform)**
  - core improvements
    - homotopy, vv4, documentation/help, myriad little things… (499 git commits)
    - multi-physics support: master/slave equation engines; spintronic, thermal, mechanical, opto, biochemical NILs and equation engines
  - case studies
    - MIT FE-FET model: ModSpec implementation + circuit experiments
    - Purdue FE-FET: ill-posedness problems discovered and communicated
    - R3: small Verilog-A macro syntax issue, being discussed w Colin
    - BSIM3: discontinuities, vds=0 notch in dIDS/dVDS found
    - MVS1: ill-posedness issues discovered, working with Dimitri to address
  - C++ MAPP
    - ModSpec + DAEAPI + Equation Engines coded in C++
    - working with MATLAB algorithm implementations (via mex interface)

- **ModSpec C++ API and Xyce-ModSpec Interface**
  - now working with Xyce 6.4; handover to Sandia in progress
Examples: Homotopy on Memristors, FE-FETs

Homotopy on memristor model (v-i-s characteristics)

3d (v vs i vs s)

Top view (v vs i)

Side view (v vs s)

MIT FE-FET model: char curves, “inverter” transfer characteristics
Examples: MAPP Multi-Physics Capabilities

Electro-Thermal

ModSpec Core

- I/O name
- I/O type
- I/O nodes
- eeNIL
- thermalNIL

Equations:
- \( p_{wr.t} = -i_{pn} \cdot v_{pn} \)
- \( 0 = i_{pn} - R_{0} + k \cdot (t_{temp.t} - T_{0}) \)

vsoc-thermistor: transient using GEAR2 LMS solver

Master Equation Engine

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

Spintronic

Bio-chemical

J. Roychowdhury, Berkeley
Progress: Education/Outreach

- Concepts/code taught in two classes at Berkeley
  - EE219A (Numerical Simulation and Modelling)
    - fundamental concepts of simulation/modelling
      - equation formulations, device models, analyses (DC/AC/transient)...
      - code components of MAPP (simplified)
  - EE290A (Numerical Simulation and Modelling 2)
    - MAPP used for HWs, demos and projects
      - table-based model work (Archit Gupta)
      - Chebyshev polynomial based PSS and transient (Kosta Trotkovsky)
- Feb 4, 2016: MAPP/VAPP Hands-on Workshop at Berkeley
  - ~25 attendees; MAPP and VAPP software released for workshop
- CICC 2015 paper/presentation on MAPP (thanks → Colin/Larry)
- Well-posed memristor/RRAM paper draft (up on arXiv)
- Table-based modelling paper (submitted to ICCAD)
- Chebyshev poly. transient paper (in prep. for ASP-DAC)
- Upcoming talk at CMC meeting (June 9, Austin)
- (in progress) book on simulation-ready modelling (World Scientific)
Our other Presentations and Posters

- Posters: 12pm
  - VAPP: Gokcen Mahmutoglu
  - STEAM (Table Based): Archit Gupta
  - MAPP Multi-Physics: Tianshi Wang
  - Well-posed Memristor/RRAM models: Tianshi Wang

- 2pm: Tianshi's talk on well-posed Memristor and RRAM models
Plans for '16-17

- By Aug 31, 2016
  - updated release of MAPP; first public release of VAPP
    - homotopy, multi-physics, memristor/RRAM models, ...
  - Xyce-ModSpec interface integrated with Xyce 6.5(?)
  - NEEDS Modelling Guide – first version + code
  - MVS debugging (w Dimitri)

- By September 2017
  - VAPP: many enhancements (more VA constructs supported, better error reporting, C++ ModSpec generation), testing on many more models, ...
  - MAPP: table-based models, memristor/RRAM updates, hierarchical ModSpec, modelling guide updates, refactoring, ...
  - C++-MAPP release