Modeling Tools Built Upon the HDL Foundation

H. Alan Mantooth

Department of Electrical Engineering

University of Arkansas

Fayetteville, AR 72701

mantooth@uark.edu

MSCAD Laboratory

- Knowle
  - The students who contributed to Paragon:
    O. Abbasi, A. Austin, V. Chaudhary, Y. Feng
    M. Francis, C. Gathercole, H. Gunupudi, N. Hingora,
    X. Huang, J. Kutchka, P. Mallick, J. Mao, H. Nguyen,
    E. Pettis, C. Vemulapally, and W. Zheng.
  - The organizations that funded Paragon and rel
    activities
    NSF, DARPA, ONR, SRC and the UA
- http://mixedsignal.eleg.uark.edu

Lynguent

Synopsys

Wal*Mart

Intel

Boeing

Mentor

Apache

Purdue

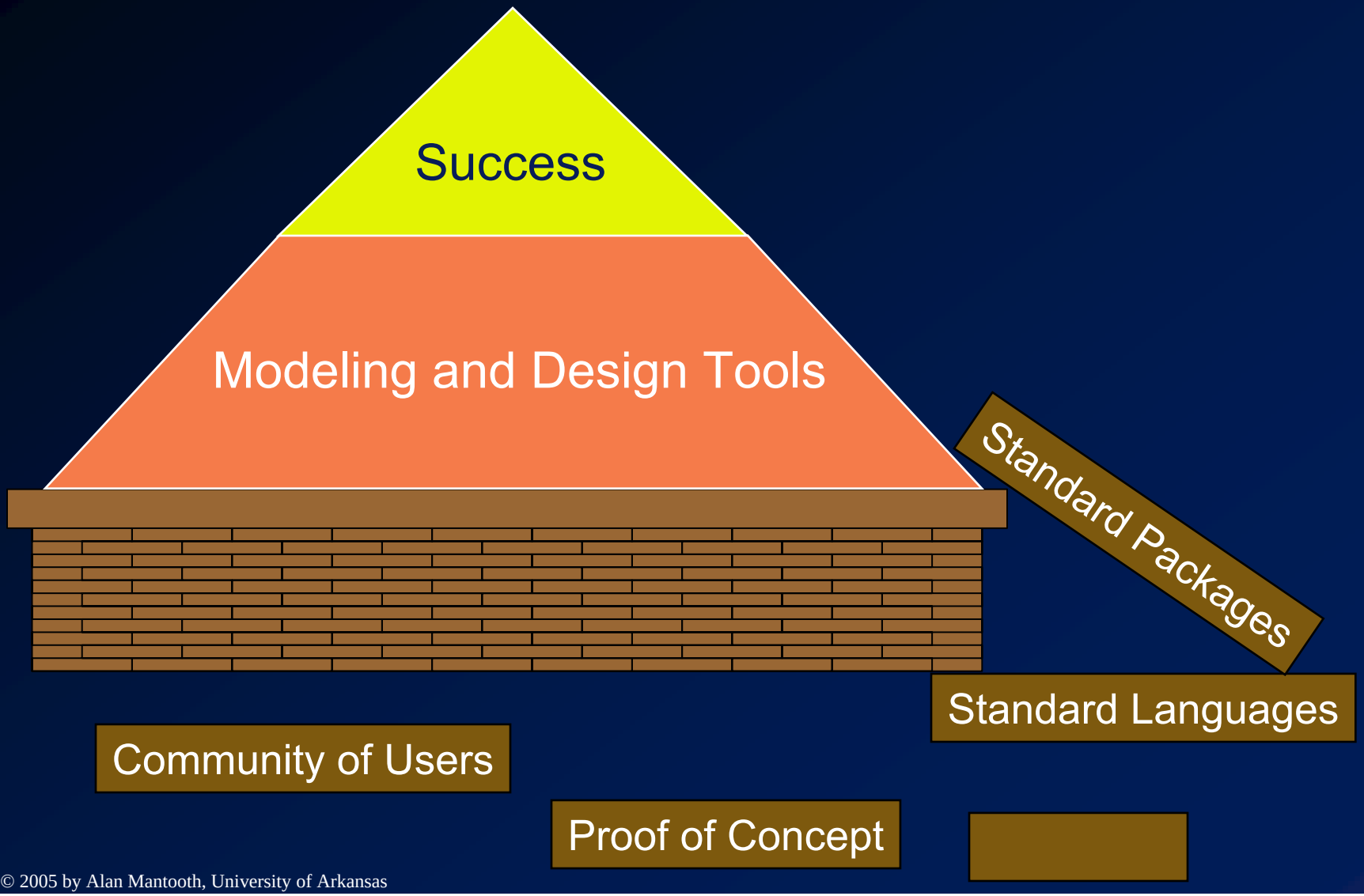
Microsoft

Intellisense

Outline

- The HDL Technological Foundation
- The Paragon Modeling Environment
  - What is Paragon?
    - Current Features and Limitations
    - Architecture
  - How do you use Paragon? How can Paragon simplify compact modeling?
    - Examples/Demo

The HDL Foundation



Anecdotal Evidence of Need

Concurrent design activity

Electronics for Extreme Environments

What's the definition of a *MUST* Have?

- Are mixed-signal designs successfully taping out today?
- Are advanced power electronic designs being manufactured today?
- Are mixed-technology systems being designed and built successfully today?

…so how many of these situations depend on the HDL foundation for success?

How many more could benefit and become dependent on this technology with improved tools?

“Behavioral” Modeling

- Rapidly evolving designs require availability of abstractions for unavailable circuits (models created AS a design activity)
- Designs based on lots of reuse benefit greatly from multiple levels of abstraction in system design (models created as an offline activity)
- Embedded systems drive analog performance to new levels. Leading edge designs demand abstractions to manage complexity.

Semiconductor Device Modeling

- Device Models are Vital to Circuit Designers
- SPICE
  - IC-based history
  - Few power device models available (mostly Si)
  - New models not easy to implement
- HDL-based simulators preferred
  - Relative ease of model implementation
  - Poor execution speed (compilers help)
  - Debugging is still poor

Outline

- The HDL Technological Foundation
- The Paragon Modeling Environment
  - What is Paragon?
    - Current Features and Limitations
    - Architecture
  - How do you use Paragon? How can Paragon simplify compact modeling?
    - Examples/Demo

Paragon Modeling Environment

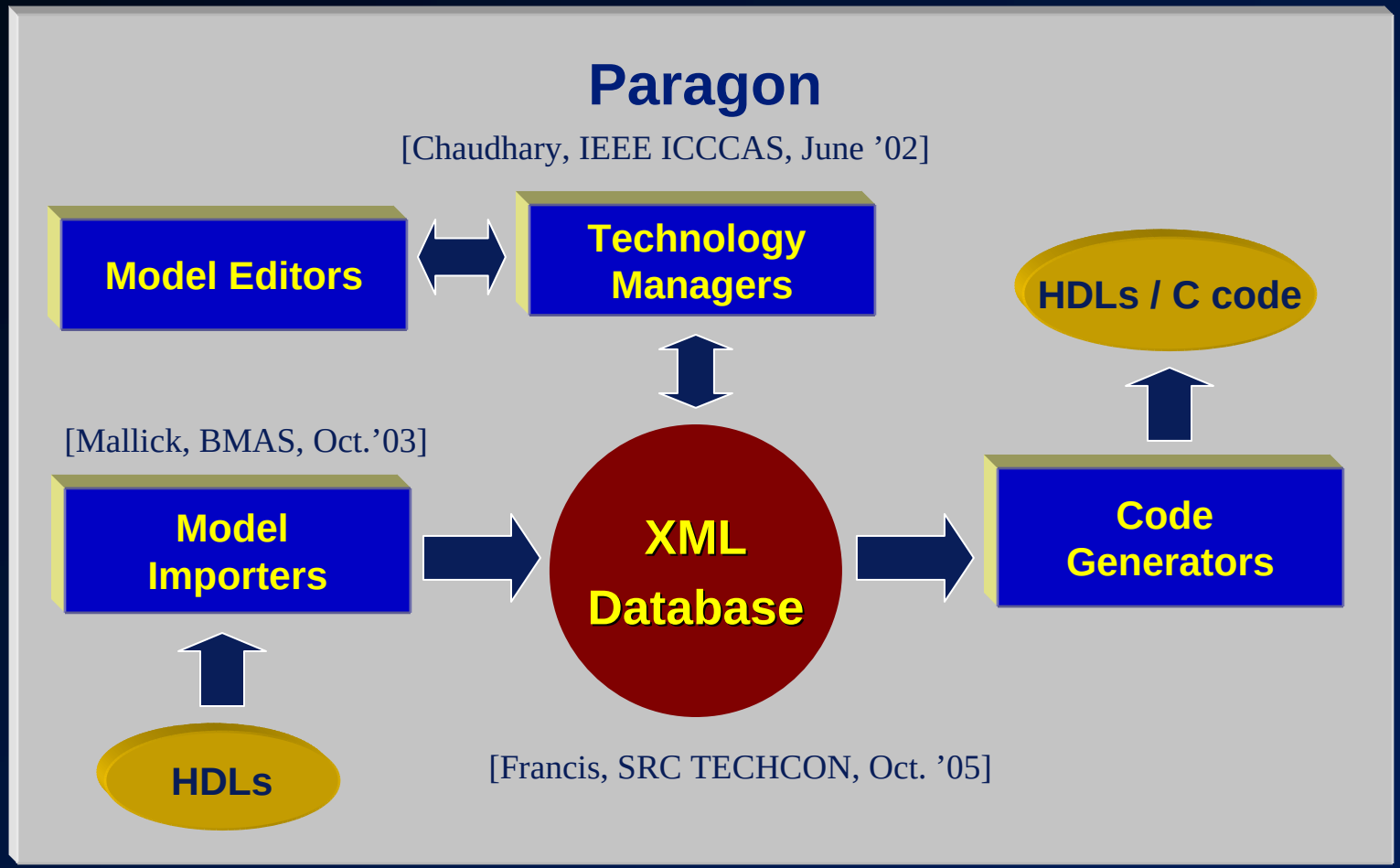
An HDL-based modeling tool for:

- Behavioral model creation*
- Compact model creation*
- Model visualization†
- Model diagnosis & debugging†
- Model characterization†
- HDL model import-export*
- Extensibility by users†

* - features have been extensively explored

† - has limitations compared to ultimate goal

Paragon Architecture



Abstract Model Representation

- Model information saved in a Common Model Interchange Format (CMIF)
- CMIF is basically XML/MathML
- Independent of specific language/simulator
- Able to capture constructs of VHDL-AMS, Verilog-AMS and MAST
- XML is used because it is
  - extensible
  - simple, flexible and structured
  - open-source and standardized
  - lots of existing XML-based tools
  - enables easy sharing of model data

CMIF: A common model interchange format

- XML-based: Open source
  - Many parsers available (SAX/DOM,etc)
  - Many utilities available: XSLT, CSS, etc
  - Easily extensible!

Consider effort of adding a new language front-end (VHDL/Verilog) compared to integrating a freely available XML parser!

EDIF or GDSII for analog models!

CMIF: A common model interchange format

Language semantics: VHDL-AMS vs CMIF

```
generic (     Temperature: real := 300.0;       --Temperature
              Iss: real := 1.0e-14;             -- bulk junction saturation current
              q: real := 1.6e-19;               -- Electronic Charge);
```

```
<interface>
    <parameter name="Temperature" type="real" default="300.0">
        <comment>Temperature</comment>
    <parameter name="Iss" type="real" default="1.0e-14">
        <comment>bulk junction saturation current</comment>
    <parameter name="q" type="real" default="1.6e-19">
        <comment>Electronic Charge</comment>
</interface>
```

1) Which is easier to write? Which is easier to parse?
2) What if I want to add a process/instance flag? Which would be easier to modify?

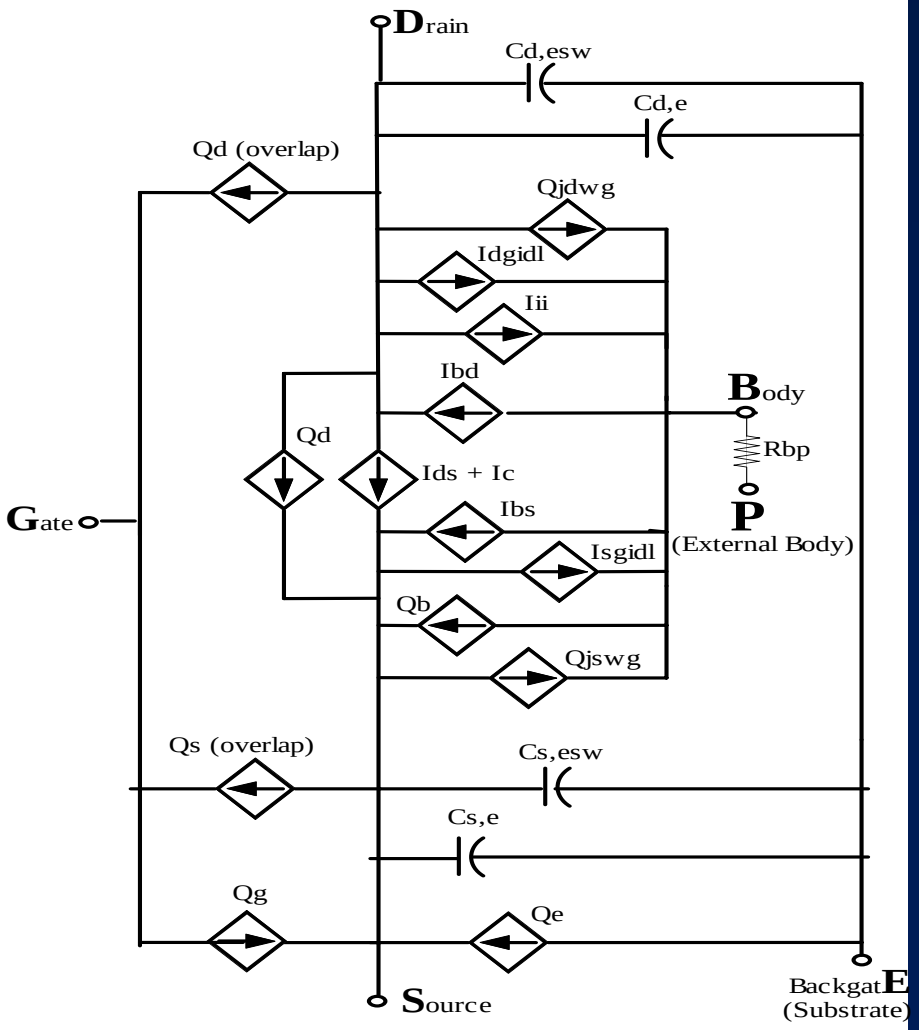
HDLs are for modeling, CMIF is for interchange!

BSIMSOI Model Topology

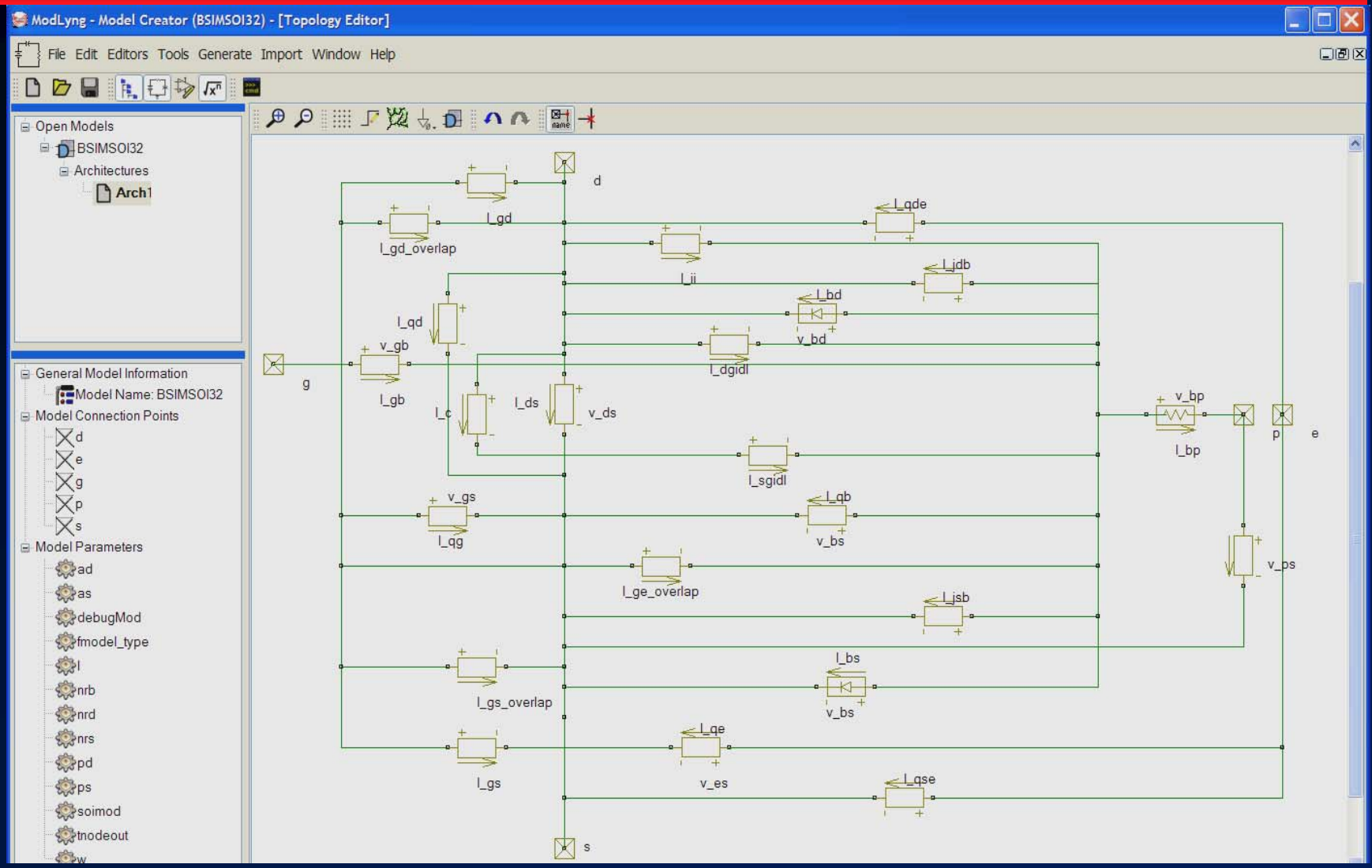
Large Signal topology of BSIMSOI, v 3.2 MOSFET model

- 5 external connection points
- 1 internal body node

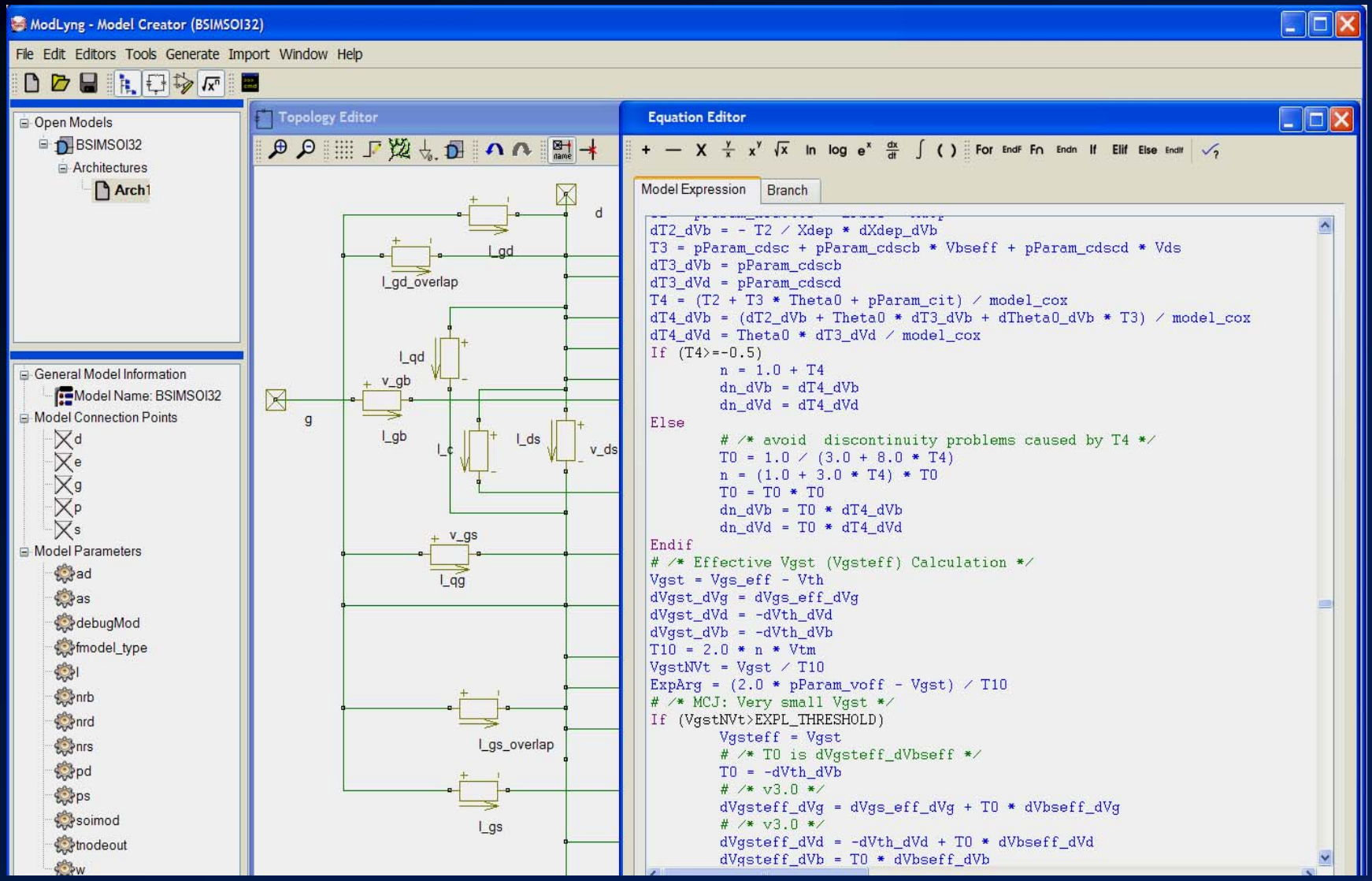
Using Paragon to create this device model begins with capturing this picture of the effects within the device



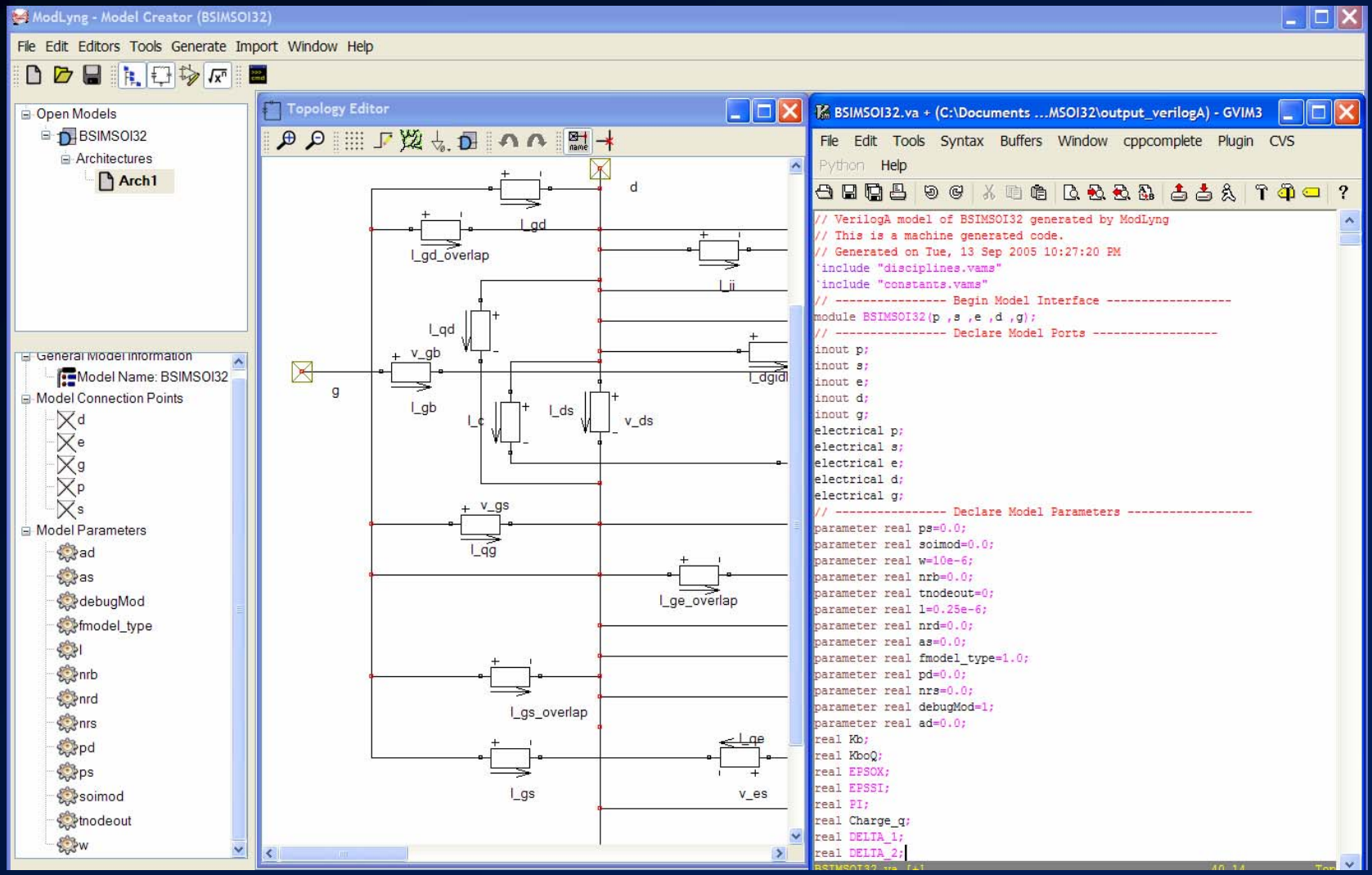
BSIMSOI Model (in Paragon)



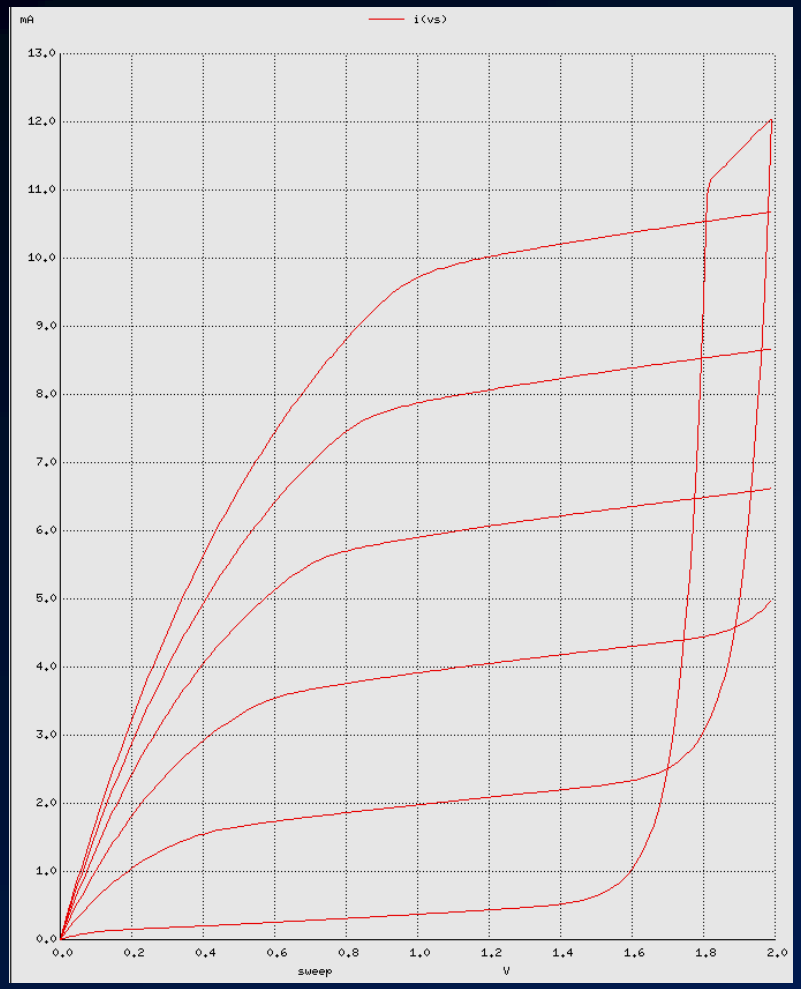
BSIMSOI Model (in Paragon)

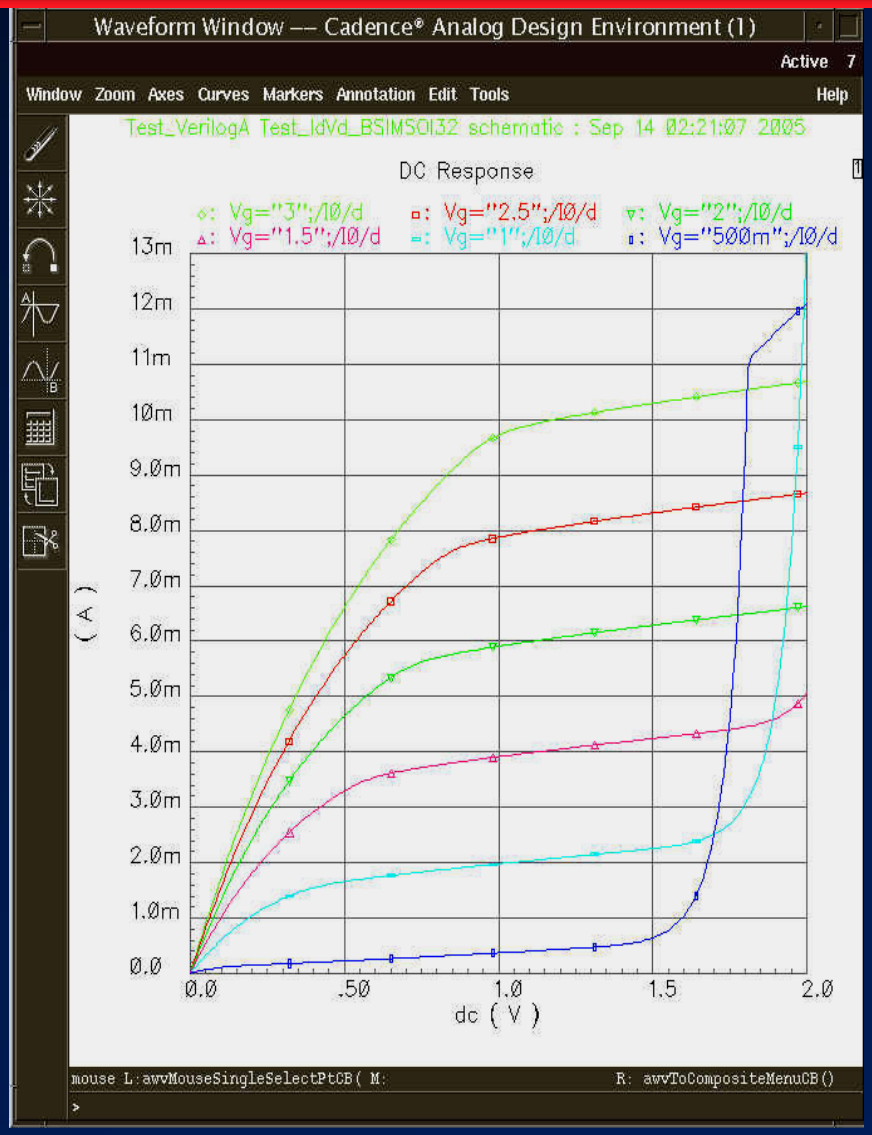


BSIMSOI Model (in Paragon)

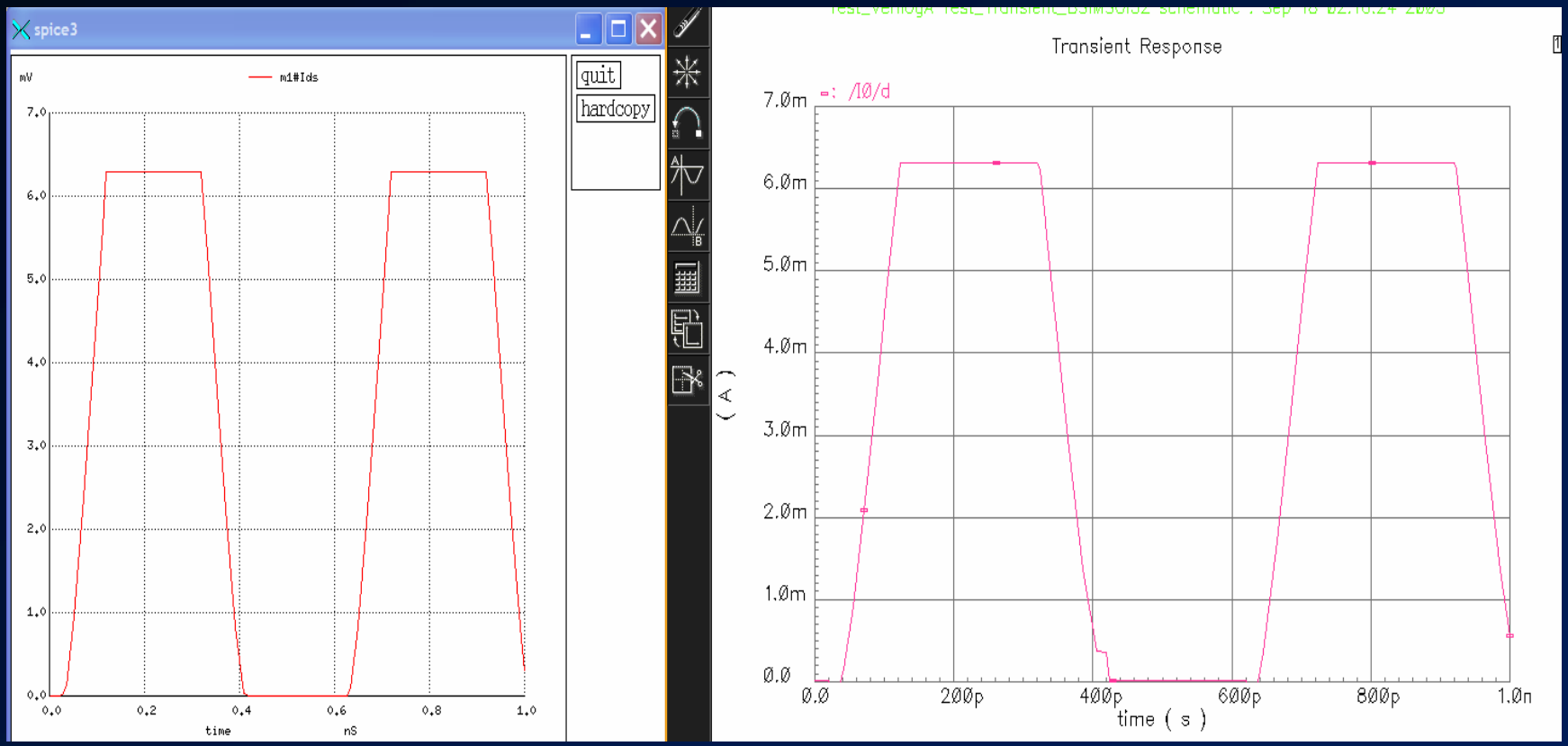


SPICE and Spectre results (DC)

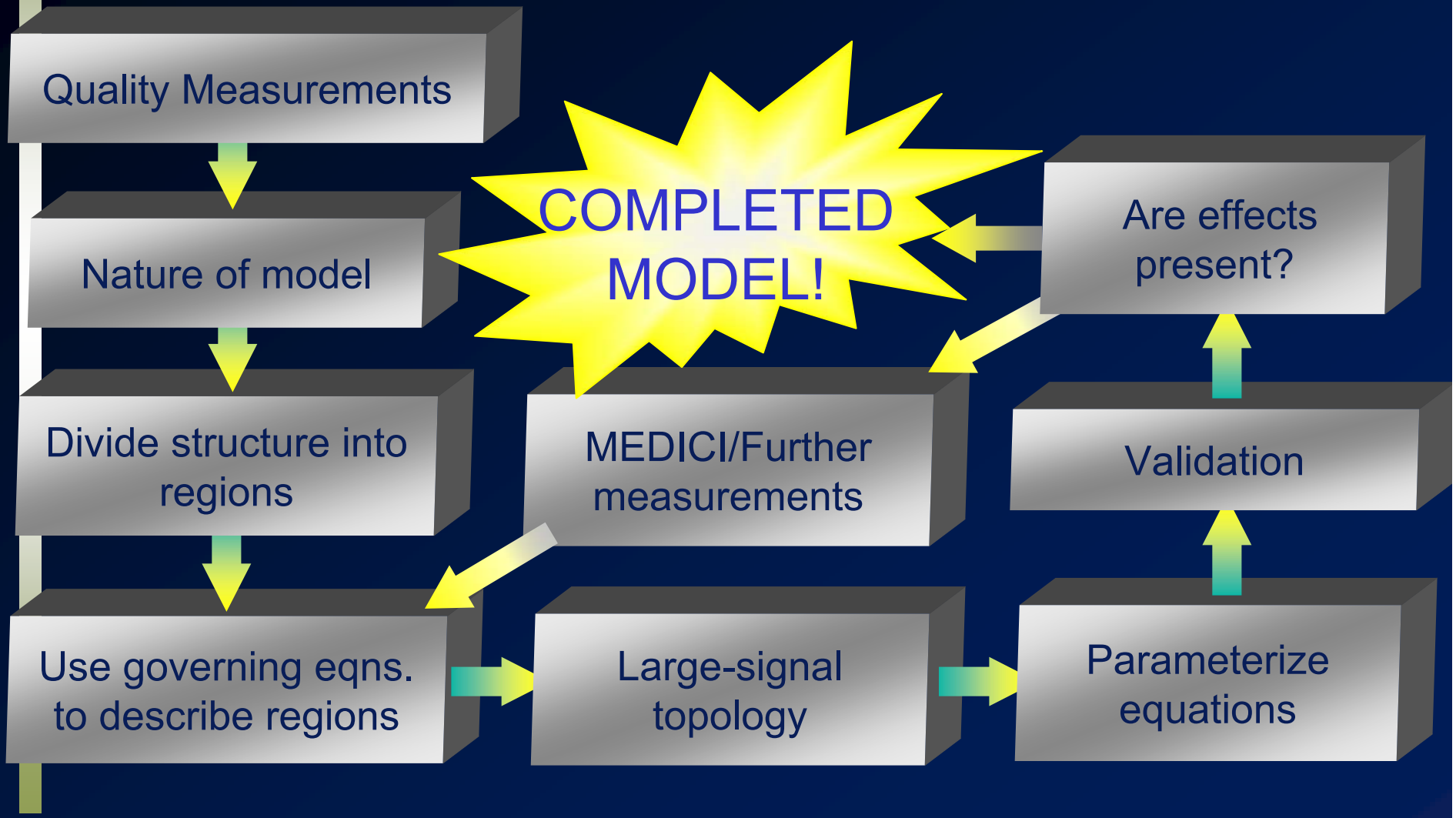




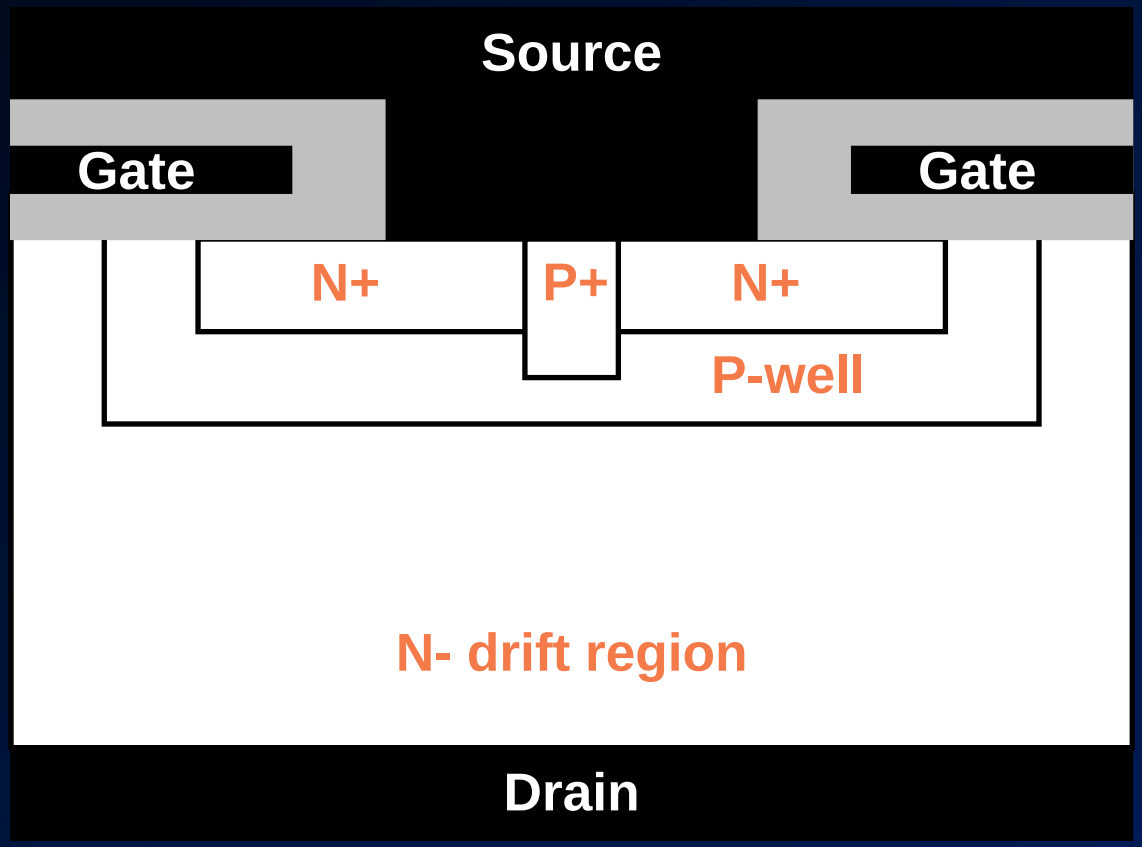
SPICE and Spectre results (Transient)



Device Modeling Methodology

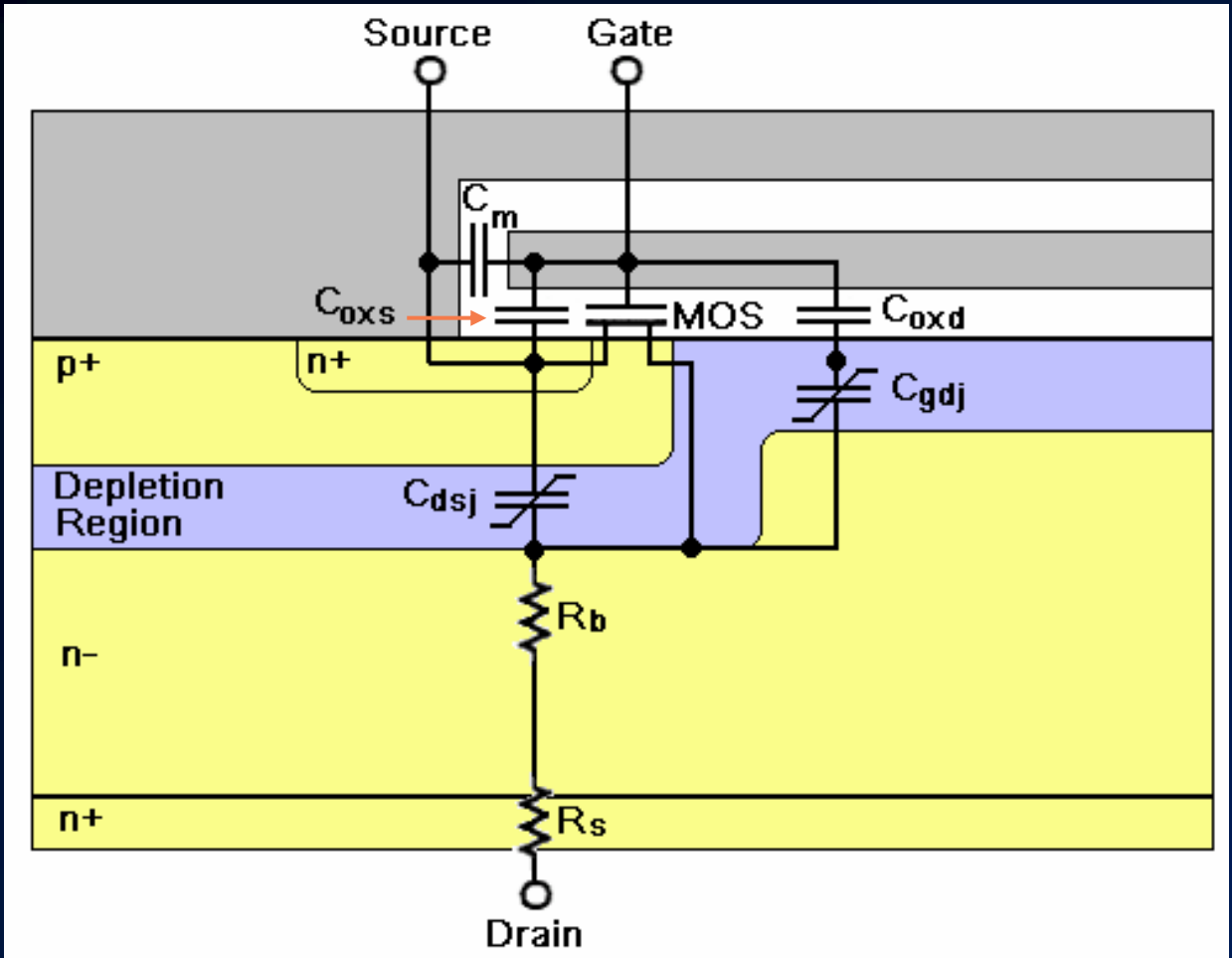


SiC DiMOSFET



- Similar to Si VDMOSFET structure

Model Formulation



- Large-signal model topology

Modeling Tools