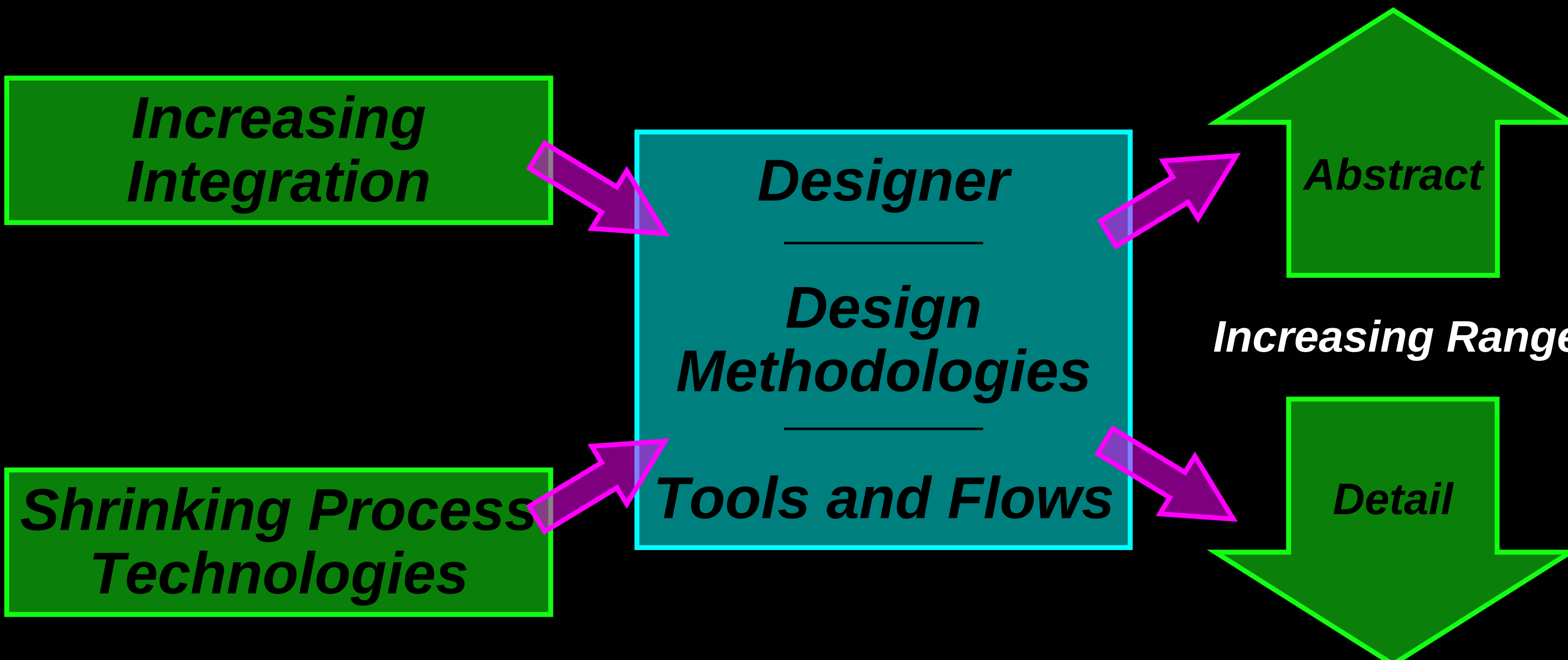
Raising Modeling to a New Low

Ted Vucurevich
CTO – Advanced Research and Development

Important Changes are Occurring



Importance of Modeling

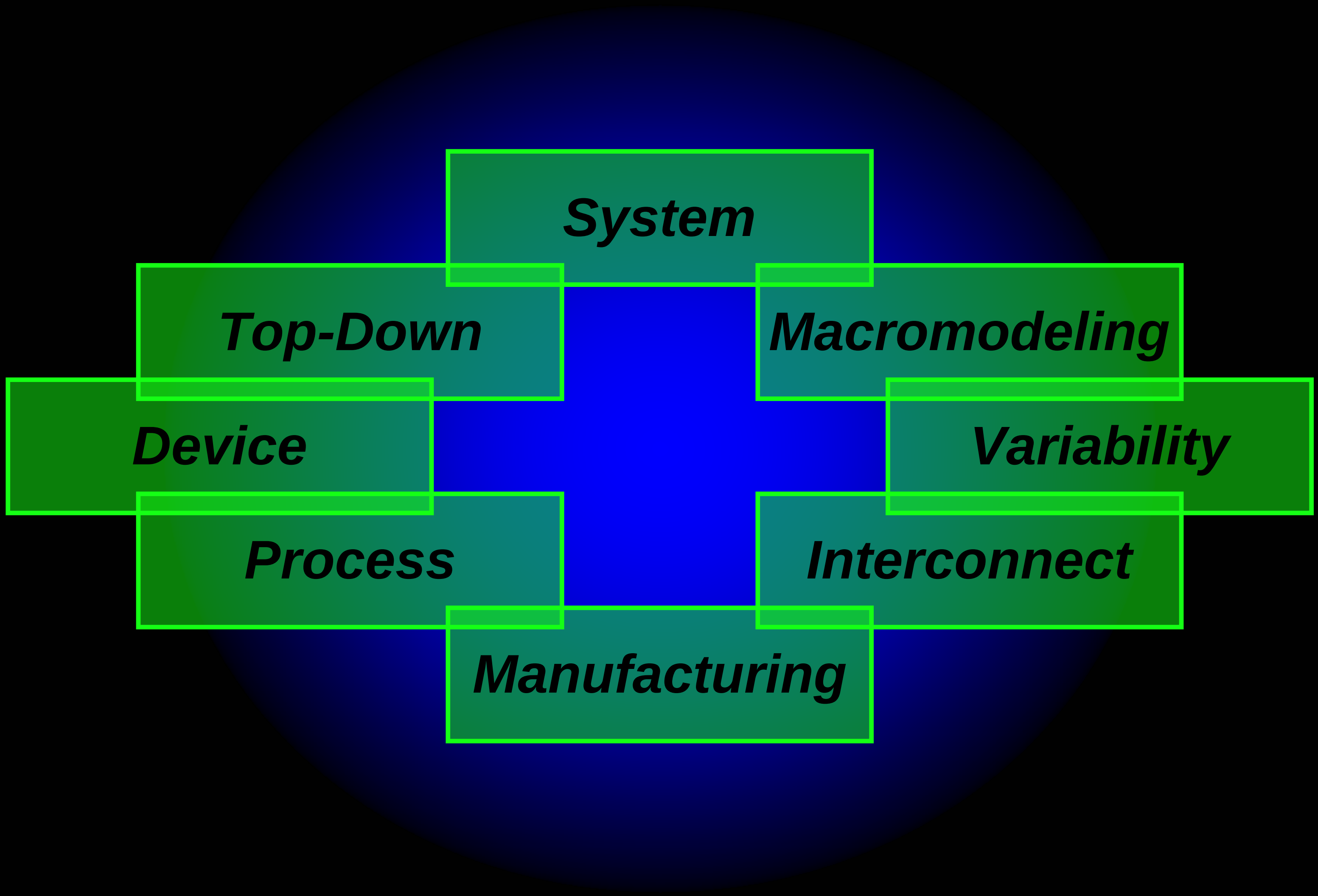
- Modeling fights complexity
  - Encapsulate what is essential, discard the rest

Complexity

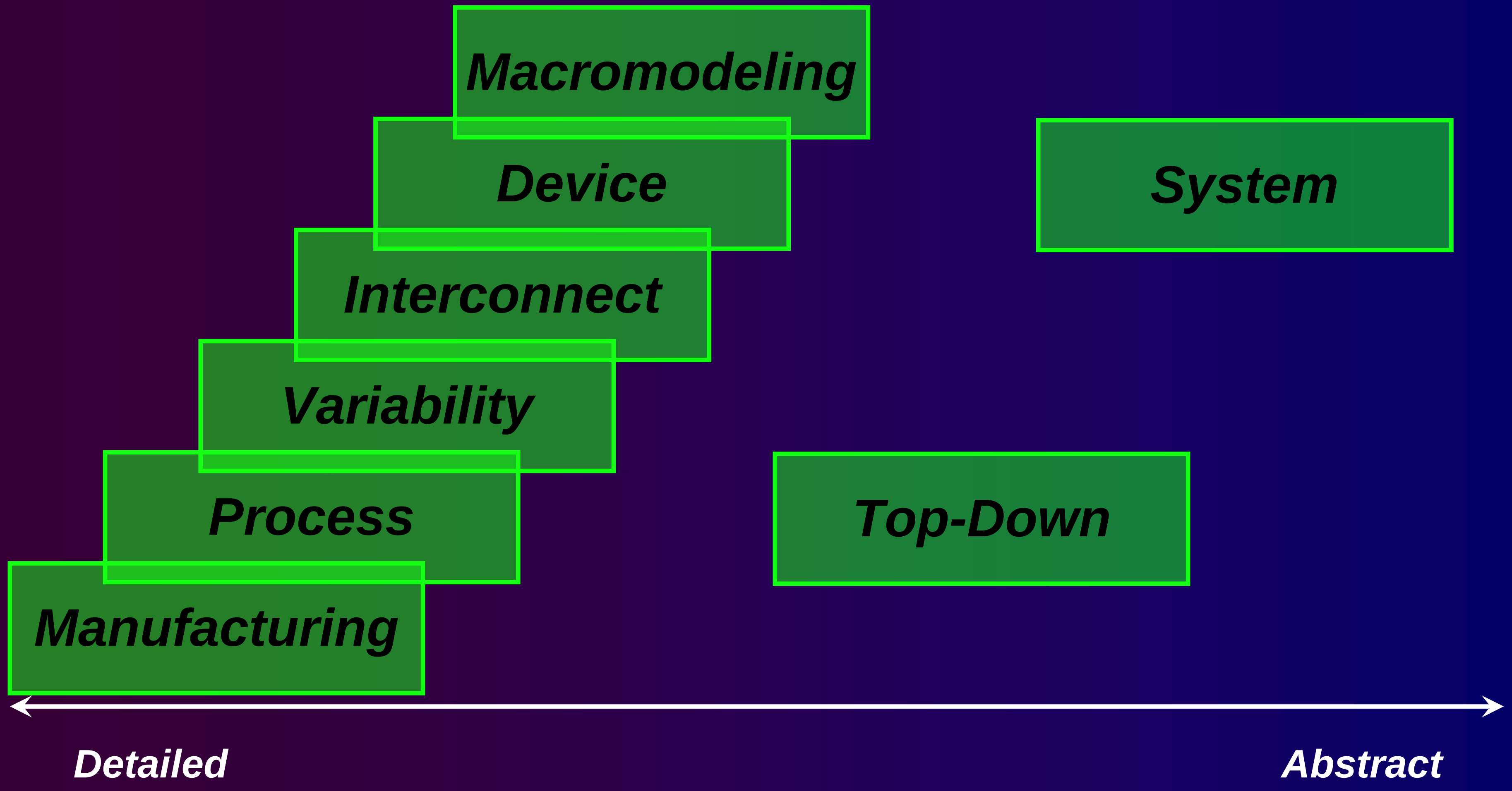
Modeling

cadence

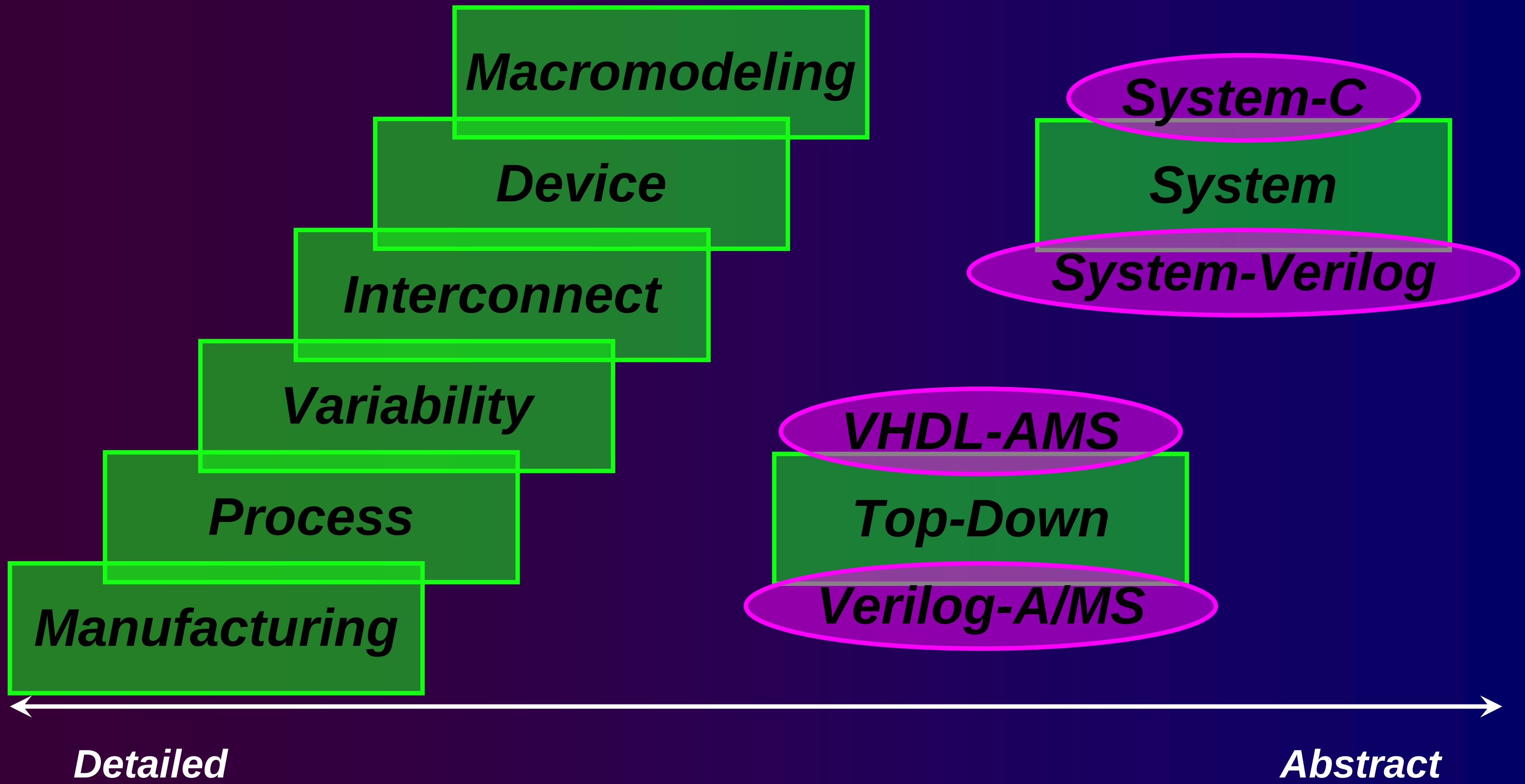
In Need of Better Modeling



Most Modeling Tasks are Low Level …



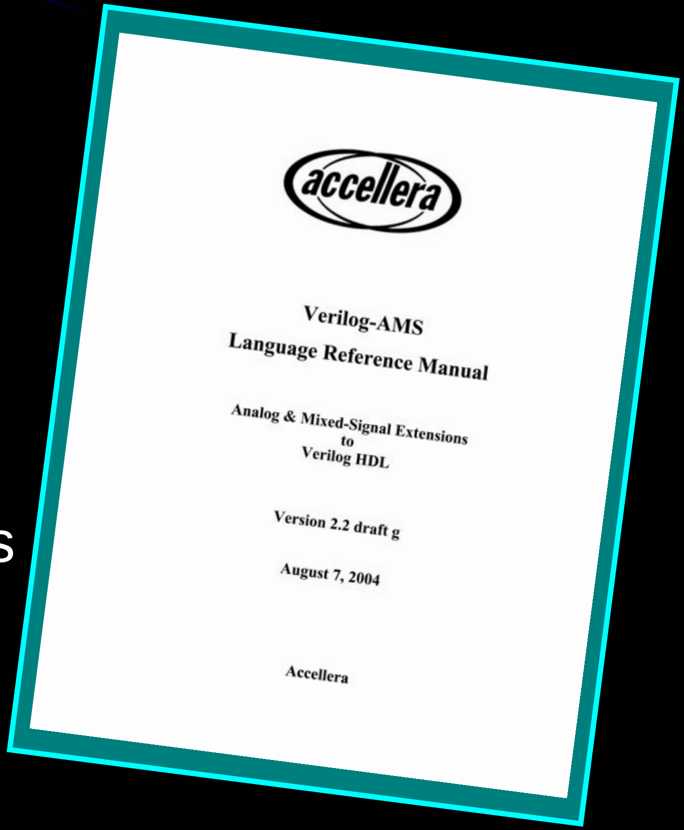
Most Modeling Tasks are Low Level …
Too Low for Existing Languages



Need a New Modeling Substrate that Supports Low Level Modeling

The Compact Modeling Extensions to Verilog-A/MS

- Allows low level models to be defined in a standardized modeling language
  - Device modeling
  - Macromodeling
  - Interconnect modeling
- Paramsets standardize SPICE .model files
  - Vendor independent model files
  - Directly supports modeling of variation


accellera

Verilog-AMS
Language Reference Manual

Analog & Mixed-Signal Extensions
to
Verilog HDL

Version 2.2 draft g

August 7, 2004

Accellera

Paramset Example

- Paramset creates alternate view with a different parameter set
- May be many paramsets for each model
- Plays role of *parameterized* .model card

```
module diode (a, c);
  parameter real is=10f from (0:inf);
  parameter real tf=0 from [0:inf);
  parameter real cjo=0 from [0:inf);
  parameter real phi=0.7 exclude 0;
  inout a, c;
  electrical a, c;
  branch (a, c) res, cap;
  real qd;

  analog begin
    I(res) <+ is∗(limexp(V(res)/$vt) – 1);
    qd = tf∗I(res) – 2∗cjo∗phi∗sqrt(1 – V(cap)/phi);
    I(cap) <+ ddt(qd);
  end
endmodule
```

```
paramset jdiode diode;
  parameter real l=0.25u;
  parameter real w=0.25u;
  .is = 16u ∗ l ∗ w;
  .tf = 15p;
  .cjo = 1.6 ∗ l ∗ w;
  .phi = 0.65;
endparamset
```

```
jdiode #(.l(200n), .w(1u)) D1 (.a(n1), .c(n2));
```

Dynamic Instance Parameters

- Today constrained to l, w, as, ad, ps, pd, etc.
- Use more parameters to model variation
  - Δxl, Δxw, ΔV_T, etc. to model mismatch
- Use fewer parameters to improve performance
  - The fewer parameters, the better the cache performance
  - Eliminate *as*, *ad*, *ps*, *pd* parameters when simulating from schematics
    - Instead, give as functions of *l*, *w*

Benefits of Compact Model Extensions

Quality

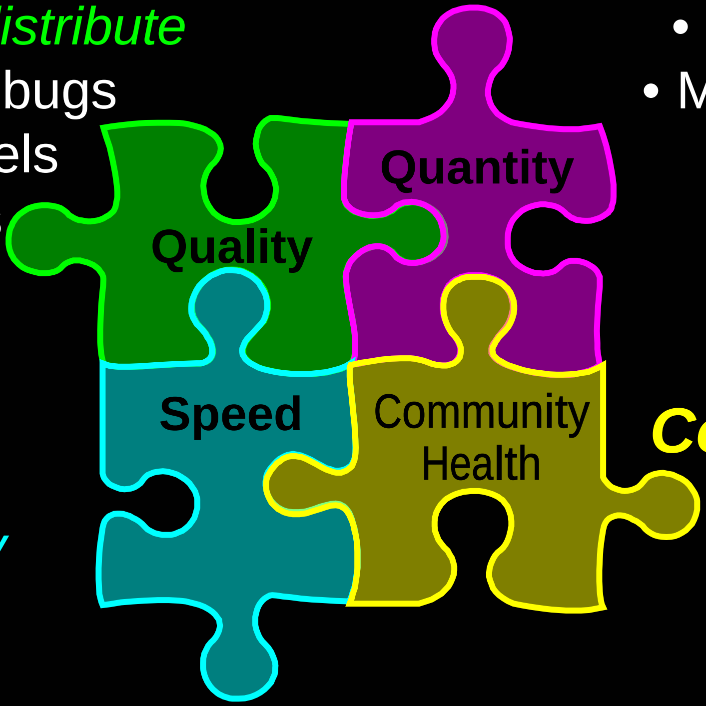
Models are smaller & easy to implement, refine, distribute

- Less code → fewer bugs
- More accurate models
- More robust models
- Better compatibility

Quantity

Inexpensive to add models

- More types of models
- More variety of models
- Tailored models



Speed

Models developed & distributed quickly (weeks vs. years)

- Updated models
- Fewer kludges

Automated code optimization

- Models run faster

Community Health

Fewer middle men between model users and developers

- Closer ties
- More modeling researchers
- More variety in research
- More specialty models

Why Compiled Verilog-A Models will be Faster than Hand-Coded Models

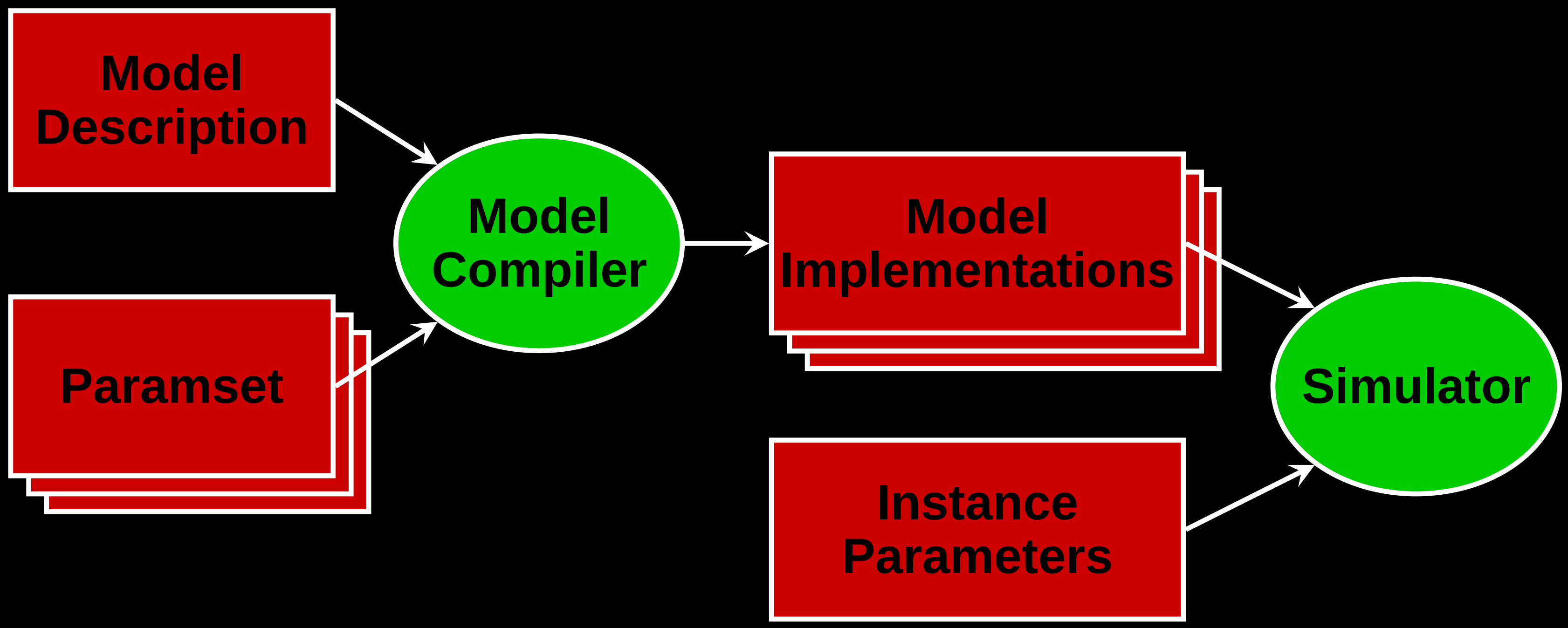
Tailored Compilation

- For multiprocessors
  - Separate out independent threads (like resistive & capacitive parts)
- For specialty simulators
  - Simplify or partition model

Exploit Difficult Trade-offs That Occur When Hand Coding Model

- Must trade-off efficiency against complexity, implementation time
- With compiler, implementations are cheap → multiple versions
  - Versions for DC & tran,
  - Versions for Newton and Samanski, etc.
- Perform detailed dependency analysis
  - Only evaluate code when necessary

Paramset and Model Compilation



- Paramset parameters become instance parameter
- All other model parameters are given fixed values and compiled out
- Instance parameters can be chosen dynamically!

Impact of the Compact Model Extensions

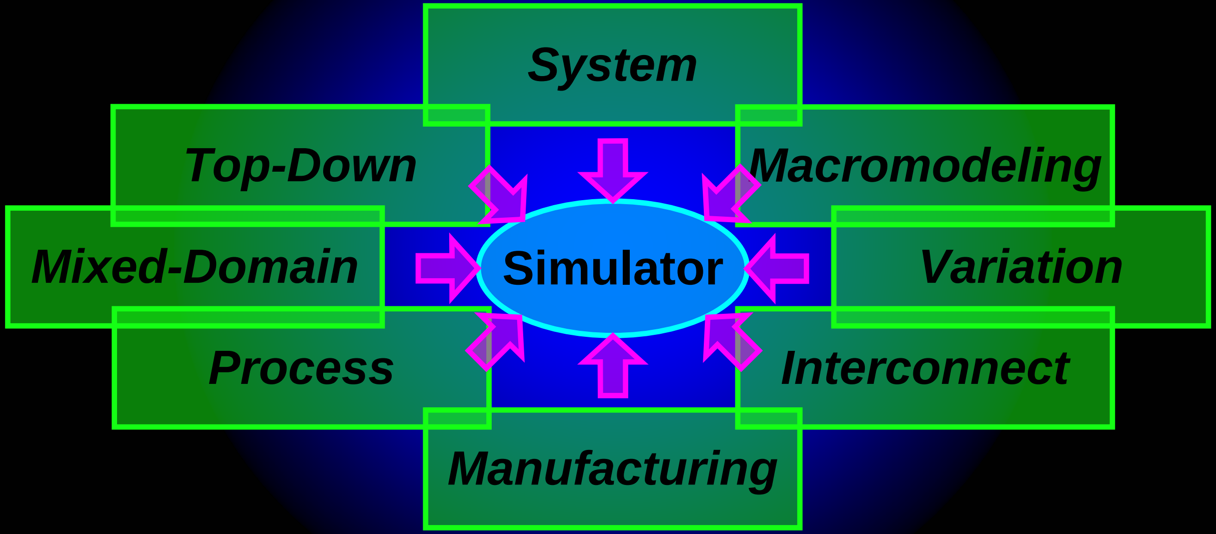
Forces the development of highly efficient model compilers.

- Verilog-A compact models must be as fast or faster than hand-coded models to succeed
  - Advantages of Verilog-A compact models are too compelling to allow them to fail
- Vendors will compete to provide best compiler
  - As they do today in Verilog & VHDL

Impact of the Compact Model Extensions

Forces the development of highly efficient model compilers.

- Allows addition of a wide variety of other models



- These models may not themselves be important enough to drive development of a model compiler

Impact of the Compact Model Extensions

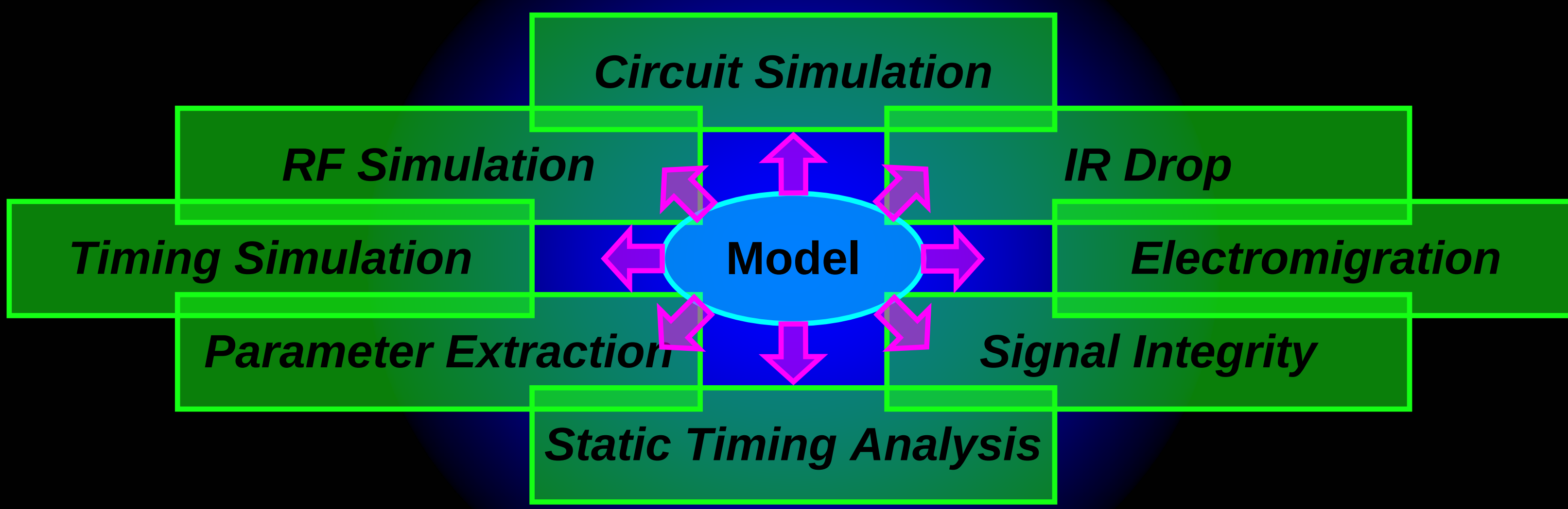
Frees the models from the simulator.

- Language is standardized, so models are portable
- Can move to any simulator from any vendor

Impact of the Compact Model Extensions

Frees the models from the simulator.

- One model for all transistor analysis



Impact of the Compact Model Extensions

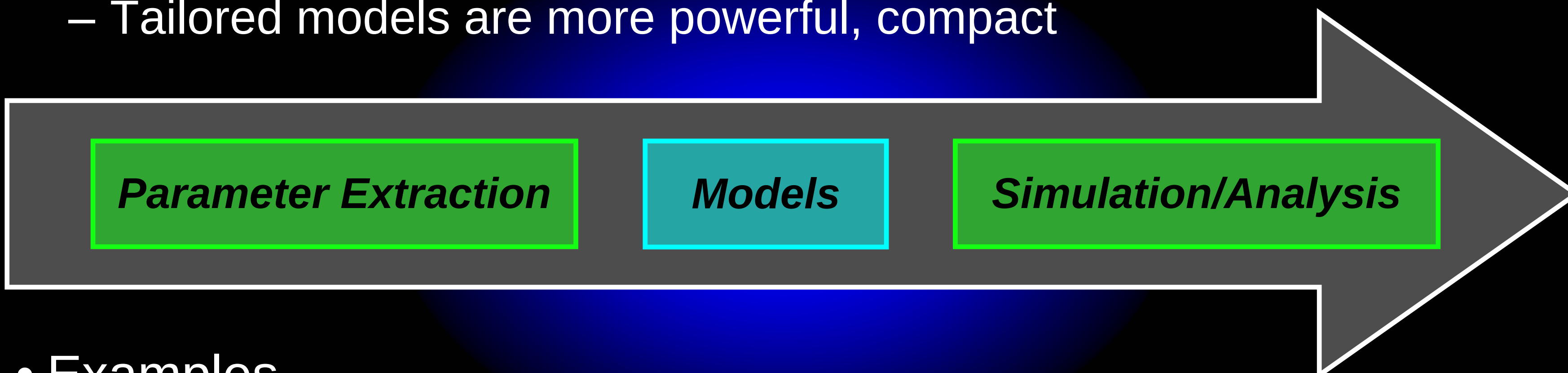
Brings models into the tool flow.

- A standardized language makes it easy for tools to generate model descriptions and parameters

Impact of the Compact Model Extensions

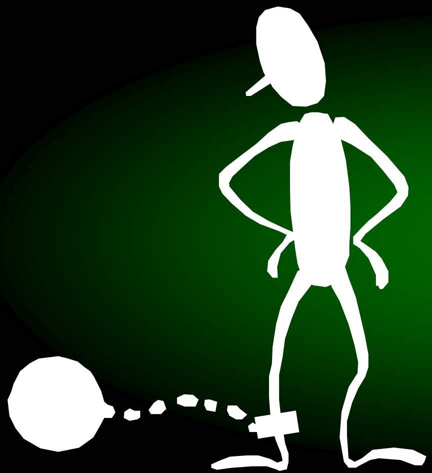
Brings models into the tool flow.

- Extractors could produce actual models
  - Not just parameters
  - Tailored models are more powerful, compact



- Examples
  - Interconnect extraction, variability extraction, macromodeling

Important Paradigm Shift



Captive Models

For the past 30 years …

Models have been built-in to the simulator and largely 'untouchable'

Free Models

Soon, models will be easily added or modified

It will take some time before the importance of this change to be recognized and fully exploited

Mismatch Example

- Step 1: model die-to-die and across die variation

```
statistics rhostats;
   process begin
      rho00=gauss(.mean(100), .std(20));
      rho01=gauss(.mean(100), .std(20));
      rho10=gauss(.mean(100), .std(20));
      rho11=gauss(.mean(100), .std(20));
   end
   mismatch begin
      drho=gauss(.mean(0), .std(5));
   end
endstatistics
```

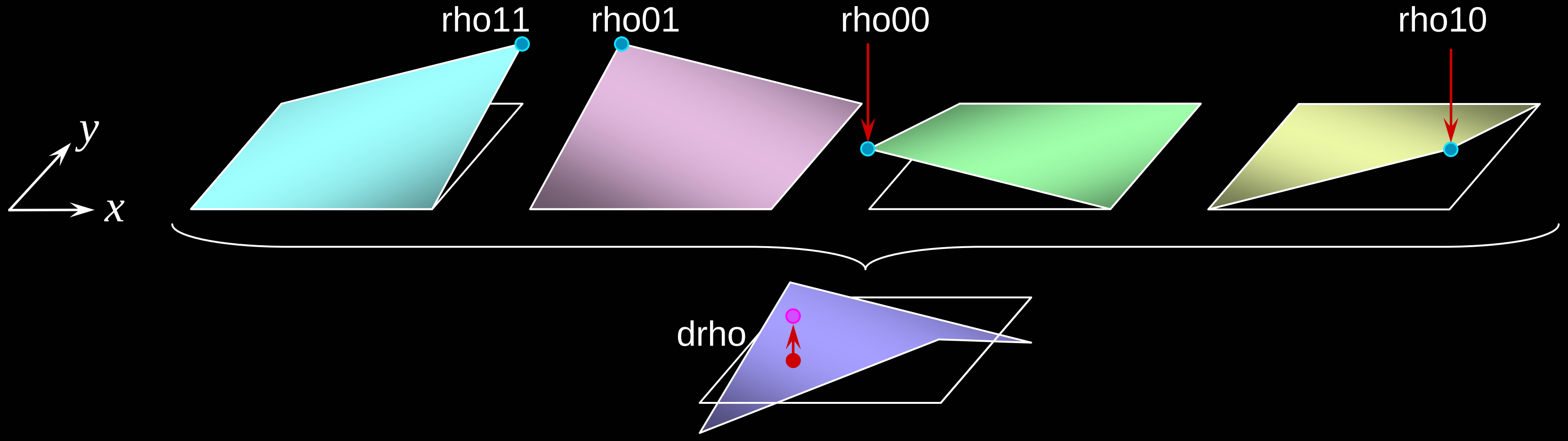
Die-to-Die Variation

Across Die Variation

Mismatch Example

- Step 2: model variation across die vs. *xy*

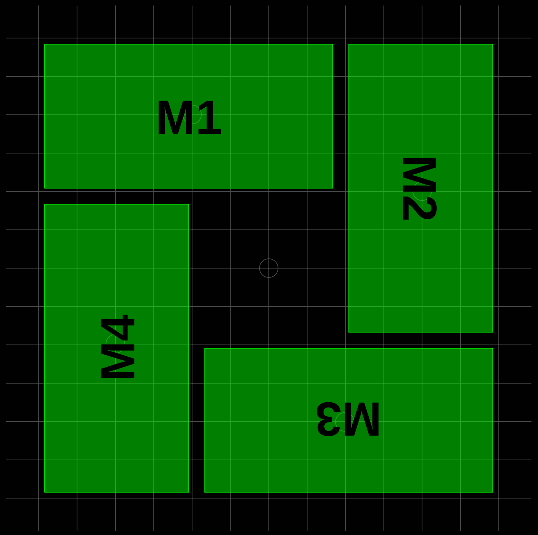
```
paramset n180nm bsim6;
    ...
    rho = rhostats.drho/sqrt($m∗$n) + (
            (1 – $x)∗(1 – $y)∗rhostats.rho00 + $x∗(1 – $y)∗rhostats.rho10 +
            (1 – $x)∗$y∗rhostats.rho01 + $x∗$y∗rhostats.rho11
      );
    ...
endparamset
```



Mismatch Example

- Step 3: extractor passes location and orientation when instantiating

```
n180nm #( ..., .$x(–2), .$y(4), .$angle(0)) M1 ( ... )
n180nm #( ..., .$x(4), .$y(2), .$angle(-90)) M2 ( ... )
n180nm #( ..., .$x(2), .$y(–4), .$angle(180)) M3 ( ... )
n180nm #( ..., .$x(–4), .$y(–2), .$angle(90)) M4 ( ... )
```



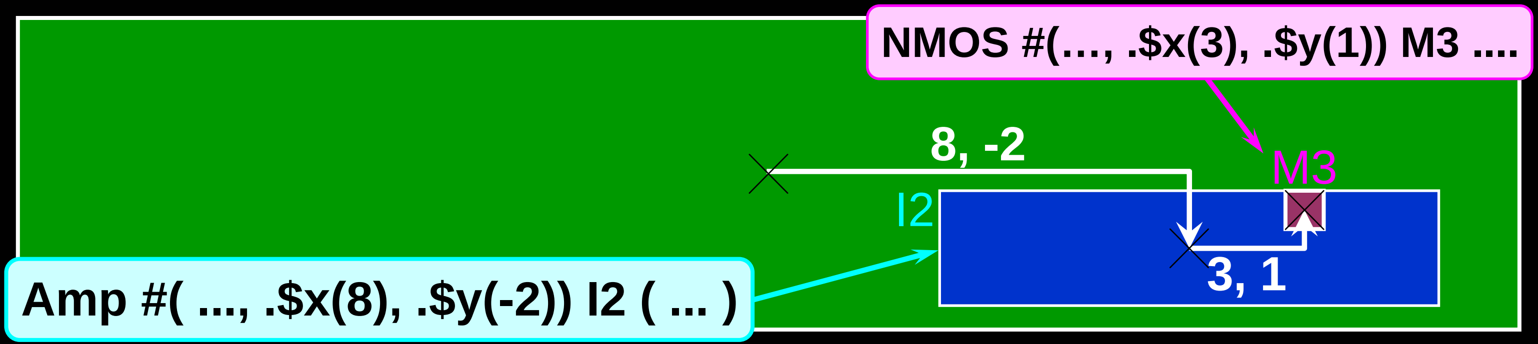
Mismatch Example

Things to notice

1. The intrinsic model does not include effects of variation
2. Effect of variation is modeled in paramset
   - In this way, variation can be included in any model
   - Fits well with established approach
3. Mismatch is modeled as a function of relative position & size
   - No need for specifying pairwise mismatch
4. Position is specified hierarchically

Hierarchical Implicit Parameters

- $x, $y, $angle are a new type of parameter
  - – Modeled after multiplicity or ‘M’-factor parameter of Spice
- They are implicitly declared, and so supported by all components.
- Their value is given relative to the next higher level of hierarchy, and they accumulate through hierarchy



- In this case, they allow the placement and orientation of components to be known
- Allows models that take into account physical placement
  - – Such as mismatch

Effect of the Neighborhood

- The last example showed how model could be dependent on physical placement

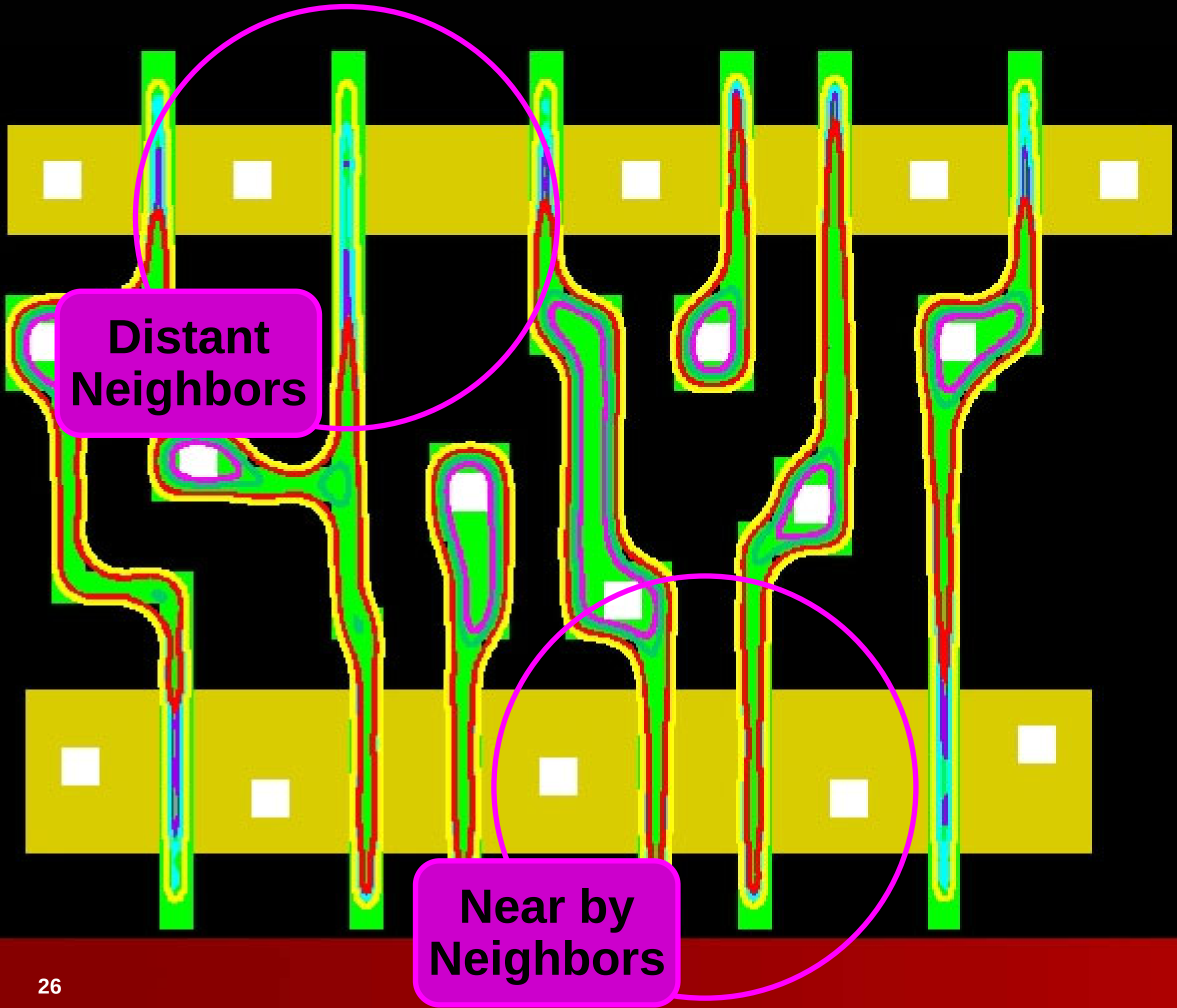
This implies that each device in a simulation could be unique, which could have important implications for the simulator, especially in these days of 'hierarchical' simulation.

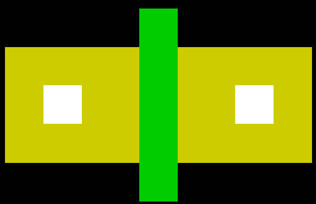
- But the model is independent of what is next to it

That is not sufficient for sub-wavelength features

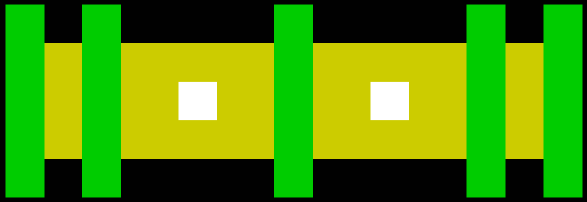
Proximity Effects

- Identically drawn devices can be imaged quite differently due to effect of neighbors





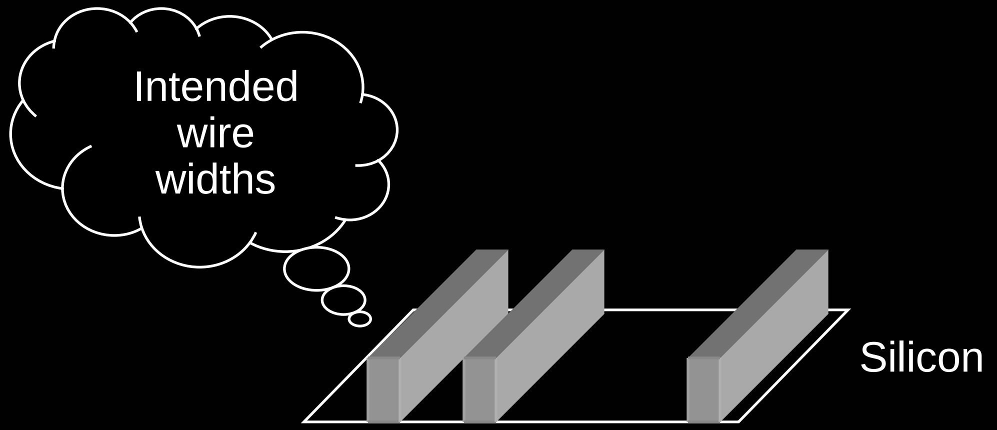
What we use to model



What we must model now

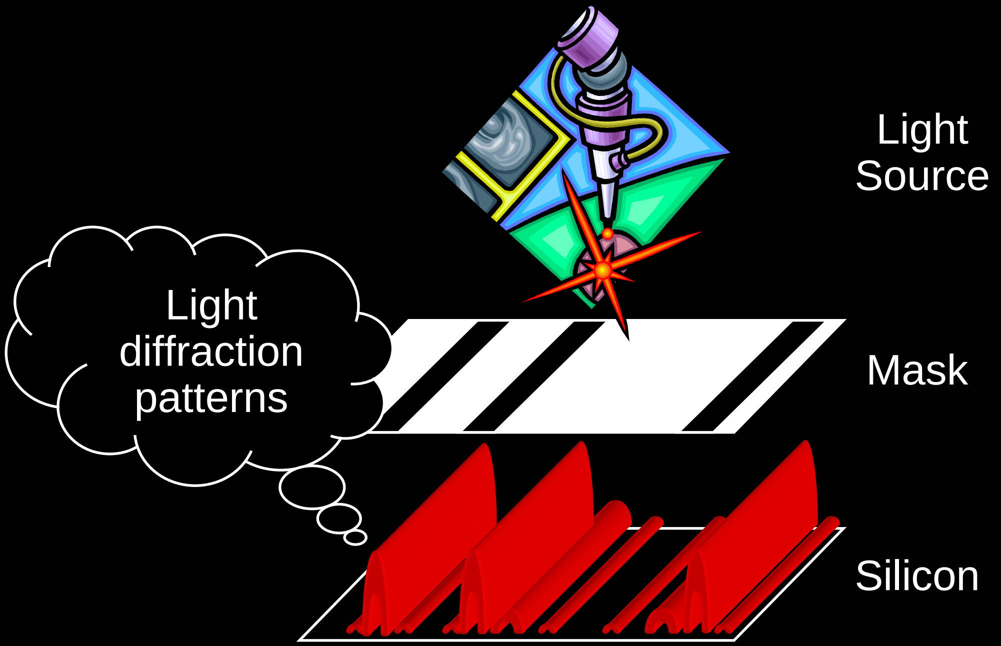
Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement



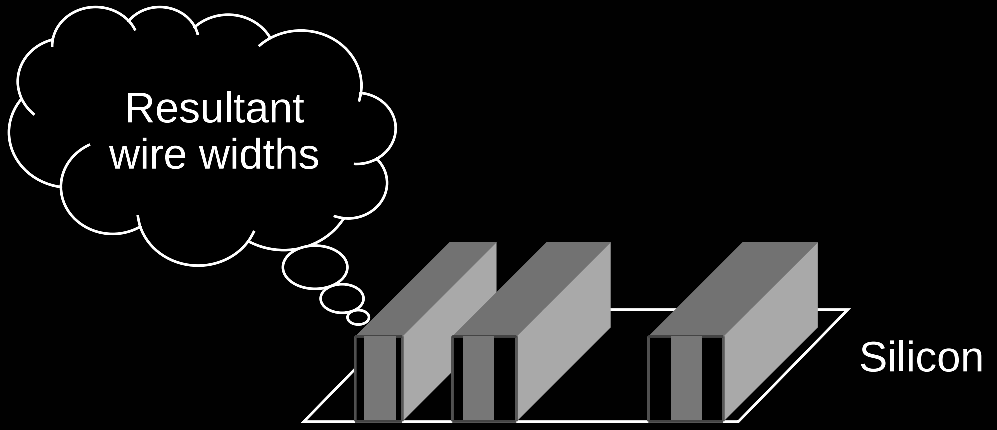
Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement



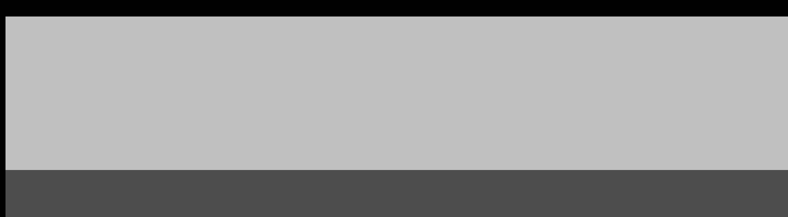
Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement



Modeling Proximity Effects

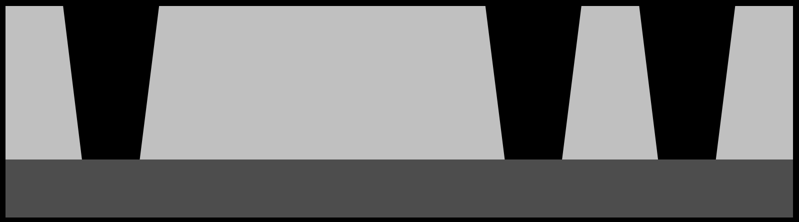
- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions



Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions

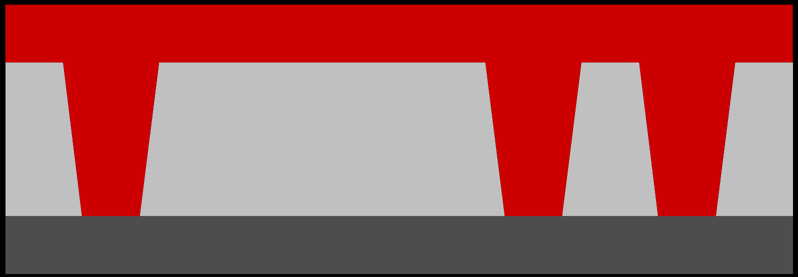
Cut Trenches for Interconnect



Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions

Add Layer of Copper

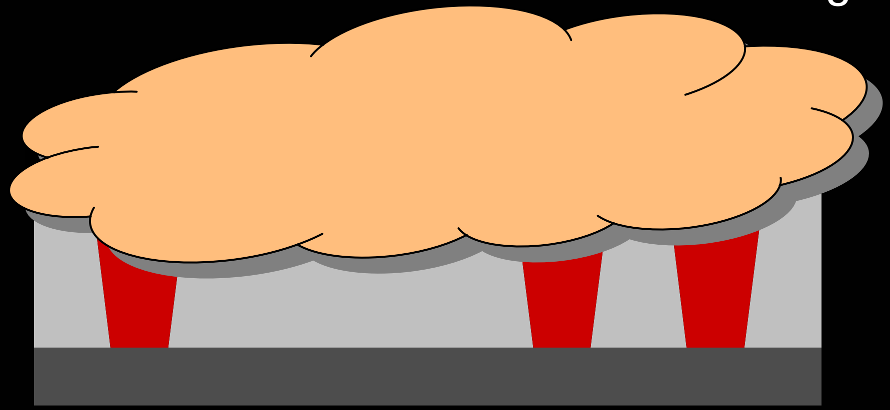


cadence

Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions

Perform Chemical Mechanical Polishing



Variation due to CMP

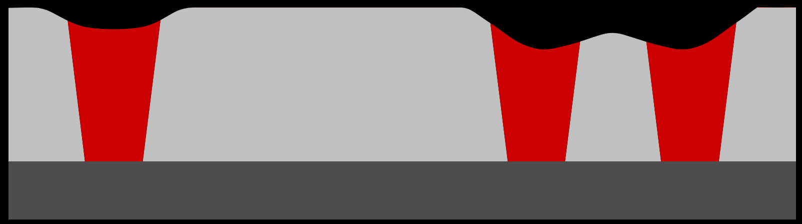
ΔR: +40% ~ –20%

ΔC: –30% ~ +10%

Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions

Thickness of wires depends on density
of wires in the neighborhood



Variation due to CMP

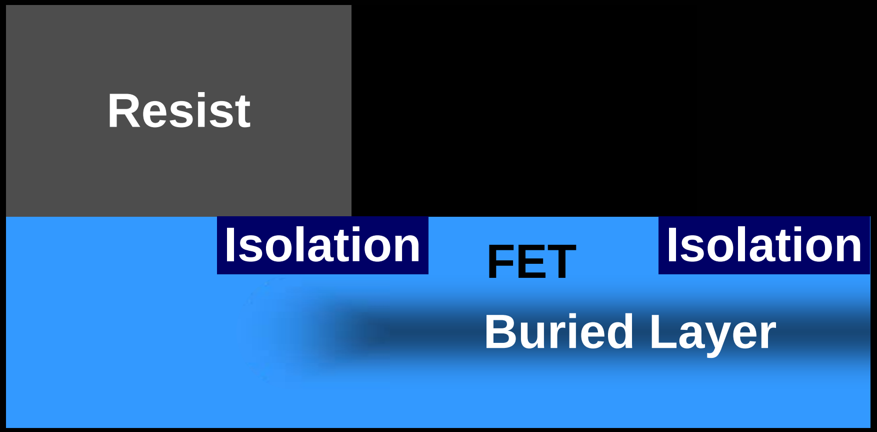
ΔR: +40% ~ –20%

ΔC: –30% ~ +10%

cadence

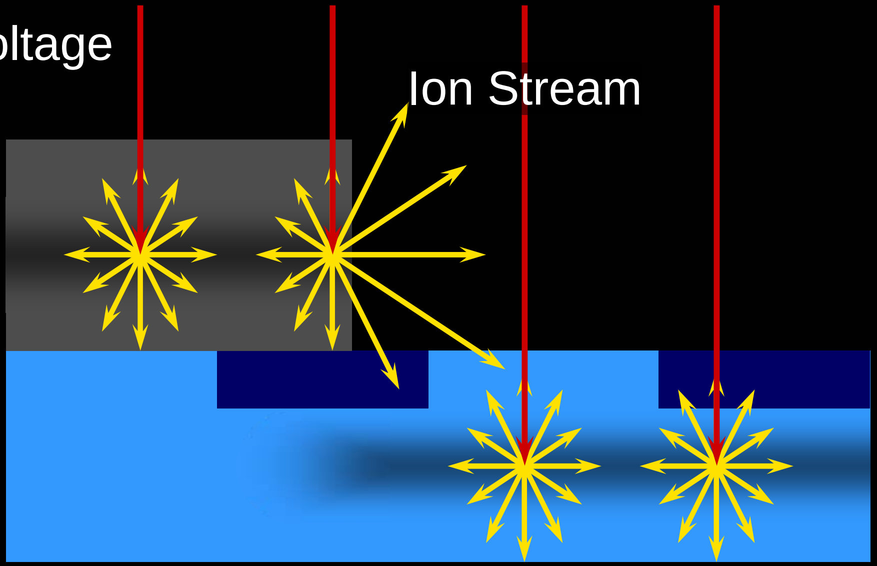
Modeling Proximity Effects

- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions
- Mask Proximity Effect on Buried Layer Implantation
  - Changes threshold voltage for devices near edge of mask



Modeling Proximity Effects

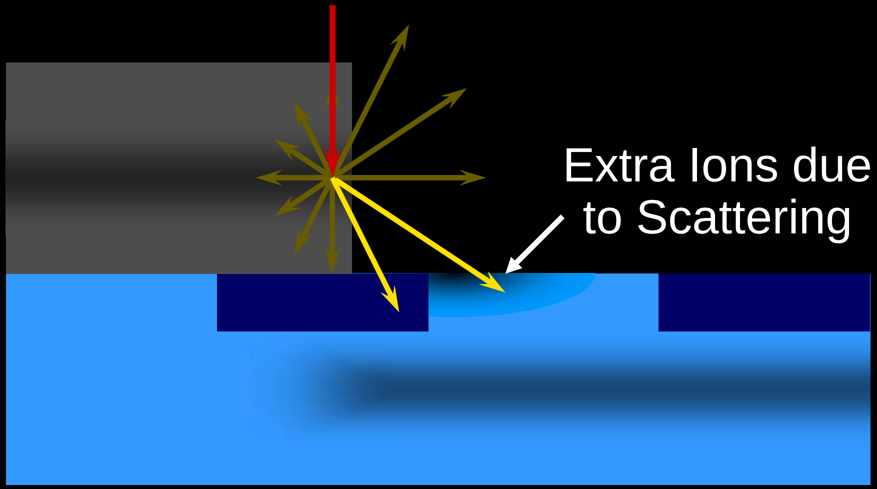
- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions
- Mask Proximity Effect on Buried Layer Implantation
  - Changes threshold voltage



cadence

Modeling Proximity Effects

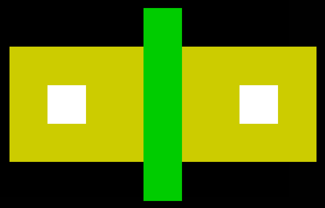
- Lithography (interference patterns)
  - Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - Changes line size, more in high density regions
- Mask Proximity Effect on Buried Layer Implantation
  - Changes threshold voltage near mask edge



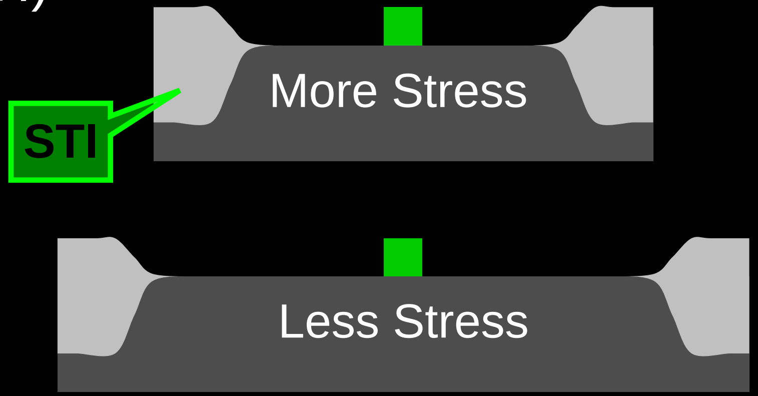
cadence

Modeling Proximity Effects

- Lithography (interference patterns)
  - – Changes line width and placement
- Metal Dishing from Chemical Mechanical Polishing (CMP)
  - – Changes line size, more in high density regions
- Mask Proximity Effect on Buried Layer Implantation
  - – Changes threshold voltage near mask edge



- Stress due to Shallow Trench Isolation (STI)
  - – Stress affects mobility
  - – The closer the STI is to the channel, the greater the effect on mobility



Modeling Proximity Effects

- All effects require more sophisticated interaction between extractor and simulator than is practical today
- These effects can all be handled with paramset parameterization
- This is not sufficient in all situations …
  - When producing macromodels, extractor produces entire model

Moving Up: Macromodeling

- A huge impediment to use of behavioral models is a lack of models

Top-Down Models

Relatively easy to develop by hand

Pull from a library and modify to fit need

Bottom-Up Models

Much tougher to develop by hand

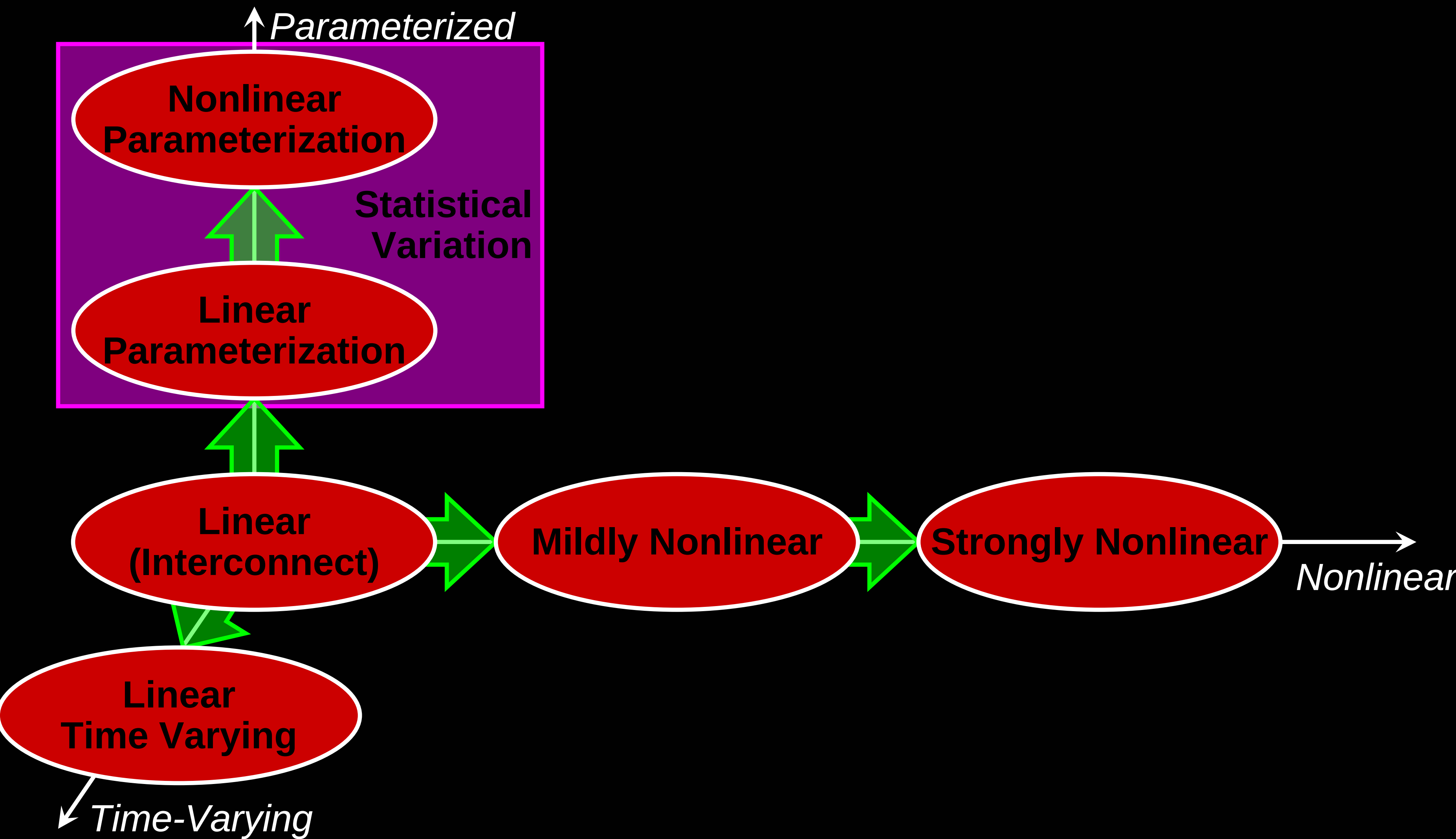
But here we have the circuit from which we can build a model

“Automatic Macromodeling”

Macromodeling: A Difficult Challenge

- A huge and extremely challenging space
- Attack one subspace at a time
  - Making good progress

Macromodeling Challanges



Macromodeling

- Continued tomorrow, with Jacob’s keynote presentation

cadence