



# Parallel Circuit Simulation at SNL and Fault Tolerance

<http://www.cs.sandia.gov/Xyce>

10 June 2002

**Rob Hoekstra**, Scott Hutchinson, Eric Keiter  
Computational Sciences Department



Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.





# Agenda



## Xyce Project Overview

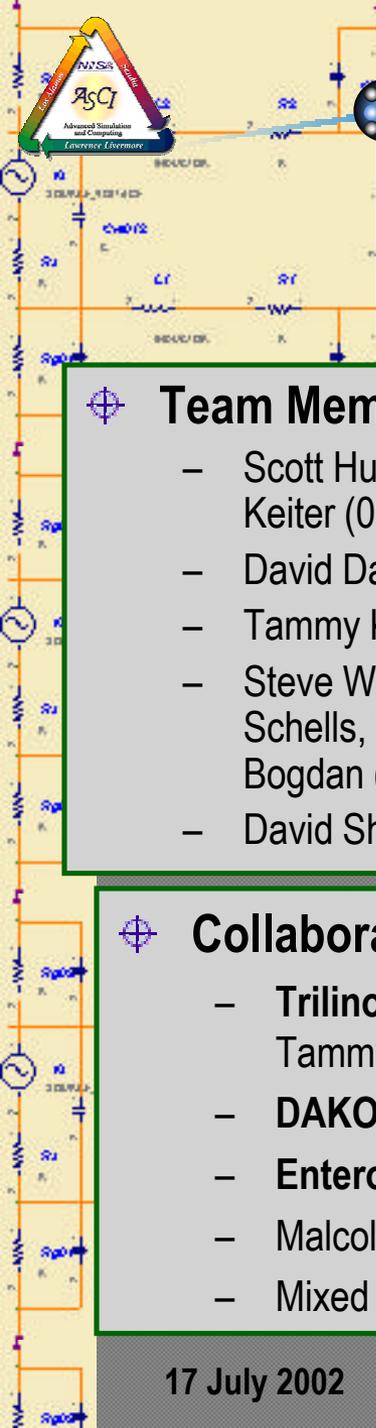
- ASCI Milestones
- Releases

## Background

- Problem Formulation
- Solvers

## Fault Tolerance

## Summary



# Multi-Center Project – 9200, 1700, 8400, 2600, 2300

**A HPEMS  
Project**

## Team Members

- Scott Hutchinson , Rob Hoekstra, Eric Keiter (09233)
- David Day (09214)
- Tammy Kolda (08950)
- Steve Wix (PI), Lon Waters, Regina Schells, Thomas Russo, Carolyn Bogdan (01734)
- David Shirley (Abba Tech.)

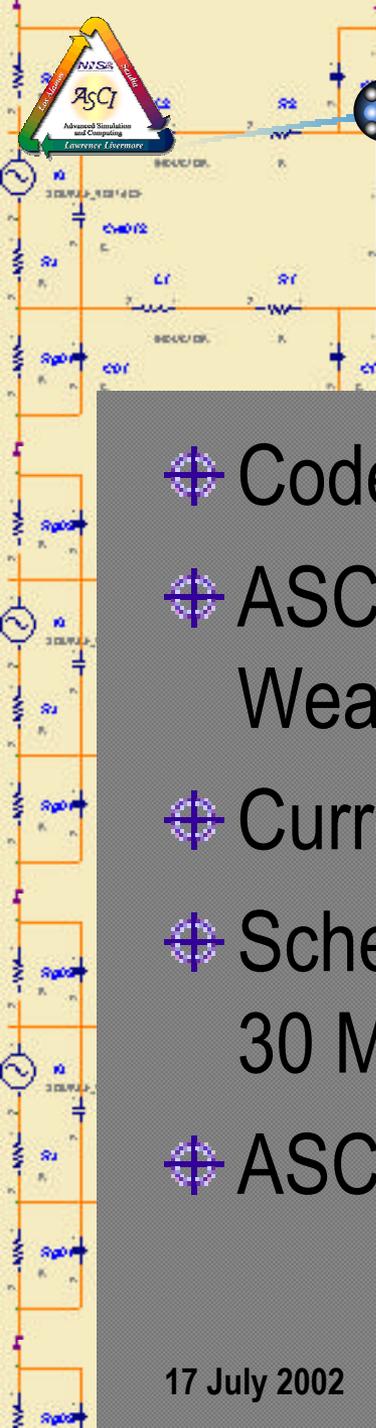


## Customers

- Bill Ballard, Ken Marx, Steve Brandon (08418)
- Marty Stevenson, Fred Anderson, Pat Smith (02612)
- George Laguna (02338) David Shirley (Abba Tech.)
- Bob Brocatto (01735), John Dye (02331), Mark De Spain (02125), John Tenney (12333)

## Collaborators

- **Trilinos / Epetra / NOX** (Mike Heroux, Roger Pawlowski, Tammy Kolda)
- **DAKOTA** (Bart van Bloemen Waanders)
- **Entero** (David Gardner, Joseph Castro, Mark Gonzales)
- Malcolm Panthaki (CoMeT Solutions)
- Mixed Signal (Phil & Dale)



# Xyce Status

- ⊗ Code-gen to First Parallel Run: 6 Months
- ⊗ ASCI Milestone Supporting Calculation on Real Weapon Sub-system: 18 Months
- ⊗ Currently in Beta 2 Release
- ⊗ Scheduled Version 1.0 Release in October, 2002: 30 Months
- ⊗ ASCI Level 1 Hostile Milestone FY03

# Beta 2 Release

- ⊕ Dry Run for Official 1.0 Release
- ⊕ GNU AutoTooled
- ⊕ Supported Platforms
  - SGI (32 & 64 bit, MPI)
  - Linux (MPICH, MPILAM)
  - FreeBSD (MPICH, MPILAM)
  - Compaq Alpha Tru64 (MPI)
  - CPlant (Ross, Ross2) (MPI)
  - Cplant (CA) PENDING
  - Windoze (Serial)
- ⊕ Documentation
- ⊕ SQE Gap-Assessment
- ⊕ Third Party Software Process
  - Trilinos
  - Xpetra
  - SuperLU & Epetra\_SLU
  - Zoltan & Zoltan\_C++
  - Chaco & Chaco\_C++
  - Expression

# Release Documentation

Xyce User's Guide, Version Beta 2

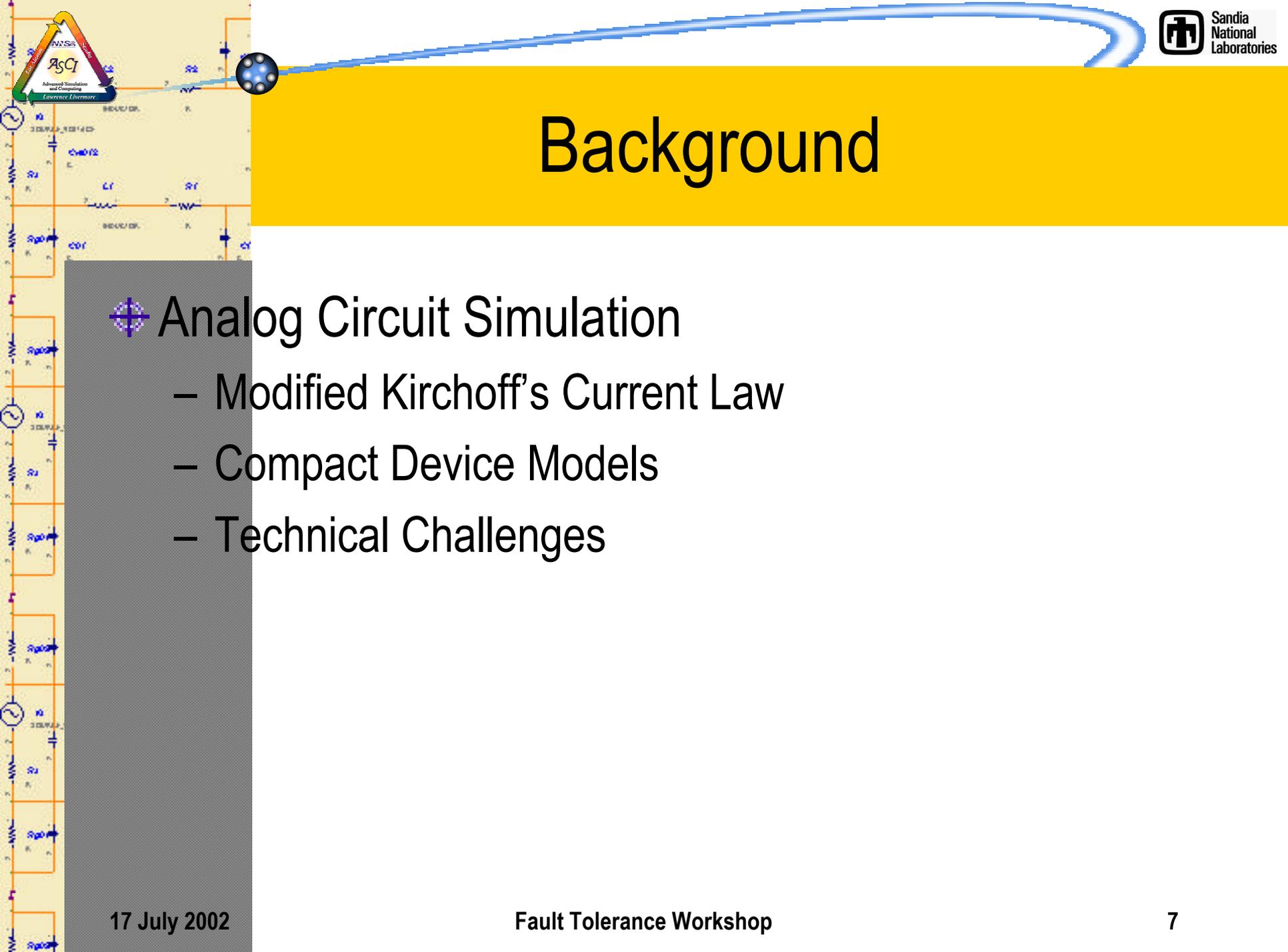
Xyce Design – Theory and Implementation, Version Beta 2

SQE Documentation

– Release and Distribution Management

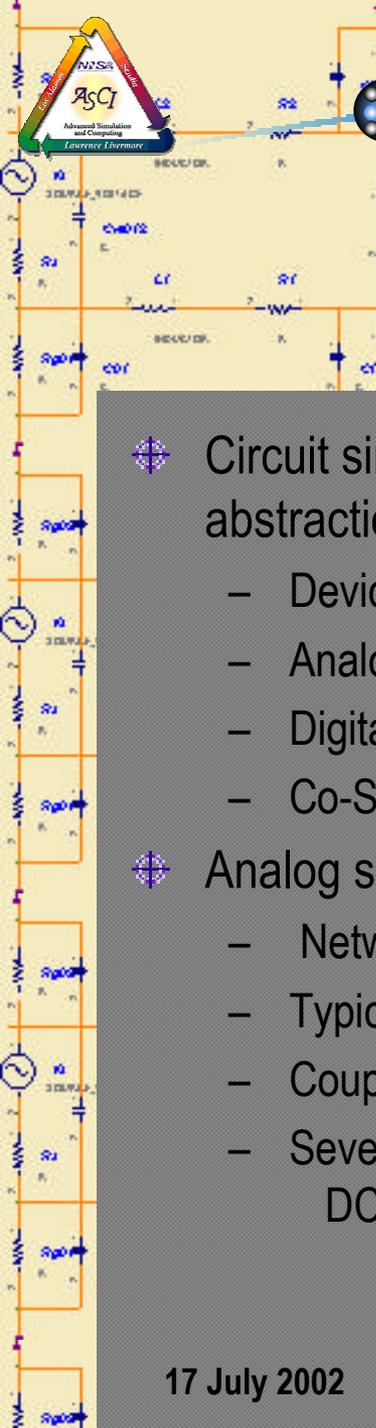
– Third Party Software Configuration Management Plan

The Xyce Parallel Electronic Simulator – an Overview, Proceedings of Parallel Computing 2001



# Background

- ⊠ Analog Circuit Simulation
  - Modified Kirchoff's Current Law
  - Compact Device Models
  - Technical Challenges



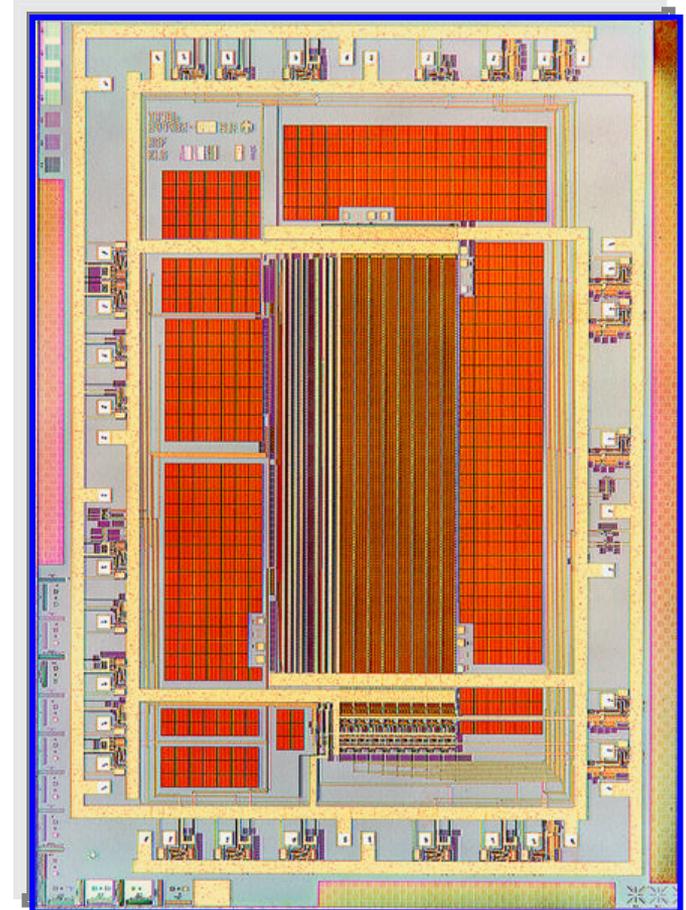
# Circuit Simulation

⊕ Circuit simulation is applied at several levels of abstraction:

- Device (PDE)
- Analog (ODE/DAE) ← 
- Digital (VHDL)
- Co-Simulation (Circuit + Software)

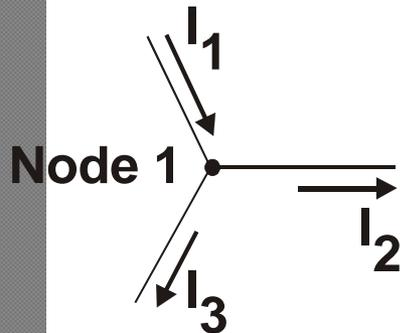
⊕ Analog simulation models:

- Network of devices
- Typically described by ODEs
- Coupled via Kirchoff's current and/or voltage laws
- Several analysis options: DC Sweep, DC Operating Point, Transient, AC Analysis



# Modified KCL Formulation

- ⊕ Xyce (and most other circuit codes) represent the problem using modified KCL (Kirchoff's current law) equations.
- ⊕ KCL : sum of all currents into a circuit node equals zero.



**KCL , Node 1:**

$$f = I_1 - I_2 - I_3 = 0$$

- ⊕ Each node: 1 KCL equation, 1 nodal voltage variable.
- ⊕ N nodes -> N KCL equations -> N node voltage variables
- ⊕ A KCL formulation becomes a modified KCL formulation when it is necessary to include some current variables.

# Device Example: Resistor

- ⚡ Resistors and Capacitors are the simplest circuit elements.
- ⚡ Most devices are a combination of resistive and capacitive elements.
- ⚡ Resistor load:

$$V = I * R \quad (V = V_1 - V_2)$$

$V_1$  = voltage at Node 1.

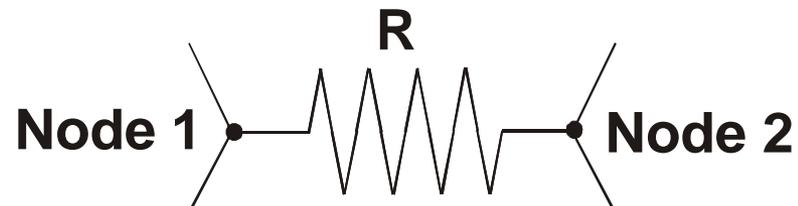
$V_2$  = voltage at Node 2.

$$I = G * V = G(V_1 - V_2)$$

where  $G = 1/R =$  conductance.

$G$  may be constant (linear)

or  $G = f(V)$  (nonlinear)

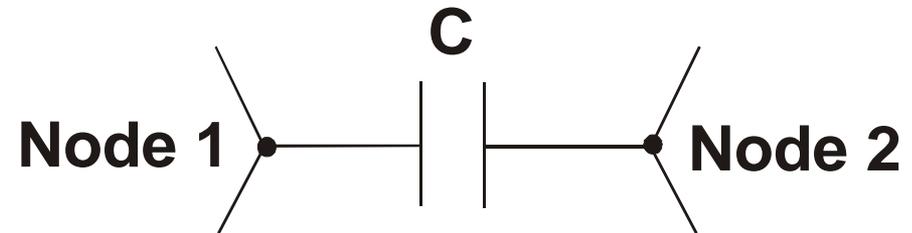


$$\mathbf{J} \quad \Delta \mathbf{X} = -\mathbf{f}$$

$$\begin{bmatrix} G & -G \\ -G & G \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} -I_R \\ I_R \end{bmatrix} \quad \begin{matrix} KCL_1 \\ KCL_2 \end{matrix}$$

# Device Example: Capacitor

- ⊗ Capacitors are charge storage devices.
- ⊗ Capacitor load:
- ⊗  $I_C = C \, dV/dT = dQ/dT$
- ⊗  $C = \text{capacitance} = C(V)$
- ⊗  $Q = \text{charge stored}$



$$\mathbf{J} \quad \Delta \mathbf{X} = -\mathbf{f}$$

$$\begin{bmatrix} C/\Delta T & -C/\Delta T \\ -C/\Delta T & C/\Delta T \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} -I_c \\ I_c \end{bmatrix} \quad \begin{array}{l} KCL_1 \\ KCL_2 \end{array}$$

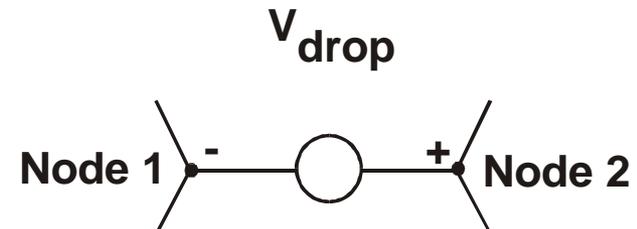
- ⊗ Time derivatives are obtained from the Time Integration Package

# Device Example: Voltage Source

- Independent voltage sources require current variables.
- Current through a voltage source not a function of voltage drop across the source (no  $I = V/R$  relationship!).
- Matrix stamp has no diagonal elements.
- Result: "one's pairs".

$$\mathbf{J} \cdot \mathbf{X} = -\mathbf{f}$$

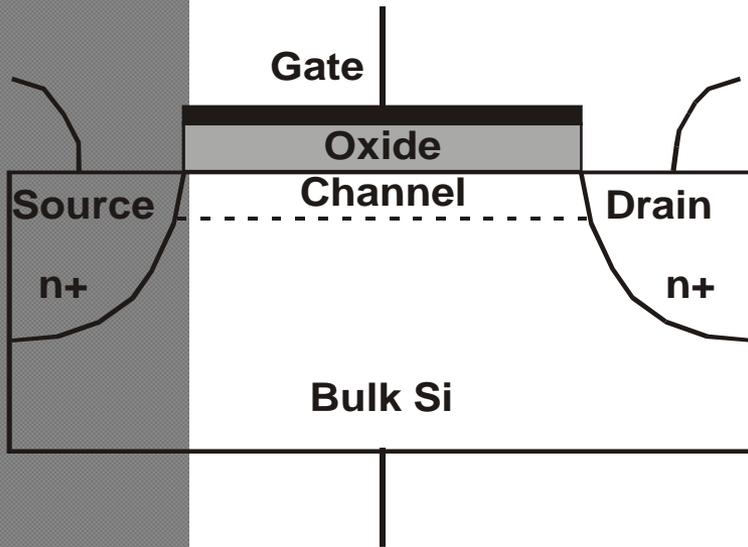
$$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta I_V \end{bmatrix} = \begin{bmatrix} -I_V \\ I_V \\ -V_1 + V_2 + V_d \end{bmatrix} \begin{matrix} KCL_1 \\ KCL_2 \\ Constr \end{matrix}$$



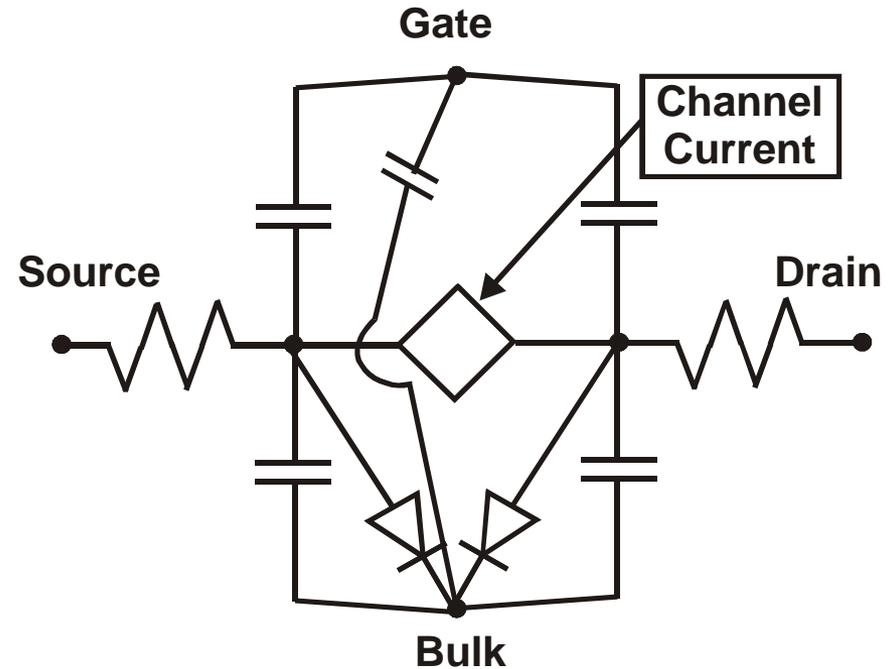
# Device Example: MOSFET BSIM3

- ⊕ BSIM3 = Berkeley Short Channel IGFET Model v.3
- ⊕ Established, 1996, by SEMATECH as an industry-wide standard for simulating integrated circuit MOSFETs
- ⊕ Crucial for ASCI milestone. Prerequisite for Rad-Hard Pentium simulation.
- ⊕ BSIM3 is one of the most complex device models in widespread use.
  - over 400 user-defined parameters.
  - In Xyce, the \*.C associated with this device ~ 13,000 lines.

# Device Example: MOSFET BSIM3

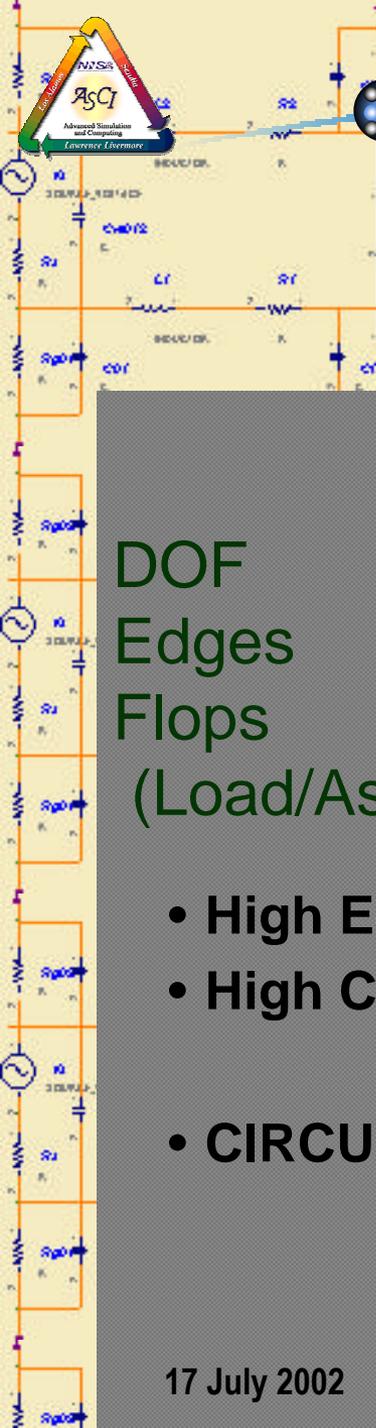


N-channel MOSFET cross-section



MOSFET Equivalent Circuit

✦ The complexity lies in the derivation of the nonlinear resistances, capacitances, and currents.



# Circuits VS. PDE's

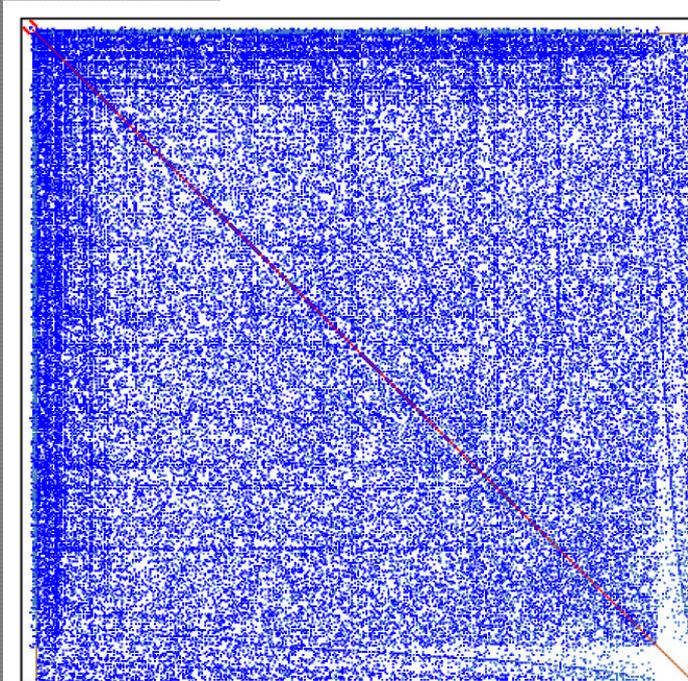
Mesh Node	Voltage Node	MOSFET
<5	1	0-3
~1-10	1- >10,000	4
1s-100s	0	>1,000

DOF  
Edges  
Flops  
(Load/Assembly)

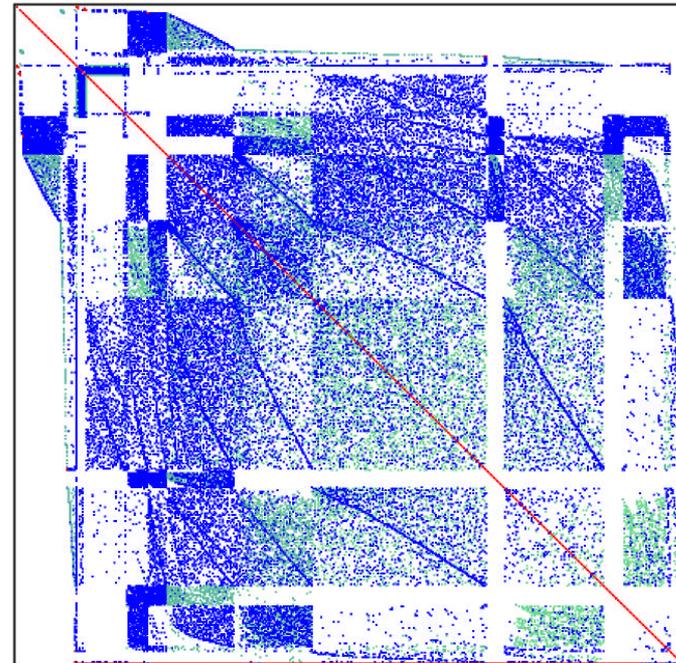
- High Edge Count → Dense Rows
- High Cost Load Calculations → Load Imbalance
- **CIRCUITS ARE NETWORKS RATHER THAN MESHES**

# Rad Hard Pentium Multiplier

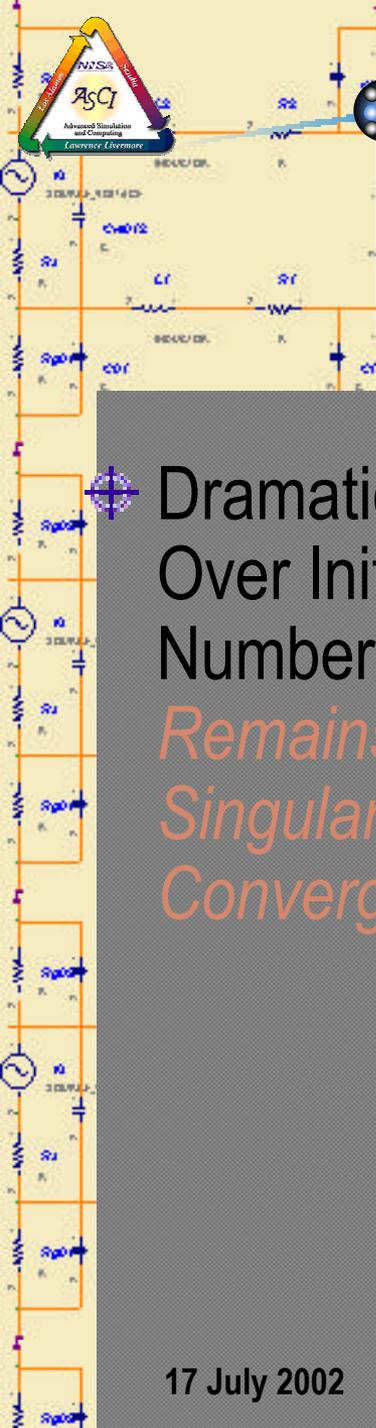
- ~70,000 MOSFET Transistors; ~25,000 Equations
- Max Edge Degree >5,000



Original

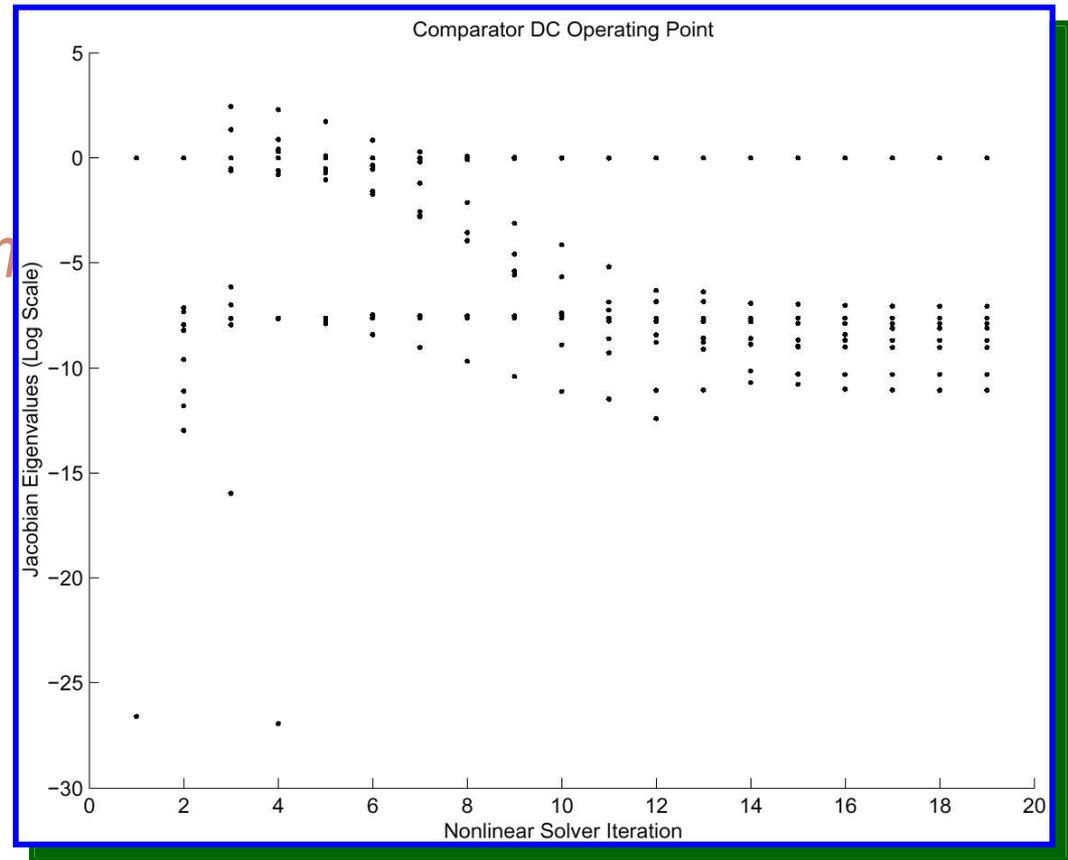


Block RCM



# DC-OP Eigenvalues

Dramatic Improvement Over Initial Condition Number *but the System Remains Numerically Singular until Convergence!*



# Circuit Simulation Challenges

Previously, no scalable, parallel circuit code existed

→ Attempts have resulted in ~50% efficiency on up to 8 processors.

Sandia has no history of developing large scale circuit codes

→ We're discovering new challenges as we develop new capability!

## ⊕ Algorithmic (time, nonlinear and linear solutions)

- Stiff, coupled DAE's → Different characteristics than PDEs
- Highly nonlinear (device model discontinuities, hysteresis, etc.)
- Large, ill-conditioned sparse Jacobian matrices present unique ordering and preconditioning challenges

## ⊕ Implementation

- Circuit problems can be very heterogeneous in terms of both the devices and the topology  
→ **There is no "characteristic" circuit!**
- Different computational phases scale differently



# Xyce

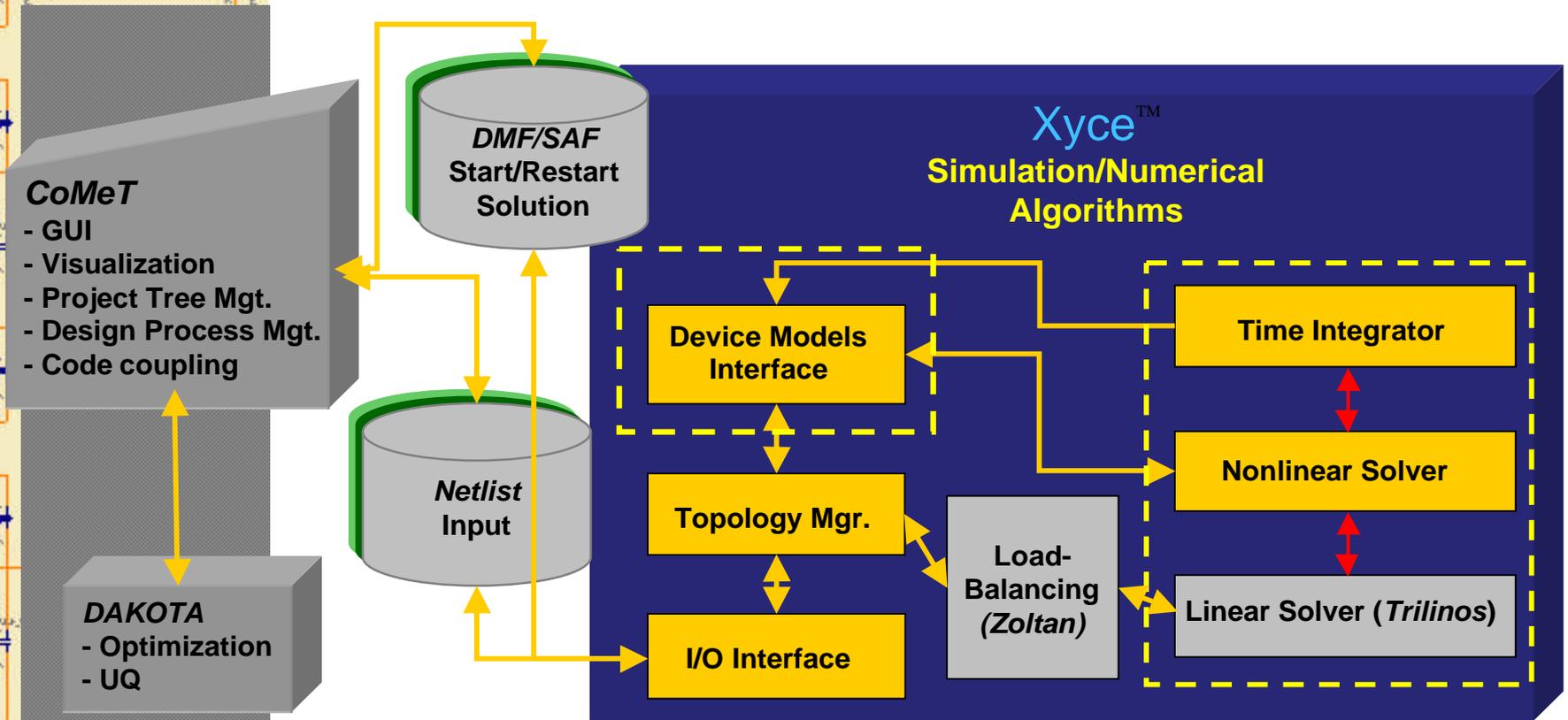
## Architecture

- Overview
- Topology
- Solvers
- Parallel

## Algorithms

- Partitioning
- Linear Solution

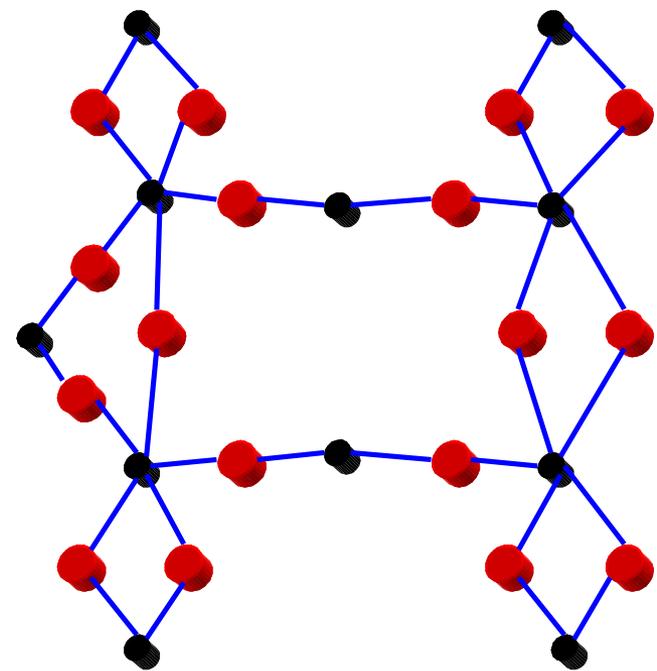
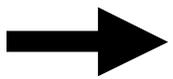
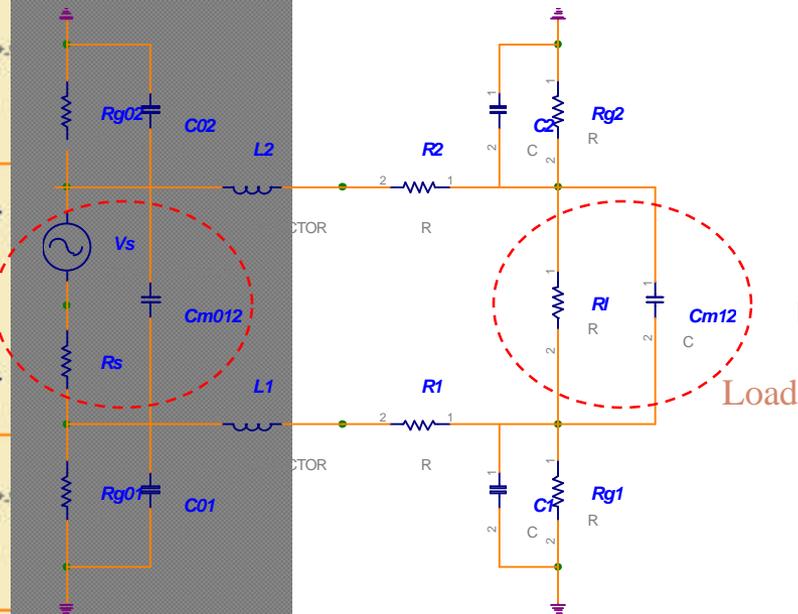
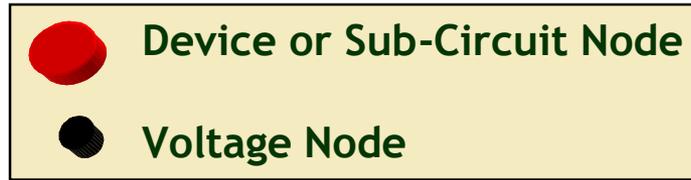
# Xyce Architecture

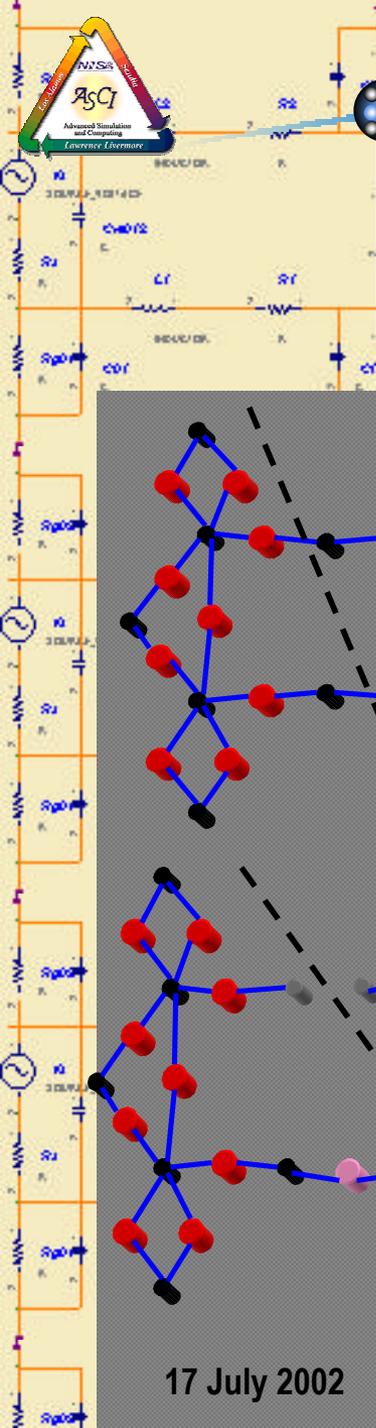




# Distributed Topology

## Generalized Distributed Topological Support





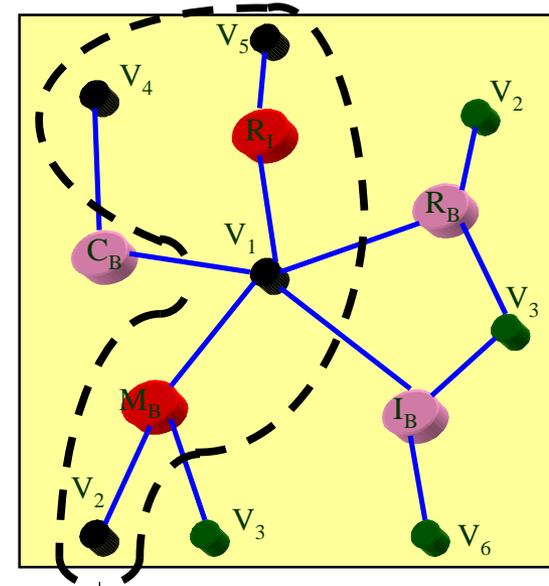
# Device "Ghosting"

## "Owned"/Internal node

- processor loads associated rows

## "Not Owned"/External node

- Dev-node: Load to V-node rows
- V-node: Reference for nonlocal data
- Requires global communication to update distribute shared solution vector data



Owned Not Owned

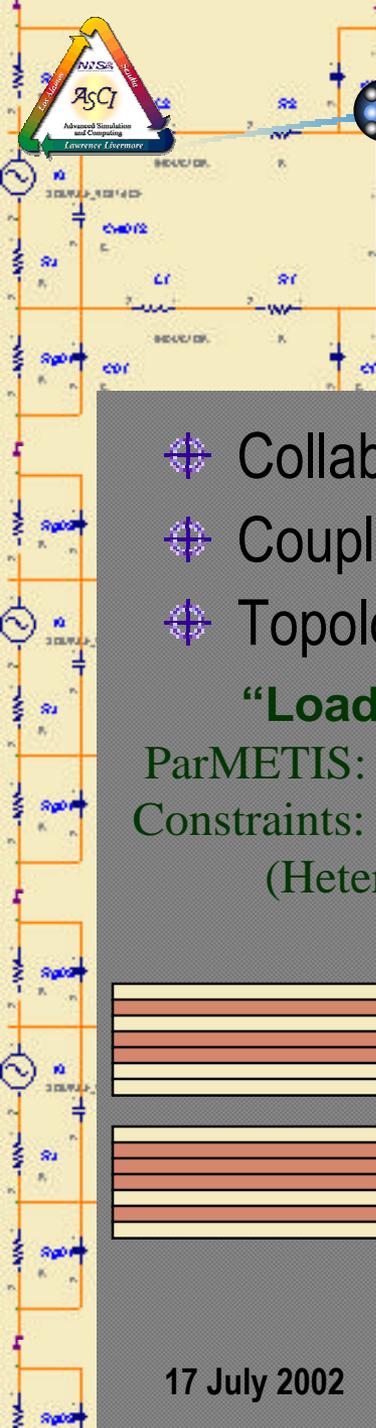
Device Node		
Voltage Node		

# Distributed/Parallel Tools

- ⊗ Communicator (Trilinos & Xyce)
  - Communication abstraction layer: serial, MPI, (PVM)...
- ⊗ Directory (Trilinos & Xyce) → Global Data Lookup
  - Distributed Hashed Database
- ⊗ Accessor (Trilinos & Xyce) → Global Data Access
  - Low cost data migration of basic data types and arrays
- ⊗ Migrator (Xyce) → Global Data Migration
  - Large scale data migration of arbitrary class structures (Devices, Ckt Nodes, etc)
- ⊗ Generic Factory (Xyce, In Progress)
  - Template Based Arbitrary Class Factory
- ⊗ Smart Pointer (In Progress)
  - Policy and Trait Based Pointer allowing user controlled reference counting, storage method, error checking, etc.

# Solvers

- ⌘ Close Collaboration with ASCI Algorithms at Sandia  
 (John Shadid, Mike Heroux, David Day, Roger Pawlowski, Tammy Kolda)
- ⌘ Time-Integration package:
  - Backward Euler, BDF2, Trapezoidal
  - Adaptive step-size control, Discontinuity breakpointing
- ⌘ Nonlinear Solver Package
  - Inexact Newton, Modified Newton, Steepest Descent, Trust Region, Hybrid
  - Globalization: Interval halving, Bank & Rose, Backtracking, More'-Thuente
  - Replaced by: NOX
- ⌘ Linear-Algebra Services Package: **TRILINOS**
  - Rich interface for tighter nonlinear/linear solver coupling
  - Flexible Norm Support
  - Dynamic Load Balancing - ZOLTAN



# Load Balance/Partitioning

- ⊕ Collaboration with Trilinos and Zoltan efforts (ASCI Algorithms)
- ⊕ Coupled Load and Solve Phases
- ⊕ Topology-ZOLTAN : TRILINOS-ZOLTAN Interface

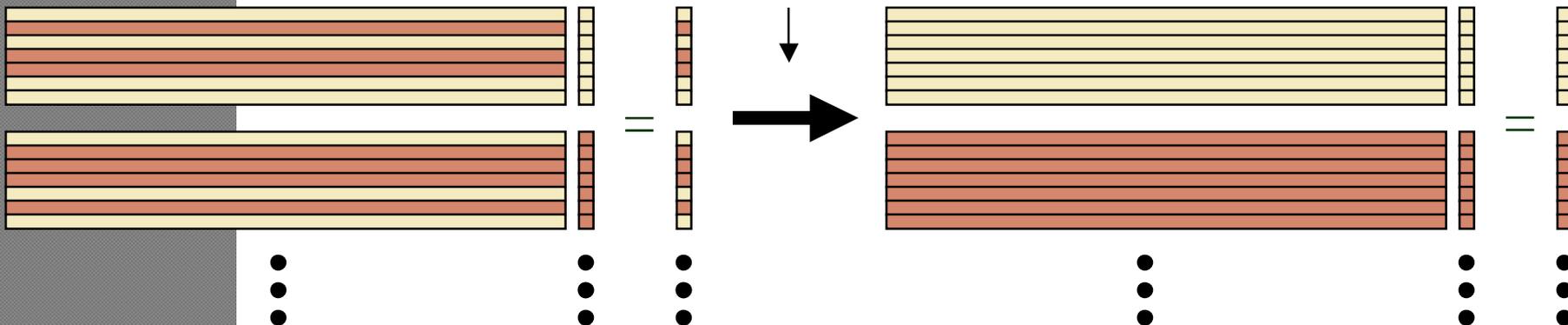
## “Load” Partitioning:

ParMETIS: “PartKway” and diffusion  
 Constraints: Load Balance  
 (Heterogeneous/Weighted)

## “Solve” Partitioning:

ParMETIS: “PartKway”  
 Constraints: Communication  
 Ordering  
 Preconditioning

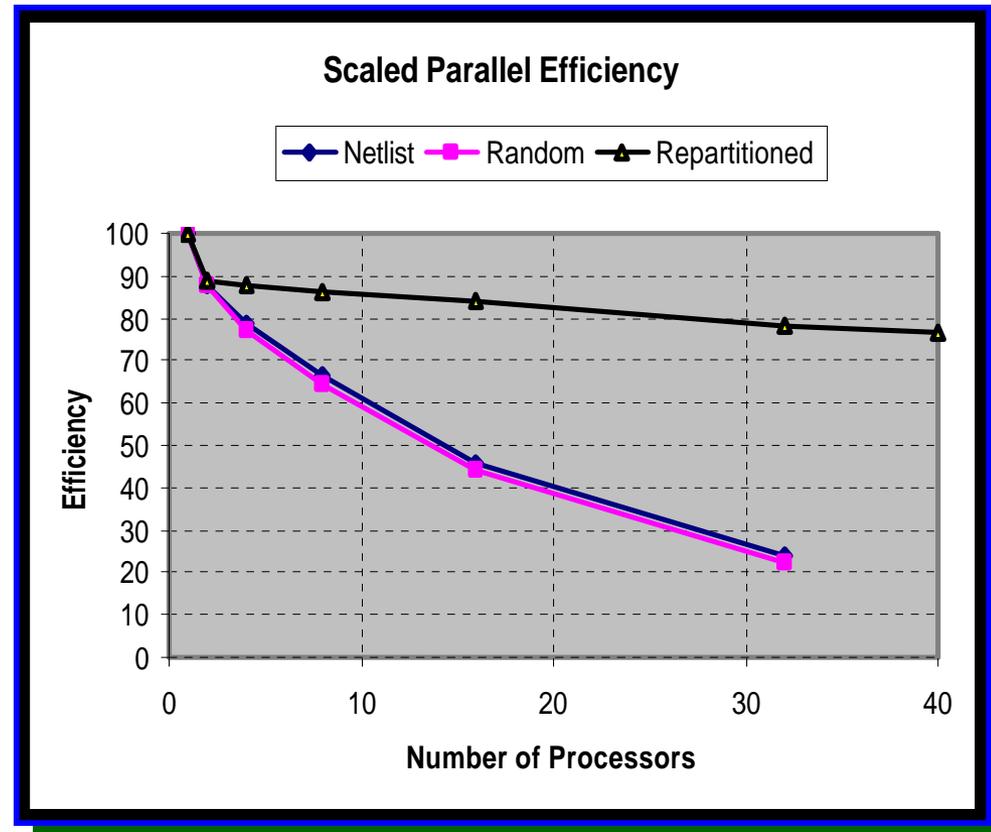
Communication Cost: Minimize

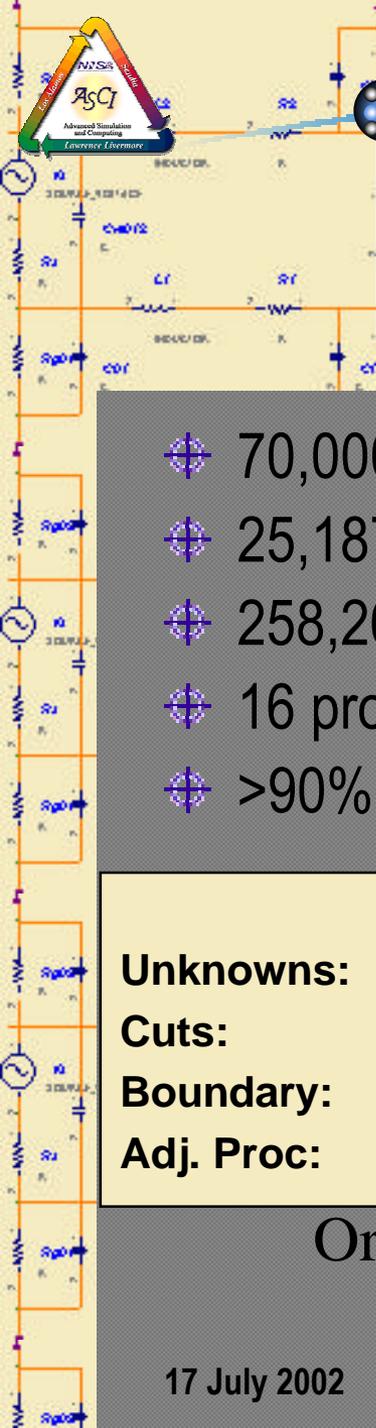




# Parallel Efficiency

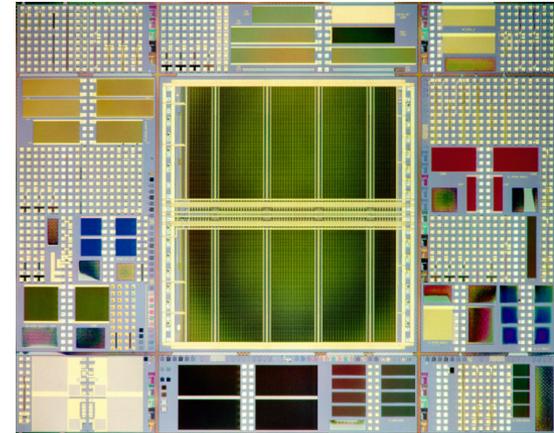
- Scaled Problem Size
- 3500 devices/processor on SGI Origin
- ~5 minute solve time
- Partitioning
  - Netlist Order
  - Random
  - Repartitioned
- Dramatic improvement in scalability for re-partitioned problem





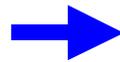
# Rad Hard Pentium Multiplier

- ⊕ 70,000 MOSFETs (of ~2 million)
- ⊕ 25,187 Unknowns
- ⊕ 258,265 NonZeroes
- ⊕ 16 processors
- ⊕ >90% of MOSFETs connect to power



	MIN	MAX	SUM
Unknowns:	1555	1566	25187
Cuts:	9399	77235	259126
Boundary:	1508	1553	24697
Adj. Proc:	15	15	240

Original



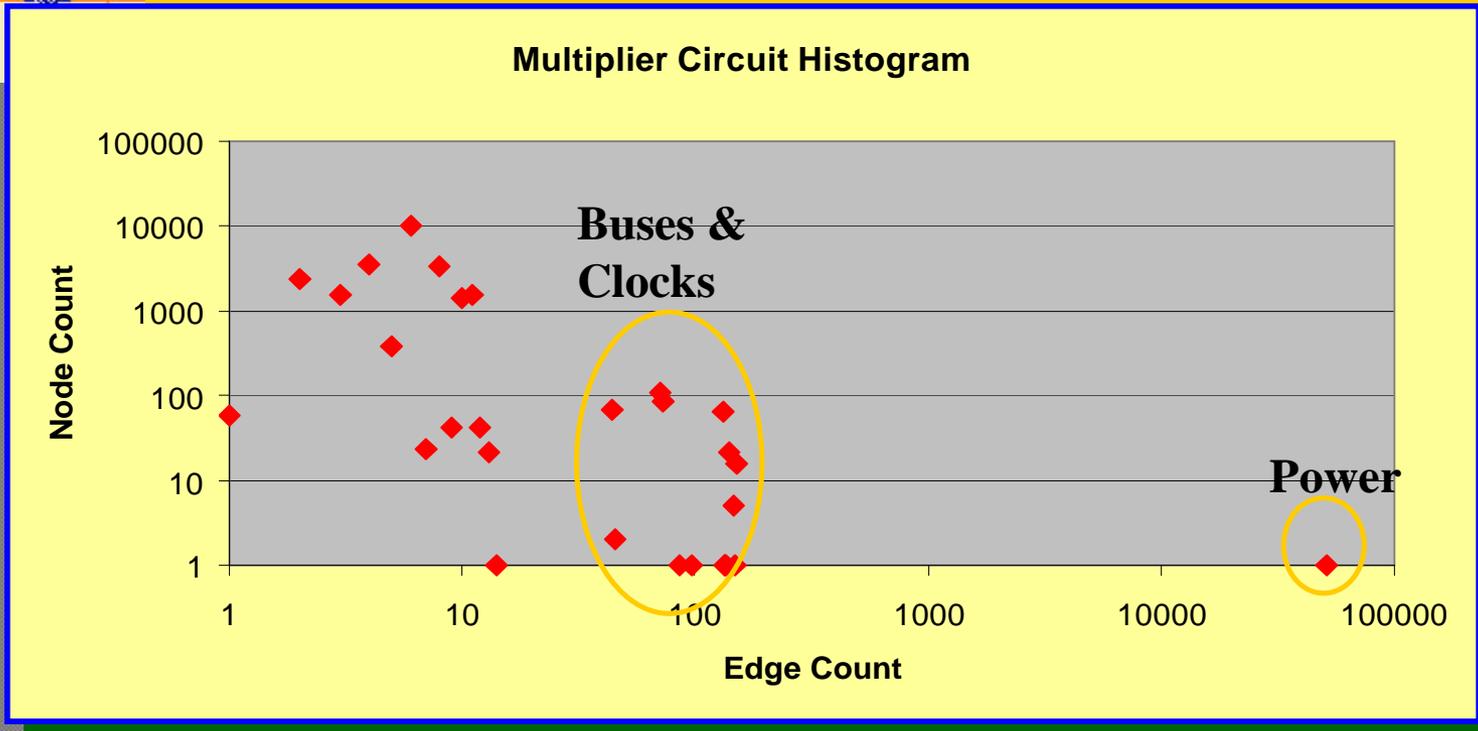
	MIN	MAX	SUM
Unknowns:	915	1995	25187
Cuts:	2592	47253	150800
Boundary:	844	1796	22653
Adj. Proc:	10	15	222

Repartitioned

58%

92%

# Rad Hard Pentium Multiplier



## Digital Circuits:

- Power node generates very dense row ( $\sim 0.9 \cdot N$ )
- Bus lines and clock paths generate order of magnitude increases in bandwidth

# Distributed Sources

- ⊕ Dramatic Decrease In Communication Volume → CUTS
- ⊕ Minimal Decrease In Communication Count → ADJ PROC

Parmetis – PartKway  
Linear System

	MIN	MAX	SUM
Unknowns:	915	1995	25187
Cuts:	2592	47253	150800
Boundary:	844	1796	22653
Adj. Proc:	10	15	222

CHACO – Multilevel-KL

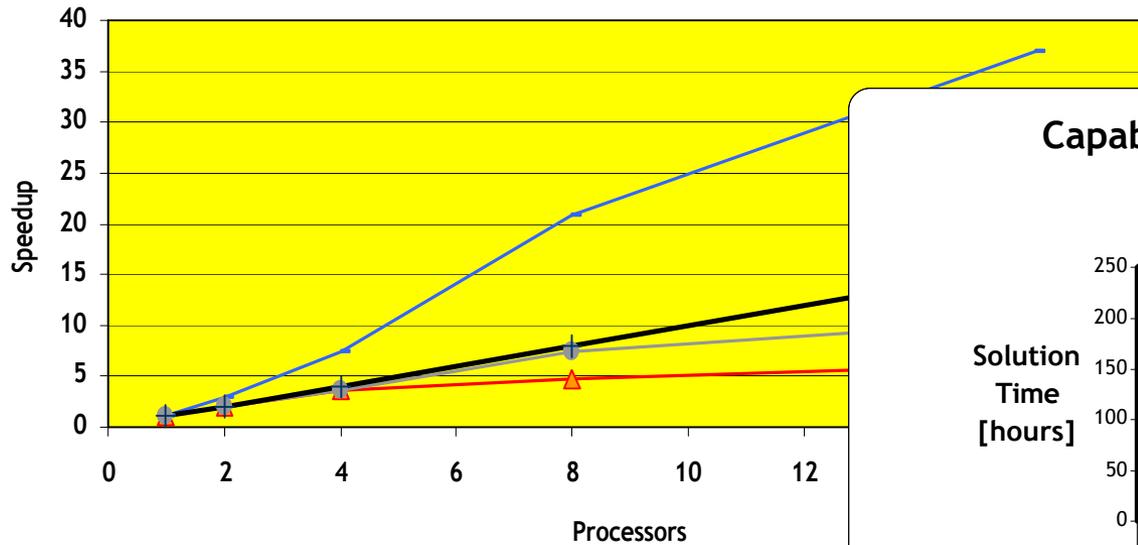
Circuit  
25% imbalance  
Distrib. Ind. Vsrc's

	MIN	MAX	SUM
Unknowns:	4856	7567	95756
Cuts:	417	1296	5719
Boundary:	377	1047	8807
Adj. Proc:	9	15	194

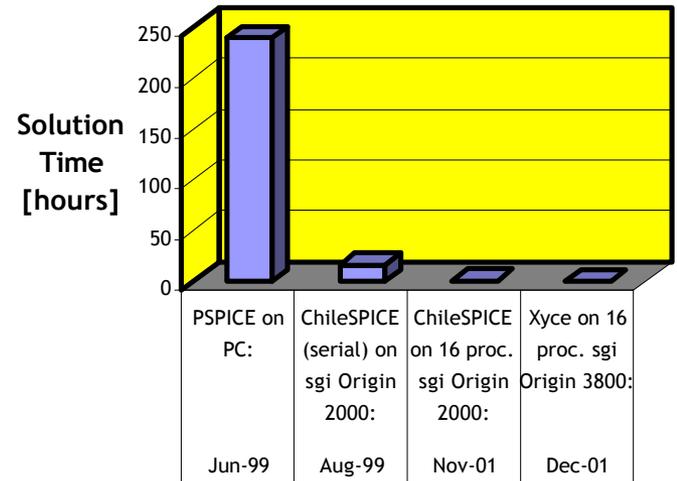


# RHP Multiplier Scaling

**RHP Multiplier Speedup**  
71,097 Devices, 28,609 Equations



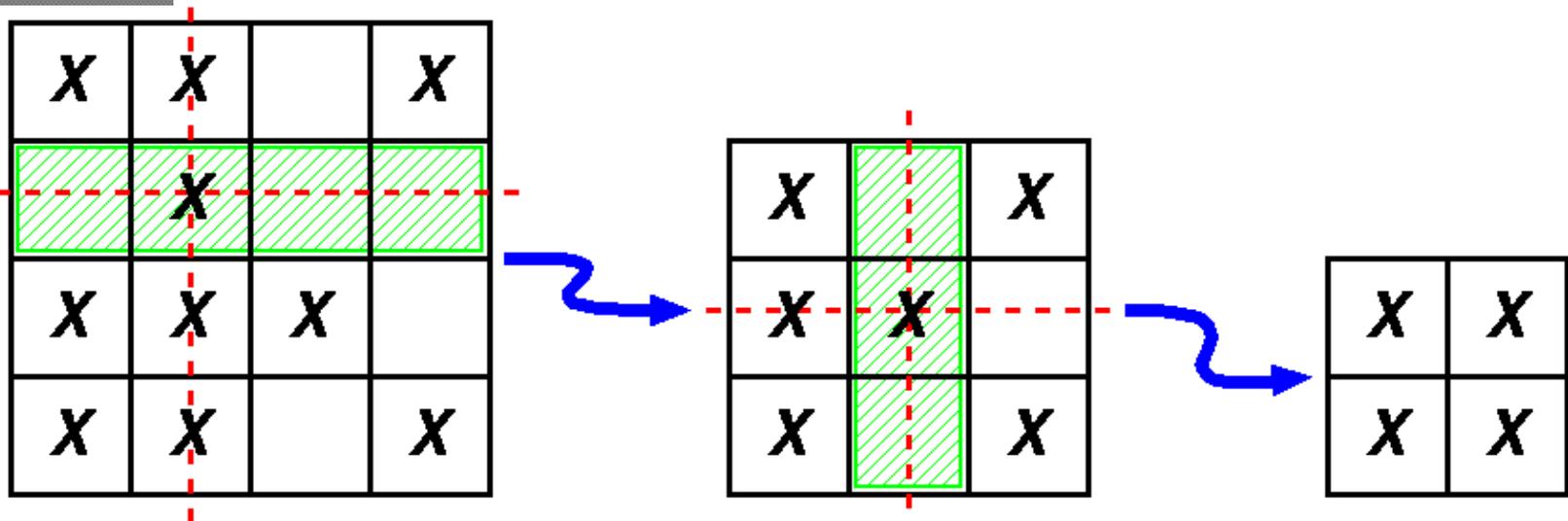
**Capability Timeline Example**  
RHP Multiplier Circuit



# Linear System Reduction

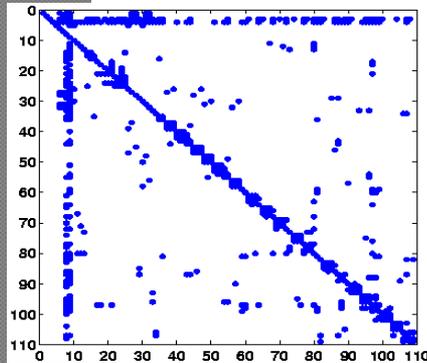
## Reduction to "Essential" System

- Constraints ('Singleton' Rows)
- Auxiliary Equations ('Singleton' Columns)
- Based on Achim Basermann's Results (NEC Europe)

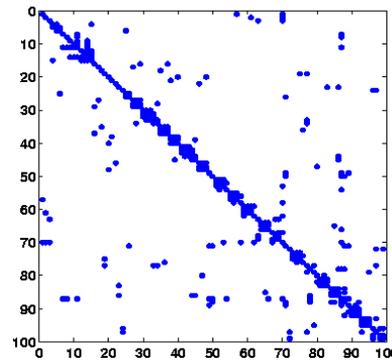


# Linear System Reduction

- ⊗ David Day's Block Algorithm
- ⊗ Implementation in Trilinos and XPetra
  - Global Search Algorithm
  - Sparse global GAXPY operation (BC Reduction)
  - Reduced System Remapping
  - Block Local Reordering (Zoltan)
  - Distributed Backsolve (Low Bandwidth)
  - Interface



**109 rows, 624 nonzeros**  
**Condition:  $8.3 \cdot 10^6$**



**99 rows, 385 nonzeros**  
**Condition:  $2.9 \cdot 10^3$**

$U_{11}$	$S_{12}$	$S_{13}$
	Reduced System <sub>22</sub>	$S_{23}$
		$U_{33}$

33: Global Backsolve

22: Global GAXPY  
 Iterative Solve

11: Global GAXPY  
 Global Backsolve

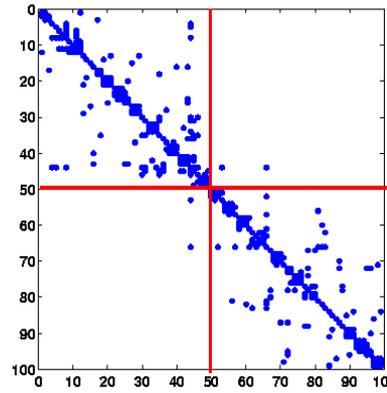
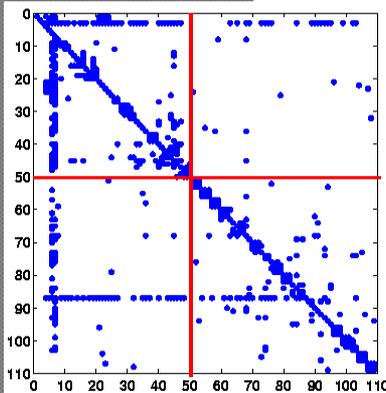


# Schur Complement Preconditioning

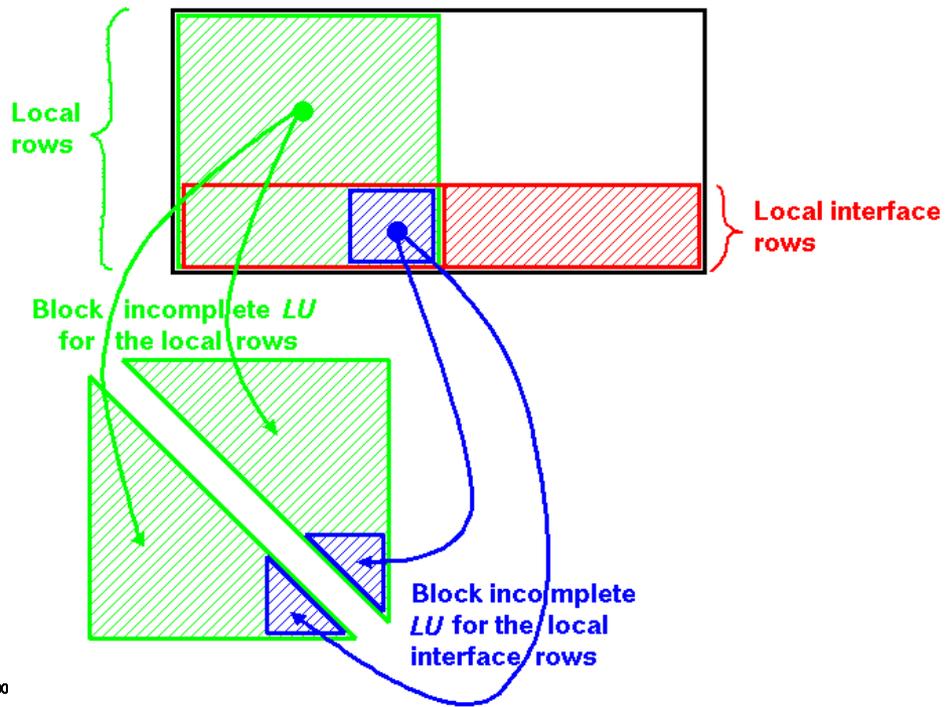
$$A = \begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} B & 0 \\ E & S \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ 0 & I \end{pmatrix}$$

$$S = C - EB^{-1}F$$

$$LU = \begin{pmatrix} L_B & 0 \\ E & L_S \end{pmatrix} \begin{pmatrix} U_B & L_B^{-1}F \\ 0 & U_S \end{pmatrix}$$



Original matrix Matrix without global nodes



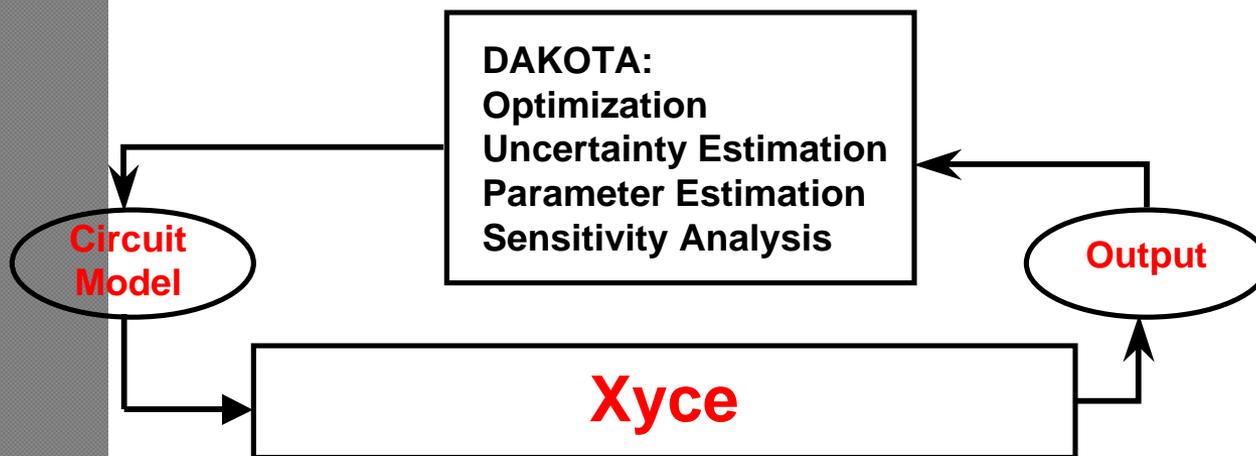
Saad and Sosonkina

# Coupled Simulation

- ✦ DAKOTA (Bart van Bloemen Waanders, Eric Keiter)
  - Optimization
  - Sensitivity
- ✦ PDE Semiconductor Devices (Eric Keiter)
  - Internal (1D & 2D)
  - Charon (New 3D FE Code)
- ✦ Digital/Mixed Signal (Univ. of Cincinnati)
- ✦ Radiation (ITS, NuGET, Entero)

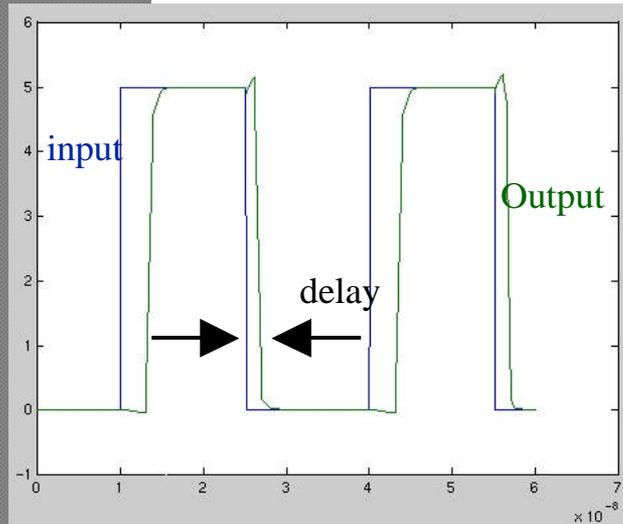
# Xyce/DAKOTA Black Box Optimization: Minimize Delay

- DAKOTA is a framework of tools for optimization, uncertainty estimation, and sensitivity analysis, for use with massively parallel computers.
- Design Goal: find optimal channel width and lengths for NMOS & PMOS devices to minimize delay of input and output signal
- Bigger devices: more charge storage, longer delays.
- Smaller devices: more expensive, more difficult to layout.

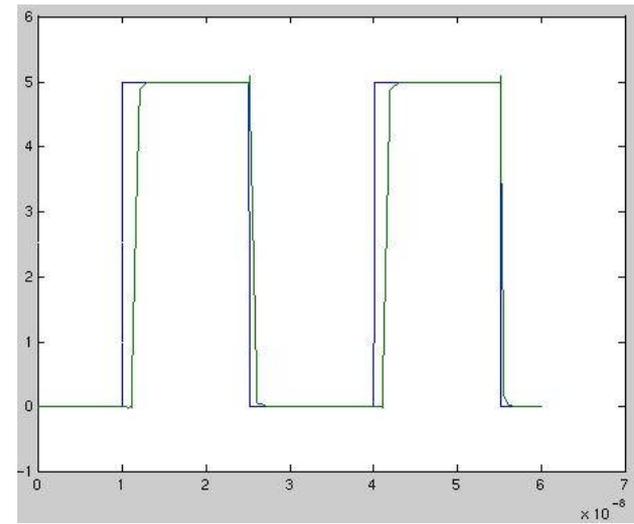


# Xyce/Dakota Results Comparator Circuit

## Nominal Design



## Final Design

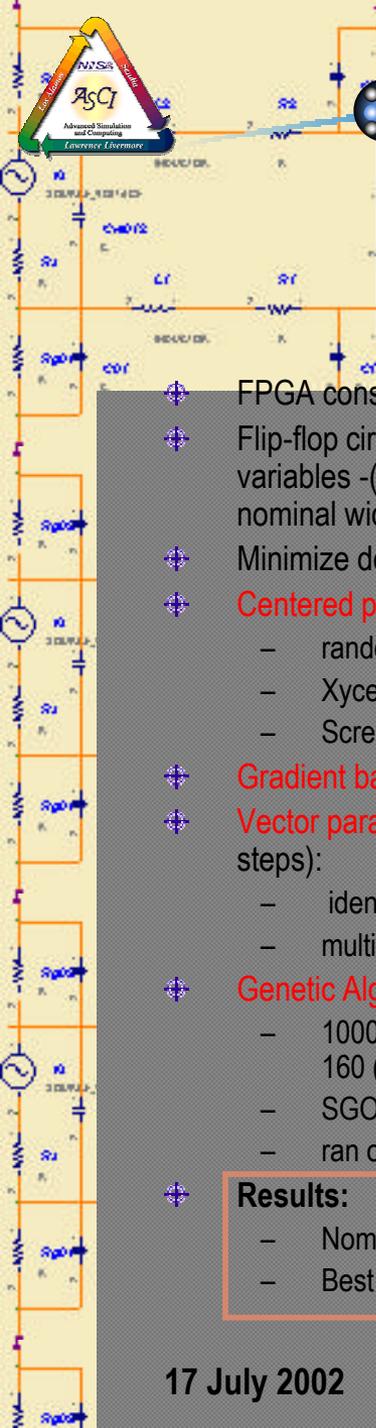


Channel length =  $2E-6$ , width =  $2E-6$

Channel length =  $1E-6$ , width =  $5E-6$

- Found solution in 6 fcn evaluations using gradient based method vs 50 fcn evaluations using coordinate pattern search
- Comparator circuit: ~ 20 MOSFETs.
- Smooth Objective function.

# Milestone FPGA circuit flip-flop device optimization

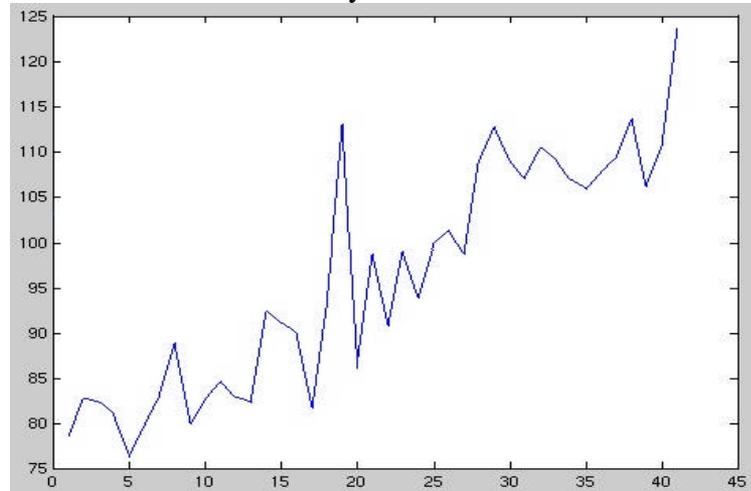


- ⊗ FPGA consists of XOR, AND, flip-flop sub-circuits
- ⊗ Flip-flop circuit - 34 devices divided into 12 design variables -(6 x widths/lengths) chosen based on nominal width and length specifications
- ⊗ Minimize delay between input and output signal
- ⊗ **Centered parameter** study results:
  - random lower values
  - Xyce terminate in certain design space
  - Screening for gradient based method
- ⊗ **Gradient based method** (npsol-sqp) failed,
- ⊗ **Vector parameter** study from initial point to bounds (40 steps):
  - identified non-smooth behavior
  - multi-modal
- ⊗ **Genetic Algorithm** study identified best design
  - 1000 function evaluations using population size of 160 (7 cycles)
  - SGOPT pga\_real (W.E.Hart)
  - ran overnight on 8 proc Linux cluster

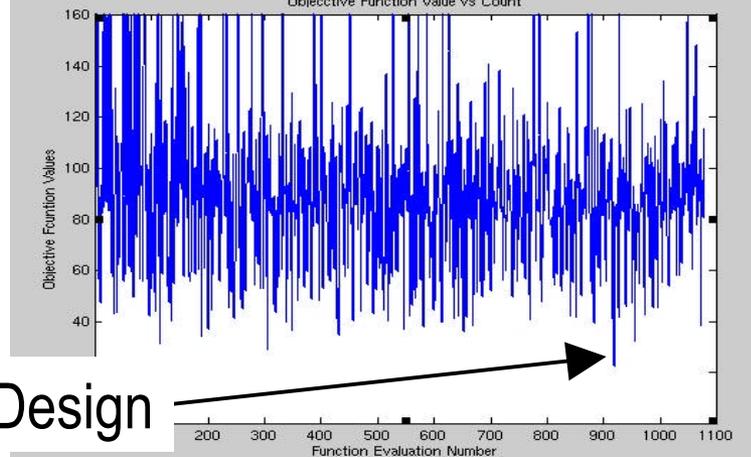
## Results:

- Nominal : 112.1
- Best : 29.2

Vector Parameter Study - non-smooth behavior



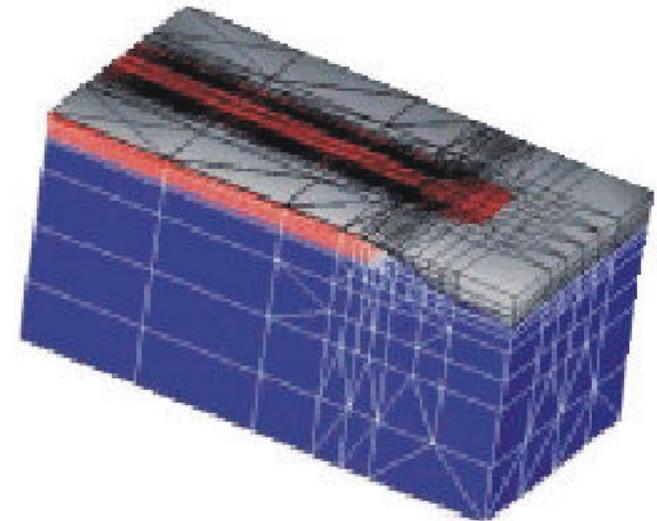
Objective Function Value vs Count



Best Design

# PDE Device Simulation

- ⊕ **Long term:** CSRF to develop **Charon**, a 3D reacting flow simulator. (see the next talk with Gary Hennigan)
  - Initial application: 3D device simulation.
  - Finite Element discretization (unusual in device simulation).
- ⊕ **Short term:** LDRD for Device-Circuit coupling.
  - 2D unstructured mesh device simulator built inside of **Xyce**.
  - Prototype for **Charon**.
  - Finite volume discretization - same as most commercial device codes: DaVinci, Pisces, Taurus, etc.



3D MOSFET mesh from  
DaVinci

# PDE Device Simulation

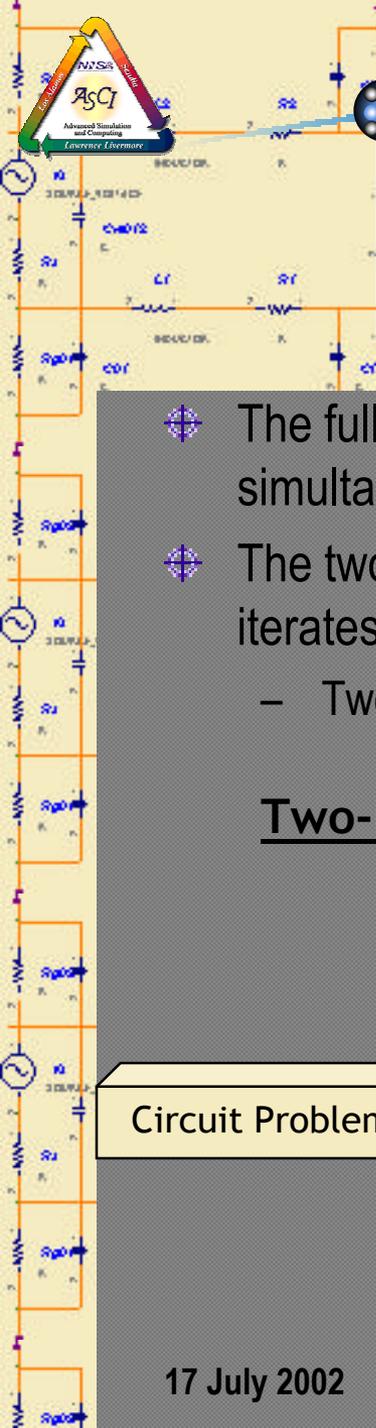
- Self-Consistently models transport of two charged species: electrons and holes.
- Numerically solves the semiconductor equations:

$$\text{Poisson:} \quad \nabla \cdot (\epsilon \nabla \Psi) = q (p - n + N_D^+ - N_A^-)$$

$$\text{Continuity:} \quad \frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot J_n - U + G \quad \frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot J_p + U - G$$

$$\text{where} \quad J_n = q \nabla D_n n - q \mu_n n \nabla \Psi \quad \text{and} \quad J_p = -q \nabla D_p p - q \mu_p p \nabla \Psi$$

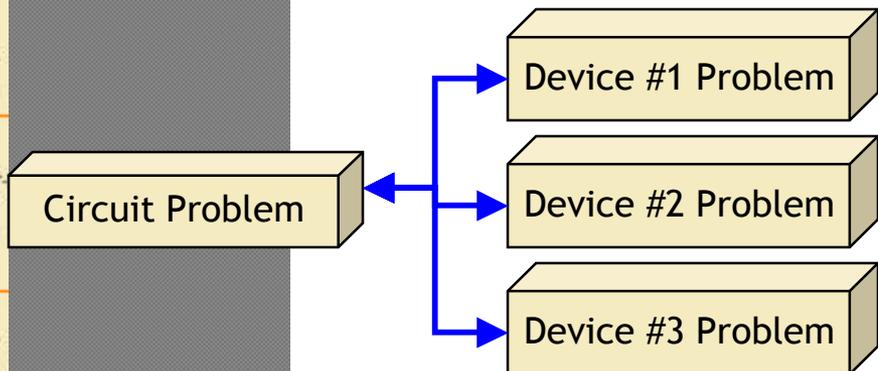
- These equations comprise the **drift-diffusion formulation**, which is the most common equation set for device simulation.



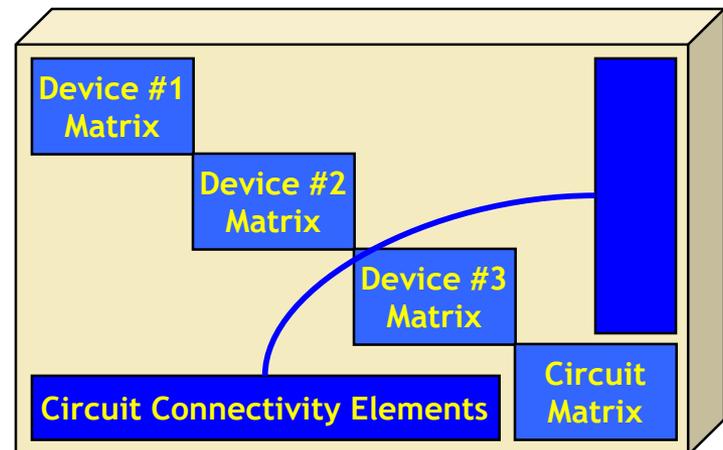
# Circuit-Device Coupling Approaches: Full-Newton and Two-Level Newton

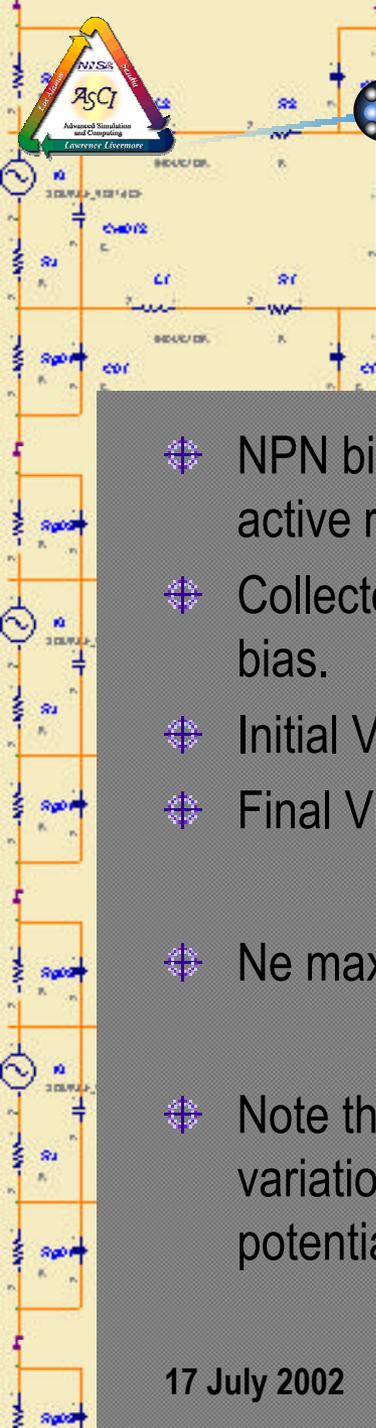
- ⊗ The full-Newton algorithm solves the PDE device and the ODE circuit nodes simultaneously using the same Jacobian matrix.
- ⊗ The two-level Newton algorithm solves the ODE circuit at one level and sub-iterates on the PDE device on a second level.
  - Two-level in Xyce will still use same Jacobian.

## Two-Level Newton



## Full Newton

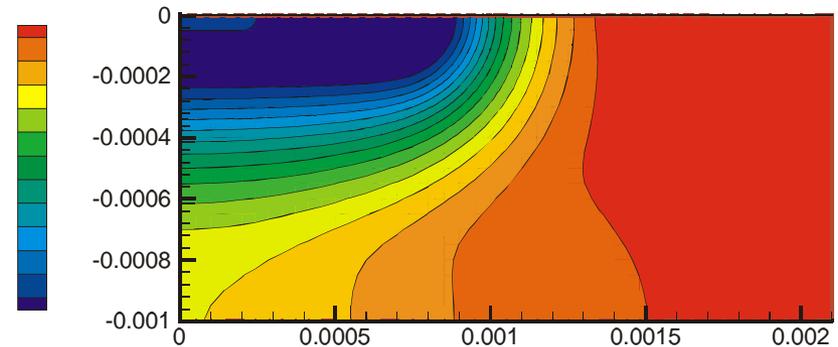




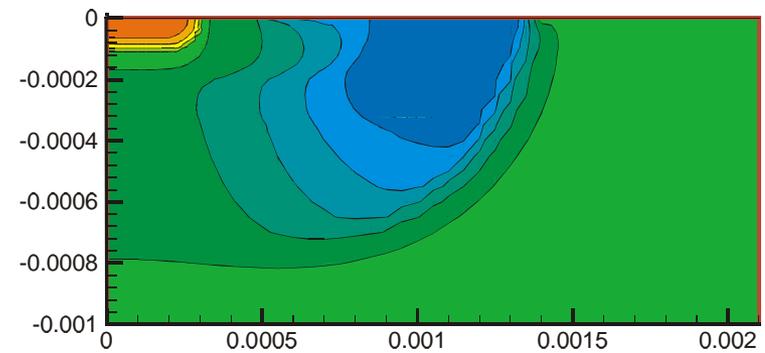
# DC(Steady State) Coupled Circuit/PDE Simulation

- ⊕ NPN bipolar transistor in the active regime.
- ⊕ Collector at a high positive bias.
- ⊕ Initial  $V_{max}$ :  $\sim 1.0V$
- ⊕ Final  $V_{max}$ :  $\sim 5.5V$
- ⊕  $N_e$  max:  $1.0e+19$  per ccm
- ⊕ Note the initial potential variation, due to the built-in potential of the device.

⊕ **Electrostatic Potential**



$1.0e+19$



⊕ **Electron Density**



# Fault Tolerance

## ✦ CURRENT

- Checkpoint
- Restart

## ✦ Issues

## ✦ Future

# Checkpoint/Restart

## ⊕ Full Featured Serial and Parallel Support

- Flexible checkpoint frequency control
- Independent repartitioning of restart run
- Pipelined through control node (Proc 0)

## ⊕ Efficient Storage

- Packed Data → Relatively Small Data Sets
- Transient nonlinear system state snapshot
- Internal Device State
- Solver Conditions (Breakpoints, Time Step, etc.)

# Questions

## ⊕ Output Commit Rate

- User Determined/Problem Dependent
- Relatively Small Data Sets

## ⊕ Checkpoint Data Size

- Only Full Checkpoint Currently
- Small: 0.1 → 10 Mbytes

## ⊕ Communication

- MPI, Single Threaded
- Driven by Linear Solver



# Questions

- ❖ Process Synchronization
  - Relatively Tight Due to Linear Solver
- ❖ Memory ?
- ❖ Bandwidth ?

# Issues/Future

- ✦ Requirements Similar to Transient PDE Simulation
- ✦ Fault Tolerance with Coupled Physics
  - **ASCI FY03 Level 1 Milestone**
  - PDE Devices
  - Radiation, Thermal : EXTERNAL CODES
- ✦ ASCI Platforms
  - MPI Variants
  - Custom OS and Compilers?

# Summary

- ✦ Xyce Version 1.0 Release (Oct., 2002)
- ✦ FY03 ASCI Electrical Milestone
- ✦ Fault Tolerance
  - Trilinos Solver Framework
  - Automated Checkpoint/Restart
  - Coupled Simulations