Xyce™

Parallel Circuit Simulation

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

April 3, 2002

Robert Hoekstra
Computational Sciences Department
Sandia National Laboratories
Albuquerque, NM, USA

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.

Sandia
National
Laboratories

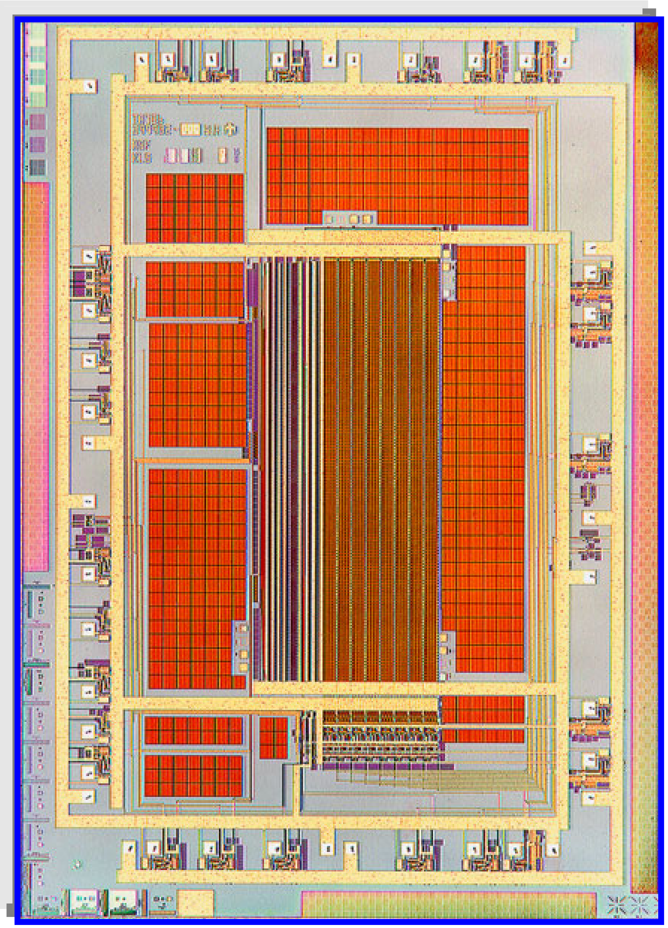
Overview

- Circuit Simulation
- Xyce
  - Device Models
  - Time Integration
  - Nonlinear Solver
  - Linear Solver
  - Partitioning/Load Balance
  - Optimization
- Conclusions

Circuit Simulation

- Circuit simulation is applied at several levels of abstraction:
  - Device (PDE)
  - Analog (ODE/DAE)
  - Digital (VHDL)
  - Co-Simulation (Circuit + Software)

Xyce™



- Analog simulation models a network of devices, typically described by ODEs, and coupled via Kirchoffs current and voltage laws
  - Several analysis options: DC Sweep, DC Operating Point, Transient, AC Analysis

Parallel Circuit Simulation Challenges

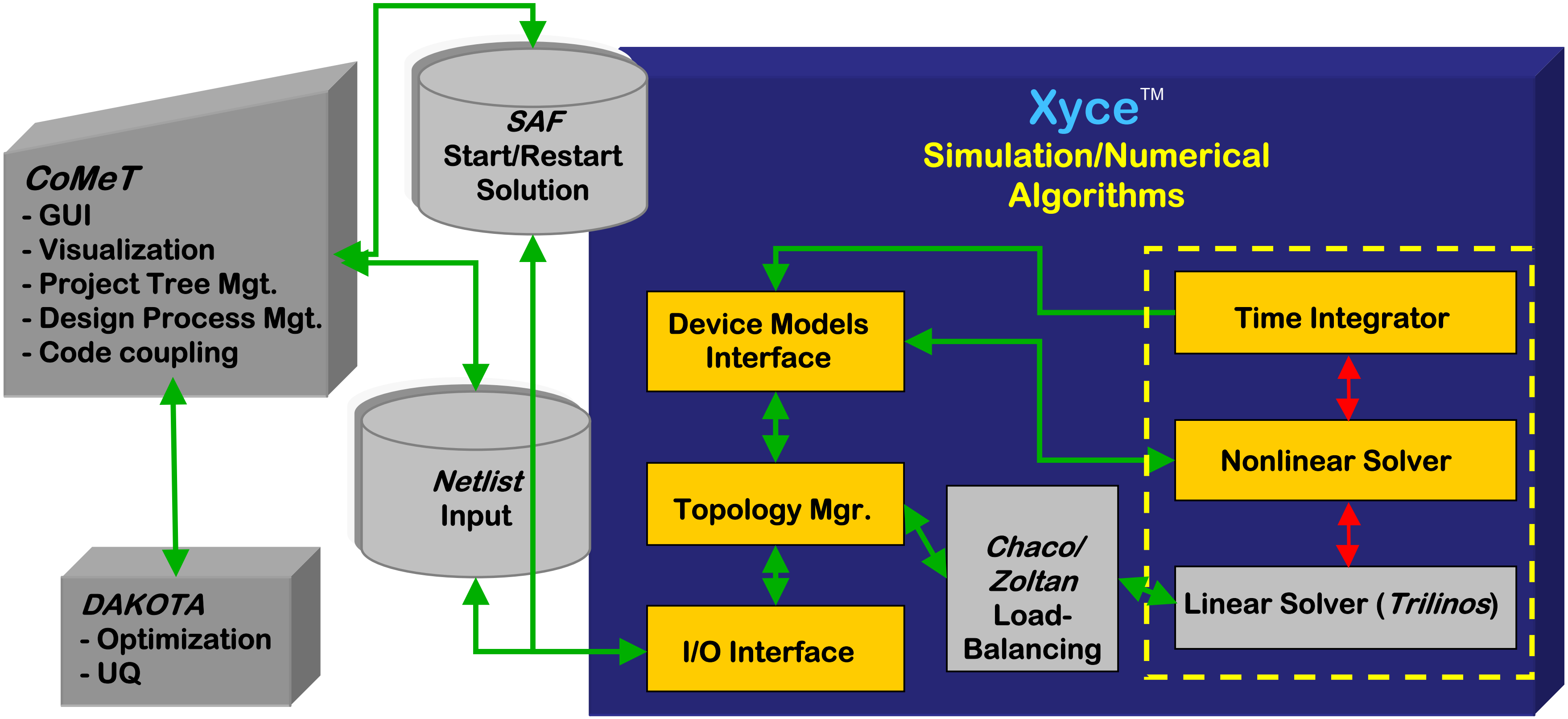
- ***Algorithmic (time, nonlinear and linear solutions)***
  - Stiff, coupled DAE's ➔ Different characteristics than PDEs
  - Highly nonlinear (device model discontinuities, hysteresis, etc.)
  - Large, ill-conditioned sparse Jacobian matrices present unique ordering and preconditioning challenges
- ***Implementation***
  - Circuit problems can be very heterogeneous in terms of both the devices and the topology
  - Different computational phases scale differently
- **Parallel Scalability**
  - Limited success with previous codes
  - Less than 8 processors, ~50% efficiency
- **SNL's First Effort**
  - No previous development effort at SNL
  - Application of state-of-the-art numerical tools developed at SNL

Sandia HPEMS Team

- ***Team Members***
  - Scott Hutchinson, Eric Keiter, Rob Hoekstra (09233)
  - David Day, Mike Heroux (09214)
  - Steve Wix (PI), Lon Waters, Regina Schells, Thomas Russo, Carolyn Bogdan (01734)
  - David Shirley (Abba Tech.)
  - Malcolm Panthaki (UNM)

- ***Customers***
  - Bill Ballard, Ken Marx, Steve Brandon (08418)
  - Marty Stevenson, Fred Anderson, Pat Smith (02612)
  - Doug Weiss (02343), George Laguna (02338)
  - Bob Brocatto (01735), John Dye (02331), Mark De Spain (02125), John Tenney (12333)

Xyce™ Kernel & Libraries

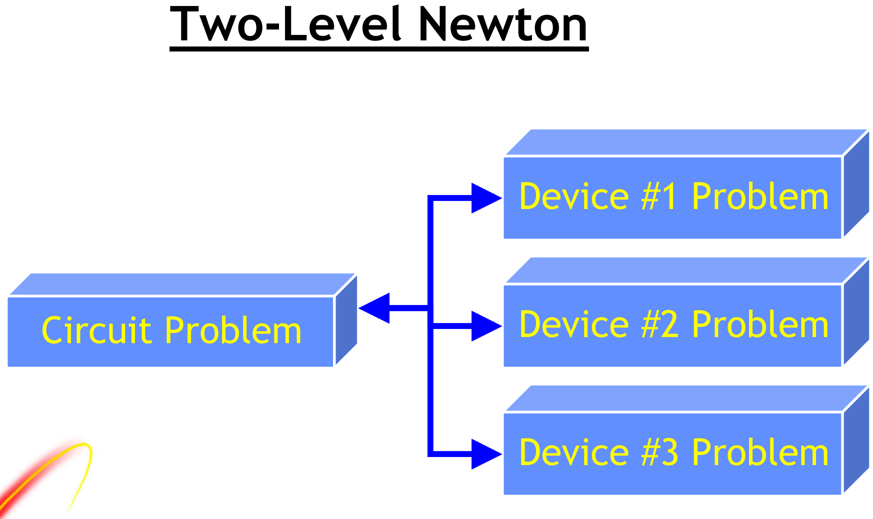


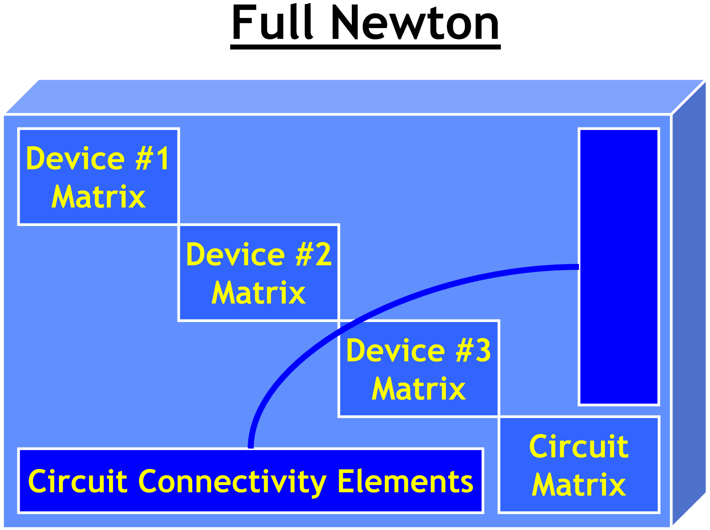
Device Modeling

- Modified Implementations of Spice3f5 Models
  - R-L-C, Mutual Inductance
  - Sources: Indep., Dep., Expression Based
  - Semiconductors: Diode, BJT, MOS1&3, BSIM3
- Semiconductor Junction “Voltage Limiting” Formulation
  - Similar to Spice
  - Modified to accommodate standard Newton update formulation
- Environmental Effects (SNL specific)
  - Temperature
  - Radiation
- PDE Modeling

PDE Devices: Motivation

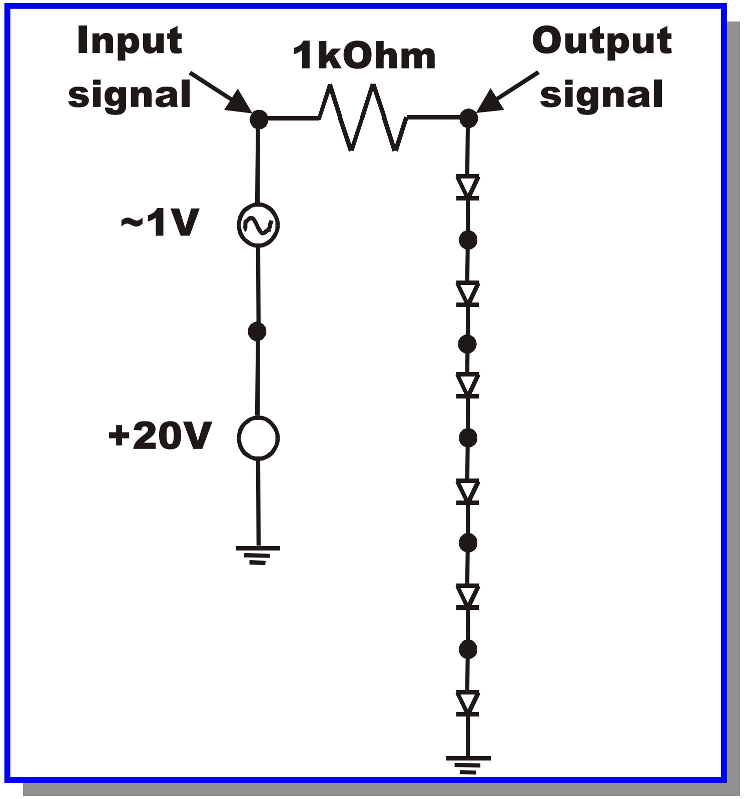
- Many radiation problems (like SEU) are only meaningful when you consider entire circuit, but need high fidelity of a PDE simulation.
- The goals of implementing PDE devices within Xyce are:
  - To investigate coupling issues between circuit and PDE device level simulation (LDRD)
  - Provide (long term) a PDE device simulation capability - *Charon*.

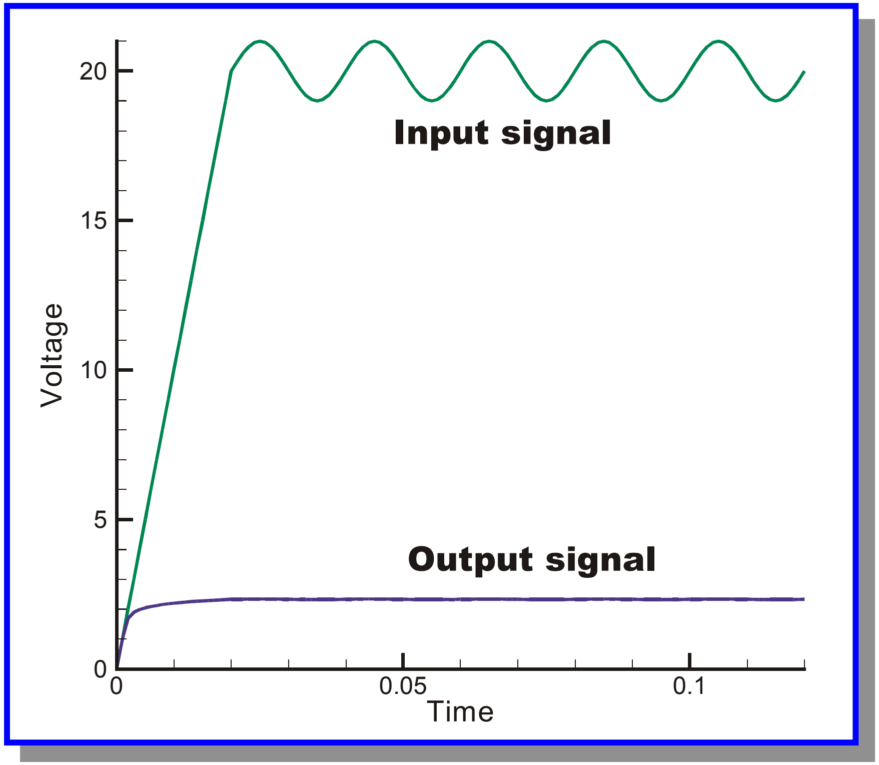




PDE Example: Voltage Regulator

- 6 PDE diodes in series





- 1D Drift-Diffusion Finite Volume PDE Diode (100 cells)
- PDE Devices implemented by Eric Keiter

Solver Technology

- Close Collaboration with *ASCI Algorithms* at Sandia

- Time Integration - John Shadid, David Ropp
  - Initial stages of development
- Nonlinear Solvers - Tammy Kolda, Roger Pawlowski, Andrew Salinger
  - NOX

Time Integration

- Techniques
  - Backward Euler, BDF2, Trapezoidal Integration
  - Adaptive step-size control
  - Discontinuity Breakpointing
- Error estimation
  - Spice
    - weighted maximum norm
    - includes aux. variables
  - Xyce
    - weighted RMS norm
    - soln variables only
- Future, Needs?
  - Adaptive techniques in Pspice
  - Flexible error estimation

Nonlinear Solvers

- Spice
  - Non-std Newton formulation, $Jx_{new}=Jx_{old}+f$
  - *In situ* limiting of semiconductor junction voltages
- Xyce
  - Current
    - Methods: Inexact Newton, Modified Newton, Steepest Descent, Adaptive Method
    - Globalized Searches: Interval Halving, Bank & Rose, Backtracking
  - Modified "voltage limiting" formulation
    - Impacts viability of globalized search methods
    - Hysteresis
  - NOX: Tammy Kolda, Roger Pawlowski
    - Newly integrated
    - Techniques such as Trust Region and More'-Thuente search
    - LOCA integration - continuation techniques

Nonlinear Solver Improvements

Nonlinear Solution Procedure:

Solution update: $x_{k+1} = x_k + \alpha_k p_k$

Update direction: $p_k = -B_k^{-1} \nabla f_k$

where $B_k = \begin{cases} I, & \text{Steepest descent} \\ J_k, & \text{Newton's method} \end{cases}$

Update direction: $\cos\theta_k = \dfrac{-\nabla f_k^T p_k}{\|\nabla f_k\| \|p_k\|}$

Descent reqm't.: $\cos\theta_k \geq \delta > 0, \quad \text{for all } k$

RHP Adder Circuit Operating Point Example:

	Simple Newton	*Steepest Descent + Newton + Line Search*
Number of Steps (Jacobian Evaluations)	562	141
Number of Linear Solutions	562	136
Number of Residual Evaluations	563	517
Solution Time (secs.)	6944	1789

Linear Solvers

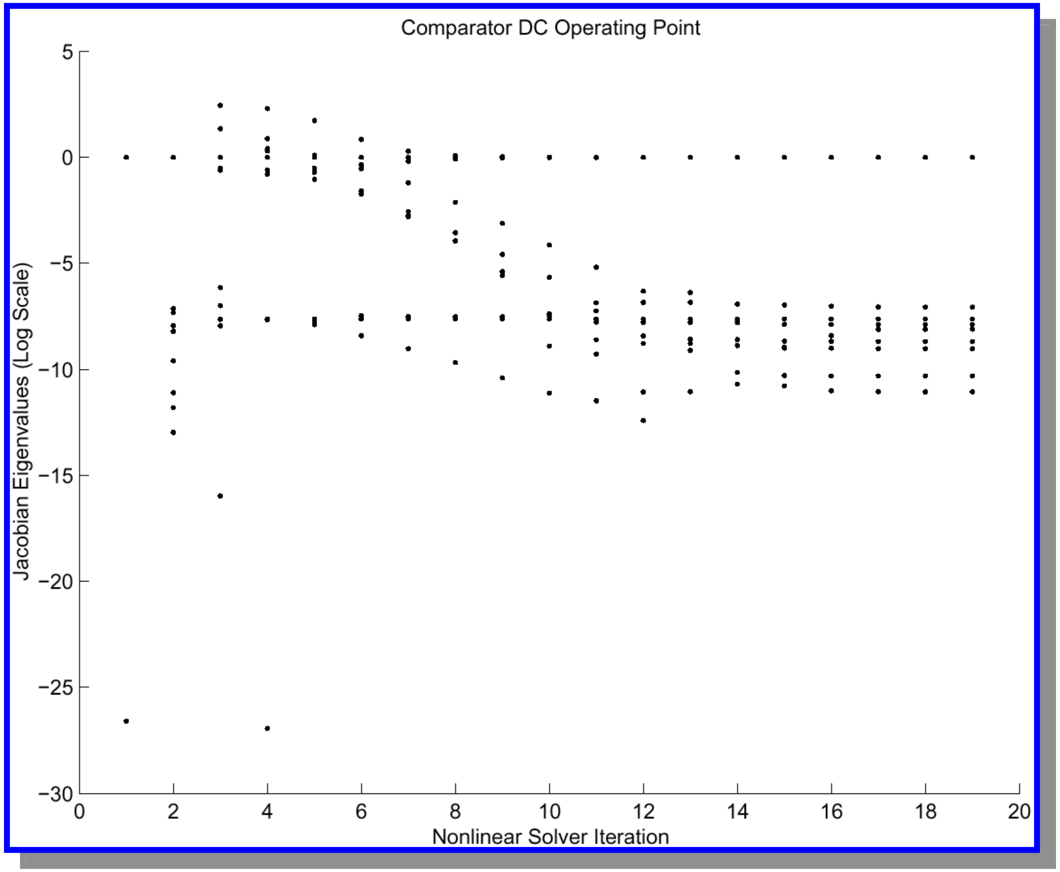
- Sparse Direct
  - Custom solver used by Spice3f5
    - Twins Reordering
    - Current/Voltage Scaling
  - SuperLU used by Xyce
  - Proven efficiency and robustness for small/serial simulations
- Distributed Sparse Iterative (TRILINOS)
  - GMRES
  - ILUT
  - Diagonal Perturbation

Trilinos Solver Framework

- Abstraction Layer for Solvers
  - Aztec, PETSc, ML, SuperLU, Ifpack
  - Common API
  - Compositional Class structures
- Petra (Epetra, Tpetra)
  - Concrete Implementaton of Linear Objects
  - Vector, MultiVector, RowMatrix, Operator
- Future Usage
  - Algebraic Multi-Level/Schur Complement
  - Block Preconditioning

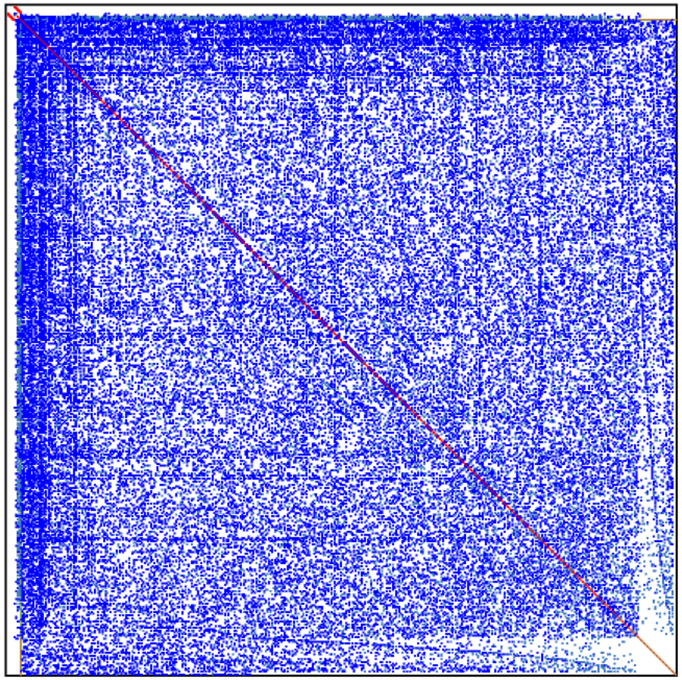
Comparator DC-OP Eigenvalues

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

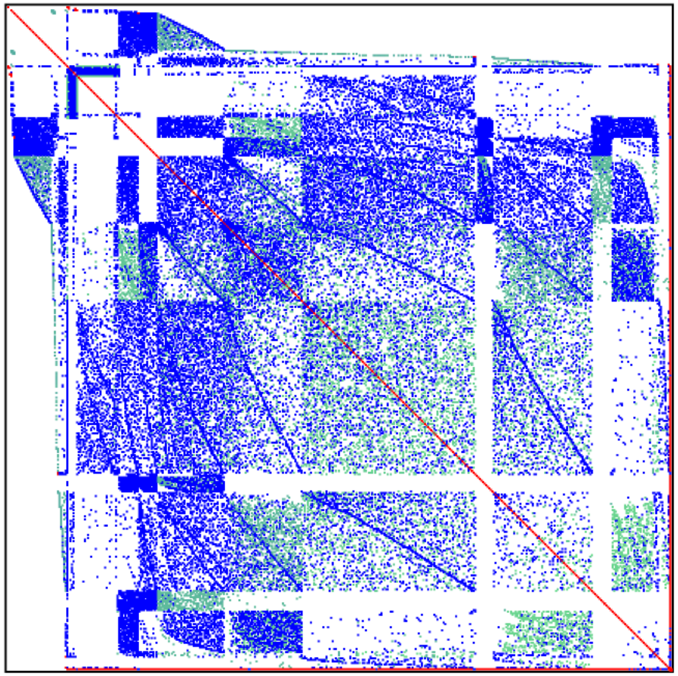


RHP Multiplier – Block RCM

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



Original

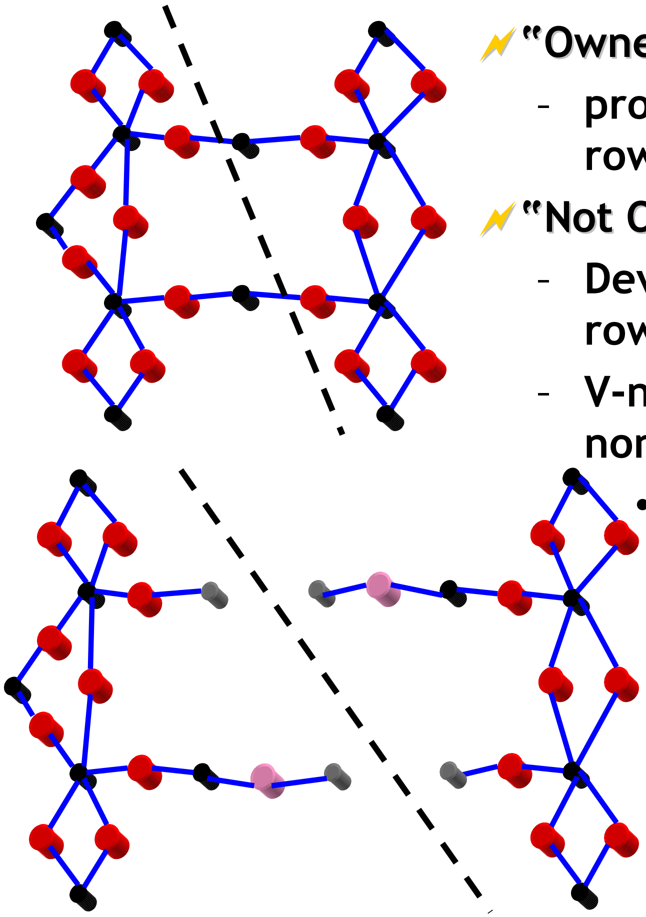


Block RCM

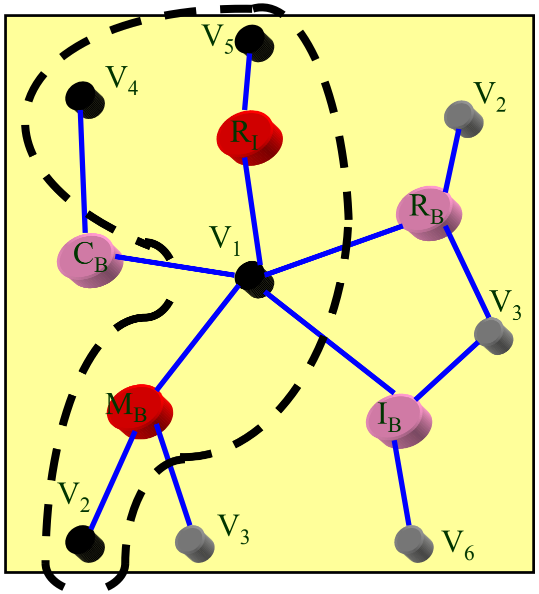
Dynamic Partitioning & Load Balancing

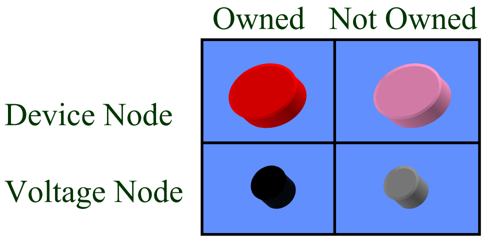
- Parallel Topological Description
- Close Collaboration with Sandia's Trilinos OO Solver Library & Zoltan Parallel Load-Balancing Tools (*ASCI Algorithms*)
- *Dynamic Partitioning & Migration*
  - Linear System (*in Trilinos*)
  - Circuit Network - Ghosting & Non-ghosting
- "Global Lookup Directory" support
  - Dependency Resolution
  - Migration

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





Load Balance / Partitioning

- Coupled Load and Solve Phases
  - Competing Criterion
- Topology-ZOLTAN : TRILINOS-ZOLTAN Interface

“Load” Partitioning:

ParMETIS and CHACO

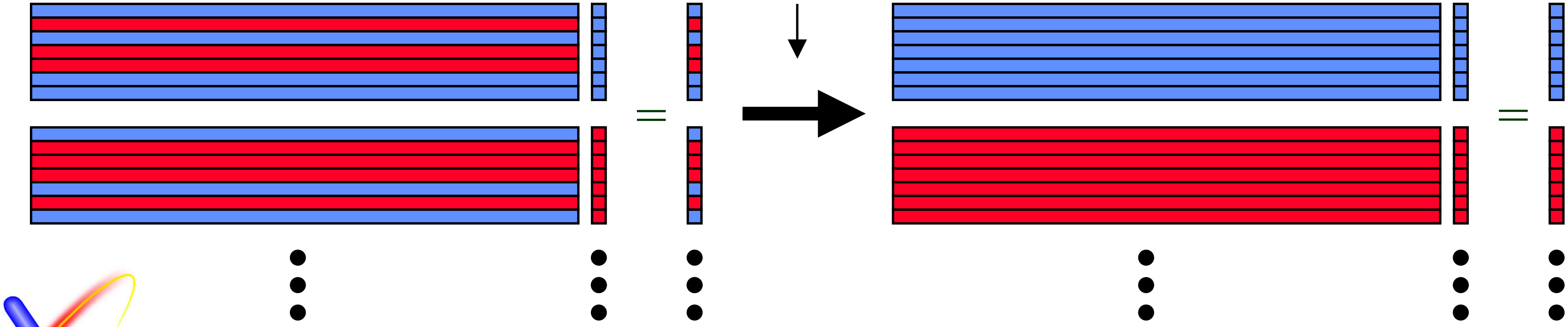
Constraints: Load Balance
(Heterogeneous/Weighted)

“Solve” Partitioning:

ParMETIS

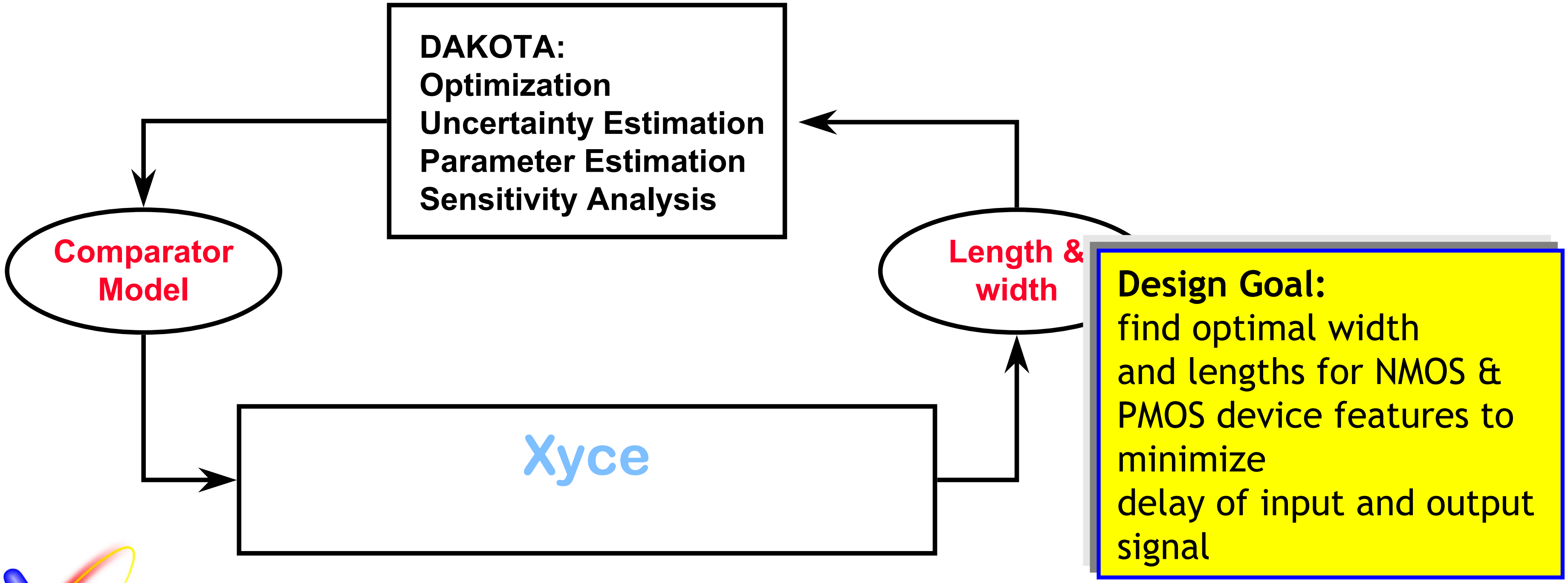
Constraints: Communication
Ordering
Preconditioning

Communication Cost: Minimize



Optimization (DAKOTA)

- Optimization of Circuit & Device Performance
- DAKOTA is a framework of tools for optimization, uncertainty estimation, and sensitivity analysis, for use with massively parallel computers. (Bart van Bloemen Waanders, Eric Keiter)

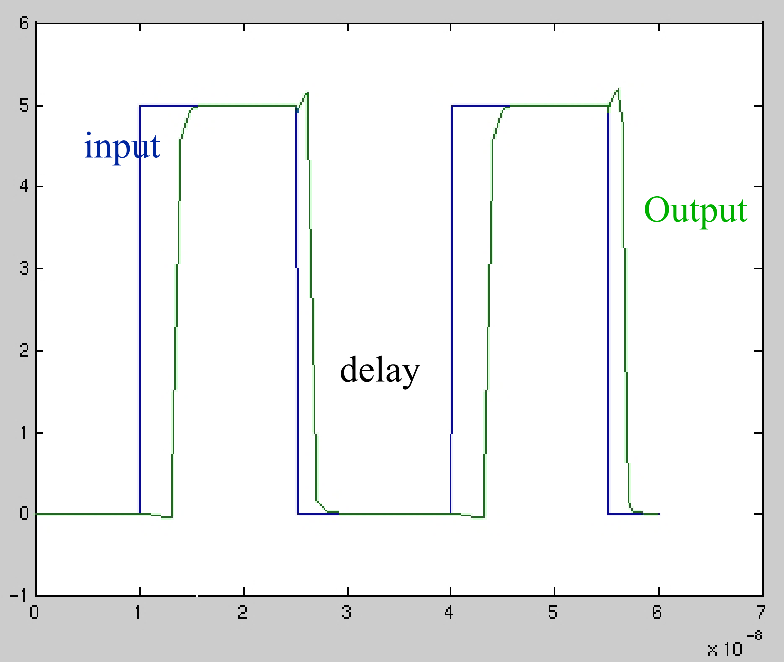


DAKOTA:
Optimization
Uncertainty Estimation
Parameter Estimation
Sensitivity Analysis

Design Goal:
find optimal width and lengths for NMOS & PMOS device features to minimize delay of input and output signal

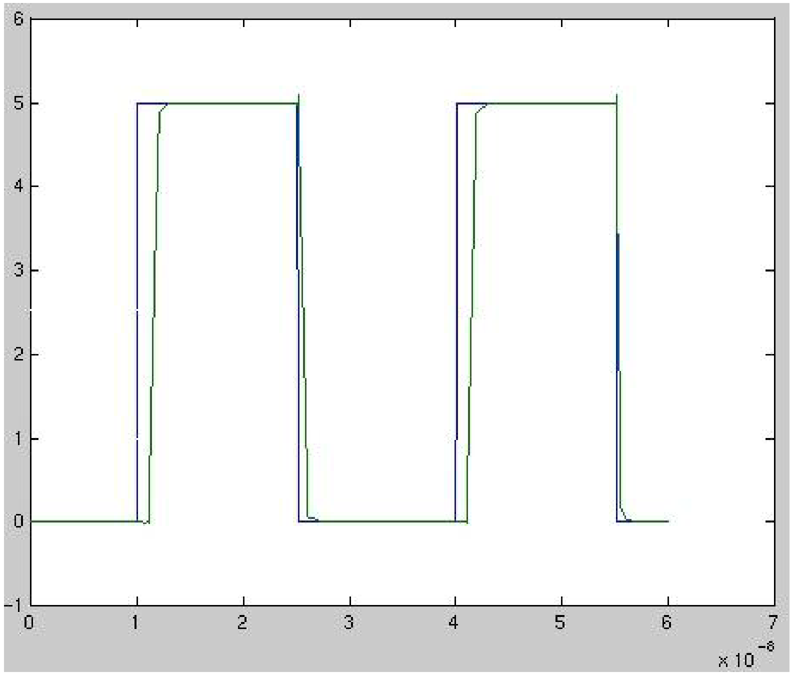
Xyce/Dakota Minimize Delay Results Comparator Circuit

Nominal Design



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

Final Design



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

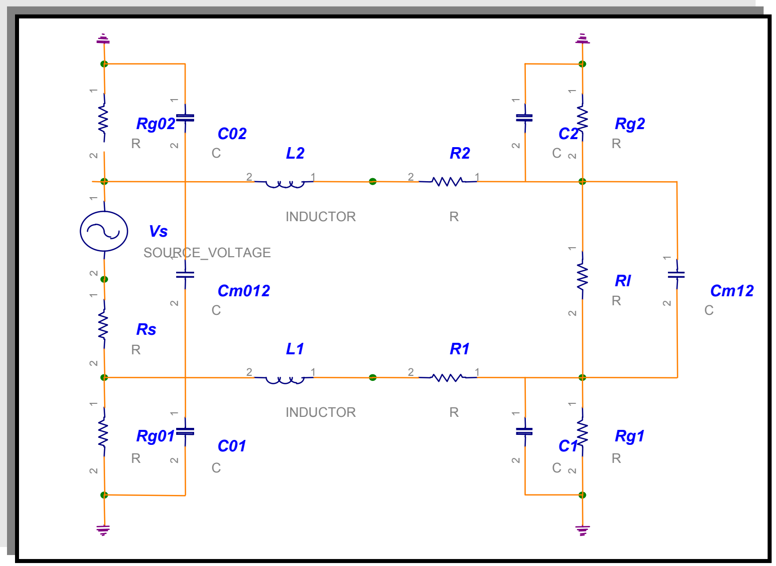
* Found solution in 6 fcn evaluations using gradient based method vs 50 fcn evaluations using coordinate pattern search

Results

- "Microstrip" transmission line
  - Partitioning
  - Scaling
- Pentium Multiplier
  - Partitioning
  - Independent Voltage Source Distribution
  - Scaling

μStrip (Transmission Line)

- Simple RLC Network
- Low Connectivity
- Easily Scaled
- 16 proc/60,000 devices



	MIN	MAX
Unknowns:	1475	1536
Cuts:	821	1431
Boundary:	513	845
Adj. Proc:	2	6

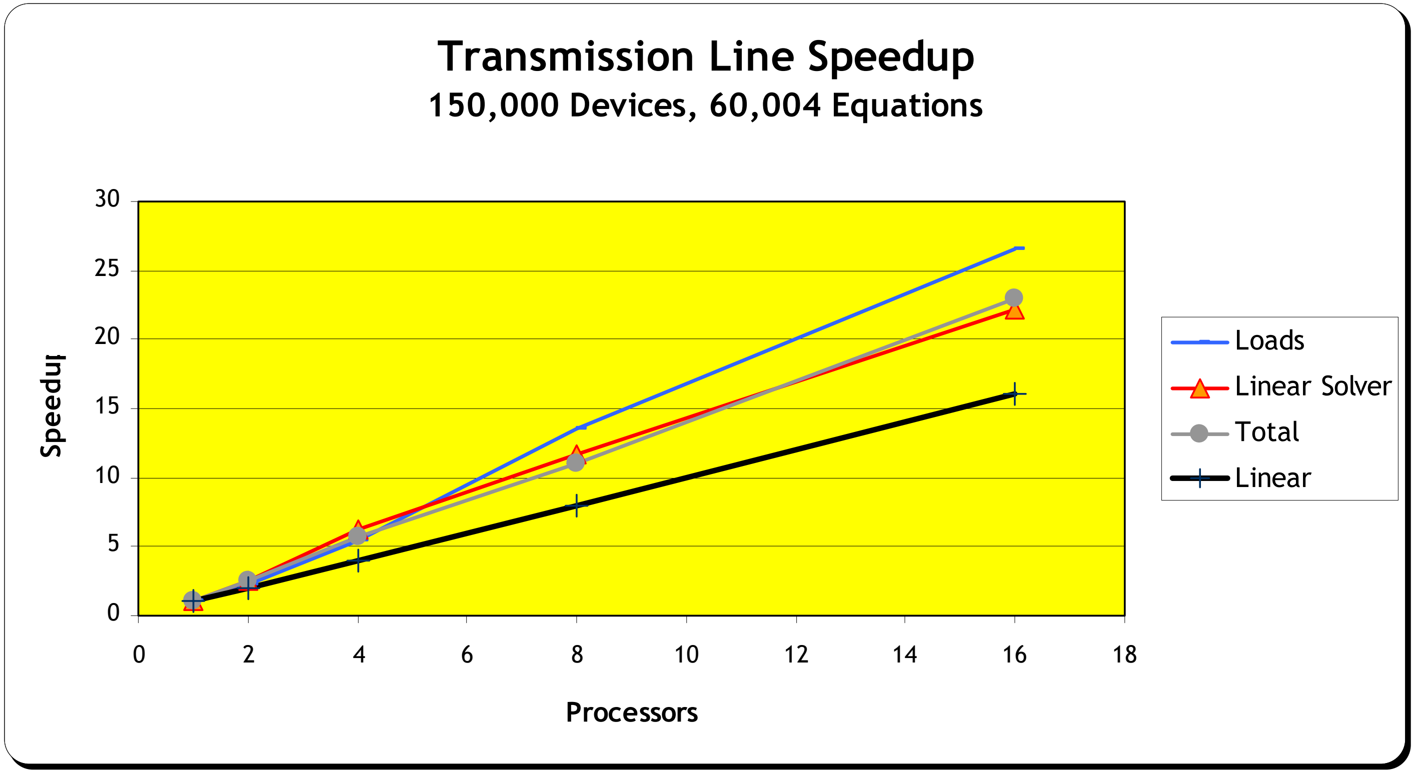
Original

→

	MIN	MAX
Unknowns:	1493	1508
Cuts:	2	7
Boundary:	2	5
Adj. Proc:	1	2

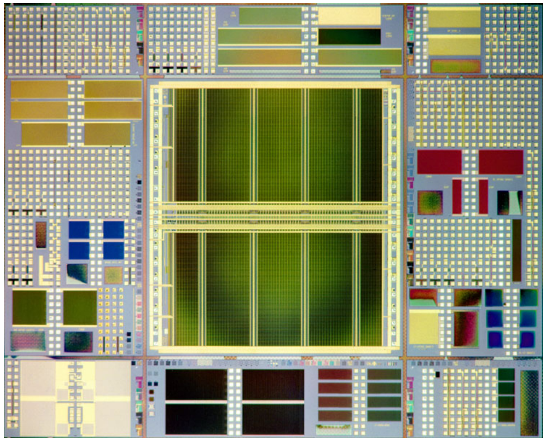
Repartitioned

Fixed Problem Size Parallel Scaling



Rad Hard Pentium Multiplier

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



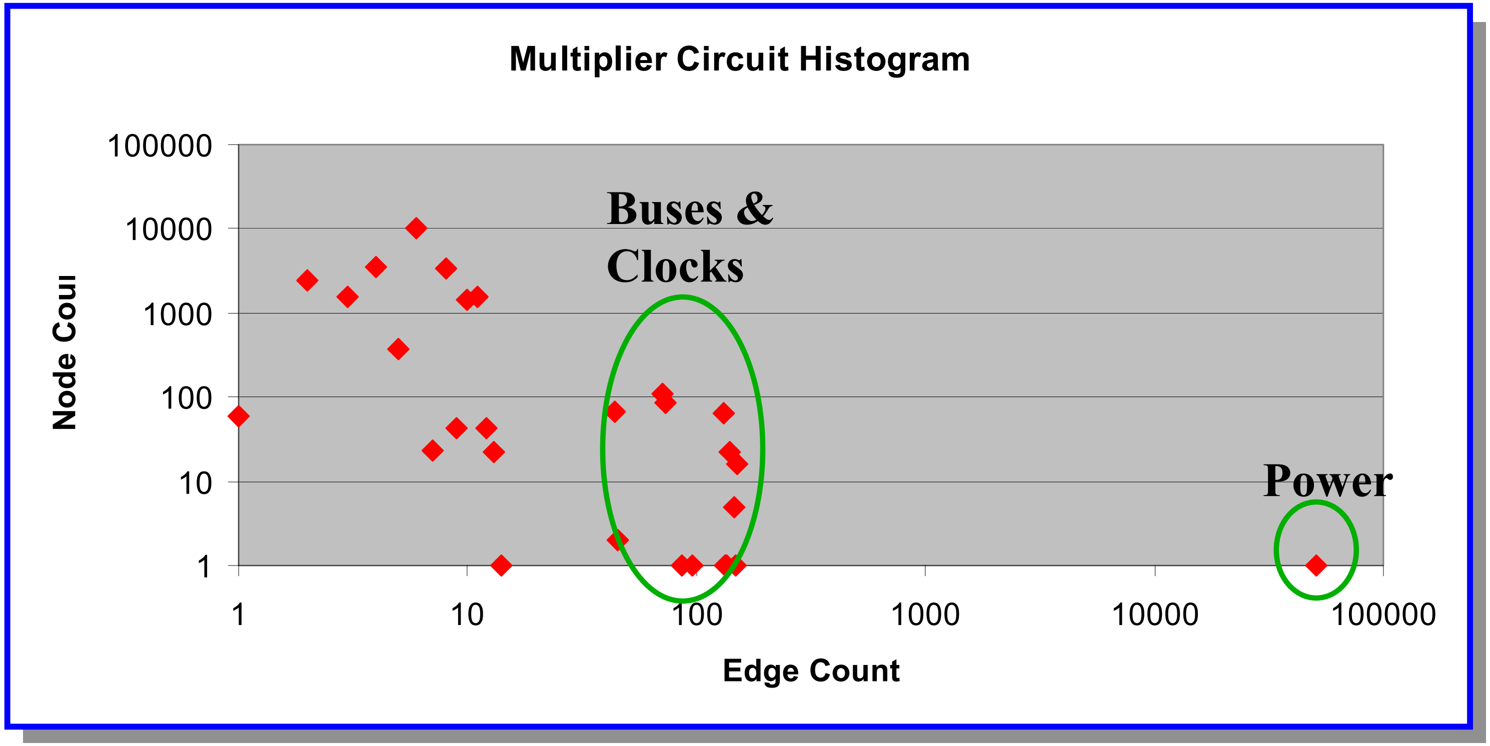
	MIN	MAX	SUM
Unknowns:	1555	1566	25187
Cuts:	9399	77235	259126
Boundary:	1508	1553	24697
Adj. Proc:	15	15	240

Original

	MIN	MAX	SUM	
Unknowns:	915	1995	25187	
Cuts:	2592	47253	150800	58%
Boundary:	844	1796	22653	
Adj. Proc:	10	15	222	92%

Repartitioned

Rad Hard Pentium Multiplier



- For Digital Circuits
  - Power node generates very dense row (~0.9*N)
  - Bus lines and clock paths generate order of magnitude increases in bandwidth

Distibuting Vsrc's

- Since Vsrc's act as independent BC's, distribute them across partitions to eliminate dependencies

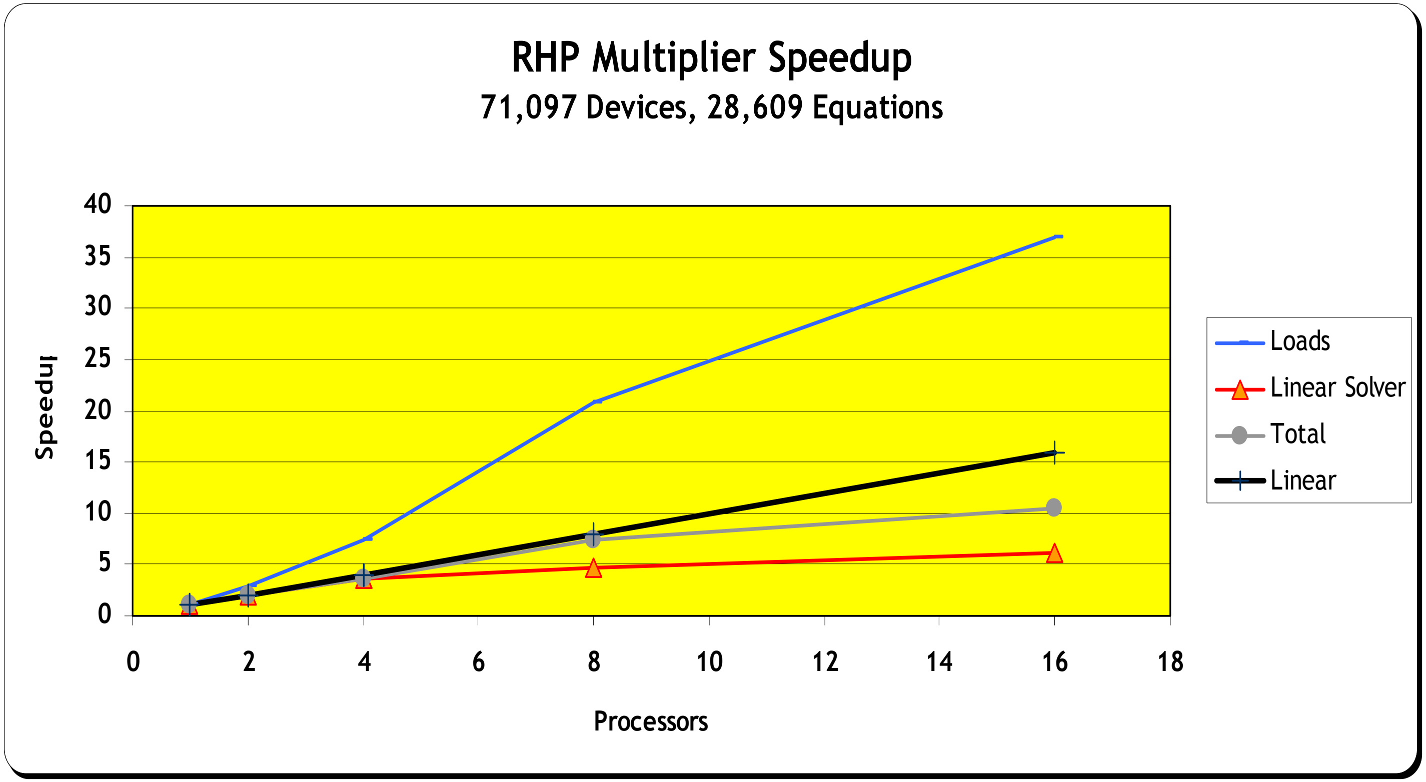
Parmetis – PartKway
Linear System

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

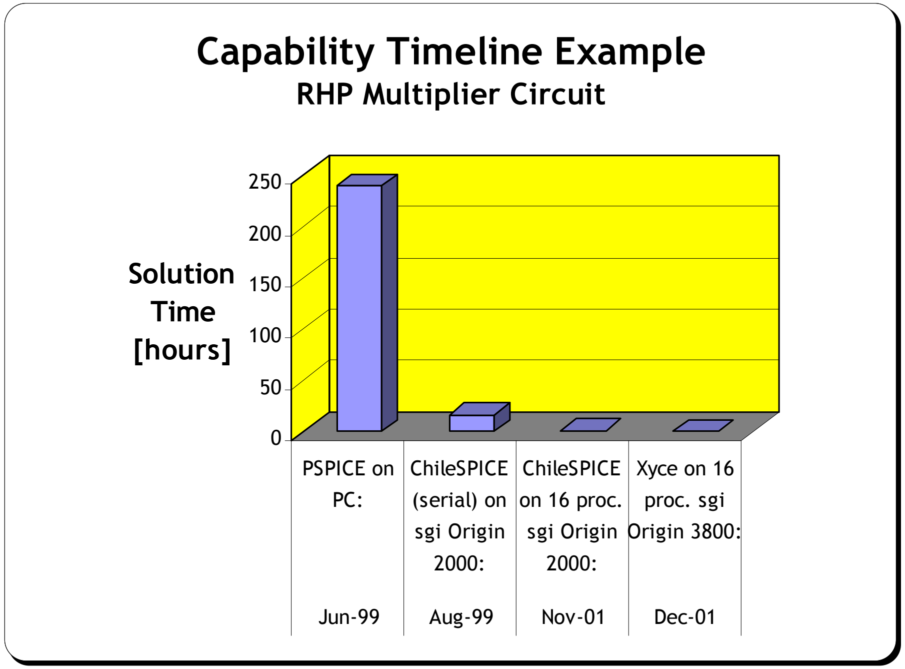
CHACO – Multilevel-KL
Circuit
25% imbalance
distrib. Ind. Vsrc's

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

Fixed Problem Size Parallel Scaling



Xyce Performance



- RHP Adder subckt on SGI Origin 3800, MIPS 400 MHz R12k Processors, 8 MB cache
- *NOTE: minimal optimization performed*

Program Phase	MFlops	Efficiency (800 MFlops max.)
Loads (Residual + Jacobian)	11.3	1.4 %
Linear Solve (Trilinos/Aztec)	96.6	12.1 %
Total Solution	32.9	4.1 %

Future Work

- Solution Algorithms
  - Waveform Relaxation
  - Trust-region Methods
  - Homotopy Methods
  - Algebraic Multi-Level\Schur Complement
- Partitioning Methods
  - Hypergraph
  - Multi-Constraint/Objective
- Coupling
  - Multi-Physics (Thermal, Radiation)
  - Multi-Fidelity (Software, Digital, Analog, PDE)

Summary

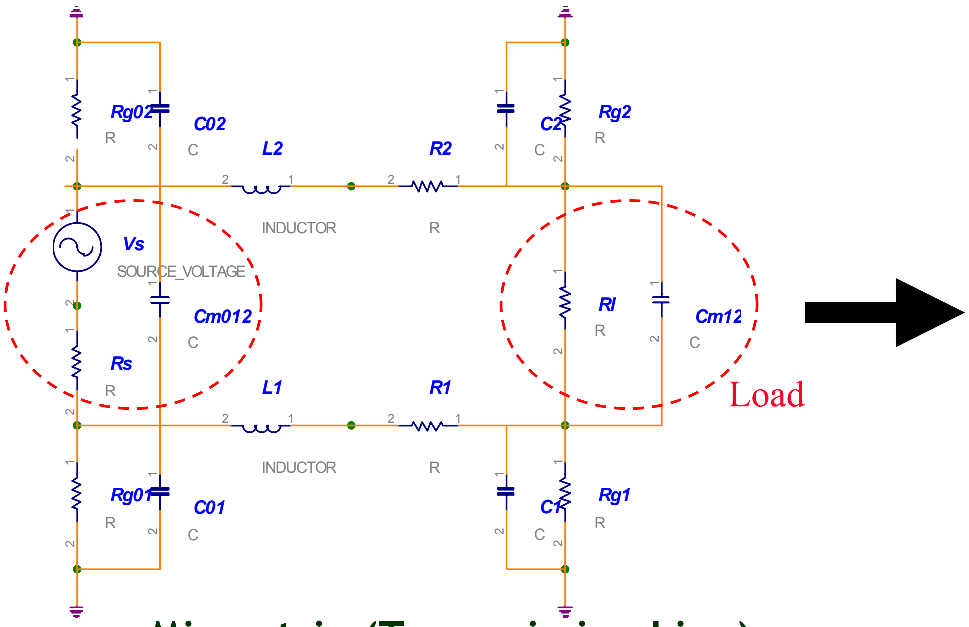
- ***A New Sandia Capability***
  - A scalable, parallel circuit code
  - Ability to run large-scale circuit problems
  - Contributing to Sandia’s many electrical-design communities
- ***Unique Challenges***
  - Circuit simulation presents problems not found in PDE-type codes
  - SNL specific needs: multi-fidelity/physics, etc.
- ***Technical Innovations***
  - Novel time-integration, nonlinear and linear solution methods
  - Partitioning heterogeneous problems
  - Test bed for algorithm development: solution and partitioning methods

Topology

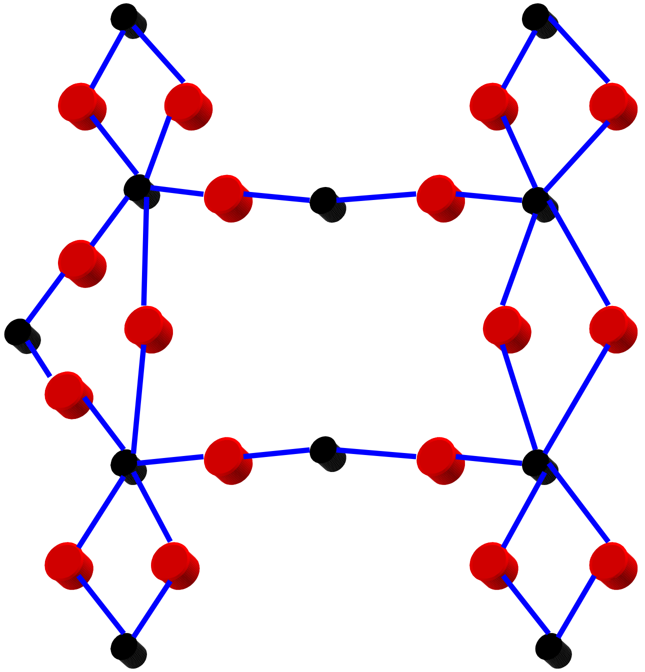
- Generalized Topological (Graph/Network) Description
  - Circuit
  - Linear System

🔴 Device or Sub-Circuit Node

● Voltage Node



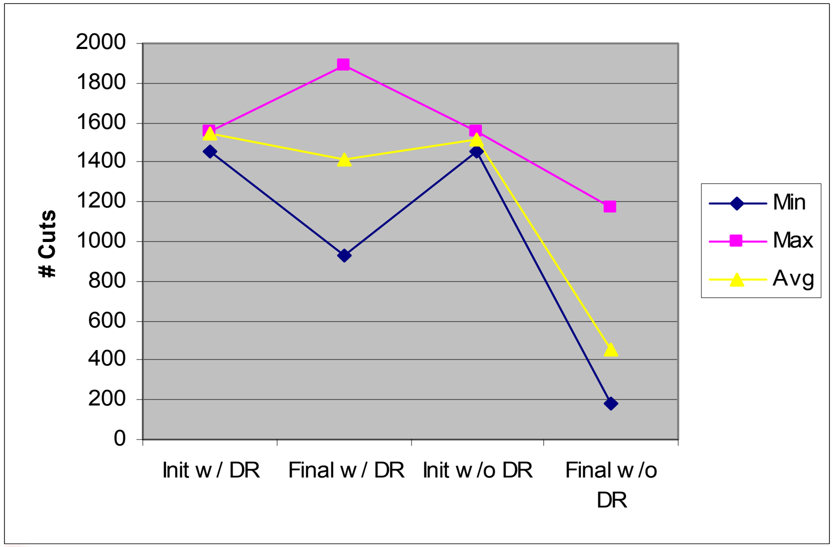
Microstrip (Transmission Line)



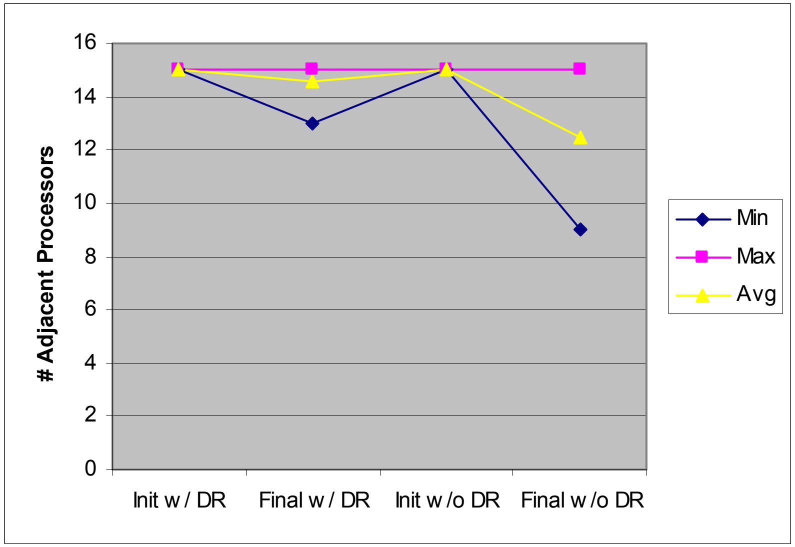
RHP Multiplier - Partitioning

- Extraction of dense row improves partitioning
  - Cut set average 1400 → 420
  - Adjacent processor average 14.5 → 12.3

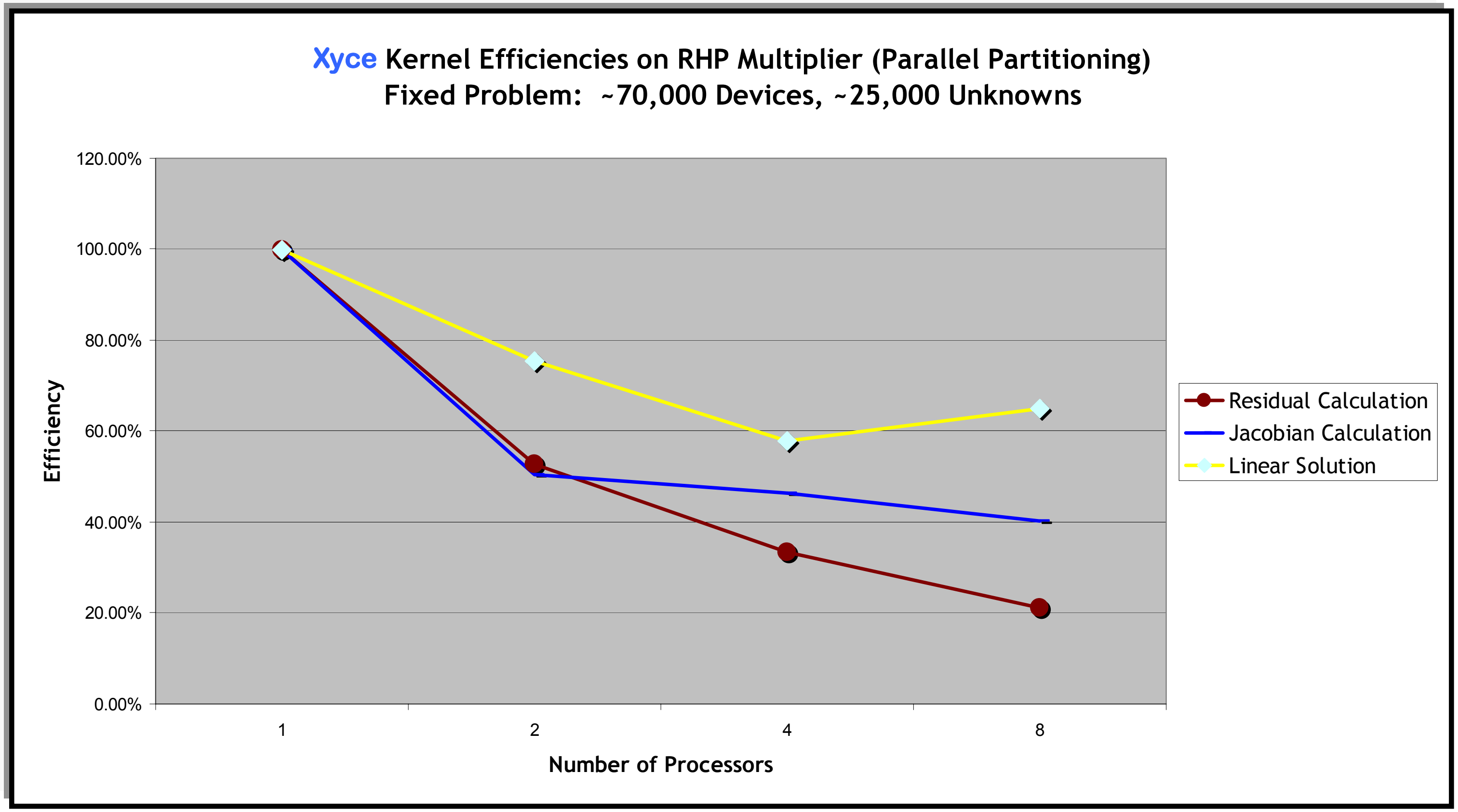
Edge Cuts



Adjacent Processors

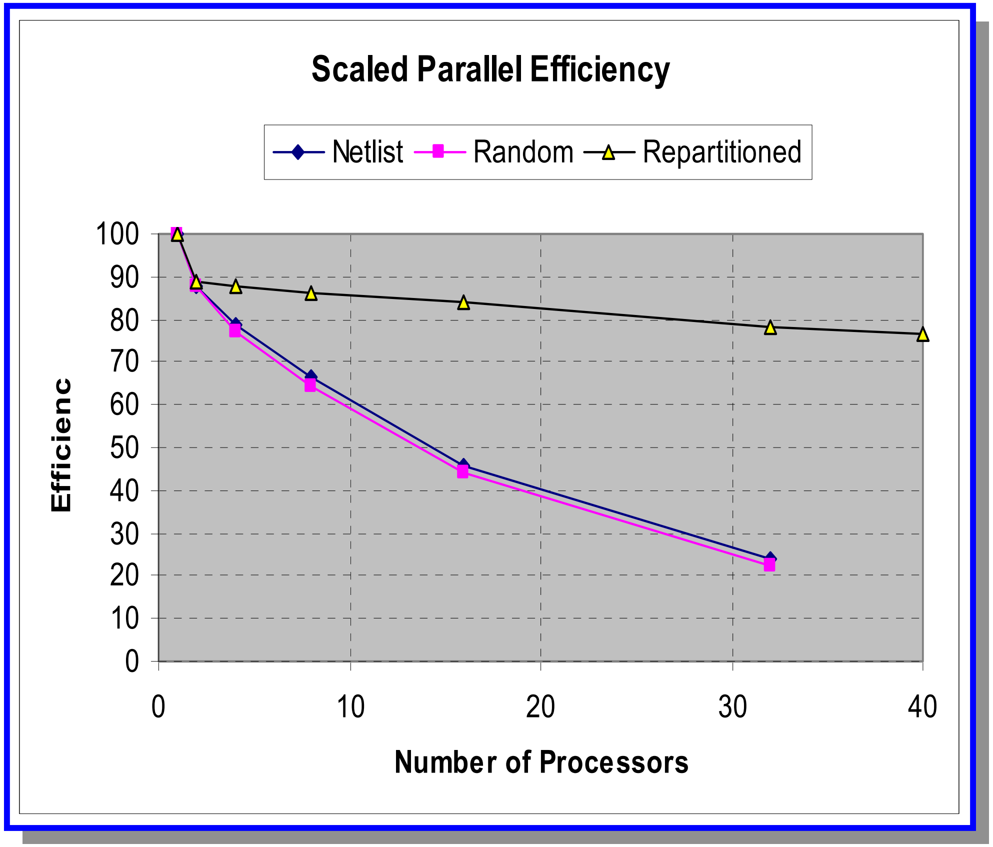


Rad Hard Pentium Multiplier



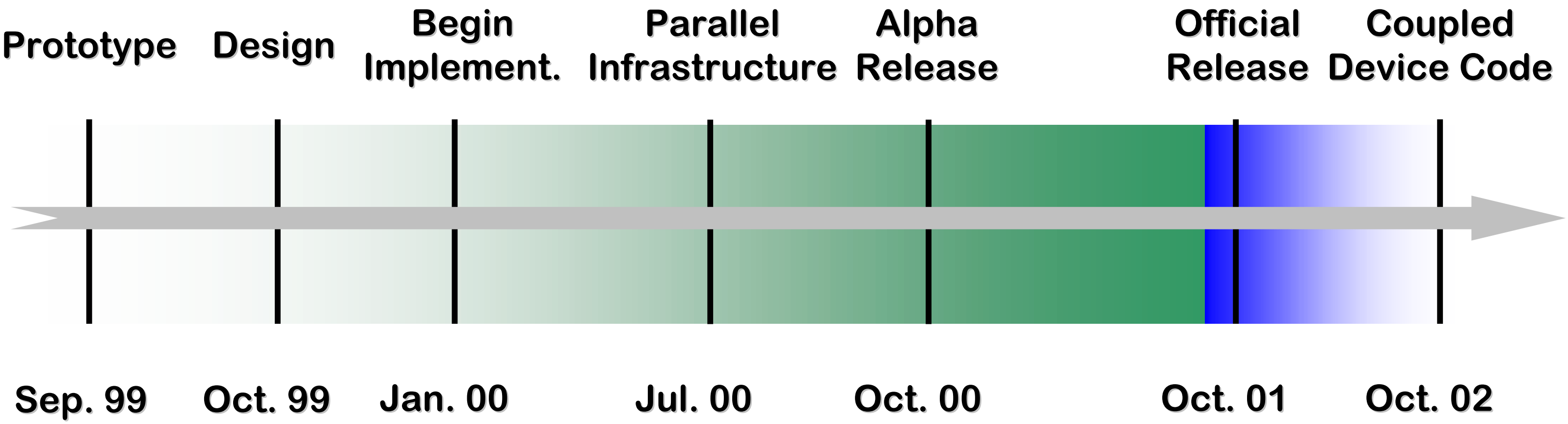
µStrip (Transmission Line) Scaling

- 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

Xyce™ Status



Overall Code Status:

•Currently able to run large, complex (e.g., RHP) circuits in parallel.

•Over 130k lines of C++ generated since January 00 (excluding libraries).

PDEs vs. Circuits

	Mesh Node	Voltage Node	MOSFET
DOF	<5	1	0-3
Edges	~1-10	1- >10,000	4
Flops (Load/Assembly)	1s-100s	0	>1,000

- **High Edge Count → Dense Rows**
- **High Cost Load Calculations → Load Imbalance**

- **CIRCUITS ARE NETWORKS RATHER THAN MESHES**

Weighting and Migration

LOAD

- Node Weighting
  - Device Node: Load Cost Function (Flops)
  - Voltage Node: Adjacent device costs due to “ghosting” cost
- Edge Weighting
  - Adjacent device cost
- Node/Device Migration
  - Node/Model/Device Blocks with “packing” facilities
  - Zoltan Migration Facility
- Algorithms
  - Chaco Multi-KL

SOLVE

- Initially implemented in Xyce
- Generalized support added to TRILINOS
- Node/Edge Weighting
  - Constant
  - Future: Preconditioner
- Algorithms
  - ParMETIS PartKway