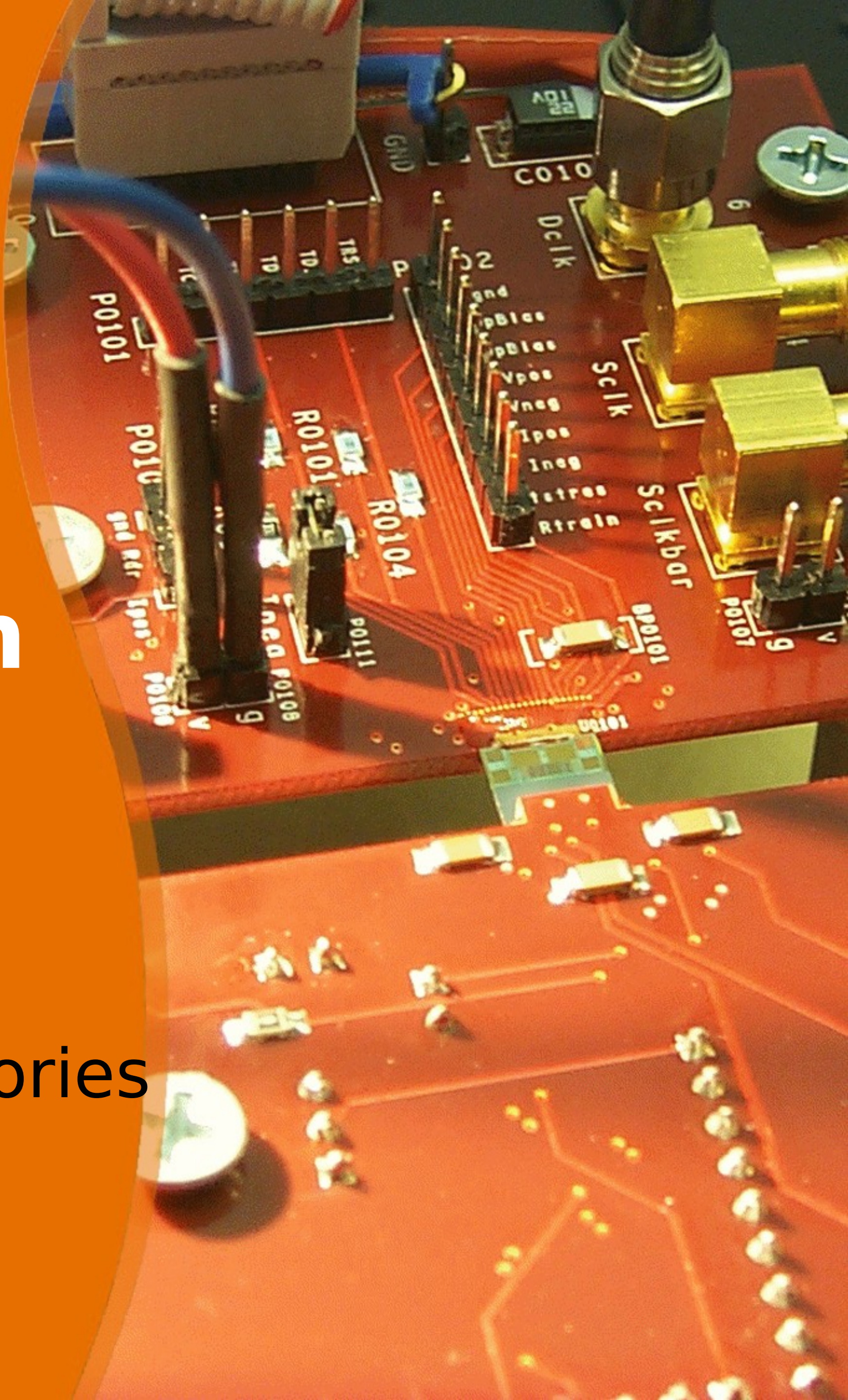
Sun microsystems

Multi-threading in Electric™, a Java VLSI CAD Tool

Gilda Garretón

Sun Microsystems Laboratories

JUG.cl May 2008

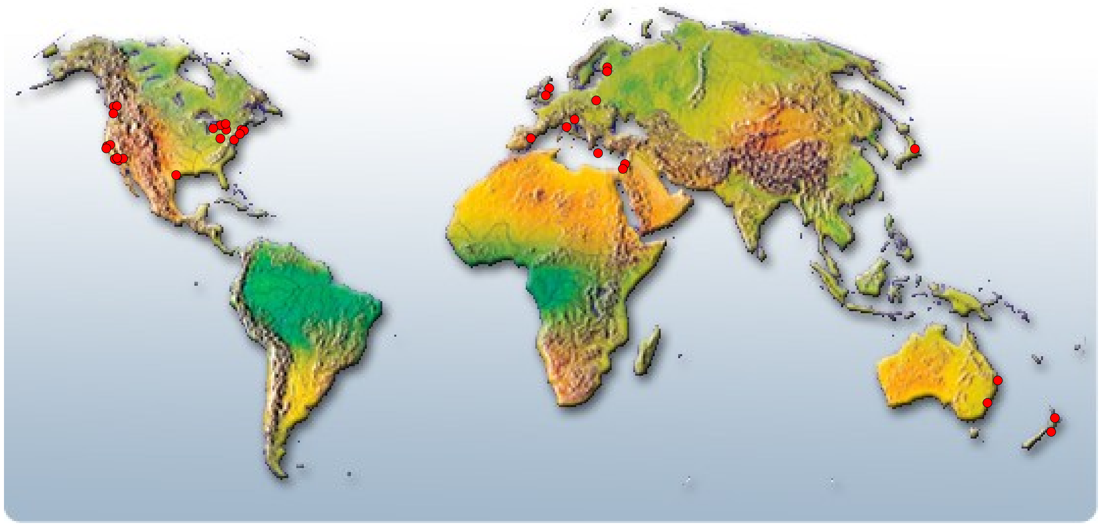


Agenda

- Who we are
- Research activities
- Electric, a Java VLSI framework
- Multi-Threading 2 projects
- Lessons learned

Sun Microsystems Laboratories

- Center for innovation
- Mission
- Research Strategy
- Collaboration



Our Technology Changes the World

Developed in Sun Labs,
transferred to Sun products and to the outside world

Sun Labs Project Focus

System Hardware

System Software

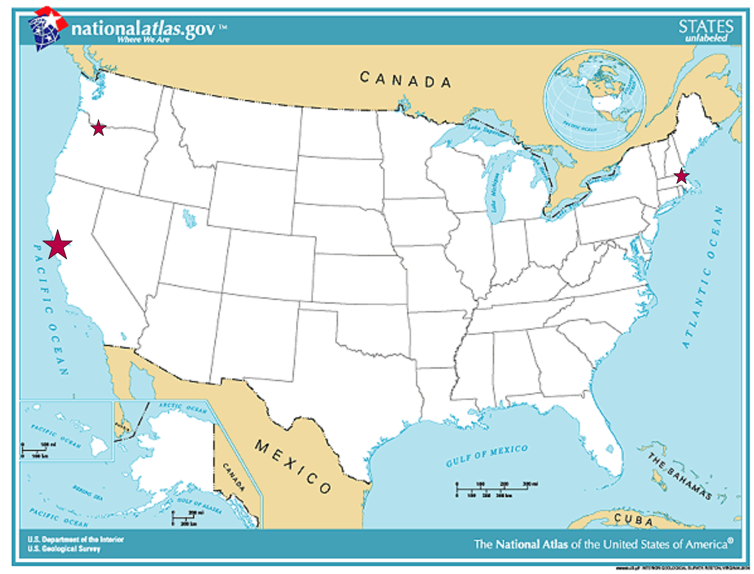
System Science

Network Clients

VLSI Research Group

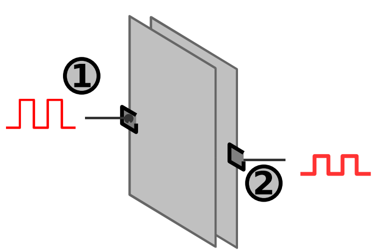
- Diverse group (~20)
  - > Hardware (14)
  - > Software (4)
  - > Both (2)
- Education
  - > BS, MS, PhD
- Rank
  - > 1 PE, 4 DEs and 1 Sun Fellow
- Collaboration aims innovation
  - > Research: circuit (HW) and CAD (SW)

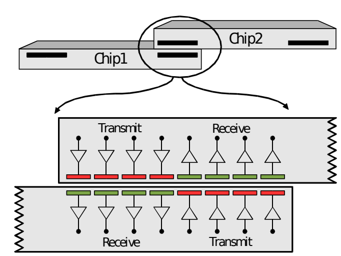
Moscow



VLSI Circuit Research

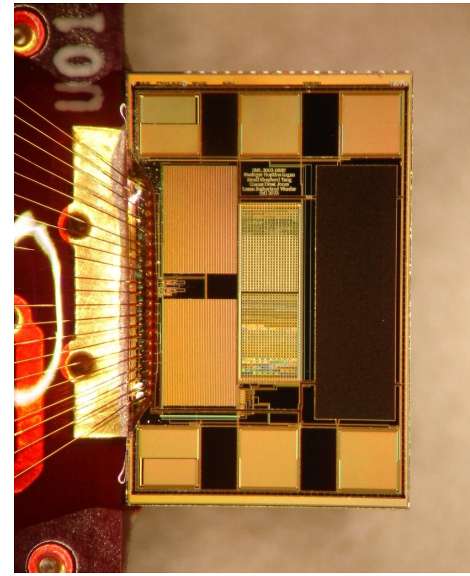
- Design circuits to enable novel architectures
- Low-power circuits
- High-speed circuits
- Asynchronous circuits
- Communication Links
  - > Proximity Communication



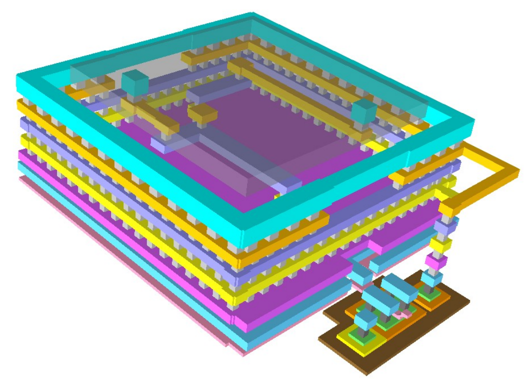


VLSI CAD Research

- Challenges
  - > Size and performance
  - > Hierarchical representation
  - > Shrinking technologies
- Research topics
  - > Handling of large data set
  - > Geometric algorithms
  - > Visualization and collaboration
- Software Research
  - > Client/server model
  - > 2D/3D visualization
  - > **Multi-Threading**



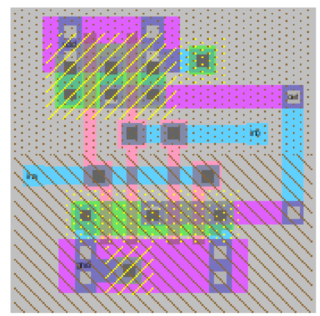
Collaboration with Harvey Mudd College

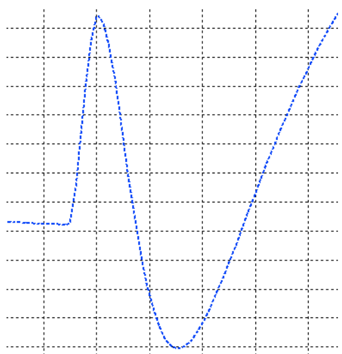


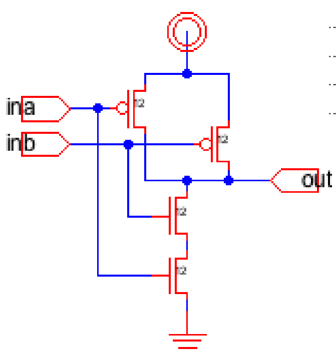
Metal Cage Structure

Making Test Chips: Custom VLSI Chip Design Flow

Chip Building Process

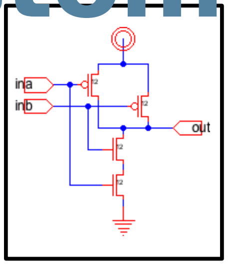


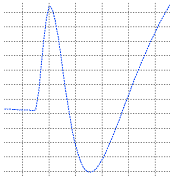


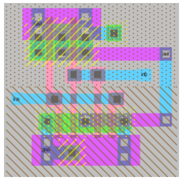


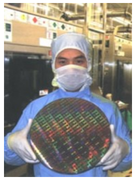
CAD Testing Process

Making Test Chips: Custom VLSI Chip Design Flow



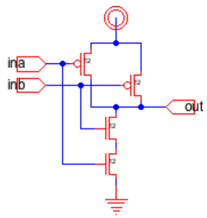


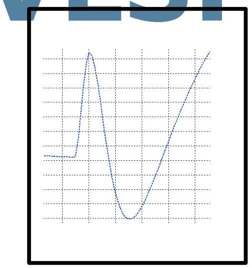


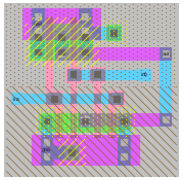


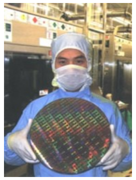
1. **Schematic Entry and Transistor Sizing**
   > Logical Effort support

Making Test Chips:
Custom VLSI Chip Design Flow



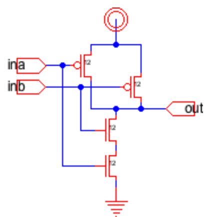


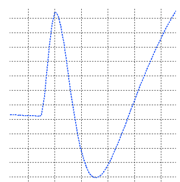


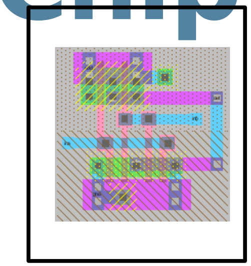


1. Schematic Entry and Transistor Sizing
2. **Simulation**
   > Interface to CAD Tools of Our Choice
   > Waveform Viewer and Cross-Probing

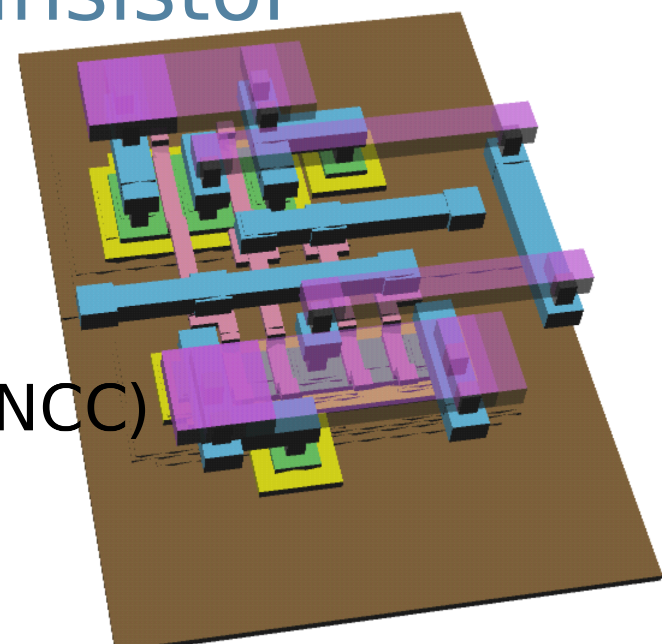
Making Test Chips: Custom VLSI Chip Design Flow



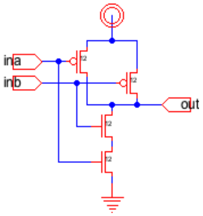


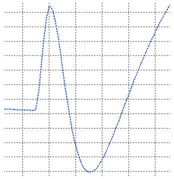
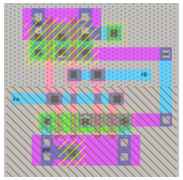
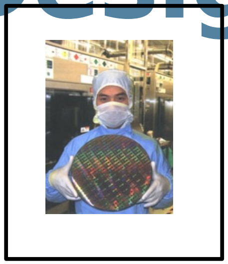


1. Schematic Entry and Transistor Sizing
2. Simulation
3. **Layout**
   > Network Consistency Checker (NCC)
   > Design Rules Checker (DRC)
   > Automatic fill generator
   > Routing, auto stitching tool

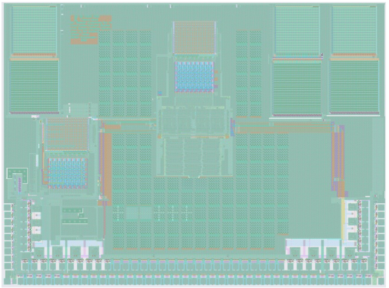
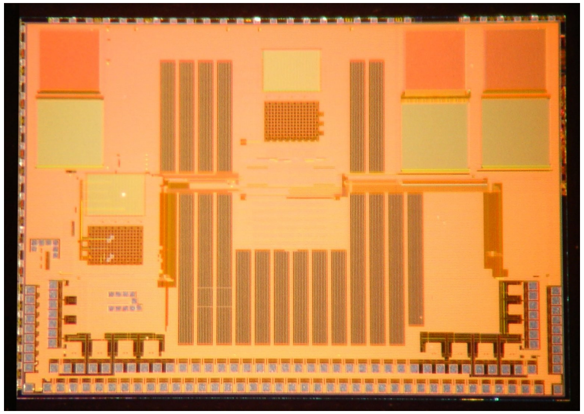


Making Test Chips: Custom VLSI Chip Design Flow



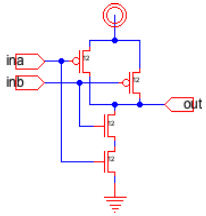

1. Schematic Entry and Transistor Sizing
2. Simulation
3. Layout
4. **Fabrication**

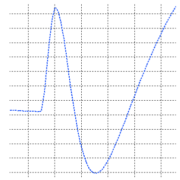



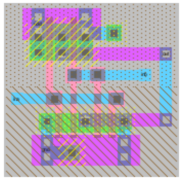
Foundry Input

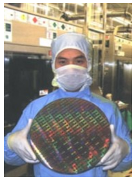
Foundry Output

Making Test Chips: Custom VLSI Chip Design Flow

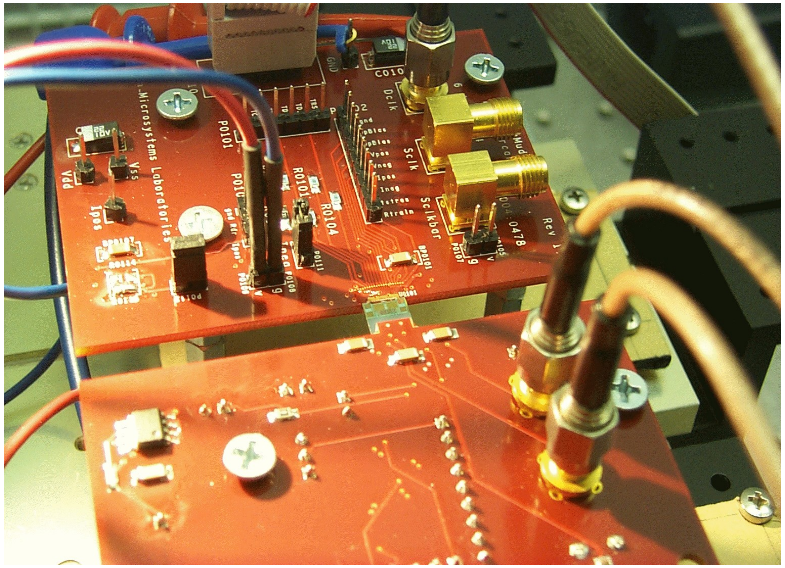








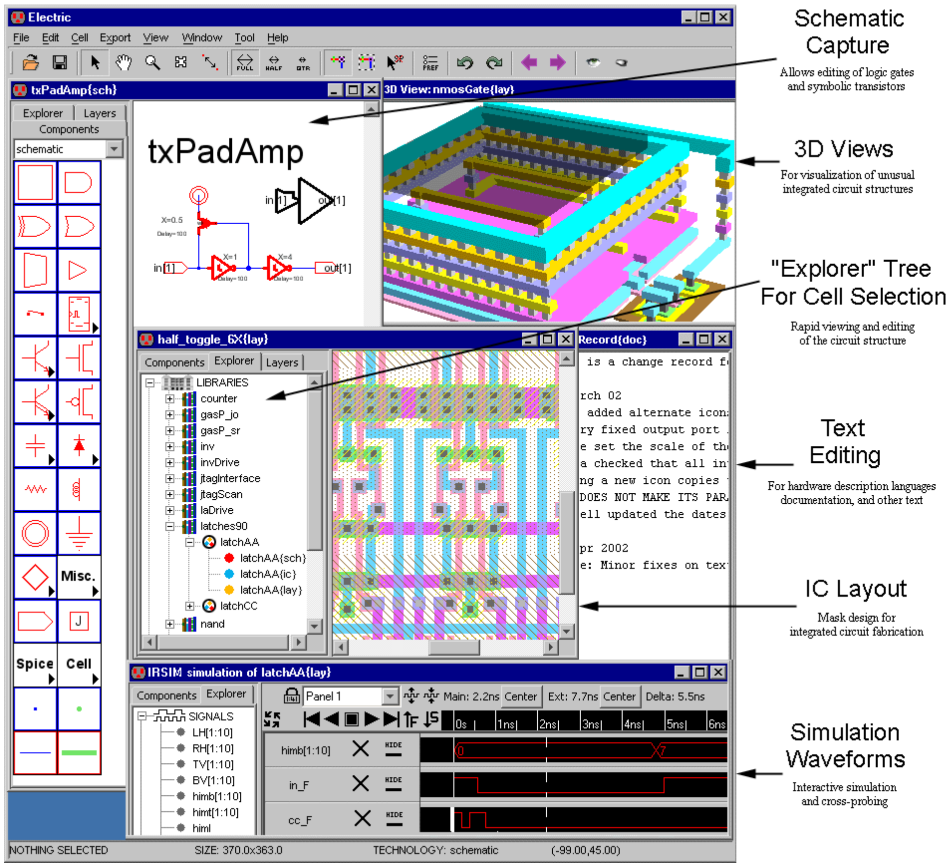
1. Schematic Entry and Transistor Sizing
2. Simulation
3. Layout
4. Fabrication
5. **Testing**



PxIO Testing Board

Electric VLSI Design System

- **Java** development platform
- Technical benefits for HW/SW
- External/Open source community
  - > www.staticfreesoft.com
  - > www.opensparc.net/
  - > Google VLSI forum
- Dynamic development cycle



Electric Framework

Multi-Threading in Electric

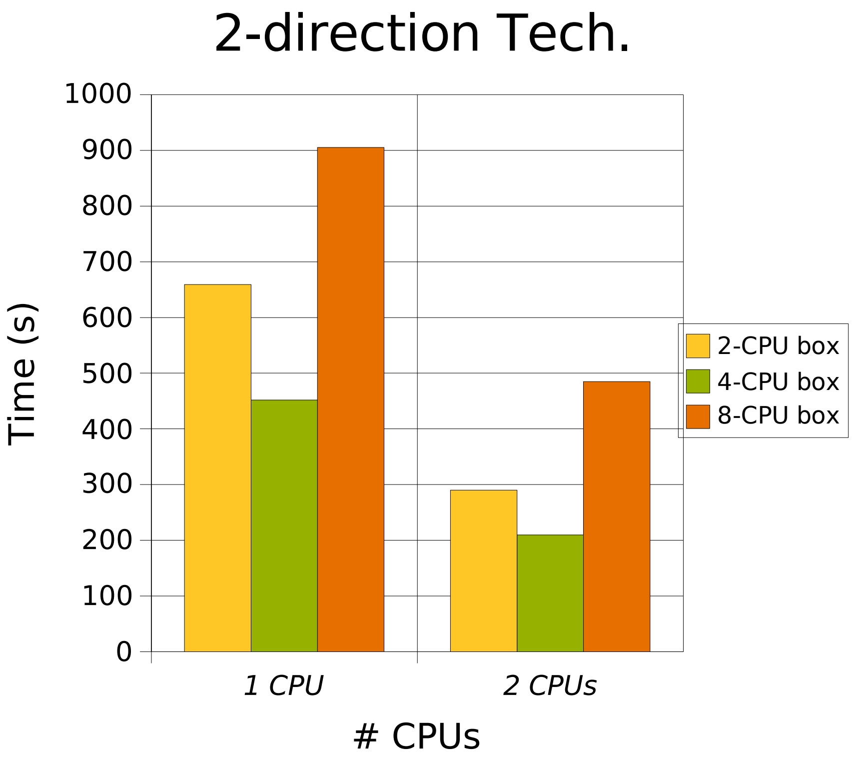
- Used in schematics/layout phases
  - Analysis tools
  - Route and placement
- 2 Multi-threading projects
  - Using directly JDK threading classes: routing tool
  - Using threading capabilities in Electric: Integrated solution: DRC tool
- Available in next GNU release: June 2008

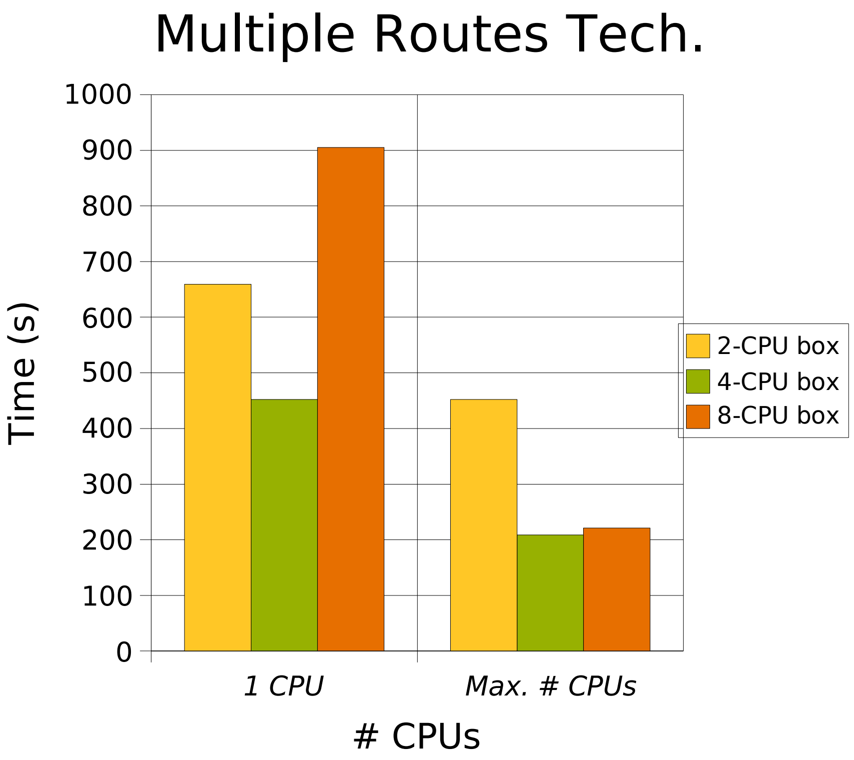
Multi-Threading a Routing Tool

- Routing using cost functions
  - > Based on Dijkstra search
  - > Two parallel techniques
    - Tech 1: two directions (A<->B)
    - Tech 2: multiple routes
- Implementation
  - > Runtime.getRuntime().availableProcessors()
  - > Class RouteInThread extends Thread
  - > Class Semaphore
    - acquireUninterruptibly()
    - release()

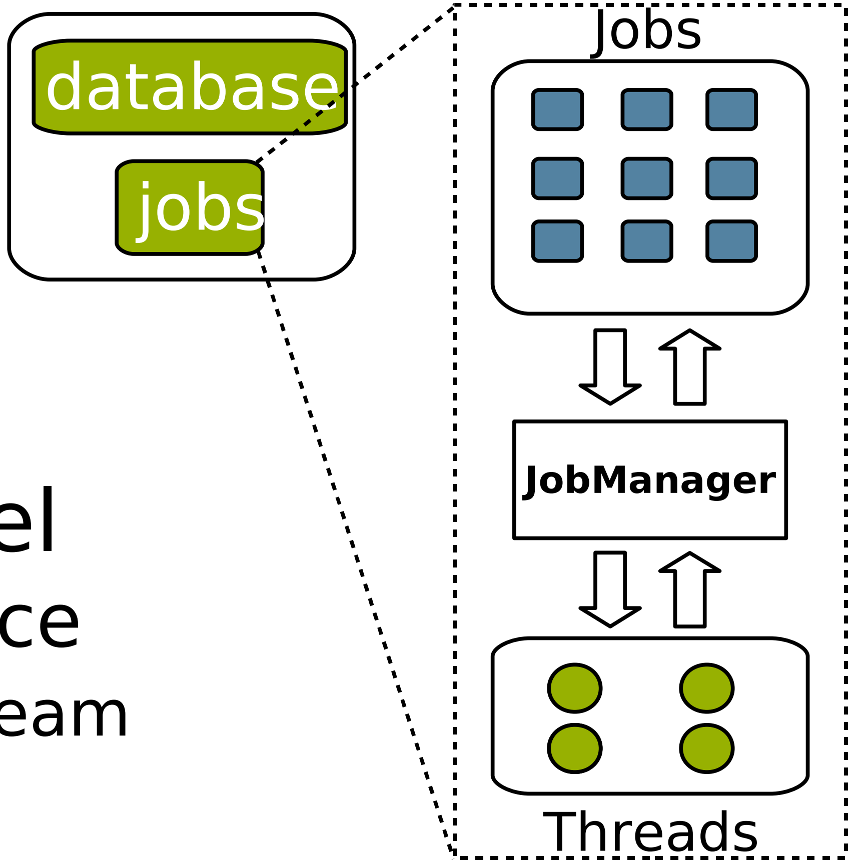
Multi-Threaded Routing Results

Example: 682 segments to route





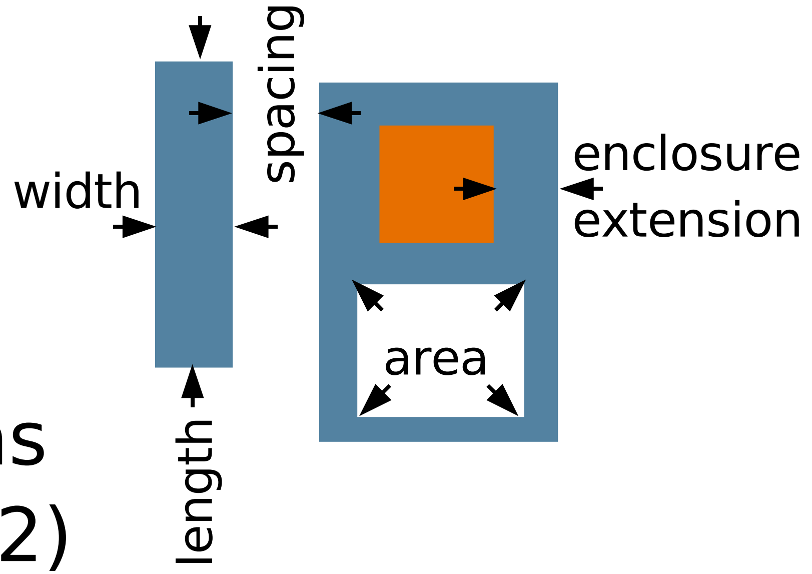
Multi-Threading a DRC Tool

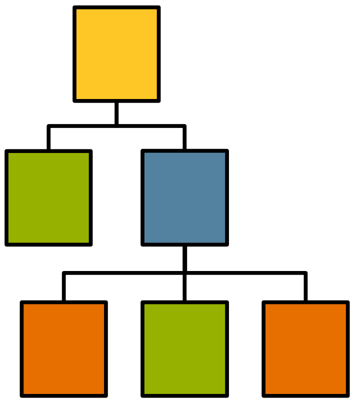


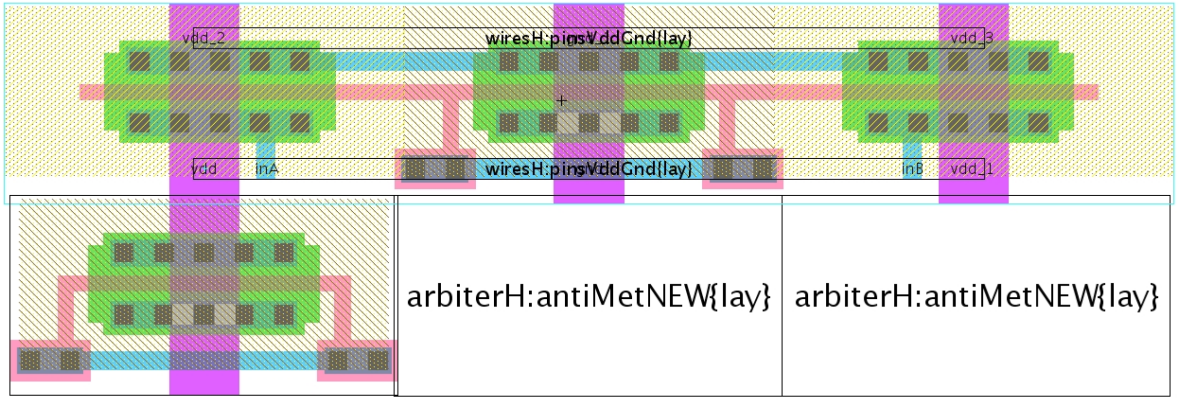
- Electric's core
- Jobs Manager
  - > Pools of jobs
  - > Java threads
- Client/server model
  - > Serializable interface
    - ByteArrayOutputStream
    - ObjectInputStream
- Class MultiTaskJob<Key,Task,Result> extends Job

DRC Introduction

- Verify correctness
  - > Recommended rules
  - > Geometrical conditions
    - w(l1), f(l1, l1), g(l1,l2)
- Single-Threaded (STD) algorithms

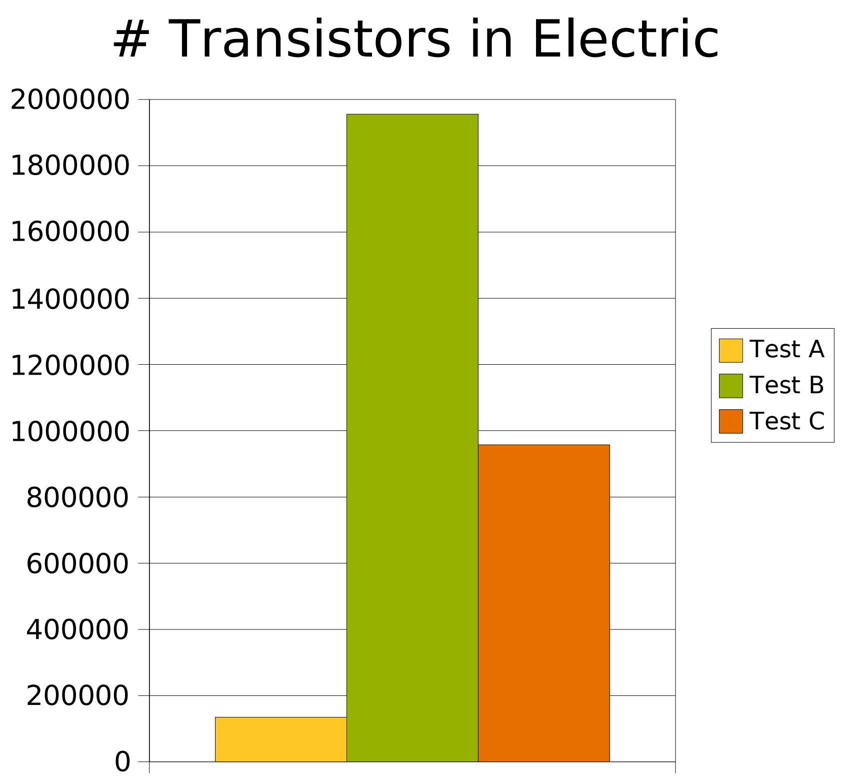






Single-Threaded DRC Results

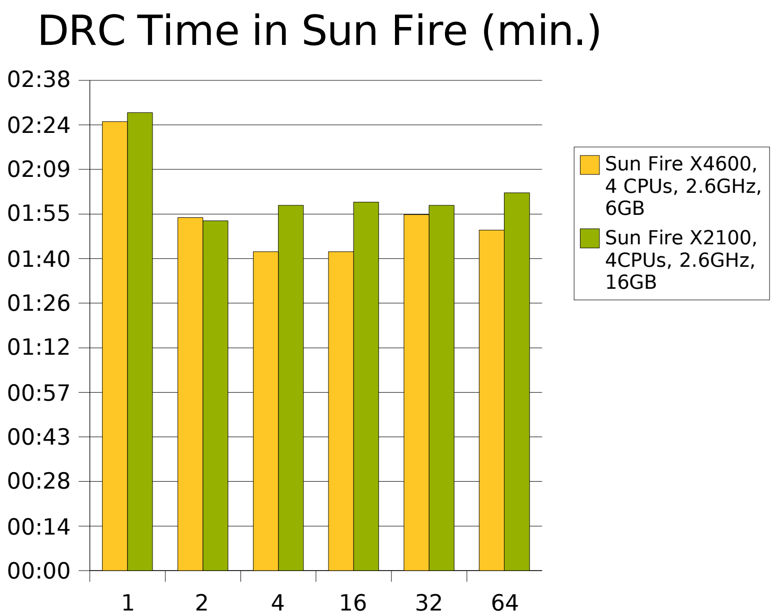
Time consuming in layouts

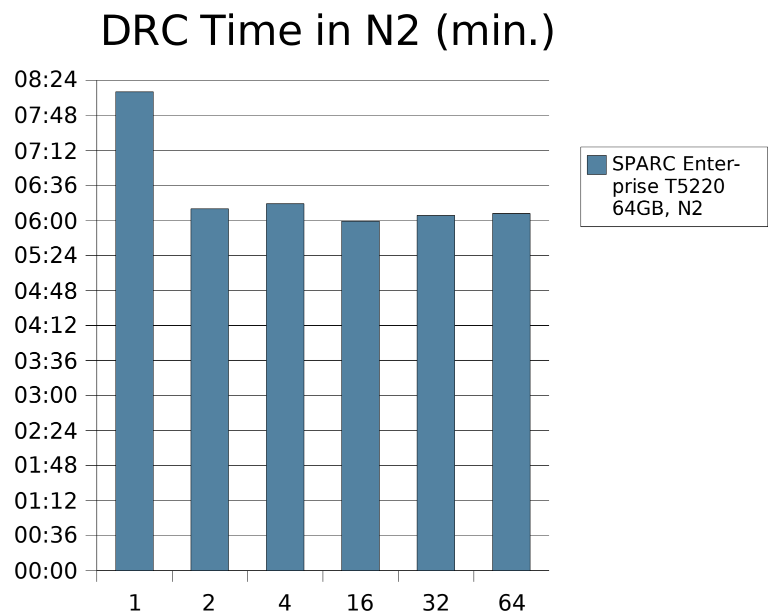


* AMD64, 2.8GHZ, 16GB

Multi-Threaded DRC Results

- Simplest approach due to STD code
  - Less than 1 week to implement
- Best results in N2 with 16T and in Sun Fire with 4T





Lessons Learned (1/2)

- Applications will require changes to utilize multi-threading
  - Changes need not be significant
  - More than one way to multi-thread
- Porting to MTD Java is feasible
  - Application-specific multi-threading strategies
- Software infrastructure matters!
  - Multi-threading using existing job mechanism
  - Profiling tools understand results
    - SunStudio Collect/Analyzer

Lessons Learned (2/2)

- MTD Routing Tool
  - > Speedup of over 4X on an 8-CPU machine
- MTD DRC Tool
  - > Results per layer
    - Up to 41% in Sun Fire X4600
    - Up to 37% in N2 with 16 threads
  - > Future work: geometry partition

Questions?

Gilda Garretón
Sun Microsystems Laboratories

JUG.cl May 2008