Calibrating Achievable Design

Jason Cong, Wayne Dai, Andrew B. Kahng, Kurt Keutzer, and Wojciech Maly
Carnegie-Mellon University, UC Los Angeles, UC Santa Cruz, and UC Berkeley

Achievable Design

- Technology: dynamic CMOS, Cu/low-k, SOI, vertical devices, ...
- Quality: power, speed, area, reliability, cost/yield, ...
- This Theme: Enable the understanding of achievable design, as well as more efficient and effective development of design technology
**CAD Life Cycle Questions**

- What does the design problem look like?
- How can I quickly develop the right design tool?
- Did I really solve the problem?
  - Did design process improve?
  - Did achievable design envelope get bigger?

**Technology Extrapolation**

- What does the design problem look like?
- Need fundamental facts and data points to anchor the process of bounding the achievable envelope of design
- Can be with respect to:
  - manufacturing process, materials, physical phenomena
  - specific CAD optimizations of circuit topology/embedding
  - system architecture and packaging
- Are properly extrapolated via:
  - "inference chains"
  - response surface modeling and parameter optimization
- Drive the EDA vision of future design issues, methodology
  - fundamental limits, fundamental truths, stakes in the ground
**Example Technology Extrapolation Questions**

- What is the maximum possible clock frequency for a given process and die size?
- When does inductance matter?
- What design tradeoffs must be made to maintain reasonable supply currents?
- What is the necessary number of package pins/balls for power/ground distribution?
- At what geometries, supply voltages will domino lose most advantages over static CMOS?
- What is an optimum design strategy from a manufacturing cost point of view?

**Time-to-Market and QOR in CAD**

- How can I quickly develop the right design tool?
- Problem: Currently takes 5-7 years to get a leading-edge algorithm into production tools
  - Result: Must solve today’s design problems with yesterday’s CAD technology
- Problem: Published descriptions insufficient to enable replication or even comparison of algorithms
  - Result: Cannot identify, evaluate or advance the CAD technology leading edge
- Issues for the entire CAD field
  - productivity of CAD tool development (time-to-market)
  - quality of the resulting CAD tools (quality-of-result)
- Our Solution: Reuse ++
**CAD-IP Reuse**

- Componentized IPs for CAD ≡ plug-and-play solvers for key problems
  - “IP socket” ≡ problem formulation + data interfaces + test data + QOR measures
- Addresses the time-to-market problem
  - CAD-IPs plug-and-play into alternative design flows
  - reduce delays from invention to evaluation to availability
  - → shorter design cycle for new tools, rapid prototyping of new flows
- Addresses the QOR problem
  - focus on the “right problems” (+ mechanism to specify these)
  - reduced barriers to entry for leading-edge research
  - standards for evaluation, reporting, documentation
  → more mature framework for developing new CAD-IP
- New GSRC infrastructure for CAD-IP Reuse: The Bookshelf

**Metrics**

- Did I really solve the problem?
- Design optimization must be founded on
  - an understanding of what should be optimized by which heuristic
  - an understanding of design as a process
- “Metrics” supports ideal of “measure, then improve”
  - design becomes less of an art and more of a formal discipline
  - design process optimization enabled through framework of recording, mining, measuring, diagnosing, and then improving
- Infrastructure
  - data mining / visualization / diagnosis infrastructure
  - project-specific design process data collection
**CAD Life Cycle Answers**

- What does the design problem look like?
  - GTX: GSRC Technology Extrapolation

- How can I quickly develop the right design tool?
  - CAD-IP Reuse via the GSRC Bookshelf

- Did I really solve the problem?
  - Metrics of the design process

---

**Synergies**

Feasibility / sanity checkers to embed within a tool flow

- GTX
  - Which problems are critical?
  - What will instances look like?

- CAD-IP Reuse
  - Estimates of best-optimized design, optimal tradeoffs

- Metrics
  - Models, measures of algorithmic activity

Optimized design processes, calibration data for modeling CAD optimization

Objective functions, tool QOR metrics
Overview of Session

- GTX: The GSRC Technology Extrapolation System
  ▲ Dirk Stroobandt

- Future Axes for Achievable Design
  ▲ Wayne Dai

- CAD-IP Reuse via the GSRC Bookshelf
  ▲ Igor Markov

- A Metrics System for Continuous Improvement of Design Technology
  ▲ Andrew B. Kahng

A. Caldwell, A. B. Kahng, F. Koushanfar,
H. Lu, I. Markov, M. Oliver and
D. Stroobandt
University of California, Los Angeles
Overview

- Introduction
- Previous Roadmapping Efforts
- Overview of GTX: Goal and Structure
- Fundamental Features
- Demonstration
- Example Use Scenarios:
  - Roadmap emulation, development, and maintenance
  - Roadmap evaluation and comparison

Introduction: Technology Extrapolation

- Evaluates impact of
  - design technology
  - process technology
- Evaluates impact on
  - achievable design
  - associated design problems
- Questions to be addressed
- Sets new requirements for CAD tools and methodologies
- Roadmaps: familiar and influential example
Introduction: Roadmapping

- Roadmapping efforts drive development of design technology:
  - System architects, designers, CAD managers use roadmaps to determine
    - tough problems
    - risks, ...
  - EDA suppliers use roadmaps to determine
    - R&D investment
    - product pipeline
  - Government and consortia use roadmaps to determine levels of investment
- Roadmaps serve as a guide to the most promising directions, the most critical problems

Roadmap Process and Its Implications

Basic Technological Assumptions

Basic Methodological Assumptions

Models and Discussion

Implications to the Community

Translation to Specific Research Agendas

Research Proposed to Implement Agenda

“Timing closure is a hard problem and will only get harder”

“We will fund research on timing-aware partitioning”
**Roadmap Process**

Basic Technological Assumptions → Models and Discussion → Implications to the Community → Models and Discussion

- **Basic Methodological Assumptions**
- **Implications to the Community**
  - "Timing closure is a hard problem and will only get harder"
  - "Here's how my work is critical for addressing your problem"
  - "I can make a breakthrough in technology or methodology"

New models

R. Newton, ICCAD99 panel

---

**Introduction: Roadmapping**

- **Difficulties of roadmapping**
  - No crystal ball
  - New technologies, new circuit techniques, new design methodologies and tools
  - Always difficult to predict achievable design, especially in the future

- **Roadmaps rely on**
  - Models: technology projections, design attributes, design tools
  - Calibrations: measurements of technology and design parameters
Roadmapping in the past

◆ Previous and ongoing efforts
  ▲ ITRS Roadmaps
  ▲ Tools: SUSPENS, GENESYS, RIPE, BACPAC, ...
  ▲ Numerous tools in industry

◆ Observations
  ▲ Predict “same” parameters but with different assumptions, inputs
  ▲ Lack of documentation and visibility into internal calculations
  ▲ Single inference chain for a given output (hard-coded modeling)
  ▲ Inflexible: user cannot define studies of other, related parameters
  ▲ Near-total duplication of effort
  ▲ Missing: models of CAD tools and optimizations (what is really “achievable”?)
  ▲ Missing: scope, comprehensive coverage

Questions To Ask About Roadmaps

◆ How do different roadmap predictions compare?
◆ How to evaluate underlying models? (sanity checks)
◆ How do we reuse and extend models to encompass new aspects of technology, new axes of achievable design?
◆ What is the impact of modeling choices on predictions?

Need a new infrastructure, new concept!
**Previous Systems Versus Ideal System**

- Same parameters but different assumptions
- Inflexible, not easy to add other studies
- Hard-coded, no easy changes
- No internal visibility
- Duplication of effort

**Goals of A New Technology Extrapolation System**

- **Flexibility**
  - Interactively edit chains of relations between parameters
  - Define new parameters and relations between them
  - Perform specific studies (but different studies at different times)
- **Quality**
  - Continuous improvements
  - World-wide participation of experts
- **Transparency**
- **Prevention of redundant effort**
Goals of New Technology Extrapolation (cont.)

- Flexibility
- Quality
- Transparency
  - Open-source mechanism
  - Models are visible to the user
- Prevention of redundant effort
  - Permanent repository of first choice
  - Adoptability and maintainability

GTX: GSRC Technology Extrapolation System

- GTX is set up as a framework for technology extrapolation

  - Flexibility, quality, visibility allow a “living roadmap”:
    - emulate existing roadmap (modeling) efforts
    - develop new roadmaps (models)
    - evaluate roadmaps (models)
    - compare roadmaps (models) to each other
**GTX Structure**

- **Knowledge representation:**
  - **Parameters** (description of technology, circuit and design attributes)
  - **Rules** (methods to derive unknown parameters from known ones):
    - ▼ closed-form models
    - ▼ executable algorithm implementations
    - ▼ table-lookups
  - **Rule chains** (serialized user-defined rules)
    - ▼ interactive specification and comparison of alternative modeling choices

- **Implementation**
  - ▼ Execution by a derivation engine to perform studies
  - ▼ Embedding into GUI for ease of use, interactivity, display of results

- See poster for details of GTX framework

---

**Knowledge Representation**

- **Rules and parameters are specified separately from the derivation engine**

- **Human-readable ASCII grammar**

```plaintext
#parameter dl_chip #rule BACPAC_dl_chip
#type double #description
#units {m} #output
double {m} dl_chip;
#default #inputs
de1e-2 double {m^2} dA_chip;
#description #body sqrt(dA_chip)
chip side length #reference #reference
#endparameter #endrule
```
**Knowledge Representation (cont.)**

- Rules and parameters are specified separately from the derivation engine
- Human-readable ASCII grammar
- Benefits:
  - Easy creation and sharing of parameters / rules by multiple users
    - D. Sylvester and K. Cao: device and power modules that “drop in” to GTX
  - Extensible to models of arbitrary complexity (specialized prediction methods, technology data sets, optimization engines)
    - Avant! Apollo or Cadence SE P&R tool: just another wirelength estimator
  - Applies to any domain of work in semiconductors, VLSI CAD
    - Transistor sizing, single wire optimizations, system-level wiring predictions,…

---

**Parameter and RuleNaming**

- Importance of consistent naming cannot be overstated
- Naming conventions for parameters
  
  
  
  [preposition] _ <principal> _ {qualifier} _ <place> _ {qualifier} _ <adverbial> _ [index] _ [unit]

  
  
  
  Example: r_int_tot_lyr_pu_dl

- Benefits:
  - Relatively easy to understand parameter from its name
  - Distinguishable (no two parameters should have the same name)
    - \( r\_\text{int} \) (interconnect resistance) = \( r\_\text{int} \) (interconnect resistivity) ?
  - Unique (no two names for the same parameter)
    - \( R\_\text{int} = R\_\text{wire} \) ?
  - Sortable (important literals come first)
Additional Features

- Optimization over a collection of rules (with constraints)
  - Example: buffer insertion for minimal delay with area constraints
- Executables can be called
  - Example: various optimizations of global delay through IPEM (Interconnect Performance Estimation Models, J. Cong, UCLA)
- Internal code rules for optimizations
  - Example: optimization of number and size of repeaters for global wires
- Storing of calibration data (e.g., “technology files”) for known process, design points

Additional Features (cont.)

- Visualization (plotting, printing, saving to file)
- Sweeping over sets of input values
  - Example: clock frequency for different Rent exponents and varying logic depth
GTX: Open and User-friendly

- Openness in grammar, parameters and rules
  - Easy sharing of data in research environment
  - Contributions from other groups
- Allows developing of proprietary models
  - Separation between supplied (shared) and user-defined parameters / rules
  - GTX offers usability behind firewalls
  - Framework for sharing results instead of data is planned
- Multi-platform (SUN Solaris, Windows, Linux)

Demonstration
GTX Current Status

◆ **Emulation of:**
  ▲ Cycle-time models of SUSPENS (with extension by Takahashi), BACPAC, Fisher (ITRS)
  ▲ Interconnect tuning studies

◆ **Main modules**
  ▼ Clock / power
  ▼ SOI
  ▼ Domino logic
  ▼ Device and Power
  ▼ Global interconnect
  ▼ System-level power
  ▼ Packaging
  ▼ Reliability and Yield
  ▼ ...

---

GTX Current Status

◆ **Evaluation** of cycle-time models
  ▲ Parameter sensitivity

◆ **Comparison** between cycle-time models
  ▲ Model sensitivity

◆ **Development** of new models
  ▲ Model of via impact on required routing resources (number of layers, pitch, etc.)
**Evaluation: Parameter Sensitivity**

- Change parameter values and observe resulting difference in outputs

- See poster on Sensitivity Analysis for further details

**Comparison: Model Sensitivity**

- Replace rule in a model's rule chain by another model's rule and observe the difference in outputs

- See poster on Sensitivity Analysis for further details
**Development: Via Impact Model**

- **Goal:** Model impact of vias on layer track utilization
- Only taking into account area taken by via is not enough
- Stochastic model of the number of wires blocked by vias used to estimate the via impact
- Via impact model improves prediction of number of layers needed for the routing
  - Verified with recent 4LM block, Cadence Silicon Ensemble P&R
- See poster on Via Impact Model for further details

**Conclusion**

- GTX: A new framework for roadmapping models and technology extrapolation efforts
  - Flexible and extensible
  - Enables easy reuse of models
  - Provides a common parameter base between all models
  - Provides user interaction
  - Relies on open-source and contributions by expert users
  - “Living Roadmap”
- Technology extrapolation becomes easier
- More principled understanding of requirements for CAD tools
**GTX Project Information**

- Design: A. Caldwell, A. B. Kahng, I. Markov, M. Oliver
- Implementation: M. Oliver
- Knowledge gathering and implementation: A. B. Kahng, F. Koushanfar, H. Lu, D. Stroobandt
- Detailed information and downloading of prototype version of GTX: [http://www.gigascale.org/GTX](http://www.gigascale.org/GTX)
- To contact the developers, ask questions, send comments, or to contribute with models to be included in GTX, please send E-mail to GTX@cs.ucla.edu

**Acknowledgements**

- MARCO/DARPA GSRC
- F.W.O. (Belgium) for D. Stroobandt’s grant to visit UCLA
- Dr. Phil Fisher, Dr. Dennis Sylvester and Kevin Cao for providing access to their models and helpful inputs
- Professors Ken Rose, James Meindl, Scott Wills and Kurt Keutzer for fruitful discussions
Future Axes of Achievable Designs: System-In-Package (SIP)

Wayne Dai
Huaizhi Wu, Shuo Zhang

Outline

◆ Opportunities and challenges for System-On-Chip (SOC)
◆ System-In-Package (SIP) as a generalization of SOC
  ▲ RF ICs and high-Q passive integration
  ▲ Modeling and analysis of integrated spiral inductors
  ▲ Memory and logic integration
  ▲ I/O rerouting and trade-off analysis
◆ Soft wire planning based on rubber-band routing
◆ Soft block packing based on Bounded Slicing Grid (BSG)
◆ Chip-package co-design of P/G distribution system
Opportunities for System-On-Chip (SOC)

- **100+ million** transistors can be made on a single die
- **$10^{17}$** transistors is being made in a year
- Each new year makes almost the same amount of transistors as all of the previous years combined
- The cost of a transistor is approaching “zero cost”
  
  $10^{-5} \frac{\text{¢}}{\text{transistor in DRAM}}, 10^{-6} \frac{\text{¢}}{\text{transistor in CPU}}$

- Electronic market will be driven by non-PCs rather than by PCs

  “The electronics market is a phenomenally elastic market”

  — Gordon Moore

Challenges for SOC...

- The cost of a memory cell for embedded DRAM is 20 times higher than that for discrete DRAM.
  
  “A couple of years ago we really thought that the embedded DRAM model would be the panacea for many applications. It’s not always the right thing. In many applications it still remains much cheaper to do it with multichip modules. It gives you satisfactory performance and often for lower cost.”

  — John Kelly, General manager of IBM Microelectronics

- There is a wide disparity in revenue per square inch among various blocks on chip.
  
  (DSP & μP $150\text{~}~$200/in², FPGA $120/in², Analog $35/in², Memory $50\text{~}~$60/in²)

  “You’re basically diluting your high-value logic pieces with all these other low-value pieces, yet you’re adding cost because you’re adding mask levels.”

  — Clark Fuhs, VP and Director of Dataquest’s Semiconductor Manufacturing Programs
... Challenges for SOC

◆ It is difficult to integrate noisy digital circuits with noise-sensitive analog circuits.

“We have system-on-chip now that they are really 'system on chips'. We do it that way because it’s most cost-effective, and the customer will prefer it that way because it offers more flexibility.”

— John Dickson, President of Lucent Technologies’ Microelectronics Group

◆ It is very difficult, if not impossible, to integrate high-Q passive components with high-frequency ICs.

◆ The new challenge is not how many transistors can be built on a single chip, but rather how to integrate diverse technologies together, predictably and cost-effectively.

System-In-Package — A generalization of SOC

◆ A giant chip rather than a miniaturized circuit board: preserving on-chip electrical environment
System-In-Package — A generalization of SOC

- Overcome formidable integration barriers without compromising individually optimized chip technologies
- Match or exceed SOC performance with lower cost, by preserving on-chip electrical environment
- Reduce time to market by taking advantage of today’s capabilities
- Unlock the full potential of IC technology
- Unleash the innovation of designers
- SIP brings new “axes” of achievable designs

SIP: RF ICs and High-Q Passive Integration ...

- 7mm x 7mm substrate
- Transceiver IC (Lucent W2020)
- Surface - Mountable Varactor
- Integrated Thin-film Passives

Single-Package GSM Transceiver

SIP Eliminate Spurious Resonance Below 3GHz
... SIP: RF ICs and High-Q Passive Integration

Cross Section of Lucent SIP

- L = 1 ~ 100 nH
- Q = 10 ~ 90
- Variation < 3%
- Mismatch < 1%

Modeling and Analysis of Integrated Spiral Inductors

Test Coupon on Wafer

Substrate Current

Edge Effect Simulation

Modeling of Integrated Inductor

Reflection Amplitude

Reflection Phase
**SIP: DRAM and ASIC Integration**

Chip-on-Chip

Wafer-Level Assembly

Memory-on-logic Logic-on-memory

I/O Rerouting

---

**SIP: DRAM and FPGA Integration**

1.4 cm

UCSC FPMCM III: Chip-on-Chip

I/O Rerouting on FPGA
All-angle I/O Rerouting

- Given package size and number of pins, what is maximum pin pitch?
- Given number of pins, what is minimum package size?
- Given package size, what is maximum total pins?

Soft Wire Planning Based on Rubber-Band Routing

- Rubber band Extraction
- Geometric Transformation
- Even Wire Distribution
Rubber-Band Routing

Topological Routing → Canonical Form → Rubber-band Sketch → Spoke Creation

Geometric Routing → Geometry Transformation → Extended Rubber-band Sketch

Octilinear Routing

“Wrong Way” Routing vs. Octilinear Routing

- Worst-case coupling reduced: 50% in “Wrong way” routing, 70% in octilinear routing
- Layer utilization reduced: 20% in “Wrong way” routing, 30% in octilinear routing

- What are the noise/area/delay tradeoffs inherent in octilinear vs. rectilinear routing?
- What is the cost of reducing worst-case coupling by X%?
Noise Avoidance Local Routing

- Rubber-Band Routing
- Geometry Routing

Before Wire Sizing

After Wire Sizing

Noise Avoidance Global Routing

- Global crosstalk estimation based on route topology, receiver noise margins and clock phases.
- 20% average reduction in crosstalk, 3% average increase in max density

Local crosstalk metric based on net coupling length
- 19.8% reduction after global route, 18.5% reduction after local route
Chip-Package Co-Design

- For a gigascale integrated system, a die may not be able to be packaged if the die and package are not co-designed and co-simulated.

- Package design should be part of IC design planning.
  - On-package global clock distribution for lower power and easier skew management
  - Area I/O power and ground distribution for better noise control

Chip-Package Co-design of Power/Ground Distribution System...

- How many P/G pins needed?
- How much decoupling capacitance? Too little noisy power supplies Too much unpredictable LC resonance increase die area

Resistive drops
- Very low
- Low
- Medium
- High

Inductive drops
- Low
- Medium
- High
- Low

# Layers
- High
- Medium
- Low

Planes

Grid Mesh Planes

Cross Traces
...Chip-Package Co-design of Power/Ground Distribution System

- On IC, hybrid full-wave techniques are applied for different types of P/G structures

![Signal Trace between Two Planes on Package Level](image1.png)

![A Pulse Propagate down the Via and onto the Trace](image2.png)

<table>
<thead>
<tr>
<th>Mesh Density</th>
<th>ASTAP on IBM 3090 Mainframe</th>
<th>Decomposition Method on IBM R/6000-350 Workstation</th>
<th>Ratio of CPU Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 x 30</td>
<td>1 m 55.29 s</td>
<td>0.18 s</td>
<td>640</td>
</tr>
<tr>
<td>42 x 42</td>
<td>5 m 42.73 s</td>
<td>0.35 s</td>
<td>900</td>
</tr>
<tr>
<td>60 x 60</td>
<td>19 m 30.88 s</td>
<td>0.74 s</td>
<td>1982</td>
</tr>
</tbody>
</table>

- In package, EM fields are decomposed into two modes (J. Fang, UCSC):
  - Strip-line mode fields propagate along metal traces
  - Parallel-plate mode field propagate between adjacent planes
  - Three to four orders of magnitude faster than ASTAP

<table>
<thead>
<tr>
<th>Iterations</th>
<th>IMET</th>
<th>MEI</th>
<th>MoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Inversion</td>
<td>1.6s</td>
<td>3.2s</td>
<td>4.8s</td>
</tr>
<tr>
<td>Total</td>
<td>2.7s</td>
<td>5.1s</td>
<td>7.4s</td>
</tr>
</tbody>
</table>

Collaborations

- Worked with Austin lab, IBM on soft block packing for GHz processor design
- Worked with Intel on chip-package co-design of P/G distribution systems
- Worked with Bell Labs, Lucent on modeling and analysis of integrated passives
Concluding Remarks

- System-On-Chip should be generalized to System-In-Package (SIP).
  - SIP provides new opportunities for gigascale integration.
  - SIP creates new dimensions for achievable design.
- Area I/O opens up a new paradigm for trading off on-chip interconnect versus on-package interconnect.
- Soft wire planning and soft block packing are core engines for chip-package co-design.
- Top level of IC and bottom level of package need to be co-designed and co-simulated.

CAD-IP Reuse: Case Studies and Infrastructure

Andrew E. Caldwell, Andrew B. Kahng
and Igor L. Markov
University of California, Los Angeles
Outline

◆ Motivation: the necessity of CAD-IP Reuse
◆ What is CAD-IP?
◆ Pitfalls in developing CAD-IP
  ▲ Case I: Creating CAD-IP of questionable value
  ▲ Case II: Roadblocks to creating needed CAD-IP
  ▲ Case III: Developing CAD-IP for obsolete contexts
◆ Our solution: CAD-IP reuse via the MARCO/GSRC Bookshelf
◆ Current status of the Bookshelf
◆ How to contribute

Challenges for CAD

◆ Difficult to manage changing data models
  ▲ for tool integration
  ▲ for new use scenarios
  ▲ due to changing technology and design processes
◆ Difficult to develop useful algorithms, CAD tools, etc.
  ▲ time to market: 5-7 year delay from publishing to first industrial use
  ▲ quality of results: unmeasurable, unpredictable, basically unknown
◆ A possible solution: cultivate flexibility and reuse
  ▲ low cost “update” of previous work to support reuse
  ▲ future tool/algorith development biased towards reuse
**Analogy: Hardware Design and CAD Tool Design**

- **Hardware design is difficult**
  - complex electrical engineering and optimization problems
  - mistakes are costly
  - verification and test not trivial
  - few can afford to truly exploit the limits of technology (full custom)
  - A Winning Approach: Hardware IP reuse

- **CAD tools design is difficult**
  - complex software engineering and optimization problems
  - mistakes can be showstoppers
  - verification and test not trivial
  - few can manage the complexity of leading-edge approaches
  - A "Surprising" Idea: CAD-IP reuse

**What is CAD-IP?**

- **Data models and benchmarks**
  - context descriptions and use models
  - testcases and good solutions

- **Algorithms and algorithm analyses**
  - mathematical formulations
  - comparison and evaluation methodologies for algorithms
  - executables and source code of implementations
  - leading-edge performance results

- **Traditional (paper-based) publications**
  - 6-page 2-column papers with tables, references, URLs, etc.
Challenges to QOR in CAD Research

◆ Research in mature areas at risk of becoming inefficient
  ▲ incremental research - difficult and risky
    ▼ implementations not available    duplicated effort
    ▼ too much trust    which approach is really the best?
    ▼ some results may not be replicable
    ▼ ‘not novel’ is common reason for paper rejection
  ▲ exploratory research - paradoxically, lower-risk
    ▼ novelty for the sake of novelty
    ▼ yet, novel approaches must be well-substantiated

◆ Pitfalls for CAD-IP
  ▲ Case I: Creating CAD-IP of questionable value
  ▲ Case II: Roadblocks to creating needed CAD-IP
  ▲ Case III: Developing CAD-IP for obsolete contexts

Case I: Creating CAD-IP of Questionable Value

◆ Recent hypergraph partitioning papers report FM implementations 20x worse than leading-edge FM

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>LIFO-FM</th>
<th>Ibm01</th>
<th>Ibm02</th>
<th>Ibm03</th>
<th>Ibm04</th>
<th>Ibm05</th>
<th>Ibm06</th>
</tr>
</thead>
<tbody>
<tr>
<td>2% Paper1</td>
<td>Min(100)</td>
<td>450</td>
<td>648</td>
<td>2459</td>
<td>3201</td>
<td>2397</td>
<td>1436</td>
</tr>
<tr>
<td></td>
<td>Ave</td>
<td>2701</td>
<td>12253</td>
<td>16944</td>
<td>20281</td>
<td>3420</td>
<td>16578</td>
</tr>
<tr>
<td>Paper2</td>
<td>Min(100)</td>
<td>366</td>
<td>301</td>
<td>1588</td>
<td>1014</td>
<td>2640</td>
<td>1008</td>
</tr>
<tr>
<td></td>
<td>Ave</td>
<td>594</td>
<td>542</td>
<td>2688</td>
<td>1802</td>
<td>3382</td>
<td>1746</td>
</tr>
</tbody>
</table>

▲ previous lack of openness caused wrong conclusions, wasted effort
  ▼ some “improvements” may only apply to weak implementations
  ▼ duplicated effort re-implementing (incorrectly?) well-known algorithms

▲ difficult to find the leading edge
  ▼ no standard comparison methodology
  ▼ how do you know if an implementation is poor?

◆ To make leading-edge apparent and reproducible
  ▲ publish performance results on standard benchmarks
  ▲ peer review (executables, source code?)
Case II: Roadblocks to Creating Needed CAD-IP

◆ “Best approach” to global placement?
  ▲ recursive bisection (1970s)
  ▲ force-directed (1980s)
  ▲ simulated annealing (1980s)
  ▲ analytical (1990s)
  ▲ hybrids, others

◆ Why is this question difficult?
  ▲ lastest public placement benchmarks are from 1980s
  ▲ data formats are bulky (hard to mix and match components)
  ▲ no public implementations since early 1990s
  ▲ new ideas are not compared to old

◆ To match approaches to new contexts
  ▲ agree on common up-to-date data model
  ▲ publish good format descriptions, benchmarks, performance results
  ▲ publish implementations

Case III: Developing CAD-IP for Obsolete Contexts

◆ Global placement example
  ▲ much of academia studies variable-die placement
    ▼ row length and spacing not fixed
    ▼ explicit feedthroughs
  ▲ majority of industrial use is fixed-die
    ▼ pre-defined layout dimensions
  ▲ HPWL-driven vs. routability- or timing-driven
  ▲ runtimes are often not even reported
  ▲ this affects benchmarks and algorithms

◆ Solution: perform sanity checks and request feedback
  ▲ explicitly define use model and QOR measures
  ▲ establish a repository for up-to-date formats, benchmarks etc.
  ▲ peer review (executables, source code?)
Bookshelf: A Repository for CAD-IP

- “Community memory” for CAD-IP
  - data models
  - algorithms
  - implementations

- Publication medium that enables efficient algorithm research
  - benchmarks, performance results
  - algorithm descriptions and analyses
  - quality implementations

- Simplified comparisons to identify best approaches

- Easier for industry to communicate new use models

Bookshelf: Addressing Inefficiencies

- Summary of inefficiencies
  - lack of openness and standards
  - huge duplication of effort
  - incomparable reporting
  - improvement difficult
  - lack of standard comparison/latest use models
  - best approach not clear
  - no feedback from industry
  - outdated use models

- Summary of proposed solutions
  - widely available, up-to-date, extensible benchmarks
  - consistent performance reporting for leading-edge approaches
  - available detailed descriptions of algorithms
  - peer review of executables (source code?)
  - credit for quality implementations

- Assists academic research

- Leads to faster industry adoption
The GSRC Bookshelf for CAD-IP

- Bookshelf consists of slots
  - slots represent active research areas with “enough customers”
  - collectively, the slots cover the field
- Who maintains slots?
  - experts in each topic collaborate to produce them - anyone can submit
- Currently, 10 active slots
  - SAT (U. Michigan, Sakallah)
  - Graph Coloring (UCLA, Potkonjak)
  - Hypergraph Partitioning (UCLA, Kahng)
  - Block Packing (UCSC, Dai)
  - Placement (UCLA, Kahng)
  - Global Routing (SUNY Binghamton, Madden)
  - Single Interconnect Tree Synthesis (UIC, Lillis and UCLA, Cong)
  - Commitments for more: BDDs, NLP, Test and Verification

What’s in a Slot?

- Introduction
  - why this area is important and recent progress
  - pointers to other resources (links, publications)
- Data formats used for benchmarks
  - SAT, graph formats etc.
  - new XML-based formats
- Benchmarks, solutions, performance results
  - including experimental methodology (e.g., runtime-quality Pareto curve)
- Binary utilities
  - format converters, instance generators, solution evaluators, legality checkers
  - optimizers and solvers
  - executables
- Implementation source code
- Other info relevant to algorithm research and implementations
  - detailed algorithm descriptions
  - algorithm comparisons
Current Progress on the CAD-IP Bookshelf

- Bookshelf@gigascale.org
  - 33 members (17 developers)
- Main policies and mechanisms published
- 10 active slots
  - inc. executables, performance results for leading-edge partitioners, placers
- First Bookshelf Workshop, Nov. 1999
  - attendance: UCSC, UCB, NWU, UIC, SUNY Binghamton, UCLA
  - agreed on abstract syntax and semantics for initial slots
  - committed to XML for common data formats
  - peer review of slot webpages
- Ongoing research uses components in the Bookshelf

Uses of the Bookshelf

- Improvement of algorithms through enhanced comparisons
  - better partitioning implementations
- UMN → UCSC → UCLA flow (assembled in days)
  - UMN partitioner (hMetis) generates blocks
  - UCSC floorplanner places blocks
  - UCLA placer (Capo) arranges cells within each block
    - can output LEF/DEF to invoke Cadence WRoute
  - supports comparisons
    - partitioner tolerance
    - aspect-ratio limits on soft blocks
  - use model: fixed-die, low whitespace
    - need a nearly perfect packing from floorplanner
    - where to allocate the whitespace?
Contributing to Bookshelf

- Request membership in the bookshelf group at http://www.gigascale.org
  - ask for "developer" membership (as applicable)
  - go over existing slots related to your research
  - browse mail archives

- Are you creating a new bookshelf slot?
  - Yes (no existing slots are appropriate)
    - use slot template from Web page (fill in the blanks)
    - write intro
    - give references to relevant sites
    - mail the URL (or the HTML) to bookshelf developers
  - No (contribution to an existing slot)
    - agree with the maintainers of the slot about your contribution
      - e.g., convert LaTeX tables from conference papers into HTML by tth
    - mail URL or your contributions to slot maintainers
**Motivation: Complexity of the Design Process**

- Ability to make silicon has outpaced ability to design it
- Complex data, system interactions
- SOC
  - more functionality and customization, in less time
  - design at higher levels of abstraction, reuse existing design components
  - customized circuitry must be developed predictably, with less risk
- Key question: “Will the project succeed, i.e., finish on schedule and under budget while meeting performance goals?”
- SOC design requires an organized, optimized design process

---

**Value of CAD Tools Improvement Not Clear**

- What is $ value of a “better” scheduler, mapper, placer?
- What is $ value of GUI, usability, ...?
- What is the right objective?
  - min wirelength $\neq$ routable
  - min literals $\neq$ amenable to layout
- Value well-defined only in context of overall design process
**What is the Design Process?**

- Not like any “flow/methodology” bubble chart
  - backs of envelopes, budgeting wars
  - changed specs, silent decisions, e-mails, lunch discussions
  - ad hoc assignments of people, tools to meet current needs
  - proprietary databases, incompatible scripts/tools, platform-dependent GUIs, lack of usable standards
  - design managers operate on intuition, engineers focus on tool shortcomings
- Why did it fail?
  - “CAD tools”
  - “inexperienced engineers”
- Must measure to diagnose, and diagnose to improve

**What Should be Measured?**

- Many possibilities
  - running a tool with wrong options, wrong subset of standard
  - bug in a translator/reader
  - assignment of junior designer to project with multiple clocks
  - difference between 300 MHz and 200 MHz in the spec
  - changing an 18-bit adder into a 28-bit adder midstream
  - decision to use domino in critical paths
  - one group stops attending budget/floorplan meetings
- Solution: record everything, then mine the data
Design Process Data Collection

- What revision of what block was what tool called on?
  - by whom?
  - when?
  - how many times? With what keystrokes?
- What happened within the tool as it ran?
  - what was CPU/memory/solution quality?
  - what were the key attributes of the instance?
  - what iterations/branches were made, under what conditions?
- What else was occurring in the project?
  - e-mails, spec revisions, constraint and netlist changes, ...
- Everything is fair game; bound only by server bandwidth

Unlimited Range of Possible Diagnoses

- User performs same operation repeatedly with nearly identical inputs
  - tool is not acting as expected
  - solution quality is poor, and knobs are being twiddled
- Email traffic in a project:
  - missed deadline, missed revised deadline; people disengaged; project failed
- On-line docs always open to particular page
  - command/option is unclear
**METRICS System Architecture**

![Diagram of METRICS System Architecture]

**METRICS Transmitter**

- **No functional change to the tool**
  - use API to send the available metrics

- **Low overhead**
  - example: standard-cell placer using Metrics API → < 2% runtime overhead
  - even less overhead with buffering

- **Won't break the tool on transmittal failure**
  - child process handles transmission while parent process continues its job

```
initToolRun()
sendMetrics()
sendMetrics()
sendMetrics()
```
**METRICS Transmitter**

![Diagram of METRICS Transmitter]

**Transmitter Example**

```c
/** API Example **/
int main(int argc, char * argv[ ] )
{
    ...
    toolID = initToolRun( projectID, flowID );
    ...
    printf( "Hello World\n" );
    sendMetric( projectID, flowID, toolID,
                "TOOL_NAME", "Sample" );
    sendMetric( projectID, flowID, toolID,
                "TOOL_VERSION", "1.0" );
    ...
    terminateToolRun( projectID, flowID, toolID );
    return 0;
}
```

```c
## Wrapper example
($File, $PID, $FID) = @ARGV;
$TID = `initToolRun $PID $FID`;
open ( IN, "<$File" );
while ( <IN> )
{
    if ( /Begin\s+(\S+)\s+on\s+(\S+.*)/)
    {
        system "sendMetrics $PID $FID $TID "
            TOOL_NAME $1";
        system "sendMetrics $PID $FID $TID "
            START_TIME $2";
    }
    ...
}
system "terminateToolRun $PID $FID $TID";
```
Example of METRICS XML

```xml
<?xml version="1.0" ?>
<METRICSPACKET>
  <REQUEST>
    <TYPE> TOOL </TYPE>
    <PROJECTID> 173 </PROJECTID>
    <FLOWID> 9 </FLOWID>
    <PARAMETER> 32 </PARAMETER>
  </REQUEST>
  <METRICS>
    <PROJECTID> 173 </PROJECTID>
    <FLOWID> 9 </FLOWID>
    <TOOLID> P32 </TOOLID>
    <DATETIME> 93762541300 </DATETIME>
    <NAME> TOOL_NAME </NAME>
    <VALUE> CongestionAnalysis </VALUE>
  </METRICS>
</METRICSPACKET>
```

Current Testbed: A Metricized P&R Flow
**METRICS Reporting**

- **Web-based**
  - Platform independent
  - Accessible from anywhere

- **Example: correlation plots created on-the-fly**
  - Understand the relation between two metrics
  - Find the importance of certain metrics to the flow
  - Always up-to-date

---

**METRICS Reporting**

Diagram showing the interaction between WEB Browser, Java Servlet, Oracle8i, 3rd Party Graphing Tool (Excel, Lotus), and Local Graphing Tool (GNUPlot) with Inter/Intra-net.
Example Reports

Congestion vs WL

# Via vs WL

METRICS Standards

◆ Standard metrics naming across tools
  ▶ same name → same meaning, independent of tool supplier
  ▶ generic metrics and tool-specific metrics
  ▶ no more ad hoc, incomparable log files

◆ Standard schema for metrics database
**Generic and Specific Tool Metrics**

<table>
<thead>
<tr>
<th>Generic Tool Metrics</th>
<th>Placement Tool Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>tool_name char</td>
<td>num_cells integer</td>
</tr>
<tr>
<td>tool_version char</td>
<td>num_nets integer</td>
</tr>
<tr>
<td>tool_vendor char</td>
<td>layout_size double</td>
</tr>
<tr>
<td>compiled_date mm/dd/yyyy</td>
<td>row_utilization double</td>
</tr>
<tr>
<td>start_time hh:mm:ss</td>
<td>wirelength double</td>
</tr>
<tr>
<td>end_time hh:mm:ss</td>
<td>weighted_wl double</td>
</tr>
<tr>
<td>tool_user char</td>
<td></td>
</tr>
<tr>
<td>host_name char</td>
<td></td>
</tr>
<tr>
<td>host_id char</td>
<td></td>
</tr>
<tr>
<td>cpu_type char</td>
<td></td>
</tr>
<tr>
<td>os_name char</td>
<td></td>
</tr>
<tr>
<td>os_version char</td>
<td></td>
</tr>
<tr>
<td>cpu_time hh:mm:ss</td>
<td></td>
</tr>
</tbody>
</table>

| Routing Tool Metrics         |                                        |
| num_layers integer           |                                        |
| num_violations integer       |                                        |
| num_vias integer             |                                        |
| wirelength double            |                                        |
| wrong-way_wl double          |                                        |
| max_congestion double        |                                        |

**Current Status**

- Completion of METRICS server with Oracle8i, Servlet, and XML parser
- Initial transmittal API in C++
- METRICS wrapper for Cadence P&R tools
- Simple reporting scheme for correlations
**Additional Infrastructure**

- Industrial standard network discovery
  - Jini, UPNP (Universal Plug & Play), SLP (Salutation)

- Security
  - Encryption for XML data
  - SSL (Secure Socket Layer)
  - User id & password authentication (reporting)
  - Registered users (transmitting)

- 3rd party reporting tool
  - MS Office integration, Crystal report, ...

- Data mining

---

**METRICS Demo**

- Transmission of metrics
  - API inside tools
  - Perl wrapper for log files

- Reporting
  - Correlation reports
  - Progress on current tool run, flow, design
Potential Benefits to Project Management

◆ Accurate Resource Prediction At any point in Design Cycle
  ▲ up front estimates for People, Time, Technology, EDA Licenses, IP re-use...
  ▲ go/no go at earliest point

◆ Accurate Project Post-mortems
  ▲ Everything tracked - tools, flows, users, notes
  ▲ Optimize for next Project based on past results
  ▲ No “loose”, random data or information left at Project end (log files!!!)

◆ Management Console
  ▲ Web-based, status-at-a-glance of Tools, Designs, Systems at any point in project

◆ No wasted resources
  ▲ prevent out of sync runs
  ▲ no duplication of data or effort

Potential Benefits to Tools R&D

◆ Methodology for continuous tracking data over entire lifecycle of instrumented tools

◆ More efficient analysis of realistic data
  ▲ no need to rely only on extrapolations of tiny artificial “benchmarks"
  ▲ no need to collect source files for test cases, and re-run in house

◆ Facilitates identification of key design metrics, effects on tools
  ▲ standardized vocabulary, schema for design/instance attributes

◆ Improves benchmarking
  ▲ apples to apples, and what are the apples in the first place
  ▲ apples to oranges as well, given enough correlation research
Potential Research Enabled by METRICS

- **Tools:**
  - scope of applicability
  - predictability
  - usability

- **Designs:**
  - difficulty of design or manufacturing
  - verifiability, debuggability/probe-ability
  - likelihood of a bug escape
  - $ cost (function of design effort, integratability, migratability, ...)

- **Statistical metrics, time-varying metrics**

- **What is the appropriate abstraction of manufacturing process for design?**
  - Impact of manufacturing on design productivity
  - Inter- and intra-die variation
  - Topography effects
  - Impact, tradeoffs of newer lithography techniques and materials

Ongoing Work

- **Work with EDA, designer community to establish standards**
  - tool users: list of metrics needed for design process optimization
  - tool vendors: implementation of the metrics requested with the standardized naming

- **Improve the transmitter**
  - add message buffering
  - “recovery” system for network / server failure

- **Extend METRICS system to include project management tools, email communications, etc.**

- **Additional reports, data mining**
Summary: Calibrating Achievable Design

Andrew B. Kahng
University of California, Los Angeles

Ten Year Vision

◆ Focused and efficient CAD R&D
  ▲ Aware of design process context
  ▲ Interoperable and reusable
  ▲ Driven by process, applications, architecture contexts

◆ Principled, reliable and effective R&D
  ▲ Right problems solved at right time
  ▲ Measurable impact on design technology
**First Year Progress**

- GTX1.0 release
  - Platform-independent engine, GUI; flexible definition/display of studies
  - New models of (1) optimization effects in layers between individual wires/devices and system architecture, (2) global interconnect resource
  - Insights on sensitivities, accuracy of existing extrapolations
- CAD-IP Reuse
  - Convergence on policies and standards, underlying data model
  - 10 Bookshelf slots active, contents being used in research
- Metrics
  - Metrics server with Oracle8i, Java servlet and XML parser
  - Initial transmittal API in C++
  - Metrics wrapper for Cadence P&R tools
  - Web-based reporting of correlation analyses

**Current Team**

- Jason Cong (30%)
- Wayne Dai
- Andrew B. Kahng
- Kurt Keutzer (20%)
- Wojciech Maly (50%)