Computer Aided Design of Digital Systems I

Professor Massoud Pedram
Dept. of EE
University of Southern California
Spring 2003

Instructor Info

- Massoud Pedram
- Office: EEB 344
- Office Hours: Thursdays 9-12, EEB 344
- Email: pedram@ceng.usc.edu
Soroush Abbaspour
Office: EEB 206
Office hours: Mondays 10-12, EEB 201
Email: soroush@sahand.usc.edu

When: Fridays, 9-11:50 am
Where: OHE 100
Class Web Page:
http://www-classes.usc.edu/engr/ee-s/680
Readings

Required Textbook:
- We will not be using a textbook for this class, but will instead use handouts of powerpoint slides and published papers from the literature. These will be accessible from the class web site.

Recommended Readings:

Course Objectives

- EE680 is a CAD course aimed at graduate students who want broad exposure to the algorithms and data structures of modern VLSI design tools.
- EE680 is an applied algorithms class, where the application area is VLSI design. We focus on the class of designs called Application Specific Integrated Circuits where the goal is to go from a high-level design down to a mask layout both correctly and quickly.
**Prerequisites**

- EE457
- EE477L or EE577a
- CSCI 455x
- Graduate student standing

---

**Syllabus**

- Introduction to Physical Design
- Partitioning
- Placement
- Global Routing
- Detailed Routing
- Static Timing Analysis
- Clock Design
- Supply Network Design
- Parasitic Extraction
Grading

- Homework: 30%
  - Late homework will not be accepted
- Midterm Exams: 40%
  - Two in-class, open-book exams, each 20%
- Final Project: 30%
  - Requires people to work in groups of two

Policies

- No “Incomplete” grades will be given for the course, except under very extreme circumstances.
- You are not permitted to submit extra work in an attempt to raise your grade.
- You are responsible for all assigned readings and information presented in class, including due dates, assignments, exams and so forth. Moreover, you are expected to attend all class meetings.
- Scholastic misconduct will not be tolerated. Scholastic dishonesty includes, but is not limited to: cheating on assignments or exams; plagiarizing; or interfering with another student’s work.
Modern Physical Design: Algorithm Technology Methodology

Based on ICCAD-00 tutorial by Chow/Kahn/Sarrafzadeh with contributions from Keutzer

Outline

- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R
Micro-electronic Industry

Micro-electronic Industry: US$920 billion  
personal computers,  
cellular phones,  
networking equipment,  
video games, cameras

Supported by

Semiconductor Industry: $US150 billion  
microprocessors, micro-controllers, DSPs,  
integrated chipsets

Semiconductor Industry

US $150 Billion

- Analog 15.2%
- MOS Micro 33.7%
- MOS Logic 14.8%
- MOS Memory 18.3%
- Bipolar Digital 0.9%
- Discrete & Opto. 13.2%
Semiconductor Manufacturing

- Design Methodology
- New Process
- New Material
- New Procedure

System specification

Design \(\rightarrow\) Fabrication \(\rightarrow\) Packaging \(\rightarrow\) Test

Tools: design simulation emulation

- Equipment
- Material
- Equipment
- Equipment

$3 billion $33 billion $23 billion

IC $150 billion

Making Sausages

- Piglet
- Pig
- Pork
- Sausages
- Packaging
Semiconductor Industry

- Rapid advancement in technology
- Miniaturization
- Low cost
- Unbounded market needs
- Vertical integration
- Capital intensive
- Customization
- Short product life
- Brain power intensive

Technology

- rapid advancement in technology
- miniaturization, low cost
- cheaper, smaller, faster systems
- greater market needs

Moore’s Law
Miniaturization (Production)

Thickness of skin is 100 m
Diameter of a piece of hair is 50 m
Finger nails grow by 1 m in 10 minutes

Miniaturization

[Diagram showing data for WW and Far East]
Influencing CAD: Moore’s Law

Microprocessors

NTRS: CHIP Frequency (GHz)
Low Cost

- A transistor cost $30 in 1960
- 8080 (5,000 tx) cost $150 in 1974: 3 cents/tx
- Pentium-II (7,500,000 tx) cost $225 in 1997: 0.003 cents/tx

- 1,000 fold decrease from 1960 to 1974
- 1,000,000 fold decrease from 1960 to 1997

Vertical Integration

Design → Masks → Foundry → Packaging → Test

- Grow Crystals
- Wafers
- Slice/Polish Wafer
- Chemicals
- Lead frame
Capital Intensive (Construction Cost)

Moore's Law also applies!

Customization and Short Product Life

Complex Design Work

Short Design Cycle

Brain Power

Design Power
Tools and Design Methodology

Underlying Architecture adds yet Another Dimension

Tools & Design Methodology Are Inextricably Interwoven-like an “Electronic” DNA

Where does CAD fit in

Electronic Systems

Real World

Silicon Foundries

FDA Industry

Semiconductor Industry
CAD at every level...

Real World

Speech processing
Signal processing
Performance analysis

System synthesis
HW/SW co-design

Formal verification
Logic synthesis
Place and route
Circuit simulation

Device simulation
Process modeling

Increased Device and Context Complexity

Exponential increase in device complexity—increasing with Moore’s law (or faster)!
System context in which devices are deployed (e.g. cellular radio) are increasing in complexity as well exponential increases in design productivity

We have exponentially more transistors!
Deep Sub Micron effects

Smaller geometries are causing a wide variety of effects that we have largely ignored in the past:
- Cross-coupled capacitances
- Signal integrity
- Resistance
- Inductance

Design of each transistor is getting more difficult!

Heterogeneity on CHIP

Greater diversity of on-chip elements
- Processors
- Software
- Memory
- Analog

More transistors doing different things!
Stronger Market Pressure

- Time-to-Money

Decreasing design window
Less tolerance for design revisions

Exponentially more complex, greater design risk, greater variety, and a smaller design window!

A Quadruple-Whammy

- Time-to-Money
- Heterogeneity

Complexity

Differences
Evolution of EDA Industry

- 1978: Transistor entry - Calma, Computervision
- 1985: Schematic Entry - Daisy, Mentor, Valid
- 1992: Synthesis - Cadence, Synopsys
- 1999: McKinsey S-Curve

Design Process:

- **Design**: specify and enter the design intent
- **Verify**: verify the correctness of design and implementation
- **Implement**: refine the design through all phases
Case Study: Alpha Performance History

![Performance History Chart]

Date of Introduction

Case Study: Complexity Trends

- Process scaling has continued steadily
- Planarization has enabled an increase in the number of interconnect layers
- Transistor counts have increased dramatically with the L2 cache SRAMs
- Additionally, design team size has increased ~40% per generation
- Opportunities to manage complexity and productivity
  - Fundamental understanding and modeling of process and circuit element behaviors
  - High level design methods
  - CAD
  - Design reuse
  - Micro-architecture
Case Study: Power Dissipation Trends

- Power consumption is increasing
  - Better cooling technology needed
- Supply current is increasing faster!
- On-chip signal integrity will be a major issue
- Power and current distribution are critical
- Opportunities to slow power growth
  - Accelerate Vdd scaling
  - Low k dielectrics & thinner (Cu) interconnect
  - SOI circuit innovations
  - Clock system design
  - Micro-architecture

Case Study: Performance Trends

- Performance has increased significantly (7x) faster than frequency
- Performance tracks transistor count when L2 cache ignored
  - Transistor budget has increased more than performance when L2 cache is considered but benchmarks did not reflect larger applications
- Opportunities to continue performance improvements
  - Continued scaling of devices, interconnect and dielectrics
  - Clock distribution
  - Micro-architecture
  - System design
Case Study: Challenging Design Trends

- Micro-architecture and logic design are stressed as frequency has increased faster than scaling.
- Further reducing the number of gate delays per cycle will be difficult.
- Cycles to communicate across chip track with frequency.
- Clock edge rates are not scaling.
- Opportunities to continue performance increases.

Outline

- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R
### Overall Road Map Technology Characteristics

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TECHNOLOGY NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (DRAM HALF-PITCH) (nm)</td>
<td>250</td>
<td>180</td>
<td>130</td>
<td>100</td>
<td>70</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td>DENSE LINES (MPU GATES) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (ASIC) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td>750</td>
<td>1250</td>
<td>2100</td>
<td>3500</td>
<td>6000</td>
<td>10000</td>
<td>1693</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>375</td>
<td>600</td>
<td>1200</td>
<td>1600</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>375</td>
<td>600</td>
<td>1200</td>
<td>1600</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
</tbody>
</table>

### Overall Road Map Technology Characteristics (Cont’d)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TECHNOLOGY NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (DRAM HALF-PITCH) (nm)</td>
<td>250</td>
<td>180</td>
<td>130</td>
<td>100</td>
<td>70</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td>DENSE LINES (MPU GATES) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (ASIC) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td>750</td>
<td>1250</td>
<td>2100</td>
<td>3500</td>
<td>6000</td>
<td>10000</td>
<td>1693</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>3674</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>375</td>
<td>600</td>
<td>1200</td>
<td>1600</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>375</td>
<td>600</td>
<td>1200</td>
<td>1600</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
</tbody>
</table>

### Overall Road Map Technology Characteristics (Cont’d)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TECHNOLOGY NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (DRAM HALF-PITCH) (nm)</td>
<td>250</td>
<td>180</td>
<td>130</td>
<td>100</td>
<td>70</td>
<td>50</td>
<td>35</td>
</tr>
<tr>
<td>DENSE LINES (MPU GATES) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (ASIC) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td>750</td>
<td>1250</td>
<td>2100</td>
<td>3500</td>
<td>6000</td>
<td>10000</td>
<td>1693</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>3674</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>375</td>
<td>600</td>
<td>1200</td>
<td>1600</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>375</td>
<td>600</td>
<td>1200</td>
<td>1600</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
<tr>
<td>DENSE LINES (On-chip, across-chip clock) (high-performance) (MHz)</td>
<td></td>
<td></td>
<td></td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
</tr>
</tbody>
</table>
Technology Scaling Trends

- **Interconnect**
  - Impact of scaling on parasitic capacitance
  - Impact of scaling on inductance coupling
  - Impact of new materials on parasitic capacitance & resistance
  - Trends in number of layers, routing pitch

- **Device**
  - $V_{dd}$, $V_t$, sizing
  - Circuit trends (multi-threshold CMOS, multiple supply voltages, dynamic CMOS)
  - Impact of scaling on power and reliability
Technology Scaling Trends (Cont’d)

- Scaling of x0.7 every three years
  - .25u .18u .13u .10u .07u .05u
  - 5LM 6LM 7LM 7LM 8LM 9LM
- Interconnect delay dominates system performance
  - consumes 70% of clock cycle
Cross coupling capacitance is dominating
- cross capacitance? 100%, ground capacitance? 0%
- 90% in 0.18u
- huge signal integrity implications (e.g., guardbands in static analysis approaches)

Multiple clock cycles required to cross chip
- whether 3 or 15 not as important as fact of “multiple” > 1

Deep-Submicron Interconnect Complexity

Estimated Number of Nets At-Risk
- Risk Factors:
  - Interconnect Delay
  - Signal Integrity
  - Electromigration
  - Process Variations

At-Risk Nets (millions)

Technology (?)
**Scaling of Noise with Process**

- Cross coupling noise increases with
  - process shrink
  - frequency of operation
- Propagated noise increases with decrease in noise margins
  - decrease in supply voltage
  - more extreme P/N ratios for high speed operation
- IR drop noise increases with
  - complexity of chip size
  - frequency of chip
  - shrinking of metal layers

**New Materials Implications**

- Lower dielectric
  - reduces total capacitance
  - doesn’t change cross-coupled / grounded capacitance proportions
- Copper metallization
  - reduces RC delay
  - avoids electromigration (factor of 4-5 ?)
  - thinner deposition reduces cross cap
New Materials Implications (Cont’d)

- Multiple layers of routing
  - enabled by planarized processes; 10% extra cost per layer
  - reverse-scaled top-level interconnects
  - relative routing pitch may increase
  - room for shielding

Technical Issues in UDSM Design

- New issues and problems arising in UDSM
  - catastrophic yield: critical area, antennas
  - parametric yield: density control (filling) for CMP
  - parametric yield: subwavelength lithography implications
    - optical proximity correction (OPC)
    - phase-shifting mask design (PSM)
  - signal integrity
    - crosstalk and delay uncertainty
    - DC electromigration
    - AC self-heat
    - hot electrons
Technical Issues in UDSM Design (Cont’d)

Current context: cell-based place-and-route methodology
- placement and routing formulations, basic technologies
- methodology contexts

Technical Issues in UDSM Design (Cont’d)

- Manufacturability (chip can’t be built)
  - antenna rules
  - minimum area rules for stacked vias
  - CMP (chemical mechanical polishing) area fill rules
  - layout corrections for optical proximity effects in subwavelength lithography; associated verification issues
- Signal integrity (failure to meet timing targets)
  - crosstalk induced errors
  - timing dependence on crosstalk
  - IR drop on power supplies
Technical Issues in UDSM Design (Cont’d)

- Reliability (design failures in the field)
  - electromigration on power supplies
  - hot electron effects on devices
  - wire self heat effects on clocks and signals

Noise Sources

- Analog design concerns are due to physical noise sources
  - because of discreteness of electronic charge and stochastic nature of electronic transport processes
  - example: thermal noise, flicker noise, shot noise
- Digital circuits due to large, abrupt voltage swings, create deterministic noise which is several orders of magnitude higher than stochastic physical noise
  - still digital circuits are prevalent because they are inherently immune to noise
- Technology scaling and performance demands make noisiness of digital circuits a big problem
These effects have always existed, but become worse at UDSM sizes because of:
- finer geometries
  - greater wire and via resistance
  - higher electric fields if supply voltage not scaled
- more metal layers
  - higher ratio of cross coupling to grounded capacitance
- lower supply voltages
  - more current for given power
- lower device thresholds
  - smaller noise margins

Why Now (Cont’d)

Focus on interconnect
- susceptible to patterning difficulties
  - CMP, optical exposure, resist development/etch, CVD, ...
- susceptible to defects
  - critical area, critical volume
**Design Productivity Gap**

“Design Productivity Gap” = “failure of Design Technology”

- Number of available transistors grows faster than designer ability to design them well
  - Increased design effort, risk, turnaround time (TAT)
  - fewer designs are worth trying
- Manufacturing non-recurring engineering (NRE) cost also increasing (mask set)
  - fewer designs are worth trying
- “Workarounds” sacrifice quality, value of designs
  - even with workarounds, fewer designs worth trying
- This is a semiconductor industry problem, not an EDA problem
### The Design Productivity Gap

**“How many gates can I get for $N?”**

<table>
<thead>
<tr>
<th>Year</th>
<th>Technology</th>
<th>Chip Complexity</th>
<th>Frequency</th>
<th>Staff</th>
<th>Cost*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>250 nm</td>
<td>13 M Tr.</td>
<td>400 MHz</td>
<td>210</td>
<td>90 M</td>
</tr>
<tr>
<td>1998</td>
<td>250 nm</td>
<td>20 M Tr.</td>
<td>500</td>
<td>270</td>
<td>120 M</td>
</tr>
<tr>
<td>1999</td>
<td>180 nm</td>
<td>32 M Tr.</td>
<td>600</td>
<td>360</td>
<td>160 M</td>
</tr>
<tr>
<td>2002</td>
<td>130 nm</td>
<td>130 M Tr.</td>
<td>800</td>
<td>800</td>
<td>360 M</td>
</tr>
</tbody>
</table>

* @ $ 150 k / Staff Yr. (In 1997 Dollars)

Source: SEMATECH

### Headcount and Moore’s Law

Headcount is **growing and growing**

Moore’s Law also applies ...
ASSP: 44% WallTime, 39% Total Effort After First Tape-out

Time and Effort Allocation by First Tape-out

<table>
<thead>
<tr>
<th>Percent of Total Project Duration</th>
<th>Percent of Total Project Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>40%</td>
<td>40%</td>
</tr>
<tr>
<td>60%</td>
<td>60%</td>
</tr>
<tr>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

First Tape-out

% Data Source: Collett International Inc.’s Design Productivity Management System (DPMS) database.

** ASSP (Application Specific Standard Product): Standard “off-the-shelf” IC product that has been designed to implement a specific application function.

ASSP Design Productivity +27% Annually

Design Productivity Trend

Key

△ = ASSP*** design

* Data Source: Collett International Inc.’s Design Productivity Management System (DPMS) database.

** Methodology: The design productivity trendline is the ordinary-least-squares (OLS) regression line. 27% is the compound annual growth rate between 06/94 & 06/98.

*** ASSP (Application Specific Standard Product): Standard “off-the-shelf” IC product that has been designed to implement a specific application function.
Silicon Complexity and Design Complexity

- Silicon complexity: physical effects cannot be ignored
  - fast but weak gates; resistive and cross-coupled interconnects
  - subwavelength lithography from 350nm generation onward
  - delay, power, signal integrity, manufacturability, reliability all become first-class objectives along with area
- Design complexity: more functionality and customization, in less time
  - reuse-based design methodologies for SOC
- Interactions increase complexity
  - need robust, top-down, convergent design methodology

Guiding Philosophy in the Back-End

- Many opportunities to leave $$$ on table
  - physical effects of process, migratability
  - design rules more conservative, design waivers up
  - device-level layout optimizations in cell-based methodologies
- Verification cost increases
- Prevention becomes necessary complement to checking
Guiding Philosophy in the Back-End (Cont’d)

- Successive approximation = design convergence
  - upstream activities pass intentions, assumptions downstream
  - downstream activities must be predictable
  - models of analysis/verification = objectives for synthesis
- More “custom” bias in automated methodologies

Implications of Complexity

- UDSM: Silicon complexity + Design complexity
  - convergent design: must abstract what’s beneath
  - prevention with respect to analysis/verification checks
  - many issues to worry about (all are “first-class citizens”)
  - apply methodology (P/G/clock design, circuit tricks, …) whenever possible
- must concede loss of clean abstractions: need unifications
  - synthesis and analysis in tight loop
  - logic and layout: chip implementation planning methodologies
  - layout and manufacturing: CMP/OPC/PSM, yield, reliability, SI, statistical design, …
Implications of Complexity (Cont’d)

- must hit function/cost points that maximize $/\text{wafer}$
  - reuse-based methodology
  - need for differentiating IP / customization

Outline

- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R
Implementation

**Design**: specify and enter the design intent

**Verify**: verify the correctness of design and implementation

**Implement**: refine the design through all phases

---

**Physical Design**

Transform sequential circuit netlist into a physical circuit
- *place* circuit components
- *route* wires
- transform into a mask

Or for FPGA’s
- *place* look-up tables
- *route* wires

Diagram:
- *netlist* → *library* → *physical design* → *layout* → *physical layout*
Standard Cell Layout

Gordian Placement Flow

Fig. 1. Data flow in the placemnt procedure GORDIAN.
Library, Netlist, and Aspect Ratio

Netlist - >100K cells from library

Library/module generators

Size and aspect ratio of core die

Setting up Global Optimization

Fig. 1. Data flow in the placement procedure GORDIAN.
Resulting Layout

Partitioning

Fig. 1. Data flow in the placement procedure GORDIAN.
Layout after Min-cut

Final Placement

Fig. 1. Data flow in the placement procedure GORDIAN.
Slicing Tree

Other details – slotting constraints

A. E. Dunlop, B. W. Kernighan,
A procedure for placement of standard-cell VLSI circuits, IEEE Trans. on CAD, Vol. CAD-4, Jan., 1985, pp. 52-56

Slotting constraints
Generating Final Placement

Fig. 1. Data flow in the placement procedure GORDIAN.

Standard Cell Layout
Global Layout

Grid-Graph Model
Checker-Board Graph (also use slicing structure)

Channel Routing

Basic Terminology:
- tracks
- trunk
- branch
- via
- end

- channel
  - Fixed pin positions on top and bottom edges
- pins
- Classical channel: no nets leave channel
- Three-sided channel possible
Two-Dimensional Compaction

X then Y 1D Compaction  Y then X 1D Compaction

Final Placement

Fig. 10: Macrocell design with standard cell blocks [68] and [69].
Outline

- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R

Example: Defect-related Yield Loss

- High susceptibility to spot defect-related yield loss, particularly in metallization stages of process
- Most common failure mechanisms: shorts or opens due to extra or missing material between metal tracks
- Design tools fail to realize that values in design manuals are minimum values, not target values
- Spot defect yield loss modeling
  - extremely well-studied field
  - first-order yield prediction: Poisson yield model
  - critical-area model much more successful
  - fatal defect types
Defect-related Yield Loss

Critical Area for Short Circuits

Critical Area for Shorts
Critical Area for Short Circuits

Approaches to Spot Defect Yield Loss

- Modify wire placements to minimize critical area
- Router issue
  - router understands critical-area analyses, optimizations
  - spread, push/shove (gridless, compaction technology)
  - layer reassignment, via shifting (standard capabilities)
  - related: via doubling when available, etc.
- Post-processing approaches in PV are awkward
  - breaks performance verification in layout (if layout has been changed by physical verification)
  - no easy loop back to physical design: convergence problems
Example: Antennas

- Charging in semiconductor processing
  - many process steps use plasmas, charged particles
  - charge collects on conducting poly, metal surfaces
  - capacitive coupling: large electrical fields over gate oxides
  - stresses cause damage, or complete breakdown
  - induced $V_t$ shifts affect device matching (e.g., in analog)

Antennas

- Charging in semiconductor processing
  (Antenna effects refer to permanent gate oxide damage caused by electrical discharges during manufacturing)
- Standard solution: limit the “antenna ratio”
  - antenna ratio $= \frac{(A_{poly} + A_{M1} + \ldots)}{A_{gate-ox}}$
  - e.g., antenna ratio $< 300$
  - $A_{Mx}$? metal ($x$) area electrically connected to node without using metal ($x+1$), and not connected to an active area
Antennas

- General solution == bridging (break antenna by moving route to higher layer)
- Antennas also solved by protection diodes
  - not free (leakage power, area penalties)
- Basically, annoying-but-solved problem
  - not clear whether today’s approaches scale into the future
  - (today, mostly post-processing approaches)

Macroscopic Process Effects

Dummy Fill controls several types of process distortions:

- CMP, SOG
- RIE
- CVD

R. Pack, Cadence
Subwavelength Optical Lithography

- WYSIWYG (layout = mask = wafer) failed starting with 350nm generation
- Optical lithography: feature size limited by diffraction
- Available knobs
  - aperture: OPC
  - phase: PSM

Optical Proximity Correction (OPC)

- Aperture changes to improve process control
  - improve yield (process window)
  - improve device performance

With OPC  No OPC  OPC Corrections

Original Layout
OPC Terminology

Phase Shifting Masks (PSM)

- conventional mask: glass
  - Chrome
  - Phase shifter
  - 0 E at mask 0
  - 0 E at wafer 0
  - 0 I at wafer 0

- phase shifting mask:
Field-Dependent Aberration

Field-dependent aberrations cause placement errors and distortions.

Center: Minimal Aberrations  
Edge: High Aberrations

Design-Manufacturing Interface Changes EDA

- Closely related to foundry capital expenditure
- Unites EDA with much of mask industry, even process development
- Expands scope of physical “verifications”, moves awareness upstream into “syntheses” (logic, layout)
- Very comprehensive changes to data model, infrastructure, flows
- Unified, front-to-back solutions will win
Process Variation Sources

- Design (manufacturing variability) Value
- **Intrinsic variations**
  - **Systematic**: due to predictable sources, can be compensated during design stage
  - **Random**: inherently unpredictable fluctuations and cannot be compensated
- **Dynamic variations**
  - Stem from circuit operation, including supply voltage and temperature fluctuations
  - Depend on circuit activity and hard to be compensated
- **Correlations**
  - Tox and Vth0 are correlated due to
    \[ \frac{|Q|}{T_{ox}} \]
  - Line width and spacing are anti-correlated by one; ILD and interconnect thickness also anti-correlated

Technology Trend Over Generations

<table>
<thead>
<tr>
<th>Technology</th>
<th>100nm</th>
<th>100nm</th>
<th>100nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lefl (µm)</td>
<td>0.10 ± 15%</td>
<td>0.12 ± 15%</td>
<td>0.09 ± 15%</td>
</tr>
<tr>
<td>Tox (V)</td>
<td>40 ± 4%</td>
<td>42 ± 4%</td>
<td>33 ± 4%</td>
</tr>
<tr>
<td>Vth0 (V)</td>
<td>0.40 ± 12.5%</td>
<td>0.42 ± 12.5%</td>
<td>0.27 ± 15.5%</td>
</tr>
<tr>
<td>Resist (Ω)</td>
<td>550 ± 10%</td>
<td>450 ± 10%</td>
<td>450 ± 10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interconnect</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>c (µm)</td>
<td>0.5 ± 3%</td>
<td>3.2 ± 5%</td>
<td>2.8 ± 5%</td>
</tr>
<tr>
<td>w (µm)</td>
<td>0.28 ± 30%</td>
<td>0.80 ± 30%</td>
<td>0.20 ± 30%</td>
</tr>
<tr>
<td>s (µm)</td>
<td>0.28 ± 30%</td>
<td>0.80 ± 30%</td>
<td>0.20 ± 30%</td>
</tr>
<tr>
<td>t (µm)</td>
<td>0.45 ± 10%</td>
<td>1.25 ± 10%</td>
<td>0.45 ± 10%</td>
</tr>
<tr>
<td>ILDh (µm)</td>
<td>0.65 ± 15%</td>
<td>1.80 ± 15%</td>
<td>0.45 ± 15%</td>
</tr>
<tr>
<td>Rvia (Ω)</td>
<td>46 ± 20%</td>
<td>50 ± 20%</td>
<td>54 ± 20%</td>
</tr>
<tr>
<td>W (µm)</td>
<td>61.01</td>
<td>1061</td>
<td>45.19</td>
</tr>
</tbody>
</table>

- Values are from ITRS, BPTM, and industry; red is 3s
- From ongoing work at UCSD/UCB/Michigan; some values may be incorrect (e.g., Rvia)
Variation Sensitivities: Local Stage

- Sensitivity evaluated by the percentage change in performance when there is $3\sigma$ variation at the parameter.
- For local stage, device variations have larger impact on line delay and interconnect variations have stronger impact on crosstalk noise.

Mapping Design to Value (1)

Across-Wafer Frequency Variation
**Mapping Design to Value (2)**

**Goal:** combine charts (1) and (2), drive Design optimizations

**FO4 INV Delays Per Clock Period**

- **FO4 INV** = inverter driving 4 identical inverters (no interconnect)
- Half of frequency improvement came from reducing logic stages
- Other extra performance came from slower Vdd scaling, but this costs too much power
Wire Spacing and Layout Methodology

- Routing tools do not always optimize for spacing
- Stand-alone spacing
  - layout (GDSII/DEF) -> layout (GDSII/DEF)
- Need tight interface to extraction and timing simulation
- Future: built-in extraction and timing estimates

Data Aspects of Post Layout Optimization

- Jogging increases amount of data significantly
- Massive data needs striping
  - minor loss of optimality for large stripes
  - need work across hierarchy
  - fix boundary location, “look” beyond cut-line
  - need propagate net information
- Must support multi-processing for reasonable TAT
Wire Spacing and Shielding

- Pre routing specification
  - convenient, handled by router
  - robust but conservative
  - may consume big area
- Post routing specification
  - area efficient—shield only where needed & have space
  - ease task of router
  - sufficient shielding is not guaranteed
- Either way: definite interactions w/ fill insertion, possible interactions w/ phase-shifting (M1,M2?)

Opportunities for Via Strengthening

- Add cut holes where possible
  - wire widening may need larger/more vias
  - “non square” via cells
- Increase metal-via overhang
  - non uniform overhang
Tradeoffs: Speed / Power / Area
Must compromise and choose between often competing criteria
For given criteria (constraints) on some variables, make best choice for free variables (min cost) => Need to be on boundary of feasible region

Wire spacing example
before spacing
after spacing

Courtesy M. Berkens, DAC99
Many different kinds of delay/area optimization are possible

Many optimizations are somewhat independent

- use several different optimizations. Apply whichever ones are applicable

---

**Optimization at Layout Level**

- Size Transistors
- Space/size wires
- Add/delete buffers
- Modify circuit locally
Transistor Sizing: Area Delay Curve

- Feasible Region
- Infeasible Region
- Optimal Curve: lowest area for a given delay

Min delay → Required Delay → Delay
Min cost → Area

Architecture and application are well matched

- Scenario: Lots of capacitance in wires
  - will it buy me speed: Yes
  - will it save me power: “Yes” (qualified)
Circuits of constant cost $W_1 + W_2 = \text{Cte}$

Circuits of constant delay: 15ns

Faster circuits inside this curve

Note: Actually circuit delay is Posynomial ~ Convex

Transistor Sizing Methods

- **Exact Solutions**
  - gradient Search
  - convex Programming

- **Approximate methods** (very good solutions)
  - iterative improvement on critical path (e.g. TILOS)
Convex Programming Outside Delay Case

- Add more and more bounds
- guess new solution (deep) inside bounds

```
New guess delay is too slow so add new bound: Tangent to curve of equal delay at new guess. New feasible region is to the left (region which contains required delay).
```

Convex Programming Inside Delay Case

- New guess delay is adequate but try and improve cost

```
Add a bound to force search into region of lower cost. New bound is constant cost curve passing through new guess. New feasible region is below new bound.
```
Transistor Sizing: Approximate Solutions

Circuit delay affected only by delay of critical path. Upsize by small amount transistors on crit path with biggest $D1/D2 = \text{improvement/cost}$.
Repeat until timing met

Transistor Sizing: TILOS method

$\eta_2 = \text{slowdown of } T\eta_1 = \text{speedup of } T$

Effective speedup of $T = \eta_1 \cdot \eta_2 = 5$

- Increase $X_{tr}$ on critical path with largest effective speedup per unit area
- effective speedup: $T$
Short Circuit Power Optimization

- Critical path methods miss short circuit power

Critical path

Short circuit power burned in all of these gates due to slow input rise time. Gates not on critical path

- Increase $I_{\text{slow}}$ until capacitive power increase for driving $I_{\text{slow}}$ is more than decrease in S.C. power

- sweep circuit from outputs to inputs

TILOS Optimization Trajectory

Note: Min Size $\neq$ Min Power
Buffer Insertion: Area-delay tradeoffs

- Optimal curve is envelope of curves
- Jump to buffered curve during timing optimization

Local Re-synthesis

- Pass Xtr re-synthesis, logic reorganization
- Gate collapsing

- $T_p$ conducts $\iff$ $N_1$ conducts. Replace $T_p$ with $N_1$
  - Repeat for $P_2$ and $T_n$ for correct NMOS/PMOS
Gate Collapsing: Example

Trade off drive-capability/logic-levels
- Intrinsic Delay
- RC Delay
- reduce number of transistors (area)

Outline
- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R
Custom Methodology in ASIC(?) / COT

- How much is on the table w.r.t. performance?
  - 4x speed, 1/3x area, 1/10x power (Alpha vs. StrongArm vs. “ASIC”)
  - layout methodology spans RTL synthesis, auto P&R, tiling/generation, manual
  - library methodology spans gate array, std cell, rich std cell, liquid lib, ...

- Traditional view of cell-based ASIC
  - Advantages: high productivity, TTM, portability (soft IP, gates)
  - Disadvantages: slower, more power, more area, slow production of std cell library

- Traditional view of Custom
  - Advantages: faster, less power, less area, more circuit styles
  - Disadvantages: low productivity, longer TTM, limited reuse

Custom Methodology in ASIC(?) / COT (Cont’d)

- With sub-wavelength lithography:
  - how much more guardbanding will standard cells need?
  - composability is difficult to guarantee at edges of PSM layouts, when PSM layouts are routed, when hard IPs are made with different density targets, etc.
  - context-independent composability is the foundation of cell-based methodology!

- With variant process flavors:
  - hard layouts (including cells) will be more difficult to reuse

- Relative cost of custom decreases
- On the other hand, productivity is always an issue...
Custom Methodology in ASIC(?) / COT (Cont’d)

- Architecture
  - heavy pipelining
  - fewer logic levels between latches
- Dynamic logic
  - used on all critical paths
- Hand-crafted circuit topologies, sizing and layout
  - good attention to design reduces guardbands
- The last seems to be the lowest-hanging fruit for ASIC

Custom Methodology in ASIC(?) / COT (Cont’d)

- ASIC market forces (IP differentiation) will define needs for xtor-level analyses and syntheses
- Flexible-hierarchical top-down methodology
  - basic strategy: iteratively re-optimize chunks of the design as defined by the layout, i.e., cut out a piece of physical hierarchy, reoptimize it ("peephole optimization")
    - for timing/power/area (e.g., for mismatched input arrival times, slews)
    - for auto-layout (e.g., pin access and cell porosity for router)
    - for manufacturability (density control, critical area, phase-assignability)
  - DOF’s: diffusion sharing, sizing, new mapping / circuit topology sol’s
  - chunk size: as large as possible (tradeoff between near-optimality, CPU time)
Custom Methodology in ASIC(?) / COT (Cont’d)

- antecedents: IBM C5M, Motorola CELLERITY, DEC CLEO
- “infinite library” recovers performance and density that a 300-cell library and classic cell-based flow leave on the table

Custom Methodology in ASIC(?) / COT (Cont’d)

- Supporting belief: characterization and verification are increasingly a non-issue
  - CPUs get faster; size of layout chunks (O(100-1000) xtors) stay same
  - natural instance complexity limits due to hierarchy, layers of interest
  - Compactor-based migration tools are an ingredient?
    - migration perspective can infer too many constraints that aren’t there (consequence of compaction mindset)
    - little clue about integrated performance analyses
Custom Methodology in ASIC(?) / COT (Cont’d)

- Tuners are an ingredient? (size, dual-Vt, multi-supply)
  - limit DOFs (e.g., repeater insertion and clustering, inverter opts
  - cannot handle modern design rules, all-angle geometries
  - not intended to do high-quality layout synthesis
- Layout synthesis is an ingredient?
  - requires optimizations based on detailed analyses
    (routability, signal integrity, manufacturability), transparent links to characterization and verification

Custom Methodology in ASIC(?) / COT (Cont’d)

- “Layout or re-layout on the fly” is an element of performance- and cost-driven ASIC methodology going forward
- “Polygon layout as a DOF in circuit optimization” is a very small step from “polygon layout as a DOF in process migration”
  - designers are already reconciled to the latter
Outline

- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R

Clear Thinking: Basics of Design Convergence

- What must converge?
  - logic, timing, and spatial embedding
  - support front-end signoff, provide predictable back-end

- Ways to achieve Convergence through Predictability
  - correct by construction ("assume, then enforce")
    - constraints and assumptions passed downstream; not much goes upstream
    - ignores concerns via guardbanding
    - separates concerns as able (e.g., FE logic/timing vs. BE spatial embedding)
**Clear Thinking: Basics of Design Convergence (Cont’d)**

- construct by correction (“tight loops”)
  - logic-layout unification; synthesis-analysis unification, concurrent optimization
- elimination of concerns
  - reduced degrees of freedom, pre-emptive design techniques
  - e.g., power distribution, layer assignment / repeater rules, GALS/LIS

**What Must A Design Closure Tool Look Like**

- **Input**
  - RT-level HDL + technology + constraints
- **Output**
  - “go”: recipe for invocation and composition of “commodity” SP&R
  - “no go”: diagnosis of RTL code problems
Logical and physical hierarchies co-evolve

- spatial: top-down coarse placement
- logic/timing: implementable RTL
- limits of human fanout, organizations always have hierarchy

Details (must construct, predict, ignore, eliminate, ...)

- pin optimizations, interconnect planning, hierarchy reconciliations, budgeting mechanisms, compatibility with downstream SP&R, ...

What Must A Design Closure Tool Look Like (Cont’d)

RTL partitioning

- understand interaction b/w block definition and placement quality
- recognize and cure a physically challenged logic hierarchy

Global interconnect planning and optimization

- symbolic route representations to support block plan ECOs

Controllable SP&R back end (including power/clock/scan)

Incremental / ECO optimizations, and optimizations that are “robust” under partial or imperfect design knowledge
Need RTL Planning Technology

- Better estimators (“initial WLMs”)
  - to account for resource, topological heterogeneity
  - to account for optimizations (placement, ripup/reroute, timing)
  - “earliest RTL signoff with detailed P&R knowledge”

Observation: Commoditized SP&R

- RTL-to-GDSII will commoditize SP&R market sectors
- Many solutions are reasonable and will survive in the marketplace
  - RTL-down SP&R becomes a “commodity”
- No solution is complete
- Key missing pieces include RTL partitioning; hierarchy and block management; real working RTL diagnosis and signoff
- Individual point technologies (e.g., global placement or detailed routing) become less valuable
  - integration is most important
Classic Picture

Combining Logical and Physical
Required Advance in Design System Architecture

Yesterday  1000nm

Today  180nm

Tomorrow  50nm

Required Advance in Design System Architecture

Yesterday  1000nm

Today  180nm

Tomorrow  50nm

Required Advance in Design System Architecture

Yesterday  1000nm

Today  180nm

Tomorrow  50nm

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Multiple design files are converged into one efficient Data Model
- Disk accesses are eliminated in critical methodology loops
- Verification of Function, Performance, Testability and other design
criteria all move to earlier, higher levels of abstraction followed by
equivalence checking and
assertion driven design optimizations
- Industry Standard interfaces for data access and control
- Incremental modular tools for optimization and analysis

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools

Planning / Implementation Methodologies (Cont’d)

- Centered on logic design
  - wire-planning methodology with block/cell global placement
  - global routing directives passed forward to chip finishing
  - constant-delay methodology may be used to guide sizing
- Centered on physical design
  - placement-driven or placement-knowledgeable logic synthesis
- Buffer between logic and layout synthesis
  - placement, timing, sizing optimization tools
Planning / Implementation Methodologies (Cont’d)

- Centered on SOC, chip-level planning
  - interface synthesis between blocks
  - communications protocol, protocol implementation decisions guide logic and physical implementation

Performance Optimization Tool Flow
Performance Optimization Methodology

- Design Optimization
  - global restructuring optimization -- logic optimization on layout using actual RC, noise peak values etc.
  - localized optimization -- with no structural changes and least layout impact
  - repeater/buffer insertion for global wires

- Physical optimization
  - high fanout net synthesis (eg. for clock nets); buffer trees to meet delay/skew and fanout requirements
  - automatically determine network topology (# levels, #buffers, and type of buffers)
  - wire sizing, spacing, shielding etc.

- Fixing timing violations automatically
  - fix setup/hold time violations
  - fix maximum slew and fanout violations

Ultra Deep Submicron Timing

Total Delay = \( G_i + G_L + RC_w \)

- \( G_i \) = Intrinsic Gate Delay
- \( G_L \) = Gate Delay from Loading
- \( RC_w \) = Delay from Interconnect Loading

Critical Path Delay
- Electrical Optimization
- Logic Optimization

50K gate Block at 0.18 microns
KEY ISSUE: PREDICTABILITY

- Everything we do is ultimately aimed at a predictable back end (physical implementation after some handoff level of design)
- Predictability == regression models
- Predictability == an enforceable assumption
  - constant-delay paradigm (logical effort, DEC, IBM, ...)
- Predictability == fast constructive prediction
  - RT-level (Tera), gate-level flat full-chip (SPC)
- Predictability == remove the need for predictability
  - GALS, LIS
  - “protocol- / communication-based system-level design”

Problems With Physical Hierarchy

- Physical hierarchy = hierarchical organization of the core layout region
- In general, no relation to high-quality (e.g., w.r.t. timing, routability) embedding of logic
  - artificial physical hierarchy created by top-down placers
  - core region is relatively homogeneous, isotropic: imposing a hierarchy is generally harmful
- Of course, some obvious exceptions
  - regular structures (memories, PLAs, datapaths)
  - hard IP blocks
  - but these don’t fit well in top-down placement anyway
- General trend: non-hierarchical embedding approaches
The Problem With Hierarchies

- Two hierarchies: logical/functional, and physical
  - schematic hierarchy also typical in structured-custom
- RTL design = logical/functional hierarchy
  - provides valuable clues for physical embedding: datapath structure, timing structure, etc.
  - can be incredibly misleading (e.g., all clock buffers in a single hierarchy block)
- Main issues:
  - how to leverage logical/functional hierarchy during embedding
  - when to deviate from designer’s hierarchy
  - methodology for hierarchy reconciliation (buffers, repartitioning / reclustering, etc.)

Interconnect Complexities

- Interconnect effects play a major role in the increasing costs for large hard-block or rectilinear-outline based design styles
- Probabilistic wireload models fail
- Without new capabilities for soft IP design and assembly, interconnect problems will significantly impact performance and cost for emerging IC technologies
Technology Scaling

- Block sizes cannot grow as rapidly as chip sizes since block design becomes increasingly more difficult --- each block is a chip design over multiple configurations.
- If the blocks are inflexible, the global wiring problems begin to dominate all aspects of performance quality and system cost.

[Diagram showing occurrence rate with normalized wirelength and die size ratio]

Soft Blocks

- With soft, flexible blocks, the system assembly can more thoroughly exploit the available technology.
- Interconnect problem is controlled via: soft boundaries for area reshaping; re-synthesis and re-mapping for timing; smart wires; and top-down specified block synthesis.
- Cf. “Amoeba” placement, coloring analysis of “good” placements with respect to original logic hierarchy, etc.

[Diagram showing occurrence rate with normalized wirelength and die size ratio]
**Soft-Block Assembly**

- Hard rectilinear blocks make prediction of global wires extremely difficult
- Top-down constraint-driven assembly of soft fabrics: ability to significantly restructure circuit level blocks during the assembly process helps reach performance goals
  - For example, timing-critical interconnect paths can be completely restructured during assembly without changing any of the system level specification
- Key issue: how to determine the soft blocks in the first place
  - Non-classical partitioning objectives: area sensitivity, functional and clocking structure, critical timing-path awareness, matching capabilities of block placer
  - Block placement: largely unsolved issue
    - Unclear whether packing-centric or connectivity-centric approaches are best

---

**Aristo, DAC-2000**
Cadence, DAC-2000

Partitioning & Log/Phys Mapping
Block Area/Performance Estimation
Block Placement
Inter-block Routing and Buffering
Communication Logic Synthesis
Concurrent Placement, Synthesis
And Route of Cells in Blocks
Finalize Route/Extract/Back Ann.

Constraints complete and block RTLs are feasible
Ensure interblock delays are accounted for
No iterations from here down

Avant!, DAC-2000
“shared algs/data = design closure”

Design Planning
VDSM Physical Synthesis
Place & Route
VDSM Optimization
Equivalence Checking
Final Extraction
Simulation/Analysis
Physical Verification
Mask Synthesis

Design Closure Needs Consistency &
Silicon Accuracy

Extract
Delay Calculation
Analysis
Placement
Routing
Optimization

Capability is unique in the Industry
Actively managing wire delay:
- Through automatic sizing (sizing-driven placement)
- Through buffer insertion

Magma, DAC-2000 “fixed timing”

Don’t: try to accurately adapt a model to reality
- The model might be accurate, the data is generally not...

Instead: Adapt the reality to the model
- Use the simplest appropriate model
- Adapt reality (e.g. cell sizes) to keep model correct.

Don’t: iterate:
- The loops are slow, and affect tool capacity
- Many parameters are optimized simultaneously
- Unclear when (or whether) it converges.

Magma, DAC-2000 “timing closure dos and don’ts”
Magma, DAC-2000 “timing closure dos and don’ts” (Cont’d)

- Instead:
  - Pick a methodology that is correct-by-construction
  - Don’t: bolt together tools using files or ‘databases’
  - Steps do not cooperate and data is often inconsistent.
- Instead: use single data model
  - All design and analysis data simultaneously available.

Synopsys Flow Example

Chip Architect:
- Budgeting
- Estimation
- Floorplanning
- IO placement

FlexRoute:
- Detailed top-level routing

Physical Compiler:
- RTL synthesis & placement together

Micro Controller

Datapath

Analog

Gates

Physical Compiler

Hard IP

Module Compiler:
- Specialized datapath synthesis

Design Compiler:
- RTL synthesis

Chip Architect:
- Coarse routing
- Detailed placement

Detailed standard cell routing:
Cadence, Avant!, proprietary
What is the Right Methodology for SOC

- Will productivity scale adequately relative to available capacity = design complexity?
- Consider:
  - Emerging networking, telecom ICs: >20M gates, <0.11um
  - >80 soft IPs taking more than 65% of IC area
  - >5 large hard IPs (CPU, DSP, DRAM)
  - >200 small hard IPs (SRAM, FIFO, Analog, etc.)
  - >50 clock domains
  - Multiple power supplies
  - High datapath and BIST content

More Radical Methodology Changes are Required

- Flat cell-based is out of capacity
  - Cell abstraction inadequate
- Hierarchical block based is resource-intensive, insufficiently automated
  - Block packing algorithms issues
  - Difficult to automate as we did with cell-based
  - Floorplanning breaks when there are hundreds of blocks
  - Lack of “unified” and meaningful abstractions
  - Lack of network-processing methods similar to those available in the front end (Verilog)
  - Lack of automated solutions for clock, power, test
Future Physical Implementation Platforms

- Where are the cycles?
  - Distributed, heterogeneous, massively parallel platforms
  - Extremely cost-effective (Linux farms, idle desktops, …)
- Where is the productivity lever?
  - By definition, not in “commoditized” design tasks (logic optimization, technology mapping, placement, routing, …)
- Require new platforms and methodologies that decompose and distribute the design optimization problem, without loss of solution quality
  - Typical issues: decoupling of design subproblems, combination of subsolutions into single solution

Outline

- Why CAD
- Technology trends
- What is Physical Design
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of planning-based methodologies
- Implications for P&R
Cell-Based P&R: Classic Context

- **Architecture design**
  - golden microarchitecture design, behavioral model, RT-level structural HDL passed to chip planning
  - cycle time and cycle-accurate timing boundaries established
  - hierarchy correspondences (structural-functional, logical (schematic) and physical) well-established

- **Chip planning**
  - hierarchical floorplan, mixed hard-soft block placement
  - block context-sensitivity: no-fly, layer usage, other routing constraints
  - route planning of all global nets (control/data signals, clock, P/G)
  - induces pin assignments/orderings, hard (partial) pre-routes, etc.

---

Cell-Based P&R: Classic Context

- Individual block design -- various P&R methodologies
- Chip assembly -- possibly implicit in above steps
- What follows: qualitative review of key goals, purposes
Placement Directions

- Global placement
  - engines (analytic, top-down partitioning based, (iterative annealing based) remain the same; all support “anytime” convergent solution
  - becomes more hierarchical
    - block placement, latch placement before “cell placement”
  - support placement of partially/probabilistically specified design
- Detailed placement
  - LEQ/EEQ substitution
  - shifting, spacing and alignment for routability
  - ECOs for timing, signal integrity, reliability
  - closely tied to performance analysis backplane (STA/PV)
  - support incremental “construct by correction” use model

Function of a UDSM Router

- Ultimately responsible for meeting specs/assumptions
  - slew, noise, delay, critical-area, antenna ratio, PSM-amenable ...
- Checks performability throughout top-down physical impl.
  - actively understands, invokes analysis engines and macromodels
- Many functions
  - circuit-level IP generation: clock, power, test, package substrate routing
  - pin assignment and track ordering engines
  - monolithic topology optimization engines
  - owns key DOFs: small re-mapping, incremental placement, device-level layout resynthesis
  - is hierarchical, scalable, incremental, controllable, well-characterized (well-modeled), detunable (e.g., coarse/quick routing), ...
Out-of-Box Uses of Routing Results

- Modify floorplan
  - floorplan compaction, pin assignments derived from top-level route planning
- Determine synthesis constraints
  - budgets for intra-block delay, block input/output boundary conditions
- Modify netlist
  - driver sizing, repeater insertion, buffer clustering
- Placement directives for block layout
  - over-block route planning affects utilization factors within blocks
- Performance-driven routing directives
  - wire tapering/spacing/shielding choices, assumed layer assignments, etc.

Routing Directions

- Cost functions and constraints
  - rich vocabulary, powerful mechanisms to capture, translate, enforce
- Degrees of freedom
  - wire widths/spacings, shielding/interleaving, driver/repeater sizing
  - router empowered to perform small logic resyntheses
- “Methodology”
  - carefully delineated scopes of router application
  - instance complexities remain tractable due to hierarchy and restrictions (e.g., layer assignment rules) that are part of the methodology
Routing Directions

- Change in search mechanisms
  - iterative ripup/reroute replaced by “atomic topology synthesis utilities”: construct entire topologies to satisfy constraints in arbitrary contexts
- Closer alignment with full-/automated-custom view
  - “peephole optimizations” of layout are the natural extensions of Motorola CELLERITY, IBM CM5, etc. methodologies

Summary: Silicon Complexity Challenges

- Silicon Complexity = impact of process scaling, new materials, new device/interconnect architectures
- Non-ideal scaling (leakage, power management, circuit/device innovation, current delivery)
- Coupled high-frequency devices and interconnects (signal integrity analysis and management)
- Manufacturing variability (library characterization, analog and digital circuit performance, error-tolerant design, layout reusability, static performance verification methodology/tools)
- Scaling of global interconnect performance (communication, synchronization)
- Decreased reliability (SEU, gate insulator tunneling and breakdown, joule heating and electromigration)
- Complexity of manufacturing handoff (reticle enhancement and mask writing/inspection flow, manufacturing NRE cost)
Summary: System Complexity Challenges

- System Complexity = exponentially increasing transistor counts, with increased diversity (mixed-signal SOC, …)
- Reuse (hierarchical design support, heterogeneous SOC integration, reuse of verification/test/IP)
- Verification and test (specification capture, design for verifiability, verification reuse, system-level and software verification, AMS self-test, noise-delay fault tests, test reuse)
- Cost-driven design optimization (manufacturing cost modeling and analysis, quality metrics, die-package co-optimization, …)
- Embedded software design (platform-based system design methodologies, software verification/analysis, codesign w/HW)
- Reliable implementation platforms (predictable chip implementation onto multiple fabrics, higher-level handoff)
- Design process management (design team size and geographic distribution, data management, collaborative design support, systematic process improvement)