# Modern Physical Design: Algorithm Technology Methodology (Part I)

Andrew B. Kahng, UCLA
Majid Sarrafzadeh, Northwestern

## Overall Roadmap 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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISOLATED LINES (MPU GATES) (nm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic (Low-Volume—ASIC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usable transistors/cm² (auto layout)</td>
<td>8M</td>
<td>14M</td>
<td>24M</td>
<td>40M</td>
<td>64M</td>
<td>100M</td>
<td>160M</td>
</tr>
<tr>
<td>Nonrecurring engineering cost/usable transistor (microcents)</td>
<td>50</td>
<td>25</td>
<td>15</td>
<td>10</td>
<td>5</td>
<td>2.5</td>
<td>1.3</td>
</tr>
<tr>
<td>Number of Chip I/Os — Maximum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip-to-package (pins) (high-performance)</td>
<td>1515</td>
<td>1867</td>
<td>2553</td>
<td>3492</td>
<td>4716</td>
<td>6532</td>
<td>8935</td>
</tr>
<tr>
<td>Chip-to-package (pins) (cost-performance)</td>
<td>758</td>
<td>934</td>
<td>1277</td>
<td>1747</td>
<td>2386</td>
<td>3269</td>
<td>4470</td>
</tr>
<tr>
<td>Number of Package Pins/Balls — Maximum</td>
<td>568</td>
<td>700</td>
<td>957</td>
<td>1308</td>
<td>1791</td>
<td>2449</td>
<td>3350</td>
</tr>
<tr>
<td>Microprocessor/controller (cost-performance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASIC (high-performance)</td>
<td>1136</td>
<td>1400</td>
<td>1915</td>
<td>2619</td>
<td>3581</td>
<td>4899</td>
<td>6700</td>
</tr>
<tr>
<td>Package cost (cents/pin) (cost-performance)</td>
<td>0.78-2.71</td>
<td>0.70-2.52</td>
<td>0.60-2.16</td>
<td>0.51-1.85</td>
<td>0.44-1.59</td>
<td>0.38-1.36</td>
<td>0.33-1.17</td>
</tr>
<tr>
<td>Power Supply Voltage (V)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum logic Vdd (V)</td>
<td>1.8-2.5</td>
<td>1.6-1.8</td>
<td>1.2-1.5</td>
<td>0.8-1.2</td>
<td>0.6-0.9</td>
<td>0.5-0.6</td>
<td>0.37-0.42</td>
</tr>
<tr>
<td>Maximum Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-performance with heat sink (W)</td>
<td>70</td>
<td>90</td>
<td>130</td>
<td>160</td>
<td>170</td>
<td>175</td>
<td>183</td>
</tr>
<tr>
<td>Battery (W) (Hand-Held)</td>
<td>1.2</td>
<td>1.4</td>
<td>2</td>
<td>2.4</td>
<td>2.8</td>
<td>3.2</td>
<td>3.7</td>
</tr>
</tbody>
</table>
## Overall Roadmap 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>Chip Frequency (MHz)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-chip local clock</td>
<td>750</td>
<td>1250</td>
<td>2100</td>
<td>3500</td>
<td>6000</td>
<td>10000</td>
<td>16503</td>
</tr>
<tr>
<td>(high-performance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-chip, across-chip clock</td>
<td>375</td>
<td>1200</td>
<td>1600</td>
<td>2000</td>
<td>2500</td>
<td>3000</td>
<td>3674</td>
</tr>
<tr>
<td>(high-performance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-chip, across-chip clock</td>
<td>300</td>
<td>500</td>
<td>700</td>
<td>900</td>
<td>1200</td>
<td>1500</td>
<td>1936</td>
</tr>
<tr>
<td>(high-performance ASIC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-chip, across-chip clock</td>
<td>400</td>
<td>600</td>
<td>800</td>
<td>1100</td>
<td>1400</td>
<td>1800</td>
<td>2303</td>
</tr>
<tr>
<td>(cost-performance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip-to-board (off-chip) speed</td>
<td>375</td>
<td>1200</td>
<td>1600</td>
<td>2000</td>
<td>2500</td>
<td>3000</td>
<td>3674</td>
</tr>
<tr>
<td>(high-performance, reduced-width, multiplexed bus)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip-to-board (off-chip) speed</td>
<td>250</td>
<td>480</td>
<td>885</td>
<td>1035</td>
<td>1285</td>
<td>1540</td>
<td>1878</td>
</tr>
<tr>
<td>(high-performance, peripheral buses)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip Size (mm^2) (@sample/introduction)</td>
<td>280</td>
<td>400</td>
<td>560</td>
<td>790</td>
<td>1120</td>
<td>1580</td>
<td>2240</td>
</tr>
<tr>
<td>DRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microprocessor</td>
<td>300</td>
<td>340</td>
<td>430</td>
<td>520</td>
<td>620</td>
<td>750</td>
<td>901</td>
</tr>
<tr>
<td>ASIC (max litho field area)</td>
<td>480</td>
<td>800</td>
<td>900</td>
<td>1000</td>
<td>1100</td>
<td>1300</td>
<td>1482</td>
</tr>
<tr>
<td>Lithographic Field Size (mm^2)</td>
<td>22 x 22</td>
<td>25 x 32</td>
<td>25 x 36</td>
<td>25 x 40</td>
<td>25 x 44</td>
<td>25 x 52</td>
<td>25 x 59</td>
</tr>
<tr>
<td>Maximum Number Wiring Levels</td>
<td>6</td>
<td>6–7</td>
<td>7</td>
<td>7–8</td>
<td>8–9</td>
<td>9</td>
<td>10</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 (multithreshold CMOS, multiple supply voltages, dynamic CMOS)
  - Impact of scaling on power and reliability
Reachability in $t_{\text{crit}} = 80$ ps

25 x 25 mm chip

Technology Scaling Trends

- 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 $\rightarrow$ 100%, ground capacitance $\rightarrow$ 0%
  - 90% in .18u
  - huge signal integrity implications (e.g., guardbands in static analysis approaches)
Ultra Deep Submicron Timing

\[ \text{Total Delay} = G_i + G_L + RC_w \]

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

Critical Path Delay
Electrical Optimization
Logic Optimization

50K gate Block at 0.18 microns

Noise Sources

- Analog design concerns are due 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 made noisiness of digital circuits a big problem
Deep-Submicron Interconnect Complexities

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

Technology (μ) vs. At-Risk Nets (millions)

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
  - thinner deposition reduces cross cap
- 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 technology
  - 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
- Current context: cell-based place-and-route methodology
  - placement and routing formulations, basic technologies
  - methodology contexts
Technical Issues in UDSM Design

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

Why Now?

- 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
- Focus on interconnect
  - susceptible to patterning difficulties
    - CMP, optical exposure, resist development/etch, CVD, ...
  - susceptible to defects
    - critical area, critical volume
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 (two types of short circuits, one type of open)
Critical Area for Short Circuits

Critical Area for Shorts

Critical Area for Short Circuits

Critical Area for Shorts
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
• Standard solution: limit antenna ratio
  • antenna ratio = \( \frac{A_{\text{poly}} + A_{M1} + \ldots}{A_{\text{gate-ox}}} \)
  • 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

• 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
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
- 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, …
- must hit function/cost/TAT points that maximize $/wafer
  - reuse-based methodology
  - need for differentiating IP → **custom**-ization

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

Courtesy M. Berkens, DAC99
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

Wire spacing example

before spacing  after spacing
Performance Optimization Methodology
Design 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

Optimization Methods

- 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

- Area
- Min cost
- Min delay
- Required Delay
- Delay

Feasible Region
- You are here
- You need to be here
- No assignment of sizes can produce a result here

Infeasible Region

Optimal Curve
- lowest area for a given delay

You need to be here

You are here
**Transistor sizing**

**What will it buy me?**

- **Scenario:** Lots of capacitance in wires
  - Will it buy me speed: Yes
  - Will it save me power: “Yes” (qualified)

---

**Transistor Sizing**

**Convexity + Dual Goals**

- Circuits of constant cost $W_1 + W_2 = Cte$
- 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 $D_1/D_2$ = improvement/cost. Repeat until timing met.
Transistor Sizing
TILOS method

- Increase Xtr on critical path with largest per unit effective speedup: $T$

$\delta_1 = \text{speed of } T$
$\delta_2 = \text{slowdown of } T$

Effective speedup of $T = \delta_1 - \delta_2 = 5$

Critical Path

Short Circuit Power Optimization

- Critical path methods miss short circuit power

- 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

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

- Starting Point
- Infeasible Region
- Required delay
- Delay
- Power
- Reduce S.C.
- Fix timing
- Downsize
- Feasible Region

Note: Min Size ≠ Min Power

Buffer Insertion
Area delay tradeoffs

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

Feasible Region
Is the Union of both feasible regions

Area of Min Size buffer

Add buffer at this point

Optimization Trajectory

Area
Delay
Local Re-synthesis

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

\[ T_P \text{ conducts } \iff N_1 \text{ conducts. Replace } T_P \text{ with } N_1 \]
- repeat for \( P_2 \) and \( T_n \) for correct NMOS/PMOS

Gate Collapsing

Example

- Trade off drive-capability/logic-levels
  - Intrinsic Delay \( \rightarrow \) RC Delay
  - reduce number of transistors (area)
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 (e.g. 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

---

Performance Optimization Tool Flow

```
Architectural/ Behavioral Design

RTL Design

Floor Planning

Logic Synthesis

Placement

Optimization

Global Routing

Detailed Routing

Electrical Analysis

Estimation

EM Analysis

Power Analysis

Timing Analysis

Signal Integrity Analysis

Clock Analysis

Extraction
```
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 syn, 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

• 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

- 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

- 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)
- antecedents: IBM C5M, Motorola CELLERTY, DEC CLEO
- “infinite library” recovers performance, density that a 300-cell library and classic cell-based flow leave on the table
Custom Methodology in ASIC(?) / COT

- 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

- 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

- “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
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.

- 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
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.

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), …
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
- 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 CMS, etc. methodologies

Planning / Implementation Methodologies

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