Modern Physical Design:
Algorithm
Technology
Methodology

Andrew B. Kahng UCLA
Majid Sarrafzadeh Northwestern

Introduction

- This tutorial will cover "the latest word" in physical chip implementation methodology and physical design (PD) algorithm technology.
- The target audience consists of
  - system and circuit designers who would benefit from understanding tool capabilities in this arena,
  - CAD engineers (both R&D and support),
  - design project managers,
  - academic researchers.
- Familiarity with basic PD methodology is assumed.

Trade-Off: Depth vs. Breadth

- Broad spectrum of possible material
- Only ~6-7 hours for presentation
- Not all possible topics covered in slides, not all slides covered in talks
  - ask questions if you'd like to hear about something in particular, esp. related to methodology or particular P&R techniques
- All tutorial materials will be available in softcopy at
  - http://vlsicad.cs.ucla.edu/ICCAD99TUTORIAL
  - http://www.ece.nwu.edu/nucad/ICCAD99TUTORIAL

Overview of the Tutorial

- PART I: Technology and Methodology Context Setting (9:00 - 10:00)
- PART II: Fundamental Physical Design Formulation and Algorithms (10:00 - 12:00)
  - Coffee Break (10:30 - 10:45)
  - Lunch (12:00 - 1:00)
- PART III: Interaction with Upstream Floorplanning and Logic Synthesis (1:00 - 2:00)
- PART IV: Interaction with extraction, analysis, and performance validation (2:00 - 3:30)
  - Coffee Break (3:30 - 3:45)
- PART V: Linkage to Custom Layout (3:45 - 4:45)
- Conclusion (4:45 - 5:00)

Modern Physical Design: Algorithm Technology Methodology (Part I)

Andrew B. Kahng UCLA

Majid Sarrafzadeh Northwestern

Outline

- Technology trends
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of classic planning-based methodologies
- Implications for P&R

Overall Roadmap Technology Characteristics

YEAR OF FIRST PRODUCT SHIPMENT	1997	1999	2002	2005	2008	2011	2014
TECHNOLOGY NODE DENSE LINES (DRAM HALF-PITCH) (nm)	250	180	130	100	70	50	35
ISOLATED LINES (MPU GATES) (nm)	200	140	100	70	50	35	25
Logic (Low-Volume—ASIC)‡							
Usable transistors/cm2 (auto layout)	8M	14M	24M	40M	64M	100M	160M
Nonrecurring engineering cost /usable transistor (microcents)	50	25	15	10	5	2.5	1.3
Number of Chip I/Os – Maximum							
Chip-to-package (pads) (high-performance)	1515	1867	2553	3492	4776	6532	8935
Chip-to-package (pads) (cost-performance)	758	934	1277	1747	2386	3268	4470
Number of Package Pins/Balls – Maximum							
Microprocessor/controller (cost-performance)	568	700	957	1309	1791	2449	3350
ASIC (high-performance)	1136	1400	1915	2619	3581	4898	6700
Package cost (cents/pin) (cost-performance)	0.78-2.71	0.70-2.52	0.60-2.16	0.51-1.85	0.44-1.59	0.38-1.36	0.33-1.17
Power Supply Voltage (V)							
Minimum logic Vdd (V)	1.8–2.5	1.5–1.8	1.2–1.5	0.9–1.2	0.6–0.9	0.5–0.6	0.37-0.42
Maximum Power							
High-performance with heat sink (W)	70	90	130	160	170	175	183
Battery (W)—(Hand-held)	1.2	1.4	2	2.4	2.8	3.2	3.7

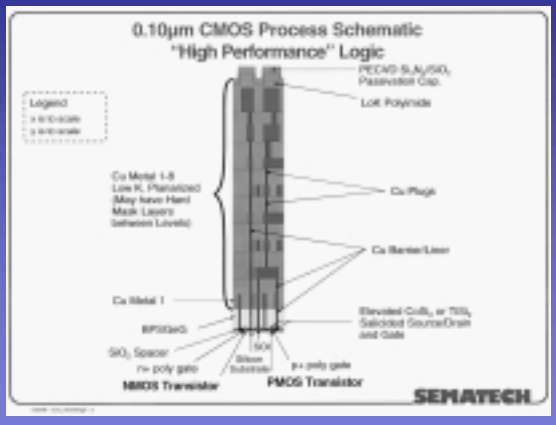
Overall Roadmap Technology Characteristics (Cont'd)

YEAR OF FIRST PRODUCT SHIPMENT	1997	1999	2002	2005	2008	2011	2014
TECHNOLOGY NODE DENSE LINES (DRAM HALF-PITCH) (nm)	250	180	130	100	70	50	35
Chip Frequency (MHz)							
On-chip local clock (high-performance)	750	1250	2100	3500	6000	10000	16903
On-chip, across-chip clock (high-performance)	375	1200	1600	2000	2500	3000	3674
On-chip, across-chip clock (high-performance ASIC)	300	500	700	900	1200	1500	1936
On-chip, across-chip clock (cost-performance)	400	600	800	1100	1400	1800	2303
Chip-to-board (off-chip) speed (high-performance, reduced-width, multiplexed bus)	375	1200	1600	2000	2500	3000	3674
Chip-to-board (off-chip) speed (high-performance, peripheral buses)	250	480	885	1035	1285	1540	1878
Chip Size (mm2) (@sample/introduction)							
DRAM	280	400	560	790	1120	1580	2240
Microprocessor	300	340	430	520	620	750	901
ASIC [max litho field area]	480	800	900	1000	1100	1300	1482
Lithographic Field Size (mm2)	22 x 22 484	25 x 32 800	25 x 36 900	25 x 40 1000	25 x 44 1100	25 x 52 1300	25 x 59 1482
Maximum Number Wiring Levels	6	6–7	7	7–8	8–9	9	10

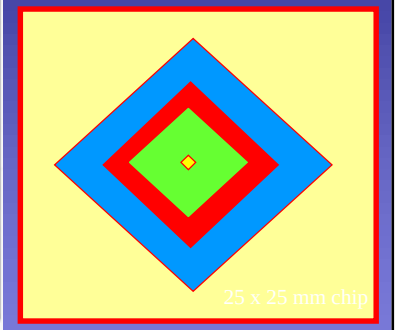
Technology Scaling Trends

- 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

Technology Scaling Trends



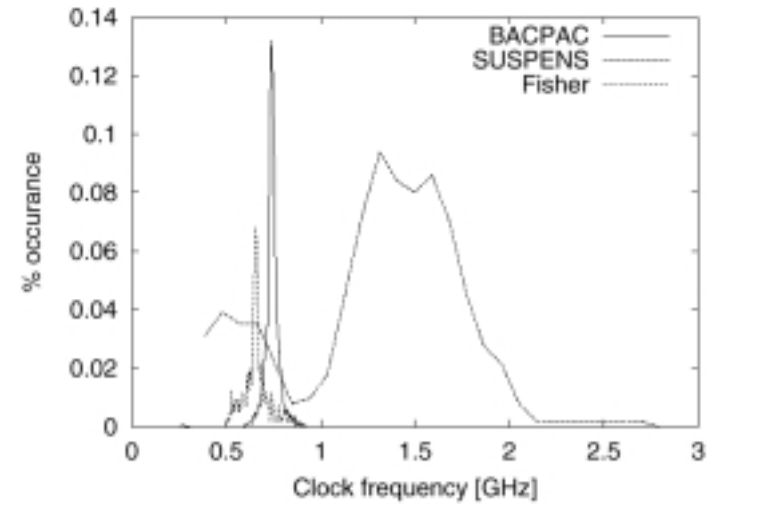
Reachability in τ_{crit} = 80 ps



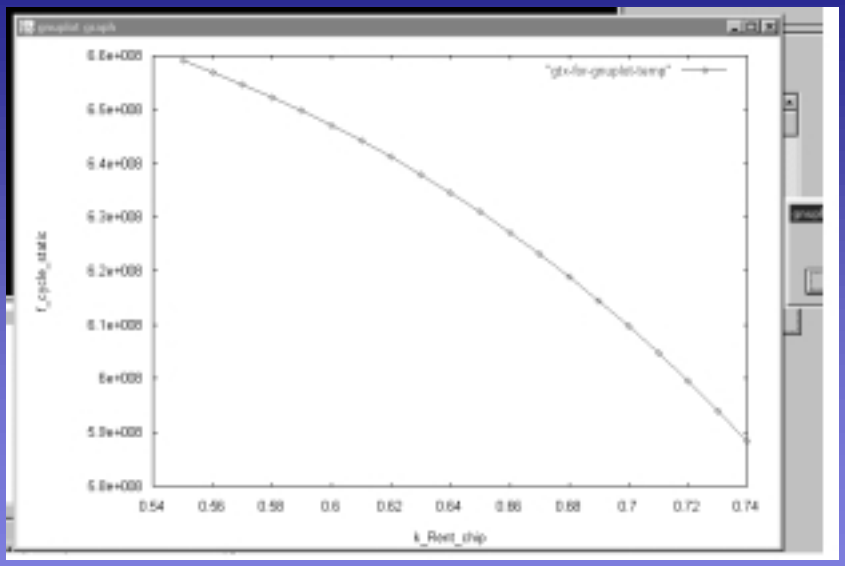
Technology Scaling Trends

- Scaling of x0.7 every three years
  - .25u .18u .13u .10u .07u .05u
  - 1997 1999 2002 2005 2008 2011
  - 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 .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

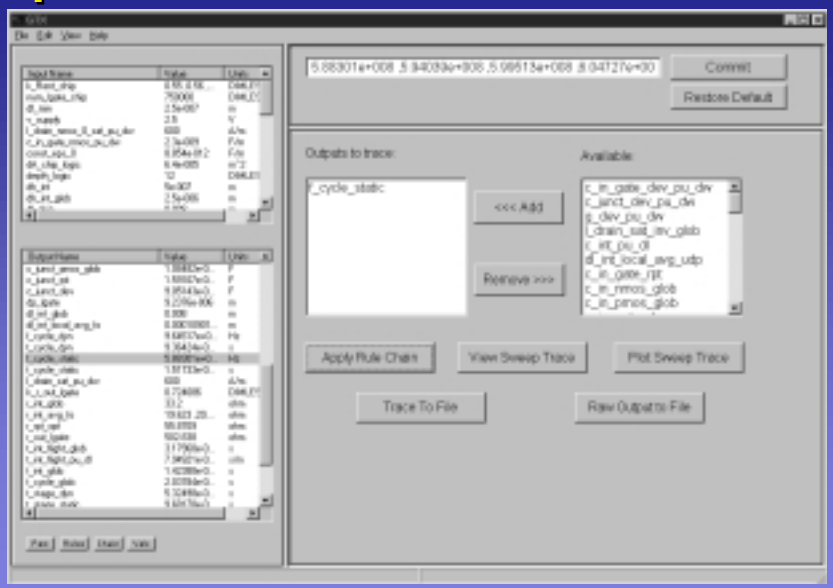
Technology Extrapolation Sensitivity

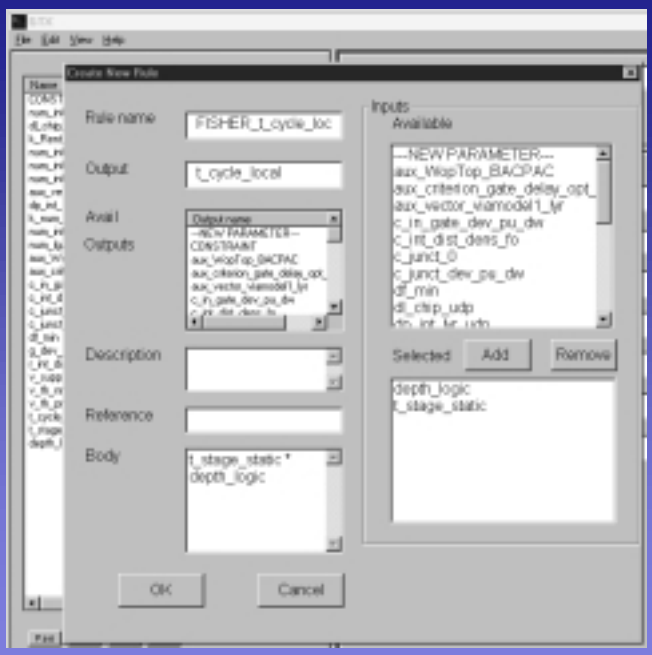


Technology Extrapolation Sensitivity



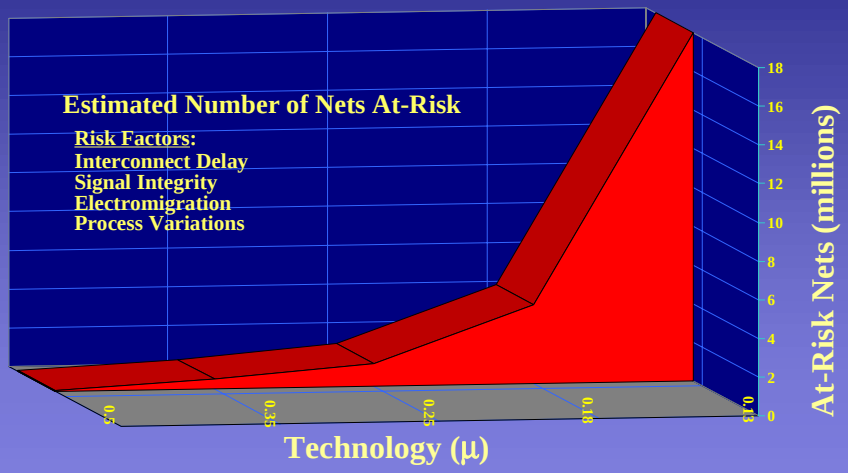
http://vlsicad.cs.ucla.edu/GSRC/GTX/





15 ICCAD Tutorial: November 11, 1999 © Andrew B. Kahng Majid Sarrafzadeh

Deep-Submicron Interconnect Complexity



16 Courtesy Hormoz/Muddu, ASIC99 ICCAD Tutorial: November 11, 1999 © Andrew B. Kahng Majid Sarrafzadeh

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
- 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

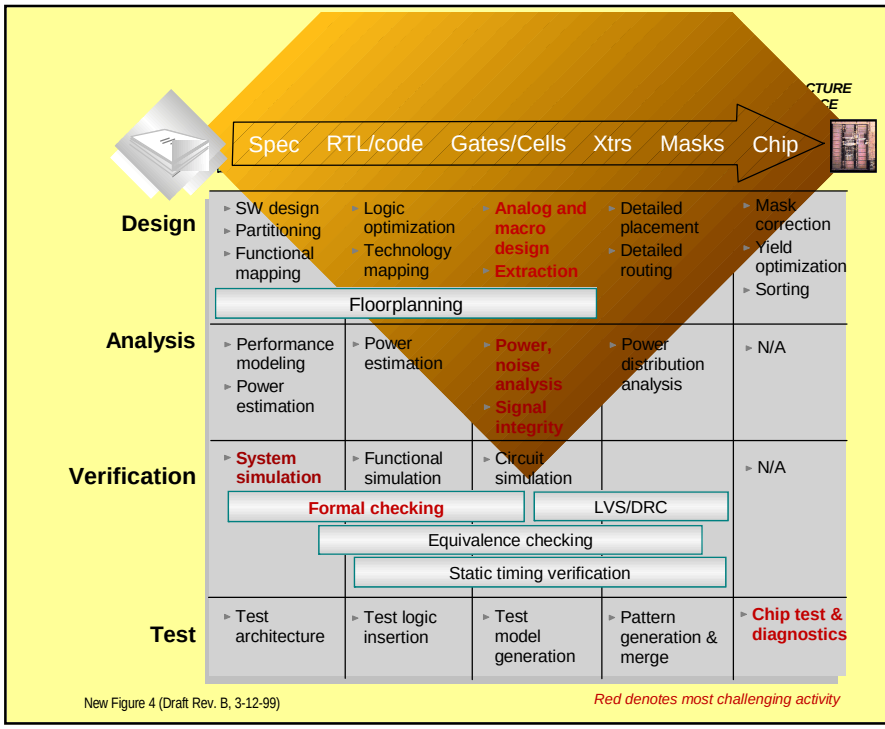
- New issues and problems arising in UDSM technology
  - catastrophic yield: critical area, antennas
  - parametric yield: density control (filling) for CMP
  - parametric yield: subwavelength lithography implications
    - optical proximity correction (OPC)
    - phase-shifting mask design (PSM)
  - 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?

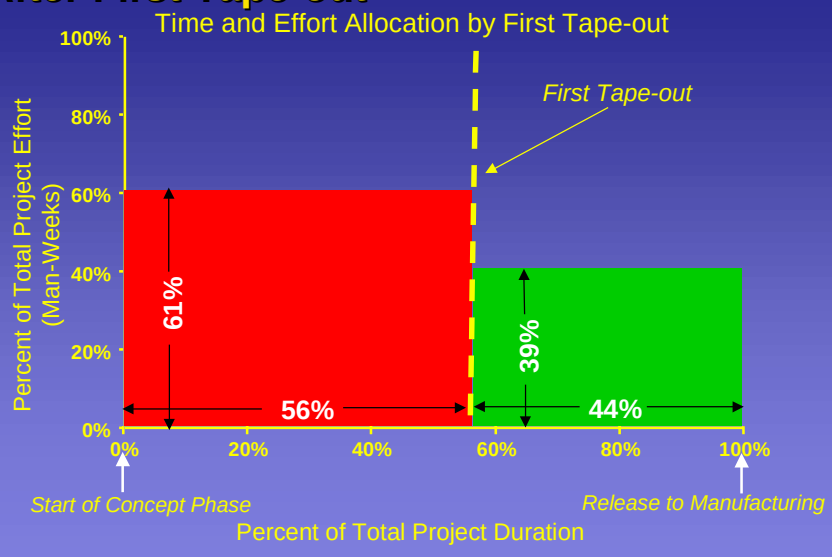
- These effects have always existed, but become worse at UDSM sizes because of:
  - finer geometries
    - greater wire and via resistance
    - higher electric fields if supply voltage not scaled
  - more metal layers
    - higher ratio of cross coupling to grounded capacitance
  - lower supply voltages
    - 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



New Figure 4 (Draft Rev. B, 3-12-99)

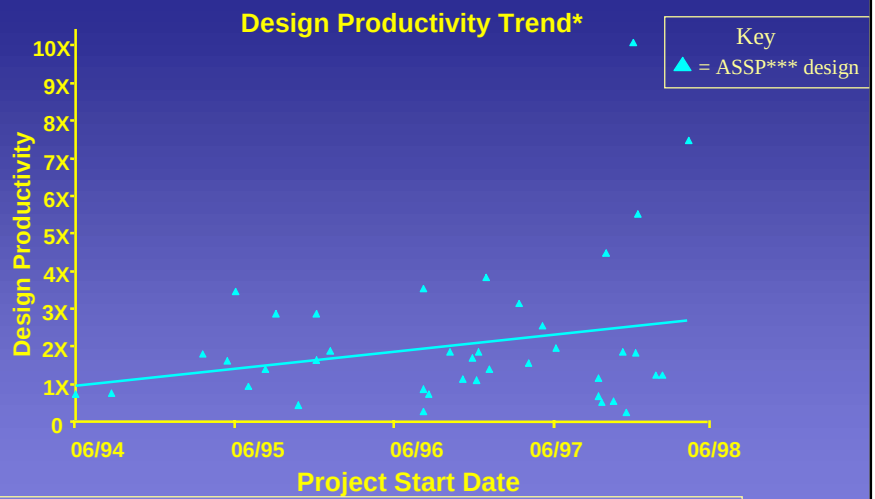
Red denotes most challenging activity

ASSP: 44% WallTime, 39% Total Effort After 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



* 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
- 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

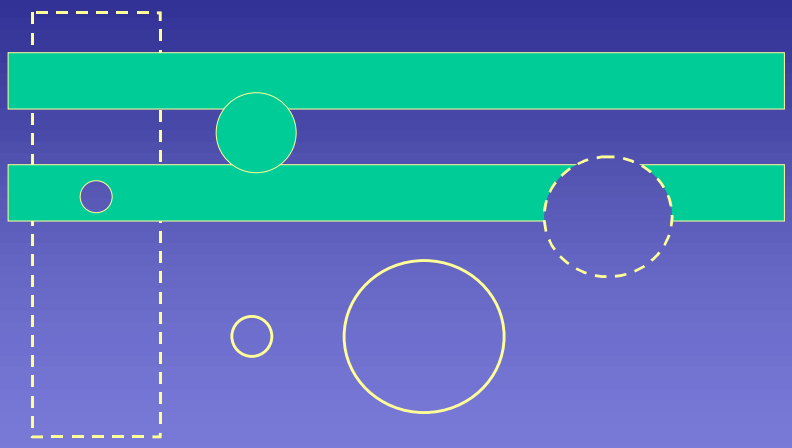
Outline

- Technology trends
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of classic planning-based methodologies
- Implications for P&R

Example: Defect-related Yield Loss

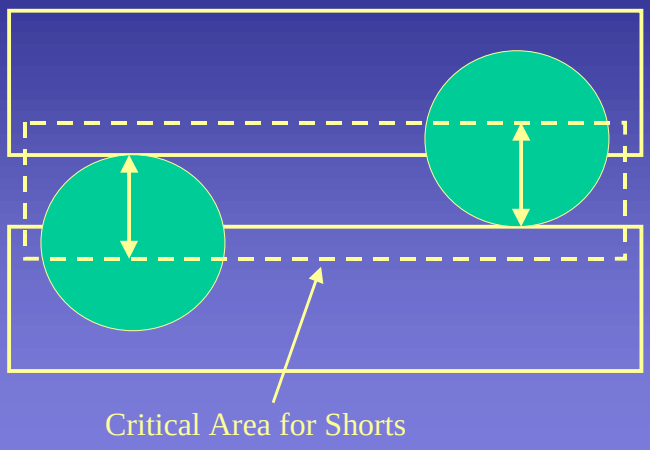
- High susceptibility to spot defect-related yield loss, particularly in metallization stages of process
- Most common failure mechanisms: shorts or opens due to extra or missing material between metal tracks
- Design tools fail to realize that values in design manuals are minimum values, not target 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** (two types of short circuits, one type of open)

Defect-related Yield Loss

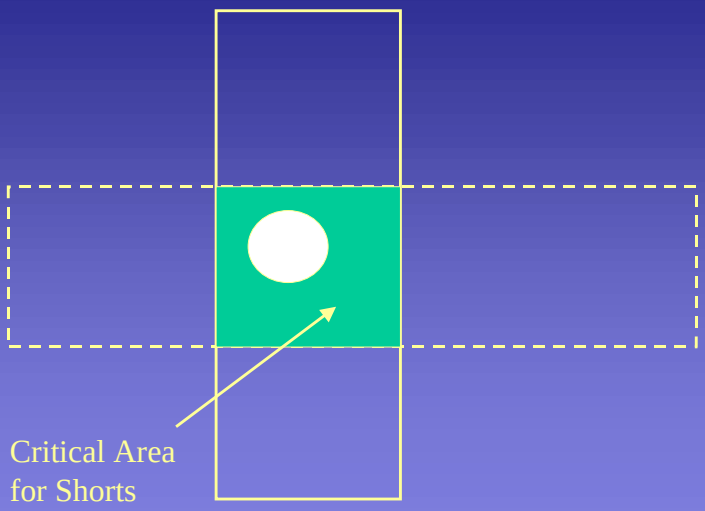


fatal defect types (two types of short circuits, one type of open)

Critical Area for Short Circuits



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
- Standard solution: limit antenna ratio
  - antenna ratio = (A_{poly} + A_{M1} + ...) / $A_{gate\text{-}ox}$
  - 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

- Charging in semiconductor processing
- Standard solution: limit antenna ratio
- 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)

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

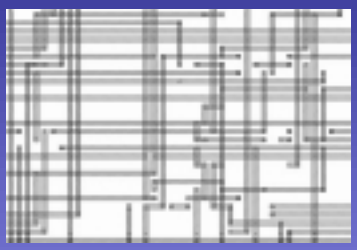
Wire Spacing and Shielding

- 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?)

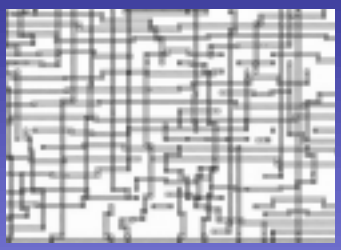
Opportunities for Via Strengthening

- Add cut holes where possible
  - wire widening may need larger/more vias
  - “non square” via cells
- Increase metal-via overhang
  - non uniform overhang

Wire spacing example



before spacing



after spacing

Courtesy M. Berkens, DAC99

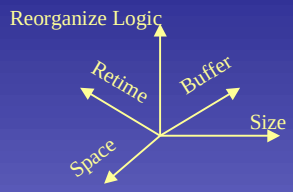
Outline

- Technology trends
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of classic planning-based methodologies
- Implications for P&R

Performance Optimization Methodology

- 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

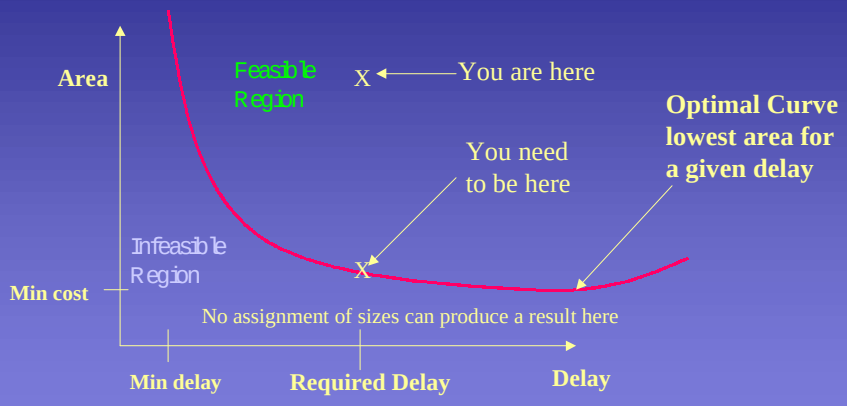


- Many different kinds of delay/area optimization are possible
- Many optimizations are somewhat independent
  - use several different optimizations. Apply whichever ones are applicable

Optimization at Layout Level

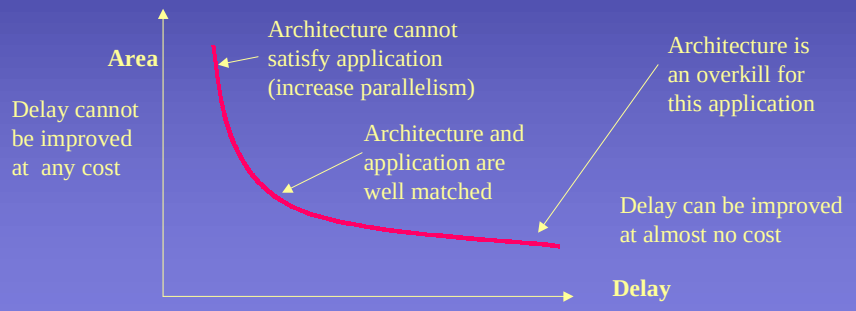
- Size Transistors
- Space/size wires
- Add/delete buffers
- Modify circuit locally

Transistor Sizing
Area Delay Curve

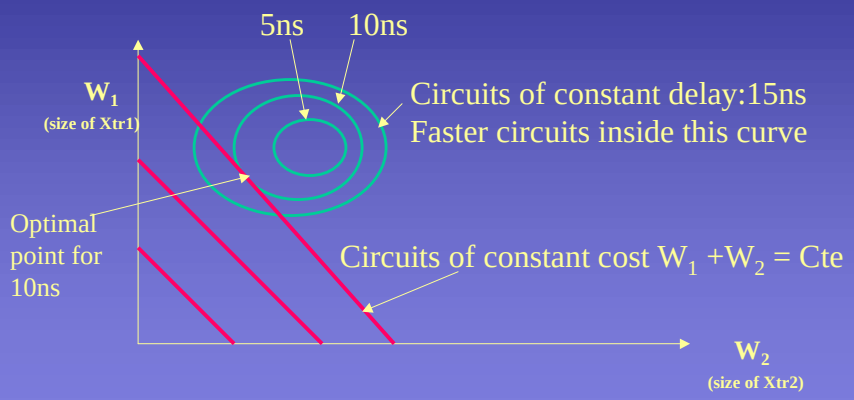


Transistor sizing
What will it buy me?

- Scenario: Lots of capacitance in wires
  - will it buy me speed: Yes
  - will is save me power: "Yes" (qualified)



Transistor Sizing
Convexity + Dual Goals



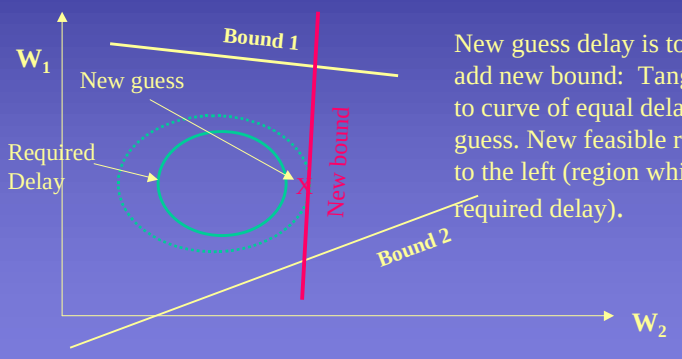
Note: Actually circuit delay is Posynomial ~ Convex

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

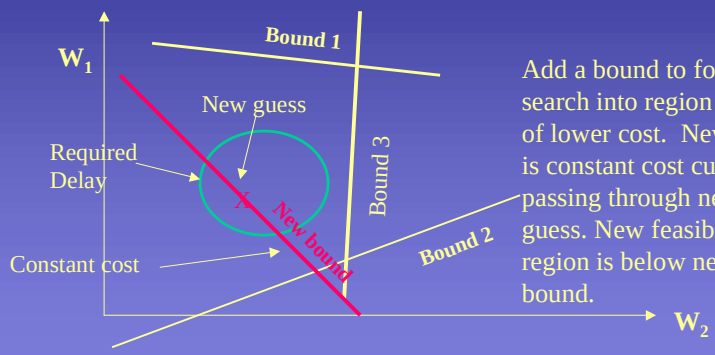
- Add more and more bounds
  - guess new solution (deep) inside bounds



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).

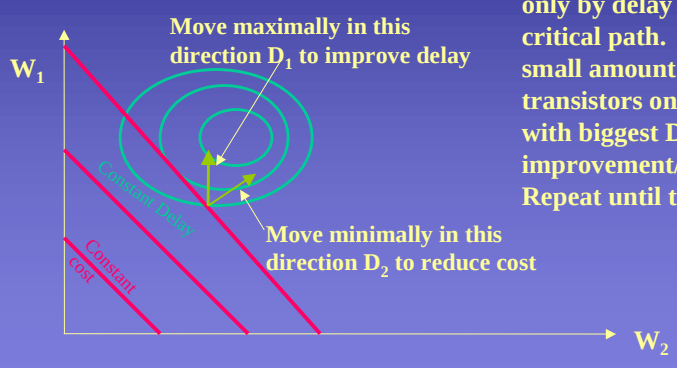
Convex Programming
Inside Delay Case

- New guess delay is adequate but try and improve cost



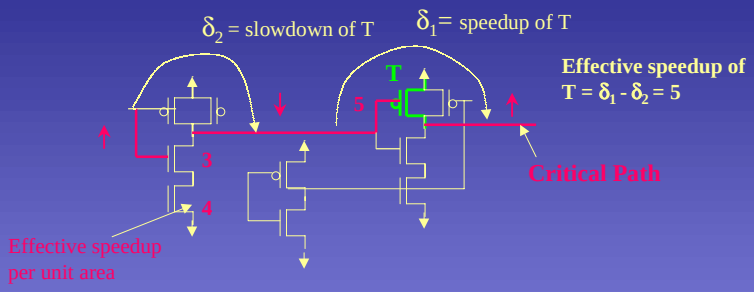
Add a bound to force search into region of lower cost. New bound is constant cost curve passing through new guess. New feasible region is below new bound.

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 = improvement/cost. Repeat until timing met

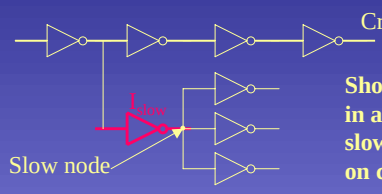
Transistor Sizing
TILOS method



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

Short Circuit Power Optimization

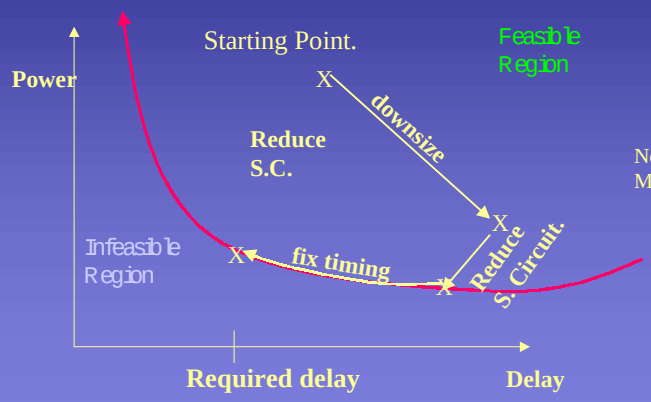
- Critical path methods miss short circuit power



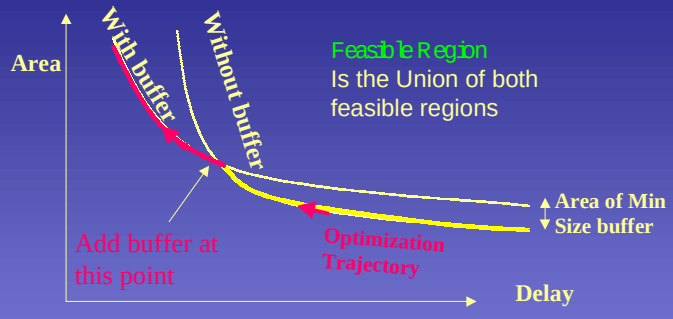
Short circuit power burned in all of these gates due to slow input rise time. Gates not on critical path

- Increase I_{slow} until capacitive power increase for driving I_{slow} is more than decrease in S.C. power
  - sweep circuit from outputs to inputs

TILOS Optimization Trajectory



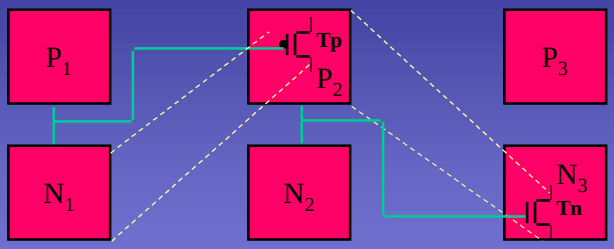
Buffer Insertion
Area delay tradeoffs



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

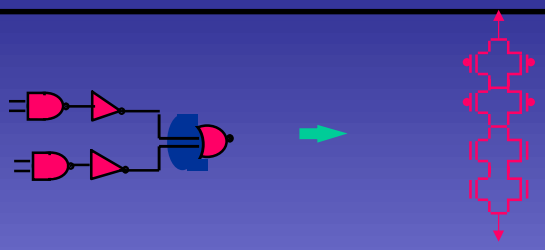
Local Re-synthesis

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



- T_P conducts <=> 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

- Technology trends
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of classic 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 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 CELLERITY, 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

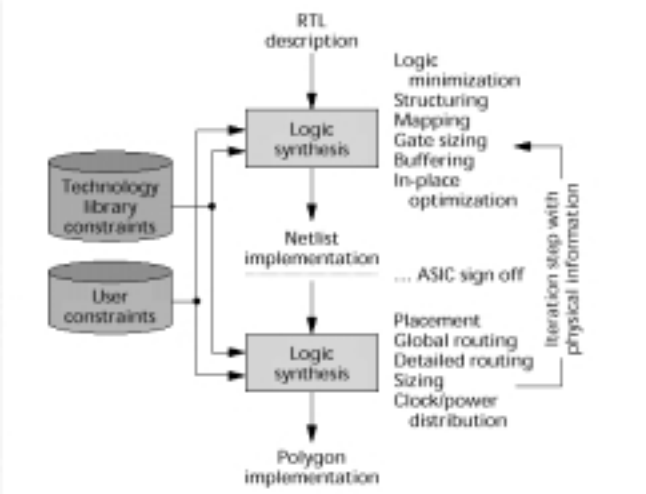
Custom Methodology in ASIC(?) / COT

- "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

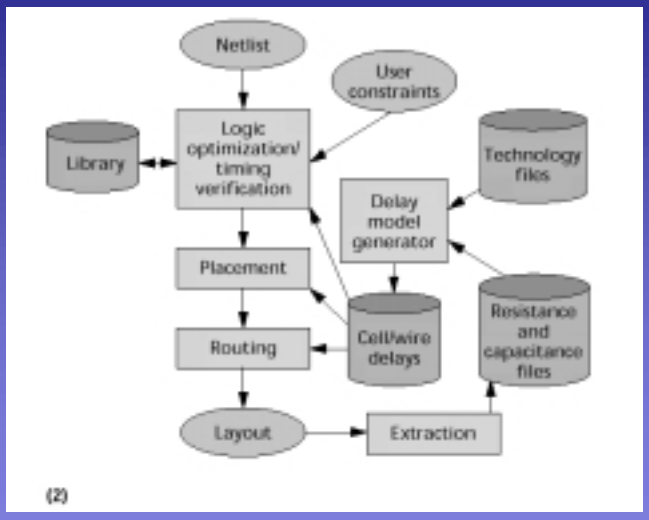
Outline

- Technology trends
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of classic planning-based methodologies
- Implications for P&R

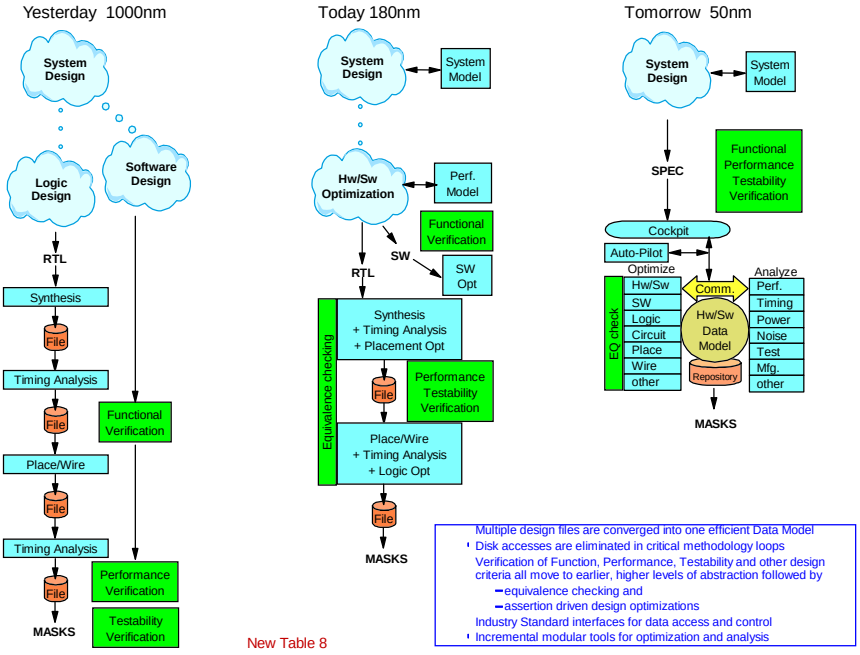
Sylvester-Keutzer: Classic Picture



Sylvester-Keutzer: Combining Logical and Physical



Required Advance in Design System Architecture



New Table 8

- 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

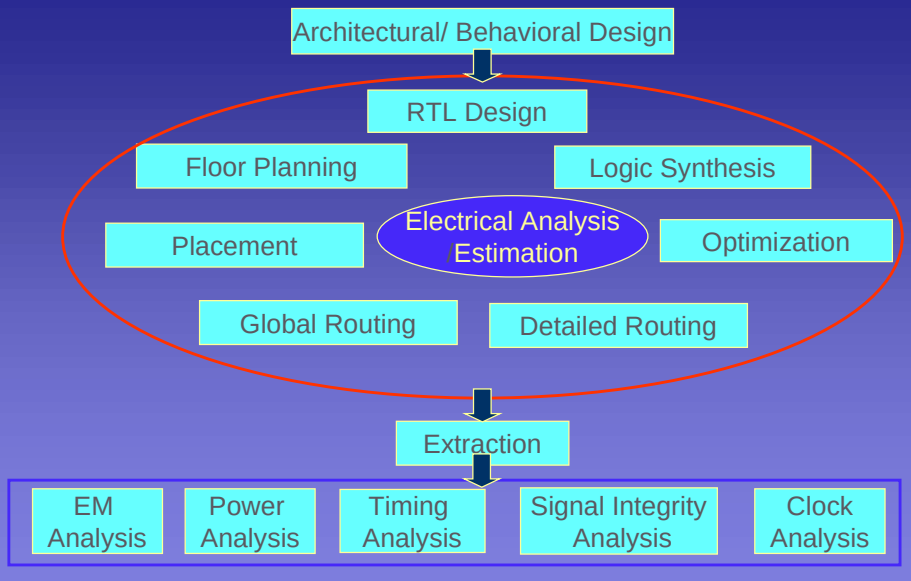
Planning / Implementation Methodologies

- 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

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

Performance Optimization Tool Flow

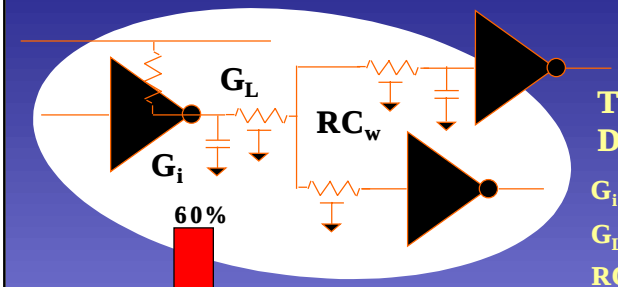


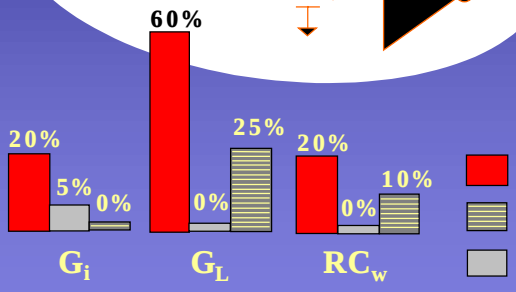
Courtesy Hormoz/Muddu, ASIC99

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 Loa

RC_w= Delay from Interconn 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, estimatable 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

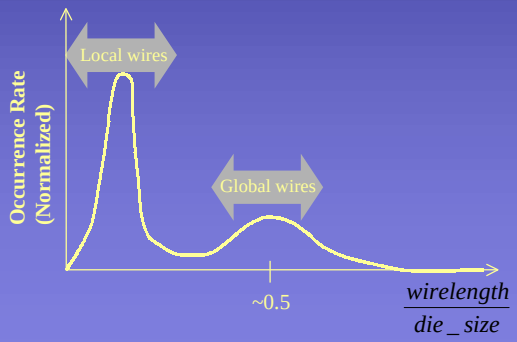
- Physical hierarchy = hierarchical organization of the core layout region
- In general, no relation to high-quality (e.g., w.r.t. timing, routability) embedding of logic
  - artifactual physical hierarchy created by top-down placers
  - core region is relatively homogeneous, isotropic: imposing a hierarchy 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

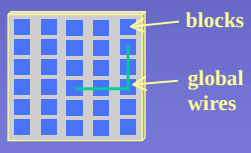
The Problem With Hierarchies

- 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

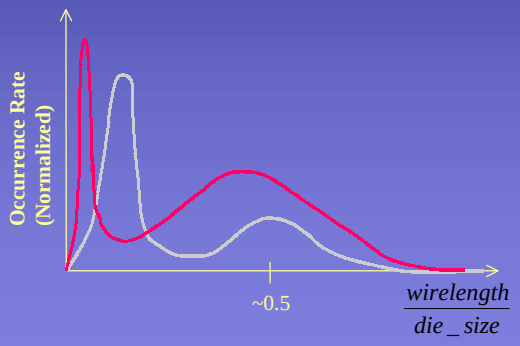


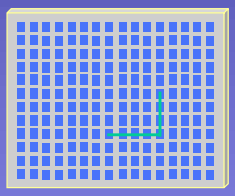


Courtesy Pileggi, MARCO GSRC

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

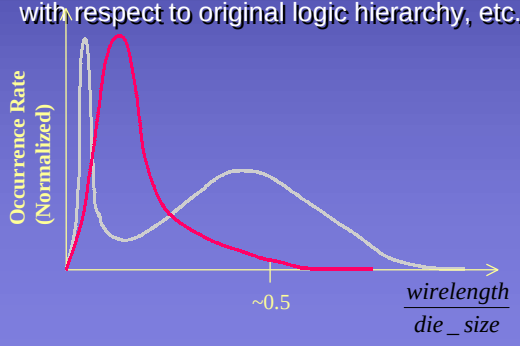


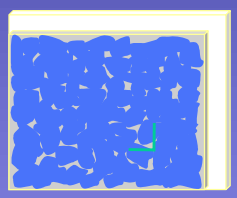


Larger chip with finer feature sizes

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 re-shaping; 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.





Superior timing, power and cost

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

Outline

- Technology trends
- Post-layout optimization methodologies
  - manufacturability and reliability
  - performance
- Custom or custom-on-the-fly methodologies
- Flavors of classic 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.
- 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
- 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

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 hey are inherently immune to noise
- Technology scaling and performance demands made noisiness of digital circuits a big problem