Generalized Constraint Generation for Analog Circuit Design

Edoardo Charbon, Enrico Malavasi*,
Alberto Sangiovanni-Vincentelli

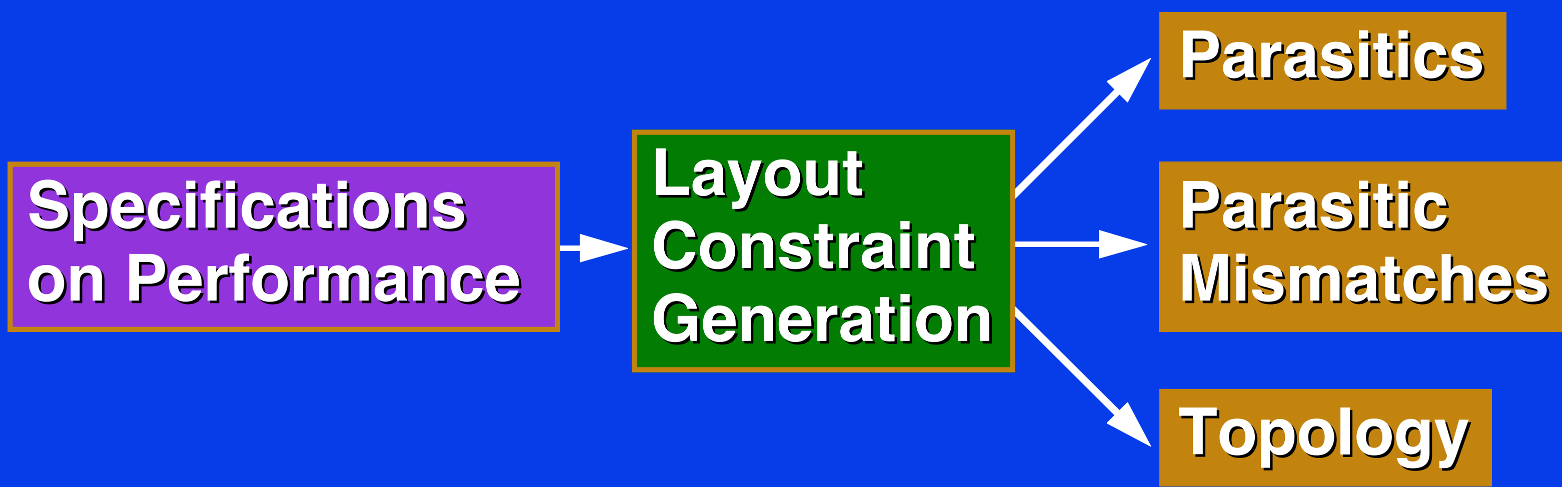
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley

* Dipartimento di Elettronica e Informatica
Università di Padova, Italy

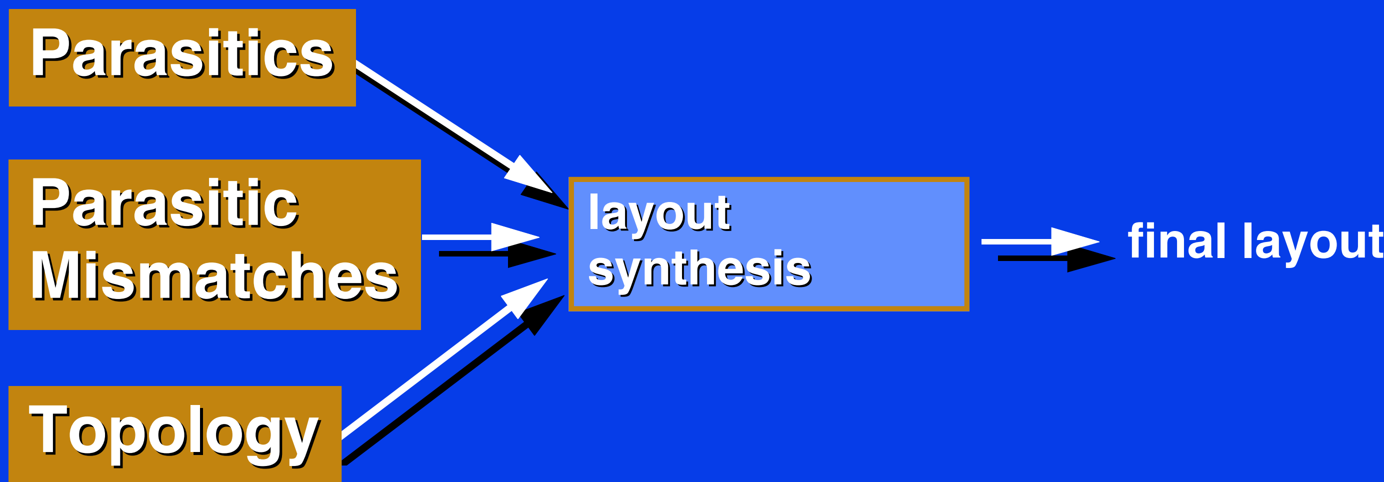
Outline

- Objective and Background
- Constraint Generation Problem
- Modeling Performance Dependence on Parasitics
- Modeling Performance Dependence on Technology
- Results
- Summary

Objective


Specifications on Performance
Layout Constraint Generation
Parasitics
Parasitic Mismatches
Topology

Objective

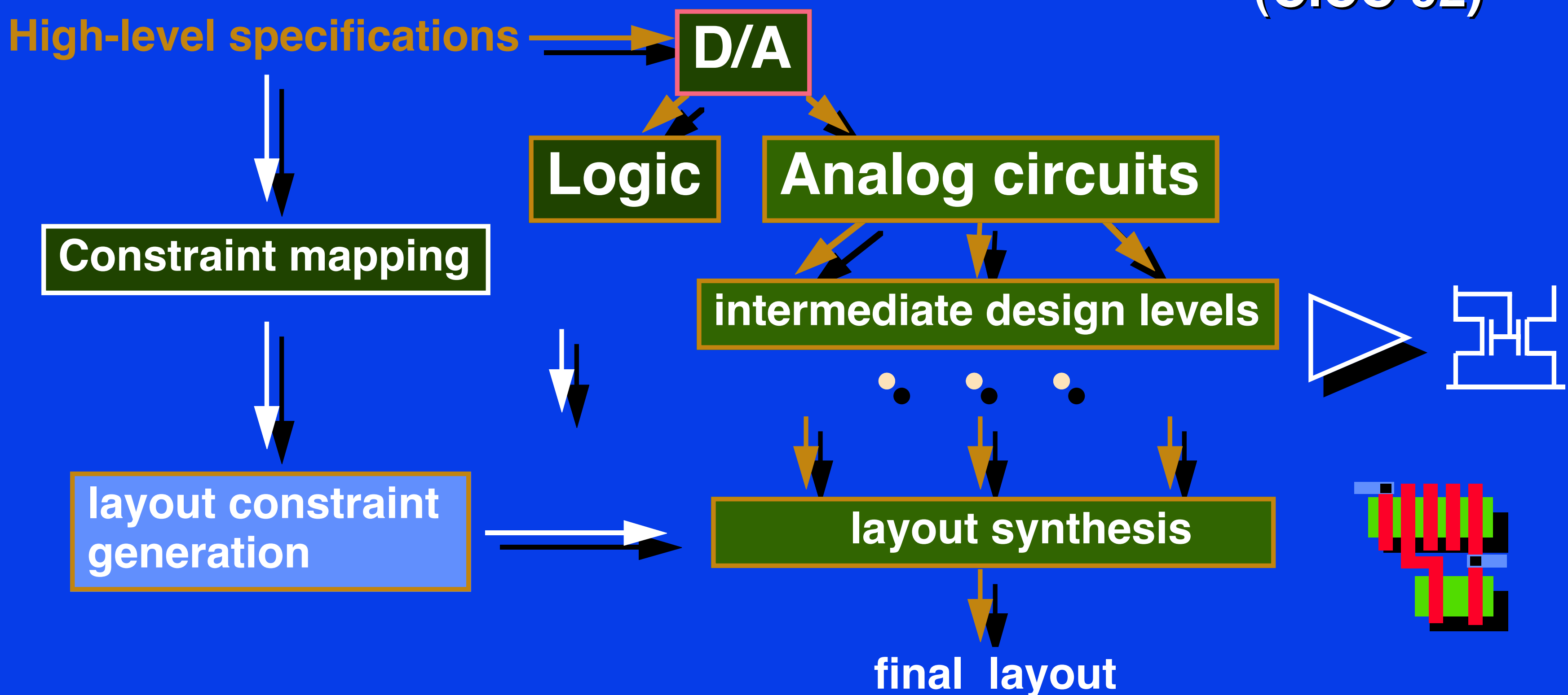


Constraints should

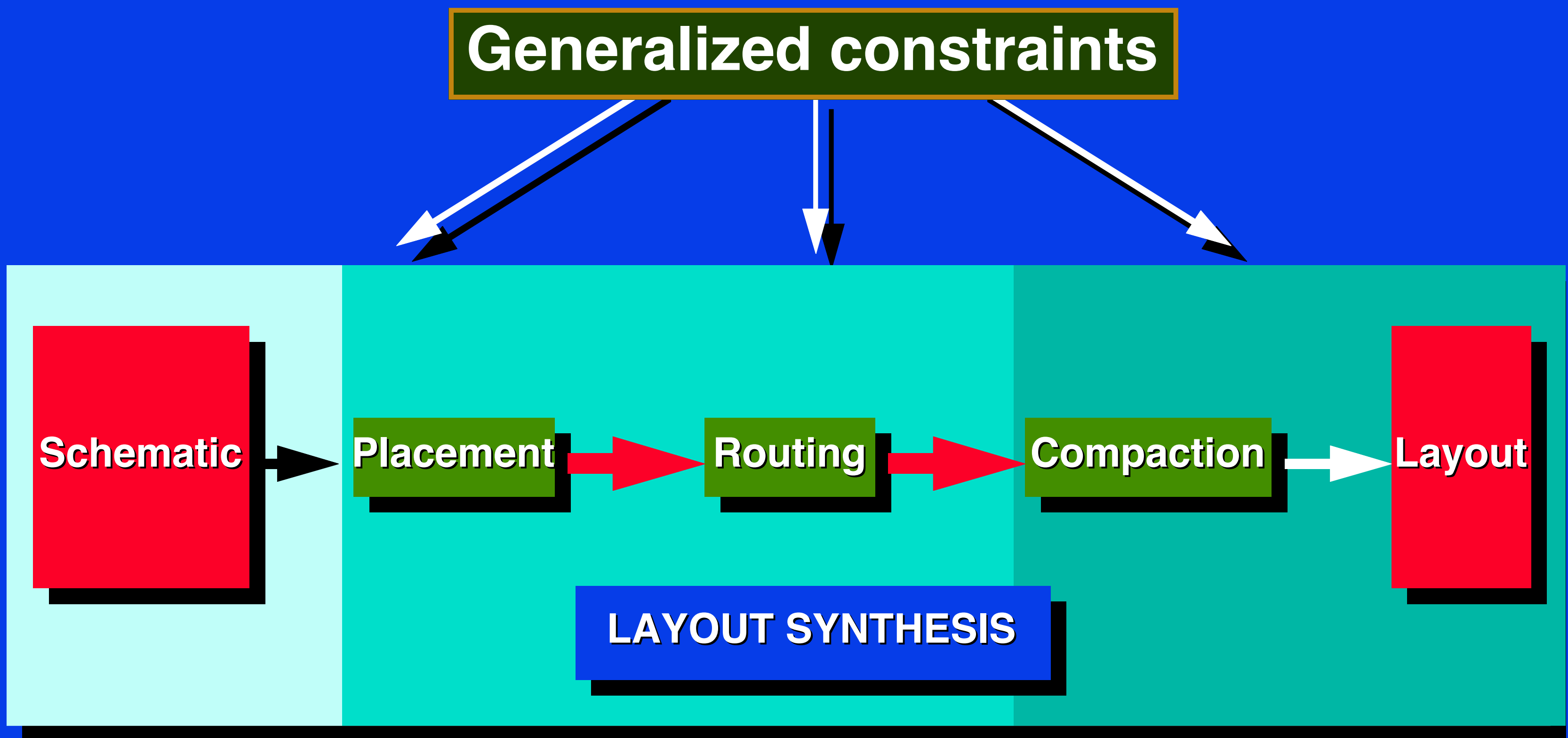
- be general enough to be used at each stage of layout synthesis
- allow maximum flexibility of each layout tool

Background

- Layout synthesis fits in top-down, constraint-driven design (CICC 92)



Use of Layout Constraints



Outline

- Objective and Background
- Constraint Generation Problem
- Modeling Performance Dependence on Parasitics
- Modeling Performance Dependence on Technology
- Results
- Summary

Constraint Generation Problem

Classes of layout constraints

- **Parasitic constraints =**
  Bounds on the absolute value of parasitics

- **Matching constraints =**
  Bounds on the mismatch of parasitics

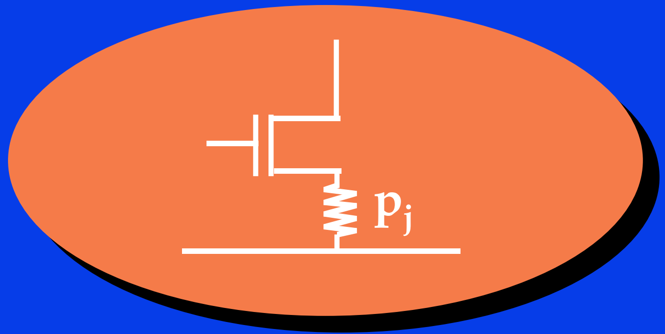
- **Topological constraints =**
  Constraints on the relative position and orientation of layout objects (device matching and symmetry)

Parasitic Constraints

Problem formulation (Choudhury 91)

- Given a set of performances $\{W_i\}$, $i = 1, \ldots N_w$ and a set of parasitics $\{p_j\}$, $j = 1, \ldots N_p$
- Find suitable $p_j^{(bound)}$, s.t.

  if $p_j \le p_j^{(bound)}$, for all j



- Then it is guaranteed that

  $\Delta W_i \le \Delta W_i^{(max)}$, for all i

where $\Delta W_i^{(max)}$ is the maximum degradation ΔW_i of performance W_i .

Matching Constraints

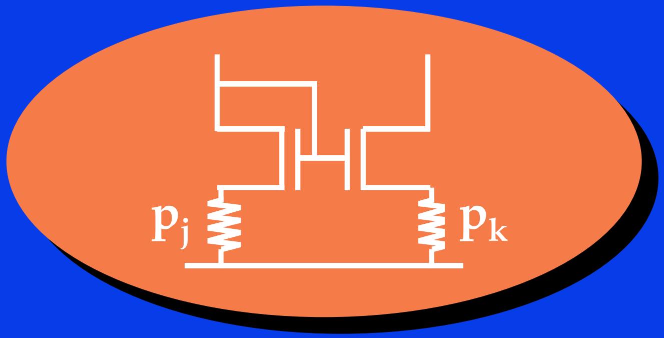
Problem formulation

- Find a suitable $\Delta\, p_j^{(neg\_mismatch)}$ and $\Delta\, p_j^{(pos\_mismatch)}$, s.t.

  if $\Delta\, p_j^{(neg\_mismatch)} \leq \Delta\, p_j \leq \Delta\, p_j^{(pos\_mismatch)}$, for all j

- Then it is guaranteed that

  $\Delta\, W_i \leq \Delta\, W_i^{(max)}$, for all i



Topological Constraints

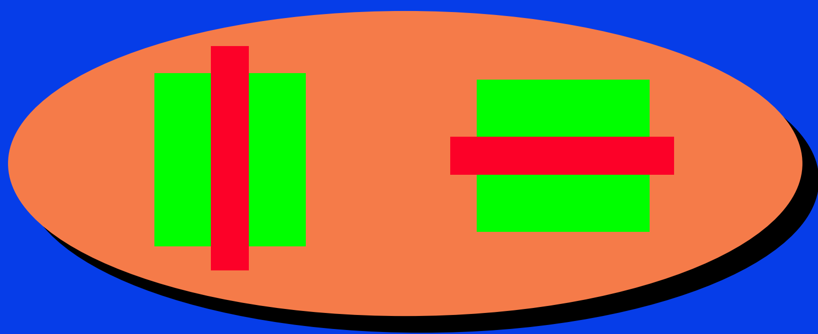
Problem formulation

- find a suitable distance $d_{jk}^{(bound)}$ and a relative orientation $r_{jk}^{(bound)}$ between device j and k, s.t.

  if $d_{jk} \leq d_{jk}^{(bound)}$, $r_{jk} \leq r_{k}^{(bound)}$ for all j,k

- Then it is guaranteed that

  $\Delta W_i \leq \Delta W_i^{(max)}$, for all i



Outline

- Objective and Background
- Constraint Generation Problem
- Modeling Performance Dependence on Parasitics
- Modeling Performance Dependence on Technology
- Results
- Summary

Modeling Performance Dependence on Parasitics

Effect of parasitics on performance

- Linearize performance around its nominal value

$$\Delta W^{+} = \sum_j S_j^{+} p_j \quad \text{and} \quad \Delta W^{-} = \sum_j S_j^{-} p_j$$

- $\Delta W^{+/-}$ = pos. / neg. performance degradation
- $S_j^{+/-}$ pos. / neg. sensitivity of W wrt p_j
- +/- dropped for simplicity from now on but always implied

Modeling Performance Dependence on Parasitics

Effect of parasitic mismatch on performance

- For a set of n mismatched parasitics

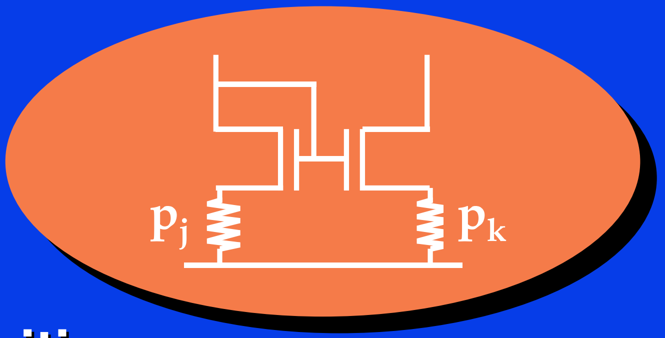
  Nominal value: $p = (1/n) \sum_j p_j$

  Mismatch between p_j and p_k :

  $\Delta p_{jk} = p_j - p_k$

- Then

  $\Delta W = S_p \, p + \sum_{(j,k)} S_{\Delta pjk} \, \Delta p_{jk}$



- S_p : sensitivity of W wrt nominal parasitic p_j
- $S_{\Delta pjk}$: sensitivity of W wrt mismatch Δp_{jk}
- S_p and $S_{\Delta pjk}$ obtained by simple manipulations of original sensitivities

Generating Parasitic Constraints

Range of parasitic component

$$p_j^{(min)} \le p_j \le p_j^{(max)}$$

Range of parasitic mismatch

$$p_j^{(min)} - p_j^{(max)} \le \Delta p_j \le p_j^{(max)} - p_j^{(min)}$$

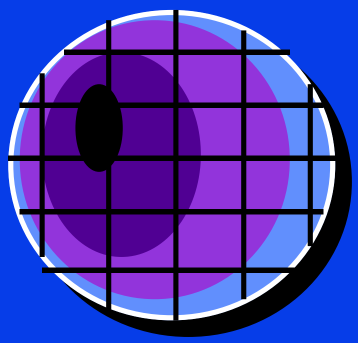
- **Non critical** parasitics eliminated based on their negligible cumulative effect on performance
- **Critical** parasitics automatically generated using constrained optimization
- **Objective:** maximize flexibility of layout tools

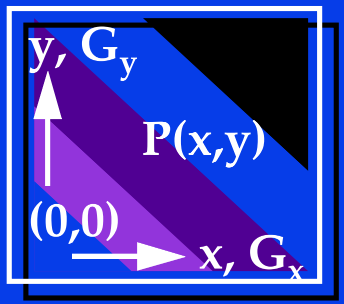
Outline

- Objective and Background
- Constraint Generation Problem
- Modeling Performance Dependence on Parasitics
- Modeling Performance Dependence on Technology
- Results
- Summary

Modeling Performance Dependence on Technology

- Technology parameters, e.g. t_{ox}, N_A, etc. , vary within the wafer along irregular patterns
- If chip is sufficiently small, gradients can be assumed constant in the x- and y- directions





- Technology parameter P(x,y) can be modeled as

$$P(x,y) = P(0,0) + G_x\, x + G_y\, y = P(0,0) + \Delta P$$

Modeling Performance Dependence on Technology

- Degradation of performance W due to all technology gradients acting on device D_j

$$\Delta W_{Dj} = \sum_i S_{j,Pi}\ \Delta P_i$$

where $S_{j,Pi}$: sensitivity of W wrt technology parameter P_i acting on D_j

- Example: μCox, N_A, N_D → MOS devices
  polysilicon thickness → resistances

Modeling Performance Dependence on Technology

- Performance degradation due to mismatch of devices D_j and D_k

$$\Delta W_{Dj,k} = \sum_i S_{j,k,Pi} \, \Delta P_i$$

where $S_{j,k,Pi}$: sensitivity of W wrt mismatch of technology parameter P_i acting on D_j and D_k

- Assuming statistical independence of technology parameters

$$\sigma^2(\Delta W_{Dj,k}) = \sum_i |S_{j,k,Pi}|^2 \sigma^2(\Delta P_i)$$

Topological Constraint Generation: Device Matching

- Assuming linear relation between technological parameters mismatch and distance d_{jk} and relative orientation of device pairs r_{jk} (Pelgrom 1989),

$$\sigma^2(\Delta P_{jk}) = A_{jk}\, d_{jk} + B_{jk}\, r_{jk}$$

Example of ΔP : $\Delta K'$, ΔVt for a pair of MOS transistors

- Then,

$$\sigma^2(\Delta W_{Dj,k}) = \sum_i |S_{j,k,Pi}|^2 \, (A_{jk}\, d_{jk} + B_{jk}\, r_{jk})$$

- Constraints on d_{jk} and r_{jk} are determined using our constrained optimization approach

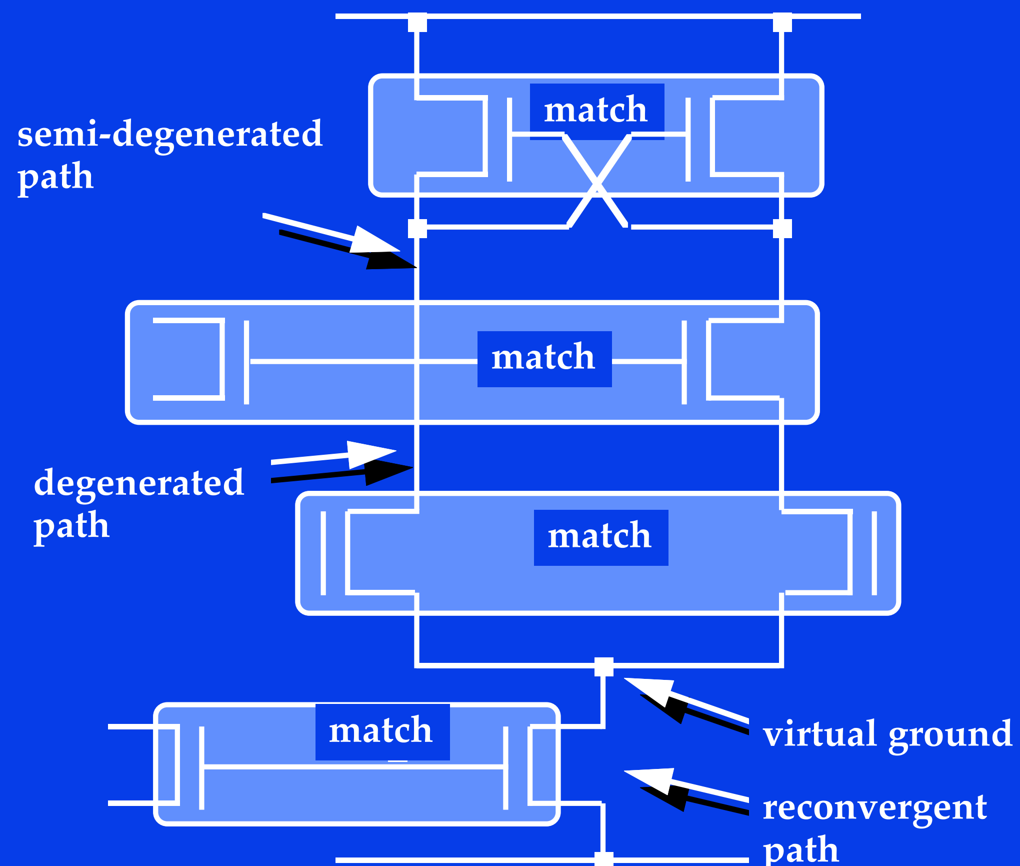
Topological Constraint Generation: Device and Interconnect Symmetry

- Matching constraints alone guarantee performance
- With device and interconnect symmetry, can achieve
  - higher flexibility of layout synthesis
  - better efficiency of layout tools
  - better robustness of the general approach

Topological Constraint Generation: Device and Interconnect Symmetry

- Candidates to symmetrization are
  - matched devices
  - matched interconnect, i.e. nets with matched parasitics
  - interconnect of fully differential signal paths
- These requirements have been used to drive a symmetry-constraint generating algorithm

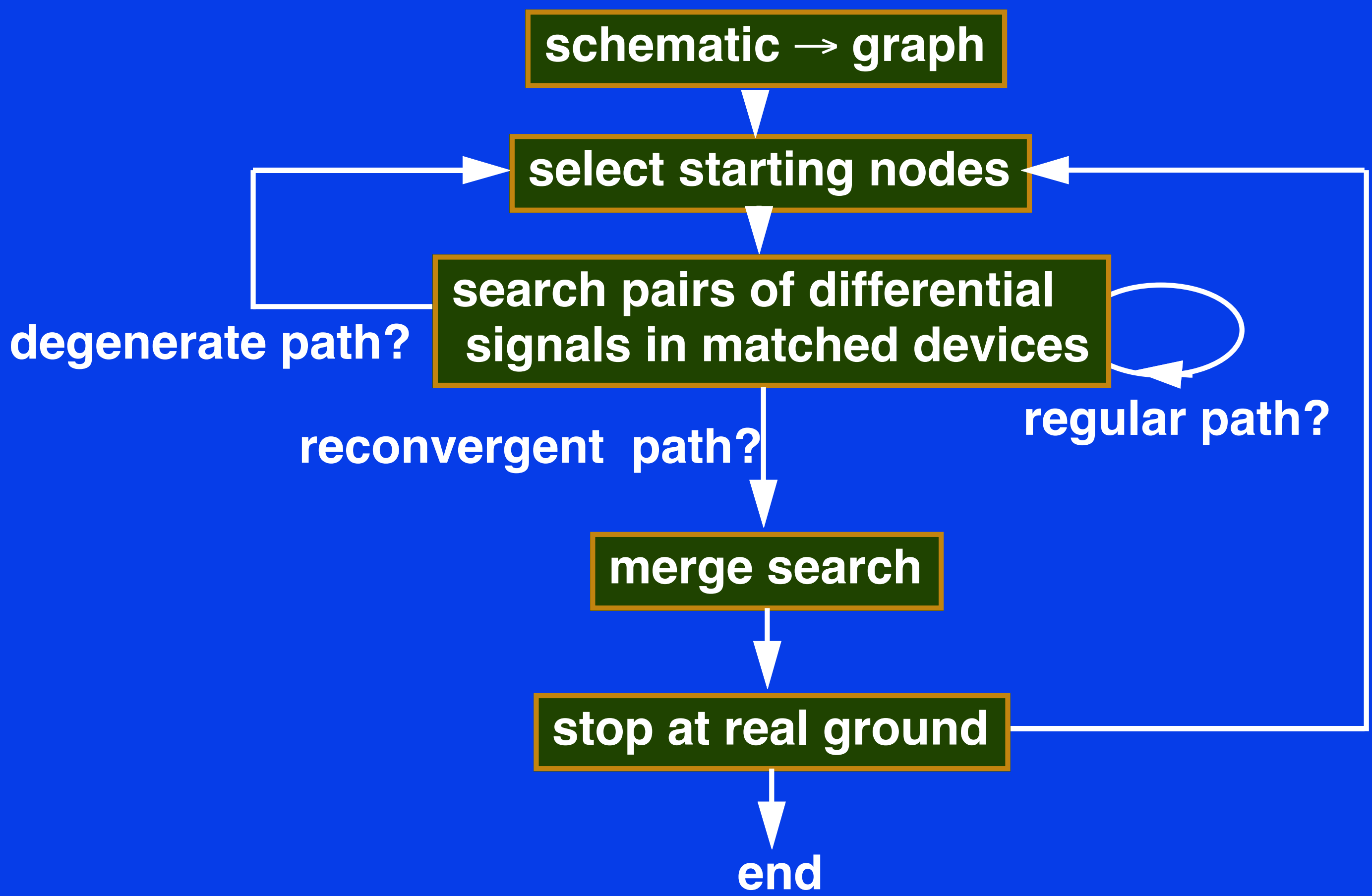
Symmetry Constraint Generating Algorithm



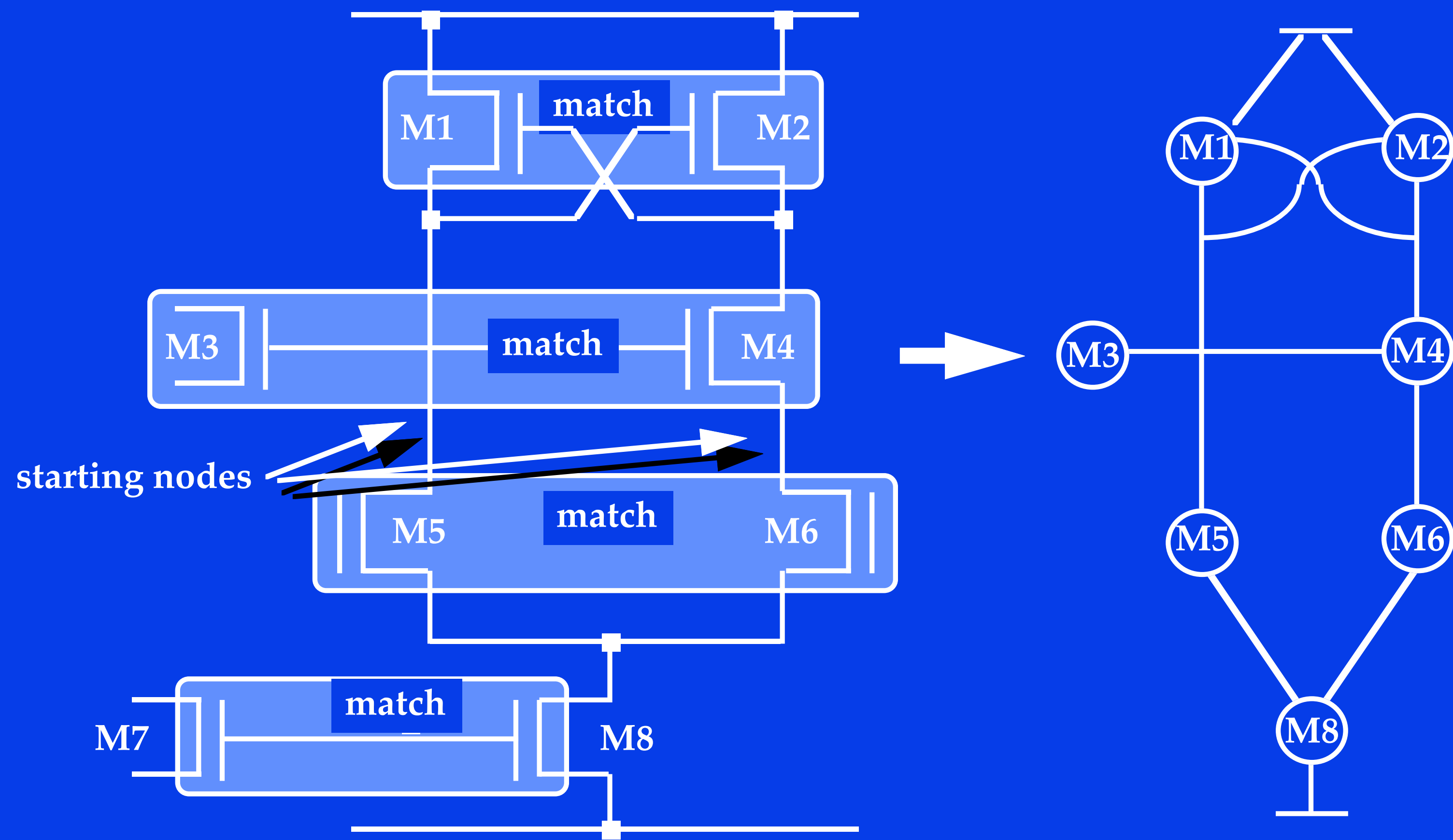
Definitions:

- **Virtual ground** =
  signal independent node
- **Reconvergent path** =
  path into which fully differential paths merge
- **Degenerated path** =
  devices on a search path are matched to devices on an incompatible path
- **Semi-degenerated path** =
  a node is at both sides of the symmetry axis

Symmetry Constraint Generating Algorithm



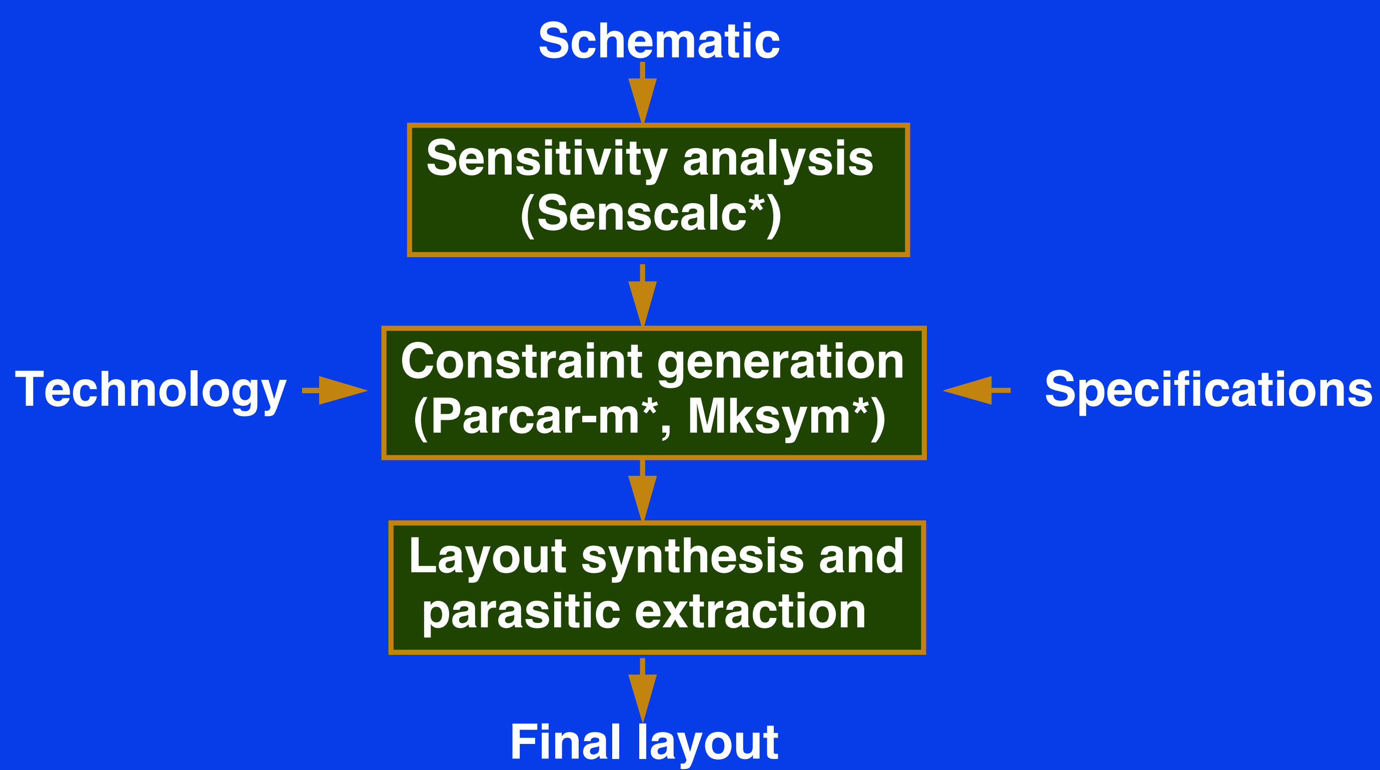
Symmetry Constraint Generating Algorithm: Example



Outline

- Objective and Background
- Constraint Generation Problem
- Modeling Performance Dependence on Parasitics
- Modeling Performance Dependence on Technology
- Results
- Summary

Results

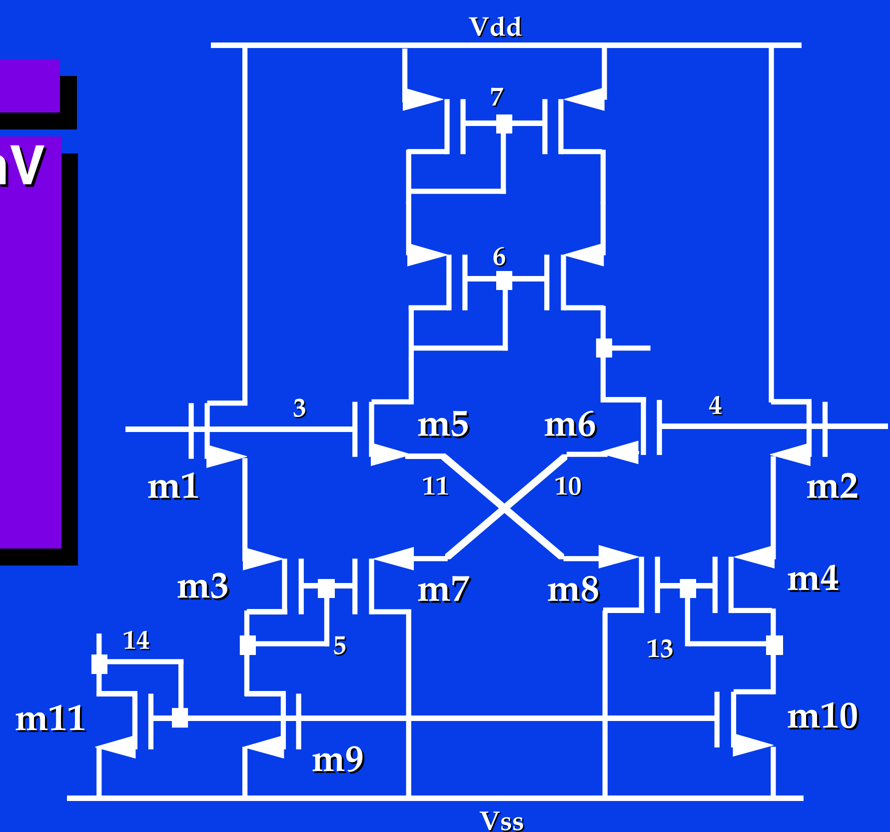


*) The program is written in C and runs under the UNIX operating system.

Class AB Amplifier

Perform.	Nominal	Specification
V_{off}	14mV	-8mV / +2mV
UGB	500MHz	-/+ 3MHz
PM	106 deg	-/+ 3deg
LFG	28 dB	- 3dB /•

Devices	Technology
15	scmos1.6μ



Generated Constraints

Maximum capacitive mismatch

Net pair	Net pair	Constraint
3 - 9	4 - 6	62.41 fF
11-gnd	10-gnd	80.97 fF

Maximum resistive mismatch

Branch	Branch	Constraint
Vdd-m12.s	Vdd-m14.s	0.1 Ω
Vss-m9.s	Vss-m10.s	0.1 Ω

Device matching constraints

Distance	Rotation	Devices
200 λ	0	m12-m14
200 λ	0	m9-m10

Symmetry Constraints

Axis	Symmetric devices
0	m1-m2; m3-m4; m7-m8; m9-m10
1	m12-m14; m13-m15; m5-m6

Tool	CPU (DECstation 5000/125)
Parcar-m / Senscalc	7.4 seconds
Mksym	0.1 seconds

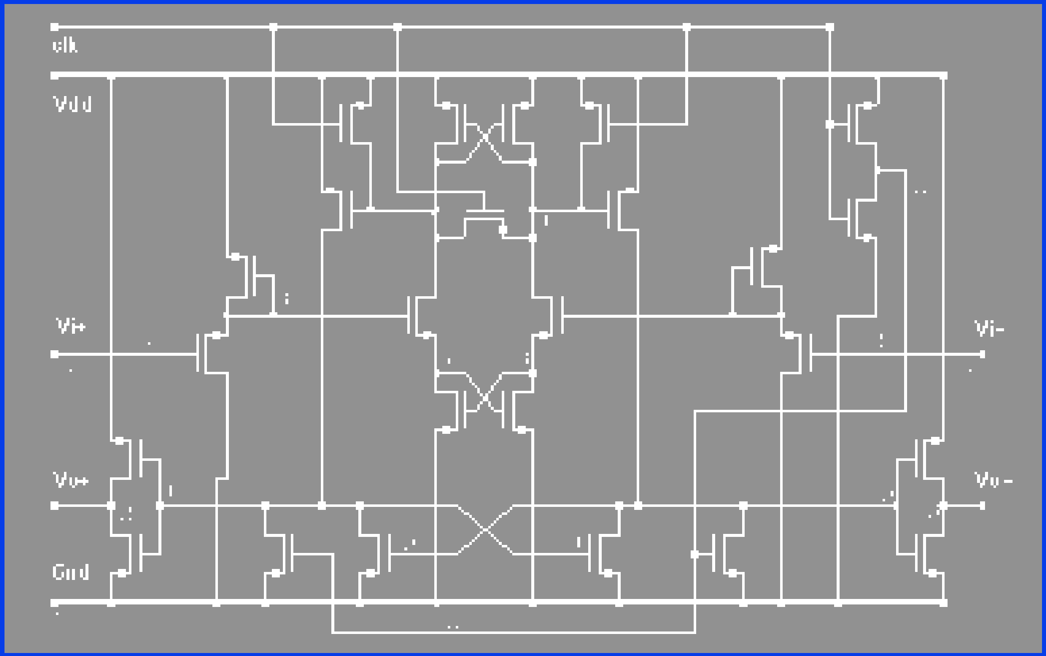
Clocked Comparator

Devices

25

Technology

scmos1.6μ



Layout

CPU time for constraint-driven layout synthesis

442 .9 seconds

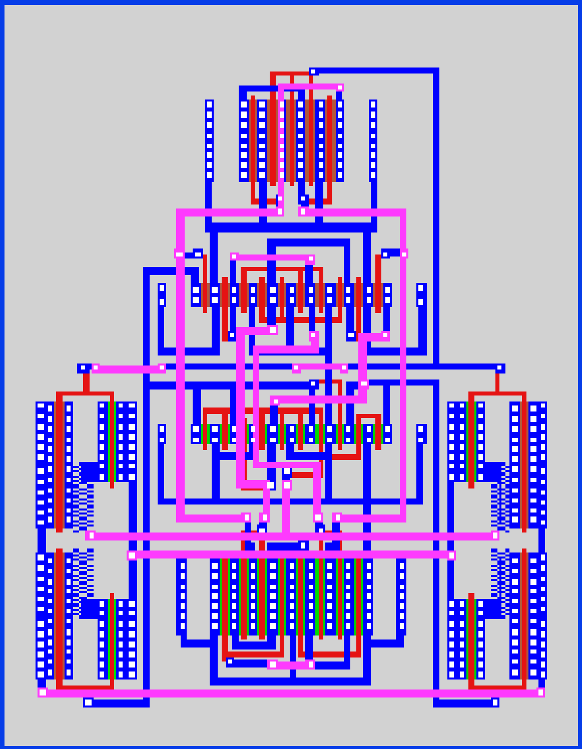
70 parasitic constraints

24 symmetries

6 device matching constraints

Technology

scmos1.6μ



Performance

Perform.	Nominal	Specification	Extracted data*
V_{off}	1.0mV	+/-0.5mV	1.4mV
t_{LH}	2 .49nsec	+/- 0.25nsec	2..69nsec
t_{HL}	2 .42nsec	+/- 0.25nsec	2..63nsec

*) Performed with Spice3 (level 3) on fully extracted layout.

Analog Benchmarks

- Constraints generated for a set of industrial circuits

Benchmark	# devices	# parasit.	# symmetry	# matching	CPU (DEC5000125)
fastcomp	25	70	24	6	10.9
philips	27	594	14	16	2151
ab	15	211	14	13	2.6
comp	26	24	14	8	0.1
opamp1	13	79	5	4	11.9
ota731	22	383	11	20	198.4

Conclusions

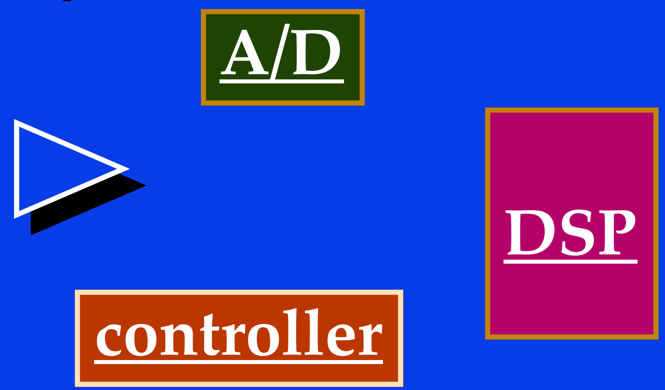
- A generalized methodology for generating
  - parasitic constraints
  - bounds on parasitic mismatch
  - device matching requirements
  - topological symmetry constraints
- Sensitivity analysis and constrained optimization used to find quantitative bounds on layout geometries and parasitics
- Effect of parasitics and parasitic mismatches on performance
- Influence of technology gradients on performance

Future Work

- Refine modeling of technology induced performance degradation
- Extend the methodology to distributed parasitics
- Apply the methodology to the extraction process

Background

- Complex mixed-signal systems are in simpler sub-components



- Top-down approach driven by specifications generally adopted for digital circuits
- Top-down, constraint-driven methodology for analog and mixed-signal circuits (CICC92)

Top-Down, Constraint-Driven Design Methodology

Higher-level constraints required
<u>by</u> previous design level

↓

Intermediate design levels

↓

Constraints imposed <u>to</u> the next
design level

At each design level

- **Mapping higher-level onto lower-level design constraints**
- **Design optimizations and architecture selection**

Top-Down, Constraint-Driven Design Methodology

Performance specifications <u>from</u> previous design level

↓

Constraint generation

↓

Layout synthesis