Introduction to Automatic Design Optimization for Analog Circuits

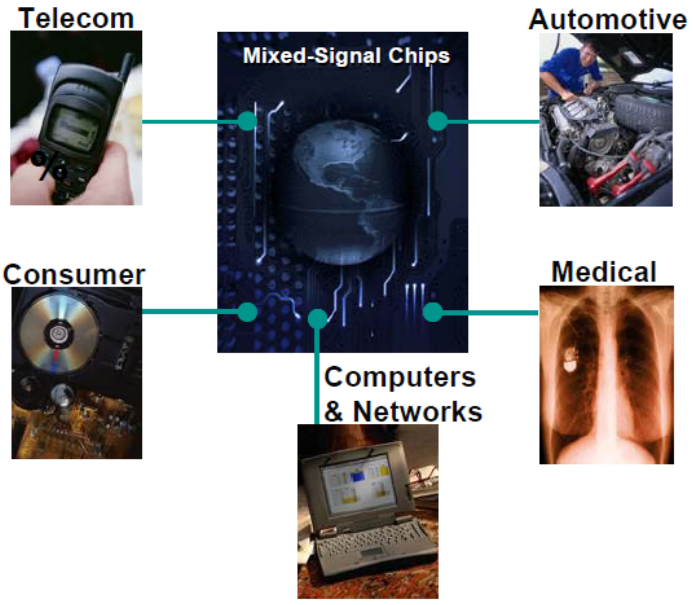
Prof. Chien-Nan Liu
Institute of Electronics
National Chiao-Tung Univ.

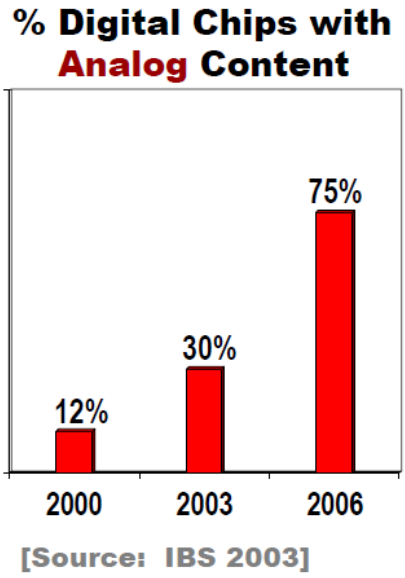
Tel: (03)5712121 ext:31211
E-mail: jimmyliu@nctu.edu.tw
http://mseda.ee.nctu.edu.tw/jimmyliu

Outline

- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

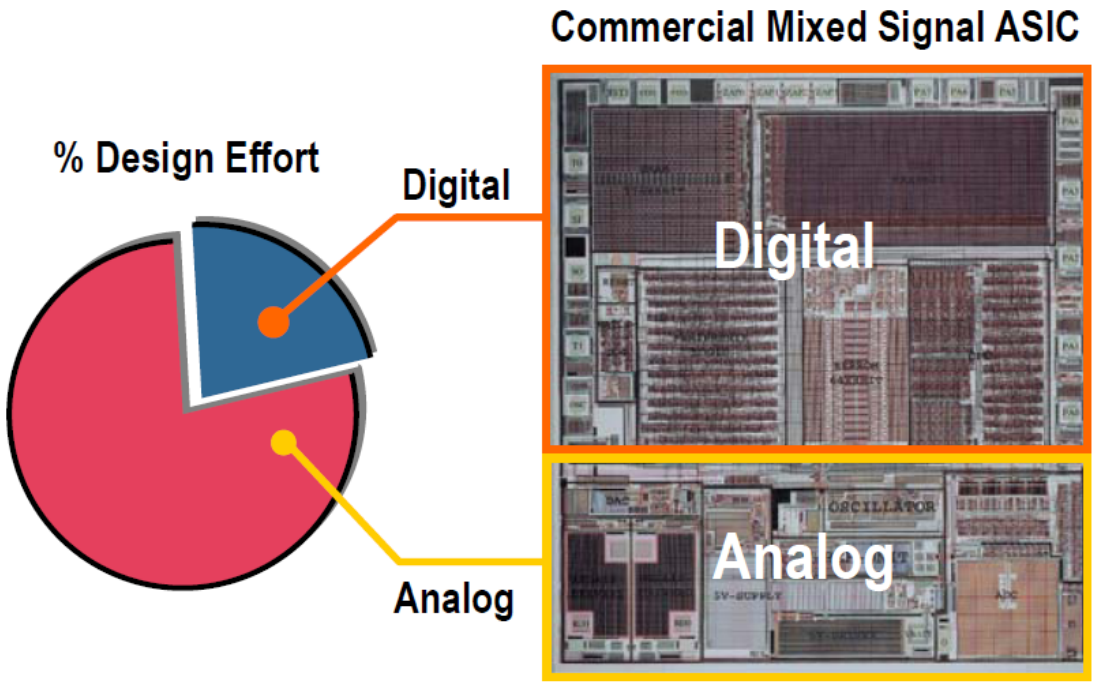
Many "Mixed-Signal" in the World





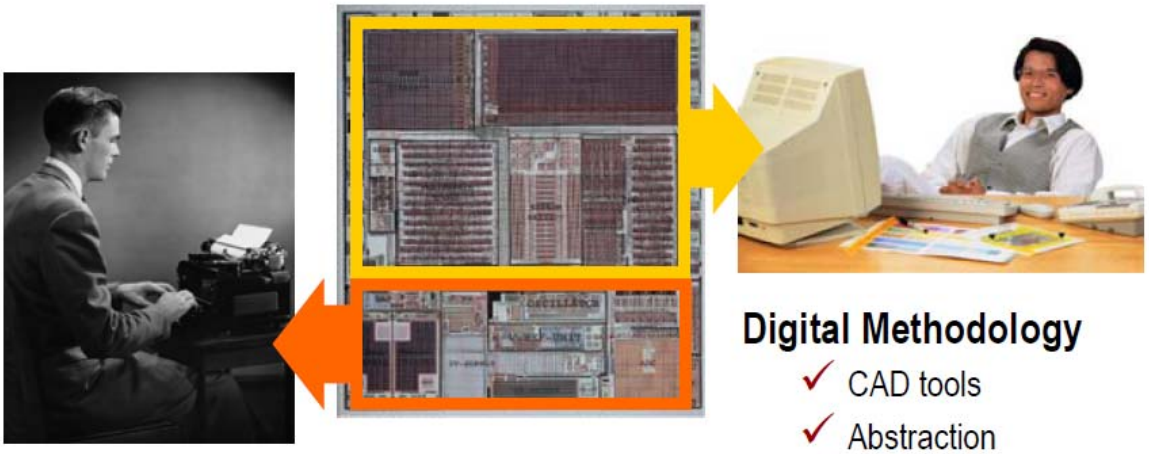
Courtesy : R. A. Rutenbar, CMU

The Mixed-Signal Design Problem



Courtesy : R. A. Rutenbar, CMU

Why This Happens ??



Analog Methodology

- ☒ CAD tools
- ☒ Abstraction
- ☒ Reuse & IP

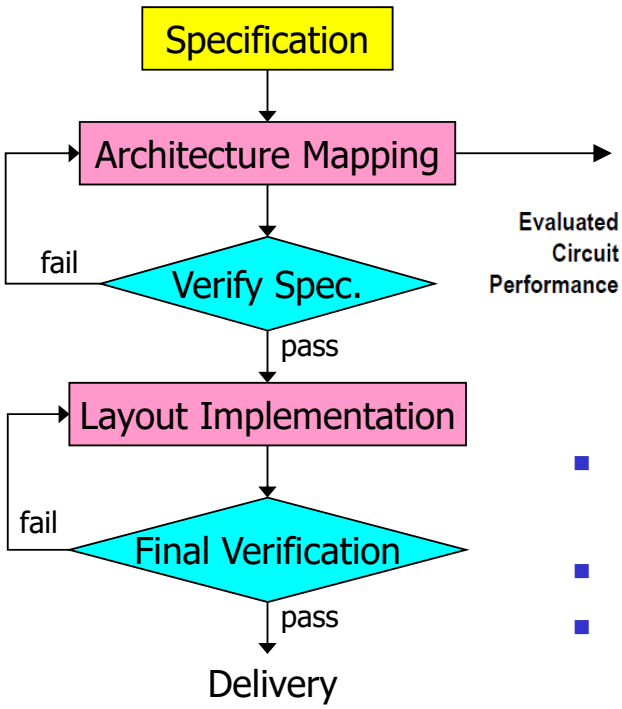
Digital Methodology

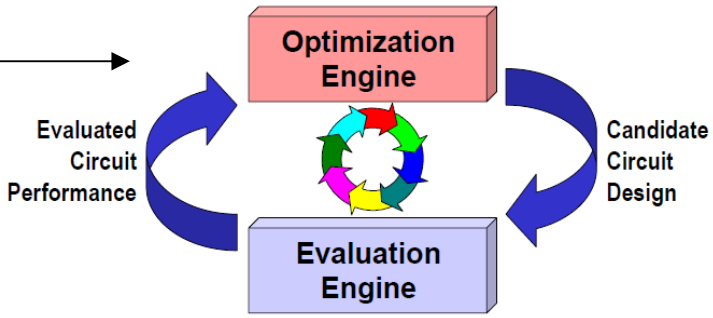
- ✓ CAD tools
- ✓ Abstraction
- ✓ Reuse & IP

Courtesy : R. A. Rutenbar, CMU

Typical AMS Design Flow

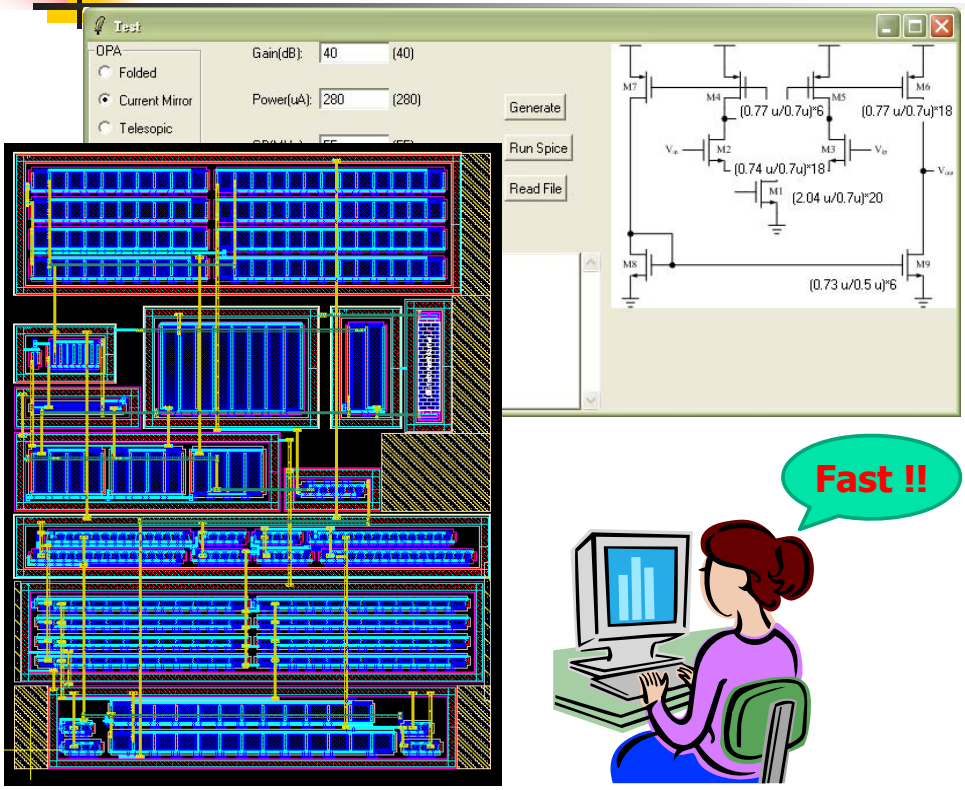
- Manual designing analog circuits is not easy !!

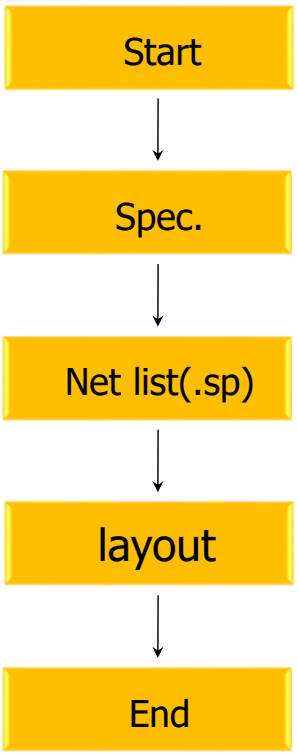




- Often require many iterations
  - Hard to meet all spec.
- Simulation time is long …
- Optimality is not guaranteed

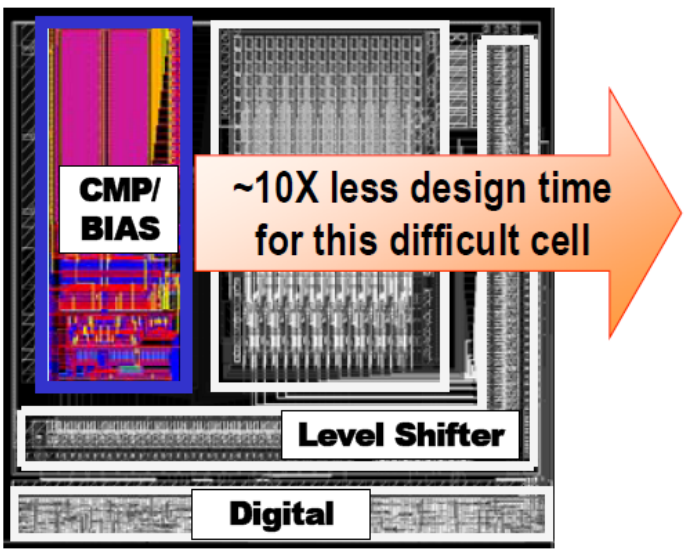
Synthesis Tool for Analog Circuits



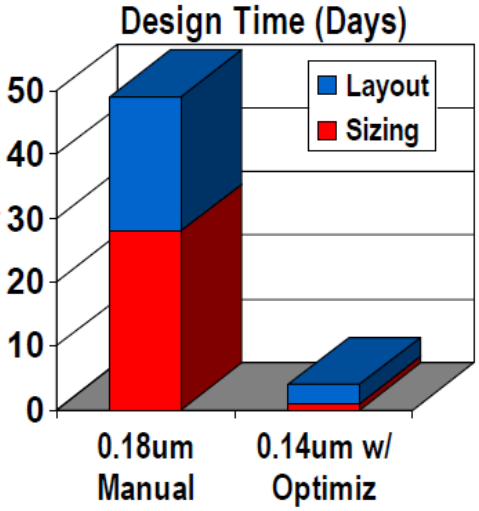


Industrial Results

- Toshiba result:
  - Porting a data converter from 0.18um to 0.14um



~10X less design time for this difficult cell



Courtesy : R. A. Rutenbar, CMU

Analog Synthesis is the Trend …

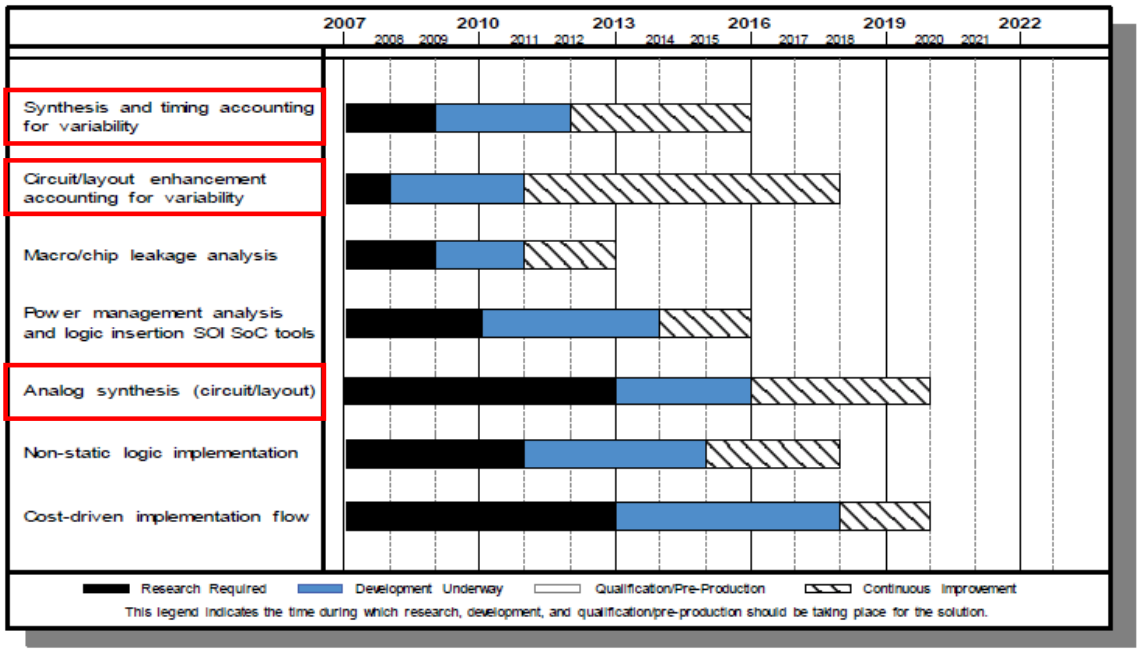
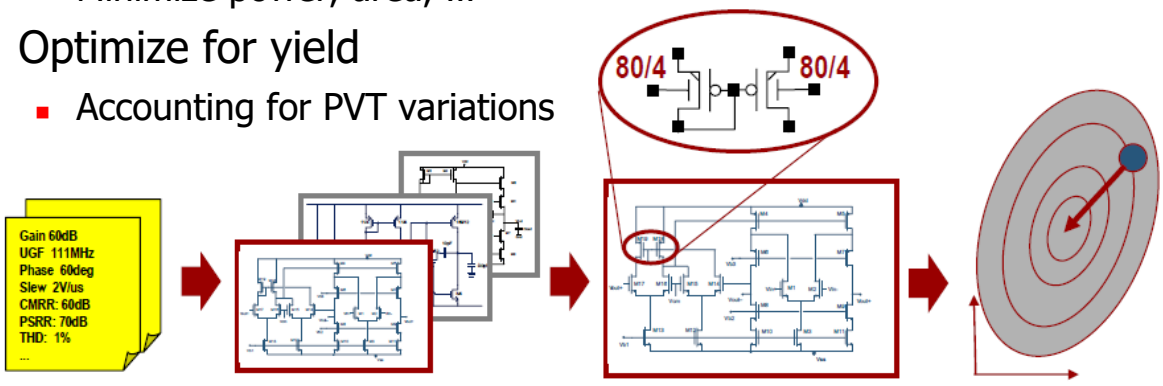


Figure DESN5 Logical/Circuit/Physical Design Potential Solutions

Source : ITRS Roadmap 2007 Edition, SIA

Synthesis = Automation + Optimization

- Optimize for performance
  - Minimize power, area, …
- Optimize for yield
  - Accounting for PVT variations



Generate proper specs

Design proper circuit topology

Design proper device sizing/biasing

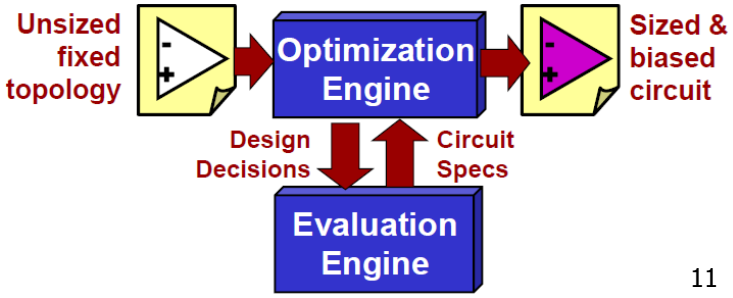
Optimize for centering, yield

Focus for automation / optimization

Courtesy : R. A. Rutenbar, CMU

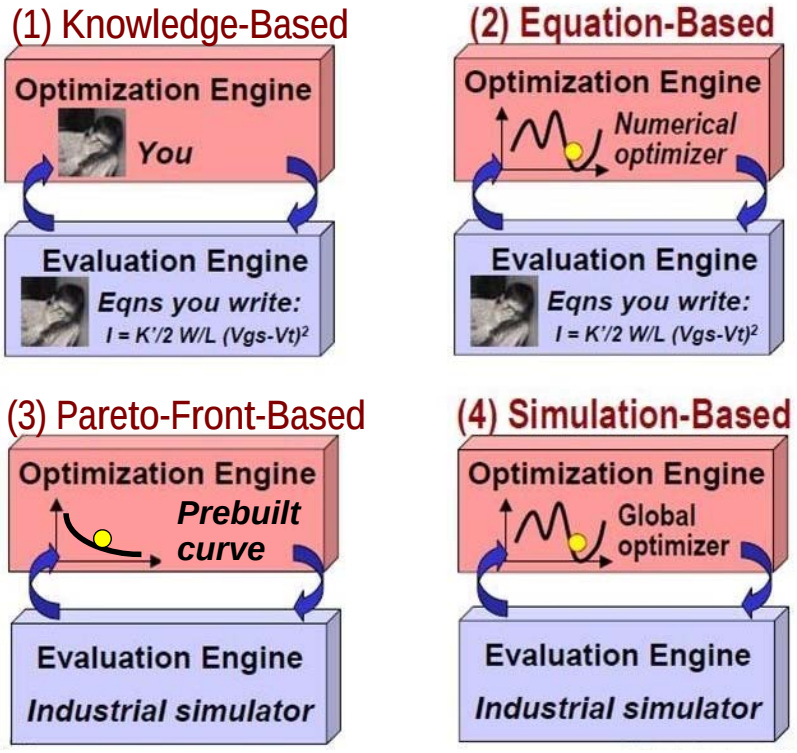
Existing Approaches

- Automatic circuit sizing are often classified into 4 types:
  - Knowledge-based optimization
    - Require codifying extensive circuit knowledge
  - Simulation-based optimization
    - Require costly circuit simulation
  - Pareto-front-based optimization
    - Require a lot of simulations to build the tradeoff curves
  - Analytical equations-based optimization
    - Rough equations limit the accuracy of final solutions



Courtesy : R. A. Rutenbar, CMU

Comparisons of Diff. Approaches



Courtesy : R. A. Rutenbar, CMU

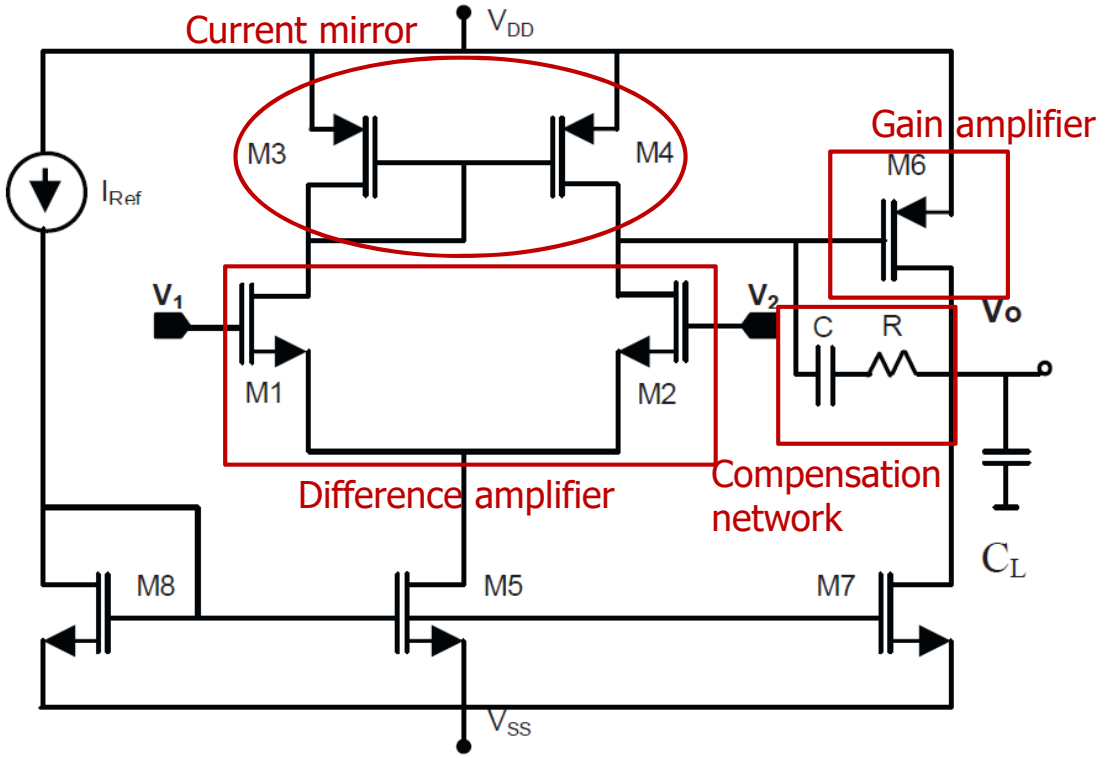
Outline

- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

Knowledge-Based Optimization

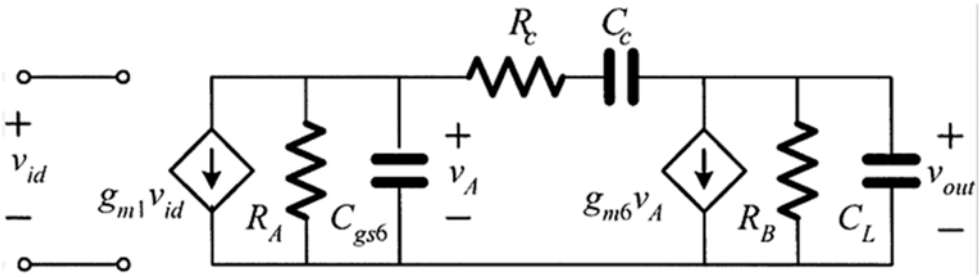
- Develop a step-by-step design flow based on designers' experience
  - For specific circuits only
- Typically fast to provide a rough design solution
  - May still require manual adjustment on the final circuit to meet design specifications
- Optimality cannot be guaranteed
  - Quality depends on cases (often over-design)
  - Workable, but may not the best

Example: Two-Stage OPA

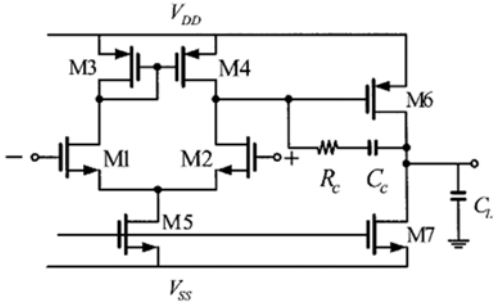


Basic Op-Amp Equations

- Small-signal equivalent circuit



- $R_A = r_{ds2} // r_{ds4}$ $\quad R_B' = r_{ds6} // r_{ds7}$
- $A_0 = g_{m1} g_{m6} R_A R_B$
- $\omega_{p1} \cong \dfrac{1}{g_{m6} R_A R_B C_C}$ $\quad \omega_u' \cong A_0 \omega_{p1} = \dfrac{g_{m1}}{C_c}$



Source : J. Mahattanakul, J. Chutichatuporn, "Design Procedure for Two-Stage CMOS Opamp With Flexible Noise-Power Balancing Scheme," IEEE Trans. Circuits and Systems, vol. 52, no 8, 2005, pp. 1508–1514.

Design Procedure (1/3)

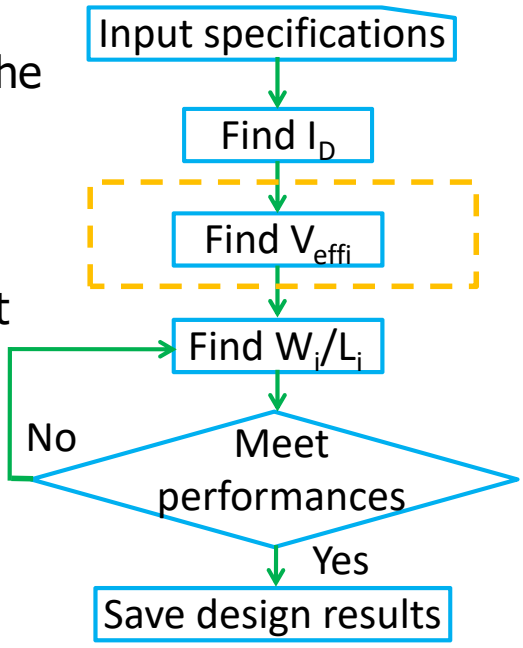
- Decide the bias point first
- Determine the device sizes from the given specifications
- Optimality cannot be guaranteed
  - Depend on the initial design
- Manual adjustment loops still exist

$$I_5 = SR(C_C + C_L)$$

$$V_{eff\,3} = V_{DD} - V_{CM\,(Max)} + V_{TN}$$

$$V_{eff\,5} = V_{CM\,(Min)} - V_{SS} - V_{TN} - V_{eff\,1,2}$$

$$V_{eff\,1,2} = SR/\omega_u$$



Source : J. Mahattanakul, J. Chutichatuporn, "Design Procedure for Two-Stage CMOS Opamp With Flexible Noise-Power Balancing Scheme," IEEE Trans. Circuits and Systems, vol. 52, no 8, 2005, pp. 1508–1514.

Design Procedure (2/3)

- Develop those steps based on designers' knowledge

Step 1

$$C_C = \frac{16kT}{3\omega_u S_n(f)}\left[1 + \frac{SR}{\omega_u (V_{HR}^{CM+} + V_{tn})}\right]$$

Step 2

$$I_{D7} = SR(C_c + C_L)$$

Step 3

$$L_6 = \sqrt{\frac{3\mu_p V_{HR}^{out+} C_c}{2\omega_u (C_c + C_L)\tan(\phi_M)}}$$

Step 4

$$W_6 = \frac{2SR(C_c + C_L)}{\mu_p C_{ox}\left(V_{HR}^{out+}\right)^2} L_6$$

Step 5

$$I_{D5} = C_c SR$$

Source : J. Mahattanakul, J. Chutichatuporn, "Design Procedure for Two-Stage CMOS Opamp With Flexible Noise-Power Balancing Scheme," IEEE Trans. Circuits and Systems, vol. 52, no 8, 2005, pp. 1508–1514.

Design Procedure (3/3)

Step 6

$$(W/L)_{1,2} = \frac{\omega_u^2 C_c}{\mu_n C_{ox} SR}$$

Step 7

$$(W/L)_{5,8} = \frac{2SRC_c}{\mu_n C_{ox}\left(V_{HR}^{CM-} - V_{tn} - SR/\omega_u\right)^2}$$

Step 8

$$(W/L)_7 = \left(\frac{C_c + C_L}{C_c}\right)(W/L)_{5,8}$$

Step 9

$$(W/L)_{3,4} = \frac{(W/L)_6}{2(W/L)_7}(W/L)_{5,8}$$

Step 10

$$(W/L)_9 = \frac{2C_c SR}{\mu_p C_{ox} V_{HR}^{out+}\left(V_{DD} - V_{HR}^{out+} - 2|V_{tp}|\right)}$$

Source : J. Mahattanakul, J. Chutichatuporn, "Design Procedure for Two-Stage CMOS Opamp With Flexible Noise-Power Balancing Scheme," IEEE Trans. Circuits and Systems, vol. 52, no 8, 2005, pp. 1508–1514.

Experimental Results

- The results can meet all specifications
  - Over-design or not ??

Specification		Result
Gain (dB)	≥80	85.1
Gain-Bandwidth (MHz)	≥5	6
Slew Rate (V/μs)	≥5	5.2
Phase Margin (degree)	>65	65
Power (μW)	min	207

GB=Gain-Bandwidth, SR=Slew Rate, PM=Phase Margin

Source : J. Mahattanakul, J. Chutichatuporn, "Design Procedure for Two-Stage CMOS Opamp With Flexible Noise-Power Balancing Scheme," IEEE Trans. Circuits and Systems, vol. 52, no 8, 2005, pp. 1508–1514.

Outline

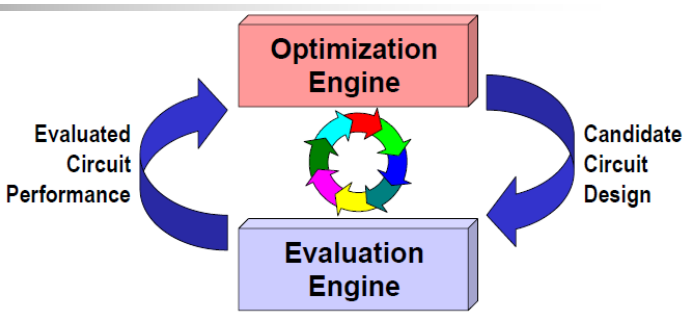
- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

Simulation-Based Optimization

- Given an initial sizing, check the circuit performance by simulations and adjust the device sizes accordingly
  - simulation → sizing → simulation → sizing …
- Non-deterministic optimization approaches are often adopted to search for an optimal circuit
  - Simulated annealing (SA), genetic algorithm (GA), …
- Accurate but time-consuming
  - The optimization results are the same with the simulation results
  - Hundreds of simulations may be required → long simulation time

Improve Simulation-Based Approach

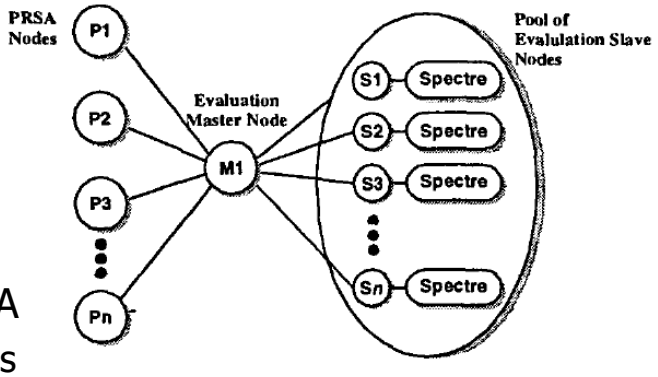
- Typical solutions to improve the efficiency of simulation-based approaches:
- *Less search:*
  - Fewer samples require less simulation time
  - Wider search often yields better solutions
- *Parallel circuit evaluation:*
  - Use faster simulator, but often has some limitations
- *Parallel circuit search:*
  - Multiple points are searched in parallel and synchronized in some manner --> require special optimization algorithm

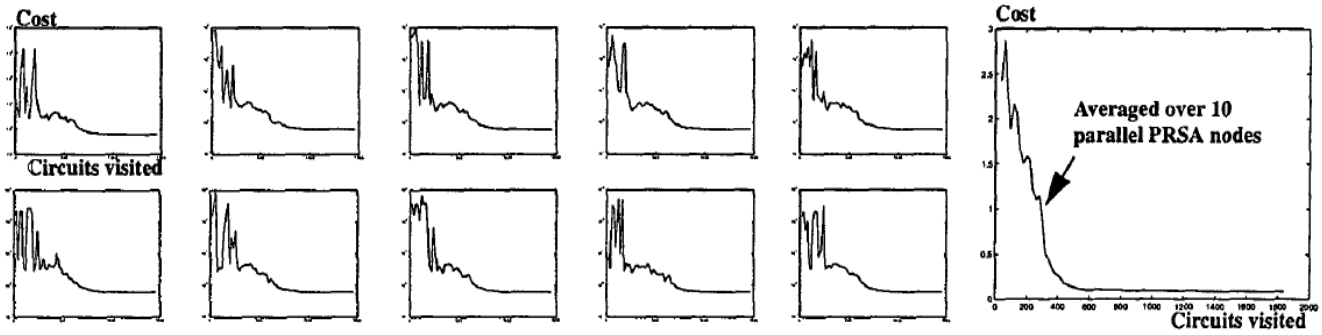


Source : M. Krasnicki, R. Phelps, R. A. Rutenbar, L. R. Carley, "MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Cells," DAC'99, pp. 945-950, 1999.

Parallel Recombinative SA (PRSA)

- Distribute works in parallel
- Each CPU creates a standard annealing optimization
  - Schedule is truncated to 1/N
- Regard each annealer as one element of the population in GA
  - Allow them to exchange results





Source : M. Krasnicki, R. Phelps, R. A. Rutenbar, L. R. Carley, "MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Cells," DAC'99, pp. 945-950, 1999.

Experimental Results

- Simulation time is reduced by parallel search
  - From hours to minutes
  - Still a long computation time for such small circuits

Custom OPA Design

Attribute	Manual Design	Auto-Synthesis: Spec.	Result
CLoad (pF)	1.25	1.25	
Vdd (V)	5	5	
DC Gain (dB)	71.2	≥ 71:	110
UGF (MHz)	47.8	≥ 48:	70
Phase Margin (deg)	77.4	≥ 77:	84
PSRR - Vss (dB)	92.6	≥ 93:	131
PSRR - Vdd (dB)	72.3	≥ 72:	108
Output Swing (V)	± 1.4	± 1.4:	± 1.45
Settling Time (ns)	-	↓:	29
Active Area ($10^3\mu^2$)	68.7	↓:	23
Circuits Evaluated			70,000
CPU Time (minutes)			219

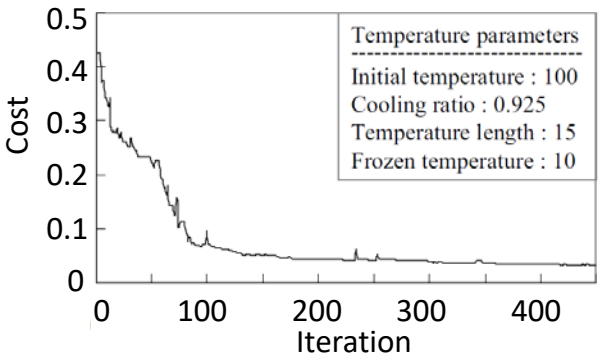
Folded Cascode OPA

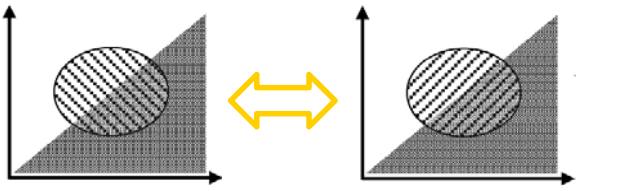
Attribute	Auto-Synthesis: Spec.	Result
CLoad (pF)	1	
Vdd (V)	5	
DC Gain (dB)	≥ 70:	71.4
UGF (MHz)	≥ 10:	24.3
Phase Margin (deg)	≥ 60:	69
PSRR - Vss (dB)	≥ 40:	111
PSRR - Vdd (dB)	≥ 40:	132
Output Swing (V)	± 1.35:	± 1.37
Settling Time (ns)	≤ 100 :	50
Active Area ($10^3\mu^2$)	≤ 68 :	11
Circuits Evaluated		60,000
CPU (minutes)		152

Source : M. Krasnicki, R. Phelps, R. A. Rutenbar, L. R. Carley, "MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Cells," DAC'99, pp. 945-950, 1999.

Search Space Reduction

- Simulated annealing (SA) is used in this work to search the optimal solution
- Determine the bias point first in this work
  - Make sure the transistors work in proper modes
  - Reduce unnecessary search space
- Transistor sizes can be calculated from bias point
  - One-to-one mapping





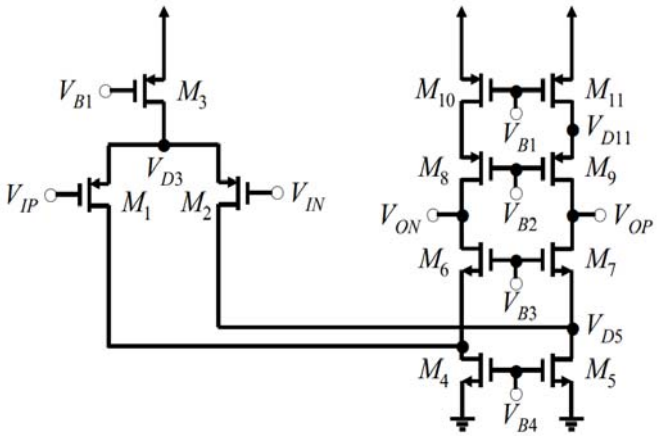
Bias-Point Space Transistor-Size Space

Source : C.-W. Lin, P.-D. Sue, Y.-T. Shyu, S.-J Chang, "A Bias-Driven Approach for Automated Design of Operational Amplifiers," VLSI-DAT, 2009, pp. 118-121.

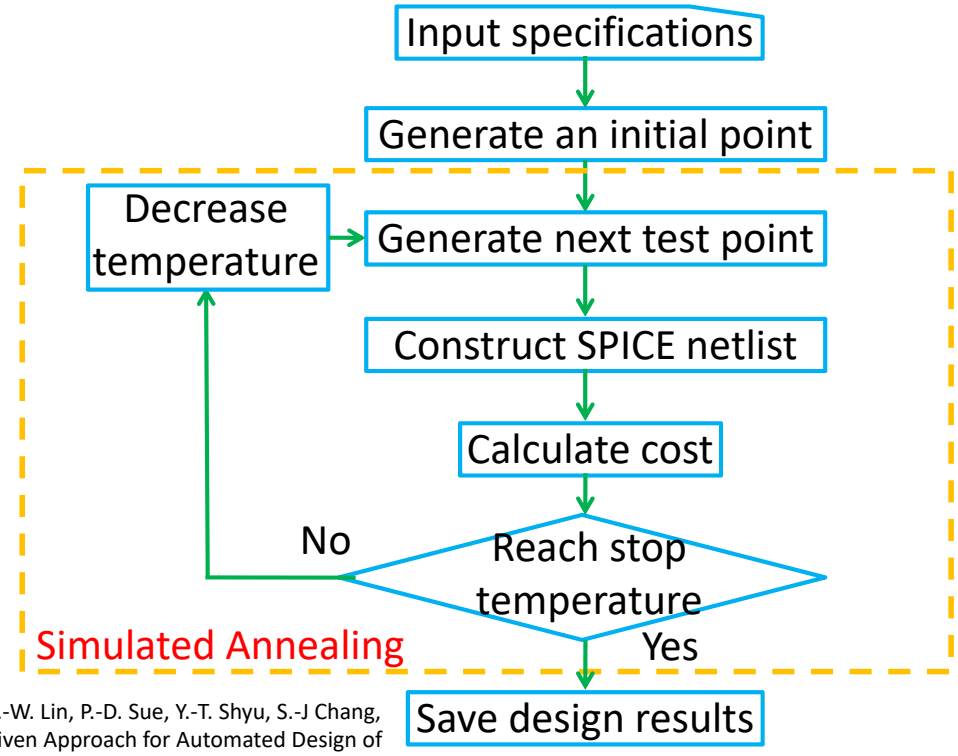
Operation Region Constraints

- All transistors are operated in saturation region
  - $V_{GS} - V_T > 0$
- The variables can be limited by the following equations

$$
\begin{aligned}
V_{B1} &\le V_{DD} - |V_{TP}| - \Delta V \\
V_{B1} &\ge V_{D3} - |V_{TP}| + \Delta V \\
V_{B1} &\ge V_{D11} - |V_{TP}| + \Delta V \\
V_{D3} &\le V_{B1} + |V_{TP}| - \Delta V \\
V_{D3} &\ge V_{ICM} + |V_{TP}| + \Delta V \\
&\vdots
\end{aligned}
$$



SA-Based Optimization Flow



Source : C.-W. Lin, P.-D. Sue, Y.-T. Shyu, S.-J Chang, "A Bias-Driven Approach for Automated Design of Operational Amplifiers," VLSI-DAT 2009.

Experimental Results

- Target circuit: two-stage OPA (11 transistors)
- The final results meet all specifications
  - No modeling error in this approach
- Design time is relatively longer than other approaches
  - Contains 450 sizing + layout iterations
  - Each iteration requires a circuit simulation

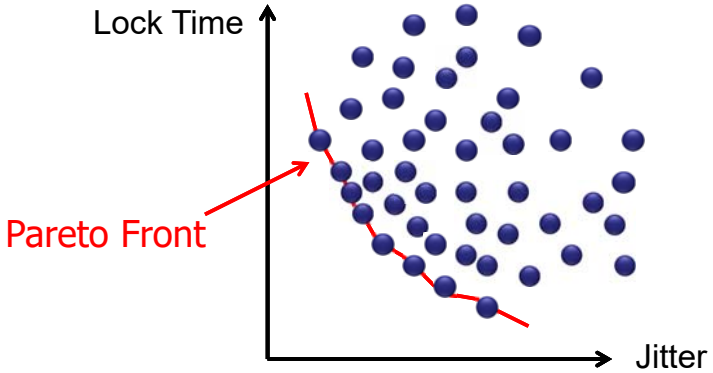
Performance	Spec.	0.13 um	0.18 um	0.25 um
DC Gain (dB)	> 60	61.2	64.6	62.6
UGF (MHz)	> 450	461.4	463.7	457.6
PM (degree)	> 65	65.8	67.6	65.0
SR (V/us)	> 500	500.4	511.8	698.8
Power (mW)	minimum	1.00	1.18	2.79
Time (**min.**)		44	57	26

Outline

- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

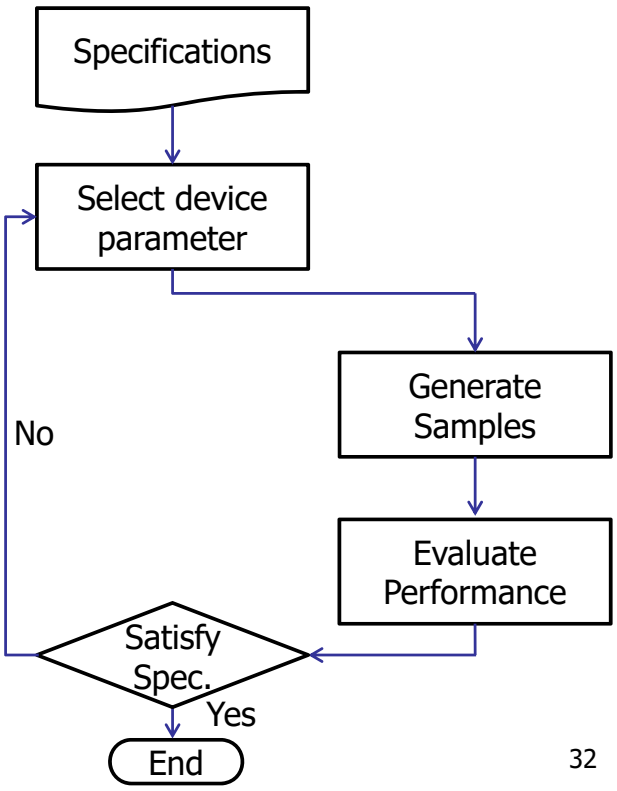
What is Pareto Front ?

- With a lot of design samples and their corresponding performances, the "best" points under different requirements form a *trade-off curve* → **Pareto Front**
- Need a lot of samples to find out the Pareto front
  - Similar to building a database in advance
  - Can be reused for the same circuit with the same technology



Pareto-Front-Based Ckt. Sizing

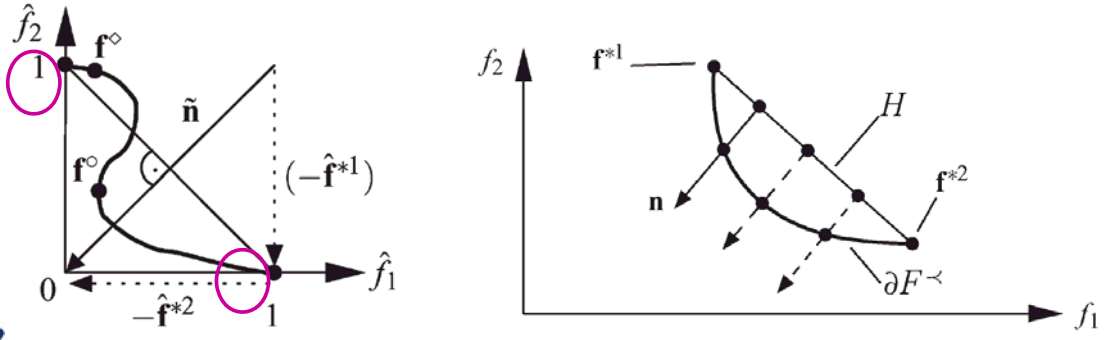
- Given the required performance, choose the corresponding design point at the Pareto Front
  - The corresponding device sizes can be obtained from the database
- The sizing step is fast, but building the database is time-consuming
  - Iteratively generate samples
  - Spend more than 4 hours in this case



Source : S. K. Tiwary, "Pareto Optimal Modeling for Efficient PLL Optimization", MSMC '07

Pareto Front Generation

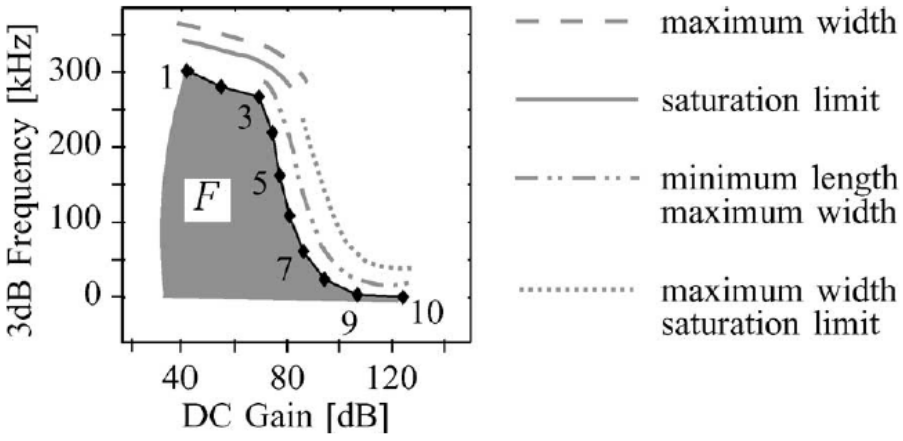
- Analog design often requires a multi-objective cost func.
  - $F(\vec{P}) = \sum_{i=1}^{M} \omega_i \cdot P_i(\vec{D})/Spec_i$ ω_i = weight coefficient of each performance
    $P_i(D) / Spec_i$ = *i-th* performance and its spec.
- Normal-Boundary Intersection (NBI) is a multi-objective searching method
- f^{*i} is the min. value that this performance can achieve
  - The middle point of minimum points is the next sample



Source : G. Stehr et al. , "Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier–Motzkin Elimination", IEEE Trans. on CAD, vol. 26, no. 10, Oct. 2007

Pareto Front with Diff. Constraints

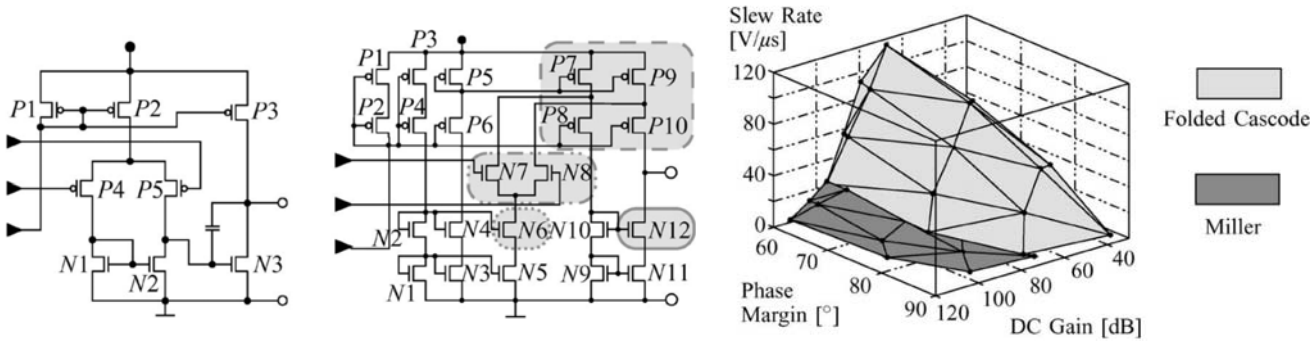
- This example shows that Pareto fronts do not have to be convex in the presented method
  - Still work under different constraints



Source : G. Stehr et al. , "Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier–Motzkin Elimination", IEEE Trans. on CAD, vol. 26, no. 10, Oct. 2007

Exp. Results of Pareto Generation

- Implement in C++
- Two OPA circuits are used as examples
  - Run time is more than a half hour



Amplifier	# Transistor	# Pareto points	# Simulation	Time (min : sec)
Miller	8	16	3264	31:18
Folded Cascade	22	16	5434	49:30

Source : G. Stehr et al. , "Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier–Motzkin Elimination", IEEE Trans. on CAD, vol. 26, no. 10, Oct. 2007

Outline

- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

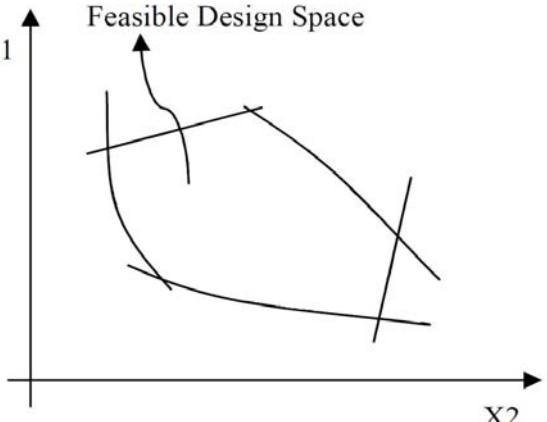
Equation-Based Optimization

- Circuit performances are often represented as non-linear equations
  - Ex: gain = $[g_{m2}/(g_{ds2}+g_{ds0})]*[g_{m6}/(g_{ds6}+g_{ds7})]$
- Design specifications can be viewed as the constraints for those non-linear equations
  - Solving those non-linear constraints can obtain the feasible solutions that meet all specifications
- *Non-linear programming* (NLP) solvers can find a feasible solution with minimum (or maximum) cost
- Simplified equations and parameter models often limit the accuracy of equation-based approaches
  - The predicted circuit performance is different to real simulation results

GP-Based Optimization (1/2)

- *Geometric programming* (GP) is a special case of non-linear programming with special forms
- In GP problems, any of its local optimal point is also a global optimal point
  - Solving GP problems is efficient in most cases

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \le 1 \quad i = 1,\ldots,m \\ & g_j(x) = 1 \quad j = 1,\ldots,p \\ & x_k \ge 0 \quad k = 1,\ldots,n \end{aligned}$$



- *g(x)* is in monomial form:

→ $g(x) = Cx_1^{a_1}x_2^{a_2}x_3^{a_3}.....x_n^{a_n}, C > 0, a_i \in R$

- *f(x)* is in posynomial form:

→ a summation of one or more monomial terms.

x1 and x2 are design variables

GP-Based Optimization (2/2)

- Circuit equations should be transformed into GP forms, for example: (folded cascode OPA)

$$Gain_{spec} \times \left(\frac{g_{d\,1} g_{d\,6}}{g_{m\,1} g_{m\,6}} + \frac{g_{d\,4} g_{d\,6}}{g_{m\,1} g_{m\,6}} + \frac{g_{d\,8} g_{d\,10}}{g_{m\,1} g_{m\,8}} \right) \leq 1$$

- Transistor models should be converted into GP-compatible form also

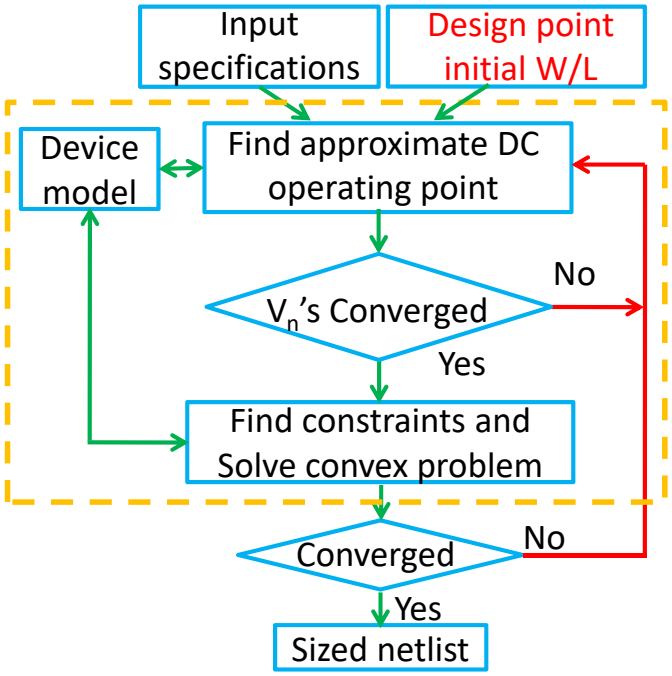
$$g_d = k_1 I_D^{a_1} L^{b_1} C^{c_1} \qquad g_m = k_2 I_D^{a_2} L^{b_2} W^{c_2}$$

- Curve fitting is often required to build those equations
  - Typical equations are often not in standard GP forms
  - Regression errors may lead to significant prediction error on circuit performance
  - In previous paper, g_d may have up to 65% avg. error

Source : P. Mandal, V. Visvanathan, "CMOS Op-Amp Sizing Using a Geometric Programming Formulation," IEEE Trans. on CAD, vol. 20, no. 1, 2001, pp. 22-38.

Iterative GP Approach (1/2)

- One method to solve the accuracy issue is using iterative approaching
- Solve convex programming to obtain the device sizes based on an initial point
  - Determine the approximate DC operating point first
  - May have prediction error due to inaccurate model
- Revise the GP models according to the errors and solve the equations again
  - Converged after several iterations



Source : P. Mandal, V. Visvanathan, "CMOS Op-Amp Sizing Using a Geometric Programming Formulation," IEEE Trans. on CAD, vol. 20, no. 1, 2001, pp. 22-38.

Iterative GP Approach (2/2)

- May not meet the spec. still after converged
  - Especially for the short channel transistors
  - Prediction accuracy is limited by modeling errors
- Manual adjustments are still required for the final circuits
  - Optimality cannot be reserved
  - Not convenient

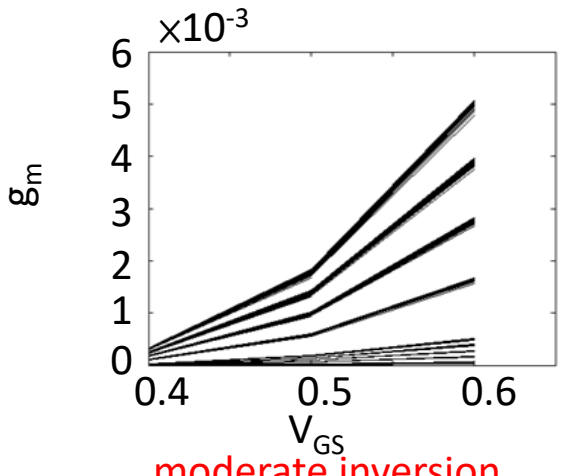
Specification		Result
Gain(dB)	≥60	58
GB(MHz)	≥10	9
Swing(V)	≥3	4.8
SR(V/us)	≥20	17.3
PM(degree)	≥60	62.3
Power(μW)	min	280
CPU time(s)		49.9

Source : P. Mandal, V. Visvanathan, "CMOS Op-Amp Sizing Using a Geometric Programming Formulation," IEEE Trans. on CAD, vol. 20, no. 1, 2001, pp. 22-38.

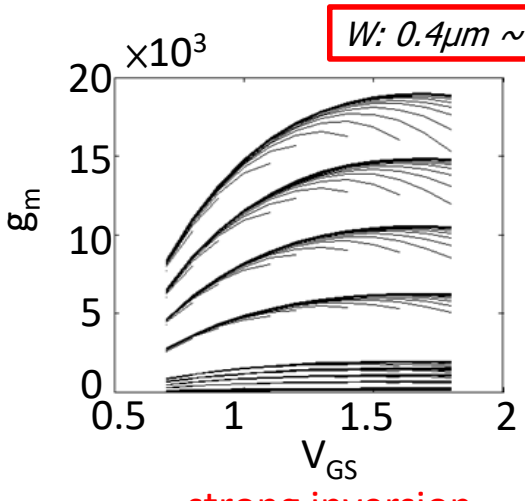
Nonlinear Parameters

- Some transistor parameters (ex: g_m, g_{ds}, …) exhibit very different behaviors in different operation regions
  - Accurate curve fitting is not easy

W: 0.4μm ~ 100μm



moderate inversion



strong inversion

Source : G. Wei and R. Hornsey, " A power optimization method for CMOS Op-Amps using sub-space based geometric programming," DATE, 2010, pp. 508-513.

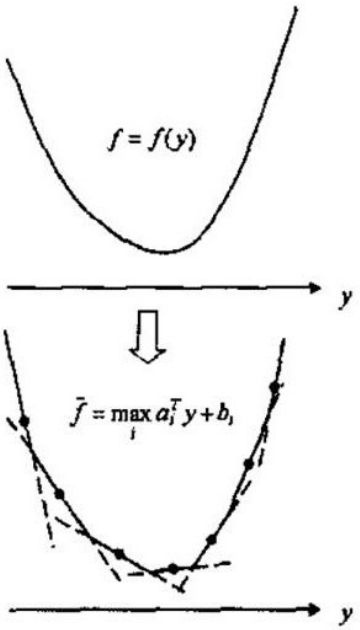
Sub-Space Modeling

- Use a piece-wise model to improve accuracy
  - Not necessary to be linear in each piece

TABLE I. SUB-SPACE MAP FOR ALL PARAMETERS BUT 1/gm

V_{gs} (V) \ L	0.5~1μm	1~2μm	2~5μm	5~10μm	10~21μm
0.4~0.5	mod1	mod4	mod7	mod10	
0.5~0.6	mod2	mod5	mod8	mod11	
0.6~0.7	mod3	mod6	mod9	mod12	
0.7~0.8	st1	st9	st17	st25	st33
0.8~1.0	st2	st10	st18	st26	st34
1.0~1.2	st3	st11	st19	st27	st35
1.2~1.4	st4	st12	st20	st28	st36
1.4~1.5	st5	st13	st21	st29	st37
1.5~1.6	st6	st14	st22	st30	st38
1.6~1.7	st7	st15	st23	st31	st39
1.7~1.8	st8	st16	st24	st32	st40

mod #: model in moderate inversion region; st #: model in strong inversion region



Experimental Results

- Sub-space based PWL (PWL_{sub}) modeling approach significantly improve the accuracy of g_m *and* g_{ds}

TABLE II. Comparison of Mean Relative Error (%) for Different Models

model	g_m	$1/g_m$	g_{ds}	I_{ds}	C_{gd}	C_{gs}	C_{gb}	C_{db}	C_{sb}	C_{dg}
PWL_{sub}	2.15	0.47	3.37	9.36	3.42	4.19	4.96	2.03	4.32	2.24
PWL [7]	---	1.70	9.40	---	---	3.10	---	---	---	---
GAP [8]	13.00	---	7.21	---	0.28	4.32	---	0.18	---	---

- The relative errors (RE) between the predicted results and HSPICE results are reduced (but still not accurate enough)

Performance	Spec.	Predic.	HSPICE	RE
A_V	≥ 40dB	76.7dB	80dB	4.1%
GBW	≥ 10MHz	10MHz	9.04MHz	10.1%
SR	≥ 5.7V/μs	6.52V/μs	6.16V/μs	5.8%
PM	≥ 70°	70°	65°	7.7%
I_{static}	minimum	15.89μA	16.75uA	5.1%

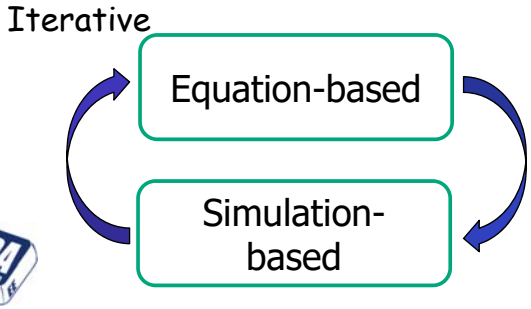
Not meet the spec …

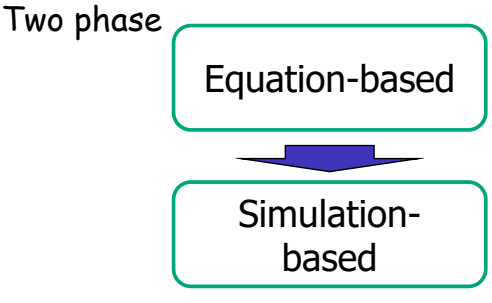
Outline

- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

Simulation-Equation Based Method

- Take advantage of both simulation and equation based methods
- Iterative simulation-equation-based (ISE) method
  - Use simulation results to correct performance equations
- Two-phase simulation-equation-based method
  - Phase 1: Do global search by equation-based search
  - Phase 2: Do local search by simulation-based search





Iterative Method

- Use initial performance evaluation equations to obtain an initial design
- Do simulation to verify the initial design and correct the parameters in the equations
- Use the corrected equations to do the search again until the result is converged
- Advantage:
  - Required number of simulations is greatly reduced
  - Accuracy can be improved with the feedback from simulation
- Disadvantage:
  - The optimality is not guaranteed → only near optimal result
  - Convergence rate depends on the quality of initial parameters

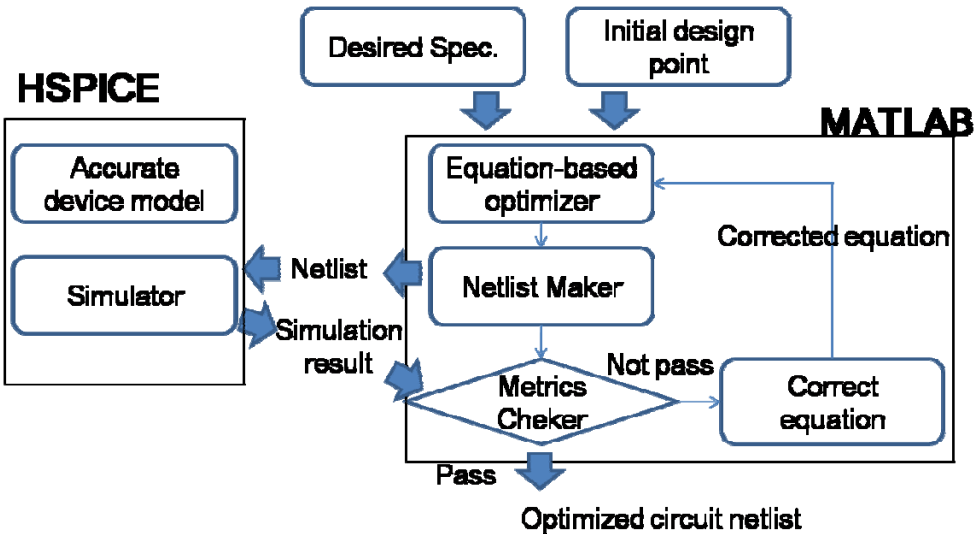
[1] M.H Maghami., F. Inanlou, R. Lotfi, "Simulation-Equation-Based Methodology for Design of CMOS Amplifiers Using Geometric Programming," IEEE Int'l Conf. on Electronics, Circuits and Systems, 2008, pp.360-363.
[2] T. Kahookar Toosi, E. Zhian Tabasy, H. Sarbishaei, R. Lotfi, "ISECAD: An Iterative Simulation-Equation-Based Opamp-Design CAD Tool", ISCAS, 2006.

Flow Chart of Iterative Method

- Evaluation model

Be correct by simulation result

  - Performance = f (design param., convergence param.)



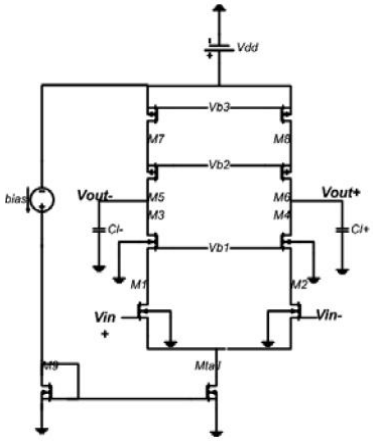
Source : M.H Maghami., F. Inanlou, R. Lotfi, "Simulation-Equation-Based Methodology for Design of CMOS Amplifiers Using Geometric Programming," IEEE Int'l Conf. on Electronics, Circuits and Systems, 2008, pp.360-363.

Experiment Results (Iterative)

- The optimization process has only 7 iterations
  - Finished in 30 seconds (faster than simulation-based method)
- The final result still does not meet spec. completely
- Although the prediction accuracy is already improved, it is still not accurate enough.

Not satisfy spec.

Performance measure	Spec.	GPCAD	Hspice
DC gain (db)	> 70	73.2	73.8
Unity gain BW (MHz)	> 100	100	92.7
SR (V/ μs)	> 20	62	62
Power (mW)	As minimum as possible	0.12	0.12

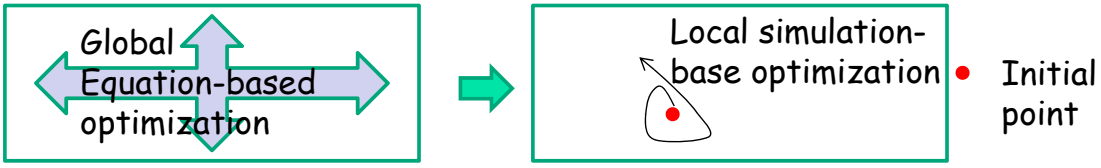


Source : M.H Maghami., F. Inanlou, R. Lotfi, "Simulation-Equation-Based Methodology for Design of CMOS Amplifiers Using Geometric Programming," IEEE Int'l Conf. on Electronics, Circuits and Systems, 2008, pp.360-363.

Two-Phase Method

- Phase 1: use performance evaluation equation to do global search
  - Obtain a rough estimation about the final result
- Phase 2: use simulation tool to do local search around the result in phase 1
  - Obtain optimal result with accurate performance evaluation
- Accuracy is improved a lot, but still requires long time in simulation-based local search

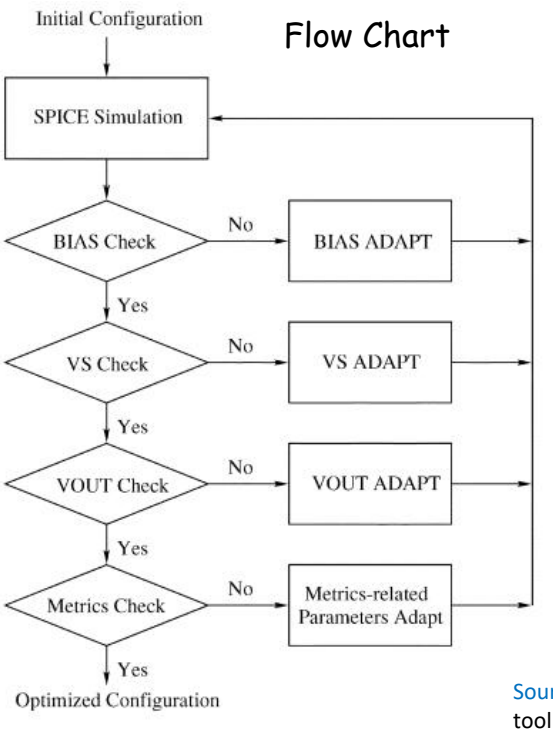
All design parameter space



Source : J. Yuan, N. Farhat, and J. Van der Spiegel, "GBOPCAD: a synthesis tool for high-performance gain-boosted opamp design," IEEE Trans. on Circuits and Systems-I, vol. 52, no. 8, Aug. 2005.

Experimental Result (2-Phase)

Flow Chart



Circuit spec

	Constraint
Gain (dB)	> 45
f_{ndgb} (MHz)	maximize
Biasing Currnet (mA)	< 0.8
Voltage Swing (V)	> 0.4
Output Voltage Tolerance (V)	< 0.1

	Equation -based result	Simulation -based result
	post-routine performance	post-SA per-formance
Gain(dB)	46.97	45.18
Unity-gain bandwidth (MHz)	816.9	888.94 Better!
Phase Margin (°)	60.0	60.0
Voltage Swing (V)	0.928	0.900
Biasing Current (mA)	3.6	3.6
Time (min.)	8	60 Cost long time

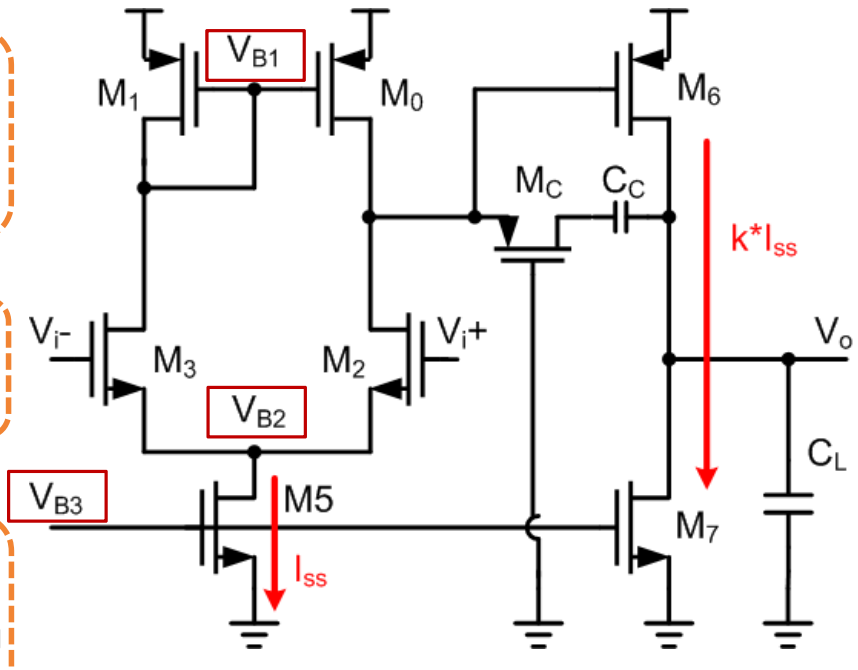
Source : J. Yuan, N. Farhat, and J. Van der Spiegel, "GBOPCAD: a synthesis tool for high-performance gain-boosted opamp design," IEEE Trans. on Circuits and Systems-I, vol. 52, no. 8, Aug. 2005.

Outline

- Motivation
- Knowledge-Based Design Automation
- Simulation-Based Optimization
- Pareto-Front-Based Optimization
- Equation-Based Optimization
- Simulation-Equation-Based Optimization
- Bias-Driven Optimization

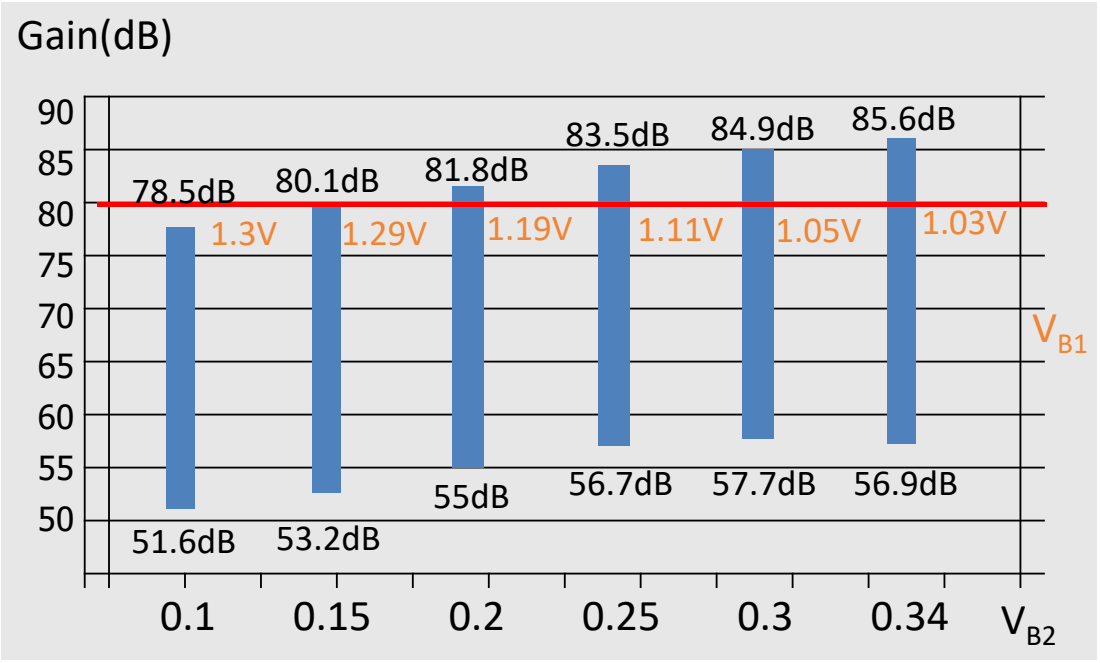
The Range of DC Operating Point

- For V_{B1}
  - $V_{DD}-V_{B1}>V_{TP}$
  - $V_{ICM}-V_{B1}<V_{TN}$
  - $V_{OCM}-V_{B1}<V_{TP}$
- For V_{B2}
  - $V_{B3}-V_{B2}<V_{TN}$
  - $V_{ICM}-V_{B2}>V_{TN}$
- For V_{B3}
  - $V_{B3}\geq V_{TN}$
  - $V_{B3}-V_{OCM}<V_{TN}$
  - $V_{B3}-V_{B2}<V_{TN}$



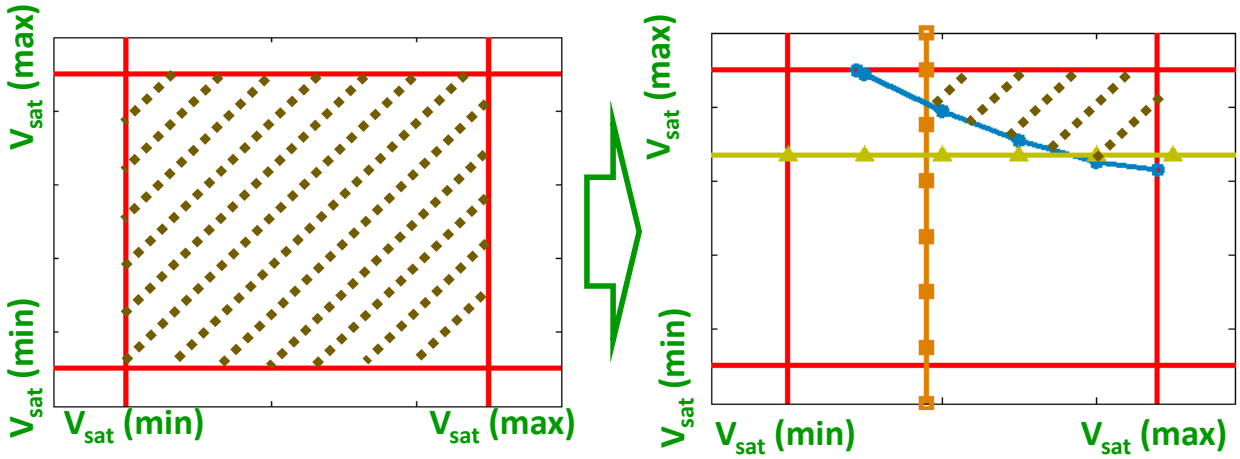
Performance vs. Operating Point

- Saturation mode cannot guarantee enough performance



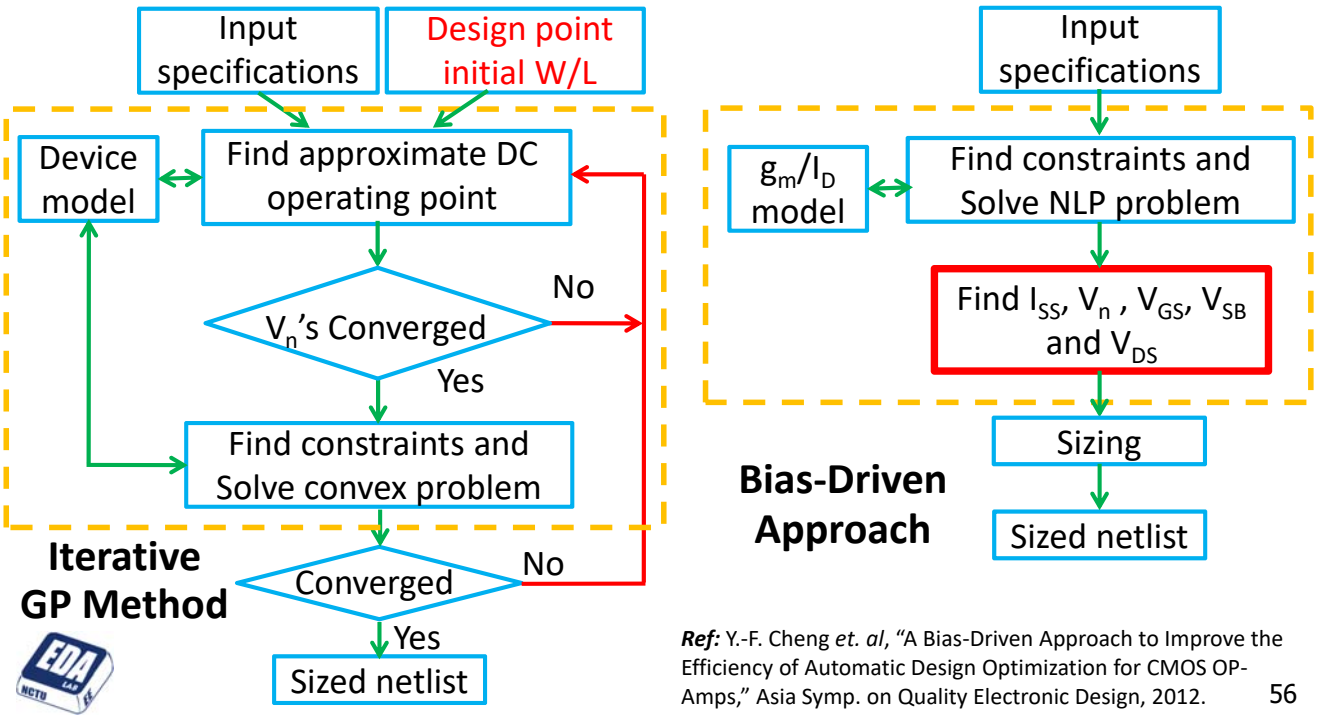
Operating Range Reduction

- The relationship between bias points and circuit performance can reduce the feasible operating range
  - Improve searching efficiency !!



Bias-Driven Optimization Flow

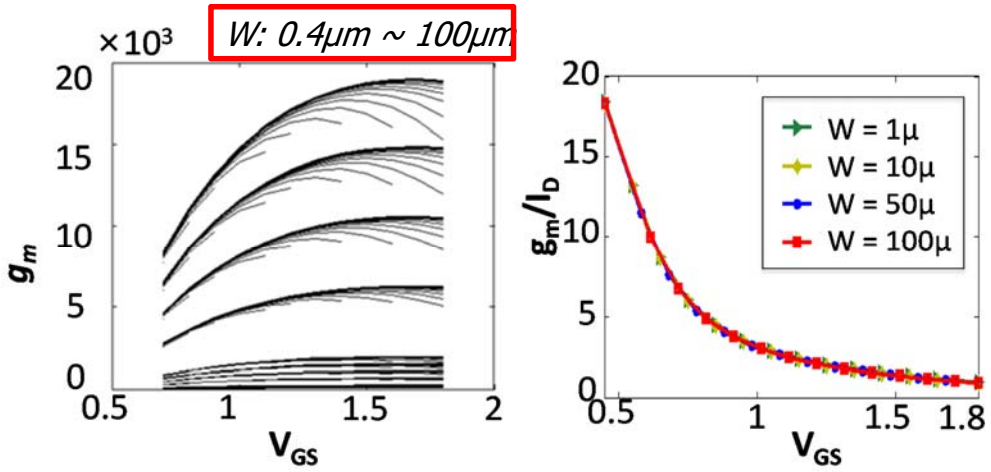
- Solve internal voltages instead of device sizes



Ref: Y.-F. Cheng *et. al*, "A Bias-Driven Approach to Improve the Efficiency of Automatic Design Optimization for CMOS OP-Amps," Asia Symp. on Quality Electronic Design, 2012.

g_m vs g_m/I_D

- Some transistor parameters (ex: g_m, g_{ds}, …) are hard to be modeled accurately → size dependent
- g_m/I_D is independent to the transistor sizes
  - Easier to be modeled

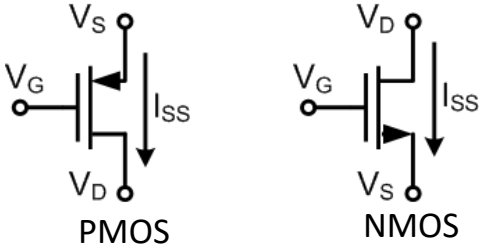


Source : G. Wei and R. Hornsey, " A power optimization method for CMOS Op-Amps using sub-space based geometric programming," DATE, 2010, pp. 508-513.

g_m/I_D Table Construction

- g_m/I_D value is decided when the three node voltages of a transistor are fixed
  - Can be recorded by a simple look-up table (LUT)
- Real simulation data is used to construct the tables
  - Improve modeling accuracy
- Table construction is a one-time effort
  - Can be used for all transistors with the same technology

Transistor Model



NMOS V_G = 0.9, V_{DD} = 1.8

	0	0.1	0.2	…	1.8
0	0	2.3	2.7	…	3.9
0.1	X	0	3.6	…	5.5
0.2	X	X	0	…	8.8
⋮	⋮	⋮	⋮	0	⋮
1.8	X	X	X	X	0

Performance Constraints

- Rewrite the equations with g_m/I_D and g_{ds}/I_D

Gain:

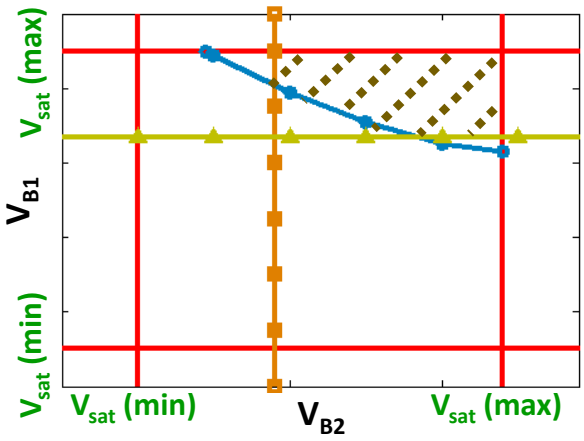
$$\frac{\frac{g_{m2}}{I_{D2}}}{\frac{g_{ds2}}{I_{D2}}+\frac{g_{ds0}}{I_{D0}}}\times\frac{\frac{g_{m6}}{I_{D6}}}{\frac{g_{ds6}}{I_{D6}}+\frac{g_{ds7}}{I_{D7}}}\geq Gain_{spec}$$

Phase margin (PM):

$$\frac{\frac{g_{m2}}{I_{D2}}\times(C_C+C_L)}{C_C\times\frac{g_{m6}}{I_{D6}}\times 2k}<\tan(\frac{\pi}{2}-PM_{SPEC})$$

Gain-Bandwidth (GB):

$$\frac{\frac{g_{m2}}{I_{D2}}\times I_{SS}}{C_C}\geq GB_{spec}$$



Objective Function

- Objective function

$$\alpha\times\sum_{i=1}^{n}\frac{(\frac{gm_i}{I_{Di}})\times I_{SS}}{(V_{gsi}-V_T)}+\beta\times\left((1+k)\times I_{SS}\times V_{DD}\right)$$

cost of area — cost of power

α : weight of area
β : weight of power

- Example
  - Optimize for power
    - α=0, β=1
  - Optimize for power and area
    - α=0.5, β=0.5

Solve NLP Problem

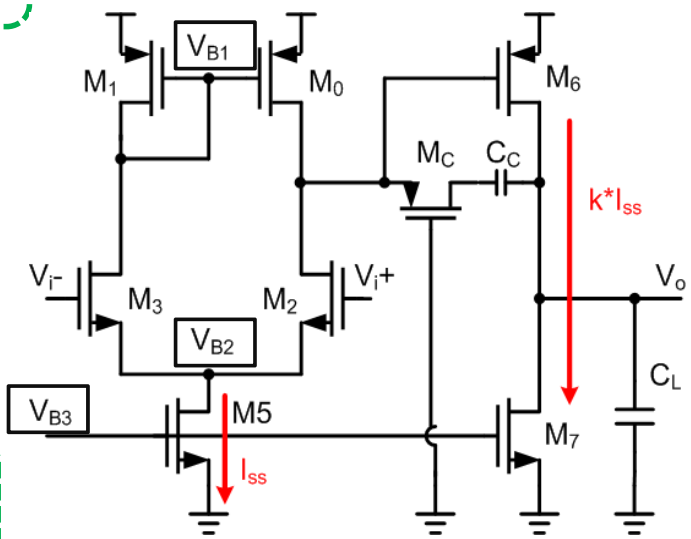
g_m/I_D model

Specification

Solve by Nonlinear Programming

I_{SS} = 5.25μA, k = 10
V_{B1} = 0.96V, V_{B2} = 0.31V V_{B3} = 0.69V, C_C=392fF

Vdd = 1.8V



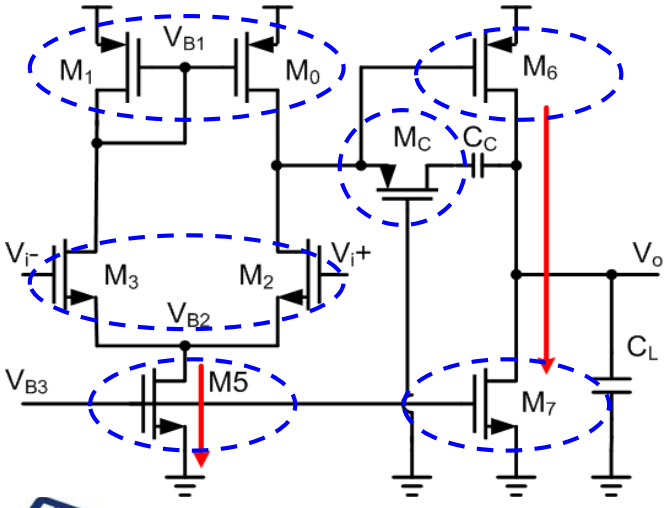
Device Sizing Table

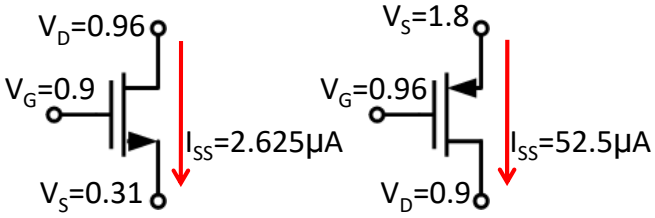
- Obtaining MOS size from current equation has large error
  - $I_d = K*(W/L)*(V_{gs}-V_t)^2$ → rough approximation
- Use simulation data to construct a lookup table
  - Given an unit-sized transistor, record its (V_{gs} , V_{ds}) vs I_d
  - I_d is proportional to the (W/L) ratio
    → $I_x / I_0 = (W/L)_x / (W/L)_0$
- Constructing the sizing table is also a one-time effort
  - Can be reused for all transistors with the same technology
  - A fast approach to reduce the equation error

Ex: NMOS sizing table			
V_{gs} (V)	V_{ds} (V)	I_d (μA)	S_x (μm / μm)
1.00	1.60	16.01	2.00 / 1.00
1.10	1.60	21.51	2.00 / 1.00
1.20	1.60	27.62	2.00 / 1.00

Transistor Sizing

- Search the sizing table to find the transistors size





M2

W*M	Current
1.62	2.52μA
1.64	2.55μA
1.66	2.58μA
1.68	2.62μA
1.7	2.65μA
1.72	2.68μA
1.74	2.71μA
1.76	2.74μA

M6

W*M	Current
20	49.4μA
20.25	50.0 μA
20.5	50.7 μA
20.75	51.3 μA
21	51.9 μA
21.25	52.5 μA
21.5	53.1 μA
21.75	53.7 μA

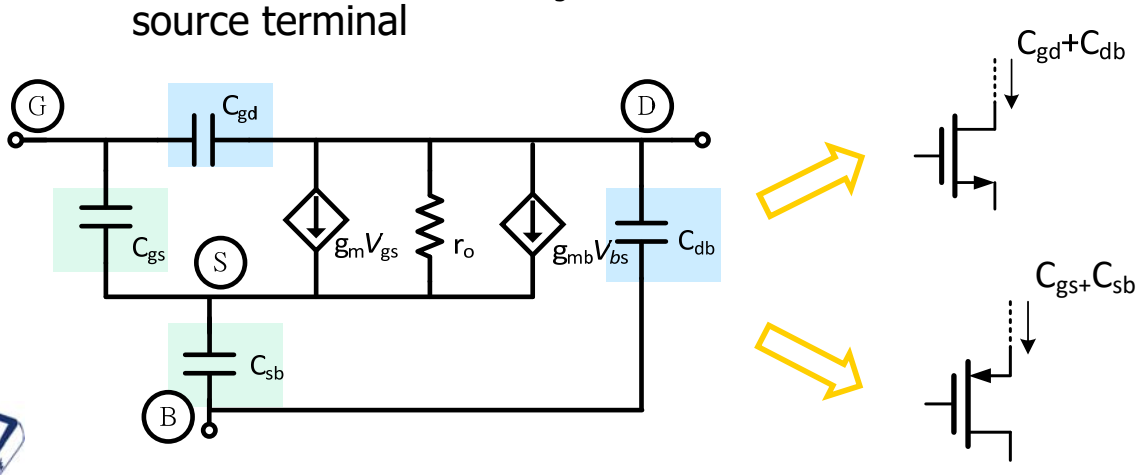
Optimize for Power and Area

- Bias-driven approach can meet all spec at the first time → efficient !!

Specification		SA	GP	Bias-driven
C_L (pF)	1	1	1	1
Gain (dB)	≥70	80.5	80.7	81.7
GB (MHz)	≥40	45.9	38.8	42.3
SR (MV/s)	≥30	31.5	28.5	36.8
OSR	≥1	1.28	1.04	1.14
PM (degree)	≥60	60.2	60.4	60
Power (μw)	min	160	120.1	113.9
Area (um^2)	min	55.48	37.69	32.7
Iteration		226	3	1
Time (s)		801	14.84	6.01
speed up ratio		1	53.9	133.3

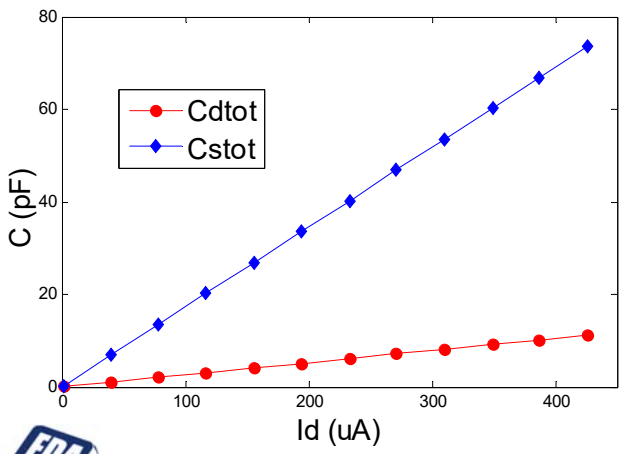
Parasitic Capacitance Consideration

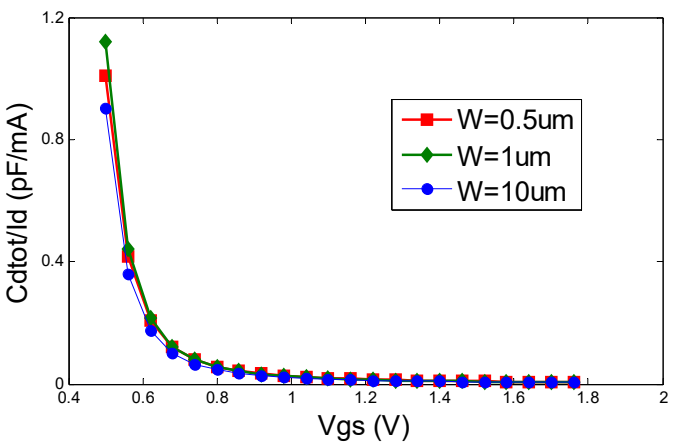
- The consideration of C_{gs}, C_{gd}, C_{db}, C_{sb} is necessary to predict bandwidth and phase margin
  - Output capacitance is $C_{gd}+C_{db}$ ($=C_{dtot}$) when output is at drain terminal
  - Output capacitance is $C_{gs}+C_{sb}$ ($=C_{stot}$) when output is at source terminal



Capacitance Prediction

- The value of capacitance increase with current
  → C_{stot}/I_D and C_{dtot}/I_D are almost constant, too
- Similar to g_m/I_D, a table is constructed to predict internal capacitance





*C_{stot}= C_{gs}+C_{sb}, C_{dtot}=C_{gd}+C_{db}

Body Effect Transconductance (g_{mb})

- Body-effect is the variation of the threshold voltage(V_{THN}) with V_{SB}

$$g_{mb} = \left(\frac{\partial i_D}{\partial v_{SB}} \right)_{V_{SB}=\text{const.}}^{I_D=\text{const.}} = \frac{\partial}{\partial v_{SB}} \left[\frac{KP_n}{2} \frac{W}{L} (V_{GS} - V_{THN})^2 \right]_{V_{SB}=\text{const.}}^{I_D=\text{const.}}$$

$$= KP_n \frac{W}{L} (V_{GS} - V_{THN}) \left(- \frac{\partial V_{THN}}{\partial v_{SB}} \right)$$

  - V_{THN} is considered in the parameter tables
- g_{mb} value is decided by DC voltage V_{SB}, but takes effects only when AC voltage $v_{sb} \neq 0$

Source : R. Jacob Baker, "CMOS circuit Design, Layout, and Simulation", Third edition.

Extend Parameter Tables

- C_{stot}/I_D , C_{dtot}/I_D and g_{mb} can also be modeled by the same method for g_m/I_D
- With the same index, the lookup table can be extended to include those non-ideal effects
- Multiplying the value in the table by the real drain current can get the required parameter
  - Simulation is not required in the synthesis flow

V_G	V_D	V_S	g_m/I_D	g_{ds}/I_D	C_{dtot}/I_D	C_{stot}/I_D	g_{mb}/I_D
:	:	:	:	:	:	:	:
1	0.5	0.1	4.11	0.41	2.24E-10	5.99E-11	1.15
1	0.5	0.2	6.11	0.53	4.33E-10	1.17E-10	1.61
1	0.5	0.3	10.27	0.77	1.18E-9	3.25E-10	2.58
:	:	:	:	:	:	:	:

Accuracy Improvement

		Gain (dB)	Pole1 (kHz)	Pole2 (MHz)	UGBW (MHz)	Pm (˚)
Spec		>75			>100	>60
simulation		76.60	16.86	187.40	100.23	61.65
Predict1	value	73.83	x	x	133.55	x
	Error(%)	3.62	x	x	33.24	x
Predict2	value	73.83	23.34	143.86	95.48	56.41
	Error(%)	3.62	38.42	23.33	4.74	8.50
Predict3	value	75.90	18.36	185.82	100.77	61.54
	Error(%)	0.91	8.90	0.84	0.54	0.18

Predict1: prediction without C_{stot}, C_{dtot} prediction & using rough performance equation
Predict2: prediction with C_{stot}, C_{dtot} prediction & using transfer function
Predict3: including g_{mb} considering in the transfer function

Source : Y.-F. Cheng, "A Bias-Driven OP-Amp Sizing Approach with Improved Prediction of Frequency Response and Channel Length Effect", Master thesis of NCU, 2012.

More References ...

- Books:
  - R. A. Rutenbar, G. G. E. Gielen, and B. Antao, Eds., "Computer-Aided Design of Analog Integrated Circuits and Systems", Hoboken, NJ: Wiley-IEEE, Apr. 2002.
  - H. E. Graeb, "Analog Design Centering and Sizing", New York: Springer-Verlag, 2007.
- Survey Papers:
  - G. G. E. Gielen, R. A. Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", Proceedings of the IEEE, vol. 88, no. 12, pp. 1825-1852, Dec. 2000.
  - R. A. Rutenbar, G. G. E. Gielen, J. Roychowdhury, "Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs", Proceedings of the IEEE, vol. 95, no. 3, pp. 640-669, Mar. 2007.