

Copyright © 1995, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**CONSTRAINT-DRIVEN ANALYSIS AND SYNTHESIS
OF HIGH-PERFORMANCE ANALOG IC LAYOUT**

by

Edoardo Charbon

Memorandum No. UCB/ERL M95/115

19 December 1995

COVER PAGE

**CONSTRAINT-DRIVEN ANALYSIS AND SYNTHESIS
OF HIGH-PERFORMANCE ANALOG IC LAYOUT**

Copyright © 1995

by

Edoardo Charbon

Memorandum No. UCB/ERL M95/115

19 December 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Constraint-Driven Analysis and Synthesis of High-Performance Analog IC Layout

by

Edoardo Charbon

**Doctor of Philosophy in Engineering-Electrical Engineering
and Computer Sciences**

University of California at Berkeley

Professor Alberto Sangiovanni-Vincentelli, Chair

Performance in analog circuits is often critically linked to the physical implementation and the technology used for fabrication. While for digital circuits several synthesis techniques have been proposed and are of common use, in the analog domain, most of the high performing systems and chips are designed largely by hand.

In this prospective we have proposed a methodology for the semi-automated synthesis of full-custom analog IC layout at medium to high frequencies. The methodology guarantees that all performance constraints be met when feasible, otherwise infeasibility is detected as soon as possible, thus providing a robust and efficient design environment. In the proposed approach, performance specifications are translated into lower-level bounds on parasitics or geometric parameters, using sensitivity analysis. Bounds can be used by a set of specialized layout tools performing stack generation, placement, routing, compaction and extraction. For each tool, a detailed description is provided of its functionality, of the way constraints are mapped and enforced, and of its impact on the design flow.

A major advantage of the methodology is the reduction of time-consuming re-design loops often needed in purely bottom-up approaches, provided that accurate models for all relevant parasitics are available. For this reason, a wide variety of compact and highly sophisticated parasitic models has been generated to cover a sufficiently wide frequency spectrum. The effects of currents generated by high-speed digital switching circuits and the mechanisms governing noise injection have also been modeled and used to drive the design towards viable solutions.

A considerable number of examples drawn from industrial applications has been generated and fabricated in various technologies to illustrate the effectiveness of the approach.



Professor Alberto Sangiovanni-Vincentelli
Dissertation Committee Chair

a Tokiko

Contents

List of Figures	x
List of Tables	xv
1 Introduction	1
1.1 Computer-Aided Design of Engineering Systems	1
1.2 Physical Assembly of Analog and Mixed-Signal ICs	2
1.3 Bottom-Up vs. Top-Down Approaches	3
1.4 Generalized Constraint-Driven Layout Design	4
1.5 Theoretical Aspects of Constraint-Based Approaches	6
1.6 Organization of the Dissertation	7
2 Literature Survey	8
2.1 The Origins of Computer-Aided-Design	8
2.1.1 Circuit Simulation	8
2.1.2 Digital Timing Analysis and Event-Driven Simulation	9
2.1.3 Circuit Optimization	10
2.2 Early Work in Computer-Aided-Design for Analog ICs	11
2.2.1 Silicon Compilation	11
2.2.2 Knowledge-Based Techniques	11
2.2.3 Algorithmic Methodologies	12
2.3 The First Complete Design Systems	12
2.3.1 IDAC/ILAC	12
2.3.2 OPASYN	13
2.3.3 OASYS/ACACIA	14
2.4 Evolution of Approaches	15
2.4.1 Silicon Compilation and Module Generation	15
2.4.2 Knowledge-Based Systems	16
2.4.3 Hybrid and Human-Driven Systems	17
2.5 Constraint-Based Approaches	19
2.5.1 Foundations	19
2.5.2 First Constraint-Driven Design Tools	19
2.5.3 Later Implementations	19

3	Generalized Constraint Generation	21
3.1	Problem Formulation	22
3.2	Mapping Specifications onto Layout Constraints	23
3.3	Methods for the Evaluation of Sensitivities	25
3.4	Constraint Generation Engine	26
3.4.1	Absolute Parasitic Constraints	26
3.4.2	Constraints on Mismatch	27
3.4.3	Device and Interconnect Symmetry	31
3.5	Constraint Generation: A Case Study	36
4	Placement	40
4.1	Evolution from the Digital to the Analog Domain	41
4.1.1	Placement Problem Formulation	41
4.1.2	Constructive or Schematic-Driven Techniques	43
4.1.3	Branch-and-Bound Search and Partitioning-Based Techniques	44
4.1.4	Quadratic Optimization-Based Techniques	45
4.1.5	Iterative Improvement Techniques	46
4.2	Simulated Annealing and Analog Placement	48
4.2.1	Terminology	48
4.2.2	Characterizing Analog Constraints	50
4.2.3	Slicing-Tree vs. Flat Representation of the Workspace	53
4.3	Modifying Basic Algorithms	54
4.3.1	Standard Features	54
4.3.2	Modifications of the Standard Features	59
4.3.3	Configuration Space	60
4.3.4	Cost Function	62
4.3.5	Move-Set	65
4.4	Module Generation	70
4.4.1	Terminology	71
4.4.2	Stack-Generation Algorithm	71
4.4.3	Analog Constraints and Computational Cost	74
4.4.4	Importance of Creating Alternative Modules	76
4.4.5	Module Replacement Criteria	78
4.5	Performance Models and Constraint Enforcement	81
4.5.1	Deterministic Model	82
4.5.2	Non-Deterministic Parasitic Constraint Enforcement	87
4.6	Substrate-Aware Placement	90
4.6.1	Modeling Switching Noise	91
4.6.2	Modifying the Original Placement Algorithm	92
4.6.3	Advanced Features: Thermal Analysis	97
4.7	Placement with Analog Constraints: A Case Study	97
4.7.1	Module Generation	97
4.7.2	Placement Algorithm	100

5	Routing	102
5.1	Performance-Driven Analog Routers	103
5.2	Maze Routing and the A* Algorithm	104
5.3	Routing of RF Circuits and MMICs	107
5.4	Parasitic Modeling and Constraint Generation	109
5.5	Routing Phases	113
5.5.1	Constructive Routing	113
5.5.2	Refinement	115
5.6	RF and Microwave Routing: A Case Study	119
6	Symbolic Compaction	123
6.1	Compaction Problem Formulation	124
6.2	Compaction with Analog Constraints	126
6.3	Constraint Enforcement Techniques	127
6.3.1	DRC	127
6.3.2	Constraints on Stray Capacitances	128
6.3.3	Preservation of Electrostatic Shields	132
6.3.4	Symmetry Constraints	133
6.4	Algorithmic Considerations	134
6.5	Wire Length Minimization	136
6.6	Compaction with Analog Constraints: A Case Study	136
7	Extraction	139
7.1	General Extraction Methodology	140
7.1.1	Extraction Tools and Organization	140
7.1.2	Constraint-Based Schematic Simplification	141
7.2	CMOS Parasitic Modeling	143
7.2.1	Substrate Capacitance	143
7.2.2	Crossover Configuration	144
7.2.3	Parallel Lines on the Same Layer	145
7.2.4	Interconnect Lines on Different Layers	145
7.3	Technology Gradient Effects: Mismatch Modeling	146
7.4	RF Parasitic Modeling	149
7.4.1	Modeling Single Interconnect Lines	150
7.4.2	Modeling Coupled Microstrip Lines	151
7.4.3	Modeling Microstrip Discontinuities	152
7.4.4	Modeling 3-D Discontinuities	153
7.5	Superconductor Parasitic Modeling	153
7.5.1	Analytical Model Generation for Superconducting Inductances	153
7.5.2	Modeling a Single Line	155
7.5.3	Modeling Coplanar Lines	155
7.5.4	Modeling Non-overlapping Lines	156
7.5.5	Modeling Overlapping Lines	156
7.5.6	Model Characterization	157
7.5.7	Example of Complete Superconductor Extraction	158

8	Substrate-Aware Analysis and Optimization	159
8.1	Importance of Substrate in Mixed-Signal Systems	160
8.2	Modeling Substrate Transport and Thermal Behavior	163
8.2.1	Background	163
8.2.2	Green's Function-Based Methods: Basics	167
8.2.3	Using the Green's Function in Substrate Analysis	168
8.2.4	Computing the Green's Function in Multi-Layered Substrates	170
8.2.5	Substrate Extraction Algorithm	174
8.2.6	Thermal analysis	176
8.2.7	Schemes for Efficient Solution of Large Substrate Problems	177
8.3	Switching Noise Sources	179
8.3.1	Substrate Injection Mechanisms	180
8.3.2	Substrate Reception Mechanisms	184
8.4	Substrate Conductivity and Technology	185
8.5	Techniques for Substrate-Aware Optimization	188
8.5.1	Constraint Generation for Substrate Parasitic Effects	188
8.5.2	Substrate Transport Evaluation in Iterative Algorithms	191
8.5.3	Template-Based Substrate Extraction	197
8.5.4	Evaluating Effects of Scaling and Technology Migration	203
9	Experimentation	211
9.1	Analog Benchmark Library	211
9.1.1	COMPL	212
9.1.2	FASTCOMP	214
9.1.3	MPH	216
9.1.4	Other CMOS Benchmarks	221
9.2	Mixed-Signal Benchmark Library	222
9.2.1	The RAMDAC System	223
9.2.2	The $\Sigma - \Delta$ CONVERTER System	237
9.3	RF and Microwave Benchmark Library	246
10	Conclusions	250
10.1	Conclusions	250
10.2	Future Work	251
A	Convergence of Modified Placement Algorithms	253
A.1	Modification of Search Space	253
A.2	Substrate-Aware Placement	254
B	Compaction Roundoff Calculations	255
C	Green's Function Related Theory	258
C.1	Non-Zero Depth Contact Calculation	258
C.2	Scaling Coefficient of Induction Matrix	259

D Sensitivity Analysis	261
D.1 Canonical Representation of Performance	261
D.2 Coefficient of Potential and Technology Parameters	262
E RF Parasitic Models	265
E.1 Closed Form Expressions for Microstrip Lines	265
E.2 Microstrip Line Discontinuities	267
E.3 3-D Discontinuities	267
F Superconducting Models	268
F.1 Single Line	268
F.2 Coplanar Lines	269
F.3 Non-Overlapping Lines	269
F.4 Overlapping Lines	269
G Software Availability	270
Bibliography	271

List of Figures

1.1	<i>Constraint-driven layout design: Traditional design partition into tasks has been modified by adding information paths between layout phases.</i>	4
1.2	<i>The organization of each layout phase. The internal feedback path provides information to the constraint generator. External feedback paths provide information on the reasons of failure to meet performance specifications. . . .</i>	5
3.1	<i>Pseudo-code of the graph-based symmetry constraint generator</i>	33
3.2	<i>Degenerate path</i>	35
3.3	<i>Semi-degenerate path</i>	35
3.4	<i>Clocked comparator COMPL</i>	36
4.1	<i>Object definition</i>	41
4.2	<i>Pseudo-code of a generic iterative algorithm</i>	47
4.3	<i>Pseudo-code of a generic SA algorithm</i>	48
4.4	<i>Normalized energy landscape in typical placement problem</i>	49
4.5	<i>Topological constraints in analog circuit design: (a) symmetry and matching; (b) well minimization</i>	51
4.6	<i>Slicing-tree space representation for layout optimization algorithms</i>	53
4.7	<i>Accounting for routing channels: the Halo algorithm</i>	55
4.8	<i>Routing estimation techniques</i>	56
4.9	<i>Dynamically adjustable terminals in modules</i>	57
4.10	<i>Placement flow diagram for PUPPY-A</i>	59
4.11	<i>Dynamically adjustable modules available to the placement tool</i>	60
4.12	<i>Library of CMOS modules</i>	61
4.13	<i>SA and well definition</i>	63
4.14	<i>Placement using virtual symmetry axes</i>	64
4.15	<i>Abutment and separation of modules</i>	66
4.16	<i>Updating symmetry axes during the annealing</i>	67
4.17	<i>Updating well regions during the annealing</i>	67
4.18	<i>Updating contact locations</i>	68
4.19	<i>Modeling contacts within modules</i>	68
4.20	<i>Contact resistance as a function of relative position within a module</i>	69
4.21	<i>Derivation of feasibility region for the contact realization</i>	69

4.22	<i>(a) Mapping of a circuit schematic onto a graph; (b) Chaining algorithm in LDO (Courtesy of Enrico Malavasi)</i>	72
4.23	<i>Splitting of large transistors (Courtesy of Enrico Malavasi)</i>	73
4.24	<i>(a) Transistor split in two modules; (b) Layout minimizing the capacitance of net D; (c) Layout minimizing the capacitance of net S</i>	75
4.25	<i>Enforcement of symmetry in LDO: (a) first alternative based on a common-centroid design style; (c) second fully symmetric layout; (b) trade-off</i>	77
4.26	<i>Example of a folded cascode opamp. The bubbles represent all sub-circuits created by the module generator on the ground of well type</i>	78
4.27	<i>Alternative implementations of the differential pair and its active load</i>	79
4.28	<i>Alternative implementations of interconnect: (a) on metal1 or metal2; (b) on metal1 and poly; (c) on metal1 and metal2</i>	83
4.29	<i>PUPPY-A's shaping function with twofold interconnect implementation</i>	85
4.30	<i>Coupling (a) without and (b) with vertical shielding. Indirect shielding effects due to the presence of the other interconnect have been added to the cross-over capacitance models</i>	85
4.31	<i>(a) Cross-coupling between MET1 and MET2 interconnect; (b) configuration avoiding cross-over; (c) heuristic for crossover probability estimation</i>	87
4.32	<i>(a) Simple injection model; (b) Proposed injection model</i>	92
4.33	<i>Mapping of substrate onto fully connected graph $G_S(V, E)$.</i>	93
4.34	<i>(a) Initial contact grid; (b) Reshuffling of contacts at high temperatures; (c) Resulting grid at lower temperatures</i>	94
4.35	<i>Resistive network reacting to high-temperature and low temperature contact reshuffling</i>	95
4.36	<i>Heuristic for the combined use of Sherman-Morrison and gradient-based methods</i>	95
4.37	<i>Small number of contacts translating within the workspace</i>	96
4.38	<i>Clocked comparator COMPL - Two alternative full-stacked implementations</i>	98
4.39	<i>Placement of comparator COMPL obtained with PUPPY-A</i>	100
5.1	<i>Propagation of path length estimate from source s to target t through node x</i>	105
5.2	<i>Generic A* routing algorithm</i>	106
5.3	<i>Grid allocation in a typical maze router</i>	106
5.4	<i>Flow diagram of the tool</i>	109
5.5	<i>(a) Microwave specification; (b) Flexibility function for constrained optimization</i>	110
5.6	<i>Interconnect model for microstripline with multiple bends</i>	112
5.7	<i>Expansion of wiring in the workspace (a) with coarse and (b) with tight constraints</i>	114
5.8	<i>Stub construction: (a) development from source; (b) completion</i>	115
5.9	<i>Effects of a via structure on a to-be-built interconnect line. (a) Structure set-up; (b) Interconnect characteristic impedance deviation as a function of the location relative to the via; (c) Feasibility zone</i>	116
5.10	<i>Distributed parasitics acting on interconnect determine feasibility zones</i>	117
5.11	<i>Refinement algorithm in CORAL</i>	118

5.12	<i>Expansion of a stub within feasibility zone</i>	119
5.13	<i>Schematic of TWA</i>	119
5.14	<i>Layout after CORAL's constructive routing</i>	121
5.15	<i>Final layout of TWA</i>	122
5.16	<i>Manual layout of TWA</i>	122
6.1	<i>Mono-dimensional compaction algorithm</i>	125
6.2	<i>Orthogonal mono-dimensional compaction iterations</i>	125
6.3	<i>Iterative mono-dimensional compaction algorithm</i>	126
6.4	<i>Constraint graphs associated with (a) horizontal and (b) vertical constraints. The symbols L/R and T/B relate to the left/right and top/bottom coordinates respectively</i>	127
6.5	<i>Topological and parasitic Constraints: (a) symmetry and matching; (b) parasitic; (c) lateral shielding; (d) vertical shielding</i>	129
6.6	<i>Required spacing for controlling capacitive coupling</i>	129
6.7	<i>Algorithm for the insertion of additional spacing for wire decoupling</i>	130
6.8	<i>Pseudo-code of procedure modify-graph</i>	131
6.9	<i>Minimum and maximum wire spacing constraints deriving from connections to fixed-distance terminals</i>	132
6.10	<i>Compacted layout of COMPL with all analog constraints enforced</i>	138
7.1	<i>Methodology of selective extraction</i>	140
7.2	<i>Simple interconnect line</i>	144
7.3	<i>Crossover configuration</i>	144
7.4	<i>Parallel interconnect lines</i>	145
7.5	<i>Interconnect lines on different layers: (a) non-overlapping; (b) overlapping</i>	146
7.6	<i>Single microstrip line over lossy substrate</i>	150
7.7	<i>Coupled microstrip lines</i>	151
7.8	<i>Typical Discontinuities in RF and microwave circuits</i>	152
7.9	<i>Analytical model generation for superconducting inductances</i>	154
7.10	<i>Single line</i>	155
7.11	<i>Coplanar lines</i>	156
7.12	<i>(a) Overlapping; (b) Non-overlapping lines</i>	157
7.13	<i>(a) Layout of a two-junction SQUID; (b) Extracted schematic using INDEX</i>	158
8.1	<i>Main and spurious currents in an inverter during transition</i>	161
8.2	<i>Impact of spurious noise signals to a differential pair</i>	162
8.3	<i>Substrate modeling using RC mesh (Courtesy of Ranjit Gharpurey)</i>	166
8.4	<i>Substrate boundaries and contact resistance modeling</i>	169
8.5	<i>Partition schemes for substrate contacts</i>	171
8.6	<i>Multi-layer doping profiles (Courtesy of Ranjit Gharpurey)</i>	172
8.7	<i>Discretization of non-abrupt doping profiles</i>	174
8.8	<i>Pseudo-code of the substrate resistance extraction algorithm</i>	175
8.9	<i>Discretization of the substrate surface</i>	176
8.10	<i>Direct and indirect current-flow paths (Courtesy of Ranjit Gharpurey)</i>	177

8.11	<i>Pseudo-code of the simplified substrate extraction scheme</i>	179
8.12	<i>Partitioning of substrate for the simplification algorithm</i>	180
8.13	<i>Typical IC substrates: (a) high-resistivity; (b) low-resistivity</i>	181
8.14	<i>Injection and reception mechanisms (Courtesy of Ranjit Gharpurey)</i>	182
8.15	<i>Body Effect in MOSFETs (Courtesy of Ranjit Gharpurey)</i>	184
8.16	<i>Storing one DCT for nominal parameter set and a number of DCTs for each computed sensitivity</i>	187
8.17	<i>Pseudo-code of the substrate sensitivity extraction algorithm</i>	188
8.18	<i>The principle and modeling of local generators</i>	189
8.19	<i>Constraint check</i>	191
8.20	<i>Contact transformation and modifications in the potential matrix</i>	192
8.21	<i>Sensitivity of resistive macro-model from transformation of a component and its contacts</i>	194
8.22	<i>Single contact moving in direction \mathbf{v} by an infinitesimal amount</i>	195
8.23	<i>Computation of δp_{ij}</i>	195
8.24	<i>200x200 DCT of the Green's Function for a commercial substrate</i>	196
8.25	<i>Computation of update matrix $\delta \mathbf{c}$ based on contact displacement relative to template</i>	197
8.26	<i>Pseudo-code of the template-based substrate extraction algorithm</i>	197
8.27	<i>Speed-up mechanism for the extraction of large substrates</i>	198
8.28	<i>Elimination of all non-critical conductances and contacts</i>	199
8.29	<i>Pseudo-code of the modified template-based substrate extraction algorithm</i>	201
8.30	<i>Similar landscape and displacement of contact i and j</i>	202
8.31	<i>Partitioning of substrate to minimize the number of different contacts for which $\nabla_{\mathbf{v}} \mathbf{c}$ need be computed explicitly</i>	203
8.32	<i>Accuracy in function of the distance of the true contact from the pre-computed contact</i>	204
8.33	<i>(a) Two-dimensional scaling in the event of re-design; (b) Three-dimensional scaling in technology migration</i>	204
8.34	<i>Plot of the dependence of each component of the $\bar{\mathbf{R}}$ matrix as a function of the contact layer depth</i>	205
8.35	<i>Plot of the dependence of $\bar{\mathbf{R}}$ as a function of the contact layer depth and related sensitivities</i>	206
8.36	<i>Scaling in x- and y-direction. Relocation of contacts and area scaling</i>	207
8.37	<i>Sensitivity of entry \mathbf{Y}_{55} in a 10x10 grid as a function of a translation in (a) x- and (b) y- direction of all the contacts in the grid</i>	208
9.1	<i>Complete layout of COMPL, (a) without enforcement, (b) with enforcement of analog constraints</i>	214
9.2	<i>Schematic of the clocked comparator FASTCOMP</i>	215
9.3	<i>Complete layout of FASTCOMP, with enforcement of all analog constraints</i>	217
9.4	<i>Details of the routing of FASTCOMP. Left: no parasitic constraints enforced. Right: all parasitic constraints successfully enforced</i>	217
9.5	<i>Schematic of MPH.</i>	218
9.6	<i>Complete layout of MPH, obtained enforcing all analog constraints</i>	220

9.7	<i>PLL schematic</i>	223
9.8	<i>VCO block diagram and schematic of one delay cell</i>	224
9.9	<i>PFD schematic</i>	224
9.10	<i>LPF schematic: (a) no substrate coupling; (b) with substrate coupling</i>	224
9.11	<i>Programmable divider: (a) block diagram; (b) single-phase flip-flop</i>	225
9.12	<i>VCO architecture generated by VCOGEN</i>	227
9.13	<i>Layout of eight-stage VCO</i>	228
9.14	<i>(a) Charge pump (CP); (b) Low-pass filter (LPF)</i>	229
9.15	<i>(a) Output signal of divider; (b) Injected current; (c) Model for substrate injection</i>	231
9.16	<i>Evaluation of peak-to-peak switching noise at the receptor site</i>	231
9.17	<i>Estimated level of switching noise signal amplitude as a result of the cumulative injection of the dividers during the annealing: (a) high temperature; (b) medium temperature; (c) low temperature</i>	232
9.18	<i>Error in substrate injection estimation using: (a) combined heuristic; (b) gradient-based method only. Evolution of total substrate violations using: (c) combined heuristic; (d) no substrate control</i>	233
9.19	<i>Placed PLL within the RAMDAC</i>	234
9.20	<i>Placed and routed PLL within the RAMDAC</i>	235
9.21	<i>Dependence from doping levels: (a) sub-set of R; (b) sensitivity</i>	235
9.22	<i>Dependence from contact depth: (a) sub-set of R; (b) sensitivity</i>	236
9.23	<i>Dependence from doping profiles: (a) sub-set of R; (b) sensitivity</i>	236
9.24	<i>$\Sigma - \Delta$ CONVERTER architecture</i>	237
9.25	<i>Schematic of the OTA</i>	238
9.26	<i>Schematic of the bias circuitry</i>	239
9.27	<i>Schematic of the comparator used in the $\Sigma - \Delta$ CONVERTER</i>	240
9.28	<i>Schematic of clock generator</i>	241
9.29	<i>Schematic of latch</i>	241
9.30	<i>Placement by PUPPY-A of the OTA (Courtesy of H. Chang and E. Felt)</i>	242
9.31	<i>Placed, routed, compacted OTA (Courtesy of H. Chang and E. Felt)</i>	243
9.32	<i>Layout of the clock generator (Courtesy of H. Chang and E. Felt)</i>	243
9.33	<i>Layout of the latch (Courtesy of H. Chang and E. Felt)</i>	244
9.34	<i>Layout of $\Sigma - \Delta$ CONVERTER (Courtesy of H. Chang and E. Felt)</i>	245
9.35	<i>Schematic of PCN38</i>	248
9.36	<i>Performance of PCN38</i>	248
9.37	<i>Final layout of PCN38</i>	248
C.1	<i>Non-zero depth contacts and dimensions</i>	259
F.1	<i>General configuration</i>	268

List of Tables

3.1	<i>Notation for parasitics and performance functions</i>	32
4.1	<i>Capacitances in the stacks generated for the clocked comparator COMPL . . .</i>	98
4.2	<i>Conditions of operation for the placement tool used in the synthesis path. The symbols P, S and M denote parasitic, symmetry and matching constraints, respectively</i>	101
5.1	<i>Performance specifications for TWA</i>	120
5.2	<i>Constraints on critical interconnect lines</i>	120
5.3	<i>CPU times required for the synthesis of TWA on a DEC Station 5000/240 .</i>	121
5.4	<i>Estimated performance of circuit TWA after layout completion</i>	121
6.1	<i>Comparison of CPU time for graph + LP vs. LP alone</i>	135
7.1	<i>Comparison between extracted and hand-computed parasitics</i>	158
8.1	<i>Substrate extraction in presence of varying technology parameters using method I (full extraction) and method II (sensitivity-based extraction)</i>	207
8.2	<i>Mean and variance of the entries of matrix \bar{R} as a function of depth variance. All values are referred to a mean depth of $1\mu m$. The execution times are reported for a uniform 10×10 contact grid</i>	209
8.3	<i>Selection of most suitable technology based on the probability of satisfying all constraints on substrate coupling resistances</i>	210
9.1	<i>COMPL: bounds on capacitive and resistive parasitics</i>	212
9.2	<i>Conditions of operation for the routing and compaction tools used in the synthesis path. The symbols P, S and M denote parasitic, symmetry and matching constraints, respectively. The net scheduling is based on a cost function which accounts for the "difficulty" of enforcing a set of desired constraints on a given net</i>	213
9.3	<i>COMPL: performance</i>	213
9.4	<i>COMPL: CPU time for each layout phase</i>	213
9.5	<i>FASTCOMP: bounds on capacitive and resistive mismatch</i>	216
9.6	<i>FASTCOMP: performance</i>	218
9.7	<i>FASTCOMP: CPU time for each layout phase</i>	218

9.8	<i>MPH: performance</i>	219
9.9	<i>MPH: CPU time for each layout phase</i>	220
9.10	<i>Nominal performance of benchmark circuits</i>	221
9.11	<i>Constraint generation for the given benchmark circuits</i>	222
9.12	<i>Measure of success of the performance-driven methodology</i>	222
9.13	<i>PLL specifications</i>	226
9.14	<i>Parameter constraints obtained for the VCO by behavioral optimization of the PLL</i>	226
9.15	<i>Constraints obtained by the sensitivity analysis</i>	227
9.16	<i>CPU times for the design and module generation obtained on a DEC Station 5000/125 and on a DEC AlphaServer 2100 5/250 (†)</i>	228
9.17	<i>Placement statistics obtained on a DEC AlphaServer 2100 5/250</i>	232
9.18	<i>Noise injector and receptor statistics in the components of the PLL</i>	234
9.19	<i>CPU times on a DEC AlphaServer 2100 5/250 for the trend analysis for the proposed experiments on the PLL with 311 noise sources / receptors. The CPU times include DCT, parameter and sensitivity computation. For the calculation of 311 contacts the inversion of matrix \mathbf{P} was performed in 1525.0 seconds.</i>	237
9.20	<i>$\Sigma - \Delta$ CONVERTER design specifications</i>	238
9.21	<i>Design constraints for the integrator</i>	240
9.22	<i>Design constraints for the comparator</i>	240
9.23	<i>Estimated man-time for an unexperienced tool-user to perform the layout of the $\Sigma - \Delta$ CONVERTER (Courtesy of H. Chang and E. Felt)</i>	244
9.24	<i>$\Sigma - \Delta$ CONVERTER experiment results</i>	246
9.25	<i>Analytical models used in synthesis for parasitic control</i>	246
9.26	<i>Performance of a set of commercial RF benchmarks</i>	247
9.27	<i>Worst-case performance degradation form nominal of PCN38</i>	247
9.28	<i>Constraints on critical interconnect lines in PCN38 as computed using PAR-CAR. Terms of type ΔL_{xy} denote a bound to the maximum attainable length mismatch between nets x and y</i>	247

Acknowledgements

This work is the result of the efforts and dedication of many people who have contributed in many ways to my endeavor and whom I gratefully acknowledge. My sincere thanks go to my advisor, Prof. Alberto Sangiovanni-Vincentelli who taught me how to conduct serious and rigorous research, in an exceptional environment such as that of the cad group at Berkeley. His vision and outstanding background was a continuous inspiration for finding new and creative solutions to manifold problems. A special sense of appreciation goes to professors Paul Gray, Robert Meyer and Theodore van Duzer for their competence and continuous support during my years at Berkeley.

I would like to thank my wife Tokiko for her continuous and unconditional support, her humor, her kindness, and her love that have made my life meaningful, through happy and difficult phases of my journey in the United States. I thank my mother and my sister for their support to my non easy decision to leave my country and for being at my side during all my studies in this faraway land.

A special thank goes to Denis Baggi and Arokia Nathan, who are mainly responsible for my decision to undertake an academic career abroad and whose advises have proven to be an indispensable tool of survival in such a different environment. I am very grateful to Enrico Malavasi, my mentor and personal friend, for the technical support and a genuinely constructive criticism he brought in much of my work on layout for analog ICs. I would also like to present a special thank to Ranjit Gharpurey, for his competent advice in the research on substrate analysis and Peter Xiao for his work in superconducting modeling.

I gratefully thank Umakanta Choudhury, Albert Ruehli, Steve Seda, Roberto Guerrieri, John Cohn, Gary Holmlund, Bruce Donecker, and the people of The HP-EEsof Labs in Santa Rosa for the useful technical discussions that inspired much of my work.

I am also in debt with the people of my research group, Giorgio Casinovi, Andrea Casotto, Henry Chang, Steven Edwards, Eric Felt, Alper Demir, Gani Jusuf, Desmond Kirkpatrick, Sriram Krishnan, Alan Kramer, Luciano Lavagno, Chris Lennard, Ed Liu, Robert Neff, Tom Shiple, Greg Uehara, and Tiziano Villa, who contributed to my research with useful discussions, suggestions, and the great work that made Berkeley famous.

I will never forget Paolo Giusto, Andre Nieuwland, Christian Olivier, Jaijeet Roychowdhury, Sunil Khatri, Amit Narayan, and Rajeev Murgai, relentless bachelors, who made me enjoy life like no one before. And of course how to forget the Japanese con-

nection, Nagisa Ishiura, Toshi and Keiko Hattori, Masahiko and Kazuyo Takahashi, Jun Kuroiwa, Atsushi Takahara, Key and Rikako Suzuki, the kindest and most sincere people I ever met.

A special acknowledgement goes to my *Italian family* in the United States, namely Paolo Miliozzi and Ana, *Ιασων Βασιλειου*, and Luca Carloni, friends and colleagues with whom I have had the most exciting and animated political discussions in years.

My friends here in the Bay Area and back in Switzerland have constantly followed and supported my endeavors. Among them I gratefully thank Philippe Schönborn, Remedio and Jacqueline, with whom I shared the happiest moments of the last years, Fabrizio and Linda Della Corte, the closest friends I have from Ticino, Vigyan Singhal, the best roommate I ever had, Luigi Semenzato, the best windsurfer in town, Daniel Engels, the Tahitian emigré, Attila Jurecska, the toughest Hungarian on the planet, the Ranjans and the Sanghavis, who desperately but unsuccessfully tried to invite all of us to their traditional weddings in India, and last but not least Slobodan Simic, Sante Gnerre and the Italian crowd, to whom I will always be grateful for the wonderful time spent together.

I would also acknowledge the good work of our secretaries, Flora Oviedo and Kia Cooper, and our grant administrator, Elise Mills, whom I will always remember as the best and most supportive staff I ever had.

In the course of these years, several institutions supported this research, either directly or indirectly. The Swiss National Science Foundation, Asea Brown Boveri and the Lehmann Foundation contributed substantially to the research that led to this dissertation. Their support was indispensable and is greatly appreciated.

Chapter 1

Introduction

*Ed ecco verso noi venir per nave
un vecchio, bianco per antico pelo,
gridando: “Guai a voi, anime prave!*

*Non isperate mai veder lo cielo:
i’ vegno per menarvi a l’altra riva
ne le tenebre etterne, in caldo e ’n gelo.”*

Dante Alighieri, “Inferno”, Canto III

1.1 Computer-Aided Design of Engineering Systems

The main objective of the discipline known as Design Technology is the creation of *methodologies* and *tools* for the design of engineering systems, helping human designers build functionality while satisfying intended performance specifications. Over the past three decades, the development of computer aids for the design of electronic systems (CAD) has been one of the fastest growing areas of activity. In particular, CAD for the physical assembly of electronic systems, either in the form of an integrated circuit (IC) or of a printed circuit board (PCB), has become one of the largest research areas in the field.

Electronic ICs have rapidly evolved from the relatively low complexity of the early days to the high sophistication of today. The task of circuit designers has become increasingly difficult, hence the need for more advanced design supports. In particular, the study of effective methodologies for the physical assembly of high-speed analog and mixed-signal

ICs and of tools supporting it has been a very active topic of research in the past decade. This subject is also the central topic of this dissertation.

1.2 Physical Assembly of Analog and Mixed-Signal ICs

In analog systems signals are continuous functions of time. By contrast, in digital systems each signal is represented by a sequence of finite number of binary digits; therefore, these signals can take on discrete values only. Due to the binary nature of signals, digital circuits are realized using gates with only two states, each state being defined in some range of the continuous signal. This makes digital circuits to a large degree immune to various noise and parasitic sources inherent to ICs. Hence, the design effort can be directed mainly towards trade-offs between power consumption, speed, and area.

Analog circuits in general require more design freedom in order to be applied effectively, since the full spectrum of capabilities exhibited by individual devices is exploited. In most analog circuits individual devices have substantially different sizes and electrical characteristics. These circuits require optimization of various performance measures. As an example, among the performance measures for operational amplifiers are gain, bandwidth, noise, power supply rejection, dynamic range, offset voltage, etc. The importance of each performance measure depends on circuit application. For this reason, fine tuning plays a crucial role in the design of analog circuits.

Because of the rather wide range of parameter spreads in ICs, active and passive components, analog designers developed circuits which cancel out the first order variations in key parameters. However, new dependencies from second order variations of design parameters have become dominant. Typical examples are the matching of input devices in differential pairs, or capacitor matching in switch capacitor filters. Sensitivities to second order variations require much more care especially during the circuit's physical assembly, due to the numerous non-idealities and parasitics it may introduce.

For these reasons designing CAD tools for analog applications is, in general, a difficult task. Consequently, while it is sometimes possible to share CAD tools between the digital and analog portions of a circuit, such as design rule checkers, extractors and data bases, there are many tools that must be designed for use primarily on analog circuits. A general and consistent methodology is required to properly guide the tools towards the satisfaction of all specifications at the system level. In addition, design failures must be

interpreted effectively so as to organize appropriate re-design schemes.

Research on analog CAD systems has progressed at a considerably slower pace than that on their digital counterparts. Part of the reason has been the intrinsic difficulty of defining and controlling performance in analog circuits. High performance can be achieved by taking advantage of the physical characteristics of integrated devices and of the correlation between electrical parameters and their variations due to statistical fluctuations of the manufacturing process. Device matchings, parasitics, thermal and substrate effects must all be taken into account. The nominal values of performance functions are subject to degradation due to a large number of parasitics which are generally difficult to estimate with the proper accuracy before an actual layout is completed.

Another reason might be the present difficulty to identify a level of abstraction where generic models such as the ones developed for digital synthesis can be derived. All these concerns need be addressed in each phase of the design with care, since severe performance degradation, even if localized only in some components, can often jeopardize the functionality of the whole system.

1.3 Bottom-Up vs. Top-Down Approaches

The approach generally adopted by designers consists of building complex layouts bottom-up, starting from the simplest components of the systems and estimating all component specifications using rough approximations mostly derived from experience. This approach often results in a series of time-consuming design loops, hence multiple re-designs are needed for the whole system.

It is our believe that the design loops could be drastically reduced if a top-down approach were used. In a top-down approach the order of the synthesis phases is reversed. First, top-level specifications are rigorously mapped onto constraints on the physical details of the layout, in such a way that the satisfaction of the low-level constraints implies satisfaction of the overall system specifications. Then, the entire physical assembly, partitioned in its basic steps, module generation, placement, routing and compaction, is performed enforcing all physical constraints. A bottom-up verification step based on extraction concludes the assembly.

There are several advantages to this approach. First, a tight control of performance can be maintained in each phase of the physical assembly independently, hence specification

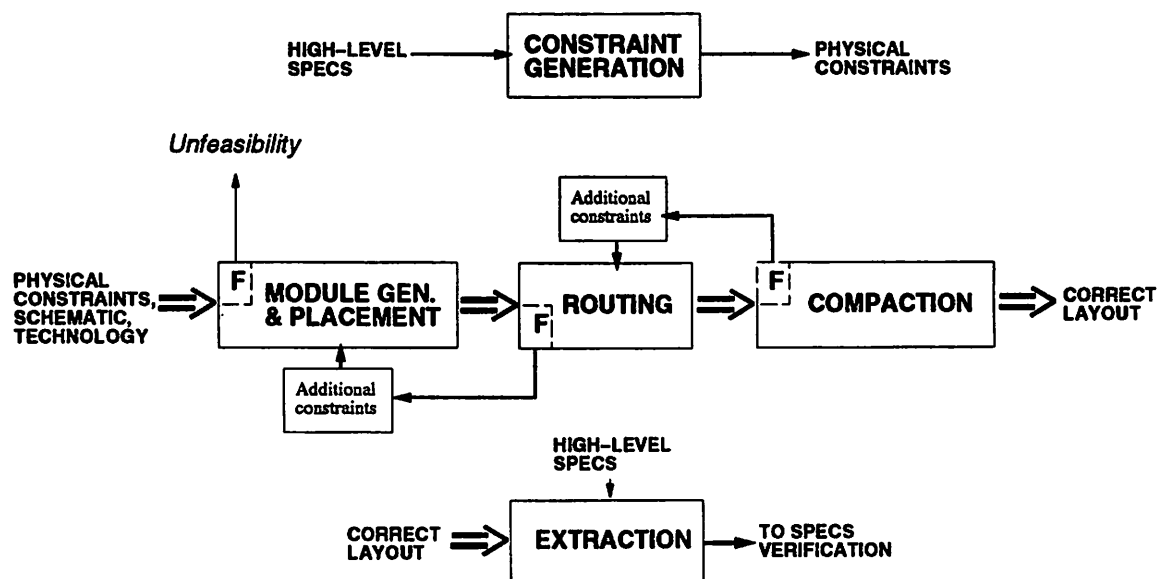


Figure 1.1: *Constraint-driven layout design: Traditional design partition into tasks has been modified by adding information paths between layout phases.*

violations can be identified early, thus enhancing the robustness of the process. Second, all physical constraints are derived so as to minimize the effort that each layout tool requires for its enforcement, hence improving its efficiency. Finally, due to the generality of the constraint generation process, the scheme can be easily extended to encompass a wide variety of non-idealities usually encountered in layout.

The scheme, called *constraint-driven layout design*, was originally formulated for a class of layout problems and then generalized by us to the extent of physical assembly for analog and mixed-signal ICs. This dissertation presents the generalized constraint-driven layout design methodology and the techniques used for each phase of the physical assembly.

1.4 Generalized Constraint-Driven Layout Design

The design flow of the constraint-driven physical assembly system is illustrated in Figure 1.1. First, *high-level specifications* are translated into a set of bounds on *low-level physical constraints*. *A priori* parasitic estimates are used to determine feasible bounds. Among all possible sets of bounds, the one maximizing the *flexibility* of the tool to be used

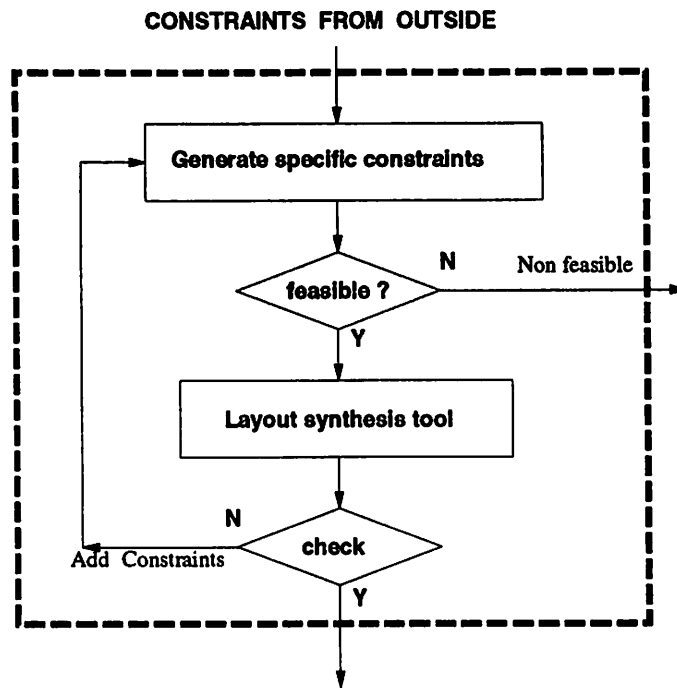


Figure 1.2: The organization of each layout phase. The internal feedback path provides information to the constraint generator. External feedback paths provide information on the reasons of failure to meet performance specifications.

is chosen. Flexibility is a function which measures how easily the tool is able to meet the given set of constraints.

Then at each step, the existence of a feasible configuration is tested, feedback paths are provided to resolve situations of infeasibility. These situations can occur as the result of partially inaccurate estimations on parasitics and other circuit non-idealities at early layout phases due to incomplete information about the physical implementation. In these cases, mechanisms are provided for the re-generation of constraints by correcting *a priori* estimations.

Each layout phase is organized as illustrated in Figure 1.2. The design task is constrained by a set of input specifications, which are either high-level performance specifications or additional design constraints introduced by other layout phases. Constraints are translated into a set of bounds on parasitics by a *constraint generator*, based on estimates of the feasible values of each parasitic. These bounds drive each tool independently. The

resulting layout is then analyzed to check whether performance specifications have actually been met. If some constraints have been violated, the values of the extracted parasitics can provide more accurate estimates to the constraint generator. The constraint generator also executes the feasibility check. In fact, low-level bounds must be feasible, i.e. they must lay between the minimum and maximum possible values estimated for the parameters. Such early detection of infeasibility provides an efficient control of design iterations, thus minimizing overall computation time. Feedback control paths provide previous design phases with information on those critical parasitics for which it was not possible to determine feasible bounds with the current configuration.

1.5 Theoretical Aspects of Constraint-Based Approaches

Any constraint-based approaches to design have the following requirements. First, a performance model must exist in some well defined form. Second, a rigorous method must be used for obtaining a set of constraints on parameters which are controlled during design and/or fabrication. Choudhury [1] used a linear performance model based on sensitivity analysis and formulated the constraint generation process in a form of a constrained quadratic programming approach. The generalization of this approach to encompass a large number of diverse constraints and a wide range of tools was first proposed by us in [2] and will be discussed in this dissertation in full detail.

With this formulation, all performance functions can be represented in a compact and rigorous way, as long as they are continuous and sufficiently regular in an interval around their nominal value. Accurate and efficient calculation of sensitivities is key to the viability of the approach. For this reason a great importance will be given to the accuracy and efficiency issues associated with parasitic estimation and performance evaluation.

We will discuss techniques for the calculation of constraints suitable for all the phases of the physical assembly and for their enforcement at all levels of design. Moreover, constraint-based techniques will be proposed and evaluated for the simplification of layout analysis both at the schematic and at the physical extraction level.

1.6 Organization of the Dissertation

In chapter 2 the state-of-the-art in CAD tools and systems for analog and mixed-signal circuits is surveyed. Chapter 3 presents an overview of sensitivity analysis and of the techniques for the mapping of high-level specifications onto low-level physical constraints. All the details of the constraint generator PARCAR are described in this chapter. In chapters 4, 5 and 6 the physical design tool-set is presented. Chapter 7 deals with the final extraction and verification steps in the design loop, while chapter 8 addresses issues related to the synthesis and analysis of ICs in the presence of substrate-related interference currents. Experimental results on industrial-strength benchmarks are reported in chapter 9, followed by conclusions in chapter 10.

Chapter 2

Literature Survey

*“O tu ch'onori scienza e arte,
questi chi son c'hanno cotanta onranza,
che dal modo de li altri li diparte?”.*

*E quelli a me: “L'onrata nominanza
che di lor suona sù ne la tua vita,
grazia acquista in ciel che sù li avanza”.*

Dante Alighieri, “Inferno”, Canto IV

2.1 The Origins of Computer-Aided-Design

2.1.1 Circuit Simulation

The last three decades have seen a tremendous increase in the complexity and sophistication of electronic systems. Designing to realize functionality while meeting a set of performance specifications, soon required tools capable of overcoming relatively inaccurate and lengthy hand analysis. Not surprisingly the first developed computer aids addressed the problem of circuit simulation and verification.

In the early 1950's digital computers started being actively utilized, for the solution of simultaneous algebraic equations describing linear electrical networks in sinusoidal steady state [3]. However only a decade later, the first viable programs were developed for the simulation of circuits in time-domain. NET1 [4] and SCEPTRE [5] used explicit-integration

and predictor-corrector techniques in the solution of integral-differential equations of non-linear systems. To maintain stability however, very small time steps were needed, hence significantly increasing the time needed to converge to a solution. Only in the mid 1960's, with the introduction of implicit integration schemes, that superior performance could be achieved. In implicit integration, the set of integral-differential equations turns to a set of static algebraic equations for each given time point. The program TRAC [6] implemented these techniques. Almost simultaneously, a second-order implicit-integration scheme was proposed which proved a better performance relative to TRAC. This research led to CIRPAC [7] and to other modifications to the method that included variable order and variable time-step implicit integration routines [8].

In the late 1960's, Howard developed a program that solved numerically a set of simultaneous nodal equations. A simple non-linear device model was used and the equations were linearized at the equilibrium using iterative methods based on Newton-Raphson and excursion limiting techniques [9]. In an independent research effort starting from a theoretical base took place at approximately the same time. Hachtel proposed a new formulation of network equations based on the Sparse-Tableau concept [10]. This approach, allowing the use of efficient techniques for the solution of large systems of linear equations, led to the development of ASTAP [11].

The experience accumulated from these research teams was eventually incorporated in the CANCER [12] and the SLIC [13] projects. With the formalization of Modified Nodal Analysis and the development of sparsity-aware pivoting and matrix reordering techniques, the CANCER project evolved into the SPICE program [14].

2.1.2 Digital Timing Analysis and Event-Driven Simulation

Following the enormous success of SPICE and the increasing importance of electrical simulation in circuit design, research in the field developed in two main directions: large-scale simulation and optimization. Early techniques, reviewed in [15], gave way to approaches purposely relaxing accuracy to achieve greatly improved simulation speed [16]. These methods, conceived for digital timing analysis, soon showed limitations in accurately simulating the effects of feedback. It was the study of numerical limitations in timing analysis that led to new techniques based on relaxation in both space and time domain. The main advantage of relaxation-based approaches is the ability of exploiting time sparsity, us-

ing the event-driven selective trace techniques first developed in digital simulators. Shortly after the development of timing simulation, mixed-mode or hybrid event-driven simulators emerged, resulting in extensive research in the field [17, 18, 19]. For a review in the field see e.g. [20, 21, 22].

More recently, this work has evolved into the development of techniques to reduce large lumped RCL circuits into a small, more tractable modal approximation of its transfer function. Thus a significantly higher efficiency can be achieved in simulating the network. A good example of this trend is represented by the Asymptotic Wave Evaluation method (AWE) [23] developed in the late 1980's, which has proven a valuable tool mainly in analysis and verification tasks.

2.1.3 Circuit Optimization

Automated design optimization [24] evolved in parallel to circuit simulation. In fact, the idea of using optimization to help design electrical circuits dates back to the early 1950's. DC biasing effects and frequency-domain matching were among the first considerations to be integrated in the optimization process [25, 26]. Catalyzed by breakthroughs in simulation techniques and a new formalized representation of circuit optimization as a general non-linear programming problem, significant effort was devoted to the improvement of optimizers' efficiency. A significant step towards achieving the goal is represented by drastic efficiency improvements in the calculation of network sensitivities, necessary for the most useful optimization algorithms. This work led to the development of several tools. In the A2OPT project [27] the simulator ASTAP [11] was used in combination with a minimizer based on the rank-one update method [28]. Constraints were accounted for in the optimization by introducing an additional penalty to the objective function. A second optimization system based on ASTAP called APLSTAP [29] was built as an interactive CAD consultant tool. A Linear Programming step was used to quantify the best trade-offs between multiple objective and constraint functions for optimally guiding the design process.

The above approaches had several disadvantages. These included a lack of flexibility at the formulation and implementation level, the relatively low degree of interactivity and serious deficiencies at the simulation level. A successful attempt to alleviate these problems was made in DELIGHT.SPICE. The tool resulted from the merger of the optimizer DELIGHT [30] and SPICE, in which an efficient sensitivity analysis package had been in-

corporated [31]. Other tools followed on the same track, where more attention was given to user-interface and flexibility issues [32]. Despite their success, numerical circuit optimization tools became soon inadequate due to the explosion in complexity of analog and mixed-signal circuits. In addition, researchers realized the enormous influence of physical implementation on performance [33, 34, 35, 36, 37], hence the necessity of optimizing at the schematic design and layout synthesis levels simultaneously.

2.2 Early Work in Computer-Aided-Design for Analog ICs

Due to the challenge posed by the problem, several research teams around the world actively began working on the creation of integrated design systems that would attack the problem of analog design in a systematic fashion. Three main schools of thought emerged to approach the problem.

2.2.1 Silicon Compilation

Introduced in the early 1980's, dedicated *silicon compilers* were advocated for the design and physical assembly of relatively complex yet highly specialized applications. The AIDE2 system [38] is one of the early examples of the trend. In AIDE2 circuit topologies, described using the C language, were mapped onto a fixed-floorplan layout based on a library of subcircuits. Libraries or library generators were provided as a complement to the compilation system [39]. More recently, AIDE2 evolved to take into account higher order effects and parasitics during the compilation process. The approach proposed in [40] was aimed at minimizing all parasitic effects by making use of linearized models of performance based on sensitivity. The fixed-floorplan layout synthesis style was replaced with a depth-first-search based *topological sort* [41] operating on clusters of "sensitive" components, i.e. devices connected to sensitive nets. Numerically computed sensitivities were used to derive a priority schedule for a digital channel router [42].

2.2.2 Knowledge-Based Techniques

PROSAIC [43], the precursor of most knowledge-based systems, was developed in the early 1980's. The approach was originally derived from the work on declarative circuit modeling at MIT [44] and later [45]. The idea consisted of creating a large data-base of

rules to be used by an inference engine driving a sequence of decisions which determined a course of actions during the design. A number of design systems using similar rule-based approaches appeared later in the literature [46, 47]. For example, BLADES [48] was based on a conventional expert system consisting of a dedicated knowledge base and an inference engine. A numerical *consultant*, generally a simulator, was used for verification purposes.

There are several disadvantages associated with rule-based systems. The creation of the knowledge base or of the rule set is generally a relatively complex process requiring the expertise of highly experienced designers. Knowledge bases are very specific to a technology and even to a very small class of problems, hence re-design and library synthesis turn-around is often quite time consuming. In addition, bounds on the quality of the design process are generally not available, so that the result obtained by these systems can vary considerably across different specifications.

2.2.3 Algorithmic Methodologies

Algorithmic design methodologies first appeared in the mid 1980's for the layout synthesis of analog and high-speed digital circuits and soon migrated to schematic design automation and optimization. LTX2 [49, 50] is the first example in this direction. In LTX2 the physical assembly problem was partitioned into placement, floorplanning, global and detailed routing, according to a classical scheme derived from the digital world [51, 52]. A 2-D placement tool, based on a modification of the Kernighan and Lin algorithm for graph partitioning [53], was used to create divided clusters of analog and digital cells [54]. During placement, separation between sensitive signal nets and large swing analog and digital signals was guaranteed by alternating sensitive and insensitive routing channels in the standard cell floorplan. Detailed routing used shielding to minimize the coupling between sensitive nets residing in the same channel [55]. This technique was successfully applied to relatively simple circuits acting as an interface with digital cores.

2.3 The First Complete Design Systems

2.3.1 IDAC/ILAC

After the initial phase, experimentation gave way to increasingly complex and more flexible systems, designed for larger mixed-signal circuits and a number of technolo-

gies. The IDAC system [56] proposed a number of innovations later to be used by other systems. Among the most notable ones, a systematic architecture selection mechanism using simplified equation-based circuit analysis and a set of predefined *synthesis strategies*, a relatively large library of circuit topologies and a layout synthesizer, ILAC [57]. ILAC's main novelty was the classification of each net based on its criticality. Parasitics on sensitive nets and coupling between noisy nodes were minimized during the routing phase. A procedural layout block generator allowed the enforcement of limited geometric constraints, such as symmetry and matching between devices. The detailed routing step, based on a gridless scan-line incremental channel router [51, Chp. 4], was semi-interactive, allowing controlled rip-up options but no spacing. The other layout phases, a slicing-tree floorplanning [58], and a best-first maze algorithm for global routing [51, Chp. 3], reflected a digital-like methodology.

2.3.2 OPASYN

A main limitation of the IDAC/ILAC system was a lack of flexibility of the design process due to the relative simplicity of the models used for circuit characterization and of the parasitic approximations used during the layout synthesis. In OPASYN [59] similar analytic models were used, however refinement were made to take into account second-order effects and parasitics that the physical implementation could introduce. The system assumed a synthesis-by-analysis approach. Optimization was based strictly on analytical models rather than simulation as in [31, 60]. In OPASYN layout was generated from a fixed-floorplan arrangement, capturing a set of important considerations in the design of analog circuits. Routing was performed disregarding any analog constraint, using the digital tool, MIGHTY [42]. The approach was strictly non-hierarchical with a number of non-interchangeable circuit topologies. The obvious disadvantage was given by the lack of flexibility within the design and the schematic optimization. A similar synthesis strategy was proposed in OAC [61], where full performance optimization was carried out during design and physical assembly. The system used fixed topology opamps on which it performed non-linear optimization to roughly size all devices. Then, detailed design was carried out to precisely take into account every parasitic component associated with the layout.

2.3.3 OASYS/ACACIA

Improved parasitic analysis techniques guaranteed a better estimation of circuit performance after fabrication. The OASYS design optimization system [62] and the layout synthesis environment ACACIA [63] were built in the late 1980's to utilize these techniques systematically for more diverse and complex circuits. The original concept of OASYS was similar to that used in IDAC, except for the fact that hierarchical decomposition was used during the design as a way of reducing a large, inherently complex optimization problem into a number of simpler ones.

Hierarchical decomposition had been proposed before for digital design [52] and, independently, for analog design such as in AN_COM [64], however design adjustments were not handled systematically. Hierarchical components were regarded as *templates-connected sub-blocks* and top-level specifications were recursively transformed during synthesis until the leaves of the design were reached and the individual sub-block specifications were generated and imposed to the automated layout generator. Some degree of flexibility was allowed in the topology of each block, hence equation-based models for each block were used to operate *backtracking* on the hierarchy, for diagnosing design failures and propose reparative strategies. Nonetheless, the rule-based nature of the system limited the exploration of a large set of feasible designs, resulting in a locally but not globally optimized circuit. Another major limitation of the system was the rather weak link between design and physical assembly, where mainly digital-oriented techniques were used in all phases of the layout synthesis.

More recently, OASYS has evolved onto the ASTRX/OBLX system [65], where the rule-based decision process has been replaced by a purely numerical optimization approach similar to that of DELIGHT.SPICE except for the use of Simulated Annealing [66] as the exclusive optimization engine. The main novelties of ASTRX/OBLX are the relaxation of the requirement that the circuit be feasible, i.e. that Kirchoff's laws be satisfied, at each annealing step and the use of AWE, in combination with symbolic analysis to quickly evaluate circuit performance.

The ACACIA environment also evolved from the digital-like layout system ANAGRAM into the KOAN/ANAGRAM II place and route system [63, 67]. KOAN, a Simulated Annealing based placement tool, could perform device-shaping and abutment on MOS transistors dynamically during the annealing. The enforcement of analog topological constraints

such as device symmetry and matching was integrated in the algorithm's cost function. ANAGRAM II a detailed line-expansion router [51, Chp. 3], supported symmetric differential routing, cross-talk avoidance and over-the-device routing. Contrarily to other approaches [68, 69], KOAN/ANAGRAM II did not use compaction as a way of further area reduction and/or performance adjustment or re-design.

Recent developments within ACACIA include RAIL [70], a power/ground synthesizer and WREN [71], a global/detailed router for signal paths. RAIL is a router based on progressive width perturbation in power busses defined on a regular grid and subject to the effects of switching noise present in the substrate and in the supply. The perturbation scheme is guided by a Simulated Annealing based algorithm. WREN decomposes each net into a minimum spanning tree and a number of alternative two-point paths is found for each net. The alternatives are re-shuffled during the annealing until a configuration is found which meets set of user-defined cross-coupling constraints. Detailed routing is carried out in a similar fashion as in [72], where a constraint-graph is modified to take into consideration various adjacency constraints reflecting noise immunity specifications. Although a number of algorithms were proposed for the minimization of passive parasitics, performance specifications were never explicitly enforced in the tool and the designer remained a key player in guiding the synthesis by determining the criticality of interconnects.

2.4 Evolution of Approaches

2.4.1 Silicon Compilation and Module Generation

Due to the dramatic increase of complexity of analog circuits of the early 1990's and the emergence of new mixed-signal circuits, silicon compilation was still regarded as an effective and powerful tool for schematic design and physical assembly. During this time the original tools migrated towards new domains of application [73, 74, 75, 76, 77, 78, 79]. New systems based on a standard cell approach, e.g. [80], had been refined to support large and possibly mixed-signal designs. At the same time, techniques for the routing of analog components in the presence of digital signals [50, 81, 82, 83], in combination with a traditional semi-fixed floorplan paradigm allowed the creation of compilers where some low-level parasitic issues were addressed. In CONCORDE [84], a compiler for successive approximation A/D converters using a set of high-performance pre-designed analog circuits

was proposed. In MxSICO [85], a 2nd order $\Sigma - \Delta$ modulator compiler and modified vertical/horizontal constraint graphs were used for cross-over balancing in sensitive nets during the routing of channels. SCF [86] proposed a more modular approach with integrated module generation and physical assembly. The suggested approach clearly goes towards a more general and flexible design system.

CADICS [87], a compiler for cyclic A/D converters, introduced the need to support the synthesis process with a behavioral model and a set of performance-driven layout tools for the generation of circuit components, as well as floorplanning and detailed routing. Thanks to behavioral modeling and simulation, the DAC performance could be quickly estimated at each stage of the optimization, thus ensuring a much broader and systematic exploration of the design space. One level of hierarchy was employed and a number of critical non-idealities were considered during the top-down synthesis. Careful parasitic extraction during the bottom-up verification phase, provided an accurate and reliable verification methodology.

The CATALYST design system for switch capacitor data converters [88] was essentially an extension to dedicated silicon compilation with the incorporation of architecture selection mechanisms based on figure-of-merit. Hierarchical system partitioning and macro-models were used for figure-of-merit calculations as well as bottom-up performance evaluations during the verification phase.

Today, silicon compilation for analog and mixed-signal applications occupies an important niche in the vast panorama of design systems. It has been shown to be well suited for specific applications, and in specific cases it could be even preferable to more general approaches. A number of surveys have appeared on the subject of compilation and module generation for specific circuits and in particular for data converters, see e.g. [89, 90, 91].

2.4.2 Knowledge-Based Systems

Due to the increasing success of algorithmic-based tools and the superior performance of module generators in dedicated applications, knowledge-based systems gradually became the environment for a set of algorithmic tools or compilers. SALIM [92, 93] for example, was a rule-based design system governing a set of layout algorithms, some of which derived from the digital domain. A PROLOG-like HDL was used to represent the design problem in a procedural fashion and through inference rules. The language was the basic

glue between specifications and layout synthesis algorithms. The generation process was bottom-up starting from transistor schematic, through grouping of analog functions into library blocks until the complete circuit was generated.

The design environment STAIC [94] mapped structural and performance specifications onto a layout description language, SPICE netlists and a data sheet. A number of intermediate and complementary descriptions was used to guide the user and the optimization tools throughout the design path. The final code, compiled by ICEWATER [95], was executed to generate the complete layout. The design methodology proposed in STAIC made extensive use of hierarchy, analytical model generation and successive refinements at each stage of the synthesis. Both knowledge-based and numerical methods were used for topology selection and semi-automated design of simple circuits. The layout generation was guided by coded rules and by pre-defined floorplan as in OPASYN .

The knowledge-based design system LAMP proposed in [96] and [97] was also based on an expert system that operated directly on the circuit primitives. The primitives were extracted from an initial schematic by means of a rule-based scheme. An iterative equation-based routine improved the circuit performance by operating a series of substitutions in the circuit topology guided by the expert system and/or by human interaction. The layout synthesis system SLAM [98, 99] used knowledge of the primitives, in combination with qualitative sensitivity analysis, to create a priority schedule for floorplanning and routing. The slicing structure-based floorplanning algorithm made use of sensitivity information to define highly sensitive zones where devices should be placed near. The channel router used again a priority schedule for an ordered generation of nets, starting from critical ones.

2.4.3 Hybrid and Human-Driven Systems

To cope with increasingly sophisticated circuits, alternative hybrid systems involving a rule-based approach to design and silicon compilation for the physical assembly have appeared. In C5 [100] for example all phases of the synthesis process were functions in a C-like HDL, while ALSYN [101] enforced additional user-determined analog-specific rules incorporated directly into the object-oriented circuit data-base. In SEAS [102] a *seed circuit* was used for initializing a Simulated Evolution engine, which generated a set of feasible variants or *mutations* to the seed. The algorithm terminated when the *score* associated with the current circuit could not be further increased. In [103] the knowledge base was present

in a form of a circuit example used as a starting point for an improvement-based synthesis. In these systems floorplan and placement were generally performed using modified versions of the Min-cut algorithm for slicing structures [104, 58]. The final layout was obtained through compaction-free maze routing [51, Chp. 3] or routing-free symbolic compaction.

In the late 1980's extensive experimentation in semi-automated analog layout systems lead to increasing the human presence in the design loop. In LADIES [105, 106] for example, a knowledge base combined with an algorithmic approach to the problem of analog synthesis was proposed. The system used a number of techniques directly imported from design automation of high-performance digital circuits [107]. A set of simple rules were used for the analysis of the schematic and for the generation of constraints on layout geometries. Matching, design rules and critical coupling were generated in this way. The layout was generated by maintaining a physical topology equivalent to that of the schematic itself. Algorithmic optimization tools for placement [108], global and detailed routing were used to generate the initial layout, which was subsequently improved using another set of rules designed to enforce the original constraints while minimizing area and wirelength. Later implementations of a similar methodology, such as ALE [109] improved the refinement phase and gradually increased the importance of algorithmic operations in the system, thus obtaining more compact layouts and significantly higher flexibility in the creation and enforcement of analog-specific constraints.

The CHIPAUDE system [110, 111] on the contrary, proposed a top-down methodology based on hierarchical decomposition and qualitative reasoning at the schematic level. The physical assembly was performed mainly using *ad hoc* generators. The synthesis environment ISAID [112] propagated specifications throughout the design hierarchy using rough parasitic estimates along the way. At early stages of the design macro-modeling [113] was used to allow specification-driven architectural selection, based on a ranking system similar to that of IDAC [56]. At later design stages models were used mainly to speedup the synthesis process. The synthesis was followed by a improvement phase based on the principles of qualitative reasoning applied to MOS design [114]. The RACHANA package was responsible for the layout generation process [115]. Using a rule-based algorithm, instances of primitives were automatically recognized from the schematic and realized using a parametrized module generator. A conventional floorplan algorithm was followed by iterative place-and-route procedures. Placed modules followed the topological order of the schematic design as in [106]. Routing was performed by an area router minimizing inner-resistance, number of

bends and capacitive parasitics.

Despite satisfactory results obtained in recent years, knowledge-based systems still lack of the necessary flexibility needed for technology migration and re-design in today's complex circuit.

2.5 Constraint-Based Approaches

2.5.1 Foundations

The constraint-based approach to design, due to Choudhury in 1990, was originated from the research on parasitic-aware channel routing [72]. A typical constraint-driven approach to layout consists of two phases. First, performance specifications are mapped onto bounds on all physical parasitics relevant to the implementation. Then, each bound is enforced during the physical assembly, hence guaranteeing the satisfaction of the original specifications. The bound generation is a complex process requiring some type of performance modeling and an optimization phase. The dependence of performance from parasitics is generally evaluated using sensitivity analysis, while the actual bound generation is performed by constrained-optimization [1, 116, 117].

2.5.2 First Constraint-Driven Design Tools

The approach in its original formulation was used to determine the weights of the edges of a constraint-graph [51, Chp. 4] representing a channel with critical nets. The original approach soon migrated to maze routing [118], placement [119], and compaction [69] tools, all integrated in the OCTTOOLS-VEM environment [120, 121]. A similar sensitivity-based constraint generation scheme was proposed by Gad El Karim [122, 123] and applied to the placement problem [124]. Schematic design followed later with the introduction of behavioral models for high-level performance characterization [125, 126].

2.5.3 Later Implementations

Using similar constraint-based approaches, others have proposed to solve specific problems in physical and schematic design. In STAT [127, 128] for example, a semi-automated layout synthesis approach with enforcement of geometric analog-specific constraints is presented. Symmetry and matching constraints are annotated directly on the

schematic as “related_to” properties. A graph, derived from schematic and based on these relations, is the starting point for the placement algorithm, which is based on a conventional *topological sort* [41]. A maze router [51, Chp. 3] has been modified to control wiring resistance and to prevent electro-migration. Technology-independent parametrized module generation completes the layout system.

The methodology proposed in ARIADNE [129], originally developed for analog circuits, has been recently employed in mixed-signal systems. The design engine of ARIADNE is a top-down hierarchical approach similar to the top-down, constraint-driven design system proposed in [126], with a comprehensive user interface in the front-end. Symbolic analysis [130, 131], is used to model circuit components, to perform architectural selections and to size devices in a similar fashion as in IDAC [56] and [59]. New topologies can be quickly characterized and incorporated in the data-base, allowing considerable design flexibility. As a byproduct of the optimization, constraints on layout geometries and parasitics can be obtained.

In LIBRA [132, 133, 134] constrained optimization and sensitivity analysis are combined to obtain compact layouts while enforcing a small set of performance specifications. Models for worst-case performance degradation due to technology deviations and resistive parasitics have been derived. Specification violations are evaluated and their elimination is attempted at each stage of the layout by building appropriate cost functions.

Chapter 3

Generalized Constraint Generation

*Ora cen porta l'un de' duri margini;
e 'l fummo del ruscel di sopra aduggia,
sì che dal foco salva l'acqua e li argini.*

*Quali Fiamminghi tra Guizzante e Bruggia,
temendo 'l fiotto che 'nver lor s'avventa,
fanno lo schermo perché 'l mar si fuggia;
e quali Padoan lungo la Brenta,
per difender lor ville e lor castelli,
anzi che Carentana il caldo senta:*

*a tale imagine eran fatti quelli,
tutto che né sì alti né sì grossi,
qual che si fosse, lo maestro felli.*

Dante Alighieri, "Inferno", Canto XV

In a constraint-based design system, performance specifications are enforced by translating them onto a format that can be handled directly by the tools responsible for the design. The process of format translation is known as *constraint generation problem* and it

is the focus of this chapter. The set of techniques described throughout the chapter have been implemented in the tools PARCAR, SENSALC, and MKSYM all part of the OCTTOOLS layout tool-set.

3.1 Problem Formulation

For a given circuit C , let us define *performance* \mathbf{K} as the finite array of all measures that evaluate a parametric behavior for C and N_K its size. Performance specifications are expressed as the maximum allowed performance degradation from nominal, due to process variance and parasitics caused by the realization of the layout details. Both absolute parasitic values and mismatch play a role in the deviation of performance measures from nominal.

Let us define V as the finite set of all possible operating points for C . Assume that \mathbf{K} can vary around an operating point $v \in V$ and let \mathbf{K}_v be the performance at v , denominated *nominal performance*. Assume that all parasitics significantly affecting performance are known. Let us define $\Delta\mathbf{K} = \mathbf{K} - \mathbf{K}_v$ as the degradation of performance \mathbf{K} from its nominal due to all known parasitics. A specification on \mathbf{K} is defined to be a constraint on the maximum allowed performance degradation $\Delta\mathbf{K}$

$$\Delta\mathbf{K} \leq \overline{\Delta\mathbf{K}} \quad (3.1)$$

The constraint generation problem consists of finding bounds on a subset of parasitics, whose enforcement guarantees the satisfaction of all performance specifications.

Problem 1 *Given a circuit C with performance \mathbf{K} and a finite set of parasitic components, find bounds on all parasitics, such that (3.1) holds.*

The solution of this problem is nontrivial for two reasons. First, performance \mathbf{K} is generally an array of non-linear functions of parasitics, often not representable in a compact form. In addition, the number of parasitics is generally much larger than the size of the performance array. Hence a naive approach based on solving inequality (3.1) with respect to each parasitic is not feasible. In the remainder of this chapter techniques are presented that address this problem in an efficient and rigorous fashion.

3.2 Mapping Specifications onto Layout Constraints

Our approach to the constraint generation problem is based on the work described in [1, 116] and [2]. Let us assume that performance \mathbf{K} is continuously differentiable around operating point v . Then, performance degradation $\Delta\mathbf{K}$ can be represented in terms of its sensitivity with respect to all relevant parasitics. We denote the number of layout parasitics by N_p , the array of all such parasitics by $\mathbf{p} = [p_1 \dots p_{N_p}]^T$, and the array of their *nominal* values by $\mathbf{p}^{(0)} = [p_1^{(0)} \dots p_{N_p}^{(0)}]^T$. Each performance K_i is a non-linear continuously differentiable function of all parasitics $K_i = K_i(\mathbf{p})$ and the array of the N_k performance functions will be indicated as $\mathbf{K} = \mathbf{K}(\mathbf{p}) = [K_1(\mathbf{p}) \dots K_{N_k}(\mathbf{p})]^T$. If all parasitics are subject to variations with respect to their nominal values, let $\Delta\mathbf{K}(\mathbf{p}) = \mathbf{K}(\mathbf{p}) - \mathbf{K}(\mathbf{p}^{(0)})$ be the corresponding degradation of \mathbf{K} due to such variations.

A generalized expression for the computation of sensitivities from a set of arbitrary performance functions has been derived in [31, 135]. With this formulation, all performance functions can be represented in a compact and rigorous way, as long as they are continuous and sufficiently regular in an interval around their nominal value, The sensitivity of K_i with respect to p_j is defined as¹

$$S_{i,j} = \left. \frac{\partial K_i(\mathbf{p})}{\partial p_j} \right|_{\mathbf{p}^{(0)}}. \quad (3.2)$$

The matrix of all sensitivities is

$$\mathbf{S} = \begin{bmatrix} S_{1,1} & \dots & S_{1,N_p} \\ \dots & \dots & \dots \\ S_{N_k,1} & \dots & S_{N_k,N_p} \end{bmatrix}.$$

Sensitivities are computed for each performance function, with respect to each parameter that may be introduced or modified in the layout phase, i.e. parasitics and geometric parameters. Several techniques have been developed for efficient numerical calculation of sensitivities in time and frequency domain [136, 137, 31]. Performance degradations are approximated by linearized expressions using sensitivities [1]. These approximations are acceptable if degradations are small compared to the nominal values. The array of all degradations of performance functions due to parasitic variations is

$$\Delta\mathbf{K}(\mathbf{p}) \approx \mathbf{S} [\mathbf{p} - \mathbf{p}^{(0)}] \quad (3.3)$$

¹Here and in what follows the non-normalized notation, first used in [116], is used for sensitivities, without loss of generality.

Before the definition of layout details, one cannot take advantage of the possible cancellation effects due to positive and negative sensitivities for different parasitics. Hence, each performance constraint is modeled only with respect to the parasitics whose sensitivity is either positive or negative, depending on the sign of the constraint itself. Assuming that the performance model of (3.3) is used, inequality (3.1) becomes

$$\Delta \mathbf{K}(\mathbf{p}) - \overline{\Delta \mathbf{K}^+} \leq \mathbf{0} \quad (3.4)$$

$$\Delta \mathbf{K}(\mathbf{p}) + \overline{\Delta \mathbf{K}^-} \geq \mathbf{0} \quad (3.5)$$

where $\overline{\Delta \mathbf{K}^+}$ and $\overline{\Delta \mathbf{K}^-}$ are the vectors of constraints, in absolute value, on the degradation of performance functions $\mathbf{K}(\mathbf{p})$ in the positive and negative direction respectively. They can be different and one of them can eventually be infinite. By substituting the linearized expression (3.3) in inequalities (3.4) and (3.5), the general problem can be rewritten as

$$\mathbf{S}^+ [\mathbf{p} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}^+} \leq \mathbf{0} \quad (3.6)$$

$$\mathbf{S}^- [\mathbf{p} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}^-} \leq \mathbf{0} \quad (3.7)$$

where \mathbf{S}^+ is the matrix of the worst-case positive sensitivities and \mathbf{S}^- is the matrix of the absolute values of the worst-case negative sensitivities:

$$\mathbf{S}^+_{i,j} = \max(0, S_{i,j})$$

$$\mathbf{S}^-_{i,j} = \max(0, -S_{i,j})$$

In the remainder of this dissertation the '+' and '-' signs have been omitted in the notations of sensitivities and constraints. Expressions (3.6) and (3.7) are given for positive and negative directions, and the general problem formulation becomes

$$\mathbf{S} [\mathbf{p} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}} \leq \mathbf{0}. \quad (3.8)$$

We want to determine an array of *bounds* $\mathbf{p}^{(\text{bound})} = [p_1^{(\text{bound})} \dots p_{N_p}^{(\text{bound})}]^T$ for all parasitics, such that inequality (3.8) holds as long as each parasitic remains below its bound, i.e.

$$\mathbf{S} [\mathbf{p}^{(\text{bound})} - \mathbf{p}^{(0)}] - \overline{\Delta \mathbf{K}} = \mathbf{0}. \quad (3.9)$$

It is necessary that all bounds be *feasible* and *meaningful*, i.e. all layout structures associated with the bounds must be physically realizable. Let $p_j^{(\text{min})}$ and $p_j^{(\text{max})}$ be respectively the minimum and maximum possible values which can be assumed by parasitic p_j , and let

$\mathbf{p}^{(\min)} = [p_1^{(\min)} \dots p_{N_p}^{(\min)}]^T$ and $\mathbf{p}^{(\max)} = [p_1^{(\max)} \dots p_{N_p}^{(\max)}]^T$. The array of bounds $\mathbf{p}^{(\text{bound})}$ must satisfy the following inequalities:

$$\begin{cases} \mathbf{p}^{(\text{bound})} - \mathbf{p}^{(\min)} \geq \mathbf{0} \\ \mathbf{p}^{(\text{bound})} - \mathbf{p}^{(\max)} \leq \mathbf{0} \end{cases} \quad (3.10)$$

The constraint generation can be reformulated as the problem of finding a solution to equation (3.9), subject to (3.10).

3.3 Methods for the Evaluation of Sensitivities

In general, the constraint generation problem requires sensitivities to be computed for a general performance K_i with respect to a set of parasitics p_j or a set of design parameters π_j . Suppose K_i is an explicit differentiable function of p_j or π_j , or it can be modeled as such. Then, the sensitivity of K_i , defined in equation (3.2), can be derived using a number of techniques, both numerical, analytical or a combination of the two [30, 138, 137, 139].

If on the contrary, K_i is strongly non-linear, two strategies can be adopted. The first consists of approximating performance K_i as a Taylor series, thus making the derivation of constraints a complex task. The second option consists of calculating *worst-case sensitivities* for K_i .

To illustrate the method, consider the array $\mathbf{\Pi}$ of all parameters π_j and $K_i = K_i(\mathbf{\Pi})$. Let us split vector $\mathbf{\Pi}$ in subvectors $\mathbf{\Pi}'$ and $\mathbf{\Pi}''$. The two vectors include the parasitics that show a linear and a non-linear behavior, respectively. $\mathbf{\Pi}'$ is defined as the vector of all parameters such that

$$|(\mathbf{S}_{i,\mathbf{\Pi}'})^T \Delta \mathbf{\Pi}' - \Delta K_i| < \epsilon, \quad (3.11)$$

with $0 < \Delta \mathbf{\Pi}' < \delta$, for some $\epsilon, \delta > 0$.

The problem of finding a worst-case sensitivity $\bar{\mathbf{S}}_{i,\mathbf{\Pi}'}$ is equivalent to that of solving

<i>maximize</i> : $\mathbf{S}_{i,\mathbf{\Pi}'}$	(3.12)
subject to:	
$\mathbf{\Pi}'' \in I,$	

where I is the feasibility interval of $\mathbf{\Pi}''$.

3.4 Constraint Generation Engine

3.4.1 Absolute Parasitic Constraints

In general, an infinite number of solutions exist for the constraint generation problem. PARCAR [116] is a constraint generator, namely a tool able to find a solution to the constraint generation problem under particular assumptions. Among all solutions, PARCAR chooses the one maximizing the layout tool *flexibility*, which is a measure of how easily the tool is able to meet the constraints. To illustrate this concept, consider the following example. Suppose that the bound for a given parasitic p_j is close to its lower limit $p_j^{(min)}$, then the implementation of a layout geometry associated with p_j will be arduous and a small number of solutions for its realization will be available. This might result into a significant loss of flexibility due to the limitations that will be necessarily imposed onto the remaining layout to be implemented. If, on the contrary, the bound is close to $p_j^{(max)}$, the effort required is lower, and the constraint easier to meet. Therefore a flexibility function defined as

$$F = 1 - \frac{\|\mathbf{p}^{(max)} - \mathbf{p}^{(bound)}\|_2}{\|\mathbf{p}^{(max)} - \mathbf{p}^{(min)}\|_2}.$$

reflects the advantage of choosing a certain set of bounds. A discussion of this definition and of the quadratic norm choice can be found in [116]. In PARCAR a geometric norm is used and the constraint generation problem is solved by minimizing a quadratic function (the geometric norm) subject to linear constraints (3.9) and (3.10), using a standard quadratic programming (QP) package.

The quality of the result depends on the estimates of parasitic limits $\mathbf{p}^{(min)}$ and $\mathbf{p}^{(max)}$, which become more and more accurate as layout details are defined during the design. The values of $\mathbf{p}^{(min)}$ and $\mathbf{p}^{(max)}$ are generally not known *a priori*. However, it is possible to compute suitable estimates, depending on the layout algorithm used. For example, the minimum value of the cross-coupling capacitance between unrouted nets can be set either to zero, or to the crossover capacitance due to unavoidable crossings. The latter estimate, however, is possible only if the router is able to detect unavoidable net crossings. This is the case for a channel router, where wire paths have been predefined in the global routing phase. With maze routing, on the contrary, the minimum value is always set to zero.

A substantial speed-up of the QP solver is achieved by removing from the problem

those parasitics whose cumulative contribution to performance degradation is negligible. A threshold value $\alpha < 1$ is defined (in PARCAR we set $\alpha = 0.01$). For each performance function K_i , all parasitic effects on performance are sorted by increasing value. The first n_i parasitics in the sorted list which satisfy

$$\sum_{j=1}^{n_i} S_{i,j} p_j^{(max)} \leq \alpha \overline{\Delta K}_i, \quad (3.13)$$

are considered non-critical with respect to the threshold α . To compensate for this simplification in the constraint generation problem, equation (3.9) is modified by replacing $\overline{\Delta K}$ with $(1 - \alpha) \overline{\Delta K}$. Notice that the sorting order may be different for different performance functions. Let P_i denote the set of n_i non-critical parasitics sorted according to performance K_i . When all performance functions are considered simultaneously, the set P of all non-critical parasitics is

$$P = \bigcap_{i=1}^{N_k} P_i.$$

The set P , determined in this way, is eliminated from further analysis. Different sorted lists are maintained for each kind of parasitics and elimination is carried out separately. This simplification can be very effective, since in most cases it allows to eliminate a relevant number, typically the 80-90% of all parasitics.

3.4.2 Constraints on Mismatch

The importance of *device matching* in integrated circuits has been shown not only for active but also for passive elements [140, 141]. Designers generally impose matching constraints on circuit devices to ensure that voltage and current mismatches in these devices be bounded. These constraints can then be mapped directly onto constraints on the geometry of the physical implementations [141]. For this reason device matching constraints are usually of qualitative nature, mostly dictated by the expertise of the designer.

Matching enforcement of device parameters or interconnect parasitics is often referred to as the maximization in correlation of electrical parameters associated with given circuit components. In general the task is accomplished by minimizing the physical distance between the components or by means of topologies specifically designed to overcome the effect of technological gradients and random mask errors.

With tighter specifications and more complex circuits however the concept of matching as the solution of a maximization problem becomes non-trivial since many, possi-

bly conflicting matching specifications might be required. This problem was first addressed by us in [2], where the rather imprecise definition of matching was replaced with a rigorous one, which allows advantageous trade-offs to be implemented. The approach consists of two phases: sensitivity characterization and constrained optimization.

Manipulating Sensitivities

Consider two parasitics p_1 and p_2 . Within the limits of linear approximation (3.3), their contribution to the degradation of performance, K_i is

$$\Delta K_i|_{1,2} = S_{i,1}p_1 + S_{i,2}p_2 = 2S_{i,p}p + \frac{S_{i,\Delta}}{2} \Delta p \quad (3.14)$$

where

$$\begin{aligned} p &= \frac{p_1+p_2}{2} & S_{i,p} &= S_{i,1} + S_{i,2} \\ \Delta p &= p_1 - p_2 & S_{i,\Delta} &= \frac{S_{i,1}-S_{i,2}}{2} \end{aligned} \quad (3.15)$$

It is evident that if

$$\left| \frac{S_{i,\Delta}}{S_{i,p}} \right| \gg 1 \quad (3.16)$$

the contribution of p_1 and p_2 to the degradation of K_i can be significantly reduced by increasing the correlation between the two parasitics, i.e. by enforcing matching between them. Inequality (3.16) determines quantitatively the benefit deriving from matching enforcement. For an arbitrary parasitic pair (ℓ, j) , their mismatch and average sensitivities are computed. If relation (3.16) holds, the mismatch $\Delta p_{\ell j}$ and the average value $p_{\ell j}$ replace p_ℓ and p_j in the list of parasitics. In our approach, the magnitude requested to ratio $\left| \frac{S_{i,\Delta}}{S_{i,p}} \right|$ is user-defined. In our tests, we have obtained good results by requiring the ratio to be at least 10, and this value has been used in all the examples of this dissertation.

From the information on the range of each parasitic, the range of variation of the average and mismatch values is computed as

$$\begin{aligned} \frac{p_\ell^{(min)} + p_j^{(min)}}{2} \leq p_{\ell j} \leq \frac{p_\ell^{(max)} + p_j^{(max)}}{2}, \\ p_\ell^{(min)} + p_j^{(max)} \leq \Delta p_{\ell j}^{(mismatch)} \leq p_\ell^{(max)} + p_j^{(min)}, \quad \forall \ell \neq j. \end{aligned} \quad (3.17)$$

One can recognize that the constraint generation problem for parasitic average $p_{\ell j}$ and a simple parasitic component are identical. The constraint generation for parasitic mismatch on the other hand is handled in a slight different manner. Suppose a solution for $\Delta p_{\ell j}$ has

a negative value, then this term could possibly generate cancellation in the approximation formulae for performance degradation ΔK_i ; of equations (3.3) and (3.6), thus creating false results. To eliminate this problem, the variable Δp_{ℓ_j} is split into two, $\Delta p_{\ell_j}^+$ and $\Delta p_{\ell_j}^-$, which are consequently added as a contribution to the approximate negative and positive components of the performance degradation. $\Delta p_{\ell_j}^-$ and $\Delta p_{\ell_j}^+$ represent the lower- and upper-bound of the allowed mismatch.

This approach is often expensive computationally, since it involves the generation of a large number of mismatches, and the constraint generation problem has to be solved on a large set of parameters. A more efficient approach is the use the matching requirement expressed in (3.16). For every pair of parasitics, their mismatch and average sensitivities are computed and compared against each other. If relation (3.16) holds, the original parasitics are discarded and substituted by their average value and the mismatch. Otherwise they are kept and mismatches between them are not considered. With this approach, the computational cost of the constraint generation problem is affected very slightly, only one being the parameter added for each group of three or more matched parasitics.

Matching constraints are often expressed as a difference between parameter ratios rather than of simple parameters. This is often the case when trying to establish specific geometric constraints from topological constraints during the technology mapping process. Given two independent parameters x and y and a performance $K_i(x/y)$ then

$$S_{i,(x/y)} = \frac{\partial K_i}{\partial (x/y)} = \frac{y^2}{y-x} (S_{i,x} + S_{i,y}), \quad (3.18)$$

and thus, the sensitivity of performance K_i with respect to the mismatch difference $\Delta \frac{x}{y} = \frac{x_1}{y_1} - \frac{x_2}{y_2}$ can be written as

$$S_{i,\Delta \frac{x}{y}} = \left(\frac{S_{i,\frac{x_1}{y_1}} - S_{i,\frac{x_2}{y_2}}}{2} \right) = \frac{y_1^2}{2(y_1 - x_1)} (S_{i,x_1} + S_{i,y_1}) - \frac{y_2^2}{2(y_2 - x_2)} (S_{i,x_2} + S_{i,y_2}). \quad (3.19)$$

Throughout this analysis it was always assumed that x_i and y_i are statistically independent variables. This is usually the case in layout since orthogonal geometries, such as channel width and length in MOS transistors, are generally influenced by independent sources.

The computation scheme for arbitrary performance starting from generalized circuit analysis can be found in Appendix D.1.

Generating Matching Constraints

Assume for simplicity but without loss of generality that all active devices can be represented by a two port. Furthermore, assume that a model relating the output to the input port is available and that only one performance K_i is considered. Let V_{D_ℓ} and I_{D_ℓ} be the quantities characterizing respectively input and output of device D_ℓ . Let $I_{D_\ell} = f(\mathbf{\Pi}_0 + \mathbf{\Pi}, V_{D_\ell})$, where $\mathbf{\Pi}$ is the vector of all deviations from a nominal value $\mathbf{\Pi}_0$, of all technological parameters affecting the device. Suppose the set of sensitivities $\{S_{i,\Pi_{\ell m}}\}$ of performance K_i with respect to all vector elements of $\mathbf{\Pi}$ associated with device D_ℓ is available. Then, in first approximation, the degradation $\Delta K_{i,D_\ell}$ of performance K_i with respect to technological deviations in device D_ℓ can be expressed as

$$\Delta K_{D_\ell} = \sum_m S_{i,\Pi_{\ell m}} \Pi_m. \quad (3.20)$$

where, for reasons that we will be clear later, the sign of sensitivities has been dropped.

Consider now a pair of devices D_ℓ and D_j , then the degradation due to the parameter mismatch of the devices can be computed as

$$\Delta K_{i,D_\ell-D_j} = \sum_m S_{i,\Delta\Pi_m} \Delta\Pi_m. \quad (3.21)$$

where $S_{i,\Delta\Pi_m}$ is the sensitivity of K_i with respect to parameter Π_m , computed using (3.15) and $\Delta\Pi_m$ are the components of vector difference $\mathbf{\Pi}(D_\ell) - \mathbf{\Pi}(D_j)$.

Now, assuming that the components $\Delta\Pi_m$ are independent random variables with zero mean, the variance of the degradation of performance K_i with respect to the variances of the mismatches of all technological parameters relevant to the pair of devices D_ℓ and D_j , is computed as

$$\sigma^2(\Delta K_{i,D_\ell-D_j}) = \sum_m |S_{i,\Delta\Pi_m}|^2 \sigma^2(\Delta\Pi_m). \quad (3.22)$$

Consider for instance a pair of matched MOS transistors. The variance of the degradation due to technological mismatches can be expressed as

$$\begin{aligned} \sigma^2(\Delta K_{i,m_1-m_2}) = & S_{i,\Delta W}^2 \sigma^2(\Delta W) + S_{i,\Delta L}^2 \sigma^2(\Delta L) + S_{i,\Delta C_{ox}}^2 \sigma^2(\Delta C_{ox}) + \\ & S_{i,\Delta\mu_n}^2 \sigma^2(\Delta\mu_n) + S_{i,\Delta V_{TO}}^2 \sigma^2(\Delta V_{TO}). \end{aligned}$$

where W , L , C_{ox} , μ_n and V_{TO} are respectively channel width, channel length, gate oxide capacitance, mobility and threshold voltage of the transistors.

In [141] a direct relation has been shown between these variances and the relative orientation and distance between device pairs. This information can be used to translate the maximum allowed performance degradation into the physical separation and relative orientation between pairs of devices. In order to do this, estimations on the minimum and maximum attainable variances for a particular process are needed, to determine the upper- and lower-bound of the performance degradation for each pair of devices. At this point the sum of all degradations due to each pair of devices in the circuit should be added to equation (3.4) and the constrained optimization solved to find the actual variances of each parameter, and consequently numerical values for the physical quantities. Notice that, for consistency, the standard deviation should be added to (3.5), thus making the optimization harder and more time consuming. In order to avoid this additional complexity, the expression for the standard deviation is linearized after substituting the single parameter variances with analytical models in the geometric quantities of interest. Thus a linear expression for the standard deviation of the degradation referred to the pair is obtained as

$$\sqrt{\sigma^2(\Delta K_{i,m_\ell-m_j})} \simeq A_{\ell j}d_{\ell j} + B_{\ell j}r_{\ell j}. \quad (3.23)$$

where A and B are quantities which depend upon the performance sensitivity of K_i with respect to the pair's technology parameters. $d_{\ell j}$ and $r_{\ell j}$ represent the distance and relative rotation of devices m_ℓ and m_j . Also in this case a simplification mechanism similar to the one proposed in [116] is used for computing the criticality of the mismatch contributions.

The requirement of statistical independence for $\Delta\Pi_m$ can be relaxed if it can be assumed that: (1) the variables are Gaussian; (2) the variance-covariance matrix \mathbf{A} associated with them is known. In this case due to the positive-definitiveness and symmetric nature of \mathbf{A} , a method based on the \mathbf{LDM}^T factorization can be used to translate the original variable set onto one where all the variables are uncorrelated and, since Gaussian, also statistically independent. The method is discussed in detail in chapter 4.

In the remainder of this dissertation, we shall refer to parasitics, parasitic mismatches and technology mismatches related to devices and interconnect according to Table 3.1.

3.4.3 Device and Interconnect Symmetry

Symmetry is often used in the layout of analog integrated circuits to minimize the effects of mismatched parasitics on certain performances such as offset voltage, Com-

<i>symbol</i>	<i>meaning</i>
R_{S_i}	degeneration resistance at the source of transistor M_i
$R_{S_{i,j}}$	mismatch between R_{S_i} and R_{S_j}
C_i	substrate capacitance of net i
$C_{i,j}$	cross-coupling capacitance between nets i and j
V_{t_i}	voltage threshold of transistor M_i .
$V_{t_{i,j}}$	mismatch between V_{t_i} and V_{t_j}
V_{dd}	Supply voltage
ω_0	Unity-gain bandwidth
A_v	Low-frequency gain
V_{off}	Systematic offset
ϕ_M	Phase margin
τ_D	Switching delay

Table 3.1: *Notation for parasitics and performance functions*

mon Mode Rejection Ratio and noise. Symmetric placement and routing forces indeed the parasitics of differential signal paths to be matched, thus reducing non-uniform and unbalanced signals. Deriving topological symmetry constraints in an automatic fashion has been traditionally associated with pattern recognition or expert systems.

An alternative approach, fully quantitative, based on rigorous matching analysis and graph-searching techniques was proposed by us in [2]. First, parasitic and device matching constraints are computed as described in section 3.4.2. Then, the circuit is mapped onto a graph. Topological symmetry constraints are derived for each node of the graph using a search algorithm driven by matching constraints. Since the ultimate goal of topological symmetry is to facilitate the respect of matching constraints on parasitics and / or devices, first priority to undergo a symmetrization process is given to those entities² on which *matching constraints* have been imposed. A second requirement is that the entities belong to *distinct differential signal paths* in order to maximally balance the signals circulating in the circuit. A third requirement is that *symmetric signal paths ultimately converge to a virtual or real signal ground*. These three basic requirements have been used in our search algorithm.

The first step consists of converting the circuit description into an undirected graph, whose nodes represent the (active and passive) devices and edges connectivity. In the second

²Devices or interconnect

```
map_onto_graph (hardware description);
add_edges (matching constraints);
create_sink_nodes (node information);
starting_nodes = differential_input;
foreach node
  if termination_condition           // verify convergence or termination conditions
    find_new_starting_node;
    continue;
  else
    search_matched_nodes;           // look for the next nodes
    search_singular_paths;         // terminate reconvergent paths
    if not_degenerate               // continue search if
      continue;                   // path is not degenerate
    if semi-degenerate              // detect and correct semi-degenerate paths
      if super_virtual
        continue;
      invalidate_path;             // no symmetry constraints in path
reorder_symmetries;
```

Figure 3.1: *Pseudo-code of the graph-based symmetry constraint generator*

step the matching constraints are introduced into the graph by adding constraint edges to it. The third step consists of finding the edges associated with real and virtual grounds. This operation is automatically performed in all nets by comparing common and differential mode gains with respect to the input. The usefulness of this characterization will be evident further on.

At this point, after initializing the search, all nodes are searched until the maximum number of symmetric paths has been found, the termination condition for each path being the presence of virtual or real ground. In the first case the symmetry path is marked *reconvergent*, thus symmetry constraints have been partially satisfied. In order to be completely satisfied the symmetric search paths must merge and proceed on the graph until a real ground is found. In the second case the symmetry constraints for the current search path(s) are completely satisfied. Thus the search ends and a new one begins from another couple of non-terminal nodes satisfying matching constraints.

If the path is found to be *semi-degenerate*, i.e. devices are found to be at both sides of the symmetry axis, a pair of edges associated to a *super-virtual ground* is searched. The symmetry constraint associated to the current search path is invalidated when no such edges are found. A *super-virtual ground* is defined as a pair of signal nets such that, if connected by a series of two identical resistors, the middle point becomes a virtual ground.

If a *degenerate path* is detected the search path is invalidated, thus nulling the symmetry constraints found associated to this path. A degenerate configuration occurs whenever a device on a search path has devices matched to it on an incompatible search path, thus forcing separate symmetry constraints to interfere with each other.

The final steps are used to map all search paths onto actual symmetry constraints. Figure 3.2 shows a simple circuit and the graph associated with it, when a degenerate path exists. Nodes "1" are visited first. Since a terminal condition occurs (real ground), the search continues in the opposite direction. After one iteration at nodes "2", the search proceeds towards the nodes "3", however since neither nodes are in a contiguous path a degenerate situation occurs. The last section of the path is therefore invalidated and the search continues from nodes "4", ending at the top, when a terminal condition occurs. Thus the search is complete resulting in the symmetry constraints M1-M2; M3-M4 and M7-M8.

Figure 3.3 illustrates a semi-degenerate situation. In this case electrical nets *A* and *B* clearly form a super-virtual ground. Thus, the constraint cycle in the symmetry constraint dependence is eliminated and the symmetry constraint is accepted.

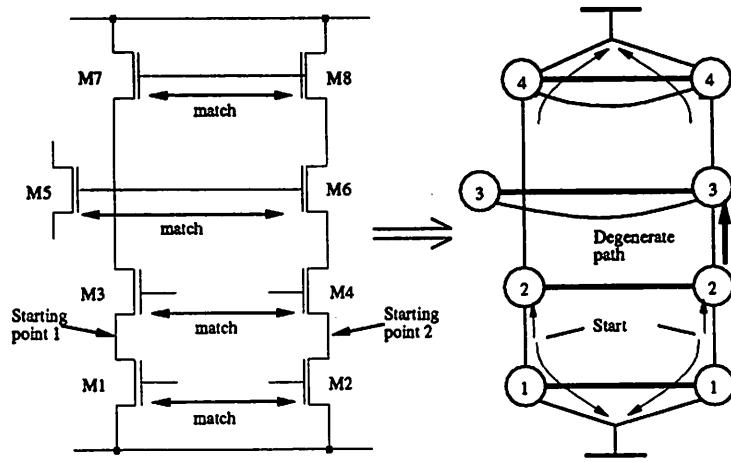


Figure 3.2: *Degenerate path*

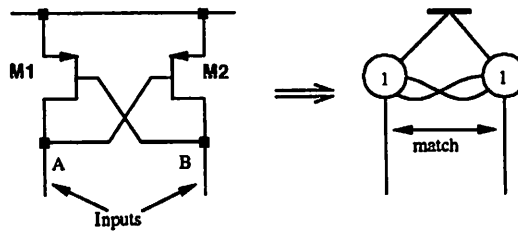


Figure 3.3: *Semi-degenerate path*

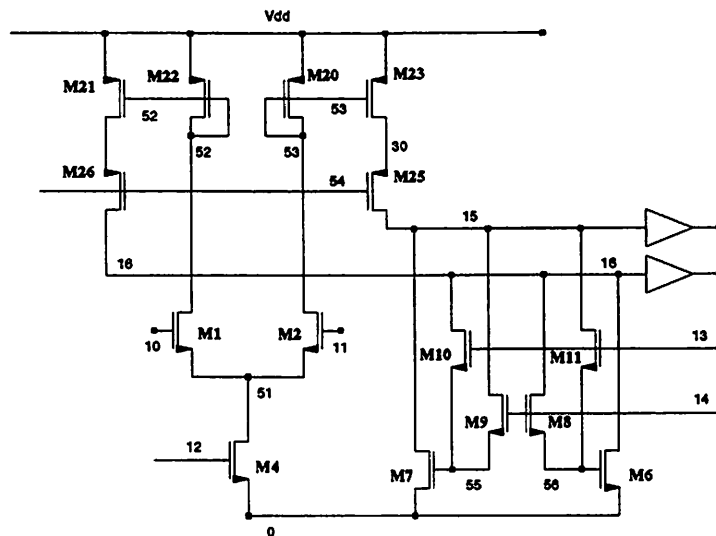


Figure 3.4: Clocked comparator COMPL

3.5 Constraint Generation: A Case Study

As a practical example, consider the clocked comparator COMPL, whose schematic is shown in Figure 3.4. This comparator has been used as a benchmark in several recent works on analog CAD [69, 142], due to its relevant performance sensitivity to layout details. Consider the following stray resistances (see Table 3.1 for notation) and the corresponding sensitivities of systematic offset V_{off} with respect to each of them:

$$\mathbf{p} = \begin{bmatrix} R_{S.1} \\ R_{S.2} \\ R_{S.3} \\ R_{S.4} \\ R_{S.6} \\ R_{S.7} \\ R_{S.20} \\ R_{S.21} \\ R_{S.22} \\ R_{S.23} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 56.53 \\ -56.53 \\ 0.202 \\ -0.202 \\ 11.83 \\ -11.83 \\ 16.76 \\ -16.76 \\ -16.72 \\ 16.72 \end{bmatrix}^T$$

Offset sensitivities to resistances are expressed in $\mu V/\Omega$. They were computed by SPICE [137] with a precision within the third digit. Therefore for each of the pairs $R_{S_{1,2}}$, $R_{S_{3,4}}$, $R_{S_{6,7}}$, $R_{S_{20,21}}$, $R_{S_{22,23}}$, $R_{S_{21,23}}$, $R_{S_{20,22}}$, $R_{S_{21,22}}$ the ratio (3.16) is $\left| \frac{S_{i,\Delta}}{S_{i,p}} \right| \geq 10^3$, i.e. the resistive mismatch is at least 10^3 times more important for offset than the absolute values of these resistances. By simplification (3.14), offset sensitivities with respect to mismatches become

$$\mathbf{p} = \begin{bmatrix} R_{S_{1,2}} \\ R_{S_{20,23}} \\ R_{S_{21,23}} \\ R_{S_{20,22}} \\ R_{S_{21,22}} \\ R_{S_{6,7}} \\ R_{S_{3,4}} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 56.53 \\ 16.76 \\ 16.74 \\ 16.74 \\ 16.72 \\ 11.83 \\ 0.201 \end{bmatrix}^T$$

The cumulative effect of all average values on performance degradation is negligible according to (3.13), and therefore they are all eliminated from \mathbf{p} . The symmetry-constraint graph-search algorithm detected the following symmetric net pairs:

$$(52, 53), (15, 16), (10, 11), (13, 14), (55, 56)$$

and the following device pairs:

$$(M_1, M_2), (M_{20}, M_{22}), (M_{21}, M_{23}), \\ (M_{25}, M_{26}), (M_6, M_7), (M_{10}, M_{11}), (M_8, M_9).$$

Performance constraints are enforced on the max switching delay τ_D and on systematic offset V_{off} :

$$\begin{aligned} \tau_D &\leq 7 \text{ ns} \\ |V_{off}| &\leq 1 \text{ mV} \end{aligned} \quad (3.24)$$

In the first steps of layout, we assume that the nominal value of all parasitics is 0, i.e. $\mathbf{p}^{(0)} = [0 \dots 0]^T$. Simulation yields a nominal value of the switching delay $\tau_D^{(0)} = 4 \text{ ns}$ and null offset. Therefore

$$\mathbf{K} = \begin{bmatrix} \tau_D \\ V_{off} \\ -V_{off} \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 4.0 \text{ ns} \\ 0.0 \\ 0.0 \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 3.0 \text{ ns} \\ 1 \text{ mV} \\ 1 \text{ mV} \end{bmatrix}$$

As expected, sensitivity analysis shows that delay is sensitive to stray capacitances, while resistances and mismatch affect only offset:

$$\mathbf{p} = \begin{bmatrix} C_{15} \\ C_{16} \\ C_{55} \\ C_{56} \\ R_{S_{-1,2}} \\ R_{S_{-20,23}} \\ R_{S_{-21,23}} \\ R_{S_{-20,22}} \\ R_{S_{-21,22}} \\ R_{S_{-6,7}} \\ R_{S_{-3,4}} \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} 36 \text{ ps/fF} & 0.0 & 0.0 \\ 36 \text{ ps/fF} & 0.0 & 0.0 \\ 47 \text{ ps/fF} & 0.0 & 0.0 \\ 47 \text{ ps/fF} & 0.0 & 0.0 \\ 0.0 & 0.056\text{mV}/\Omega & 0.056\text{mV}/\Omega \\ 0.0 & 0.016\text{mV}/\Omega & 0.016\text{mV}/\Omega \\ 0.0 & 0.016\text{mV}/\Omega & 0.016\text{mV}/\Omega \\ 0.0 & 0.016\text{mV}/\Omega & 0.016\text{mV}/\Omega \\ 0.0 & 0.016\text{mV}/\Omega & 0.016\text{mV}/\Omega \\ 0.0 & 0.011\text{mV}/\Omega & 0.011\text{mV}/\Omega \\ 0.0 & 0.201\mu\text{V}/\Omega & 0.201\mu\text{V}/\Omega \end{bmatrix}^T$$

Because of symmetries, and since the nominal value of mismatch is 0, offset sensitivities in the positive and negative direction are equal. We use the following conservative minimum and maximum parasitic estimates:

$$C^{(min)} = 1 \text{ fF}$$

$$C^{(max)} = 100 \text{ fF}$$

$$R^{(min)} = 0$$

$$R^{(max)} = 50\Omega$$

With these estimates, PARCAR computed the following set of parasitic bounds:

$$\mathbf{p}^{(bound)} = \begin{bmatrix} 71.96 \text{ fF} \\ 71.96 \text{ fF} \\ 78.52 \text{ fF} \\ 78.52 \text{ fF} \\ 1.0 \Omega \\ 7.4 \Omega \\ 7.4 \Omega \\ 7.4 \Omega \\ 7.5 \Omega \\ 19.9 \Omega \\ 49.5 \Omega \end{bmatrix}$$

Here the relation between sensitivity and tightness of bounds is evident. Only a few parameters affect critically the performance of this circuit and therefore need be bounded tightly. In practice, only the mismatch between the source resistances in the differential pair and between the two current mirrors (M_{20}, M_{23}) and (M_{21}, M_{22}) are responsible for offset. Details on the statistics for the constraint generation of this circuit are available in section 9.1.1.

Chapter 4

Placement

*Stavvi Minòs orribilmente, e ringhia:
essamina le colpe ne l'intrata;
giudica e manda secondo ch'avvinghia.*

*Dico che quando l'anima mal nata
li vien dinanzi, tutta si confessa;
e quel conoscitor de le peccata
vede qual loco d'inferno è da essa;
cignesi con la coda tante volte
quantunque gradi vuol che giù sia messa.*

Dante Alighieri, "Inferno", Canto V

In this chapter we explore the features needed by an effective placement tool to cope with analog constraints. We show how performance models derived in chapter 3 can be efficiently used to drive a Simulated Annealing placement algorithm to enforce analog constraints and ultimately a set of high-level performance specifications. Efficient techniques for fast evaluation of substrate noise and their implementation within the Simulated Annealing algorithm are described. The effects of all the proposed changes on the annealing are carefully modeled and the impact on the algorithm convergence is discussed. The placement methodology, implemented in a tool called PUPPY-A, is illustrated throughout the

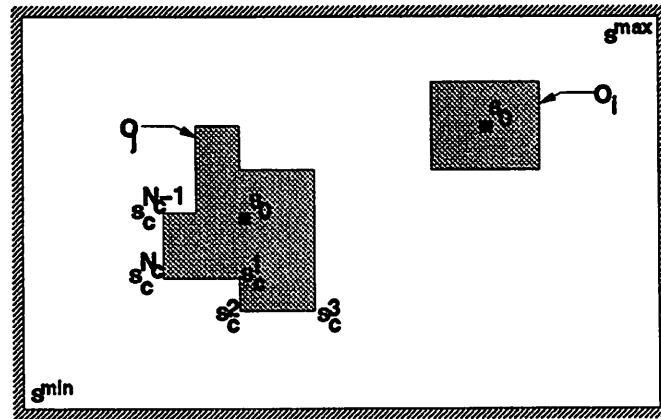


Figure 4.1: Object definition

description with an example used to highlight the main features of the approach. PUPPY-A is part of the OCTTOOLS layout tool-set.

4.1 Evolution from the Digital to the Analog Domain

In recent years layout design automation for analog integrated circuits has generated considerable interest due to the increasing complexity of the problems and the sophistication of proposed solutions. The challenges generated by denser technologies and by on-chip integration of mixed analog/digital circuits have led to more and more complex and creative algorithms. In order to cope with higher performance sensitivity to layout topology and interconnect parasitics in analog circuits, several techniques have been proposed. In most of these approaches, methodologies derived from the digital world have been used.

4.1.1 Placement Problem Formulation

The placement problem is referred to as the task of finding a physical location for a number of layout objects in order to satisfy a number of *constraints* and to minimize a *cost*. Let O_j , $j = 1, \dots, N_O$ be a layout object, $s_c^{(i)}(O_j)$, $i = 1, \dots, N_c(O_j)$ the vertices of its perimeter, and $s_0(O_j)$ its center¹, as illustrated in Figure 4.1. Let C^A be the set of all

¹The center of a polygon can be defined in a number of ways (center of mass, arbitrary edge, etc.). In this work polygons are always approximated with rectangular objects and the center is assumed to be the center of mass of the each object.

absolute constraints on the center or perimeter of object O_j , typically of the form

$$\begin{aligned} & \mathbf{s}^{(min)} \leq \mathbf{s}_0(O_j) \leq \mathbf{s}^{(max)}, \quad \forall j = 1, \dots, N_O, \text{ or} \\ & \mathbf{s}^{(min)} \leq \mathbf{s}_c^{(i)}(O_j) \leq \mathbf{s}^{(max)}, \quad \forall i = 1, \dots, N_c(O_j), \quad \forall j = 1, \dots, N_O, \end{aligned} \quad (4.1)$$

where $\mathbf{s}^{(min)} < \mathbf{s}^{(max)}$ are reference points. These constraints are required to fix the mobility of an object within boundaries determined by considerations on the entire chip or module under construction. Let C^R be the set of all relative constraints, i.e. the constraints relating pairs or groups of objects to each other, typically of the form

$$s_x(O_j) \leq s_x(O_i); \quad s_y(O_j) \leq s_y(O_i), \quad (4.2)$$

where $\mathbf{s}(O_j) = [s_x, s_y]^T$ is an arbitrary point (on the perimeter or in the center) relative to the O_j .

Assume that each object O_j can be represented in terms of a collection of simpler four-sided objects \bar{O}_j , called *primitives*. Each object \bar{O}_j is completely specified by the following features: \mathbf{s}_0 , r_0 , ℓ and w . r_0 relates to the orientation of the object, while ℓ and w to its length and width, respectively.

Furthermore, assume that a Manhattan style design is adopted for all our layouts. Then, r_0 can assume only the following self-explanatory values:

NO_ROTATE, ROTATE_90, ROTATE_180, ROTATE_270, MIRROR_X, MIRROR_Y, MIRROR_YX

For a given circuit, let us define the *placement configuration* S as the set of quadruples $\{(\mathbf{s}_0(\bar{O}_j), r_0(\bar{O}_j), \ell(\bar{O}_j), w(\bar{O}_j)) \mid \forall j = 1, \dots, N_O\}$ which determine the location and orientation of each object in the layout. When S satisfies the constraints of C^A and C^R , it is called *legal configuration*. Let us define the set of all configurations as $\{S\}$.

Finally, let us define a function $f(S)$ as the cost associated with configuration S .

The placement problem in its most general formulation consists of finding a legal configuration S associated with the minimum cost f^* . The problem can be expressed in terms of the following optimization

<i>minimize</i> : $f(S)$	(4.3)
<i>subject to</i> :	
C^A and C^R	

In this formulation the placement problem is *NP*-hard, however over the years a number of heuristics have been developed to find sub-optimal solutions in lesser CPU time.

The best known placement heuristics are generally referred to as: *constructive or schematic-driven, branch-and-bound search and partitioning-based, quadratic optimization-based, and iterative improvement techniques.*

4.1.2 Constructive or Schematic-Driven Techniques

Placement methods based on these techniques attempt to solve (4.3) by proceeding as follows

1. the circuit schematic is translated into a set of *functional units*².
2. a *module*³ for each functional unit is generated as a single item or as a set of equivalent alternatives.
3. analog constraint sets C^A and C^R are annotated into the circuit schematic in a form of *mutual relations*⁴
4. every module is placed enforcing all mutual relations

These methods were among the first to be used in digital physical assembly [143]. Step 1 is generally performed using knowledge-based [92] or pattern-recognition methods [144]. In general, modules are created in step 2 with procedural generators [92] similar to traditional silicon compilers (see chapter 2). Finally, the placement of step 4 is performed in two phases: initial placement and iterative improvement. A large number of algorithms have been proposed for the initial placement phase. Among others, force-directed and clustering algorithms [145] and topological sort [41] have been employed in most systems [92, 105, 128]. A number of iterative improvement methods have been proposed. A classical approach consists of performing global floorplanning, using for example a slicing-tree scheme [58], and exhaustive search to map the best module alternative with the physical space associated with the floorplan.

Although fast, schematic-driven methods lack of flexibility when a knowledge-base is used as the engine for the creation of modules and relations, since a new, very elaborated rule system often need be implemented for each technology. Moreover, due to the inherent

²A functional unit is defined as a single device or as a cluster of devices which perform a given function, e.g. differential pair, mirror, etc.

³A module is here defined as the physical realization of a device or sub-circuit component.

⁴A mutual relation is defined as a clause stating a type of dependence between objects, e.g. *a match b; a symmetric_to b.*

sequential nature, constructive placement algorithms are not good candidates to solve the analog placement problem due to the large number of trade-offs in implementing each module and in dealing with the complex interactions through interconnect and substrate interference, which require the ability of performing optimization at a global level.

4.1.3 Branch-and-Bound Search and Partitioning-Based Techniques

In branch-and-bound methods, search is carried out along a decision-tree. Each branch of the tree corresponds to the selection of the orientation or a constraint on a pair of modules. Each tree node corresponds to a partial configuration \bar{S} . Suppose now that a lower-bound on the objective function $f(S)$ is known, then all the branches leading to a configuration of higher cost are *pruned*. In designing the pruning techniques, one must insure that all solutions be *in principle* searched and that a solution will be found. Appropriate techniques, the key for improving the efficiency of the algorithm, are reported e.g. in [143]. The *branching* operation consists of selecting the next module or constraint to be considered by the algorithm.

Branch-and-bound criteria have been applied to a number of search algorithms for one-dimensional device placement, e.g. [146], and module generation, e.g. [147], where the size of the device chains was kept relatively small. Recently, a similar approach has been used in [148] for the placement problem in the presence of analog-specific constraints between pairs of devices. However, only layouts of limited size could be considered, while larger circuits required the use of hierarchical decomposition before applying the algorithm.

Partitioning-based methods attempt to divide the original problem into smaller sub-problems to improve the overall efficiency. A large number partitioning schemes based on the Kernighan and Lin min-cut algorithm [53] can be found in the literature, e.g. [149, 150, 104, 54] and more recently [151]. Analog-specific constraints, which are generally global to the circuit, are hard to enforce using such methods, since they need be distributed properly among all the partitions, thus creating a large overhead in maintaining full constraint consistency. In [50] a modified one-dimensional standard-cell placement algorithm derived from [54] was used, to create a floorplan separating sensitive analog circuitry from digital, fast-switching circuits. In [101] on the contrary, min-cut partitioning was employed to enforce symmetry constraints.

4.1.4 Quadratic Optimization-Based Techniques

If $f(S)$ is a measure of the total interconnect length of the circuit, problem (4.3) reduces to

$$\begin{aligned} & \text{minimize : } \sum_{i=1}^{N_n} \bar{L}(n_i) & (4.4) \\ & \text{subject to:} \\ & C^A \text{ and } C^R \end{aligned}$$

Items $n_i, \forall i = 1, \dots, N_n$ are the circuit's electrical nets and $\bar{L}(n_i)$ is the estimate of the total wiring length of n_i . The constraints in C^A and C^R represent the requirement that the cells be within chip boundaries and do not overlap with each other, respectively. Term $\bar{L}(n_i)$ can be computed, for example, as

$$\bar{L}(n_i) = \sum_{O_j, O_k \in \mathcal{N}(n_i)} \|s_0(O_j) - s_0(O_k)\|^2,$$

where $\mathcal{N}(n_i)$ is the set of all objects connected to net n_i . Problem (4.4) can be translated into quadratic program [152]

$$\begin{aligned} & \text{minimize : } \mathbf{x}^T \mathbf{Q} \mathbf{x} & (4.5) \\ & \text{subject to:} \\ & \mathbf{P} \mathbf{x} \leq \mathbf{0} \end{aligned}$$

where the $N_w \times 1$ vector \mathbf{x} represents the lengths of horizontal and vertical wiring segments connecting the objects in each net and $N_w \times N_w$ matrix \mathbf{Q} the information on the *weight* of each segment. $N_w \times N_w$ matrix \mathbf{P} is a compact representation of the constraints in C^A and C^R .

By appropriately weighting \mathbf{Q} it is possible to introduce the concept of *net criticality*, i.e. the sensitivity of performance with respect to the length of a given wiring segment. Moreover, in \mathbf{P} one can introduce explicit constraints on the length of each segment. Thus, by using the distance measure between objects and a fix point, one can implement a number of constraints typically used in analog physical assembly, such as symmetry and device matching. In general, Quadratic optimization placement is efficiently solved using QP methods, which require positive definiteness of \mathbf{Q} as necessary condition for the convergence to a solution.

More recently, a number of efficient methods for the solution of (4.5) were proposed for *standard-cell* [145, 153] and *macro-cell* [154, 155, 156] problems⁵. A number of authors have proposed techniques to introduce additional constraints related to performance. In [157] for example timing constraints are added to \mathbf{P} and heuristics based on Lagrangian Relaxation are used to efficiently find an approximate solution to the constrained problem. In [158] the net weights in \mathbf{Q} are calculated based on the expected switching activities of gates driving them.

Recently, quadratic optimization has been used to enforce a number of analog-specific constraints. In [159] for example the layout is partitioned into smaller regions, while the original analog-specific constraints are mapped onto each partition using quadratic optimization. The process is continued iteratively until each partition contains a single module. The final placement in each partition is performed using designer-assisted exhaustive enumeration on the module shapes that best adapt to the partition's shape.

The major drawback of pure quadratic optimization is the relatively low flexibility in accounting for analog-specific constraints. In methods combining quadratic optimization with other techniques, although significantly faster, global constraints such as symmetry and substrate noise immunity require complex schemes to be appropriately enforced. Furthermore, the final solution is generally not aligned with the layout grid of the workspace, hence potentially harmful adjustments are always needed at the end of the placement⁶.

4.1.5 Iterative Improvement Techniques

Iterative improvement techniques make use of a small *perturbation* to configuration S to obtain, progressively, an improvement in cost $f(S)$. Each perturbation applied to S leads to a new configuration S' which can be *accepted* or *rejected* based on some measure of the *cost improvement* $\Delta f = f(S') - f(S)$. The iteration continues until a *stopping condition* occurs. The initial configuration S_0 used as a starting point by the algorithm can be pre-optimized or random, while the solution S_f is said *global minimum/maximum* of the problem.

A great number of iterative improvement methods have been proposed. The main

⁵In standard-cell problems the size of each object O_i is identical in one or in both directions. In macro-cell problems on the contrary, the modules may have arbitrary size and shape.

⁶A solution to this problem is the use of a compaction tool based on the Constraint-Graph algorithm and Linear Programming, which does not require the initial solution to be aligned with the grid for an efficient solution of the compaction problem.

```
construct_initial_placement;           // constructively or randomly
repeat
  select_objects_to_move;
  tentatively_move_selected_objects;
  evaluate_cost;                       // calculate  $f(S)$ 
  if move_accepted                     // downhill/hill climbing criterion
    finalize_move;
  else
    undo_move;
until stopping_condition_satisfied
```

Figure 4.2: Pseudo-code of a generic iterative algorithm

differences between the methods relate to (1) the type of perturbation, (2) the evaluation of the cost improvement, (3) the acceptance/rejection criterion, and (4) the stopping condition. Two main classes of iterative improvement methods exist: the *downhill or descent* and the *hill climbing* algorithms. Descent algorithms are said *greedy*, since a cost improvement is required for the acceptance of a new configuration. A necessary condition for these algorithms to converge to a global minimum is that $f(S)$ be a convex function defined on convex set $\{S\}$ [160, p. 181]. A sufficient condition for descent algorithms to converge to a *local minimum* is given by the Global Convergence Theorem [160, p. 187].

Hill climbing algorithms are similar to descent algorithms, with the exception of the acceptance/rejection criterion, which allows the algorithm to tolerate temporary worsening in the cost. The provision was invented to make the optimizer capable of escaping local minima to reach the global minimum with higher probability⁷. The pseudo-code of a generic iterative algorithm is given in Figure 4.2. A number of selection techniques for both objects and move types have appeared in the literature. Perhaps, the randomized n -tuple interchange has received most attention due, mainly, to its simplicity. Although $n > 2$ schemes have been extensively studied [161], the advantages do not seem to justify added complexity. Hence, most of the early work in this area is based on $n = 2$ schemes [162, 163]. Over the years, cost evaluation and acceptance mechanisms have generated significant research activity in the field.

⁷Under certain conditions it is possible to show that this probability is in the limit 1.

```
create_randomized_placement;
set_initial_temperature;
repeat
  repeat
    select_objects_to_move;
    tentatively_move_selected_objects;
    evaluate_cost;
    if move_accepted
      finalize_move;
    else
      undo_move;
  until equilibrium_reached
  update_temperature;
until freezing_point_reached
```

Figure 4.3: *Pseudo-code of a generic SA algorithm*

In the remainder of this chapter, we will focus on hill-climbing algorithms. A number of approaches derived from natural phenomena have appeared in the literature [66, 164, 165, 166, 167, 168]. Simulated Annealing (SA) [66] in particular, has proven its suitability for the placement problem in a number of design styles [169, 170, 171]. Layout tools based on SA have generally great flexibility in the number and type of measures being minimized, hence the algorithm was a logical choice for many analog circuit design systems. The use and modification to the algorithm for the analog placement problem is the main topic of section 4.2.

4.2 Simulated Annealing and Analog Placement

In this section we present the generic formulation of SA and the modifications that were made to adapt it to the analog placement problem.

4.2.1 Terminology

A generic SA algorithm is shown in Figure 4.3. The algorithm is completely specified by the following:

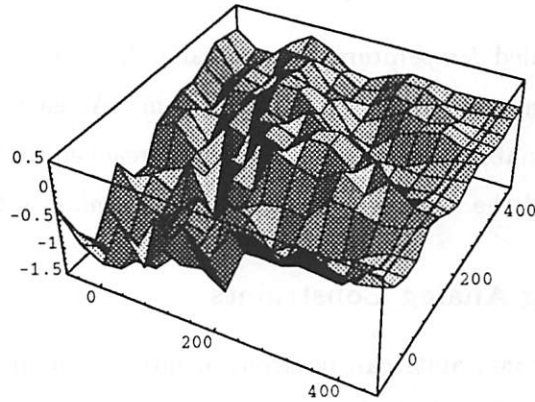


Figure 4.4: *Normalized energy landscape in typical placement problem*

- configuration space $\{S\}$
- cost function $f(S)$
- move-set M
- cooling schedule $T_{k+1} = \Gamma(T_k)$

The configuration space $\{S\}$, also called *search space* is the space of all possible placement configurations S associated with a given circuit. In bounded workspaces within a integer coordinate system, the size of $\{S\}$ grows factorially with the number of objects N_O .

In general, cost $f(S)$ is an analytic function of configuration S . Ordinarily, the cost has been used in the past to quantify basic chip features such as area and routing and to drive the algorithm towards a design-rule violation free layout. With the introduction of SA in placement tools for analog design [67, 119], the cost function has been also used to *enforce* analog-specific requirements. The plot of $f(S)$ versus all possible configurations $S \in \{S\}$ is called *energy landscape* of the placement problem. Figure 4.4 shows the energy landscape observable in a typical placement problem.

The move-set M is the collection of all legal moves which are performed on a n -tuple of objects. In most applications n can vary from one to two *during* an algorithmic run [67, 119]. The move, and hence new configuration S' , is accepted with probability 1 a cost improvement occurred, i.e. $\Delta f = f(S') - f(S) < 0$, otherwise the probability of acceptance

P is given by

$$P = e^{-\frac{\Delta f}{T}}.$$

Parameter T , called *temperature*, holds a value T_0 initially and steadily decreases to its final value T_f during the course of the algorithm. At each temperature the algorithm performs t_k isothermal moves, until equilibrium is reached for that temperature. The sequence $\{(T_k, t_k)\}_{k=1}^{\mathcal{K}}$ and the value \mathcal{K} are referred to as *cooling schedule*.

4.2.2 Characterizing Analog Constraints

Analog-specific constraints can be grouped into two main categories: *topological* and *parasitic* constraints. Topological constraints are aimed at controlling the relative location and orientation of each geometric object in the layout. Constraints on matching and symmetry belong to this category. A special case of matching relates to the need of clustering objects in a particular area of the workspace for reasons of noise immunity and latch-up suppression [172] or simply to reduce the number of separate well regions, as in the case of devices of the same type in CMOS circuits [67, 119].

Consider first symmetry constraints. Let $\mathbf{s}_0(O_j) = [x_0, y_0]^T$ be the location of an object O_j in the workspace and $r_0(O_j)$ its orientation as defined in section 4.1.1. Moreover, let $\mathcal{S}_m, \forall m = 1, \dots, N_S$ be the set of all n -tuples⁸ of objects on which symmetry constraints are to be enforced with respect to vertical axis A_m . Let $x(A_m)$ be the location of axis A_m in x-direction⁹. Then, all symmetry constraints are expressed by the following set of equations

$$\left. \begin{aligned} y_0(O_j) - y_0(O_i) &= 0, \\ x_0(O_j) - 2x(A_m) + x_0(O_i) &= 0, \\ r(O_j) &= \text{mirror_symmetric}[r(O_i)]. \end{aligned} \right\} \text{with } (O_i, O_j) = 2\text{-tuple} \in \mathcal{S}_m. \quad (4.6)$$

$$x_0(O_j) - x(A_m) = 0, \quad \text{with } (O_j) = 1\text{-tuple} \in \mathcal{S}_m.$$

Definition and enforcement of symmetry constraints¹⁰ is shown in Figure 4.5a for objects O_i, O_j and O_k .

Consider now matching constraints. Let $\mathcal{M}_m, \forall m = 1, \dots, N_{\mathcal{M}}$ be a set of matched objects, also called *matching cluster*. Assume that for each cluster, there exist a set of

⁸Term $n = 1$ in case of self-centered symmetry, otherwise $n = 2$.

⁹The problem can be formulated in an identical fashion for horizontal symmetry constraints.

¹⁰The “F” symbol in Figure 4.5 represents mirror-symmetry constraints, while “A” represents center-symmetry.

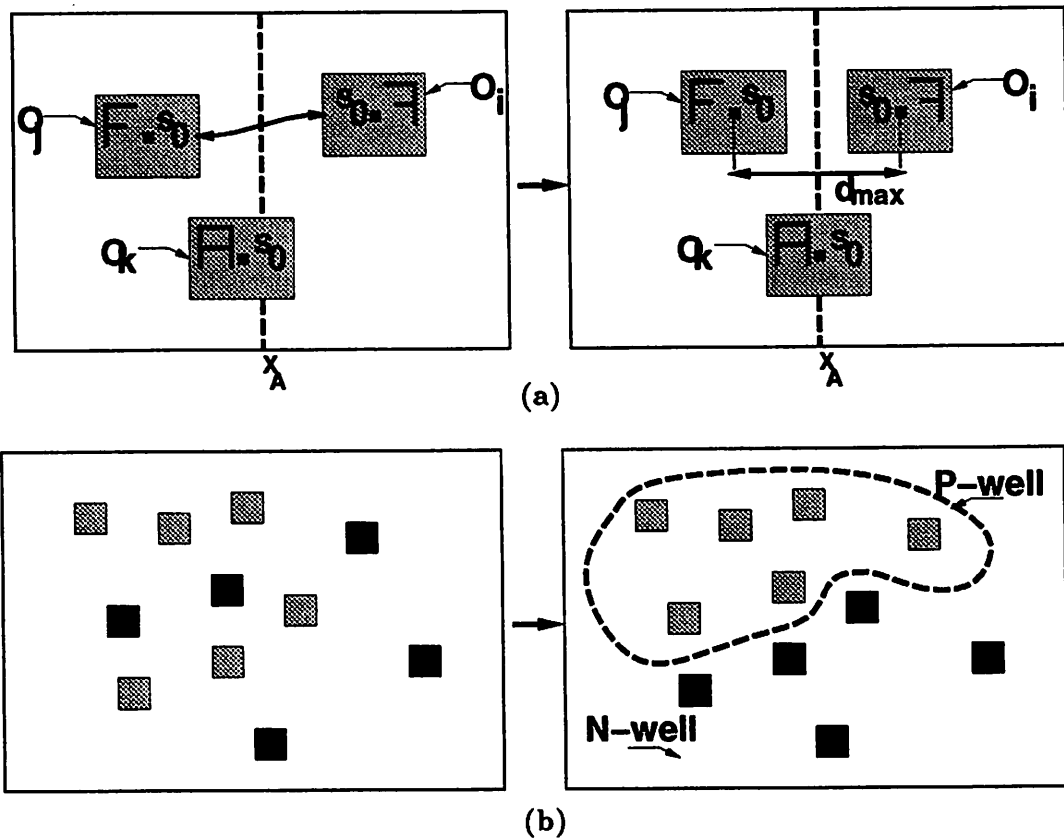


Figure 4.5: Topological constraints in analog circuit design: (a) symmetry and matching; (b) well minimization

parameters $d_{\mathcal{M}_m}^{(max)}$ and $r_{\mathcal{M}_m}^{(max)}$, i.e. the bound on the maximum distance and relative orientation mismatch associated with the objects in the cluster¹¹. Then, matching constraints are expressed by the following set of equations

$$\left. \begin{aligned} |\mathbf{s}_0(O_j) - \mathbf{s}_0(O_i)| &\leq d_{\mathcal{M}_m}^{(max)} \\ r_0(O_j) - r_0(O_i) &\leq r_{\mathcal{M}_m}^{(max)} \end{aligned} \right\} \text{ with } (i, j) = \text{pair} \in \mathcal{M}_m. \quad (4.7)$$

Figure 4.5a shows the enforcement of matching constraints on the pair O_j - O_i .

Finally, consider the special case of device matching used for the minimization of the number of wells. Expressions similar to (4.7) are used for the generation of the well regions, where $r^{(max)} \rightarrow \infty$ and $d^{(max)}$ is generally calculated on-the-fly as a fraction of the current chip section. Figure 4.5b shows a configuration in which devices of types requiring different well doping are mixed within the workspace (left-hand side). The diagram on the right-hand side shows the layout after enforcement of well-dependent matching, thus reducing the number of well structures needed and simplifying the biasing circuits.

Parasitic constraints are generally defined *explicitly* as a set of bounds on particular parasitic components or *implicitly* in terms of the global performance degradation that parasitics induce. Explicit parasitic constraints can be either derived as a solution to Problem 1 or imposed by the user. Explicit constraints are generally related to a particular parasitic element, which in turn is associated to a individual geometric structure. Thus, explicit parasitic constraints are useful when *local* decisions need be made. In placement however, all implementation details of the chip are not yet available and decisions at the highest level need be made. Hence, a more appropriate way of controlling parasitic effects is to build parasitic-based performance models and to evaluate the resulting performance degradation at each step of the algorithm.

In the literature both approaches to the control of parasitics in placement tools can be found. Cohn [173] proposed to realize some of the most critical routing artwork on-the-fly during the unfolding of the annealing, while user-defined critical net-couplings were minimized. In [174] the effects of substrate noise on user-defined critical modules were minimized using a rough approximate of substrate transport mechanisms at each step of the annealing. In our work, we have proposed for the first time the use of implicit parasitic constraints in the placement problem [119, 175, 176]. Alternatively, explicit constraints

¹¹ $d^{(max)}$ and $r^{(max)}$ can be computed using the techniques described in chapter 3 or arbitrarily imposed by the designer.

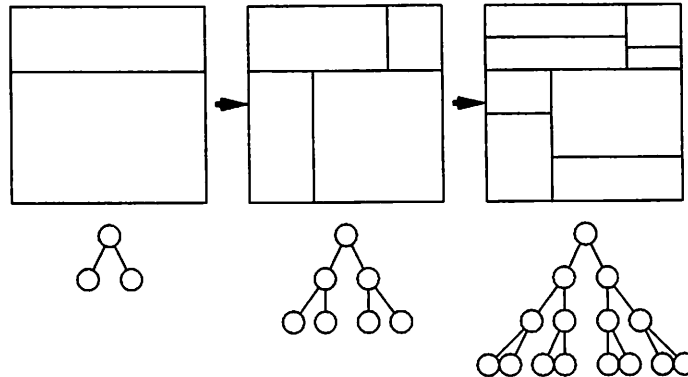


Figure 4.6: *Slicing-tree space representation for layout optimization algorithms*

were used in the other phases of the layout, namely routing, compaction and extraction. In the remainder of this chapter we will describe the techniques used for the enforcement of parasitic constraints using SA based placement algorithms.

4.2.3 Slicing-Tree vs. Flat Representation of the Workspace

Due to the irregular size and shape of analog devices, most analog placers were derived from digital-specific tools designed to work in macro-cell design style or as floor-planners. Two placement concepts using SA have acquired popularity in the digital domain. The first, due to Otten [177, 178], is based on a *slicing-tree* representing the workspace. A slicing-tree is a graph which represents the iterative partitioning of the space into successively smaller regions as illustrated in Figure 4.6. The annealer, manipulating the slices, i.e. the nodes in the graph, does not move the physical objects directly but modifies their **relative position** by altering the aspects of the slices in the tree. Hence, no overlap is possible between cells during the unfolding of the algorithm, thus potentially improving the efficiency of the approach, but simultaneously preventing the algorithm from searching desirable areas of the energy landscape accessible though non-legal configurations. This placement model was adapted to the analog problem in ILAC [57], where symmetry-preserving swaps were introduced. However, to the best of our knowledge, no provision was given to control matching or parasitic constraints during the placement stage.

The second concept, due to Gelatt & Jepsen [179], uses a *flat* representation of the workspace. According to this model, objects are translated and rotated directly by the annealer, which manipulates their **absolute position**. Each manipulation occurs on a

gridless plane and overlaps are allowed, thus the algorithm is provided with a mechanism to eliminate all overlaps from the final solution via a measure of the overlap in the cost function. In recent years a number of valid SA based macro-cell placement algorithms implementing analog-specific constraints have been developed based on Gelatt & Jepsen's placement style [67, 124, 174].

For reasons of improved flexibility, we feel that this concept is most suitable to effectively attack the analog placement problem, hence we have adopted it as a basis of our work as well [119, 175, 176].

4.3 Modifying Basic Algorithms

The enforcement of analog-specific constraints has been performed in PUPPY-A by introducing several heuristics which involve modifications to the the *configuration space* $\{S\}$, the *cost function* $f(S)$, and the *move-set* M . Let us discuss the basic algorithm in all its features and then describe each modification in details.

4.3.1 Standard Features

PUPPY-A is an interactive tool implementing a flat style SA based placement algorithm. The tool, based on a program originally designed for macro-cell digital design [170], is characterized by the following standard features.

The configuration space $\{S\}$ is fixed during the unfolding of the annealing, i.e. all the macro-cells originally designed for the chip will not be modified by the annealing. The routing space also stays unchanged at all temperatures and it is simulated by a rectangular halo around each macro-cell [180]. The thickness of the halo for a given cell side is logarithmically growing with the number of nets departing from that side¹² as shown in Figure 4.7.

The cost function $f(S)$ evaluates the impact of three factors: total chip area $f_A(S)$, total wiring $f_W(S)$ and cell-overlap $f_O(S)$. The term $f(S)$ is computed as

$$f(S) = \alpha_A f_A(S) + \alpha_W f_W(S) + \alpha_O f_O(S), \quad (4.8)$$

where the α_A , α_W and α_O are constant weights indicating the relative importance of the various components.

¹²The diagonals shown in Figure 4.7 determine the side from which each terminal is most likely to depart.

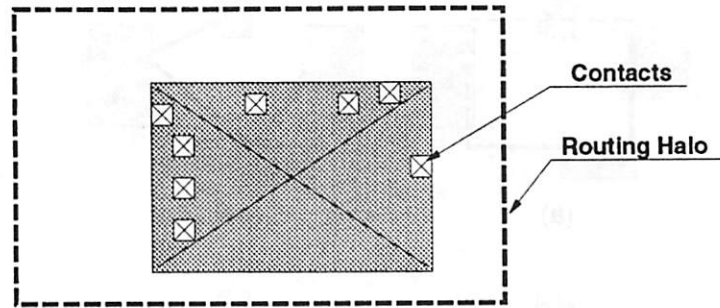


Figure 4.7: Accounting for routing channels: the Halo algorithm

The total chip area is computed as

$$f_A(S) = W(S) \times H(S),$$

where $W(S)$ and $H(S)$ are the width and height of the entire chip associated with configuration S . The values of $W(S)$ and $H(S)$ can be easily estimated using the information on the position of the left/rightmost and highest/lowest cells in the workspace.

The total wiring of the chip is estimated as

$$f_W(S) = \sum_{i=1}^{N_n} \bar{L}(n_i),$$

where $n_i, i = 1, \dots, N_n$ is a net and $\bar{L}(n_i), i = 1, \dots, N_n$ is the estimate of its wiring length. In PUPPY-A the term $\bar{L}(n_i)$ can be computed using two alternative methods: *semi-perimeter* and *non-minimum spanning tree*. The first method involves the computation of the semi-perimeter of the net bounding-box, i.e. the minimum-area box containing all terminals in the net. See Figure 4.8a. The complexity of the method is dominated by the bounding-box computation, which can be performed efficiently in $O(V)$, where V is the number of terminals for a given net [51, Chp. 2].

The second technique is a variation of the *minimum spanning tree* (MST) method, illustrated in Figure 4.8b. Let $G(V, E)$ be a fully connected graph or *clique*, where vertices V are the terminals of the net. The MST is the minimum length acyclic path of the clique connecting all V terminals. The complexity needed to find the MST is $O(V^2)$, since from each vertex of the graph every other not-previously-visited vertex must be searched [51, Chp. 2]. While solving the placement problem however, detailed routing is not known, hence relatively high inaccuracies are to be expected while estimating the wiring length,

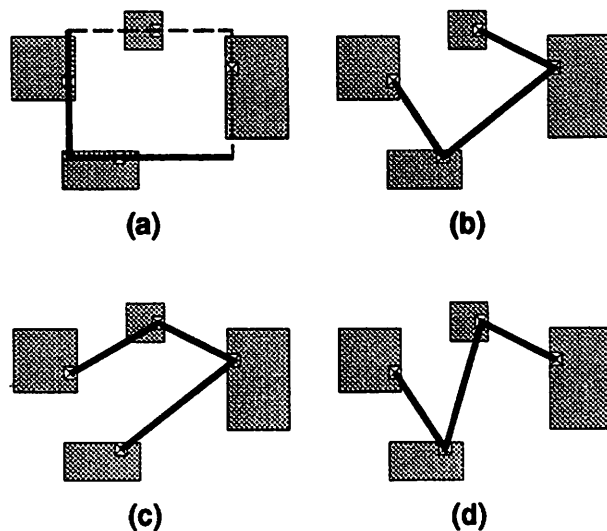


Figure 4.8: *Routing estimation techniques*

especially at high temperatures. Thus, MST can be replaced with a less accurate but more efficient method.

A good alternative is the calculation of random acyclic paths, whose length will always be greater or equal than that of the MST. Non-minimal acyclic paths (Figure 4.8c,d) are faster to build, since simple vertex enumeration, linear in V , is required. Furthermore, to improve the accuracy of the method, one could add the term `swap_segment` in the move-set, in such a way that a scheme such as that in Figure 4.8c could be replaced by the one in Figure 4.8d during the annealing. Thus, the task of reducing the path length, hence improving the accuracy of the estimation, can be left to the annealing itself. Since improved accuracy is needed particularly at low temperatures, when detailed placement decisions are to be made, the probability of selecting the appropriate moves could be further increased with the decay of the temperature.

The position of terminals in each module can be fixed, i.e. determined *a priori*, or dynamically changed during the annealing. In the latter case the cell is said *soft* and its terminals *floating*. Figure 4.9 shows a module with floating terminals and the moves being performed on the terminals.

Lastly, let us consider the overlap cost. Term $f_O(S)$ is computed as follows

$$f_O(S) = \sum_{j=1}^{N_O} \sum_{i=j+1}^{N_O} \text{Area}(O_j \cap O_i), \quad (4.9)$$

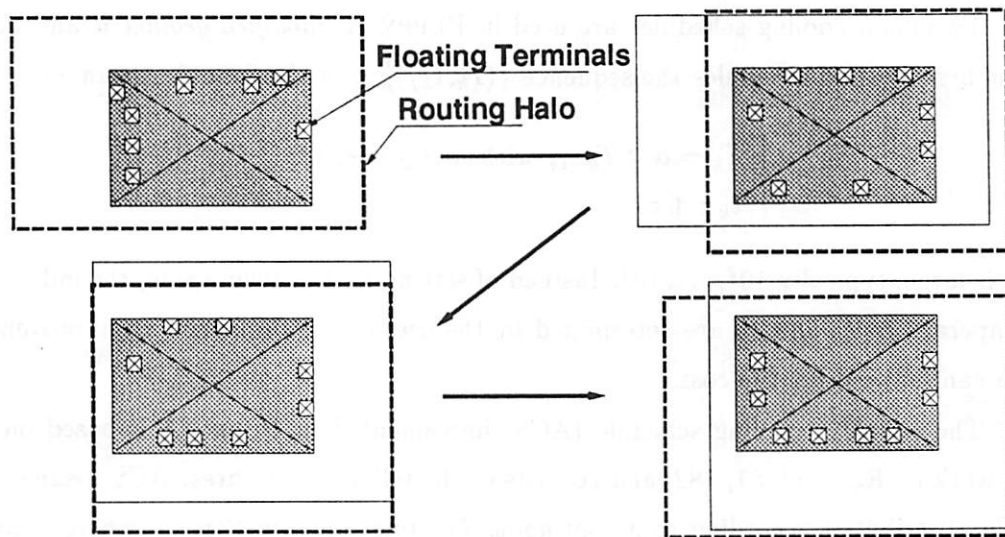


Figure 4.9: *Dynamically adjustable terminals in modules*

where the term $Area(O_j \cap O_i)$ indicates the area of the intersection of objects O_j and O_i . The process is relatively inefficient, requiring a complexity of $O(N_O^2)$.

In PUPPY-A, for efficiency reasons, each component of the cost function is computed *incrementally* during the annealing. Thus, only those components of the cost are recomputed which are modified by objects and nets directly involved in a move. As we will see, this cannot always be done when global changes occur in the workspace as e.g. in the case of substrate noise injection.

The move-set of PUPPY-A consists of

- TRANSLATE
- ROTATE
- TRANSLATE_AND_ROTATE
- SWAP
- NET_SEGMENT_SWAP

All moves except SWAP and NET_SEGMENT_SWAP involve a single object. The values that a rotation called by ROTATE can take are the same given in section 4.1.1. During the unfolding of the algorithm the probability of selection for the various move types changes and it is controlled by a mechanism aimed at maximizing those moves with the best impact on the cost [170].

Two main cooling schedules are used in PUPPY-A: *modified geometric* and *adaptive*. The first method schedules the sequence $\{(T_k, t_k)\}_{k=1}^{\mathcal{K}}$ in the following manner

$$\begin{cases} T_k = \alpha \times T_{k-1}, \text{ with } 0.9 \leq \alpha < 1.0 \\ t_k : \text{fix} \end{cases}$$

Term t_k is large, typically $10^2, \dots, 10^4$. Instead of setting \mathcal{K} to a given value, the initial and final temperatures T_0 and T_f are determined by the user or the prolonged no-improvement criterion can be used for the cost.

The adaptive cooling schedule (ACS) implemented in PUPPY-A is based on the original work by Romeo [181, 182] and consists of the following features. ACS assumes the cost to be distributed according to a continuous Gamma function $\Gamma(p, \alpha)$, where p and α are the shape and the scaling factors respectively. Hence, the expected value and variance of the cost can be expressed as a function of temperature T

$$E_T(f) = \frac{p}{\alpha + 1/T} + f^* ; \quad \sigma_T^2(f) = \frac{p}{(\alpha + 1/T)^2} , \quad (4.10)$$

where f^* is the minimum of cost f . Setting $T \rightarrow \infty$, one can find expressions for parameters $E_\infty(f)$, $\sigma_\infty^2(f)$, p and α . Moreover, equations (4.10) can be rewritten as

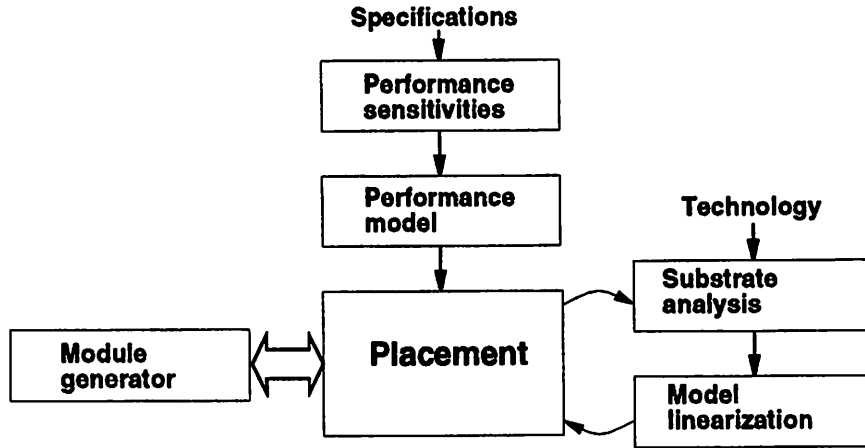
$$E_T(f) = E_\infty(f) - \frac{\sigma_\infty^2(f)}{T} \left(\frac{\alpha}{\alpha + 1/T} \right) ; \quad \sigma_T^2(f) = \sigma_\infty^2(f) \left(\frac{\alpha}{\alpha + 1/T} \right)^2 . \quad (4.11)$$

Using equation (4.11), the fact that $\alpha \gg 1/T$ and that $E_\infty(f) - E_T(f)$ is small if $\sigma_\infty^2(f) \ll T$ [181], the following criterion can be used

$$T_0 = K\sigma(f) ,$$

where K is a small positive number. The initial distribution parameters (p, α) are derived by collecting infinite temperature statistics and using (4.10) at $T \rightarrow \infty$. This distribution, along with equations (4.10), is then used to predict what the cost distribution at each temperature should be, assuming a Gamma distribution for the cost. This is done by adapting p and α at each temperature. One assumes that equilibrium is attained when the moving weighted average of mean cost $\overline{E_T(f)}$ lies within δ standard deviations $\sigma_T(f)$ from the mean $E_T(f)$ in the model of the cost distribution predicted at a given temperature, i.e.

$$|E_T(f) - \overline{E_T(f)}| < \delta\sigma_T(f),$$

Figure 4.10: *Placement flow diagram for PUPPY-A*

where δ is a non-negative user-defined real number. If for a given temperature no equilibrium is reached in at most $M < \infty$ moves, $E_T(f)$ and $\sigma_T(f)$ are set to the moving weighted averages $\overline{E_T(f)}$ and $\overline{\sigma_T(f)}$ as calculated at the current temperature. The next temperature is computed as

$$T_k = T_{k-1} e^{-\frac{T_{k-1}\lambda}{\overline{\sigma_{T_{k-1}}(f)}}},$$

where $\overline{\sigma_{T_{k-1}}(f)}$ is the moving average of standard deviation as calculated at temperature T_{k-1} and λ is the upper-bound to the distance between the stationary densities associated to temperatures T_k and T_{k-1} [181].

As we will see later, the cooling schedule can become an important factor in determining convergence for the SA algorithm.

4.3.2 Modifications of the Standard Features

A modification on the cooling schedule in PUPPY-A goes beyond the scopes of this dissertation. On the contrary, the remaining features characterizing the SA algorithm for the placement problem are an effective mechanism allowing us to adapt it to the analog placement problem. Figure 4.10 illustrates a modified version of the placement methodology for the use in analog IC design. High-level specifications are used, in combination with sensitivity analysis, to create a model for performance, which drives directly the modified cost function in the SA algorithm. Analog-specific constraints are enforced using a number of modifications in the cost function as well as in the move-set. Substrate is char-

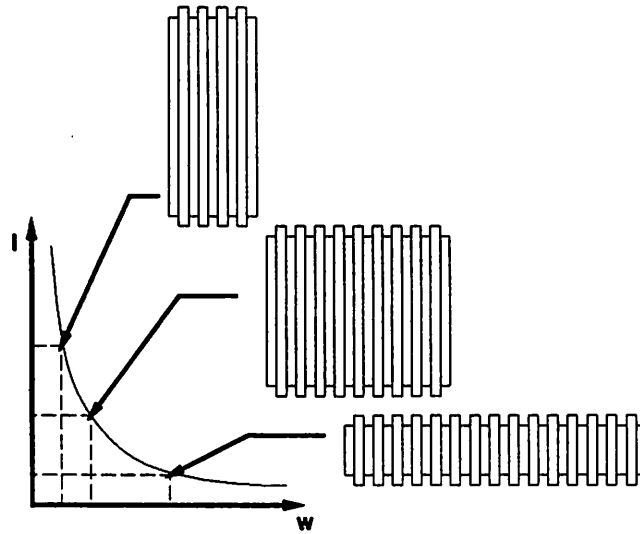


Figure 4.11: *Dynamically adjustable modules available to the placement tool*

acterized using an efficient Green's Function-based method and a number of heuristics for fast computation of the strength of currents transported through the substrate. As a final verification step, the layout is routed, fully extracted and simulated accounting for all relevant non-idealities. In case some specifications are not met, all corresponding sensitivities are weighted by a factor proportional to the severity of the violation, and the placement is repeated [119]. In what follows a discussion and justification of the various proposed modifications is presented.

4.3.3 Configuration Space

In the digital domain, a dynamically adjustable search space $\{S\}$ has been proposed as a key mechanism to improve the results of layout design [183, 184]. The first applications adopting such a strategy in the analog domain were proposed in [67] and [185], where adjustable modules enabled the placer to select from a larger set of possible realizations. Figure 4.11 shows a sample of module realizations for simple stacked transistors. Since the W/L ratio is fixed, length and width of the whole module is determined by the relation shown in the curve of Figure 4.11, which is used for the direct calculation of the module dimensions during the annealing.

An Alternative solution adopted by a number of systems, e.g. ILAC [57], SAM [124] and SALIM [92], consists of creating a library of relatively simple device realizations

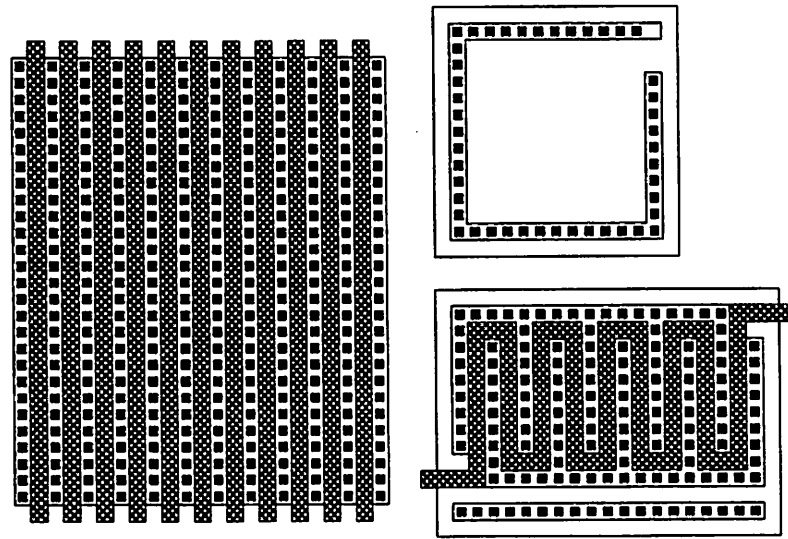


Figure 4.12: *Library of CMOS modules*

and selecting the most appropriate module, based on a set of rules. This method has the advantage of exploring a number of structures, electrically equivalent but which might improve the performance of the layout due, for example, to better matching or greater compactness. See Figure 4.12.

The problem with the first approach is that the number of alternative realizations can be very high, thus slowing down the annealing. Moreover, a relatively reduced variety of modules is available, thus preventing the algorithm from exploring configurations such as the *common-centroidal* and *interdigitated* structures. As a partial solution to this problem an additional modification to the move-set was proposed [67] which allowed the annealing to operate diffusion mergings or module abutments dynamically.

The second approach lacks of flexibility since a decision on the module realization must always be made *a priori* when the details of the placement and routing are not yet known. Furthermore, every module is generated separately, hence no or limited trade-offs involving a higher number of modules are possible and may not be supported by additional knowledge about the global appearance of the layout.

The first attempt to address these problems was proposed by us in [175, 176]. In this approach simultaneous placement and module optimization is used as an effective way to insure maximal flexibility during the placement phase, while drastically reducing the search space for all possible module implementations. First, the composite stack-module

generator LDO partitions the circuit and finds different alternative sets of modules. Each alternative solution is chosen so as to minimize a cost function accounting for all analog-specific constraints. Routability and interconnect parasitics cannot be taken into account at this stage since no information on the reciprocal position of the modules is known.

Next, all the equivalent solutions are made available to PUPPY-A. The set of moves of the annealing algorithm has been extended to include not only geometric perturbations, but also swaps between alternative solutions. In this way, placement and module optimization are performed simultaneously. The set of modules available to the placement is relatively small, since the configurations yielding large performance degradation have already been discarded. Hence, negligible computational overhead is needed with respect to standard placement with a pre-defined set. Section 4.4 discusses generation and module-interchange techniques in detail.

4.3.4 Cost Function

In PUPPY-A the cost function has been used for both minimization and constraint enforcement purposes. Let $f_{EC}(S)$ be the cost function associated with the constraint enforcement and let $f_{MIN}(S)$ be the objective of various measures being minimized. Then, the general cost for SA is defined as

$$f(S) = f_{EC}(S) + f_{MIN}(S) . \quad (4.12)$$

The term $f_{MIN}(S)$, the measure for standard minimization features, such as area, and total wiring, is defined as

$$f_{MIN}(S) = \alpha_A f_A(S) + \alpha_W f_W(S) , \quad (4.13)$$

where $f_A(S)$, $f_W(S)$, α_A and α_W have been defined in (4.8).

The term $f_{EC}(S)$, relating to constraint violations of the placement, is defined as

$$f_{EC}(S) = \alpha_O f_O(S) + \alpha_{WE} f_{WE}(S) + \alpha_S f_S(S) + \alpha_M f_M(S) + \alpha_P f_P(S) , \quad (4.14)$$

where α_O , α_{WE} , α_S , α_M , and α_P are non-negative weights. Each individual cost function is described hereafter.

$f_O(S)$: this function relates to the total overlap between all modules. The function is defined in equation (4.9). Efficient techniques for its calculation are discussed in [170].

$f_{WE}(S)$: this function relates to the cumulative set of translations needed to assign each module to its region of compatibility. For simplicity but with no loss of generality

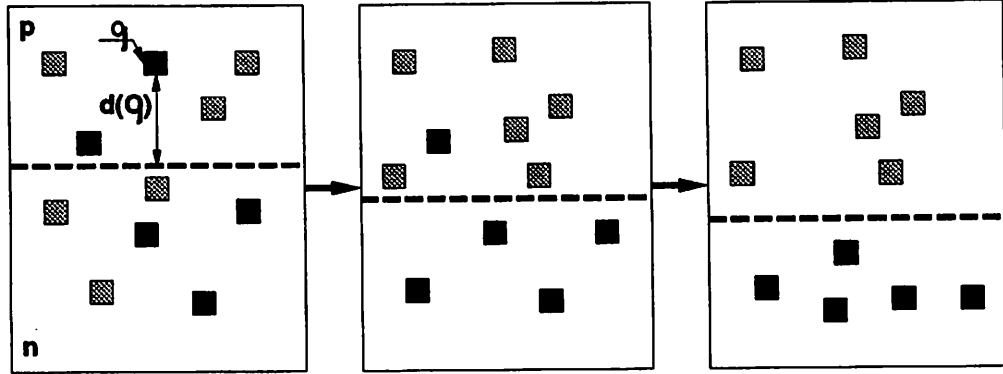


Figure 4.13: SA and well definition

consider the case of a CMOS circuit. Let \mathcal{P} , \mathcal{N} and \mathcal{I} be the sets of N-, P- and no-type¹³ modules. Modules in \mathcal{P} (\mathcal{N}) are compatible with a N(P)-type well, while modules in \mathcal{I} can be put over both well regions. Cost $f_{WE}(S)$ is defined as the sum of all translations needed to bring each module in the appropriate region

$$f_{WE}(S) = \sum_{O_j \in \mathcal{N}} d_{\mathcal{N}}(O_j) + \sum_{O_j \in \mathcal{P}} d_{\mathcal{P}}(O_j), \quad (4.15)$$

where function $d_{\mathcal{N}}(O_j)$ ($d_{\mathcal{P}}(O_j)$) is the Manhattan distance between object O_j of type N(P) and the closest edge of the well region compatible with it. If O_j is within a compatible region, then the associated cost is zero. See Figure 4.13. These regions represent the future geometric realization of the wells. Figure 4.13 shows the evolution of S towards a configuration in which all the objects are placed in two distinct well regions. Suppose now that more than one well region exists for each device type. Let us define function $d_{\mathcal{N}}(O_j, WR_i)$ ($d_{\mathcal{P}}(O_j, WR_i)$) as the Manhattan distance between object O_j and the closest edge of compatible region WR_i . Then, the minimum distance $d_{\mathcal{N}}^{(min)}(O_j)$ is defined as

$$d_{\mathcal{N}}^{(min)}(O_j) = \min_{i=1, \dots, N_{\mathcal{P}}} d_{\mathcal{N}}(O_j, WR_i), \quad \forall O_j \in \mathcal{N}$$

where $N_{\mathcal{P}}$ ($N_{\mathcal{N}}$) is the number of existing well P(N)-regions. $d_{\mathcal{P}}(O_j)$ is defined similarly.

Since it is rectangular, a region WR_i is characterized as an object, i.e. using the center $s_0(WR_i)$, its width $W(WR_i)$ and its height $H(WR_i)$. During the unfolding of the

¹³This might be the case of composite modules, i.e. small sub-circuits implementing both types, or poly resistors, which do not have restrictions on the underlying well.

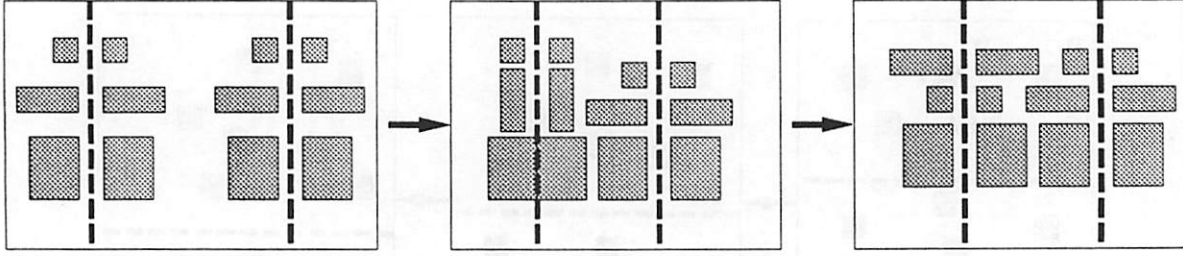


Figure 4.14: *Placement using virtual symmetry axes*

SA , $s_0(WR_i)$, $W(WR_i)$ and $H(WR_i)$ can be changed using standard moves. The same overlapping constraints can be used and abutment can be performed on them [119].

$f_S(S)$: the function relates to the cumulative set of translations and rotations needed to enforce all symmetry constraints. This cost, derived from (4.6), is defined as

$$f_S(S) = \sum_{m=1}^{N_S} \sum_{i,j \in \mathcal{S}_m} d_{\mathcal{S}_m}(O_i, O_j) + r_{\mathcal{S}_m}(O_i, O_j) ,$$

where $\mathcal{S}_m, \forall m = 1, \dots, N_S$ is a set of all tuples of objects bound by a symmetry constraint with respect to axis A_m . Terms $d_{\mathcal{S}_m}(O_i, O_j)$ and $r_{\mathcal{S}_m}(O_i, O_j)$, the translation to make (O_i, O_j) tuple symmetric with respect to A_m , are defined as

$$\left. \begin{aligned} d_{\mathcal{S}_m}(O_i, O_j) &= |y_0(O_j) - y_0(O_i)| + |x_0(O_j) - 2x(A_m) + x_0(O_i)|, \\ r_{\mathcal{S}_m}(O_i, O_j) &= \begin{cases} \alpha_r, & \text{if } r(O_j) \neq \text{mirror_symmetric}[r(O_i)] \\ 0, & \text{otherwise} \end{cases} \end{aligned} \right\} \text{if } A_m \text{ vertical axis,}$$

$$\left. \begin{aligned} d_{\mathcal{S}_m}(O_i, O_j) &= |x_0(O_j) - x_0(O_i)| + |y_0(O_j) - 2y(A_m) + y_0(O_i)|, \\ r_{\mathcal{S}_m}(O_i, O_j) &= \begin{cases} \alpha_r, & \text{if } r(O_j) \neq \text{mirror_symmetric}[r(O_i)] \\ 0, & \text{otherwise} \end{cases} \end{aligned} \right\} \text{if } A_m \text{ horizontal axis,}$$

where $x(A_m)$ ($y(A_m)$) is the position of vertical (horizontal) axis A_m and α_r is a non-negative constant. If $x(A_m)$ ($y(A_m)$) are let vary during the annealing, one can obtain better levels of global optimization and possibly more compact layout. The approach, discussed in [186], is illustrated in Figure 4.14.

$f_M(S)$: The cost due to mismatch is characterized as follows from equation (4.7)

$$f_M(S) = \sum_{m=1}^{N_M} \sum_{i,j \in \mathcal{M}_m} d_{\mathcal{M}_m}(O_i, O_j) + r_{\mathcal{M}_m}(O_i, O_j) ,$$

where $\mathcal{M}_m, \forall m = 1, \dots, N_{\mathcal{M}}$ is a matching cluster, i.e. a set of all objects bound by certain matching constraints. Terms $d_{\mathcal{M}_m}(O_i, O_j)$ and $r_{\mathcal{M}_m}(O_i, O_j)$, the translation to make (O_i, O_j) tuple matched according to the definition of equation (4.7), is defined as

$$\begin{aligned} d_{\mathcal{M}_m}(O_i, O_j) &= \max \left\{ |\mathbf{s}_0(O_j) - \mathbf{s}_0(O_i)| - d_{\mathcal{M}_m}^{(max)}, 0 \right\}, \\ r_{\mathcal{M}_m}(O_i, O_j) &= r_0(O_j) - r_0(O_i) - r_{\mathcal{M}_m}^{(max)}, \end{aligned}$$

where $\mathbf{s}_0(O_j)$ is the center of object O_j and $r_0(O_j)$ is its orientation. Terms $d_{\mathcal{M}_m}^{(max)}$ and $r_{\mathcal{M}_m}^{(max)}$, called *cluster diameter* and *cluster rotation factor* respectively, represent the maximum distance and relative orientation between any two objects within a cluster. Values for $d_{\mathcal{M}_m}^{(max)}$ and $r_{\mathcal{M}_m}^{(max)}$ are either calculated using the techniques presented in chapter 3 or imposed by the designer.

$f_P(S)$: this term relates to the cumulative effect of all parasitics on performance. Since performance generally consists of a number of measures, e.g. phase margin, low frequency gain and activation delay, a weight must be given to the violation associated with each performance component $K_i, \forall i = 1, \dots, N_k$. Let us define the set of weights $\alpha_i, \forall i, \dots, N_k$ as the *relative importance of each violation*, then term $f_P(S)$ can be interpreted as the weighted sum of all performance violations and is defined as

$$f_P(S) = \sum_{i=1}^{N_k} \alpha_i f_{i,P}(S),$$

where term $f_{i,P}(S), \forall i = 1, \dots, N_k$ is the violation of the specifications imposed on performance measure K_i . When no flexibility is allowed on the realization of the layout, $f_{i,P}(S)$ is defined as

$$f_{i,P}(S) = F \left(\max \{ \Delta K_i^+ - \overline{\Delta K_i^+}, 0 \} \right) + F \left(\max \{ \Delta K_i^- - \overline{\Delta K_i^-}, 0 \} \right),$$

where $\overline{\Delta K_i^\pm}$ is the *specification* on the positive/negative degradation of performance K_i and ΔK_i^\pm is a *model* for it. Function $F(\cdot)$ is used to weight the cost function differently according to the severity of each violation. In section 4.5 we will discuss how $F(\cdot)$ can be used to account for flexibility in the realization of the interconnect.

4.3.5 Move-Set

The original move-set in PUPPY-A was modified to include a number of techniques used by the analog placement tool. The new moves are

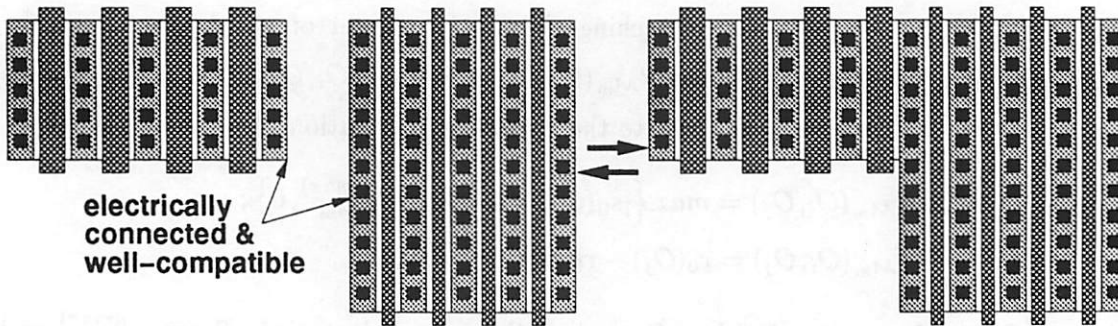


Figure 4.15: *Abutment and separation of modules*

- abut_cells
- separate_cells
- swap_alternatives
- update_axis
- update_wells
- update_well_contacts

To allow existing modules to freely abut during the unfolding of the algorithm, a new move called `abut_cells` was created. The move causes two cells which belong to compatible wells to fuse into one. The analogous move `separate_cells` is exercised **only** on those cells previously abutted. See Figure 4.15.

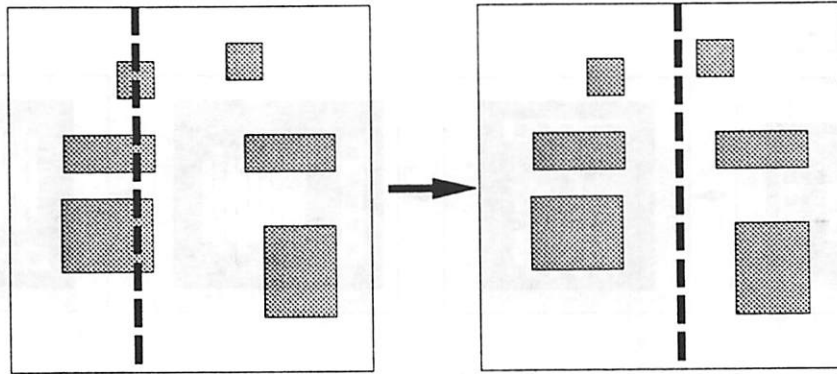
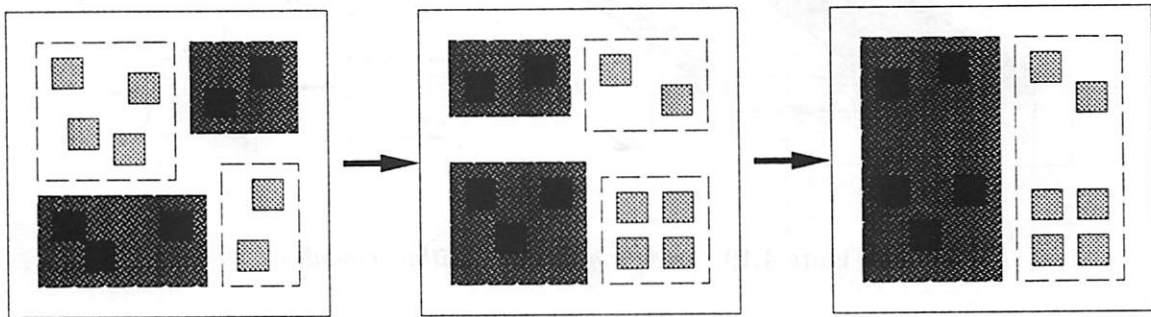
In our approach, module alternatives are generated *a priori* and the solutions which represents the best trade-offs are selected during the annealing. In order for our SA based placement engine to explore all the options at its disposal, the move-set has been expanded to include a number of new items. The move `swap_alternatives` allows the annealing to replace a module alternative realization, thus in effect moving within $\{S\}$ towards new search regions.

A move called `update_axis` has been added to allow a translation of arbitrary symmetry axis A_m . This move is used for updating the position of an existing axis either to a new random location or to that which minimized the symmetry violations related to it. In the latter case, the new location of vertical¹⁴ axis A_m is computed as follows

$$x(A_m) = \frac{1}{|\mathcal{S}_m|} \sum_{j \in \mathcal{S}_m} \mathbf{s}_0(O_j)$$

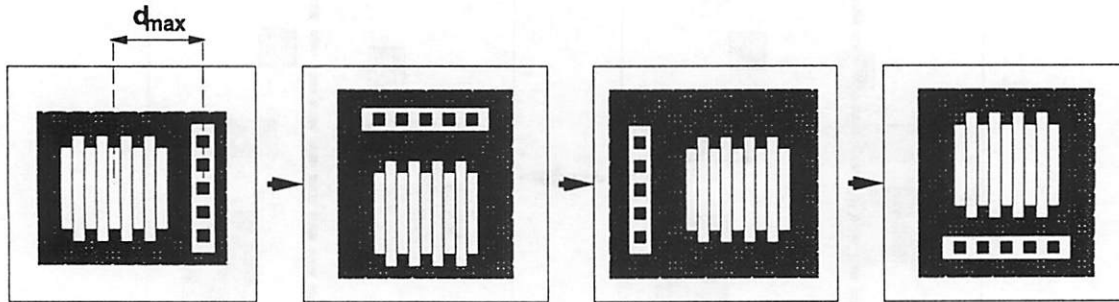
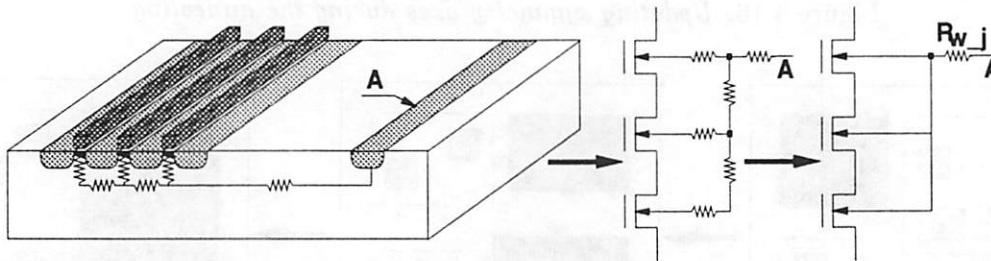
where $\mathbf{s}_0(O_j) = [x_0, y_0]^T$ is the vector of the center of object j and $|\mathcal{S}_m|$ is the cardinality of \mathcal{S}_m . See Figure 4.16.

¹⁴The same argument can be used for horizontal axes.

Figure 4.16: *Updating symmetry axes during the annealing*Figure 4.17: *Updating well regions during the annealing*

A move called `update_wells` is used for the recalculation of parameters s_0 , H and W for each well region $WR_i, \forall i = 1, \dots, N_P(N_M)$. The objects can be viewed as regular objects, hence the same overlap rules apply but only to objects of different type. Abutment and separation can be performed on each pair of wells of the same type. Figure 4.17 shows the progressive modification of the regions and an abutment procedure.

Finally, a move called `update_well_contacts` is used for the update of well contacts in each module. The mechanism is similar to the principle of the *floating terminals*. Figure 4.18 shows a well contact moving within the region of the well. The term d_{max} determines the maximum distance between devices in the module and well contact in order to avoid excessive performance degradation or latch-up. A value for d_{max} can be user-enforced [142] or it can be computed using sensitivity analysis and a local substrate simulation, as following. Let $S_{i,R_{w-j}}$ be the sensitivity of performance K_i with respect to R_{w-j} . Term R_{w-j} models the substrate resistance between the channel of each device in the module and the

Figure 4.18: *Updating contact locations*Figure 4.19: *Modeling contacts within modules*

node which is connected to the bias bus line through the contact. See Figure 4.19. Then, using the approach of chapter 3, one can derive a bound $R_{w-j}^{(bound)}$ for R_{w-j} associated with module O_j . Since the geometric structure of module O_j is known, ignoring all surrounding objects, any substrate resistance evaluator can be used to extract R_{w-j} for various positions of the contact relative to the module as shown in Figure 4.18. In our approach the Green's Function-based package SUBRES (See chapter 8) was used for the characterization. Plot 4.20 shows the value of R_{w-j} as a function of the relative position of a simple one-via contact in a typical device configuration. From the plot, one can define a *feasibility region* in which the contact should be placed for the bound inequality

$$R_{w-j} \leq R_{w-j}^{(bound)}$$

to be satisfied. This determines the region within which the contact is allowed to float during the unfolding of the SA algorithm as illustrated in Figure 4.21. Alternatively, one can add the product $S_{i,R_{w-j}} R_{w-j}$ to the estimate of performance degradation ΔK_i , thus allowing the placer to take the effects of far biasing contacts directly into account. Both these approaches are available in PUPPY-A.

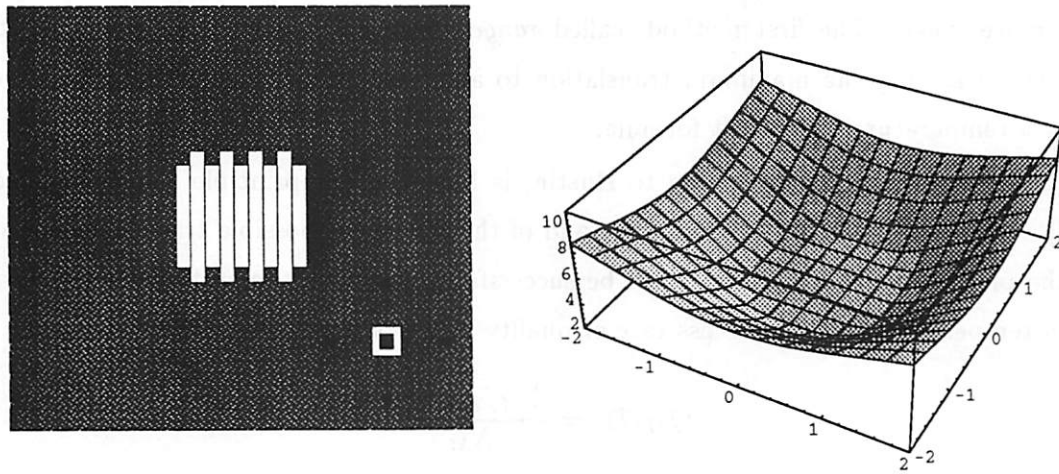


Figure 4.20: *Contact resistance as a function of relative position within a module*

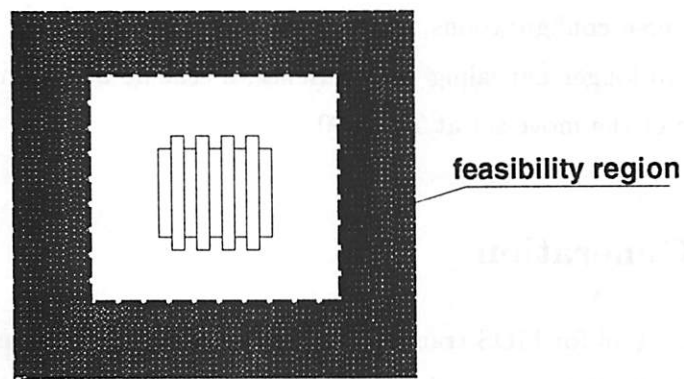


Figure 4.21: *Derivation of feasibility region for the contact realization*

Translation moves for both cells and terminals require the definition of a *move range*, which relates to the maximum possible distance at which a selected module should be randomly placed by the algorithm. There are at least two approaches to the selection of a move range. The first method, called *range limiting* is due to Kirkpatrick [66] and consists of setting the maximum translation to a pre-determined value which is reduced using a temperature-dependent formula.

The second method, due to Hustin, is based on the principle of *quality factors* [187]. The selection of the type of move and of the maximum possible translation is based on the probability that the move will be successful in reducing the cost function¹⁵. For a given temperature T_k , the success rate or quality factor of move M is determined as

$$Q_M(T_k) = \frac{\Delta_M f_{T_{k-1}}(S)}{N_M},$$

where $\Delta_M f_{T_{k-1}}(S)$ is the total cost reduction obtained at a previous temperature due to move M and N_M is the total number of accepted moves of type M .

An estimate of the probability of success $Pr_M(T_k)$ for move M at temperature T_k can be computed as

$$Pr_M(T_k) = \frac{Q_M(T_k)}{\sum_{m=1}^{M_M} Q_m(T_k)},$$

where $Q_m(T_k)$ is the quality factor of move m , $\forall m = 1, \dots, M_M$ and M_M is the total number of moves in the move-set. At high temperatures a large move-set is preferable to explore a large number of diverse configurations. At lower temperatures, i.e. $T_k \rightarrow 0$, a large move-set translates into longer annealing times. Hence, a scheme based on quality factors tends to limit the size of the move-set at $T_k \rightarrow 0$.

4.4 Module Generation

LDO [188] is a tool for MOS transistor composite stack generation. Its purpose is to generate a set of stacks containing all the transistors of the circuit, split into modules abutted with each other. Source/drain regions are shared between adjacent elements, in such a way that area and critical capacitances are minimized.

¹⁵Notice that the latter scheme is more general than the former in that it can be applied to the entire move-set whether or not a translation is involved.

4.4.1 Terminology

Let $G(E, V)$ be an undirected graph, with a set of vertices V and a set of edges E . A *path* or *chain* p from vertex v_0 to vertex v_k in G is defined as the sequence of vertices $v_0, v_1, v_2, \dots, v_k \in V$ and edges $e_1, e_2, \dots, e_k \in E$ linking e_i , via v_{i-1} and v_i , and $e_i \neq e_j, \forall i, j, i \neq j$ [189, Ch.5]. Vertices v_0 and v_k are called *endpoints* of p . The *length* of a path is the number of edges it contains. An *n-path* is a path whose length is n .

A *connected graph* is a graph $G(V, E)$ where for all $v, w \in V$ there is a path in G whose endpoints are v, w . A *complete component* C is a sub-graph of G where all pairs of vertices are adjacent. C is *maximal* when its size cannot be increased, i.e. when for each $v \in V$, either v is already in C , or v is not adjacent to some vertex of C . A maximal complete component is called *clique*.

Let $G(E, V)$ be a graph representing the circuit, where all vertices in V are nets in the circuit and for each MOS device there exist an edge in E , linking the vertices associated to the nets connected to the device terminals. A stack of n transistors in the circuit can be created if the corresponding edges in G form an n -path, as illustrated in Figure 4.22a. The device junction regions corresponding to the path endpoints are said to be *external*, while the other points in the path are called *internal*. Each full-stacked layout implementation corresponds to a *path partition* of $G(E, V)$, also defined as a set \mathcal{P} of paths such that:

$$p \cap q = \emptyset, \quad \forall p, q \in \mathcal{P} \quad (4.16)$$

$$\bigcup_{p \in \mathcal{P}} p = E. \quad (4.17)$$

Operators \cap and \cup are applied to the edge sets of the paths. Condition (4.16) is the *non-overlapping* condition that no two stacks in the layout contain the same transistor. Condition (4.17) is the *covering* condition, that each transistor must appear in a stack. In each circuit at least one trivial partition \mathcal{P}_0 exists, in which each path has exactly one edge. Such partition corresponds to separate modules and it is often the starting configuration for placement tools with automatic abutment capability such as our approach [119].

4.4.2 Stack-Generation Algorithm

Hereafter is a description of the algorithm for stack generation.

1. The circuit is mapped to graph $G(V, E)$, which is split into two or more sub-graphs $G_i, i = 1, \dots, k$ according to the well type.

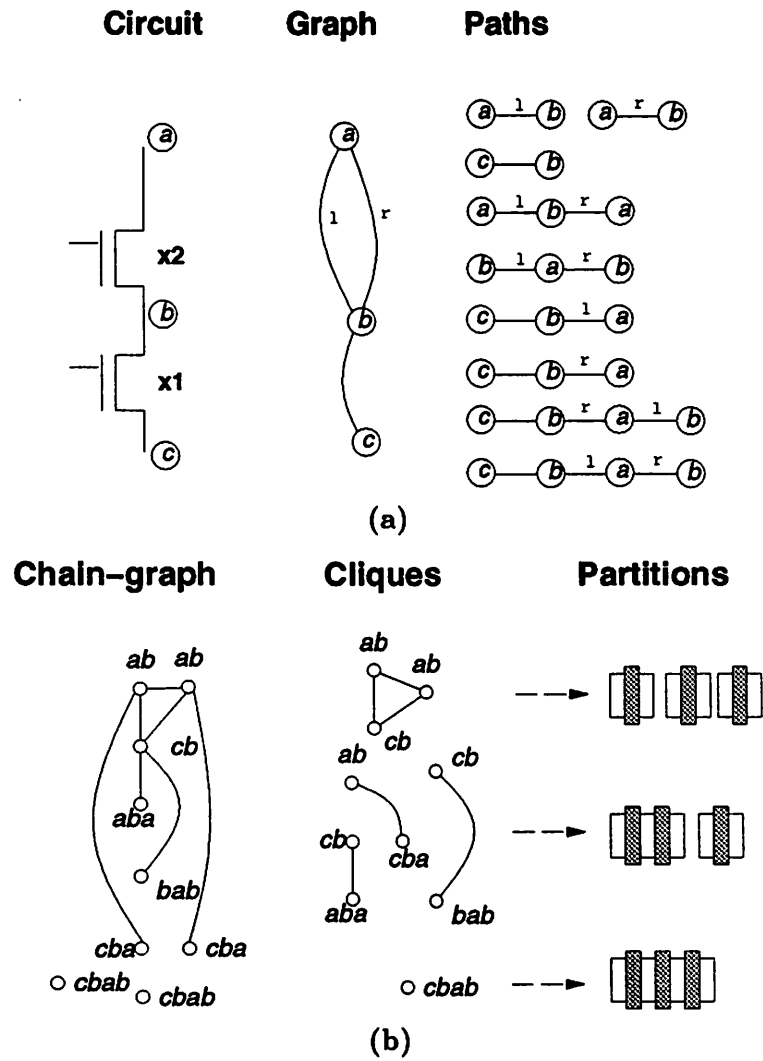


Figure 4.22: (a) Mapping of a circuit schematic onto a graph; (b) Chaining algorithm in LDO (Courtesy of Enrico Malvasi)

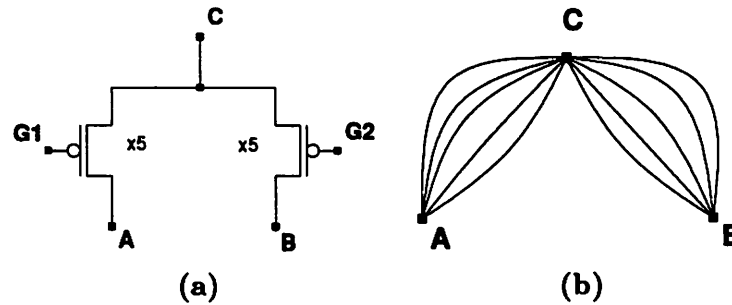


Figure 4.23: *Splitting large transistors: (a) differential pair; (b) equivalent graph* (Courtesy of Enrico Malavasi)

2. Large transistors are split into smaller modules connected in parallel.
3. Each sub-graph is split into smaller connected sub-graphs, containing only edges corresponding to modules with the same channel width.
4. Optimum path partition is carried out on each sub-graph independently.

In step 1 the circuit is mapped onto the corresponding graph as described above. All sub-graphs are generated based on the type of well to which the devices are connected. This is the first simplification made to the problem, which reduces its size significantly, with no modification of the solution set.

Step 2 consists of two phases. The first phase enforces explicit requirements of the user on the number of sub-divisions in devices. During the second phase, matched transistors are split into modules with the same width. Matching is improved by enforcing the same fringe effects on all modules. The computed width is the Greatest Common Divisor (GCD) of the widths of all matched transistors. Design-rules determine a lower bound to the width, while splitting is not applied if the GCD is smaller than this bound. After a transistor is split, the new modules become distinct matched devices, each introducing a different edge into the circuit graph and the new modules replace split transistors. Figure 4.23 shows an example of splitting in a simple differential pair.

After step 3 in each sub-graph only transistors with the same channel width are present. Stacks containing modules with different widths can still be built with the automatic abutment procedure in the placement phase as proposed in [67, 119]. All sub-graphs are disjoint with the exception of the ones which include devices related by symmetry and/or matching constraints. Such sub-graphs cannot be processed independently, therefore they are merged to form larger non-connected sub-graphs. Let $G_1(E_1, V_1)$, $G_2(E_2, V_2)$

be two such interdependent sub-graphs. They are merged into a non-connected sub-graph $G'(E', V')$, where $E' = E_1 \cup E_2$, and $V' = V_1 \cup V_2$.

In step 4, path partitioning, based on a two-phase algorithm, is performed as shown in Figure 4.22a,b. All existing paths in $G(V, E)$ are generated using a dynamic programming procedure. The problem of finding a path partition is transformed into a *clique* problem [190, p.194], [147]. Each path is associated to a vertex of a *chain-graph* G_c , whose edges link two vertices if and only if the corresponding paths are *mutually compatible*, i.e. they can coexist in the same partition. The non-overlapping condition (4.16) is necessary for mutual compatibility, hence all partitions are in G_c complete, and thus maximal, components [147]. Hence, each partition is a clique in G_c .

The coverage condition (4.17) is checked on all found cliques to determine whether they constitute a partition, otherwise they are discarded. A cost function is then used to evaluate the advantage of accepting each clique as a solution. The cheapest clique(s) is (are) the optimum solution to the partitioning problem. The algorithm described in [191] for CMOS logic cells is the basis of the approach followed in step 4. However, due to lack of flexibility to deal with analog-specific constraints, this algorithm cannot be used without appropriate modifications.

Moreover, the algorithm is computationally inefficient when applied to graphs where the number of edges is large compared to the number of vertices. In this case, the number of paths generated by the dynamic programming procedure grows almost factorially with the number of edges. The original algorithm has been modified to account for analog specifications and symmetries and to deal more efficiently with circuit graphs with a large number of edges. A cost function has been introduced to choose among different solutions the ones minimizing critical parasitics.

4.4.3 Analog Constraints and Computational Cost

The cost function exploits the fact that the junction capacitance of diffusion regions located in external positions of a stack is generally larger than that of internal regions. Capacitance can be minimized in critical nets by penalizing the nets located at the ends of a stack. The cost associated to a stack p is:

$$F(p) = \sum_i \text{cap}(n_i) \cdot \text{crit}(n_i),$$

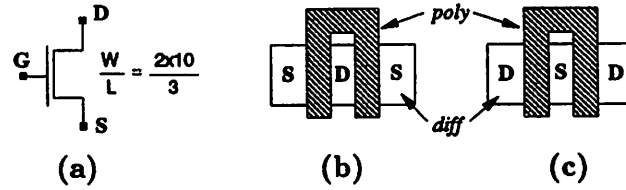


Figure 4.24: (a) Transistor split in two modules; (b) Layout minimizing the capacitance of net D ; (c) Layout minimizing the capacitance of net S

where the sum is extended to all the nets n_i connected to the source/drain regions of stack p . $\text{Cap}(n_i)$ is the junction capacitance of net n_i , while $\text{crit}(n_i)$ is its criticality weight defined on the ground of the performance sensitivity with respect to this capacitance. As an example, consider the 2-module transistor shown in Figure 4.24a and its two implementations 4.24b and 4.24c. If the capacitance on net S is more critical than that on net D , the cost of solution 4.24c is lower than that of solution 4.24b.

Let us turn our attention to capacitance estimator $\text{cap}(n_i)$ and criticality weight $\text{crit}(n_i)$. Suppose a number constraints on critical net capacitances $C^{(\text{bound})}(n_j), \forall j = 1, \dots, N_p$ have been computed using the techniques discussed in chapter 3. These bounds are used to define a set of normalized *criticality weights* for stray capacitances as

$$w(n_j) = \left[\frac{C^{\text{max}}(n_j) - C^{(\text{bound})}(n_j)}{C^{\text{max}}(n_j) - C^{\text{min}}(n_j)} \right]^2,$$

where $C^{\text{max}}(n_j)$ and $C^{\text{min}}(n_j)$ are the realizability limits for capacitive net n_j .

Let s_i be a stack module containing M_i devices of width W_i . Its $(M_i + 1)$ S/D regions are connected to $(M_i + 1)$ nets $n_j, j = 0, \dots, M_i$. Nets n_0 and n_{M_i} are in external positions. If $M_i > 1$, there are $(M_i - 1)$ other nets in internal positions. Let $F(s_i)$

$$F(s_i) = \sum_{j=0}^{M_i} k_j(W_i) \cdot w(n_j) \quad (4.18)$$

be the cost for stack s_i . Term k_j is called *position weight* and is defined as

$$k_j(W_i) = \begin{cases} C_{\text{ext}}(W_i)/C_{\text{int}}(W_i) & \text{if } j = 0 \text{ or } j = M_i \\ 1 & \text{otherwise} \end{cases}$$

where $C_{\text{int}}(W_i)$ and $C_{\text{ext}}(W_i)$ are the junction capacitance in internal and external positions respectively. The position weights account for the different junction capacitances due to net positions in the stack. The cost of a partition \mathcal{P} where all nets n_j reside is defined as

the sum of the costs associated with its stacks

$$F(\mathcal{P}) = \sum_{s_i \in \mathcal{P}} F(s_i)$$

This formulation of the cost function implicitly accounts for area minimization. In fact, suppose the criticality weights were all approximately the same. Then the partitions made of long stacks would be cheaper than partitions made of shorter ones, because in the former case both the total number of S/D regions and the number of S/D regions in external positions would be smaller. In general, the cost function is cheaper for partitions made of long stacks, with most critical nets located in internal positions.

Symmetry constraints are effective in decreasing the computational cost of the algorithm. As soon as a path is found in step 4 of the algorithm, it is checked against symmetry constraints, and discarded if they are violated. The size of the clique problem is reduced accordingly, along with the overall CPU cost [147]. Often, alternative solutions are equivalent in terms of their cost, though showing advantages under different considerations. A typical example is the circuit shown in Figure 4.25. All solutions, 4.25a, 4.25b and 4.25c are symmetrical. However, solution 4.25a optimizes matching with a common-centroid pattern, while solution 4.25b minimizes the amount of external routing between S/D regions and gates. Solution 4.25c is a trade-off, since interdigitated structures are used to some degree in order to improve matching at minimum expense of wiring. These trade-offs in physical realizations cannot be evaluated at this point, hence the decision on which alternative is postponed to when placement will be performed.

Matching is accounted for by providing proper splitting of the transistors into modules with the same channel width and by abutting matched devices into the same stacks when possible. Matching can also be improved by selecting the configurations with maximum device interleaving, and common-centroid patterns are always found when they exist.

4.4.4 Importance of Creating Alternative Modules

The tool LDO returns a number of solutions which are equivalent in terms of the associated cost but not in terms of the possible improvements in performance. Trade-offs can be done only during the placement phase, since a global view of the workspace is possible only at that point. In our approach we have proposed the use of move `swap_alternatives` which allows SA to choose a random alternative to a given module.

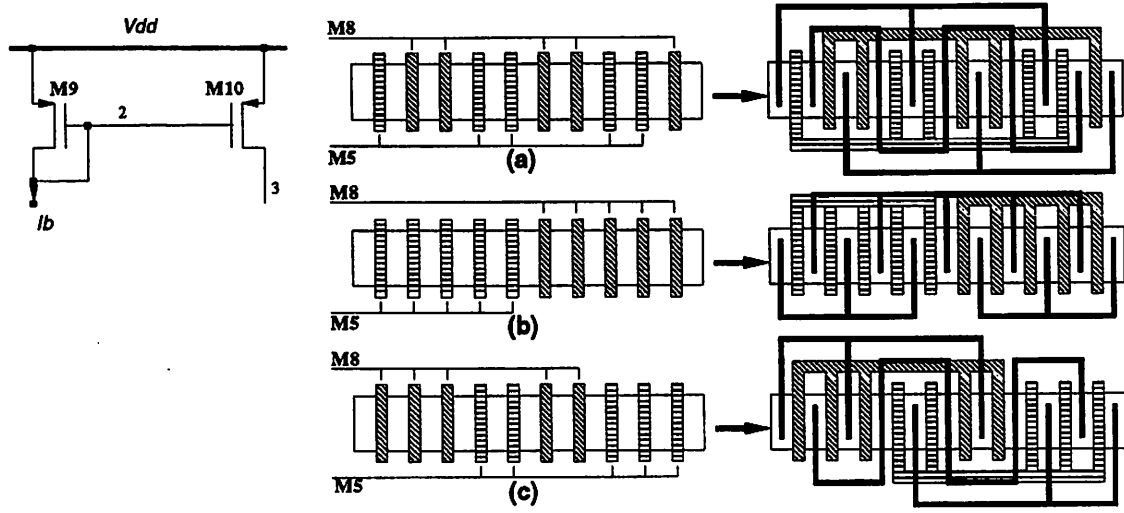


Figure 4.25: *Enforcement of symmetry in LDO: (a) first alternative based on a common-centroid design style; (c) second fully symmetric layout; (b) trade-off*

To illustrate the advantages of considering alternative module implementations during the placement, let us analyze the simple circuit shown in Figure 4.26. For simplicity, assume all transistors are equal in size. Moreover, assume that transistors ($M1, M2$), ($M3, M4$), ($M5, M6$), and ($M7, M8$) require (a) to be matched devices and (b) to be placed symmetrically with respect to a vertical axis. In addition, assume (c) that $M9$ and $M10$ also be a pair of matched devices.

This circuit can be partitioned in two sub-circuits according to the polarity of its devices¹⁶. If the transistors are implemented in a “full-stacked” design style, a possible solution for each sub-circuit could be the following scheme: ($M9, \frac{M10}{4}, \frac{M1}{2}, \frac{M1}{2}, \frac{M10}{4}, \frac{M7}{2}, \frac{M7}{2}, \frac{M8}{2}, \frac{M8}{2}, \frac{M10}{4}, \frac{M2}{2}, \frac{M2}{2}, \frac{M10}{4}$) for sub-circuit I, and ($M5, M3, M4, M6$) for sub-circuit II. The notation $\frac{Mxx}{n}$ indicates one of the n modules of width w/n into which transistor Mxx , of width w , is split. This scheme is desirable in terms of symmetry and matching constraints and it is acceptable in terms of area, since only two stacks implement the entire circuit. However, it has several drawbacks. Firstly, this solution does not allow any interleaving among transistors $M1$ and $M2$. This might result in worse offset and noise performance in presence of even modest technology gradients. Secondly, due to the size of the stack implementing sub-circuit I, nets 5 and 6 might be long and therefore involve stray resistances, large capacitances to ground and cross-coupling capacitances. This might

¹⁶The bubbles in Figure 4.26 represent different well types.

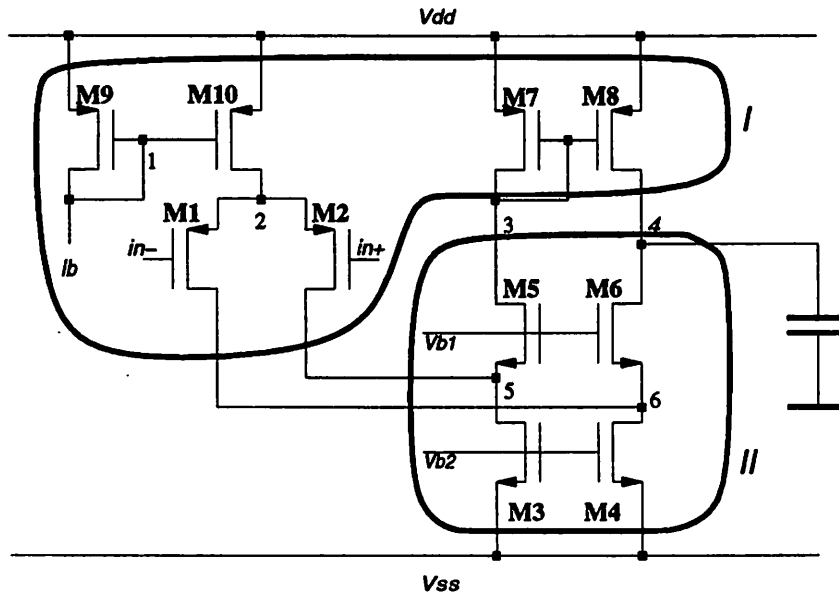


Figure 4.26: Example of a folded cascode opamp. The bubbles represent all sub-circuits created by the module generator on the ground of well type

result in poor bandwidth, due to the high criticality of these nets.

An alternative scheme for sub-circuit I, which could alleviate most of the above problems, would be to distribute all devices in three stacks: $(\frac{M9}{2}, \frac{M10}{8}, \frac{M9}{2}, \frac{M10}{2})$, $(\frac{M2}{2}, \frac{M1}{2}, \frac{M2}{2}, \frac{M1}{2})$ and $(\frac{M7}{2}, \frac{M1}{8}, \frac{M7}{2}, \frac{M8}{2})$. This implementation of sub-circuit I is perfectly equivalent to the previous one, in terms of cost. Although equivalent however, these alternatives may yield very different circuit performance, depending on the routing. In other words, no module generator would have enough information to select a solution among the two alternatives before a complete layout is actually placed and possibly routed. For this reason both alternatives must be made available to the placer in order to insure that the best realization be selected.

4.4.5 Module Replacement Criteria

The placement algorithm is responsible for finding a combination of all available alternative realizations for a given circuit. At this point the placer has a better way to operate a selection. In fact, it can make precise estimations of global wiring, parasitics (cross-over capacitance, stray resistance and capacitance, etc.) and routability. Thus, since the cost function takes all these factors into account, the decision of accepting or rejecting

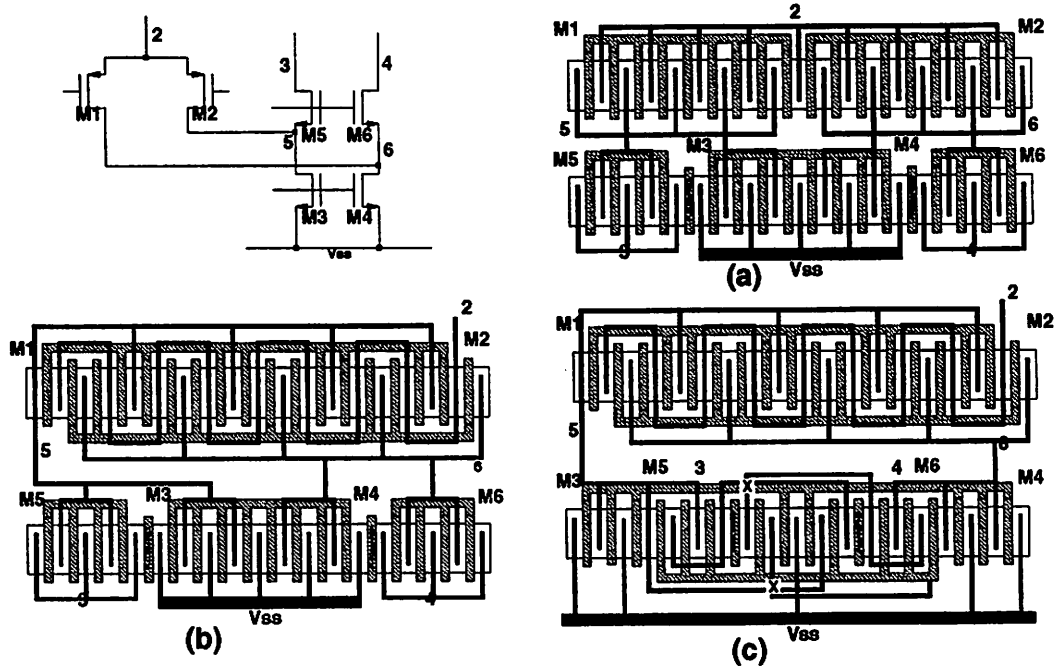


Figure 4.27: Alternative implementations of the differential pair and its active load

the new alternative is supported by a much better insight.

Capacitive and resistive parasitics are estimated using analytical models for a particular technology. Cross-over capacitances between nets are obtained from an estimate of the probability that the nets will cross after the routing as described in sections 4.5 and 7.2. From these estimates the cumulative effect of parasitics is evaluated based on the sensitivity information of performance with respect to each parasitic component in the circuit [119].

Violations to specifications can therefore be detected and included in the cost function as part of term $f_P(S)$, the calculation of which is discussed in detail in section 4.5. $f_P(S)$ is a measure for the degradation of performance K_i with respect to all parasitics present in the layout. It is approximated using a linear combination of all performance sensitivities and estimates of the parasitics. Assume that such a linearized performance representation is given. To illustrate the selection mechanism let us consider the input differential pair and its active loads from the circuit of Figure 4.26. Some implementations in a “full-stacked” design style are shown in Figure 4.27a, b, c.

Realization (a) shows minimum interleaving of all devices, this configuration rep-

resents the worst possible device matching for the pair of transistors $M1-M2$, $M3-M4$, and $M5-M6$. However interconnect parasitics are small and routing symmetry of nets 5, 6 is high.

In the third stack (c) considerably higher matching is achieved by configuring the transistor geometries according to a common-centroid pattern. This configuration requires however more wiring area and unavoidable signal path crossings are introduced. Configuration (b) is a trade-off between the previous two, with moderate interconnect length and good interleaving. Notice that no crossings are present in this configuration.

For simplicity, consider only nets 5, 6, V_{ss} , devices $M1$, $M2$, $M3$, $M4$ and performances K_1 , K_2 . For given transistor sizes and bias current, using sensitivity analysis, degradations ΔK_i for K_i , $i = 1, 2$, can be approximated as

$$\Delta K_i = \alpha_{i1} \Delta R_{S_{21}} + \alpha_{i2} \Delta R_{S_{34}} + \alpha_{i3} R_{S_{-1}} + \alpha_{i4} R_{S_{-2}} + \alpha_{i5} \Delta V_{t_{21}} + \alpha_{i6} \Delta V_{t_{34}} + \alpha_{i7} C_{56} + \alpha_{i8} C_5 + \alpha_{i9} C_6,$$

where $R_{S_{-j}}$ and $\Delta R_{S_{-jk}}$ are the degeneration resistances and resistance mismatches at the sources of devices M_j and M_k . $\Delta V_{t_{-jk}}$ are the mismatches of voltage threshold in the device pairs M_j and M_k . Parasitic capacitance C_{jk} represents the coupling between net j and k , while C_j is the substrate capacitance of net j . Moreover, terms $\alpha_{i1}, \dots, \alpha_{i9}$ represent pre-computed sensitivities of the performance model.

Suppose the specifications for K_i are given in terms of the inequality

$$\Delta K_i \leq \overline{\Delta K_i}, \quad i = 1, 2, \quad (4.19)$$

where $\overline{\Delta K_i}$ is the maximum acceptable degradation of performance K_i from nominal. Let us consider now the three alternative implementations of the differential pair of Figure 4.27.

These alternatives are all equivalent in terms of area and junction capacitances. However, realization (a) requires the smallest routing area for the interconnect of nets 5 and 6. This implies low interconnect resistances and capacitances. Moreover better matching between nets 5 and 6 are obtainable in the routing phase. Suppose now $\alpha_{i1}, \dots, \alpha_{i4}$, $\alpha_{i7}, \dots, \alpha_{i9}$ are large $\forall i$, i.e. resistive and capacitive mismatches dominate threshold voltage mismatches in affecting both performances. Then, there will be no specification violation in the sense of equation 4.19 and the contribution of $f_P(S)$ will be negligible or null. Otherwise, $f_P(S)$ will increase the cost of this configuration, thus decreasing the probability of its selection.

Consider now realization (c). This solution minimizes the threshold voltage mismatch, though at the expenses of the capacitive coupling and self capacitance of nets 5 and 6. So, if α_{i5} and α_{i6} are large $\forall i$, then the cost of this configuration will be lower. If both specifications were tight, i.e. a strong dependence of both performances of parasitic mismatches and threshold voltages was present, a trade-off configuration should be chosen. For instance, (b) represents a possible alternative configuration that could meet both specifications. Under the above conditions, the cost of this configuration is in fact lower than that of the other two. Thus the probability of acceptance is the highest among all configurations.

Clearly, if no flexibility had been allowed during the placement phase, it would not have been possible to enforce tight specifications on both performances. Hence, a constraint-driven approach to placement, in combination with module optimization is desirable, when many specifications, possibly tight, are present and trade-offs are possible. In Appendix A the convergence of the placement algorithm is proved under the same set of conditions proposed in [181] and [192].

4.5 Performance Models and Constraint Enforcement

In previous approaches [57, 67] the control of performance was carried out *indirectly*, i.e. enforcing user-defined topological constraints, such as symmetry and, to some degree, matching and cross-coupling. In fact, all these constraints are imposed based on some prior knowledge of performance dependencies. In this dissertation we have proposed a fully integrated performance-driven approach to the physical assembly of analog and mixed-signal circuits, one in which the user *may* interact but is also supported by numerical analysis to *quantify* the effects of critical parasitics on performance.

Hence, it is crucial that an accurate performance model based on parasitics is built in such a way that it can be used consistently at each step of the assembly. We recognize the importance of both deterministic parasitics and random errors in the layout. The former generally cause degradations in performance in a well defined manner, while the latter generate deviations from nominal which must be characterized using the appropriate tools of statistics.

4.5.1 Deterministic Model

The performance model which relates to *discrete* or *distributed* deterministic parasitics relies on sensitivity analysis and accurate parasitic estimation for its characterization. Let us refer to chapter 3 for the definitions of the terms describing the model. Given a circuit C , the set of operating points V , performance \mathbf{K} and its nominal value \mathbf{K}_V , call $\Delta\mathbf{K}$ the degradation of \mathbf{K} from nominal due to all parasitics. Assuming that a specification on \mathbf{K} is given (3.1), the violation of this specification can be added to the annealing cost function as described in section 4.3.

The key issue is the generation of a performance estimate sufficiently accurate and efficient to compute at each annealing step. Term $\Delta\mathbf{K}$ should also account for the alternative implementations of the interconnect which result in a different impact to performance. Let us assume that all parasitic elements characterizing C are known. Denote the array of these parasitics in terms of array $\mathbf{p} = [p_1 \dots p_{N_p}]^T$. Assuming that these parasitics represent the sole sensible cause of degradation for each performance measure, $\Delta\mathbf{K}$ can be written as a non-linear function $\Delta\mathbf{K}(\mathbf{p})$. If all parasitics are near their nominal value $p^{(0)}$ and $\Delta\mathbf{K}(\mathbf{p})$ is continuously differentiable at $p^{(0)}$, one can use a first order Taylor expansion for $\Delta K(p)$ as described in chapter 3.

$$\Delta\mathbf{K}(\mathbf{p}) = \mathbf{S} [\mathbf{p} - \mathbf{p}^{(0)}], \quad (4.20)$$

where generalized expressions for the computation of sensitivities from a set of arbitrary performance functions have been derived in [31, 135]. With this formulation, performance can be represented in a compact and rigorous way, as long as \mathbf{K} has a continuous and sufficiently regular behavior in an interval around its nominal value. The techniques for efficient numerical calculation of sensitivities in time and frequency domain have been discussed in chapter 3.

The sensitivity-based performance model of equation (4.20) cannot be directly used however to enforce a set of specifications without proper modifications. There are a number of reasons for this. (1) Since a layout is not available, the values of all parasitics are obviously not known *a priori*, hence at this stage of the design it is not possible to know the worst-case scenario. If both positive and negative sensitivities cumulatively contribute to performance degradation, then cross-cancellations could occur, thus resulting in errors in estimating parasitics. (2) Effects due to technology-related tolerances and temperature gradients need be taken into account in the sensitivity model. (3) The knowledge of the

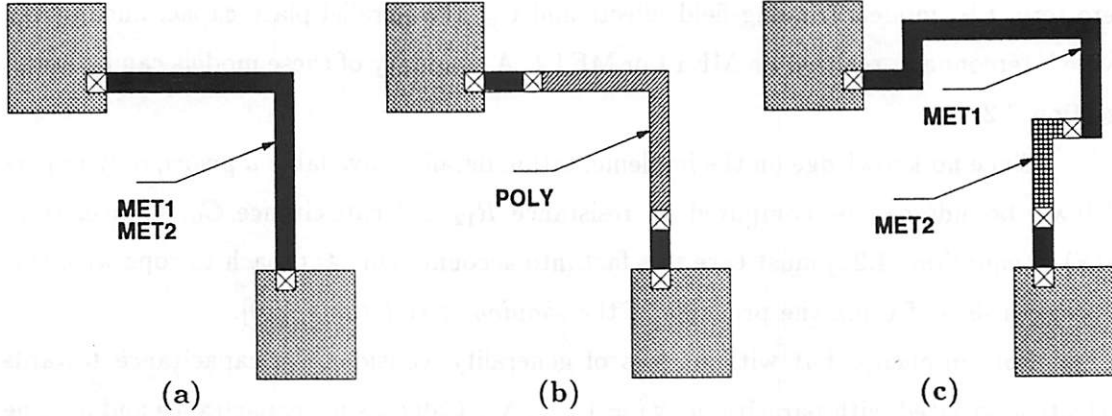


Figure 4.28: Alternative implementations of interconnect: (a) on metal1 or metal2; (b) on metal1 and poly; (c) on metal1 and metal2

implementation of the wiring and of the existence of wire-crossings is not available. Thus appropriate estimations need be carried out.

To cope with these problems we have proposed a sensitivity-based model where the negative and positive contributions to performance are kept separate [119]. Let $f_{i,P}(S)$ be the violation of the specification imposed on performance K_i , defined as

$$f_{i,P}(S) = F\left(\max\{\Delta K_i^+ - \overline{\Delta K_i^+}, 0\}\right) + F\left(\max\{\Delta K_i^- - \overline{\Delta K_i^-}, 0\}\right), \quad (4.21)$$

where all the terms have been defined in section 4.3 and function $F(\cdot)$ is described hereafter through an example.

Consider the layout in Figure 4.28. The wiring realization of Figure 4.28a involves the use of MET1 or MET2, hence the intrinsic resistance of the interconnect R_{12} will be

$$R_{12} = R_{\square 1} A_a \text{ or } R_{12} = R_{\square 2} A_a ,$$

depending on the material used for the wiring. The terms $R_{\square i}, i = 1, 2$ are the sheet resistance of MET1 and MET2 respectively, while A_a is the area of the interconnect. The realizations in Figures 4.28b,c involve the use of vias, both metalizations and a third wiring layer. Hence, resistance R_{12} could vary over a wide range of values. To characterize the capacitance towards substrate of the interconnect C_0 , we use a set of analytical models which account for its parallel-plate and fringe components as

$$C_0 = C_{fr} + C_{pp}w ,$$

where term C_{fr} models fringing-field effects and C_{pp} the parallel-plate capacitance of the w -wide interconnect, realized on MET1 or MET2. A summary of these models can be found in section 7.2.

Since no knowledge on the implementation details is available *a priori*, only upper- and lower-bounds can be computed on resistance R_{12} and capacitance C_0 . Hence, term $f_{i,P}(S)$ in equation (4.21) must take this fact into account. One approach to cope with this problem consists of using the principle of the *shaping of violations* [119].

For simplicity, but without loss of generality, consider the capacitance towards substrate associated with parasitic $p_j, \forall j = 1, \dots, N_p$. Call C_j such capacitance and assume that estimates of upper- and lower-bounds $C_j^{(min)}$ and $C_j^{(max)}$, exist. Let ρ_c be defined as the ratio $C_j^{(max)}/C_j^{(min)}$, which is identical for all parasitic capacitances. Then, degradation ΔK_i is bounded by

$$\Delta K_i^{(min)} \leq \Delta K_i \leq \Delta K_i^{(max)},$$

where

$$\begin{aligned} \Delta K_i^{(min)} &= \sum_{j=1}^{N_p} (S_{i,j} C_j^{(min)}) \\ \Delta K_i^{(max)} &= \sum_{j=1}^{N_p} (S_{i,j} C_j^{(max)}). \end{aligned}$$

In order to take this flexibility in realizing interconnect, violations must be weighted based on the severity of the violation. A way of differently weighting interconnect implementations based on the parasitics they are associated with, consists of *shaping* the violations due to the parasitics in equation (4.21) using the piecewise-linear function $F(\cdot)$, called *shaping function*. $F(\cdot)$ is given by the expression (see Figure 4.29)

$$F(\Delta K_i) = \begin{cases} 0, & \text{if } \Delta K_i^{(max)} \leq \overline{\Delta K_i} \\ \Delta K_i^{(max)} - \overline{\Delta K_i}, & \text{if } \Delta K_i^{(min)} < \overline{\Delta K_i} \leq \Delta K_i^{(max)} \\ (S_r + 1)[\Delta K_i^{(max)} - (S_r \rho_c + 1)\overline{\Delta K_i}], & \text{if } \overline{\Delta K_i} \leq \Delta K_i^{(min)} \end{cases}$$

If $S_r \gg 1$ then the values of $F(\cdot)$ for feasible and unfeasible placements differ by at least one order of magnitude. In PUPPY-A the value $S_r = 10$ has been selected.

A similar approach can be used for cross-over capacitances and resistances. Cross-over capacitances however require more care, since (1) the models are more complex; (2) the *existence* of a cross-over is not known *a priori* but it depends on a number of factors¹⁷; (3) there exists the ability of generating vertical and horizontal shielding devices.

¹⁷E.g. the type of router or even the routing schedule used.

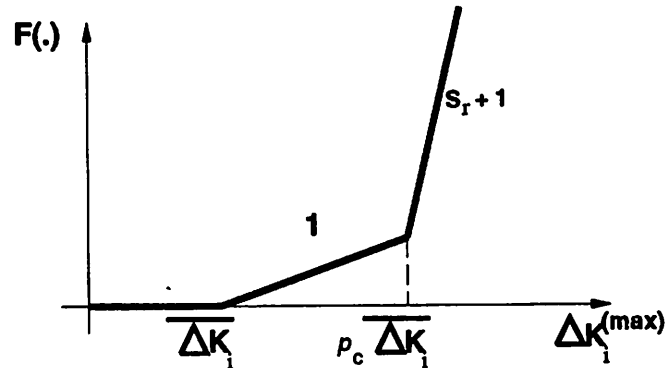


Figure 4.29: PUPPY-A's shaping function with twofold interconnect implementation

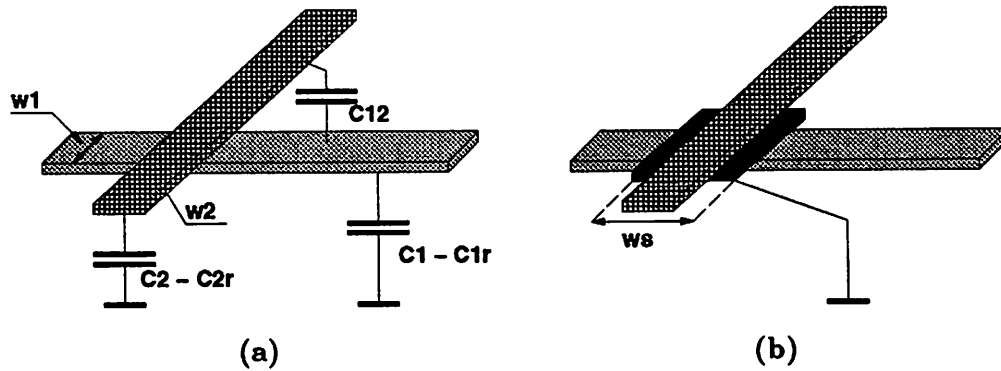


Figure 4.30: Coupling (a) without and (b) with vertical shielding. Indirect shielding effects due to the presence of the other interconnect have been added to the cross-over capacitance models

Modeling cross-coupling involves appropriate estimation of both coupling and shielding effects. Figure 4.30a shows the coupling capacitance between lines 1 and 2 and the shielding effect of line 1 onto the capacitance towards substrate of line 2 (C_{2r}). Line 2 also causes a similar but weaker reduction (C_{1r}) of the substrate capacitance of line 1. In our approach, analytical models for the coupling between line 1 and 2 are automatically derived using the tool CAPMOD [193], which also provides correction factors for the substrate capacitances of C_1 and C_2 . The model of coupling capacitance C_{12} , consisting of fringe ($C_{fr,1}$ and $C_{fr,2}$) and parallel plate components (C_{cc}), is given by

$$C_{12} = k_{12} + C_{fr,1}(w_2) + C_{fr,2}(w_1) + C_{cc}w_1w_2 ,$$

where k_{12} is a technology-dependent constant and $w_i, i = 1, 2$ is the physical width of the upper and lower wiring, respectively. The complete summary of all analytical models used

in our approach can be found in section 7.2. The correction terms C_{1r} and C_{2r} are given, for the structure in Figure 4.30a, by the expression

$$\begin{aligned} C_{1r} &= k_1 + C_{fr_1r} w_2 \\ C_{2r} &= k_2 + C_{fr_2r} w_1 + C_{fr_12} w_1 w_2 , \end{aligned} \quad (4.22)$$

where k_1 , k_2 , C_{fr_1r} , C_{fr_2r} and C_{fr_12} are technology-dependent constants. Models of simple vertical shielding structures can be generated using CAPMOD on a higher number of superimposed layers. See Figure 4.30b. The shielding effects can be accounted for in the model of equation (4.22) in terms of an additive term C_{is} , $i = 1, 2$ of the form

$$\begin{aligned} C_{1r} &\approx C_{1s} = k_1 + C_{fr_1s} w_s \\ C_{2r} &\approx C_{2s} = k_2 + C_{fr_2s} w_s + C_{fr_12s} w_s w_2 , \end{aligned}$$

where C_{fr_1s} , C_{fr_2s} and C_{fr_12s} are technology-dependent parameters and w_s is the width of the vertical shield shown in Figure 4.30b.

The estimation of parasitics also takes into account the junction capacitances of interconnected transistors. This component is relatively small if compared with the capacitance due to interconnect lines. However in case of short interconnect lines, it becomes dominant, hindering further parasitic reduction by reducing the size of the wire. A drastic reduction of this parasitic component is possible only through device abutment.

From the model of cross-over capacitances, one can construct an upper- and a lower-bound on corrected substrate capacitances C_j and cross-coupling capacitances C_{ij} . From these estimates a piecewise linear function $F(\cdot)$ can be computed in a similar manner. Since the *existence* of the cross-over is not known at the placement stage, one must estimate the probability $Pr_c(1, 2)$ of a cross-over between nets 1 and 2. In PUPPY-A $Pr_c(1, 2)$ is estimated using the following heuristic

$$Pr_c(1, 2) = \frac{\text{Area}(BB_1 \cap BB_2)}{\text{Area}(BB_1 \cup BB_2)} ,$$

where BB_i , $i = 1, 2$ is the bounding-box of net i (See Figure 4.31c) and the function Area calculates the area underlying the bounding-box. The bounding-box $BB_1 \cap BB_2$ is the shaded area in Figure 4.31c.

The capacitance $C_c(1, 2)$ associated with a probable cross-over between nets 1 and 2 is then computed as

$$C_c(1, 2) = Pr_c(1, 2) C_{12} .$$

The same reasoning can be applied to any arbitrary net pairs in the circuit.

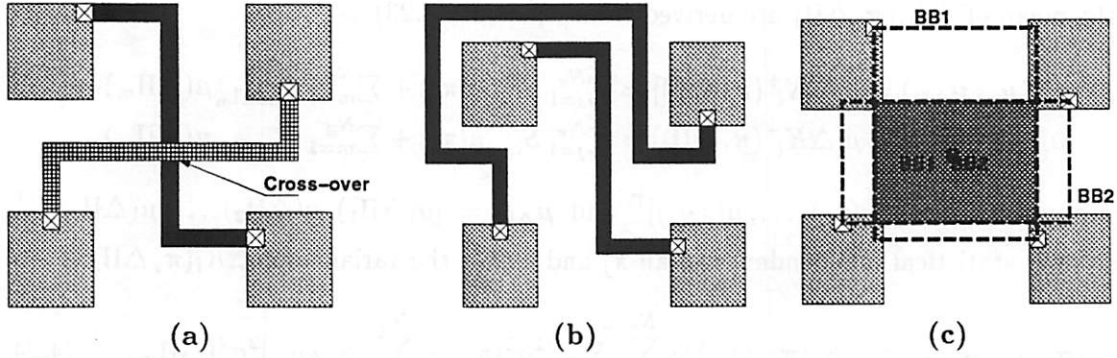


Figure 4.31: (a) Cross-coupling between MET1 and MET2 interconnect; (b) configuration avoiding cross-over; (c) heuristic for crossover probability estimation

4.5.2 Non-Deterministic Parasitic Constraint Enforcement

Let us turn our attention now to the case when parasitics and parasitic mismatches are modeled as random variables. If the distribution and/or the first moments of these variables are known, one can extend the performance model to account for random effects too. Let parasitic element $\pi_j, \forall j = 1, \dots, N_\pi$ be a random variable with finite mean $\mu(\pi_j)$ and variance $\sigma^2(\pi_j)$. Let parameter or parasitic mismatch $\Delta\Pi_m, \forall m = 1, \dots, N_\Pi$ be also a random variable with mean $\mu(\Delta\Pi_m)$ and variance $\sigma^2(\Delta\Pi_m)$. Let us characterize degradation ΔK_i based on its deterministic and non-deterministic components

$$\Delta K_i = \Delta K_i(\mathbf{p}) + \Delta K_i(\boldsymbol{\pi}, \Delta\Pi) ,$$

where $\boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_{N_\pi}]^T$ and $\Delta\Pi = [\Delta\Pi_1, \Delta\Pi_2, \dots, \Delta\Pi_{N_\Pi}]^T$.

Moreover, assume that the sensitivities of $K_i, \forall i = 1, \dots, N_k$ with respect to these parasitics are known. Assuming that all random parasitics take values close enough to their mean value, i.e. the parasitic variance is reasonably small, the random component of performance degradation $\Delta K_i(\boldsymbol{\pi}, \Delta\Pi)$ can be approximated as

$$\Delta K_i(\boldsymbol{\pi}, \Delta\Pi) \approx \sum_{j=1}^{N_\pi} S_{i,\pi_j} \pi_j + \sum_{m=1}^{N_\Pi} S_{i,\Delta\Pi_m} \Delta\Pi_m , \quad (4.23)$$

where S_{i,π_j} and $S_{i,\Delta\Pi_m}$ are the sensitivities of K_i with respect to π_j and $\Delta\Pi_m$ respectively. We now return to the explicit representation of positive and negative sensitivities, which was dropped in chapter 3 to simplify the notation. The positive and negative components

of the mean of $\Delta K_i(\pi, \Delta\Pi)$ are derived from equation (4.23) as

$$\begin{aligned}\mu_i^+(\mu_\pi, \mu_{\Delta\Pi}) &= \mu[\Delta K_i^+(\pi, \Delta\Pi)] \approx \sum_{j=1}^{N_\pi} S_{i,\pi_j}^+ \mu(\pi_j) + \sum_{m=1}^{N_\Pi} S_{i,\Delta\Pi_m}^+ \mu(\Delta\Pi_m) \\ \mu_i^-(\mu_\pi, \mu_{\Delta\Pi}) &= \mu[\Delta K_i^-(\pi, \Delta\Pi)] \approx \sum_{j=1}^{N_\pi} S_{i,\pi_j}^- \mu(\pi_j) + \sum_{m=1}^{N_\Pi} S_{i,\Delta\Pi_m}^- \mu(\Delta\Pi_m),\end{aligned}$$

where $\mu_\pi = [\mu(\pi_1), \mu(\pi_2), \dots, \mu(\pi_{N_\pi})]^T$ and $\mu_{\Delta\Pi} = [\mu(\Delta\Pi_1), \mu(\Delta\Pi_2), \dots, \mu(\Delta\Pi_{N_\Pi})]^T$. Assuming statistical independence of all π_j and $\Delta\Pi_m$, the variance of $\Delta K_i(\pi, \Delta\Pi)$ is

$$\sigma_i^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2) = \sigma^2[\Delta K_i(\pi, \Delta\Pi)] \approx \sum_{j=1}^{N_\pi} |S_{i,\pi_j}|^2 \sigma^2(\pi_j) + \sum_{m=1}^{N_\Pi} |S_{i,\Delta\Pi_m}|^2 \sigma^2(\Delta\Pi_m), \quad (4.24)$$

where $\sigma_\pi^2 = [\sigma^2(\pi_1), \sigma^2(\pi_2), \dots, \sigma^2(\pi_{N_\pi})]^T$, $\sigma_{\Delta\Pi}^2 = [\sigma^2(\Delta\Pi_1), \sigma^2(\Delta\Pi_2), \dots, \sigma^2(\Delta\Pi_{N_\Pi})]^T$, $S_{i,\pi_j} = \max\{S_{i,\pi_j}^+, S_{i,\pi_j}^-\}$ and $S_{i,\Delta\Pi_m} = \max\{S_{i,\Delta\Pi_m}^+, S_{i,\Delta\Pi_m}^-\}$. Due to the generally large number of parasitic components, by Central Limit Theorem, one can show that $\Delta K_i^\pm(\pi, \Delta\Pi)$ has a nearly normal distribution $N\{\mu_i^\pm(\mu_\pi, \mu_{\Delta\Pi}), \sigma_i^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2)\}$.

Thus, in the presence of random parasitics, at least one additional specification is needed for each performance measure K_i . Let us define $\overline{\sigma_i^2}$ the specification on the variance of performance measure K_i and $\overline{\sigma_K^2} = [\overline{\sigma_1^2}, \overline{\sigma_2^2}, \dots, \overline{\sigma_{N_k}^2}]^T$ the $N_k \times 1$ vector of specifications related to all performance measures. Hence, equations (3.4), (3.5) generalize to

$$\Delta \mathbf{K}(\mathbf{p}) + \mu_K^+(\mu_\pi, \mu_{\Delta\Pi}) - \overline{\Delta \mathbf{K}^+} \leq \mathbf{0} \quad (4.25)$$

$$\Delta \mathbf{K}(\mathbf{p}) - \mu_K^-(\mu_\pi, \mu_{\Delta\Pi}) + \overline{\Delta \mathbf{K}^-} \geq \mathbf{0} \quad (4.26)$$

$$\sigma_K^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2) - \overline{\sigma_K^2} \leq \mathbf{0}, \quad (4.27)$$

where $\mu_K^\pm(\mu_\pi, \mu_{\Delta\Pi})$ and $\sigma_K^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2)$ are $N_k \times 1$ vectors defined as

$$\begin{aligned}\mu_K^\pm(\mu_\pi, \mu_{\Delta\Pi}) &= [\mu_1^\pm(\mu_\pi, \mu_{\Delta\Pi}), \mu_2^\pm(\mu_\pi, \mu_{\Delta\Pi}), \dots, \mu_{N_k}^\pm(\mu_\pi, \mu_{\Delta\Pi})]^T \\ \sigma_K^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2) &= [\sigma_1^2(\mu_\pi, \mu_{\Delta\Pi}), \sigma_2^2(\mu_\pi, \mu_{\Delta\Pi}), \dots, \sigma_{N_k}^2(\mu_\pi, \mu_{\Delta\Pi})]^T.\end{aligned}$$

The cost function is constructed as in equation (4.21).

In case the condition of statistical independence does not hold, variance $\sigma_i^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2)$ as calculated in (4.24) must be corrected by factor $\sigma_{i_err}^2$ which accounts for the cross-correlations between all parameter pairs.

$$\begin{aligned}\sigma_{i_err}^2 &= 2 \left\{ \sum_{j,k} \text{sign}(i, k) S_{i,\pi_j}^\pm S_{i,\pi_k}^\pm E(\pi_j \pi_k) + \right. \\ &\quad \sum_{m,n} \text{sign}(m, n) S_{i,\Delta\Pi_m}^\pm S_{i,\Delta\Pi_n}^\pm E(\Delta\Pi_m \Delta\Pi_n) + \\ &\quad \left. \sum_{j,m} \text{sign}(j, m) S_{i,\pi_j}^\pm S_{i,\Delta\Pi_m}^\pm E(\pi_j \Delta\Pi_m) \right\},\end{aligned} \quad (4.28)$$

where (j, k) , (m, n) and (j, m) denote all the possible pairs of parameters and function $sign()$ is described as

$$sign(i, j) = \begin{cases} -1 & , \text{if } S_i = S^+ \text{ AND } S_j = S^- \text{ OR } S_i = S^- \text{ AND } S_j = S^+ \\ 1 & , \text{otherwise} \end{cases}$$

Operator $E(.)$ denotes the expected value.

Alternatively, one can approach the problem in the following manner. Let us define a $(N_\pi + N_\Pi) \times 1$ vector $\mathbf{x} = [\pi, \Delta\Pi]^T$ and let us assume that parameters π_j and $\Delta\Pi_m$ be Gaussian. Moreover, assume that variance-covariance matrix \mathbf{A} associated with \mathbf{x} is known. Then, \mathbf{A} is by construction a positive-definite, symmetric $(N_\pi + N_\Pi) \times (N_\pi + N_\Pi)$ square matrix and hence it can be decomposed as

$$\mathbf{A} = \mathbf{LDM}^T ,$$

where \mathbf{L} and \mathbf{M} are square orthogonal matrices and \mathbf{D} is a diagonal matrix. Let us now define a new vector \mathbf{y} as a linear combination of \mathbf{x}

$$\mathbf{y} = \mathbf{L} \begin{bmatrix} \pi \\ \Delta\Pi \end{bmatrix} .$$

One can show that, due to the orthogonality of \mathbf{L} and \mathbf{M} , the new variables of \mathbf{y} are necessarily uncorrelated and, since Gaussian, statistically independent as well. Hence, σ_i^2 can be computed by replacing vector $[\sigma_\pi^2, \sigma_{\Delta\Pi}^2]^T$ with $\sigma_y^2 = [\sigma^2(y_1), \sigma^2(y_2), \dots, \sigma^2(y_{(N_\pi+N_\Pi)})]^T$ in equation (4.24) after the appropriate sensitivity transformations.

Unfortunately in some cases process-related parameters cannot be assumed Gaussian, hence an approach based on the corrected performance variance must be used. In most processes however the cross-correlation terms are not known precisely or they are not available for all parameter pairs. A possible solution to this problem is the following. Let us assume that a lower-bound $r^{(min)}$ and upper-bound $r^{(max)}$ to all cross-correlations is known or can be estimated. Hence, by replacing all the correlation terms with $r^{(max)}$ in (4.28) and ordering in increasing size each component, one can find a set of pairs whose cumulative effect on $\sigma_{i_err}^2$ is a fraction α of $\sigma_i^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2)$ and can therefore be eliminated.

Consequently, term $\sigma_{i_err}^2$ can be re-written as

$$\sigma_{i_err}^2 = \alpha \sigma_i^2(\sigma_\pi^2, \sigma_{\Delta\Pi}^2) + 2 \left\{ \sum_{j', k'} S_{i, \pi_{j'}}^\pm S_{i, \pi_{k'}}^\pm E(\pi_{j'} \pi_{k'}) + \sum_{m', n'} S_{i, \Delta\Pi_{m'}}^\pm S_{i, \Delta\Pi_{n'}}^\pm E(\Delta\Pi_{m'} \Delta\Pi_{n'}) + \sum_{j', m'} S_{i, \pi_{j'}}^\pm S_{i, \Delta\Pi_{m'}}^\pm E(\pi_{j'} \Delta\Pi_{m'}) \right\} ,$$

where (j', k') , (m', n') and (j', m') denote all the parameter pairs which have not been eliminated. The remaining correlation factors can be estimated using some combination of $r^{(min)}$ and $r^{(max)}$, hence allowing the estimation of $\sigma_{i_err}^2$.

Hence equation (4.29) is modified as

$$\sigma_{\mathbf{K}}^2(\sigma_{\pi}^2, \sigma_{\Delta\Pi}^2) + \sigma_{\mathbf{K_err}}^2 - \overline{\sigma_{\mathbf{K}}^2} \leq 0, \quad (4.29)$$

where $\sigma_{\mathbf{K_err}}^2 = [\sigma_{i_err}^2, \sigma_{2_err}^2, \dots, \sigma_{N_k_err}^2]^T$. This method may lack in accuracy mainly due to the estimation of correlations between the most critical parameters.

4.6 Substrate-Aware Placement

In this dissertation we have considered analog-specific constraints related to topology and parasitics. The first type of constraint is of *global* nature, since it controls the relative location of various objects in the layout. The second constraint is a *local* one, since it relates, at least at low frequencies, to discrete parasitic components. Substrate effects are generally referred to in terms of parasitic pulsing currents injected by fast-switching circuits that induce performance degradation in sensitive analog circuits interfering with their operation.

Placement can be a critical step in the physical assembly since the relative position of sensitive blocks and noise injecting circuitry can influence strength and waveform of parasitic substrate currents. Traditionally, substrate-aware optimization has not been as important as substrate analysis. Substrate noise analysis has been generally addressed *a posteriori*, i.e. after completion of schematic design and physical assembly. In many design problems however, a *dynamic* substrate noise analysis would be preferable. Unfortunately, experience has shown the extreme time complexity required to accurately model substrate and estimate performance degradations due to switching noise.

Recently, a number of authors have addressed the problem of performing these tasks efficiently within physical assembly phases [70, 194, 174]. Common to these approaches is the use of a Finite Difference method for the evaluation of the electric field on a coarse grid spanning the workspace, combined with AWE for an efficient solution of the resulting system of simultaneous algebraic equations. However, these methods often cannot guarantee the accuracy needed for reliable performance estimation, due to the extremely coarse grids used. In addition, even if dense or non-uniform grids were used, at no extra cost

in computation, the alignment requirements of grid and layout objects would be so stringent to make it impossible to use the methods in iterative algorithms based on progressive and often minimal modifications. The latter problem can now be effectively addressed if e.g. *Voronoi tessellation* is used during the grid generation [195].

More recently, the use of analytical approximations has been suggested to derive a simpler model for substrate parasitics [196, 197]. A major drawback of these approach is the lack of accuracy and the strong dependence on the technology and on the physical implementation of the circuit, which might not be available at high-level design stages.

In this dissertation we advocate a constraint-driven approach to substrate-aware placement. In order for the placement tool to be effective in preventing violations to performance specifications, the following features must be implemented in the tool. First, a model for each noise injecting module must exist. The model should characterize the *waveform* and the *spatial location* where the noise is injected as precisely as possible. The waveform will determine how the noise will affect the sensitive circuit, while the location of injection will set the strength and energy of the noise. Second, a compact model of substrate transport should be available and efficient substrate current evaluation should be possible independently of the circuit configuration. Finally, a model for substrate noise absorption and its effect on performance should be defined.

We propose the generation of compact models characterizing both the spatial and the waveform components of the noise injectors in each high-frequency circuit in the layout. Moreover, we advocate the use of a set of specific constraints on the maximum energy and amplitude of the signal at a sample of frequencies of operation for each critical node in the sensitive circuit. These constraints are compared with the estimate of the injected signal, i.e. the output of the model simulating it. The transport model is obtained using efficient techniques for Green's Function-based substrate analysis. The SA-based placement tool PUPPY-A has been modified to take into account violations to the above constraints in its cost function, while, to speed-up the evaluation of substrate transport, a number of heuristics have been implemented within the placer itself.

4.6.1 Modeling Switching Noise

For the purpose of physical assembly or schematic design, switching noise is often modeled as a simple signal, generally synchronized with the clock, if one is present. A

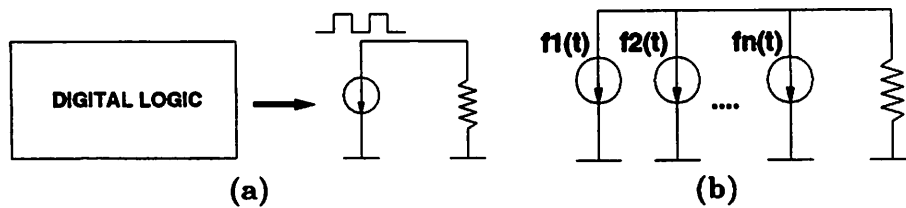


Figure 4.32: (a) *Simple injection model*; (b) *Proposed injection model*

number of examples of this modeling style can be found in the literature [70, 194, 174]. Figure 4.32a shows an example of such models. We believe that these models might be too approximate to be reliable for our placement tool. For this reason we propose an alternative method consisting of the following steps:

1. isolate various noise injecting components
2. generate a model for each injecting component
3. select a minimum number of parameters for each generator

While steps 1 and 2 can be performed manually, a rigorous technique is needed to carry out step 3. We propose the use of fitting techniques based on the minimization of the mean-square error of the signal function or of the energy of the signals. The models usually require careful modeling of the signals by use of a number of analytical functions $f_i(t)$, $i = 1, \dots, n$, characterized by a number of parameters, such as amplitude, phase, etc. Such a model is shown in Figure 4.32b. The method is illustrated in all its steps with an example in chapter 9.

4.6.2 Modifying the Original Placement Algorithm

Generally, improvements on the performance degradation due to substrate-induced switching noise can be achieved by placing noise injecting and noise sensitive modules at a certain distance or by creating special structures, such as low-resistivity guard-rings, around noise injectors. The first provision is implemented in a placer using the conventional SA move-set. The second issue is generally solved by extending the search space, allowing the annealing to choose from a number of alternative implementations for a module, including one with a guard-ring implemented around it.

In this dissertation we restrict our attention to the first option, where our Green's Function-based substrate analysis method is used for the evaluation of the substrate at

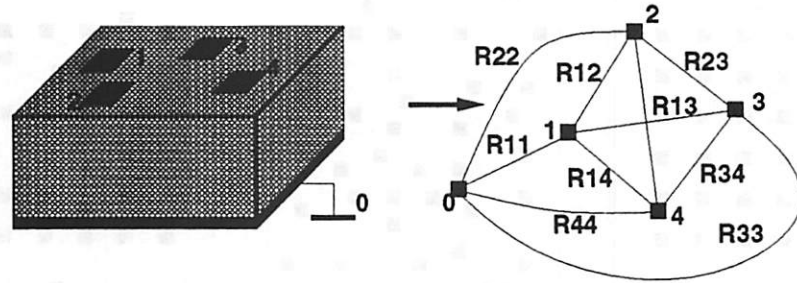


Figure 4.33: Mapping of substrate onto fully connected graph $G_S(V, E)$.

each annealing step. We approach the problem of evaluating the effects of substrate on performance in the following way.

1. generate a macroscopic model for each switching noise injector
2. generate constraints for each node of noise-sensitive modules
3. generate the resistive network associated with substrate
4. quantify violations to constraints

For each noise injecting module j a model is created which accurately reproduces substrate injected noise, taking into account both impact ionization and capacitive coupling through devices and interconnect lines. The model $V_S(\Pi_j)$ is based on a bank of independent current noise generators with a unified set of parameters represented by vector Π_j .

Then, the sensitivity of a given performance K_i is computed with respect to the parameters Π_j related to each noise source j acting on every node in the analog modules being placed. Using constrained optimization techniques [116] and the specification on the maximum positive and negative performance degradation $\overline{\Delta K_i^\pm}$, a set of bounds $\Pi_j^{(bound)}$ is generated only for a reduced set of *critical nodes* n_c . The set n_c is generated based on the cumulative effect of all parasitic noise sources acting on each node similarly as in [116].

In step 3 a given placement configuration is mapped onto a fully connected graph $G_S(V, E)$, whose vertices V are the substrate contacts and edges E are weighted by the conductance Y_{ij} or resistance R_{ij} between the corresponding vertices i and j . Figure 4.33 shows the mapping procedure. The techniques for the evaluation of the edges have been described in detail in section 8.2. The calculation of all violations in step 4 to the given constraints is carried out by solving the circuit underlying $G_S(V, E)$ and evaluating the appropriate parameters at each critical node.

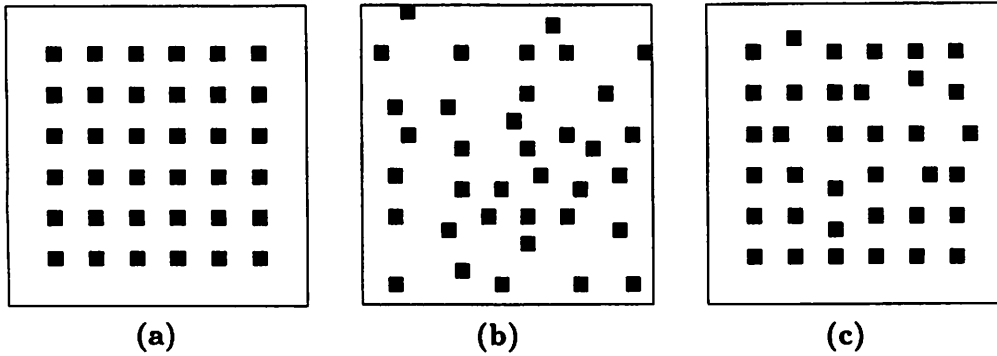


Figure 4.34: (a) Initial contact grid; (b) Reshuffling of contacts at high temperatures; (c) Resulting grid at lower temperatures

At each stage of the annealing only steps 4 and 3 need be repeated, since steps 1 and 2 are carried out only once for each chip. The efficiency of a Green's Function based substrate simulator, though high, is still insufficient for such computationally intensive algorithm as SA, hence, appropriate heuristics must be developed. In SA, at high annealing temperatures, considerable reshuffling is allowed on the components of the layout. Hence, the locations of switching noise generators and receptors can be significantly modified. At lower temperatures on the contrary, modules move by lesser amounts in average. Hence, the edges of $G_S(V, E)$ change with lower frequency and by a lesser amounts.

As an illustration consider a regular 36 contact grid shown in Figure 4.34a. Plot 4.35 shows the average variation of the resistive components of the substrate network when high-temperature (Figure 4.34b) and low-temperature (Figure 4.34c) contact perturbations occur during the unfolding of SA. On the other hand, only when changes in the edges of $G_S(V, E)$ reflect a significant change in any performance measure K_i , the entire substrate network should be evaluated along with the estimate of performance degradation ΔK_i . This observation leads us to the following heuristics for the evaluation of substrate effects after each tentative annealing move.

When a new temperature T_k is reached, the full graph $G_S(V, E)$ is solved, i.e. all the edges in V are evaluated exactly, using the Sherman-Morrison update to obtain the new matrix \mathbf{P}^{-1} . After a new move m_k and the associated translation $\mathbf{v} = [\Delta x, \Delta y]^T$ is selected by the annealing algorithm (Figure 4.37), the sensitivity of the edges of $G_S(V, E)$ can be efficiently computed using the techniques outlined in section 8.5.2. Suppose the set n_c of all critical receptors has been derived for the circuit, moreover let n_s be the set of all

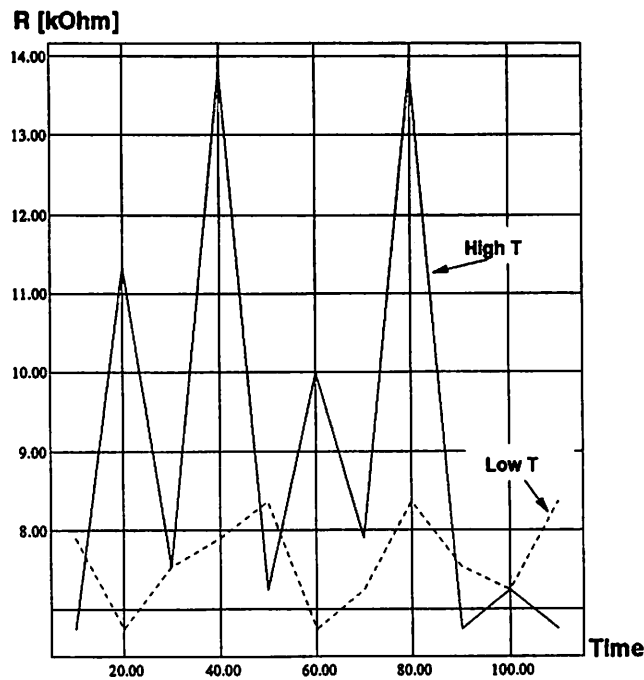


Figure 4.35: *Resistive network reacting to high-temperature and low temperature contact reshuffling*

```

foreach temperature  $T_k$ 
  evaluate_substrate_network_exactly; // use Sherman-Morrison to update
                                     // configuration at equilibrium
  repeat
     $m_k = \text{select\_move};$ 
    estimate_network_change( $m_k$ ); // use gradient-based method
    foreach node  $j \in n_c$ 
       $\Pi_j = \text{estimate\_cumulative\_noise};$ 
      if  $\Pi_j \geq \Pi_j^{(bound)}$ 
        evaluate_substrate_network_exactly;
        go_to_next_node;
        evaluate_cost_function;
        accept_or_reject_move;
    until equilibrium_reached

```

Figure 4.36: *Heuristic for the combined use of Sherman-Morrison and gradient-based methods*

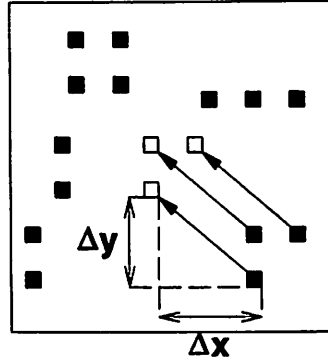


Figure 4.37: *Small number of contacts translating within the workspace*

noise injecting nodes. Let $[\mathbf{Y}_c]_{m_k}$ be the conductance matrix of all the nodes in n_c and in n_s , and let $[\Delta \mathbf{Y}_c]_{m_k}$ be its update.

By equation (8.50) term $[\Delta \mathbf{Y}_c]_{m_k}$ is estimated as

$$[\Delta \mathbf{Y}_c]_{m_k} \approx \left[\nabla_{\mathbf{v}} \mathbf{Y}^T \right]_0 \mathbf{v}, \quad (4.30)$$

where term $[\nabla_{\mathbf{v}} \mathbf{Y}^T]_0$ is defined as in equation (8.50) for matrix \mathbf{Y}_c . After updating \mathbf{Y}_c , the resistive network is solved and parameter Π_j can be evaluated for all critical nodes j . By comparing Π_j with the bound $\Pi_j^{(bound)}$ one can obtain the corresponding violation. If a violation to specifications has occurred, then a precise extraction step must be performed and the precise value for the violation is used to drive the cost of the annealing in a manner similar to [119]. Otherwise the contribution of substrate noise to node j in degrading performance K_i is considered negligible and the cost function will not take it into account. The cost relative to the remaining analog-specific constraints, as well as area and wiring length will however be computed. The placement algorithm is proved to converge to a global minimum under the same conditions of [181] and [192] when it is modified to account for noise substrate transport evaluation. See Appendix A.

Using the sensitivities with respect to technology parameters presented in section 8.4 it is possible to carry out *trend analysis* on technology migration and scaling. Trend analysis can give important insights on how performance will change with the technology and may save unnecessary re-design loops. An example of trend analysis for technology scaling is presented in section 9.2.

4.6.3 Advanced Features: Thermal Analysis

Sometimes it might be useful to consider the problem of overheating within a chip. Overheating can generate catastrophic faults, by destroying active and passive devices, and parametric faults, by causing devices to behave differently according to the temperature of the substrate in their vicinity. Substrate thermal behavior is characterized by solving equation (8.29) assuming that the backplane is an isotherm e.g. at room temperature and that the sole sources of heat are MOSFETS and substrate resistances. Other sources/sinks of heat such as pads and polysilicon resistances could be easily added to the analysis using back-annotation in the schematic.

Using our Green's Function-based method, a thermal equivalent resistive circuit for the substrate can be computed. The temperature of each device is estimated by calculating the voltage at a particular node of the circuit. Using HSPICE the sensitivity of a performance K_i with respect to the temperature of each device d can be evaluated and the corresponding sensitivity-based model can be built for degradation ΔK_i as

$$\Delta K_i = \sum_d S_{i,T_d} \Delta T_d ,$$

where S_{i,T_d} is the sensitivity of K_i with respect to temperature of device d . Term ΔT_d represents the temperature deviation of d from the nominal value of T_0 . All violations to performance specifications are computed based on this estimate and integrated directly into the cost function which drives the annealing.

4.7 Placement with Analog Constraints: A Case Study

Consider again the clocked comparator COMPL. For this circuit a set of module implementations were first generated using LDO. Then, PUPPY-A implemented the final placement by selecting the optimal combination of modules available.

4.7.1 Module Generation

Many possible stack implementations exist for this circuit. Two of such possible solutions are shown in Figure 4.38. All transistors have been grouped in four subcircuits, according to their channel widths, their matching requirements and bulk nets. Only transistors belonging to the same subcircuit can belong to the same stack. The two solutions

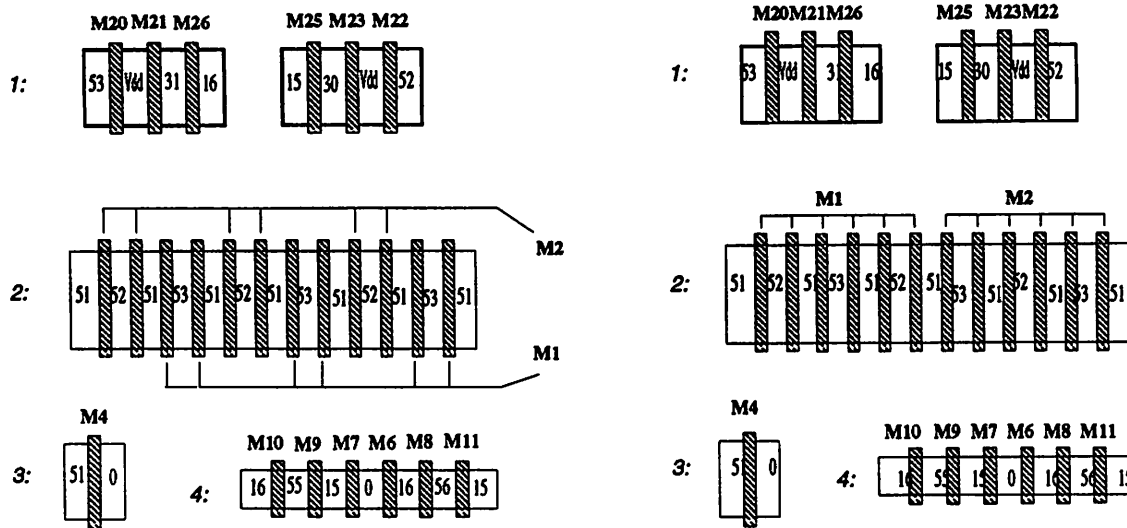


Figure 4.38: Clocked comparator COMPL - Two alternative full-stacked implementations

nets	without abutment	with abutment	% reduction
15,16	41.2 fF	34.4 fF	16.5%
55,56	13.4 fF	6.6 fF	51%

Table 4.1: Capacitances in the stacks generated for the clocked comparator COMPL

only differ by the implementation of the stack containing the input differential pair. In the first realization they are interleaved in a common-centroid pattern, which minimizes device mismatch, but usually requires a considerable area overhead, due to the complex routing required. The second solution is symmetric, but without the common-centroid structure. The choice between such alternative realizations is left to the user or it can be made automatically during the placement phase on the ground of area and routing considerations. In both solutions, critical nets 55, 56, 15, 16, whose capacitance toward the substrate strongly influences the comparator speed, have been kept in internal positions when possible. The capacitance values are reported in Table 4.1. In both cases stack abutment yielded a reduction of net capacitance. Such a reduction can be exploited to improve the flexibility of the routing stage. For example, consider nets 55 and 56. Abutment allowed each of them to be reduced by more than 6.6fF, which in our process is the capacitance of a 136 μm -

long minimum-width metal-1 wire. Therefore the router is allowed to draw longer wires for the sensitive nets, thus increasing the success rate and the robustness of the entire layout synthesis.

These capacitance values constitute new nominal values and better lower limits, and can be used to compute a new set of bounds. By using these values:

$$\begin{aligned}
 C_{15}^{(min)} &= C_{15}^{(nom)} = C_{16}^{(min)} = C_{16}^{(nom)} = 34.4 \text{ fF} \\
 C_{55}^{(min)} &= C_{55}^{(nom)} = C_{56}^{(min)} = C_{56}^{(nom)} = 6.6 \text{ fF} \\
 \max C &= 100 \text{ fF} \\
 \min R &= 0 \\
 \max R &= 50 \text{ } \Omega
 \end{aligned} \tag{4.31}$$

we obtain the following arrays:

$$\mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 5.5 \text{ ns} \\ 0.0 \\ 0.0 \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 1.5 \text{ ns} \\ 1 \text{ mV} \\ 1 \text{ mV} \end{bmatrix}$$

Here the delay degradation, due to the insertion of junction capacitances, is apparent. The next set of bounds found by PARCAR is the following:

$$\mathbf{p}^{(\text{bound})} = \begin{bmatrix} 67.1 \text{ fF} \\ 67.1 \text{ fF} \\ 48.9 \text{ fF} \\ 48.9 \text{ fF} \\ 1.0 \text{ } \Omega \\ 7.4 \text{ } \Omega \\ 7.4 \text{ } \Omega \\ 7.4 \text{ } \Omega \\ 7.5 \text{ } \Omega \\ 19.9 \text{ } \Omega \\ 49.5 \text{ } \Omega \end{bmatrix} \tag{4.32}$$

Notice that all bounds on critical capacitances have been lowered, because the degradation allowed to delay is smaller than in the previous step. In fact half of the degradation allowed at the beginning of the layout design has been introduced by junction capacitances alone, and the remaining half will be available to the remaining tools (i.e. placement and routing tools).

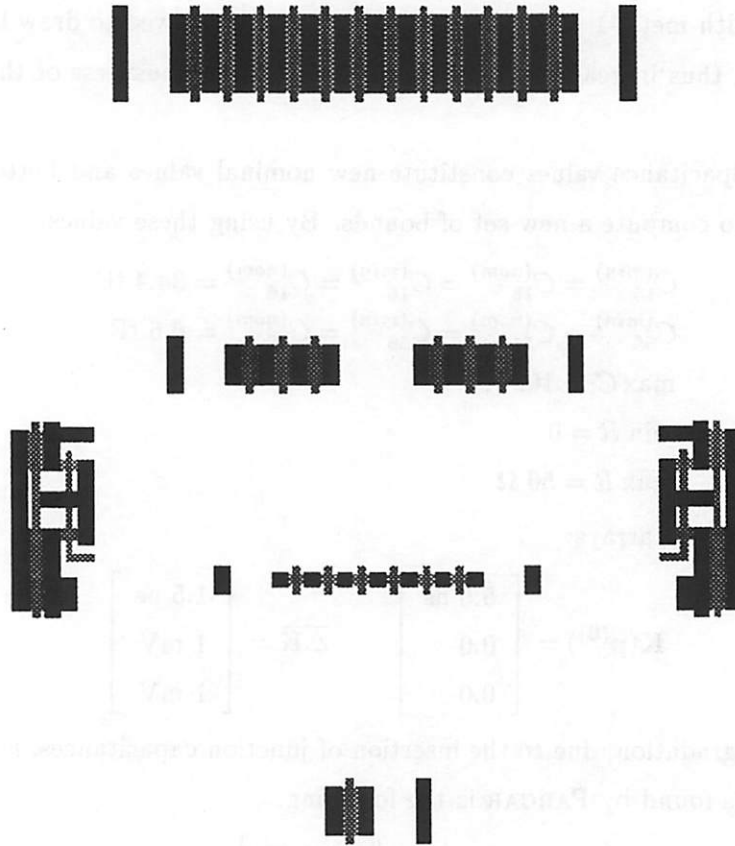


Figure 4.39: *Placement of comparator COMPL obtained with PUPPY-A*

4.7.2 Placement Algorithm

The placement of Figure 4.39 was obtained by enforcing all symmetry and matching constraints found in section 3.5 using the modified SA algorithm described in section 4.3. All parasitic constraints were enforced indirectly by evaluating their cumulative effects on performance. The algorithm's cost function was driven by the resulting performance model compared with the specifications as outlined in section 4.3. After placement, estimates of the minimum values of all critical parasitics can be drawn, taking into account the junction capacitances of all terminals and the estimated minimum length of interconnections

tool	item	conditions
PUPPY-A	cooling schedule	$T_0, K, \sigma(f), \delta$: see [182]; $t_k = 2 \times 10^3 \forall k$;
	constraints	P, S and M enforced
	optimization priority	$\alpha_A = 0.01, \alpha_W = 0.1, \alpha_O = 1, \alpha_{WE} = 1.0,$ $\alpha_S = 2.0, \alpha_M = 2.0, \alpha_P = 1.0$
	routing estimation	non-minimum spanning-tree
	interconnect dependency	$\rho_c = 2.0, S_r = 10$

Table 4.2: Conditions of operation for the placement tool used in the synthesis path. The symbols P, S and M denote parasitic, symmetry and matching constraints, respectively

between terminals:

$$\mathbf{p}^{(0)} = \mathbf{p}^{(min)} = \begin{bmatrix} 10.1 \text{ fF} \\ 10.1 \text{ fF} \\ 51.0 \text{ fF} \\ 51.0 \text{ fF} \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} \quad (4.33)$$

In this design substrate and thermal effects were ignored. For an example dealing with analog-specific constraints derived from substrate considerations, we refer to chapter 8. Table 4.2 lists the conditions under which the placement algorithm was run. The values of the priority weights were selected empirically by performing a series of experiments on a set of eight benchmarks. The values obtained were then kept constant throughout the experimentation. Details on the statistics for the placement of this circuit are available in section 9.1.1.

Chapter 5

Routing

*Non era ancor di là Nesso arrivato,
quando noi ci mettemmo per un bosco
che da neun sentiero era segnato.*

*Non fronda verde, ma di color fosco;
non rami schietti, ma nodosi e 'nvolti;
non pomi v'eran, ma stecchi con tòsco:*

*non han sì aspri sterpi né sì folti
quelle fiere selvagge che 'n odio hanno
tra Cecina e Corneto i luoghi còliti.*

Dante Alighieri, "Inferno", Canto XIII

In this chapter we present a composite routing algorithm, designed for radio-frequency (RF) and monolithic microwave (MMICs) ICs. In this approach, performance sensitivities are used to derive a set of bounds on critical parasitics and to generate weights for a cost function which drives an area router. In addition to these bounds, design often requires that the length of interconnect lines be equal to predefined values. The routing scheme enforces both types of constraints in two phases. During the first phase all parasitic constraints are enforced on all nets. Length constraints are enforced during the second phase by expanding each net simultaneously while ensuring that no additional violations

to parasitic constraints are introduced in the layout. During both phases accurate and efficient parasitic estimations are guaranteed by compact analytical models, based on 2-D and 3-D field analysis. Finally, a global check on all distributed parasitics is performed. If the original constraints are not satisfied, the weights are updated based on the severity of the violation and routing is applied iteratively.

5.1 Performance-Driven Analog Routers

In ILAC[198], module generation was the emphasis of the layout system, while analog routing was considered a secondary issue. Analog routing approaches based on a channel routing style [51, Chp. 4] have been proposed in [50, 199, 83, 133]. In these works cross-coupling capacitances were avoided using appropriate heuristics in combination with the use of a global router forcing critically coupled nets to be in different channels, however no constraints on substrate and stray resistances were enforced. Only recently performance-driven channel routing tools have appeared [71] addressing this issue.

Despite the effectiveness of channel routing for some analog circuits, a different routing style may be preferable when highly irregular shapes are present in the layout. This is the case in a large number of circuit styles and in particular in MMICs. For these reasons *area routing* [200] is often adopted in mid-frequency high-performance analog IC design. A graph, called the *routing graph*, is defined over the routing workspace. The nodes of the graph define a partition on the wiring space, the edges link each two nodes if a wiring segment can be generated between the nodes. Routing graphs are usually arranged as a grid, which can be built on the workspace in a uniform or non-uniform fashion.

A cost function is usually defined on each graph edge according to global and/or local criteria and the cost of a path is defined as the sum of all edges connecting all the nodes in the set associated with the path. Hence the routing problem can be translated into that of searching a minimum cost path. The complexity of the search is $O(n^2)$, where n is the number of nodes associated with the workspace. ANAGRAM [67], a flexible area router based on the line expansion algorithm, is an example of such an approach. Crosstalk avoidance is controlled indirectly through net classification based on relative criticality, while no provision is given for controlling the *magnitude* of crosstalk interference or any kind of explicit parasitic control. The router in STAT [128] is based on a minimum-detour maze routing algorithm driven by a weighted sum of resistivity, vias and net length.

All area routing algorithms are based on some type of local optimization, this results in a lack of global view of the interconnection problem. Effects acting on a global scale, such as the substrate noise, cannot be modeled accurately, while relatively localized effects, such as capacitive parasitics, can be directly incorporated in the cost function thus insuring enforcement of a number of parasitics during routing. An example of a constraint-driven approach to routing is the tool ROAD [201], a maze router based on the A* algorithm operating on a relative grid with dynamic allocation. In ROAD [201] a performance-directed procedure defines a set of weights which relate to the direct impact of parasitics onto performance and to the tightness of specifications. The weights are then used by ROAD in the computation of a performance-based cost function, used to drive the algorithm to a solution attempting to meet all parasitic constraints. If one or more parasitic violations cause a performance specification to be violated, a new set of weights is computed and the offending line is ripped-up or the entire layout is re-done. This scheme recently evolved into one where weight evaluation was replaced by the use of *dynamic parasitic constraints* [202] during routing. This approach eliminates the dependency of the solution from routing net scheduling, since parasitic bounds are enforced softly, i.e. constraint surplus in other interconnect lines is used to compensate violations in wiring currently being built.

Routing in MMICs is complicated by the presence of distributed parasitics. In the remainder of the chapter we will show how a maze router based on the A* algorithm can be used to drive the solution to a configuration attempting to eliminate violations to all discrete and distributed parasitic constraints.

5.2 Maze Routing and the A* Algorithm

Maze routing is an area routing method based on the Lee-Moore algorithm [203]. In its basic form, one attempts to find a connection between two nodes s and t , the *source* and the *target*, using a two-phase procedure. In the first phase, or *propagation* phase, a *wave* is generated from s and propagated along the routing graph, until one of its fronts hits terminal t . In the second phase, or *backtrace* phase, a path connecting t to s is found starting from t and proceeding while a direction always orthogonal to the wave fronts is maintained.

The A* algorithm [204] is a general method for identifying the shortest path in the graph representing the workspace. Consider the diagram in Figure 5.1. Assume that

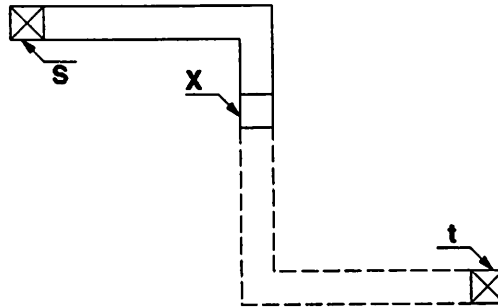


Figure 5.1: Propagation of path length estimate from source s to target t through node x

one wants to estimate the length of the path from the source to the target. Let node x be a node reached by a wave front. On x one can define a cost function $\ell(x)$ estimating the total length of the path connecting s and t

$$\ell(x) = g(x) + h'(x) , \quad (5.1)$$

where $g(x)$ is the cost of the known path from s to x and $h'(x)$ is an estimate of the cost $h(x)$ of an optimal path from x to t . In Manhattan style routing $h'(x)$ is generally given by the semi-perimeter distance between x and t while $g(x)$ can be computed exactly.

Figure 5.2 shows the complete A* algorithm. Let $R(x)$ be the set of nodes adjacent to x in the routing graph, Y the set of nodes selected for propagation, X the set of nodes reached but not yet propagated, and Z the set of all the nodes in the partial wiring path. When the algorithm completes, if the target has been reached, Z is the set of all the nodes on which propagation has been performed. The set of the nodes that fully define the path will be obtained running the backtrace phase on all the elements of Z .

The algorithm has been proved to be admissible, i.e. to always find minimum cost or optimum path if one exists, whenever $h'(x) \leq h(x)$, $\forall x$, where $h'(x)$ and $h(x)$ are the estimate and the true value of the length of the interconnect between x and the target [204]. The complexity of the algorithm is $O(n^2)$, where n is path length. However the average performance is generally much better, especially with medium size circuits. The speed of the algorithm is also determined by the accuracy of $h'(x)$ and by the structure of the graph structure [205]. Figure 5.3 shows the grid used in a typical A* based router for the computation of the path associated with the interconnect line being created. Every node of the grid represents a point in the workspace, while each edge is weighted by an amount proportional to the cost of a hypothetical interconnect line if it was crated in that

```

Y = {s};
X = ∅;
Z = {s};
repeat
  for x ∈ Y; // Propagate the nodes in Y
    X = X ∪ (R(x) - Z);
  if X = ∅
    exit; // Unroutable net
  else
    Y = {x ∈ X, such that ℓ(x) is min};
  if t ∈ Y
    exit; // Target reached
  else
    X = X - Y;
    Z = Z ∪ Y;
until forever

```

Figure 5.2: *Generic A* routing algorithm*

location. 3-D grids are used in the case of multiple layer interconnect style, while the grid is not allocated for a given layer in the areas where an obstacle exists for it. Segments within the path can be dynamically moved from one grid position to another and grid nodes can be modified to allow the implementation of new wire segments in a similar way as in [42].

The A* algorithm has been extended to multi-terminal nets, using for example a

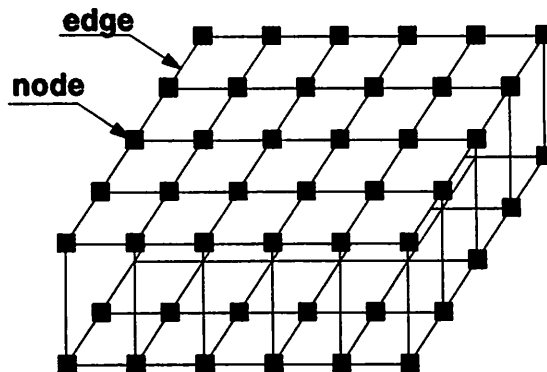


Figure 5.3: *Grid allocation in a typical maze router*

version of Prim's algorithm [206] for Minimum Spanning Trees (MST). The scheme consists of selecting one of the terminals in a net, the *seed*, as the first source from which the propagation is started. When the first path is completed, all the nodes in the path are promoted to the status of source and the algorithm is run again until no more target nodes are available.

If an estimate for $\ell(x)$ as defined in equation (5.1) is used in the cost function, the algorithm minimizes all interconnect lengths in the circuit. Moreover, the minimization is performed over all nets equally aggressively. The algorithm is appropriate in most digital designs, when area and delay optimization is needed. In analog circuits on the contrary, these factors are generally not the only concern and trade-offs must be drawn between a number of often conflicting requirements, such as stray resistances, substrate capacitances, etc. To cope with this problem a number of solutions have been proposed, see review in [201]. One solution, similar to the one we adopted in this dissertation, consists of augmenting the wiring length estimator $\ell(x)$ by a factor proportional to the sum of all the violations presently accumulated in the layout. Every violation is *weighted* by an amount relating to the sensitivity of performance with respect to the corresponding parasitic and to the tightness of the specification. An additional factor is added to account for the local area crowding. Area crowding is a function of the *congestion ratio* R between the needed space for the new wire being built, and the effective space available on the sides of edge x . The area crowding factor A_c is given by

$$A_c = \begin{cases} 0, & \text{if } R \leq 1 \\ A_{max} & \text{if } R > 1 \end{cases} \quad (5.2)$$

where A_{max} is a large constant. Later in the chapter we will see how the concept of area crowding and parasitic constraint violation will be extended to the problem of high-frequency routing.

5.3 Routing of RF Circuits and MMICs

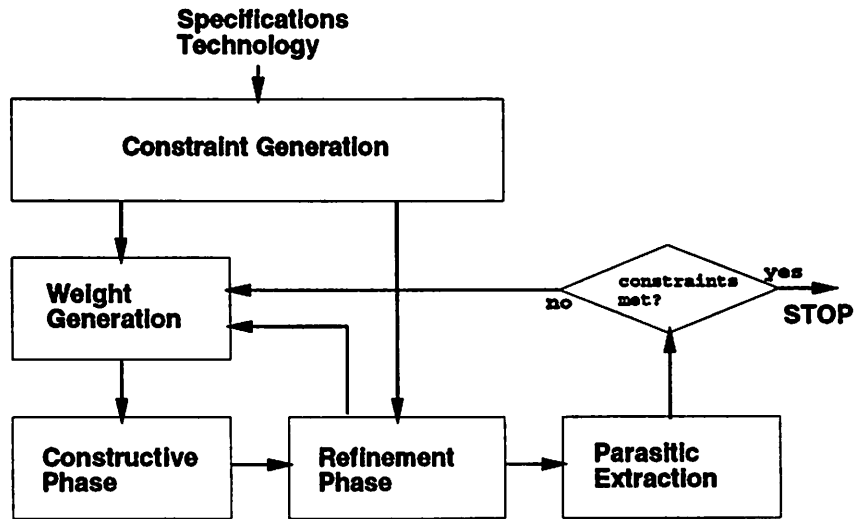
Great effort has been devoted over the years to create general purpose CAD tools for schematic design and optimization of RF and microwave circuits, e.g. [207, 208]. Layout synthesis has not received comparable attention in the literature due to the inherent complexity of the problem and to the lack of designs whose size could justify a CAD based

approach. More recently however, the increasing complexity of monolithic RF ICs and MMICs suggests that an automated approach to the physical synthesis is preferable for efficiency, reliability and yield considerations.

LINMIC[209, 207] proposed a knowledge-based interactive approach aimed at designers with minimum expertise by providing aids for low frequencies design of MMICs. However, due to the lack of an explicit reference to performance, the synthesis process could result in a large number of time-consuming iterations necessary to satisfy specifications. Other semi-automated topology-driven approaches to the routing of MMICs have also been proposed [210, 211]. These systems include template-based routines for the abutment of pre-defined cells implementing devices as well as interconnect. The knowledge of the relative position of all the cells provides the starting point of the layout realization, thus strongly limiting the flexibility of the approach and its applicability to complex designs.

We propose a constraint-based approach to the routing of MMICs. The flow diagram of the approach is shown in Figure 5.4. Firstly, high-frequency performance specifications are mapped onto a set of bounds on all classes of distributed parasitics. Then, using sensitivity analysis, a set of weights is calculated for the area router. The role of the weights is to control a cost function which penalizes those realizations with highly critical parasitics. Layout synthesis of RF and microwave circuits almost always requires that the dimensions of some interconnect lines be fixed. However, length constraints on interconnect cannot be effectively enforced during this phase. Hence, the routing or *constructive phase* is followed by a *refinement phase*. The refinement consists on progressive expansion of all nets simultaneously thus allowing enforcement of all net constraints while no new violations are created on the remaining parasitic constraints. If infeasibility is detected a new set of weights is generated and the cycle is repeated. At the completion of the layout, the entire circuit is checked against constraint violations so as to verify that all performance specifications are met. Parasitics are directly extracted from all physical geometries and estimated by means of *ad hoc* analytical models based on 2-D and 3-D field analysis. In case a specification violation occurs, the parasitics responsible for the violation are identified and a sensitivity-based scheme is used to create a new set of weights. The cycle is then restarted. The finite step loop ends when all specifications are met.

There are several advantages to this approach. A constraint-based approach to the layout of RF and microwave circuits helps drastically reduce the number design iterations by carefully modeling and controlling all relevant parasitic effects in the circuit. For a given

Figure 5.4: *Flow diagram of the tool*

technology, compact and accurate models for physical parasitics are derived only once. Hence, the synthesis and analysis of a circuit is efficient and can be effectively used within a larger semi- or fully automated design cycle. Furthermore, using a different cost weighting scheme, rapid circuit re-design for different realization and performance requirements can be efficiently accomplished. This is particularly useful in the design of large scale MMIC libraries.

5.4 Parasitic Modeling and Constraint Generation

Accurate interconnect modeling is a fundamental requirement of a constraint-driven approach to the routing of RF and microwave circuits. For reasons of efficiency closed formulae and analytical models for interconnect lines and all relevant parasitics are desirable. Since the area of application of this work is MMICs, all interconnect lines are modeled as microstrip transmission lines. Parasitic effects such as inductive and capacitive crosstalk are modeled in terms of the degradation induced on the characteristic impedance Z_0 and loss α . Alternatively at low frequencies discrete (R,C,L) parasitics can be used. Analytical models of all considered parasitics are obtained by fitting appropriate mathematical expressions to data obtained from 2-D or 3-D field solvers as proposed by [193].

Interconnect discontinuities are modeled using discrete components, while radia-

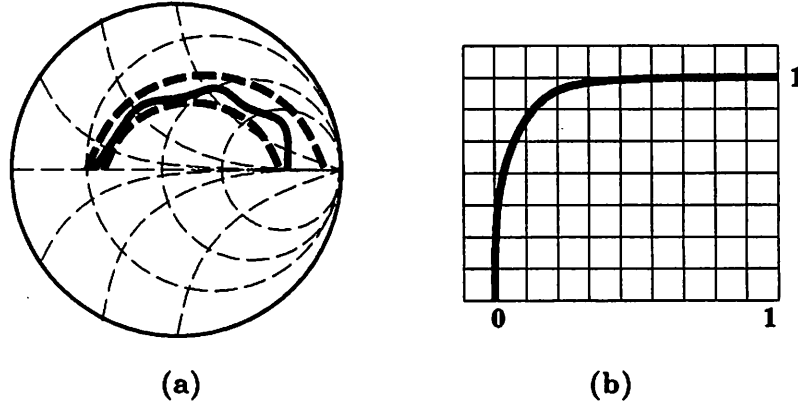


Figure 5.5: (a) Microwave specification; (b) Flexibility function for constrained optimization

tion and surface-wave propagation effects have been neglected due to the the relative small circuit size if compared with the signal wavelengths [212]. If needed, surface-wave propagation could be easily modeled using the closed forms reported in [213]. Substrate-dependent losses have been taken into account in the full model. A summary of the formulae used in our approach for estimating Z_o and α can be found in Appendix E.

The constraint generation techniques outlined in [116] and [2] need be modified to account for the distributed nature of parasitics in microwave circuits. For a given a performance K_i , let us define *performance specification* the set of inequalities which determine lower- and upper-bounds for K_i and the range of frequencies for which they are valid. For example, consider the input reflection coefficient S_{11} illustrated in Figure 5.5a. The solid line is the actual value of S_{11} obtained from the complete layout after extraction and the dotted lines are the frequency dependent specifications to performance S_{11} , or

$$\begin{aligned}
 |L_0| \leq |S_{11}| \leq |U_0|, \quad \angle L_0 \leq \angle S_{11} \leq \angle U_0, \quad \text{for } f_0 - \Delta f_0 \leq f \leq f_0 + \Delta f_0 \\
 |L_1| \leq |S_{11}| \leq |U_1|, \quad \angle L_1 \leq \angle S_{11} \leq \angle U_1, \quad \text{for } f_1 - \Delta f_1 \leq f \leq f_1 + \Delta f_1 \\
 \dots \\
 |L_n| \leq |S_{11}| \leq |U_n|, \quad \angle L_n \leq \angle S_{11} \leq \angle U_n, \quad \text{for } f_n - \Delta f_n \leq f \leq f_n + \Delta f_n
 \end{aligned}$$

In general, for a performance set $\{K_i\}$, $i = 1, \dots, N_k$ and a set of parasitics $\{p_j\}$, $j = 1, \dots, N_p$, *parasitic constraint generation* is defined as the process of creating an inequality constraint on a subset of all parasitics $\{p_j\}$: $p_j \leq p_j^{(bound)}$, to guarantee the fulfillment of all performance constraints in the entire frequency range or $\Delta K_i(f) \leq \overline{\Delta K_i(f)}$, $\forall i = 1, \dots, N_k, \forall f \in [f_{min}, f_{max}]$. $\Delta K_i(f)$ represents the total and $\overline{\Delta K_i(f)}$ the maximum al-

lowed degradation of performance K_i with respect to all parasitics within the entire range of operation. Parasitics such as cross-couplings, characteristic impedance degradation, losses and microstrip line length tolerances are constrained in this fashion. Constraints on parasitics related to short and open microstrip terminations are derived from the above constraints as described in Appendix E.

Assuming that performance is differentiable at and near its nominal value at all frequencies, it can be represented as a linear combination of its sensitivity with respect to all distributed parasitics at all frequencies. Since parasitics can contribute constructively as well as destructively to performance, positive and negative sensitivities must be considered separately for constraint computation. Positive and negative components of the performance degradation ΔK_i^+ and ΔK_i^- with respect to all parasitics belonging to the set $\{p_j\}$ are, in first approximation

$$\Delta K_i(f)^+ = \sum_j S_{i,j}(f)^+ p_j, \quad \Delta K_i(f)^- = \sum_j S_{i,j}(f)^- p_j.$$

Technology-related process variations can be taken into account by replacing nominal values of sensitivities with worst-case values [116] or by finding bounds on parasitic and device mismatches [2].

To insure that all relevant parasitics are considered during the optimization, each interconnect line to be implemented in the circuit is modeled as a microstrip line with inductive and capacitive coupling with all the other nets. A numerical sensitivity analysis is performed for each performance function using simulator MNS [210]. All parasitics, whose cumulative contribution to performance degradation is negligible, are automatically discarded. Bounds on critical distributed parasitics are calculated by using constrained optimization, the objective being the maximization of the flexibility of the layout generation process. The objective function of the optimizer, PARCAR, is a polynomial of second order monotonic in the range from zero to one, as shown in Figure 5.5b. The goal is to obtain large bounds on critical parasitics, while at the same time unnecessarily loose bounds on uncritical parasitics are allowed to be tightened.

RF and microwave circuits rely for functionality on transmission lines with a specific length. However tolerances ΔL_j^\pm from nominal value L_j should be permitted to provide more flexibility to the router

$$\ell_j = L_j, \quad \Delta \ell_j^+ \leq \Delta L_j^+, \quad \text{and} \quad \Delta \ell_j^- \leq \Delta L_j^-.$$

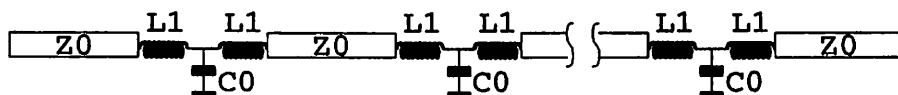


Figure 5.6: *Interconnect model for microstripline with multiple bends*

While L_j is determined by design, constraints ΔL_j^{\pm} must be computed numerically. Using MNS the sensitivity of each performance with respect to length variations can be quantified, thus, using constrained optimization, constraints on the tolerances are calculated.

Parasitics involving the detailed realization of interconnect such as bends, gaps and steps need be considered in a somewhat different way. A bended interconnect realization is shown in Figure 5.6, where the bends have been replaced with appropriate models. See Appendix E.

Consider the problem of finding a constraint on the number of bends in the microstrip line. Suppose a microstrip line of nominal length L is partitioned into N segments of length L/N each. Let us model $N - 1$ bends as in Figure 5.6. Due to the repetitive character of the model, performance sensitivities with respect to the discrete components $S_{i,L/C}$ of each bend are necessarily equal. Therefore the cumulative degradation ΔK_i of N bends is

$$\Delta K_i = \sum_{j=1}^{N-1} (S_{i,L_1} L_1 + S_{i,C_0} C_0) = (N - 1)(S_{i,L_1} L_1 + S_{i,C_0} C_0)$$

Consequently only one (L_1, C_0) pair must be considered by the optimizer. After the optimization two scenarios are possible: (a) parasitics are not critical and therefore they have been eliminated. In this case, given the maximum and minimum values for C_0 and L_1 , $N^{(bound)}$ can be computed as following

$$N^{(bound)} = 1 + \min\{\lfloor C_0^{(max)}/C_0^{(min)} \rfloor, \lfloor L_1^{(max)}/L_1^{(min)} \rfloor\}.$$

Note that if $C_0^{(min)} = 0$ and $L_1^{(min)} = 0$, the constraint on N is ∞ , thus it can be neglected. (b) parasitics are critical and bounded. In this case $N^{(bound)}$ can be obtained by replacing the maximum value of C_0 and L_1 with its bound.

$$N^{(bound)} = 1 + \min\{\lfloor C_0^{(bound)}/C_0^{(min)} \rfloor, \lfloor L_1^{(bound)}/L_1^{(min)} \rfloor\}.$$

5.5 Routing Phases

For reasons of flexibility and algorithmic reliability, an area routing scheme has been used in our approach. In this scheme all inequality constraints can be easily implemented in the cost function driving the router. Equality constraints however cannot be handled effectively via a cost function, due to the instability they induce in the algorithm [204, Chp.3]. Thus, the routing scheme has been partitioned into two phases. During the first phase, the *constructive phase*, all inequality constraints, i.e. constraints related to interconnect parasitics, are enforced. The second phase or *refinement*, enforces all equality constraints while no new parasitic violations are introduced. Both phases have been implemented in a tool called CORAL.

5.5.1 Constructive Routing

The first phase of the routing scheme consists of a maze router based on the A* algorithm [214]. The A* algorithm is based on a heuristic estimation of the cost of a path connecting the propagation node on the grid x and a terminal target. An optimal path for a net j is defined as the path minimizing a cost function $f(x)$. $f(x)$ is the estimate of the wiring length ℓ weighted by a factor proportional to interconnect crowding $K_c(x)$ and to the sum of all violations to parasitic constraints.

$$f(x) = \ell(x) \left(1 + K_c(x) + w_j^R \frac{Viol[R_j(x)]}{R^0} + w_j^C \frac{Viol[C_j(x)]}{C^0} + \frac{1}{C^0} \sum_{jk} w_{jk}^C Viol[C_{jk}(x)] + w_j^Z \frac{Viol[Z_j(x)]}{Z^0} \right),$$

where

$R_j(x)$ = integral of the estimated transmission line loss of net j at x

$C_j(x)$ = integral of the estimated substrate capacitance of net j at x

$C_{jk}(x)$ = integral of the estimated coupling between nets j and k at x

$Z_j(x)$ = local estimated characteristic impedance from nominal for net j at x

Every violation ($Viol[.]$) is calculated as the difference between the estimate of the parasitic component and its pre-computed constraint, when this value is positive, otherwise it is set to zero. See Appendix E. Parameters R^0 , C^0 and Z^0 are normalization factors. w_j^R , w_j^C , w_{jk}^C and w_j^Z are specific weights, calculated according to the equation

$$w_j = w_0 \sum_{i,j} \left[\frac{S_{i,j}(f)^-}{\Delta K_i(f)^-} + \frac{S_{i,j}(f)^+}{\Delta K_i(f)^+} \right]$$

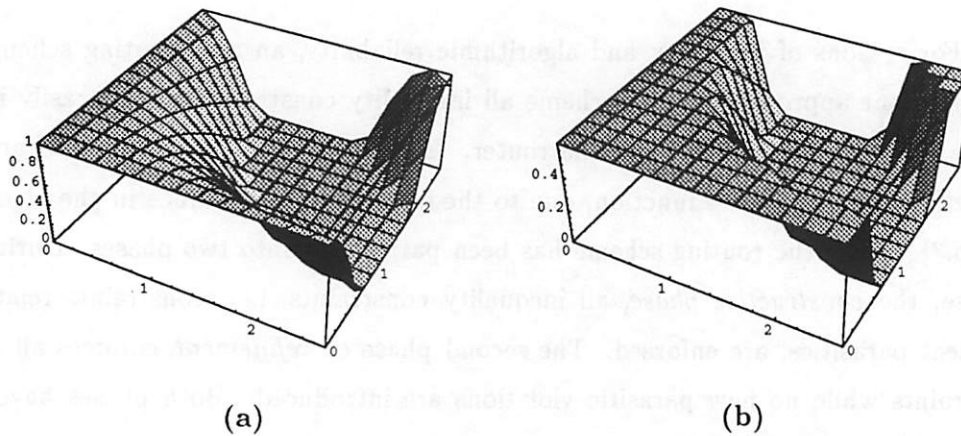


Figure 5.7: Expansion of wiring in the workspace (a) with coarse and (b) with tight constraints

where $\overline{\Delta K_i(f)}^\pm$ represents the specification of performance K_i at frequency f , $S_{i,j}(f)^\pm$ its sensitivity with respect to parasitic p_j and w_0 a normalization factor. As an illustration consider the cost of propagating a wavefront for a given net. When the constraints on parasitics associated with the net being constructed are loose, the cost function will lightly weight the cost of propagation into areas where violations will occur (Figure 5.7a). On the contrary, when constraints are tight (Figure 5.7b) propagation causing constraint violations can be generally tolerated by a lesser extent.

The constructive phase yields interconnects with a minimized length ℓ . Since ℓ cannot be further reduced, the algorithm stops if loss and substrate capacitance violations occur or if the maximum transmission line length L is exceeded. This technique can also be used for implementing stubs simply by creating a virtual target in an area where no parasitic violations can occur. Figure 5.8a,b shows the gradual construction of a stub by generating a wiring segment reaching a *virtual target* from any given source. The virtual target is positioned as far as possible from objects that may induce violations onto one or more parasitic constraints as formally defined in section 5.5.2. The routing order of the nets is determined automatically giving priority to the wiring of stubs and of nets on which parasitic constraints are tightest. This is done not to compromise the ability of the router to meet all parasitic constraints by routing first non-critical nets.

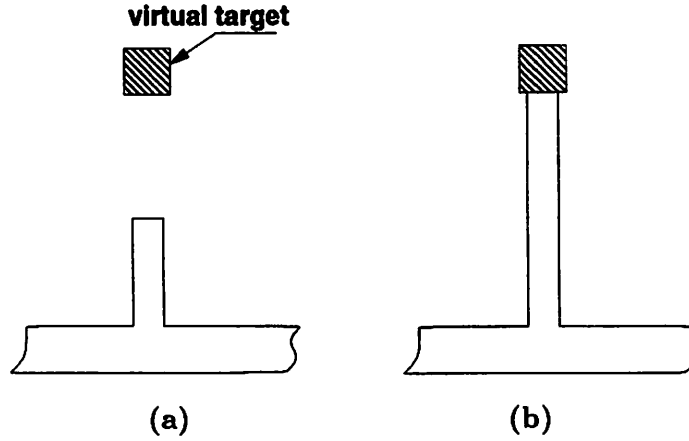


Figure 5.8: *Stub construction: (a) development from source; (b) completion*

5.5.2 Refinement

The refinement phase is responsible for the enforcement of all equality constraints while insuring that no additional violation be introduced in the layout. Refinement is only applied to the set of all nets for which at least a violation exists, call C such set. Consider the constraint on microstripline length of net j , L_j . Since previously obtained length is guaranteed to be smaller than the constraint, the interconnect needs be expanded. However the expansion must occur inside a space where no constraint violations are possible. Call this space *feasibility zone* of net j , or \mathcal{F}_j . \mathcal{F}_j is defined as the intersection of all spaces $\mathcal{B}_j(p_k)$, for which parasitic p_k associated with net j does not exceed its predefined constraint.

Assume that each parasitic can be expressed in form of a n^{th} order polynomial $\mathcal{P}_n(\mathbf{x}, \mathbf{s}, \mathbf{V})$, where \mathbf{x} is the location of a point in the interconnect and \mathbf{s} is the position of any objects responsible for p_k . \mathbf{V} is the vector of all known design parameters (interconnect width, via size, etc.). Then, the location of $\mathcal{B}_j(p_k)$'s boundaries is the locus of all \mathbf{x} that solve

$$\mathcal{P}_n(\mathbf{x}, \mathbf{s}_0, \mathbf{V}) - p_k^{(bound)} = 0, \quad (5.3)$$

for given object position \mathbf{s}_0 and parameter \mathbf{V} . As illustration consider the effect of a via structure on the characteristic impedance of the microstrip line depicted in Figure 5.9a. Given the physical dimensions of both objects, a model for the deviation of microstrip impedance ΔZ_o is derived. See Appendix E. Figure 5.9b shows a plot of ΔZ_o as a function of the location of an hypothetical transmission line relative to the via. Based on a constraint

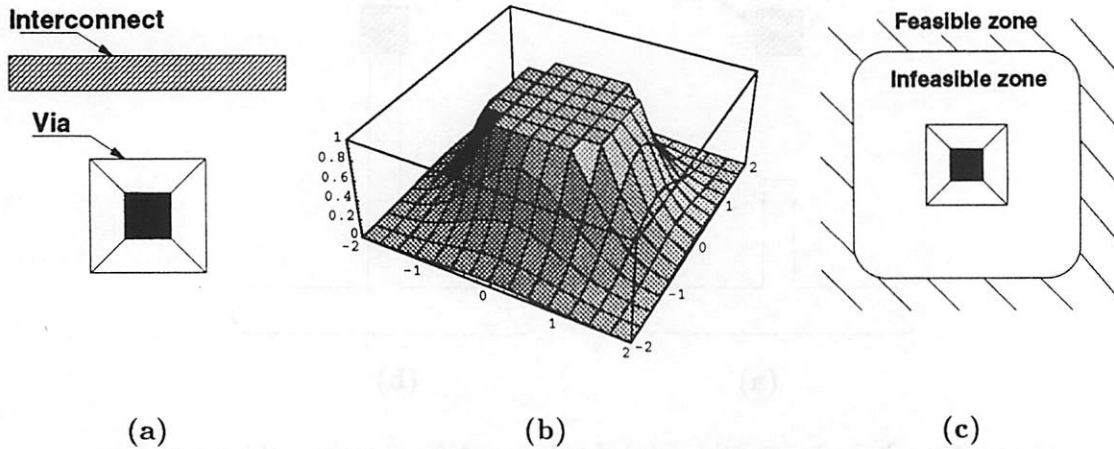


Figure 5.9: Effects of a via structure on a to-be-built interconnect line. (a) Structure set-up; (b) Interconnect characteristic impedance deviation as a function of the location relative to the via; (c) Feasibility zone

on ΔZ_o , one can derive the feasibility zone for the given transmission line as shown in Figure 5.9c. The resulting space $\mathcal{B}(\Delta Z_o)$ is a two-dimensional sphere centered in s_o with radius d_o , where d_o is computed solving Equation (5.3) for a given constraint $\Delta Z_o^{(bound)}$. Hence any implementations of the microstrip outside this space will satisfy the constraint. Given P_j , the set of all constrained parasitics relevant to net j , \mathcal{F}_j becomes

$$\mathcal{F}_j = \bigcap_{p_k \in P_j} \mathcal{B}_j(p_k).$$

The boundaries of feasibility zone \mathcal{F}_j are often a complex function of relative position and size of the surrounding layout objects and by the constraint imposed on each parasitic p_k as shown in Figure 5.10a (dotted lines).

If the initial number of bends b exceeds the maximum allocated number the refinement phase terminates. If, on the other hand, $n = N^{(bound)} - b \geq 4$, then the expansion algorithm is applied. Since only Manhattan expansion is allowed, the feasibility zone for each net is horizontally sliced in correspondence of each vertex as shown in Figure 5.10b. Between each pair of horizontal cuts a rectangle is built of width equal to the minimum zone diameter between the cuts. Call R the set of all the rectangles non-adjacent to source and target obtained in this fashion. Since it is preferable to expand interconnect as uniformly as possible in order to keep it at the center of the zone, the maximum possible number of rectangles r is selected for expansion, where $r = \lfloor \frac{n}{4} \rfloor$. This ensures that the minimum possi-

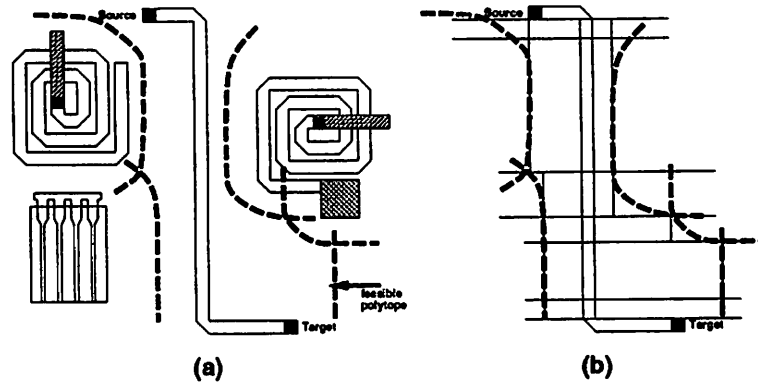


Figure 5.10: *Distributed parasitics acting on interconnect determine feasibility zones*

ble horizontal expansion is performed, thus maintaining the distance between interconnect and zone boundary as large as possible.

The interconnect can be expanded in two opposite directions in two adjacent rectangles to maximize the length increase $\Delta\ell$, or in two equal direction to minimize the number of bends. Hence, a large number of combinations for the expansion rectangles exists that obtains the desired expansion. The problem of finding an appropriate subset \bar{R} of R which maximizes the total expansion $\Delta\ell$ can be formulated as

$\underset{\bar{R}}{\text{maximize}}: \Delta\ell \tag{5.4}$
<p>subject to:</p> $n \leq N^{(\text{bound})}$

This problem can be solved using exhaustive enumeration techniques or linear programming. Due to the low number of vertices per zone, in CORAL the former solution has been adopted. An expansion step $e = \min_{j \in C} \Delta\ell_j / K$ is selected, where K is a constant proportional to the worst-case number of expansion steps of the algorithm. For each net j the set of all expansion rectangles \bar{R}_j is calculated and each net is expanded by e . The expansion is continued until either a net j reaches its nominal length L_j or the location of a zone vertex changes. If the first event occurs, net j is dropped from C , e is recomputed and the expansion continues. Otherwise, \bar{R}_j is recomputed for all nets in C and the expansion proceeds until an infeasibility is detected, i.e. the interconnect line crosses its assigned zone. In the latter case the partially expanded layout is analyzed for performance violations, a

```

set_initial_expansion_step;
set_initial_expansion_direction;
set_constrained_nets;
repeat
  compute_feasibility_boundary;
  identify_expansion_rectangles;
  if max_expansion < required_expansion
    exit;
  expand_segments;
  update_feasibility_zones;
  if no_feasibility
    exit;
  update_constrained_nets;
  update_expansion_step;
until all_constraints_met OR infeasibility_detected

```

Figure 5.11: *Refinement algorithm in CORAL*

new set of weights is generated and routing is repeated.

All parasitics responsible for the violation can be easily identified, thus the set of weights associated with these parasitics can be modified so as to insure that a new routing attempt with a modified cost function will reach a satisfactory solution. In our approach, weights are increased by an amount Δw_j , inversely proportional to the severity of the violation that the corresponding parasitics induced

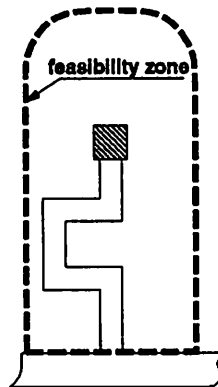
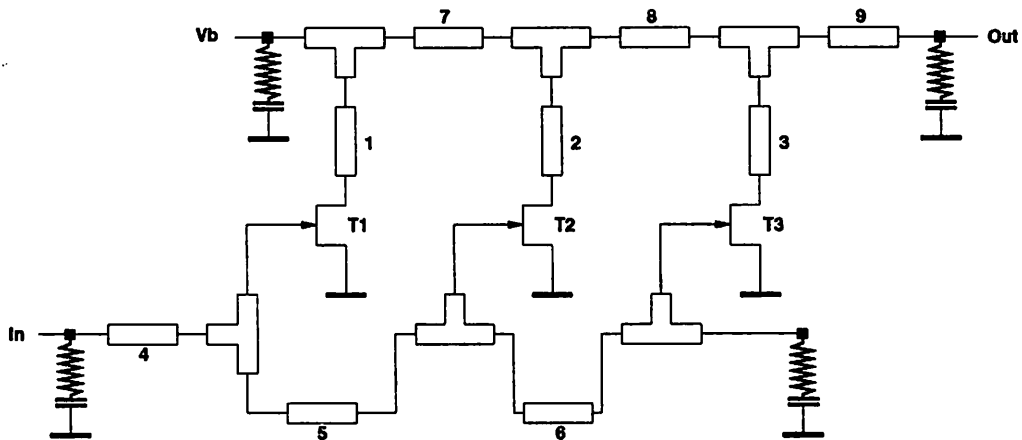
$$\Delta w_j = \frac{1 - w_j}{m_0} \sum_{i,f} \left[\frac{S_j^i(f)^-}{\Delta K_i(f)^- - \Delta K_i(f)^-} + \frac{S_j^i(f)^+}{\Delta K_i(f)^+ - \Delta K_i(f)^+} \right],$$

where m_0 is a normalization factor. The sequence of constructive/refinement steps terminates when no more constraint violations are present in the layout or when the maximum number of iterations is reached. The refinement algorithm is summarized in Figure 5.11.

Stubs are treated similarly. Consider the structure of Figure 5.8b. The initial position of the virtual target \mathbf{x}_T is determined so that it will be located in the *center* of the feasibility zone.

$$\mathbf{x}_T = \frac{1}{N_{\mathcal{F}_j}} \sum_{k=1}^{N_{\mathcal{F}_j}} \mathbf{x}_k(\mathcal{F}_j),$$

where \mathcal{F}_j is the feasibility zone associated with the target, $\mathbf{x}_k(\mathcal{F}_j)$ is a point in the boundary

Figure 5.12: *Expansion of a stub within feasibility zone*Figure 5.13: *Schematic of TWA*

of \mathcal{F}_j and $N_{\mathcal{F}_j}$ is the number of points in the boundary. After the initial stub is built, refinement is used to adjust the length of the stub using the feasible expansion algorithm. An example of the expansion algorithm applied to a stub is shown in Figure 5.12.

5.6 RF and Microwave Routing: A Case Study

Consider the traveling wave amplifier (TWA) shown in Figure 5.13. The specifications are listed in Table 5.1. In this circuit the control of transmission coefficient S_{21} over a range of frequencies spanning 18 GHz is the main objective. To achieve the objective, the circuit is first decomposed into interconnect and devices. The simulator MNS is used to

freq. range (GHz)	0-1.5	1.5-18.0	18.0-26.0
$ S_{21} $ (dB)	> 3	> 3	> -10
	< 7	< 5	< -7

Table 5.1: *Performance specifications for TWA*

freq. range (GHz)	0-1.5	1.5-18.0	18.0-26.0
1: $\Delta L_{57}, \Delta L_{68}$	<2.5%	<1.7%	<1%
2: # bends in 1,2, and 3	4	4	4
3: # bends in 7 and 8	16	16	16
4: $\Delta Z_1, \Delta Z_2, \Delta Z_3$	<0.2%	<0.1%	<0.4%
5: $\Delta \alpha_1, \Delta \alpha_2, \Delta \alpha_3$	<0.5%	<1%	<5%

Table 5.2: *Constraints on critical interconnect lines*

compute the sensitivity of S_{21} with respect to the following features related to interconnect, namely: length, characteristic impedance, and loss.

From the sensitivities, using the approach outlined in chapter 3 and in this chapter a set of constraints on each of these features is computed. Table 5.2 lists the constraints computed using PARCAR. Notice that terms ΔL_{xy} relate the maximum length mismatch between interconnect line x and y .

The constructive phase of CORAL builds minimum length feasible interconnect between the pre-placed objects as shown in Figure 5.14. After this phase only constraints 4 and 5 have been enforced. Performance S_{21} is highly sensitive with respect to terms Z_1, Z_2 , and Z_3 which are in turn highly sensitive to the vias present within each transistor implementation. Hence, interconnects 1,2 and 3 branch from T_1, T_2 and T_2 at 90° angle until a safe distance is reached for connecting with segments 7,8 and 9. A longer implementation is not selected by the algorithm since it would have violated the constraint on the loss coefficients of 1, 2, and 3. The refinement phase of CORAL enforces the length of all the critical interconnects within the tolerances permitted by constraints 1, 2 and 3. Notice, how both constraints on the number of bends were respected by appropriately implementing interconnects 1,2,3 and 7,9. All phases of the assembly and the corresponding CPU times are listed in Table 5.3. The final layout is shown in Figure 5.15 and the performance evaluation

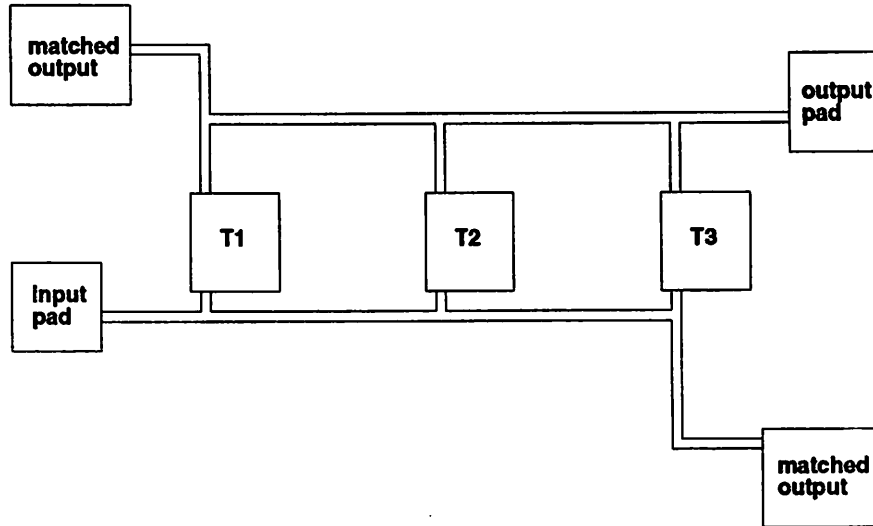


Figure 5.14: *Layout after CORAL's constructive routing*

tool	CPU time (sec)
MNS	31
PARCAR	351
CORAL	53

Table 5.3: *CPU times required for the synthesis of TWA on a DEC Station 5000/240*

freq. range (GHz)	0-1.5	1.5-18.0	18.0-26.0
$ S_{21} $ (dB)	6.26	4.30	-8.21

Table 5.4: *Estimated performance of circuit TWA after layout completion*

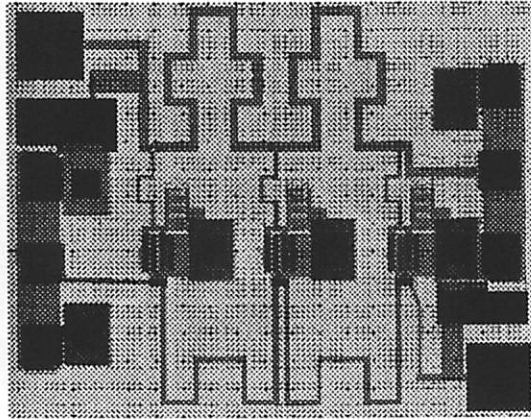


Figure 5.15: *Final layout of TWA*

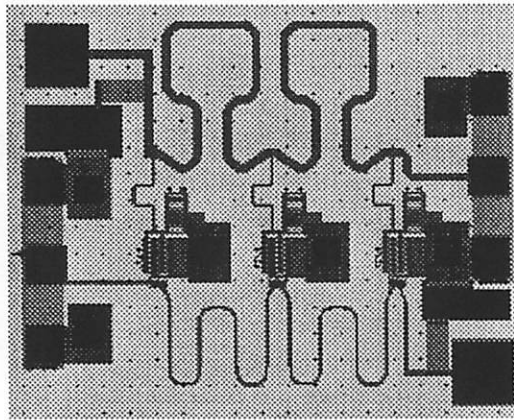


Figure 5.16: *Manual layout of TWA*

of the circuit after extraction is summarized in Table 5.4. As expected the performance specification was met at all frequencies. The circuit was fabricated and tested at the HP-EEsof Labs in Santa Rosa. The fabricated chip passed the specification test and the yield of the circuit was comparable to an identical circuit laid out manually (discrepancy $< 1\%$ for medium volume of fabrication). Figure 5.16 depicts the same circuit as an experienced designer laid it out.

Chapter 6

Symbolic Compaction

*D'anime nude vidi molte gregge
che piangean tutte assai miseramente,
e pareva posta lor diversa legge.*

*Supin giacea in terra alcuna gente,
alcuna si sedea tutta raccolta,
e altra andava continuamente.*

*Quella che giva intorno era pi molta,
e quella men che giacea al tormento,
ma pi al duolo avea la lingua sciolta.*

Dante Alighieri, "Inferno", Canto XIV

In this chapter we describe the symbolic compaction tool, which represent the last phase of the physical assembly. We show how analog constraints can be enforced during compaction, we analyze the impact of these constraints on known compaction algorithms and we present alternative solutions for efficient handling of the compaction problem. The compaction methodology is illustrated throughout the description with an example used to highlight the main features of the approach.

A compactor is in general used to reclaim empty area that has been allocated for interconnections but has not been used by the final implementation. To be useful a

compactor must reclaim the area while violating no constraints. A compactor can also be used as a *spacer*, i.e. a tool to adjust the layout by moving components so that all the design rules constraints (DRC) are satisfied. In this mode of operation the layout synthesis phase is allowed to introduce (in general minor) DRC violations in favor of faster and more efficient placement and routing procedures. Thus, in addition to reducing chip area, compaction also improves the efficiency and robustness of the entire physical assembly process.

6.1 Compaction Problem Formulation

In this section the basic formulation of the compaction problem in the presence of a complex set of constraints is outlined. For a detailed description of the problem and the algorithms, see [215] and [216, Ch.10.2]. The symbolic layout compaction problem can be formulated as a constrained minimization problem, where the target function is area:

$\text{minimize : } (x_R - x_L)(y_T - y_B) \tag{6.1}$
<p>subject to:</p> $\{C\}$

where

x_R = horizontal position of the rightmost circuit element

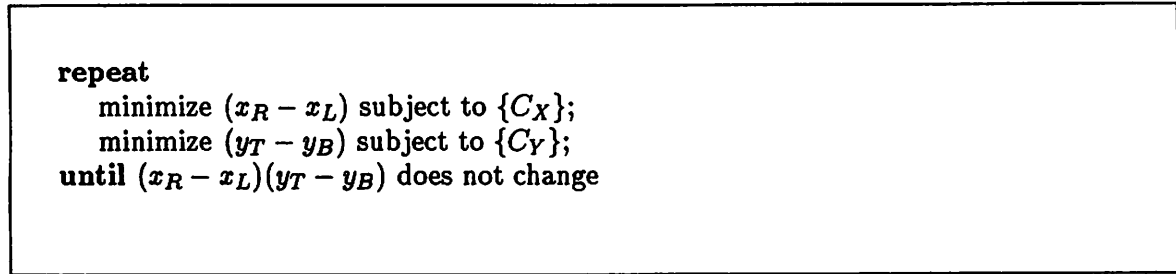
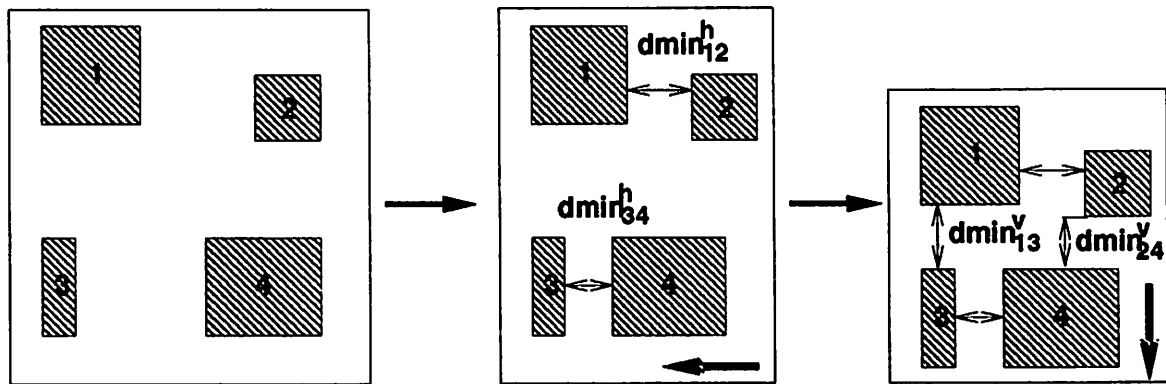
x_L = horizontal position of the leftmost circuit element

y_T = vertical position of the topmost circuit element

y_B = vertical position of the bottommost circuit element

$\{C\}$ is a set of constraints.

This problem has been proven to be NP-hard [216]. Nevertheless, a large number of heuristics can yield good quality layouts. Although interesting techniques for two-dimensional compaction [217, 218] have been developed, mono-dimensional compaction is still the most practical approach because it is simple, computationally efficient, and, in most cases, it yields good area reduction. A mono-dimensional compactor finds an approximate solution to this problem by separating the horizontal and vertical components. The algorithm is formulated in Figure 6.1. Here $\{C_X\}$, $\{C_Y\}$ are sets of horizontal and vertical constraints, respectively. Heuristics exist interrelating the two dimensions by incor-

Figure 6.1: *Mono-dimensional compaction algorithm*Figure 6.2: *Orthogonal mono-dimensional compaction iterations*

porating wire balancing algorithms [219] and introducing jogs [220]. Figure 6.2 illustrates the method pictorially. The minimum allowed vertical (horizontal) distance $d_{min_{ij}^v}$ ($d_{min_{ij}^h}$) between each pair of objects i and j in Figure 6.2 is determined by DRC and parasitic considerations [221].

Mono-dimensional compaction is usually formulated as a longest-path problem on a directed graph, which is solvable in polynomial time. However, not all analog circuit constraints (for example, symmetry constraints) can be expressed in a form suitable for a Constraint-Graph (CG) algorithm. A perturbation approach was proposed in [222], where the graph is initially solved without constraints, and symmetries are gradually enforced. This approach however can become computationally expensive with a large number of symmetry constraints. Furthermore, false over-constraints may arise depending on the order in which the symmetric objects are processed, especially when multiple symmetry axes are present. A different approach to compaction with symmetry constraints is to use Linear Programming (LP) [68] or Integer Programming (IP) [223]. The main drawback of

```
run_CG;  
run_LP;  
round_off_coordinates;  
verify_constraints;  
switch_compaction_axis;
```

Figure 6.3: *Iterative mono-dimensional compaction algorithm*

this approach is the computational intensity required when a realistic number of constraints is considered. In [68], algorithmic complexity is reduced somewhat by collapsing all objects not directly interacting with symmetric items in the layout, generating “super-constraints” which are solved by a linear program. With regard to parasitic control, we are not aware of other approaches taking analog or mixed-signal related constraints into account.

6.2 Compaction with Analog Constraints

Our approach to the enforcement of analog-specific constraints during compaction combines a CG algorithm and a linear program in sequence. The first phase enforces all design rules, while the second phase uses the result of the first as a starting point to enforce the remaining constraints. The method has been implemented in a tool called SPARCS-A. The various phases of the algorithm are illustrated in Figure 6.3 .

By its nature, the LP algorithm can return a result which is not aligned with the layout integer grid. Hence, a round-off procedure is needed to adjust the coordinates of those elements laying outside the critical path. In Appendix B we show how this procedure is justified, thus allowing us to use LP instead of a more time-expensive IP based approach. Finally, a check is performed on the result and the compaction is iterated in the orthogonal axis until a set of stopping conditions is met.

Using the CG solution as starting point is key to a significant speed-up in the solution of the linear problem. Control over cross-coupling capacitances is implemented by modifying the CG before computing the longest path. Proper distances between parallel interconnection edges are kept to maintain cross-coupling capacitances below their bounds. This is achieved by employing a heuristic which adds extra spacing between wire segments.

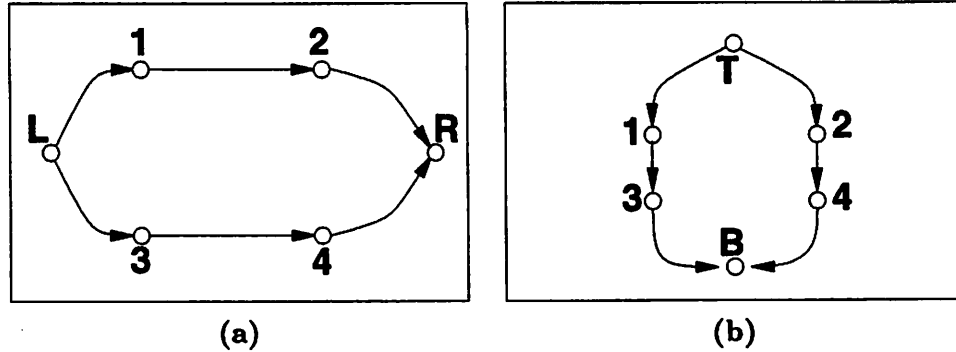


Figure 6.4: *Constraint graphs associated with (a) horizontal and (b) vertical constraints. The symbols L/R and T/B relate to the left/right and top/bottom coordinates respectively*

6.3 Constraint Enforcement Techniques

6.3.1 DRC

Consider first the Constraint-Graph (CG) longest-path algorithm [216, Ch.10.2]. The pattern of component connectivity and minimum separations required by DRC is described as a weighted, directed graph. Two graphs are actually used, one for the vertical and one for the horizontal direction. Figure 6.4 shows the graphs associated with the structures in Figure 6.2. Each geometry in the layout is represented in the graph by a vertex. For simplicity, but without loss of generality, we consider only the horizontal graph. Let x_1 and x_2 be the horizontal coordinates of vertices 1 and 2, respectively. Constraints of the form

$$x_2 - x_1 \geq K; \quad x_2 - x_1 \leq K \quad (6.2)$$

are represented as edges of weight K and $-K$ respectively, directed from vertex 1 to vertex 2. **2. Maximum spacing constraints**

$$|x_2 - x_1| \leq K_{max} \quad (6.3)$$

can be reduced to a pair of constraints of the type (6.2) by using the equivalent expressions

$$x_2 - x_1 \leq K_{max} \quad \text{AND} \quad x_1 - x_2 \leq K_{max}$$

Minimum spacing constraints

$$|x_2 - x_1| \geq K_{min} \quad (6.4)$$

are usually solved by requiring that the relative position of the objects corresponding to vertices x_1, x_2 remain unaltered during compaction. If before compaction $x_2 > x_1$, the constraint is

$$x_2 - x_1 \geq K_{min}, \text{ otherwise } x_1 - x_2 \geq K_{min} .$$

A plane-sweep algorithm is used to generate the minimum-spacing constraints [224]. The longest path in the constraint graph determines the minimum width (height) achievable by compaction. Consequently, a longest-path algorithm can be used to determine, in polynomial time, the optimum location of all components or, alternatively, the existence of over-constraint loops.

A valid alternative to the CG algorithm is constituted by LP methods. The only requirement for the use of LP is the linearity of the constraints being enforced. All constraints (6.2), (6.3) and (6.4) can hence be enforced by implementing the constraints explicitly in the linear program.

To ensure that all performance specifications be met, analog IC design often requires that a whole new class of constraints be met in addition to DRC. These additional constraints are known as *topological* and *parasitic constraints*. See Figure 6.5. Symmetry and parasitic matching (Fig. 6.5a), device matching (Fig. 6.5a), and shielding preservation (Fig. 6.5c,d) are generally referred to as topological constraints, while resistive and capacitive coupling are known as parasitic constraints (Fig. 6.5b). In the remainder of this section we discuss the techniques used to enforce topological and parasitic constraints.

6.3.2 Constraints on Stray Capacitances

The bounds on the maximum value of cross-coupling capacitance are enforced by requiring proper spacing between wire segments. The resulting constraint is of type (6.2). As an example, consider the interconnects shown in Figure 6.6a. Compaction without capacitive coupling constraints yields the minimum-area pattern shown in Figure 6.6b. However, the enforcement of a bound on the critical coupling between nets 2 and 3 might require a non-minimum distance between the wires, as shown in Figure 6.6c. In general, the value of the minimum spacing to be enforced between wire segments depends on the wire structure and shape. This value is determined by the algorithm illustrated in Figure 6.7.

In step 3 performance degradations are computed using the linearized approximation of performance degradation ΔK_i based on equation (3.3).

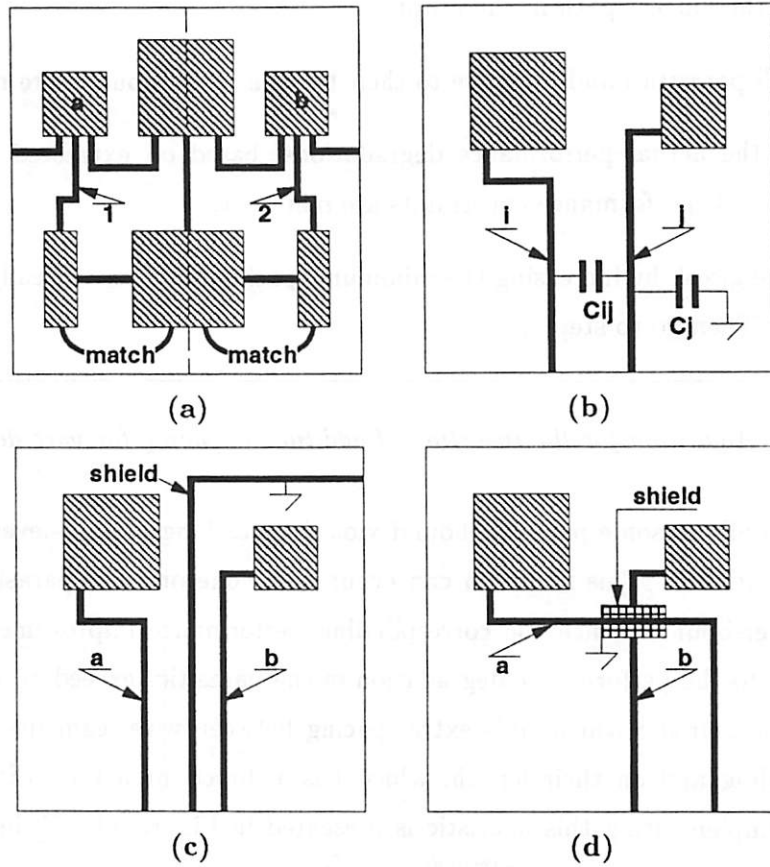


Figure 6.5: *Topological and parasitic Constraints: (a) symmetry and matching; (b) parasitic; (c) lateral shielding; (d) vertical shielding*

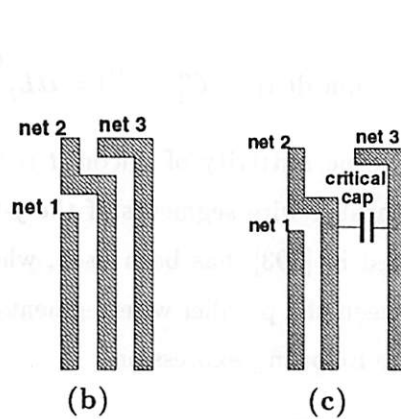


Figure 6.6: *Required spacing for controlling capacitive coupling*

1. Compute the longest path in the graph.
2. Extract all parasitics and compare to their bounds. If all bounds are met, exit.
3. Compute the actual performance degradations, based on extracted parasitics and sensitivities. If performance constraints are met, exit.
4. Modify the graph by increasing the minimum spacing between critically coupled wire segments. Then go to step 1.

Figure 6.7: *Algorithm for the insertion of additional spacing for wire decoupling*

Note that step 3 allows some parasitic bound violations as long as high-level performance constraints are satisfied. This situation can occur when one or more parasitics lay much below their upper-bounds, since the corresponding performance improvement can cancel the contribution to the performance degradation of the parasitics exceeding their bounds. Step 4 employs a heuristic which adds extra spacing between wire segments, based on the need for decoupling and on their length, which has a direct impact on the overall area. The procedure implementing this heuristic is presented in Figure 6.8. C_j indicates the j -th cross-coupling capacitance, and $C_j^{(bound)}$ is the bound on its maximum value. Function $\text{min_dist}(C_j, C_j^{(bound)})$ returns the minimum distance increment to add between parallel segments of the j -th pair of wires, to reduce their cross-coupling capacitance from C_j to $C_j^{(bound)}$. This function depends on the model used, for instance with the parallel-plate model it becomes

$$\text{min_dist}(C_j, C_j^{(bound)}) = \epsilon t L_j \frac{C_j - C_j^{(bound)}}{C_j \cdot C_j^{(bound)}},$$

where ϵ is the dielectric permittivity of silicon, t is the wire thickness, and L_j is the sum of the lengths of all parallel wire segments of the j -th net pair. In our approach, a more precise model, described in [193], has been used, where the dependence of capacitance C_j on the distance d between the parallel wire segments, with widths w_1 and w_2 respectively, is approximated by the following expression

$$C_j = k_0 + k_1 w_1 + k_2 w_2 + \frac{k_3}{d} + \frac{k_4}{d^2},$$

where k_0, \dots, k_4 are technology-dependent constants. Therefore $\text{min_dist}(C_j, C_j^{(bound)})$ be-

```

// Purpose: Add constraints to improve capacitive decoupling.
// Since this procedure is called only if some performance violation
// has been found, we know that at least one bound has been exceeded
foreach cross-coupling  $C_j$  such that  $C_j > C_j^{(bound)}$ 
     $\delta_j = \text{current\_min};$  // current min. distance between any two
                          // parallel segments contributing to  $C_j$ 
     $\delta_j = \delta_j + \text{min\_dist}(C_j, C_j^{(bound)});$ 
    foreach pair  $P_i$  of parallel segments
         $d_i = \text{current\_segm\_min};$  // current min. distance between the
                                  // segments of pair  $P_i$ ;
        if  $d_i < \delta_j$ 
            add_constraints; // add constraint to graph requiring  $d_i \geq \delta_j$ 
                          // between the segments

```

Figure 6.8: Pseudo-code of procedure modify-graph

comes a function of the wire widths too. For each cross-coupling C_j exceeding its bound $C_j^{(bound)}$, δ_j is the minimum distance to be kept between parallel segments of the j -th pair of wires. The distance increment is a function of the parasitic bound violation. Notice that spacing is added not only between the nearest segments, but also between all the segment pairs whose distances are less than δ_j .

The spacing step, implemented by procedure shown in Figure 6.8, can introduce over-constraints making the graph unsolvable. An example in which this might occur is illustrated in Figure 6.9. Two wire segments are connected to terminals A and B , whose relative position is fixed with respect to the instance of a sub-cell. When extra spacing between the segments is required, an overconstraint is generated if $\delta_j > D - W_1 - W_2$. When a positive loop is detected, a pruning procedure is invoked, which removes the newly-added spacing constraints contained in the positive weight loops. In such situations the task of decoupling the two nets is left to the remaining segment pairs. If a feasible solution involving the remaining segment pairs does not exist, then an error is reported. In that event the constraint cannot be met.

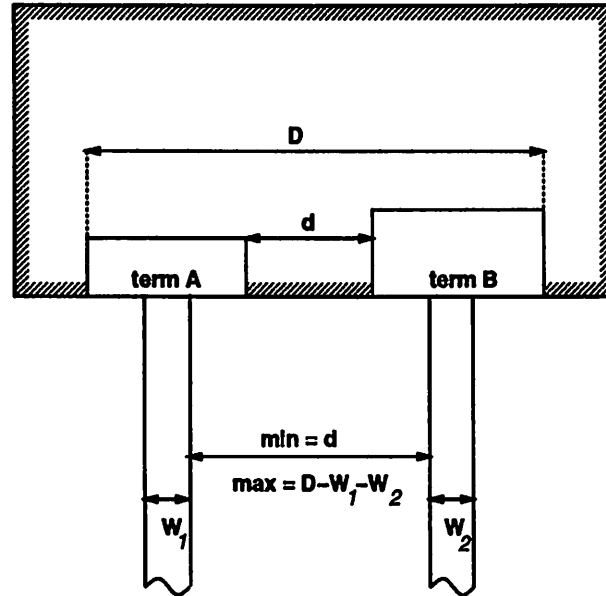


Figure 6.9: *Minimum and maximum wire spacing constraints deriving from connections to fixed-distance terminals*

6.3.3 Preservation of Electrostatic Shields

Two types of shields can be used, *vertical* and *lateral* shields. Figure 6.5d shows a vertical shield created during routing to reduce coupling between critical nets. The horizontal wire, realized with layer 3, runs over a vertical wire segment realized with layer 1, and the shield is realized with layer 2. For example, in a 2-metal/1-poly technology, the lower conductor layer 1 could be poly, the upper layer 3 could be metal2, and the shield layer 2 could be metal1). Vertical shield constraints are expressed as

$$x_2 = x_3 \quad \text{AND} \quad y_2 = y_1, \quad (6.5)$$

where x_2 , y_2 are the coordinates of the center of the shielding plate; x_3 is the x-coordinate of the horizontal wire and y_1 is the y-coordinate of vertical wire.

A lateral shield is a wire stub separating the interconnections as shown in Figure 6.5c. Since the stub has no interconnect purpose and it is often open on one side, a standard digital compactor would remove it to optimize area. Note that shields are generally inserted by the router only when absolutely necessary. If the shields were removed during compaction, the algorithm for the insertion of decoupling spacing would yield large area increments or even report some over-constraints and fail.

The spacing requirement for the lateral shield between nets a and b is

$$y_{shield} \geq y_a \quad \text{OR} \quad y_{shield} \geq y_b. \quad (6.6)$$

where y_{shield} is the location of the top of the shield; y_a that of the top of net a and y_b of net b .

Notice that because of the *OR* condition this constraint is not linear. However, the shield functionality is preserved if either one of the following linear constraints is used in its place

$$y_{shield} \geq y_a \quad \text{AND} \quad y_{shield} \geq y_b \quad (6.7)$$

$$y_{shield} = \frac{1}{2}(y_a + y_b) \quad (6.8)$$

Note that both the LP solver and the constraint graph can be used to implement constraint (6.7), but not (6.8), which can be solved only using a linear or integer program.

6.3.4 Symmetry Constraints

Consider two devices a and b between which a symmetry constraint is required with respect to a symmetry axis s , which, without loss of generality, we assume is vertical (Figure 6.5a). The constraints to be enforced are

$$x_a - x_s = x_s - x_b ; \quad y_a = y_b , \quad (6.9)$$

where x_a, y_a and x_b, y_b are the coordinates of a and b respectively and x_s is the axis position. The vertical constraints in (6.9) can be solved by the constraint graph, while the horizontal constraints have no graph representation and cannot be enforced. However, it can be introduced as a linear constraint in a linear program as

$$x_a + x_b - 2x_s = 0 . \quad (6.10)$$

Symmetry constraints on complete wire segments can be formulated as simple object symmetries on the wire endings and jogs. In Figure 6.5a, the horizontal constraints for the wires labeled 1 and 2 are represented as

$$x_{2,left} - x_s = x_s - x_{1,right}$$

$$x_{2,right} - x_s = x_s - x_{1,left},$$

where $x_{1,right}$ is the rightmost ending of segment 1, $x_{1,left}$ the leftmost ending of segment 1, $x_{2,right}$ the rightmost ending of segment 2, and $x_{2,left}$ the leftmost ending of segment 2.

In this fashion the problem of enforcing one wire symmetry constraint is reduced to satisfying two object symmetries. The wire symmetry constraints can thus be directly incorporated into the LP solver and handled in the same way as the device symmetry constraints. Notice that this methodology guarantees the preservation of existing geometric wire symmetry, not the enforcement of symmetry on arbitrarily shaped wires.

6.4 Algorithmic Considerations

Since all minimum design-rule spacing requirements are integers and each symmetric pair is also governed by minimum spacing requirements, the necessary conditions for Theorem 3, Appendix B are verified. Hence, the optimum solution to the LP problem with symmetry constraints is placed on an integer grid.

As a consequence of Theorem 3, IP techniques are not necessary to solve the compaction problem with symmetry constraints and an LP-based approach suffices. The LP solver is implemented with a simplex algorithm [225]. In this algorithm, the optimum solution is found by sequentially visiting the vertices of the polytope bounding the feasible region. As soon as the minimum is reached, it is recognized as the optimum solution by the algorithm, and no further vertices of the polytope need be visited.

The average speed of the algorithm can therefore be increased if the starting solution is, on average, closer to the optimum. Table 6.1 displays the results of applying this concept to sixteen compaction problems; by using the solution of the constraint graph as the initial starting point for the LP solver, on average, 50% in CPU time was saved. As a general rule, designers adopt symmetry enforcement in circuits with differential structure. However, not all parasitics need be matched with the same accuracy. Sensitivity analysis shows that, in general, only a relatively small number of parasitic matchings influence critical performances such as offset and CMRR. In our approach, the relative criticality of parasitics and mismatch is determined automatically with the method described in this dissertation [2].

The ratio $|S_{i,\Delta}/S_{i,p}|$ of equation (3.16) can be used as a measure of the criticality of such constraints. If this ratio is low, then symmetry is not important for performance, and hence it needs not be enforced. As a consequence, minor deviations from symmetry are still allowed in the layout after compaction, thus improving the flexibility and the success rate of the compaction phase.

circuit	direction	CPU time (s)		
		graph + LP	LP alone	% reduction
AB	horizontal	16.30	22.58	27.8%
AB	vertical	12.92	27.55	53.1%
COMPL	horizontal	8.60	12.15	29.2%
COMPL	vertical	5.50	9.78	43.8%
FASTCOMP	horizontal	13.95	23.85	41.5%
FASTCOMP	vertical	10.17	22.68	55.2%
NEWOTA	horizontal	10.80	14.78	26.9%
NEWOTA	vertical	10.00	17.03	41.3%
OPAMP1	horizontal	13.60	26.23	48.2%
OPAMP1	vertical	4.68	28.28	83.5%
OTA	horizontal	35.30	55.60	36.5%
OTA	vertical	19.33	41.38	53.3%
OTA731	horizontal	19.30	29.35	34.2%
OTA731	vertical	5.02	25.05	80.0%
FCPHIL	horizontal	20.25	29.03	30.2%
FCPHIL	vertical	5.73	37.52	84.7%
Averages	-	13.21	26.42	50.0%

Table 6.1: Comparison of CPU time for graph + LP vs. LP alone

6.5 Wire Length Minimization

Once a minimum area solution satisfying the performance requirements is obtained, a secondary optimization is performed to minimize the total interconnection length. Wire length minimization is performed by the LP solver, so that the complete set of compaction constraints (including symmetry constraints) is considered throughout the optimization. The objective function is formulated as

$$\text{minimize : } \sum_{\text{all wire segments}} (x_{i,\text{right}} - x_{i,\text{left}}), \quad (6.11)$$

where $x_{i,\text{right}}$ is the rightmost ending of wire segment i , $x_{i,\text{left}}$ the leftmost ending of wire segment i , and $x_{i,\text{right}} \geq x_{i,\text{left}}$.

This formulation of the objective function allows the LP solver to find the global minimum horizontal (vertical) wire length. Our design methodology only requires that parasitic bounds, shields, and symmetries be considered. Using total wire length as the secondary objective however, circuit performance can be improved by further reducing wire resistances and parasitic capacitances.

6.6 Compaction with Analog Constraints: A Case Study

Consider once more the clocked comparator COMPL after placement and routing performed by PUPPY-A and ROAD [201], respectively. Some of the parasitics extracted by ESTPAR (see chapter 7) at the completion of the routing phase are shown hereafter, along

with nominal performance and resulting degradation.

$$\mathbf{p}^{(0)} = \begin{bmatrix} 70.3 \text{ fF} \\ 70.4 \text{ fF} \\ 20.6 \text{ fF} \\ 18.0 \text{ fF} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 6.2 \text{ ns} \\ 756 \mu\text{V} \\ -756 \mu\text{V} \end{bmatrix} \quad \overline{\Delta\mathbf{K}} = \begin{bmatrix} 0.8 \text{ ns} \\ 244 \mu\text{V} \\ -244 \mu\text{V} \end{bmatrix}$$

The solution to the constraint-generation problem is

$$\mathbf{p}^{(\text{bound})} = \begin{bmatrix} 87.7 \text{ fF} \\ 87.8 \text{ fF} \\ 32.8 \text{ fF} \\ 28.4 \text{ fF} \\ 1.0 \Omega \\ 1.0 \Omega \\ 1.0 \Omega \\ 1.0 \Omega \\ 1.0 \Omega \\ 5.2 \Omega \\ 49.3 \Omega \end{bmatrix}$$

Notice that now resistive mismatch has become critical because of the shrunk margin allowed to offset degradation. Of the capacitive parasitics, C_{55} and C_{56} have been recognized as more critical than the others, and their bounds have been further tightened with respect to their previous values used to drive the router. Other bounds on C_{15} and C_{16} have been relaxed as a consequence. The layout produced with this set of bounds is shown in Figure 6.10. Capacitive extraction from this layout yields the following values:

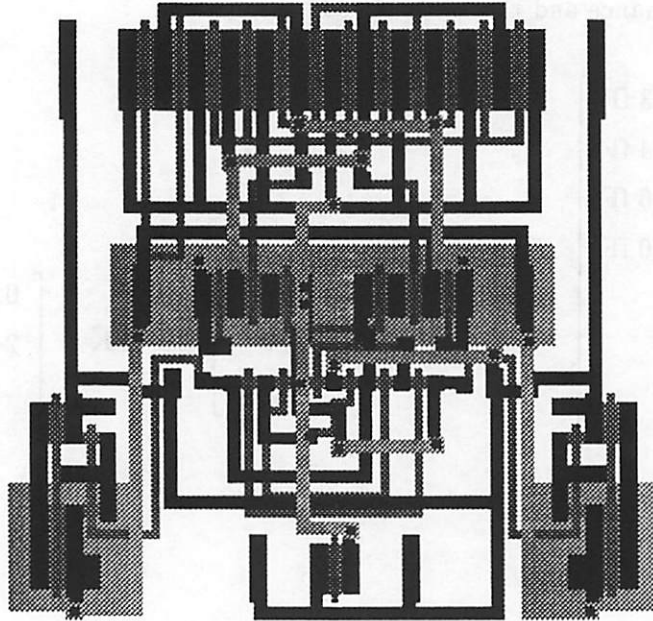


Figure 6.10: *Compacted layout of COMPL with all analog constraints enforced*

$$C_{15} = 73.9 \text{ fF}$$

$$C_{16} = 75.2 \text{ fF}$$

$$C_{55} = 18.8 \text{ fF}$$

$$C_{56} = 17.2 \text{ fF}$$

Simulation showed that in the compacted layout performance specifications were met with an offset of $743\mu V$ and a delay of $6.7ns$. Details on the statistics for the compaction of this circuit are available in section 9.1.1.

Chapter 7

Extraction

*Così di ponte in ponte, altro parlando
che la mia comedia cantar non cura,
venimmo; e tenavamo il colmo, quando*

*restammo per veder l'altra fessura
di Malebolge e li altri pianti vani;
e vidila mirabilmente oscura.*

Dante Alighieri, "Inferno", Canto XXI

In the course of this dissertation we have shown how constraint-based techniques can be used to control or reduce the impact of parasitics on performance. In this chapter we show how constraint-driven strategies can significantly enhance the efficiency of physical parasitic extractors reducing the complexity of parasitic characterizations.

This chapter also deals with issues related to parasitic and mismatch modeling and extraction techniques for a number of applications and technologies. The techniques described in the chapter have been implemented in ESTPAR, which is part of the OCTTOOLS layout tool-set.

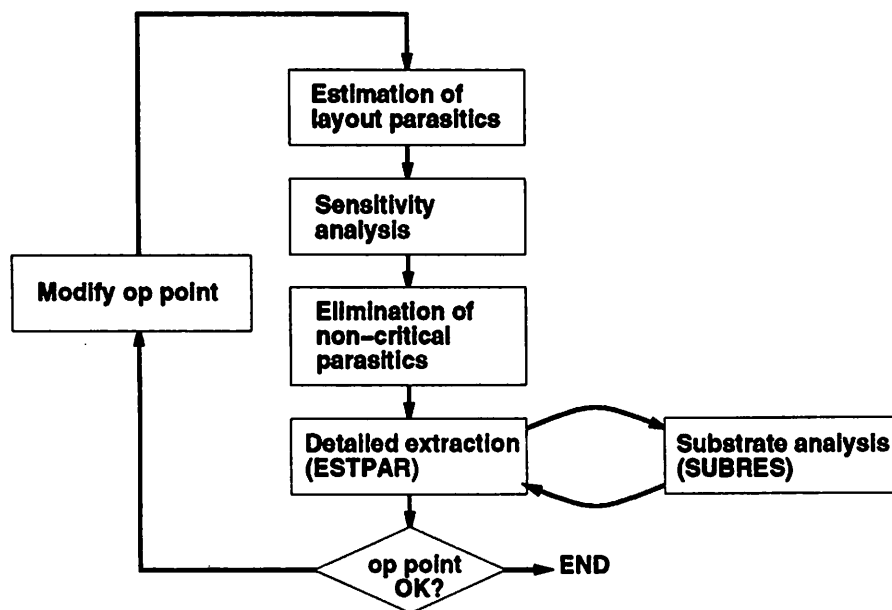


Figure 7.1: Methodology of selective extraction

7.1 General Extraction Methodology

7.1.1 Extraction Tools and Organization

Figure 7.1 shows the flow graph of the extraction methodology. In the first phase, a conservative estimate of all parasitics is computed. For each pair of terminals in the circuit the line resistance is computed as if the worst-case routing scheme were used under the highest possible congestion conditions. Capacitances to substrate are computed similarly, whereas analytical models [193] are used to accurately account for fringing effects. Capacitive cross-coupling is always assumed to take place at least once between any pairs of nets.

In the second phase, performance sensitivities are efficiently computed using the methods discussed in chapter 3. From these data, performance sensitivities with respect to parasitic mismatch are also computed. Based on the previously obtained data, an integer program is constructed and solved using either standard methods or *ad hoc* heuristics developed by us. During this phase a list of critical paths, candidates for extraction, is built.

In the final phase all critical parasitics are extracted in detail. Analytical models

for accurate computation of line-to-substrate and line-to-line capacitance extraction are used, thus ensuring that all important parasitic effects be accounted for. Assuming that the region of operation for the circuit is maintained in the neighborhood of the nominal operating point, this procedure ensures that in a selectively extracted layout, performance degradations are within the limits imposed by the constraints. In case of strongly non-linear circuits a verification phase at the end of the selective extraction must be performed in order to check whether the new operating point is close enough to the previous one. If this condition is not satisfied the process must be iterated using the new operating point for the sensitivity analysis.

All the tools shown in Figure 7.1 have been implemented in the OCTTOOLS environment [120] running under the UNIX operating system. In the remainder of the chapter we will discuss the techniques and models used in our extraction methodology.

7.1.2 Constraint-Based Schematic Simplification

Aim of *schematic simplification* is the reduction of complexity of the extracted schematic. This is often desirable for speeding-up circuit simulation and verification in large systems. In this section a constraint-based strategy is presented for achieving simplification and some considerations on the applicability and resulting accuracy are discussed. The essential idea underlying the method consists of identifying those parasitics which can be neglected, keeping the accuracy of the circuit simulation within pre-defined bounds. Let us consider circuit C , assume that the performance array K is available along with a set of specifications in form of inequality (3.1). Moreover, assume that approximated linearized expressions exist for performance degradation ΔK (3.3) and that accurate estimations of layout parasitics are also available. Using equations (3.3) and (3.1) one can determine if performance K will be within specifications. Suppose that only a certain number of parasitics can be extracted exactly, although conservative estimates can be derived for all of them. Moreover, suppose one is interested in measuring performance with a certain accuracy. To accomplish this objective one would need to determine the set of all parasitics that are essential to ensure the required accuracy. Call P^x this set. In the remainder of this section we will address the problem of minimizing the size of P^x , while ensuring accuracy. Let us introduce some useful definitions.

Definition 1 *The set P of all the parasitics of circuit C is said to be non-critical if there*

exists a partition $\mathcal{P} = (P^x, P^e)$ such that: P^e is a set of all non-critical parasitics, i.e. those parasitics whose cumulative effect on performance is a fraction α of performance specification $\overline{\Delta K}$. $P^x = P - P^e$ is the set of all remaining parasitics.

Definition 2 Let us define performance estimation accuracy A as

$$A = \max_{i=1 \dots N_k} \left| \frac{\Delta K_i^{(max)} - \Delta K_i^{(x-max)}}{\overline{\Delta K}_i} \right|, \quad (7.1)$$

where $\Delta K_i^{(max)}$ is the upper-bound of true performance degradation ΔK_i , $\Delta K_i^{(x-max)}$ the upper-bound of the extracted value of ΔK_i and $\overline{\Delta K}_i$ the specification on K_i .

Lemma 1 If P is non-critical, then for all partitions $\mathcal{P}_m = (P^x, P^e)$, $A = \alpha$

Proof : Since $P = P^x \cup P^e$ and given that upper-bounds for parasitics are known,

$$\Delta K_i^{(max)} = \sum_{j \in P^x} S_{i,j} p_j^{(max)} + \sum_{j \in P^e} S_{i,j} p_j^{(max)} = \Delta K_i^{(x-max)} + \Delta K_i^{(e-max)}.$$

By definition 1, $\Delta K_i^{(e-max)} \leq \alpha \overline{\Delta K}_i$, moreover $\Delta K_i^{(max)} \geq \Delta K_i^{(x-max)}$, hence

$$|\Delta K_i^{(max)} - \Delta K_i^{(x-max)}| \leq \alpha \overline{\Delta K}_i.$$

Since $\overline{\Delta K}_i \geq 0 \forall i$, α must necessarily be the maximum over performance array \mathbf{K} . ■

Next, for each performance measure K_i we need to find a partition \mathcal{P}_i , that minimizes the size of P^x . The problem can be formulated in terms of the following optimization

$\text{maximize : } P^e _{\mathcal{P}_i} \quad (7.2)$
subject to:
$\sum_{j \in P^e} S_{i,j} p_j^{(max)} \leq \alpha \overline{\Delta K}_i$

Let us consider now the general problem of an array of performance measures \mathbf{K} . Assume that each problem (7.2) has a unique solution \mathcal{P}_i . Then, the general partition \mathcal{P}_K , derived as

$$\mathcal{P}_K = \bigcap_{i=1}^{N_k} \mathcal{P}_i, \quad (7.3)$$

is also unique.

Moreover, (7.2) can be rewritten as a standard integer program. Let \mathbf{e} be a vector of size $N_p = |P|$, whose components e_j are boolean variables, e_j being one if parasitic $p_j \in P^e$,

zero otherwise. Then the problem reduces to

$\underset{e}{\text{maximize}} : \sum_{j=1}^{N_p} e_j \tag{7.4}$
<p>subject to:</p> $\sum_{j \in P} e_j S_{i,j} P_j^{(max)} \leq \alpha \overline{\Delta K}_i$

If the components of e are let vary continuously and constraint

$$0 \leq e_j \leq 1, \quad \forall j$$

is added to the integer program, problem (7.4) can be processed by a conventional linear program solver [216]. In our tests we used PARCAR's engine to obtain the \mathcal{P}_K using the techniques presented in chapter 3.

7.2 CMOS Parasitic Modeling

Full three-dimensional simulation of layout structures for accurate computation of interconnect parasitics is impractical for real circuits. Numerical methods based on Finite Difference, Finite Element, Integral Equation or other techniques can be used to extract exact electrical parameters of any structures [226, 227, 228, 229, 230, 231]. These methods however, due to the relatively high computational cost, cannot be efficiently used for a very large number of objects such as those present in a complex layout structure.

A better approach consists of creating models of interconnect which can be easily generated by CAD tools for layout extraction and estimation. Analytical models have been proposed for very specific capacitive structures in [232, 233]. More recently automated methods have been proposed for automatic generation of models for capacitive parasitics in a wide range of structures used in semiconducting circuits [193, 234]. Hereafter a short description of the models used in all our parasitic estimation tools is presented. The techniques used to derive the models are described in detail in [234] and were implemented in a tool called CAPMOD.

7.2.1 Substrate Capacitance

Figure 7.2 shows a simple interconnect line as it appears in most IC technologies. The

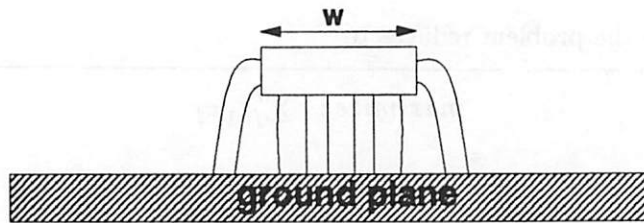


Figure 7.2: Simple interconnect line

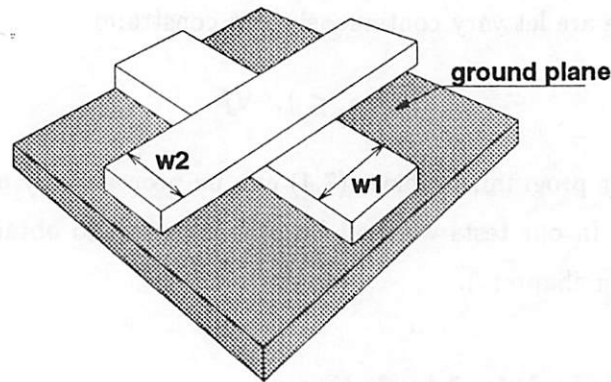


Figure 7.3: Crossover configuration

substrate capacitance of the line is represented in terms of its parallel plate and fringe components. For a given technology for each wiring layer the parallel plate capacitance per unit length only depends on the interconnect width. The fringe capacitance per unit length on the contrary is fixed. The following equation models the substrate capacitance

$$C_0 = k_0 + k_1 w, \quad (7.5)$$

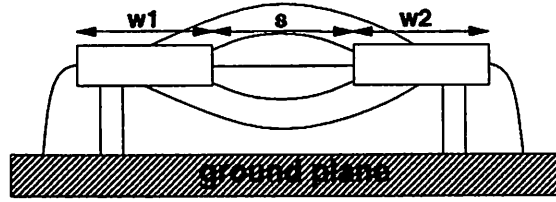
where k_0 and k_1 are real constants dependent on technology and on the type of wiring used. The coefficients are calculated only *once* for a given technology using CAPMOD.

7.2.2 Crossover Configuration

The crossover configuration, shown in Figure 7.3, is modeled by the cross-coupling

$$C_{12} = k_0 + k_1 w_1 + k_2 w_2 + k_3 w_1 w_2, \quad (7.6)$$

where k_0, k_1, k_2 and k_3 are real constants and w_1, w_2 the widths of the two lines. Due to the shielding effect of the lower line over the upper one and vice-versa, the substrate capacitance

Figure 7.4: *Parallel interconnect lines*

of both lines, calculated as in equation (7.5) needs be corrected by the following factors

$$C_{c,1} = k_0 + k_1 w_2 ; C_{c,2} = k_0 + k_1 w_1 + k_2 w_1 w_2 . \quad (7.7)$$

7.2.3 Parallel Lines on the Same Layer

Same layer parallel running lines are dominated by fringing effects, since no vertical overlapping is present and generally the ratio height/width is small¹. Figure 7.4 shows such a configuration. The model for the cross-coupling is as following

$$C_{12} = \mathcal{P}(1/s) + \mathcal{P}(1/(s + w_1)) + \mathcal{P}(1/(s + w_2)) , \quad (7.8)$$

where $\mathcal{P}(\cdot)$ is a given polynomial [234]. The correction components for substrate capacitances are

$$C_{c,1} = \mathcal{P}(1/(s+s_0)) + \mathcal{P}(1/(s+s_0+w_2)) ; C_{c,2} = \mathcal{P}(1/(s+s_0)) + \mathcal{P}(1/(s+s_0+w_1)) , \quad (7.9)$$

where s_0 is a technology- and layer-dependent constant.

7.2.4 Interconnect Lines on Different Layers

There exist two types of configurations for interconnect lines on different layers: the overlapping and the non-overlapping configuration. A non-overlapping configuration is shown in Figure 7.5a. The cross-coupling capacitance is characterized by the following model

$$C_{12} = \mathcal{P}(1/(s + s_0)) + \mathcal{P}(1/(s + s_0 + w_1)) + \mathcal{P}(1/(s + s_0 + w_2)) , \quad (7.10)$$

while correction components are

$$C_{c,1} = \mathcal{P}(1/(s+s_0)) + \mathcal{P}(1/(s+s_0+w_2)) ; C_{c,2} = \mathcal{P}(1/(s+s_0)) + \mathcal{P}(1/(s+s_0+w_1)) . \quad (7.11)$$

¹Sub-micron technologies are currently reversing this trend. However similar reasoning can be used for the generation of appropriate models which take into account high horizontal parallel plate fields.

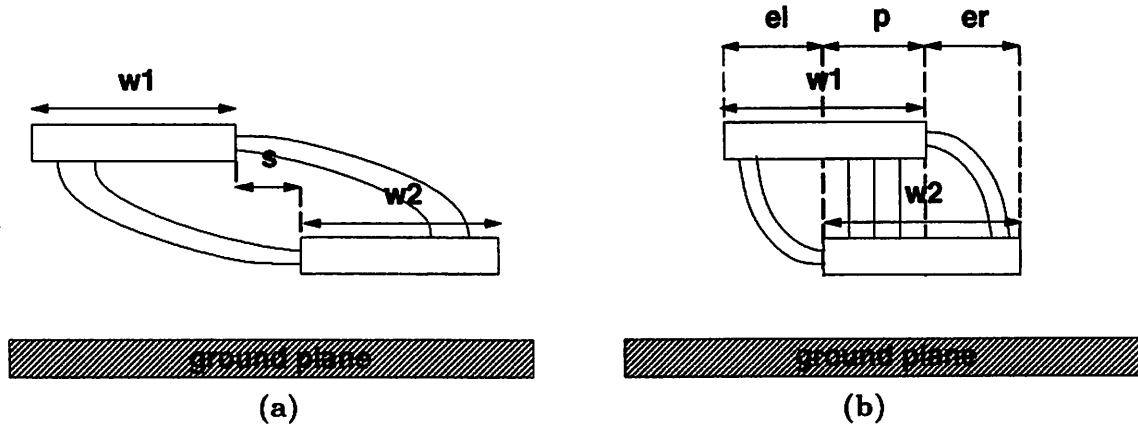


Figure 7.5: Interconnect lines on different layers: (a) non-overlapping; (b) overlapping

Overlapping interconnect lines on the contrary (Figure 7.5) can be modeled as

$$C_{12} = k_0 p + \mathcal{P}(1/(e_l + e_{l0})) + \mathcal{P}(1/(e_r + e_{r0})) , \quad (7.12)$$

where p , e_l and e_r are the cross-, left and right overlaps, respectively. e_{l0} and e_{r0} are constants determined using CAPMOD. The correction terms are determined as

$$\begin{aligned} C_{c,1} &= \mathcal{P}(1/(p + p_0)) + \mathcal{P}(1/(e_l + e_{l0})) + \mathcal{P}(1/(e_r + e_{r0})) , \\ C_{c,2} &= k_0 p + \mathcal{P}(1/(e_l + e_{l0})) + \mathcal{P}(1/(e_r + e_{r0})) . \end{aligned} \quad (7.13)$$

7.3 Technology Gradient Effects: Mismatch Modeling

A number of authors have attempted to find compact and accurate models for parasitic mismatch. In this dissertation we focus on capacitive and resistive mismatches for passive components and on MOS transistor parameter mismatches for active devices. Let us consider a pair of parasitic components p_i and p_j , mismatch M is defined as follows

$$M = 2 \frac{p_i - p_j}{p_i + p_j} .$$

Consider next a pair of active devices d_i , d_j , each characterized by a set of parameters Π_i and Π_j respectively. The mismatch of a pair of parameters is defined similarly as

$$M_{\ell m} = 2 \frac{\pi_\ell - \pi_m}{\pi_\ell + \pi_m} , \quad \forall \pi_\ell \in \Pi_i \text{ and } \pi_m \in \Pi_j .$$

In general, M and $M_{\ell m}$ are non-deterministic measures dependent of process variations, relative distance and geometry of the object pair. For this reason, mismatch is often characterized based on its mean μ_M and variance σ_M^2 . Several models exist in the literature

describing mismatch [141, 235, 236, 237, 238]. In the tools described in this dissertation a technology-independent implementation of the models proposed by Pelgrom [141] and Lakshmi Kumar [236] is being used.

In what follows a short description of the mismatch models used in this dissertation for both passive and active devices is presented. Consider a MOS device and its current-voltage relationship in triode and saturation region, respectively

$$I = K(V_{GS} - V_T - V_{DS}/2)V_{DS} ; I = \frac{K}{2}(V_{GS} - V_T)^2 , \quad (7.14)$$

where I is the drain current, K the transconductance constant, V_T the threshold voltage, V_{DS} the drain-to-source voltage, and V_{GS} the gate-to-source voltage. Let us define the means of measures I , K and V_T as \bar{I} , \bar{K} and \bar{V}_T . Let us assume that all devices are in saturation, then the variance of current mismatch can be derived [236] as

$$\frac{\sigma_I^2}{\bar{I}^2} = \frac{\sigma_K^2}{\bar{K}^2} + 4 \frac{\sigma_{V_T}^2}{(V_{GS} - \bar{V}_T)^2} - 4r \frac{\sigma_{V_T}}{V_{GS} - \bar{V}_T} \frac{\sigma_K}{\bar{K}} ,$$

where r is the correlation coefficient between mismatches in V_T and K , while σ_K and σ_{V_T} are the standard deviations of K and V_T respectively.

Using a set of well-known analytical models, one can derive expressions for σ_{V_T} and σ_K . Consider first the threshold voltage. V_T is expressed as

$$V_T = \Phi_{MS} + 2\Phi_B + \frac{Q_B}{C_{ox}} + \frac{Q_f}{C_{ox}} + \frac{qD_I}{C_{ox}} ,$$

where Φ_{MS} is the gate-semiconductor function difference, Φ_B the Fermi potential in the bulk, Q_B the depletion charge density, Q_f the fixed oxide charge density, D_I the threshold adjust implant dose, q the electron charge, and C_{ox} the gate oxide capacitance per unit area. It can be shown that Q_B , Q_f , D_I and C_{ox} are all statistically independent variables, hence σ_{V_T} can be written as

$$\sigma_{V_T}^2 = \frac{1}{C_{ox}^2} (\sigma_{Q_B}^2 + \sigma_{Q_f}^2 + q^2 \sigma_{D_I}^2) + \frac{\sigma_{C_{ox}}^2}{C_{ox}^4} (\bar{Q}_B^2 + \bar{Q}_f^2 + q^2 \bar{D}_I^2) \quad (7.15)$$

Substituting the values of the individual parametric variances in equation (7.15), one obtains

$$\sigma_{V_T}^2 = \frac{q}{LWC_{ox}^2} (\bar{Q}_B + \bar{Q}_f + q\bar{D}_I) + A_{geom}^2 (\bar{Q}_B^2 + \bar{Q}_f^2 + q^2 \bar{D}_I^2) ,$$

where terms \bar{L} and \bar{W} are the mean channel length and width of the devices, and A_{geom}^2 is a geometry-dependent parameter. Thus, given the relative position and orientation of

a device pair in terms of vector $\Delta \mathbf{v} = [\Delta x, \Delta y, \Delta r]^T$, a compact model of the threshold mismatch in the pair can be derived [141] as

$$\sigma_{\Delta V_T}^2 = \frac{A_p^2}{LW} + S_p^2(\Delta \mathbf{v}) ,$$

where A_p^2 models the area and $S_p^2()$ the spacing dependence for parameter V_T . Terms A_p^2 and $S_p^2()$ are process-dependent and can be obtained empirically from measurements on wafer test patterns.

The transconductance K is calculated as

$$K = \mu C_{ox} W/L , \quad (7.16)$$

where μ is the channel mobility, C_{ox} the gate capacitance per unit area, W the channel width and L its length. Assuming independence for all the factors on the right-hand side of equation (7.16), σ_K^2/\bar{K}^2 can be written as

$$\frac{\sigma_K^2}{\bar{K}^2} = \frac{\sigma_L^2}{L^2} + \frac{\sigma_W^2}{W^2} + \frac{\sigma_\mu^2}{\mu^2} + \frac{\sigma_{C_{ox}}^2}{C_{ox}^2} .$$

Substituting in equation (7.16) the values of the individual variances for each parameter, σ_K^2/\bar{K}^2 can be re-written as

$$\frac{\sigma_K^2}{\bar{K}^2} = \frac{1}{LW} (A_\mu^2 + A_{ox}^2) + \frac{\sigma_L^2}{L^2} + \frac{\sigma_W^2}{W^2} ,$$

where A_μ^2 is a technology-dependent geometry-insensitive model parameter and A_{ox}^2 is a position-dependent parameter [236]. The transconductance mismatch can be represented compactly as

$$\frac{\sigma_{\Delta K}^2}{\bar{K}^2} = \frac{A_K^2}{LW} + S_K^2(\Delta \mathbf{v}) + \frac{A_L^2}{WL^2} + \frac{A_W^2}{W^2L} , \quad (7.17)$$

where A_K^2 , A_L^2 , A_W^2 and $S_K^2(\Delta \mathbf{v})$ are geometry-dependent factors. Expressions for σ_W^2 and σ_L^2 have been derived using the fact that $\sigma_L^2 \propto 1/W$ and $\sigma_W^2 \propto 1/L$ [141]. However, the latter terms are generally small with respect to the other expressions in equation (7.17), hence they can be neglected, yielding

$$\frac{\sigma_{\Delta K}^2}{\bar{K}^2} \approx \frac{A_K^2}{LW} + S_K^2(\Delta \mathbf{v}) .$$

In the literature, functions $S_p(\Delta \mathbf{v})$ and $S_K(\Delta \mathbf{v})$ have been generally reported as being linear with $|v|$, however non-linear expressions can be easily supported in our tools.

Using the value derived in [236] for the variance of C_{ox} , one can easily obtain formulae for the mismatch of parasitic capacitances and resistances. Consider first capacitive mismatch. Let $C = C_{ox}WL$ be the value of a parasitic capacitance of an interconnect line W wide and L long. Then, the variance of C is given by

$$\frac{\sigma_C^2}{C^2} \propto \frac{1}{LW}.$$

Thus, the variance of the mismatch ΔC is

$$\frac{\sigma_{\Delta C}^2}{C^2} = \frac{A_{ox}^2}{LW} S_C^2(\Delta \mathbf{v}),$$

where A_{ox}^2 is an area factor related to the oxide thickness and $S_C^2(\Delta \mathbf{v})$ is the spacing dependence for capacitance pairs.

Let us characterize now stray resistance mismatches. Let $R = R_{\square}L/W$ be the value of the interconnect stray resistance. Then, assuming spatial independence of R_{\square} , the variance of R is given by

$$\frac{\sigma_R^2}{R^2} = \frac{\sigma_L^2}{L^2} + \frac{\sigma_W^2}{W^2}.$$

Hence, by the same reasoning as before, the variance of the mismatch ΔR becomes

$$\frac{\sigma_{\Delta R}^2}{R^2} = \frac{A_L^2}{WL^2} + \frac{A_W^2}{W^2L}.$$

These models were implemented in both the placement and the extraction tools for mismatch-aware synthesis and analysis.

7.4 RF Parasitic Modeling

Accurate interconnect modeling is a fundamental requirement of a constraint-driven approach such as that described in this dissertation. For reasons of efficiency closed formulae and analytical models for interconnect lines and all relevant parasitics are desirable. Since the area of application of this dissertation is that of MMICs, all interconnect lines are modeled as microstrip transmission lines.

Considerable attention has been devoted in the past to accurate and efficient modeling of parasitics relevant at RF. In particular, the study of stripline structures has been extensive. See for example [227, 239] for surveyed work in the field. In this dissertation, and for the sole purpose of its use in the router described in chapter 5, we describe a number

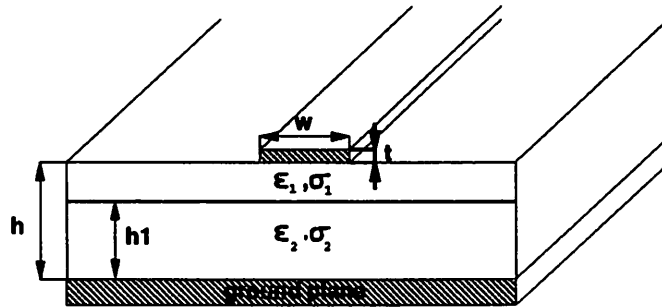


Figure 7.6: *Single microstrip line over lossy substrate*

of analytical models generated using a mixed approach: in part numerical [234] and in part fully algebraic.

Parasitic effects such as inductive and capacitive crosstalk are modeled as coupling capacitances and mutual inductances. Almost all models for interconnect can be represented by the two-dimensional setting of Figure 7.6. The transmission line is fully described by its characteristic impedance Z_o and loss α .

Capacitive and inductive couplings caused by other lines can be easily included in the model as discrete passive components calculated by numerical 2-D field solvers. When coupling is induced by three-dimensional structures, complete 3-D solutions can be used. With such methods it is possible to obtain correction terms ΔZ_o and $\Delta \alpha$ as a function of the geometries involved.

Discontinuities are modeled using discrete components, while radiation and surface-wave propagation effects have been neglected due to the the relative small circuit size if compared with the signal wavelengths. Substrate-dependent losses have been taken into account in the model.

7.4.1 Modeling Single Interconnect Lines

Contrary to simple striplines, microstrip lines are inhomogeneous transmission lines since the field between strip and ground plane is not entirely included in the substrate. Therefore the dominant mode of propagation is not purely transverse electromagnetic (TEM) but quasi-TEM.

As a consequence the effective dielectric permittivity ϵ_r of the transmission medium is lower than the substrate dielectric permittivities ϵ_1, ϵ_2 . Closed forms for characteristic

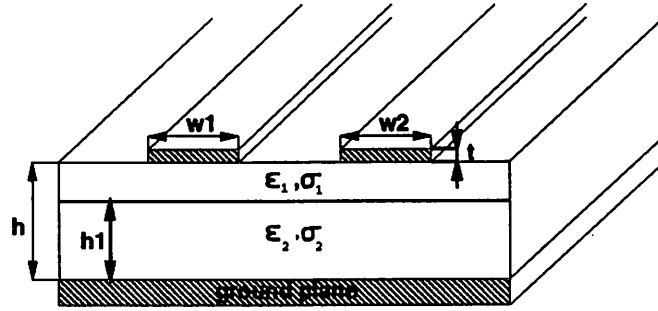


Figure 7.7: Coupled microstrip lines

impedance Z_o and effective dielectric permittivity ϵ_r have been derived using a number of *static* techniques, as reviewed in [240, 241]. Statically derived formulae are quite accurate at frequency below a few GHz. High accuracy can be obtained at higher frequencies using adequate frequency-dependent correction factors. Appendix E.1 gives a summary of the formulae used in this dissertation for the initial sizing of all interconnect dimensions.

Power losses in microstrip lines are related to the the following causes: conductor losses, dissipations in the substrate, radiation and surface-wave propagation. At the frequency of interest ($f < 40$ GHz) and for the materials used in our implementations, radiation and surface-wave propagation effects have been neglected [212]. However surface-wave propagation could be easily be modeled using the closed forms reported in [213]. Closed form expressions for conductor losses α_c and α_d substrate losses exist for microstrip lines and are reported by a number of authors [240, 241]. A summary of the formulae used in our approach for estimating the loss of microstrip lines is in Appendix E.1.

7.4.2 Modeling Coupled Microstrip Lines

Coupled microstrip lines are used in a variety of applications, namely directional couplers, filters and impedance matching networks [242]. From elemental symmetric coupler analysis, it is known that even and odd modes of propagation can be described by the odd and even mode characteristic impedances Z_{oo} and Z_{eo} . Even and odd characteristic impedances are obtained from the following equalities

$$Z_{oe} = \sqrt{\frac{L_e}{C_e}} = \frac{\sqrt{\epsilon_{eff-e}}}{c C_e}, \quad Z_{oo} = \sqrt{\frac{L_o}{C_o}} = \frac{\sqrt{\epsilon_{eff-o}}}{c C_o},$$

where $C_e(C_o)$ and $L_e(L_o)$ are the even (odd) capacitances and self inductances of the lines, ϵ_{eff-e} (ϵ_{eff-o}) is the even (odd) permittivity of the substrate and c is the wave velocity in

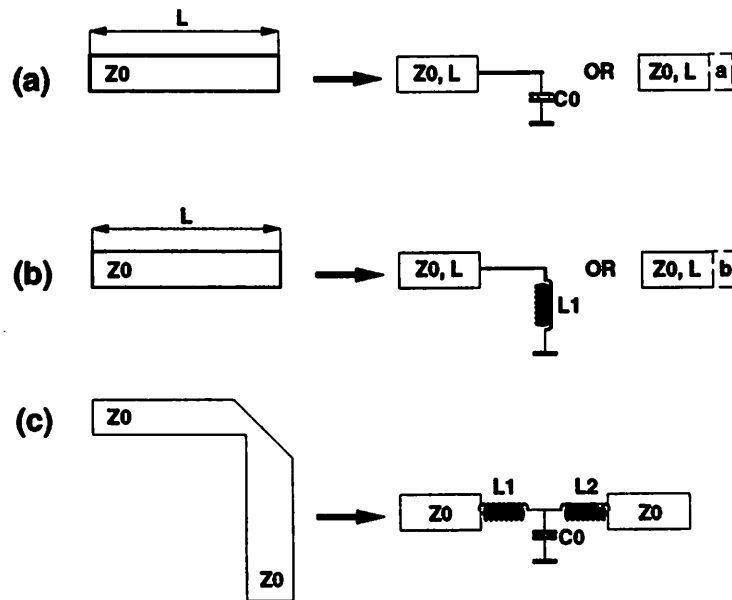


Figure 7.8: *Typical Discontinuities in RF and microwave circuits*

empty space. Semi-empirical expressions for C_e and C_o are given in Appendix E.1. Alternatively, analytical expressions for cross-coupling and substrate capacitances can also be obtained. In our implementation these expressions have been preferred for their simplicity and relatively high accuracy (See Appendix E.1).

7.4.3 Modeling Microstrip Discontinuities

The following discontinuities are generally present in microwave circuits: (a) open circuits, (b) short circuits, (c) right-angled jogs or bends, (d) gaps, (e) steps, (f) transverse slits, (g) T-junctions, and (h) cross-junctions. For the purposes of this work we consider open, short circuits and bends, while T-junctions can be easily included in the analysis. The remaining discontinuities are not of concern during the routing phase.

Discontinuities can be modeled with an equivalent circuit of discrete components (Figure 7.8). These models can be used to generate constraints on the physical dimensions of the discontinuity. Notice that configurations (a) and (b) can be also represented as transmission lines with length augmented by a factor (a,b) . Appendix E.2 summarizes the parameters as implemented in the cost function of the router.

7.4.4 Modeling 3-D Discontinuities

The presence of 3-D objects in the neighborhood of an interconnect line causes inductive and/or capacitive coupling. The patterns of the field lines are generally very complex, and can be accurately simulated through 3-D numerical analysis. Analytical models have been derived from the simulation data based on the knowledge of the physics underlying the effect as in [193]. A 3-D field solver has been used to perform extensive simulations of 3-D structures frequently encountered in layout synthesis. Analytical models have been created based on these data by fitting a polynomial function onto them. Appendix E.3 summarizes the models used in our approach.

7.5 Superconductor Parasitic Modeling

In this section we discuss how the techniques used earlier in CMOS model generation can also be applied to the modeling of very different technologies, such as those used in superconductor ICs.

Under specific conditions, which are always true in superconducting circuits, compact analytical models can be created from simulations or experimental data. Analytical models are quite useful for parasitic characterization, since an accurate insight into the model dependence on various parameters is provided. Moreover, performance-driven synthesis and extraction tools can be considerably sped up when analytical models are used to operate fast and reliable decisions in optimization processes [243].

7.5.1 Analytical Model Generation for Superconducting Inductances

A methodology similar to that used for interconnect parasitics in ICs [234] has been used to generate analytical models for superconducting parasitic inductors. The method, described in Figure 7.9, consists of four phases: configuration selection, numerical simulation, model generation and parameter fitting. In the first phase a set of layout structures among the most common is selected. In the second phase the self and mutual inductance between the conductors is numerically computed for each structure. In the last phase a configuration dependent model is created and its parameters are fitted to the simulation results.

Due to the unique property of superconductivity, interconnections are often mod-

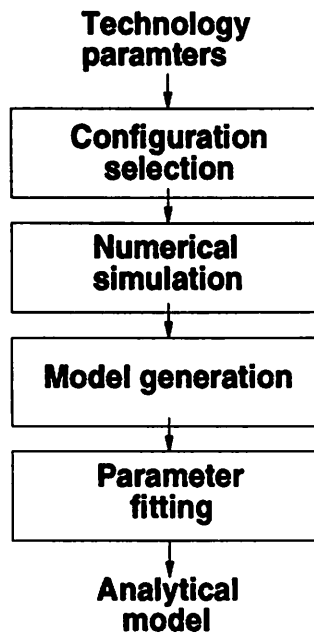
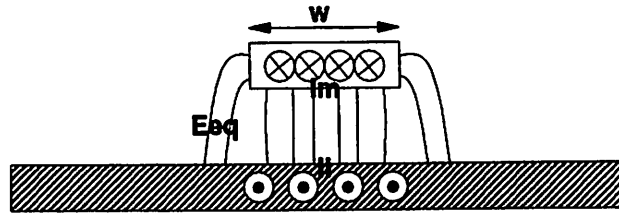


Figure 7.9: *Analytical model generation for superconducting inductances*

eled as purely inductive and capacitive lines. At frequencies of interest however the inductive component dominates, thus reducing the model to a self- and mutual inductance. Currently, the following configurations are supported: (1) a single line over the ground plane, (2) two lines on the same layer and, (3) two (overlapping) coupled lines on different layers. For all configurations per-unit-length inductances are modeled.

The magnetic flux induced in an interconnect line can be partitioned in several components. Although the exact function of the flux may not be known, its monotonical behavior may be used to determine a convenient selection of the variable in which the approximation polynomial should be expressed. For example, in the case of electromagnetic fields, the general dependence on $1/x$ can be exploited to reduce the polynomial complexity of the approximation, thus allowing a much compacter model.

This strategy, along with efficient numerical methods for accurate calculation of inductance in superconducting lines [244, 245] has been chosen to optimize the compactness and the accuracy of the models.

Figure 7.10: *Single line*

7.5.2 Modeling a Single Line

Figure 7.10 shows a single line as it appears in a typical IC technology. When a current I_M flows in the upper plate a current I_I , confined into a region below the upper plate will appear flowing in the opposite direction. This phenomenon, peculiar to superconductivity, is often analyzed by means of an analogy between magnetic flux and electrostatic fields [246]. In this analogy, the magnetic potential A corresponds to the electrical potential V and the flux density $\mathbf{B} = \nabla \times \mathbf{A}$ to the electric field \mathbf{E} . The \mathbf{B} and \mathbf{E} fields are in fact equal in magnitude but orthogonal at each point in the analogous fields, while the z -component of A is equal to V . The currents and charges in the equivalent systems are related by the equivalence $\rho_s = \mu \epsilon J_s$. Where J_s is the surface current density in the superconducting system and ρ_s is its equivalent charge in the electrostatic system.

Using the electrostatic analog one can show that the flux results from the superposition of a “parallel plate” component Φ_p and a fringe component Φ_f . While Φ_p is inversely proportional to w , Φ_f is a constant with respect to w . Consequently the self-inductance per unit length of a single superconducting line is modeled using a polynomial in $\frac{1}{w}$, as

$$L = g_0 + \frac{g_1}{w} + \frac{g_2}{w^2}, \quad (7.18)$$

where g_0, g_1 , and g_2 are real constants, calculated using methods which will be described later.

7.5.3 Modeling Coplanar Lines

In this configuration (see Figure 7.11) a design parameter s , representing the separation between the lines, has been added to the model. Using the electrostatic analogy, the mutual flux between parallel lines can be shown to have a Φ_p and Φ_f components. Φ_p is inversely proportional to the separation between the lines s . Φ_f , on the contrary decreases

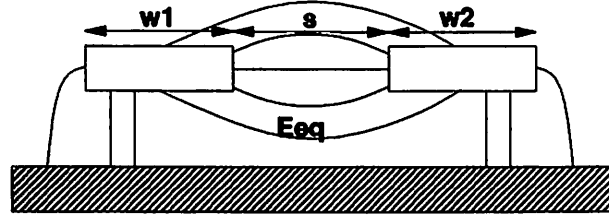


Figure 7.11: Coplanar lines

with the width of each line and with the line separation. Therefore the mutual inductance between coplanar lines is modeled as:

$$M_{cp} = P_n\left(\frac{1}{s}\right) + P_n\left(\frac{1}{s}\left[\frac{1}{w_1} + \frac{1}{w_2}\right]\right), \quad (7.19)$$

where P_n are polynomials of order n . Since part of the fringe field is intercepted by the other line, a correction term L_c has to be added to the self inductance model as calculated in 7.18. The intercepted flux is inversely proportional to the line separation s and the width of the opposite line w_2 . However, unlike the mutual flux, it does not approach infinity as s tends to zero, thus an offset term must be added to s in the model.

$$L_{c.1} = P_n\left(\frac{1}{s + s_{off}}\right) + P_n\left(\frac{1}{w_2}\right), \quad (7.20)$$

$$L_{c.2} = P_n\left(\frac{1}{s + s_{off}}\right) + P_n\left(\frac{1}{w_1}\right). \quad (7.21)$$

7.5.4 Modeling Non-overlapping Lines

In this configuration (see Figure 7.12 b) the “parallel plate” flux component Φ_p between the two layers is not present, thus only the fringe component of the flux Φ_f must be considered. The behavior of Φ_f is similar to the case 7.5.3, however the flux does not have a singularity for zero s . Thus an offset term s_{off} must be included as following:

$$M_{non-ov} = P_n\left(\frac{1}{s + s_{off}}\right) + P_n\left(\frac{1}{s}\left[\frac{1}{w_1} + \frac{1}{w_2}\right]\right). \quad (7.22)$$

For the correction inductive terms, expressions similar to eq. 7.20 and 7.21 are used.

7.5.5 Modeling Overlapping Lines

In this configuration (see Figure 7.12a) a “parallel plate” flux component Φ_p is present. Its magnitude is inversely proportional to the overlap p . The fringe component Φ_f

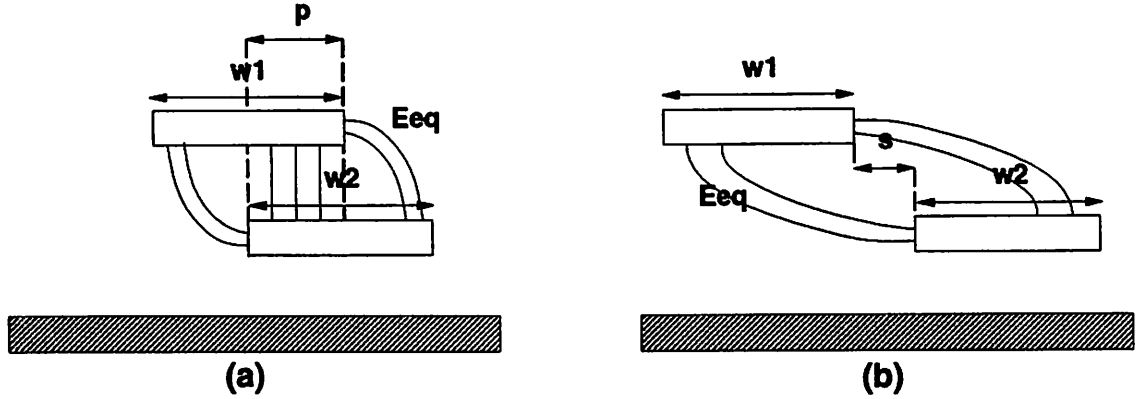


Figure 7.12: (a) Overlapping; (b) Non-overlapping lines

is subdivided in two branches, one for each non-overlapping area. The magnitude of these components is inversely proportional to e_1 and e_2 . Also in this case Φ_f will never vanish, since the lines have a vertical separation due to the isolator. Thus offset terms must be added to the final expression

$$M_{ov} = P_n \left(\frac{1}{p + p_{off}} \right) + P_n \left(\frac{1}{e_1 + e_{1-off}} \right) + P_n \left(\frac{1}{e_2 + e_{2-off}} \right). \quad (7.23)$$

Using similar reasoning, expressions for the correction terms for the self-inductances are obtained.

$$L_{c.1} = P_n \left(\frac{1}{e_1 + e_{1-off}} \right) + P_n \left(\frac{1}{e_2 + e_{2-off}} \right), \quad (7.24)$$

$$L_{c.2} = P_n \left(\frac{1}{p + p_{off}} \right) + P_n \left(\frac{1}{e_1 + e_{1-off}} \right) + P_n \left(\frac{1}{e_2 + e_{2-off}} \right), \quad (7.25)$$

where the parameters are e_1, e_{1-off}, e_2 and e_{2-off} .

7.5.6 Model Characterization

All the parameters present in the models discussed in sections 7.5.2, 7.5.3, 7.5.4, and 7.5.5 have been calculated using model generator INDMOD for a $2 \mu m$ Niobium Josephson Junction process.

The accuracy of the models ranges from 0.4% to 10% with an average of 2%. The models have been obtained for a range of widths and separations of $20 \mu m$, while the thickness was fixed to practical values dictated by the process. A list of models and relative numerical values for each parameter is available in Appendix F.

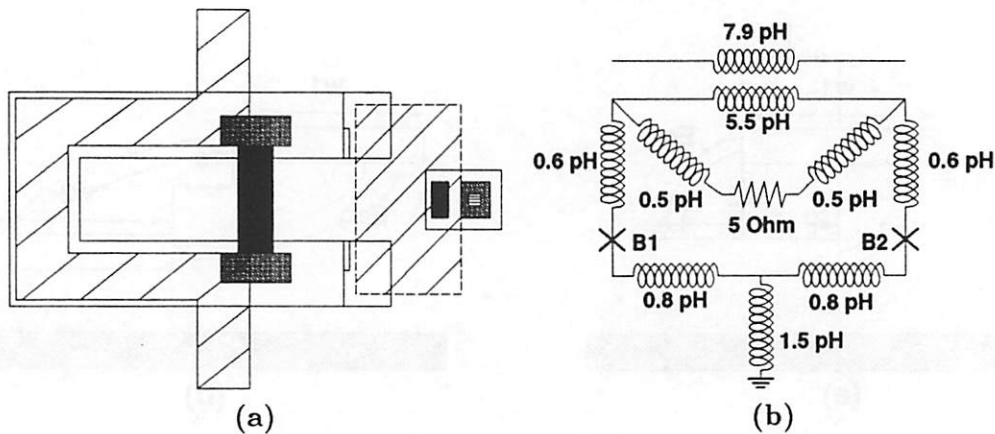


Figure 7.13: (a) Layout of a two-junction SQUID; (b) Extracted schematic using INDEX

Parasitic	Extracted (pH)	Hand (pH)
loop	6.7	7.0
control	7.9	8.0
transformer	3.9	4.2

Table 7.1: Comparison between extracted and hand-computed parasitics

The model generator INDMOD is used to generate the analytical models used by the extraction program INDEX. INDEX is a MAGIC interface used to map all layout structures onto appropriate analytical models. Details on the extraction routines can be found in [247].

7.5.7 Example of Complete Superconductor Extraction

In this section we present a simple circuit that was extracted and characterized using the programs INDEX/INDMOD. The circuit is a two-junction Superconducting Quantum Interference Device (SQUID) circuit (Figure 7.13a), consisting of a loop inductance, control line and transformer. The resulting extracted schematic is shown in Figure 7.13b. The results of the extraction are listed in Table 7.1, which also shows a comparison with the hand calculations. The extraction time was a fraction of a second. INDEX was also tested on a 5-to-32 bit serial decoder. A network simplification phase as described in [247] was also performed. The extraction of 38 junctions, 2507 inductors (345 after network simplification) and 81 resistors was completed in 5 seconds.

Chapter 8

Substrate-Aware Analysis and Optimization

*“Pape Satàn, pape Satàn aleppe!”,
cominciò Pluto con la voce chioccia;
e quel savio gentil, che tutto seppe,
disse per confortarmi: “Non ti nocchia
la tua paura; ché, poder ch’elli abbia,
non ci torrà lo scender questa roccia”.*

*Poi si rivolse a quella ’nfiata labbia,
e disse: “Taci, maladetto lupo!
consuma dentro te con la tua rabbia.*

*Non è senza cagion l’andare al cupo:
vuolsi ne l’alto, là dove Michele
fé la vendetta del superbo strupo”.*

*Quali dal vento le gonfiate vele
caggiono avvolte, poi che l’alber fiacca,
tal cadde a terra la fiera crudele.*

Dante Alighieri, “Inferno”, Canto VII

This chapter presents novel techniques for efficient and accurate computation of substrate related parasitic effects. Models for noise sources, substrate injection, reception and transport mechanisms are illustrated along with applications in optimization within the physical assembly phase. Techniques are proposed for efficient analysis of substrate configurations with arbitrary doping profiles and of the effects of temperature contours on parameters and electrical mismatches in devices. The usefulness of these techniques is demonstrated for large chips with 2,500 contacts or more.

Fast sensitivity computation techniques are presented to evaluate the dependence of performance on technological parameters such as doping levels and profiles. Finally, strategies are proposed for efficient evaluation of the impact of technology and geometric scaling on performance. The techniques for substrate extraction outlined in this chapter have been implemented in a package called SUBRES, which is the basis for Green's Function-based substrate analysis and optimization used by the extraction and placement tools.

8.1 Importance of Substrate in Mixed-Signal Systems

With the continuous increase of chip complexity, device density and circuit speed, power density has become a major source of concern in the design of reliable VLSI ICs. High power density affects circuit performance directly and indirectly. Firstly, by concentrating power in a limited area of the chip, the transient and steady state temperature of large number of devices can be far from the nominal value for which the circuit was designed. This may result in catastrophic as well as parametric faults at circuit and system level.

Furthermore, due to relatively small distances between high-swing high-frequency noise sources and sensitive devices, the substrate becomes a major carrier for spurious signals within the circuit. In mixed-signal circuits in particular, coupling of analog, relatively slow signals with quickly switching digital ones is often disastrous. To alleviate the problem, heavily over-designed structures are generally adopted, thus seriously limiting the advantages of innovative technologies. For this reason substrate modeling has received attention from mixed-signal circuit designers, attempting to integrate RF analog and baseband digital circuitry on the same chip.

In purely digital systems each signal is represented by a sequence of finite number of binary digits; therefore, these signals can take on discrete values only. Due to the binary nature of signals, digital circuits are realized using gates with only two states, each state

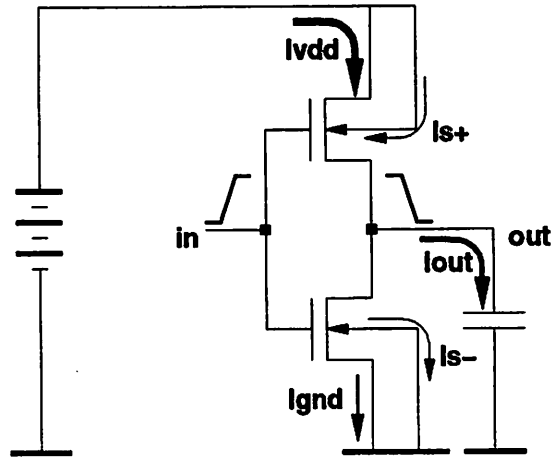


Figure 8.1: *Main and spurious currents in an inverter during transition*

being defined in some range of the continuous signal. This makes digital circuits to a large degree immune to various noise and parasitic sources inherent in ICs. On the contrary, due to the non-discrete nature of circulating signals, analog circuits are in general much more sensitive to electrical noise. A noise signal can be classified based on its nature (*thermal, shot, flicker and switching noise*) and propagation medium (*interconnect, substrate, power supply busses*). The first three noise types relate to intrinsically generated parasitic currents caused by the various physical phenomena, over which little control is possible during the physical assembly. The last noise type is more interesting to us in view of this dissertation, since with opportune design of the medium one can control the impact it may have on performance.

In digital circuits an often large number of gates undergoes a transition periodically. When a transition occurs (Figure 8.1) a spike of current is absorbed from a power bus (I_{vdd}) and used to charge a load in the signal path (I_{out}). In general, a significant portion of this current is discharged to a ground bus through direct feedthrough (I_{gnd}), or it is injected directly into the substrate through various mechanisms¹ (I_s^+ , I_s^-). The cumulative contribution of currents associated with switching gates in the circuit results in ripple noise, generally observable in power and ground busses. Similar spurious noise currents are injected into substrate at various locations, in direct proximity to switching circuitry. The collection of all spurious currents and voltages directly and indirectly generated by

¹See section 8:3.1 for details

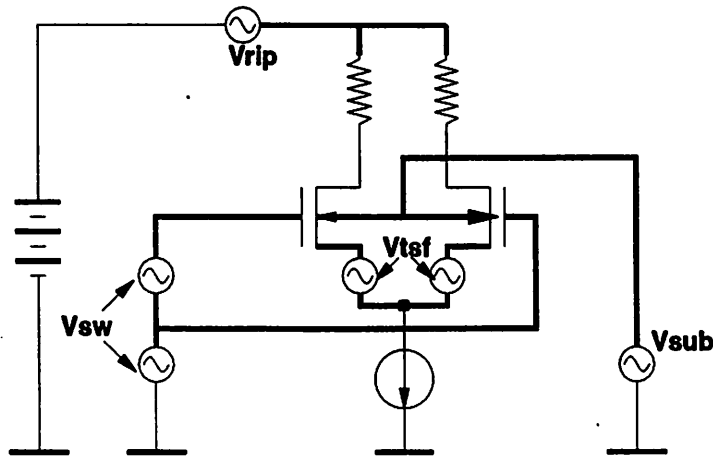


Figure 8.2: Impact of spurious noise signals to a differential pair

switching activity is called *switching noise*. Spatial location and rate of occurrence of transitions are generally time-variant and difficult to characterize exactly. In addition, due to the complexity of typical digital circuits, an exact waveform characterization of switching noise is impractical.

Alternatively, switching noise is often modeled based on its macroscopic appearance. For example, if the number of switching gates is large enough and the global switching activity of the digital circuit is uniformly distributed over a large section of the spectrum, this noise can be modeled as a single Gaussian white or pink noise source. Switching noise in this discrete form can be used to easily estimate its impact on relatively complex analog circuits, hence allowing a designer or an automated tool to determine “safe” floorplannings or global routings.

As an illustration, consider the scenario depicted in Figure 8.2. Sources V_{tsf} represent intrinsic noise generated within the devices. Sources V_{sw} , V_{sub} , V_{rip} represent switching noise transmitted directly through interconnect and substrate or indirectly through power busses. By inspection, it appears clear that a direct effect on the circuit performance is severely degraded by the presence of these noise sources. This observations will be formalized in section 8.5.1, where we shall evaluate performance degradation due to switching noise and calculate numerical bounds on the maximum switching noise which guarantees satisfaction of specifications. In section 8.5.2 techniques will be discussed for the efficient evaluation of substrate related effects during intensive phases such as floorplanning and

placement.

8.2 Modeling Substrate Transport and Thermal Behavior

8.2.1 Background

The effects of thermally inhomogeneous substrate on circuit performance were first studied by Fukahori [248]. According to this approach the substrate is first discretized into a resistive/capacitive mesh. DC/steady-state analysis is carried out by solving directly the system of simultaneous thermal and electrical equations. Transient analysis is performed by using variable time-step trapezoidal integration techniques on the system of simultaneous algebraic equations. More efficient techniques for the solution of the DC and transient analysis were proposed by Lee [249]. Direct LU factorization was replaced with the Incomplete Cholesky Conjugate Gradient iterative method in the DC solution. In the transient simulation the RC mesh was reduced to a macro-model that could be used efficiently inside a AWE-based simulator.

Extensive literature has recently appeared on electrical modeling and analysis of silicon substrate. Compact and accurate macro-models for both lightly and heavily doped substrates have been proposed by several authors, e.g. [36, 196].

Recently, attempts to introduce the effects of substrate on medium-sized integrated circuits have been made using a numerical Finite Difference method, e.g. [250]. The technique is versatile and general in nature, since it can handle lateral and vertical resistivity variations and arbitrary substrate geometries. However, to obtain accurate substrate characterization, a fine mesh is required, thus making the storage and computational effort often prohibitive.

To overcome the formidable computational complexity of the problem, sparse non-uniform grids are often used. The grid size is made fine in regions close to the substrate contacts and coarse in distant regions. The use of non-uniform or coarse grids usually involves a speed-accuracy trade-off. For large substrate problems, it is often necessary to use coarse grids to achieve reasonable computation time and storage requirements. For layout optimization, the grid has to be made very coarse to allow for successive iterations in the substrate contact positions. Since the location of the grid points is usually predefined by means of heuristics, the grid density might be too coarse in the areas in which the geometric structure of the layout is being modified, thus resulting in a considerable

computational error.

Boundary element methods can also be used for parasitic extraction. In [251, 252] a Green's Function is described in a finite uniform medium, with Neumann boundary conditions², using the technique of the separation of variables. Image-charge concepts have been used, in order to avoid the series of computations involved in the method. Despite their efficiency however, all these techniques are still too computational intensive to be used during any optimization loop. Even substrate analysis of complex structures or of thousands of contacts may be impractical or extremely time consuming.

Traditionally, substrate noise analysis has been addressed *a posteriori*, i.e. as an analysis and verification tool after completion of physical assembly. Recently, a number of authors have addressed the problem of performing these tasks efficiently within physical assembly phases [70, 194, 174]. Common to these approaches is the use of a Finite Difference method for the evaluation of the electric field on a coarse grid spanning on the workspace, combined with AWE for an efficient solution of the resulting system of simultaneous algebraic equations. However, these methods often cannot guarantee the accuracy needed for reliable performance estimation, due to the highly coarse grids used. In addition, even if dense or non uniform grids were used, at no extra cost in computation, the alignment requirements of grid and layout objects would be so stringent to make it impossible to use the methods in iterative algorithms based on progressive and often minimal modifications. The latter problem can actually be eliminated if *Voronoi* Tessellation is used during the grid generation [195]. More recently, a method for the reduction of substrate networks has been proposed, which uses congruence transforms on the original system of simultaneous algebraic equations representing the tassellated model of substrate [253]. The method has been proved to be superior than AWE in stability, efficiency and robustness.

An alternative approach to the problem of efficient substrate analysis proposes the use of analytical approximations in order to derive a simpler model for substrate parasitics [196, 197]. A major drawback of this approach is the lack accuracy and the strong dependence on technology and the physical implementation of the circuit, which might not be available at high-level design stages.

In general, silicon substrates in ICs are composed by one or more lightly doped epitaxial layers and a highly doped "core". Hence, differently conductive areas are present

²Zero normal electric field

in the vertical section of the chip, while lateral resistivity variations are due to device and well implants as well as other integrated components. The latter are junctions with the substrate and may therefore be considered equipotential. Calculating resistances between any contact locations on the substrate requires the computation of electric potential $\Phi(x, y, z, t)$ at any location (x, y, z) in the bulk. From Maxwell's equations one can show that

$$\frac{1}{\rho} \nabla \cdot \nabla \Phi(x, y, z, t) + \epsilon \frac{\partial}{\partial t} (\nabla \cdot \nabla \Phi(x, y, z, t)) = 0, \quad (8.1)$$

holds, where ϵ and ρ are respectively the local dielectric permittivity and resistivity of the substrate. In the electrostatic case equation (8.1) reduces to the Laplace equation

$$\nabla^2 \Phi = 0, \quad (8.2)$$

with either Dirichlet or Neumann boundary conditions³ or a combination of those on the surfaces. Equation (8.2) is often solved numerically using techniques based on some discretization of the workspace. A popular method, known as *box integration* technique, consists of partitioning the workspace into three-dimensional boxes indexed $\{i, j, k\}$ for dimensions $\{x, y, z\}$. Equation (8.2) is then translated into a difference equation by replacing the derivatives of Φ in each dimension as

$$\frac{\partial^2 \Phi}{\partial^2 x} = \frac{\Phi_{i+1,j,k} - \Phi_{i-1,j,k} - 2\Phi_{i,j,k}}{\Delta_i^2},$$

where similar substitutions are made for y and z -directions, $\Delta_i, \Delta_j, \Delta_k$ being the sizes of the integration box. Finding the potential in each node $\{i, j, k\}$ is equivalent to computing the voltage at the center of an appropriately sized resistive mesh, assuming that the faces in each box are equipotential. Figure 8.3 shows the discretization of space and the equivalent resistive mesh, including the resistive sizing, used for the solution of the Laplace equation in three dimensions. To find the resistance between each contact pairs, it is sufficient to set all the nodes associated with a given contact to 1 Volt and to measure the current flowing out of each other contact. The resulting system of simultaneous equations is diagonally dominant and sparse, since only seven elements in each row are non-zero. Hence, standard techniques for the solution of sparse linear systems can be applied [20]. The most commonly used methods in substrate-related literature are (1) *direct methods*, based on Gaussian elimination and LU decomposition; (2) *iterative methods*, based on Conjugate Gradient Algorithms,

³Dirichlet conditions impose a given potential, Neumann conditions a given electrical field

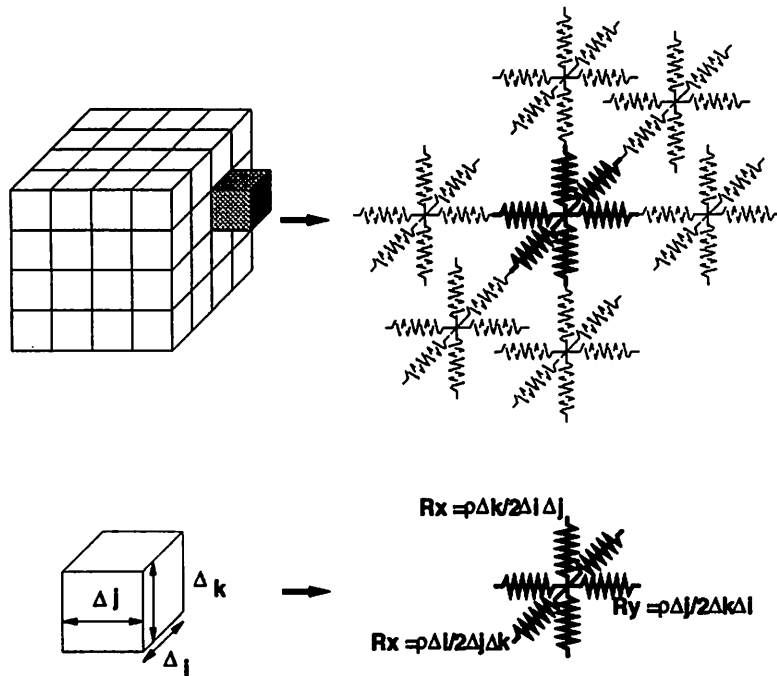


Figure 8.3: *Substrate modeling using RC mesh* (Courtesy of Ranjit Gharpurey)

Jacobi and Gauss-Seidel Relaxation schemes; (3) *hybrid methods*, i.e. combination of (1) and (2) [254, 250]; (4) *frequency domain methods*, based on AWE and similar algorithms. The literature on the subject is extensive, for a review see [255, Chp. 3].

Iterative schemes are the only practical techniques for solving the large matrices which result from the finite differences technique. In realistic problems however, accuracy requirements often impose the use of large number of nodes, typically in the millions. This can lead to prohibitive computation costs, not to mention the storage requirements.

Frequency domain analysis is quite complex for multi-layered substrates, as the different layers of silicon have different time constants. Recently, AWE has been used to address this problem efficiently [250]. In general, AWE produces an approximate representation of the frequency response by selecting appropriate low-order transfer functions to model the circuit. First, the moments of the circuit are extracted and then matched to a low-order transfer function. The second step is performed using the *Pade approximation* [256]. While this technique is elegant in its approach, its usefulness in substrate modeling problems, as discussed in [250], is questionable. The principal reason for this is that with typical substrate doping levels, the substrate impedance in silicon is dominated by the conductance

and not the susceptance up to 4-5 GHz. At these frequencies however, the emergence of skin-effect modes in high-conductivity substrates complicates the modeling problem. Thus, for most substrate modeling tasks, the additional computational complexity of AWE over a simple electrostatic model may not be justifiable. In [70] a similar conclusion was reached by the authors. Hence, despite all of the simplifying assumptions, the solution of substrate problems with more than ten to twenty contacts often becomes impractical.

In this dissertation we advocate the use of Green's Function-based methods for the analysis of substrate and optimization-based synthesis of switching noise sensitive analog ICs. Using these methods we were able to extract accurately a few thousands contacts in reasonable CPU time. In addition it was possible to perform floorplanning on a large, noise sensitive mixed-signal design. The techniques described in the following sections have been implemented in a package called SUBRES, presently available in public domain.

8.2.2 Green's Function-Based Methods: Basics

Consider two arbitrary scalar fields Φ and Ψ defined in a volume V bounded by a surface S . The divergences of vector fields $\Phi\nabla\Psi$ and $\Psi\nabla\Phi$ are given by

$$\nabla \bullet (\Phi\nabla\Psi) = \Phi\nabla^2\Psi + \nabla\Psi \bullet \nabla\Phi , \tag{8.3}$$

and

$$\nabla \bullet (\Psi\nabla\Phi) = \Psi\nabla^2\Phi + \nabla\Psi \bullet \nabla\Phi , \tag{8.4}$$

respectively.

Subtracting (8.4) from (8.3) one has

$$\nabla \bullet (\Phi\nabla\Psi) - \nabla \bullet (\Psi\nabla\Phi) = \Phi\nabla^2\Psi - \Psi\nabla^2\Phi . \tag{8.5}$$

Integrating both sides of (8.5) over volume V and applying the divergence theorem one has

$$\int_V (\Phi\nabla^2\Psi - \Psi\nabla^2\Phi)dv = \oint_S (\Phi\nabla\Psi - \Psi\nabla\Phi) \bullet \hat{\mathbf{n}}ds = \oint_S (\Phi\frac{\partial\Psi}{\partial n} - \Psi\frac{\partial\Phi}{\partial n}) ds , \tag{8.6}$$

where $\hat{\mathbf{n}}$ is the unit outward normal vector to the surface S enclosing the volume V . Let $\Phi(\mathbf{r})$ be the potential at point \mathbf{r} resulting from a localized charge density $\rho(\mathbf{r}')$, and $\Psi(\mathbf{r}, \mathbf{r}')$ the potential at \mathbf{r} due to a point charge placed at a point \mathbf{r}' . The potential functions are related to the charge densities by the Poisson equation. Thus

$$\nabla^2\Phi = -\frac{\rho}{\epsilon} ,$$

and

$$\nabla^2 \Psi(\mathbf{r}, \mathbf{r}') = -\frac{\delta|\mathbf{r} - \mathbf{r}'|}{\epsilon} ,$$

Resubstituting the Poisson equation into (8.6) and using \mathbf{r}' as the integration variable, one has

$$\Phi(\mathbf{r}) = \int_V \rho(\mathbf{r}') \Psi(\mathbf{r}, \mathbf{r}') d^3 \mathbf{r}' + \epsilon \oint_S \left(\Psi \frac{\partial \Phi}{\partial n} - \Phi \frac{\partial \Psi}{\partial n} \right) ds' , \quad (8.7)$$

where $\Psi(\mathbf{r}, \mathbf{r}')$ is known as the *Green's Function*. If the Green's Function is known, equation (8.7) allows one to determine the potential at any point in the volume V due to a known arbitrarily distributed charge density. Image-based techniques and the method of separation-of-variables (SOV) are two different approaches for evaluating the Green's Function. The methods are described in detail in [255, Chp. 3].

In the absence of any boundaries, that is in the free-space case, the function $\Psi(\mathbf{r}, \mathbf{r}')$ reduces to $1/4\pi\epsilon|\mathbf{r} - \mathbf{r}'|$, where \mathbf{r} is the observation point and \mathbf{r}' is the source point. In this case, if the field term in the surface integral falls faster than $|\mathbf{r}|^{-1}$ at large distances from the source, then as the surface S is taken to infinity, the term related to the surface integral vanishes and the potential function $\Phi(\mathbf{r})$ is related to the charge density through

$$\Phi(\mathbf{r}) = \int_V \frac{\rho(\mathbf{r}')}{4\pi\epsilon|\mathbf{r} - \mathbf{r}'|} d^3 \mathbf{r}' . \quad (8.8)$$

The function $\Psi(\mathbf{r}, \mathbf{r}')$, in the presence of finite boundaries can be imagined as consisting of two parts, the first is the free-space Green's Function and the second is the potential due to charges outside the volume V , distributed so that the required boundary conditions are satisfied on S [257].

8.2.3 Using the Green's Function in Substrate Analysis

Consider the structure in Figure 8.4. Suppose one would like to compute the complex impedances between contacts 1 and 2 and the impedance toward ground. Before proceeding with the solution of this problem it is necessary to make an important assumption. As mentioned earlier, at frequencies up to 4-5 GHz, substrate susceptance ϵ is typically much smaller than conductance σ , hence it may be ignored and all substrate impedances may be considered real. In the electrostatic case, the problem of computing the resistance between a substrate contact and all the others can be translated into that of computing the charge at the contact when set at a potential of 1 Volt, while the other contacts and the backplane are grounded. The reason for this is the following. Capacitance C_{ij} between

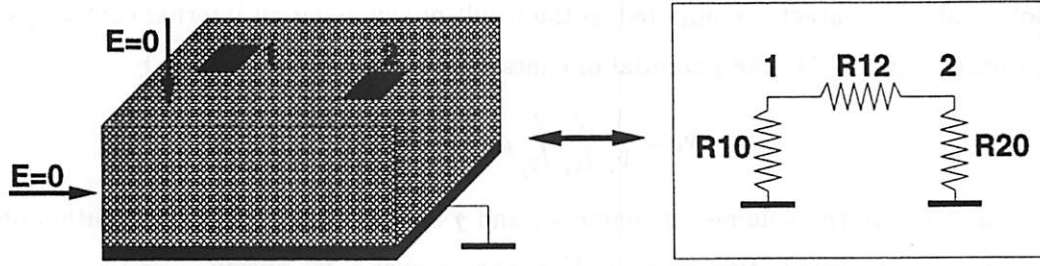


Figure 8.4: Substrate boundaries and contact resistance modeling

contacts i and j is defined as the ratio of the charge on contact j to the potential of contact i , or $C_{ij} = Q_j/\Phi_i$. By Stokes Theorem,

$$C_{ij} = \frac{1}{\epsilon} \oint_S E \cdot \hat{\mathbf{n}} ds, \quad (8.9)$$

where $\hat{\mathbf{n}}$ is the unit outward normal vector to the surface S which encompasses the contact. E is the electric field in the medium. Similarly, the resistance between contacts is defined as

$$R_{ij} = Y_{ij}^{-1} = \left[-\sigma \oint_S E \cdot \hat{\mathbf{n}} ds \right]^{-1} = -\frac{1}{\sigma \epsilon} \frac{\Phi_i}{Q_j}, \quad (8.10)$$

where σ is the medium conductivity. Note that in both the resistive and the capacitive cases the potential satisfies the Laplace equation, thus the problems can be interchanged freely.

Hence, by solving the capacitance problem (after appropriate scaling of medium susceptances), it is possible to easily obtain all substrate conductances of the resistance problem. The next step is therefore the computation of the potential everywhere due to charges placed in the substrate.

Now, let us turn our attention to solving equation (8.7). The problem depicted in Figure 8.4 represents a mixed-boundary case, since zero potential in the chip's back-plane is assumed (Dirichlet condition) and vanishing normal electric field on the other faces (Neumann condition). Under these conditions, equation (8.7) simplifies⁴ to

$$\Phi(\mathbf{r}) = \int_V \rho(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') d^3 \mathbf{r}', \quad (8.11)$$

where V is the chip's volume region and G the Green's Function.

⁴In purely Neumann boundary problems this simplification is not allowed.

The potential of a contact is computed as the result of averaging all internal contact partitions. Hence, using (8.11) the potential of contact i can be derived as

$$\bar{\Phi}_i = \frac{1}{V_i} \int_{V_i} \int_{V_j} \rho_j G dv_j dv_i ,$$

where V_i and V_j are the volumes of contacts i and j and ρ_j is the charge distribution on j . If a uniform charge distribution $\rho_j = Q_j/V_j$ is chosen over j , we obtain

$$\bar{\Phi}_i = \frac{Q_j}{V_j V_i} \int_{V_i} \int_{V_j} G dv_j dv_i . \quad (8.12)$$

The solution to equation (8.11) for each contact pairs yields the *coefficient of potential* matrix \mathbf{P} . The relation between matrix \mathbf{P} and vectors $\bar{\Phi}$, the average potential at each contact, and \mathbf{Q} , the charge associated with all contacts, is described as

$$\bar{\Phi} = \mathbf{P} \mathbf{Q} . \quad (8.13)$$

Similarly, one can derive the relation

$$\mathbf{Q} = \mathbf{c} \bar{\Phi} , \quad \mathbf{c} = \mathbf{P}^{-1} , \quad (8.14)$$

where \mathbf{c} is called *coefficient of induction* matrix. For a contact i , the capacitance to ground C_i and all mutual capacitances C_{ij} are characterized as [255]

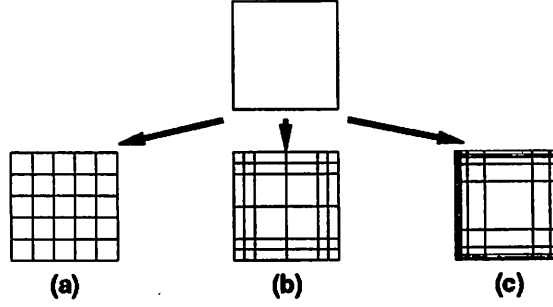
$$C_i = \sum_{j=1}^N c_{ij} , \quad C_{ij} = -c_{ij} , \quad (8.15)$$

where N is the size of matrix \mathbf{c} . Using equations (8.9) and (8.10) in combination with relations (8.15), all mutual and ground resistances can be easily derived.

The accuracy of the substrate conductance computation can be increased given the same number of partitions if a non-uniform partitioning scheme is used. Figure 8.5a shows a uniform scheme, while Figure 8.5b shows a quadratic and Figure 8.5c a logarithmic scheme. The Green's Function is identical and the computational complexity increases only marginally as a result of a modest overhead due to the partition generation scheme, which might be relatively challenging in complex contact profiles [255].

8.2.4 Computing the Green's Function in Multi-Layered Substrates

In the previous section we have seen how the Green's Function can be computed and used to derive the impedance matrix for all contacts on the surface of a single, uniform

Figure 8.5: *Partition schemes for substrate contacts*

and isotropic substrate. In this section we present the Green's Function used to represent complex multi-layer doping profiles.

Consider an array of layers of different thickness and permittivity. At each boundary between layers i and $i + 1$ the following conditions must be satisfied on the current density J_n

$$J_n = \sigma_i E_i = \sigma_{i+1} E_{i+1} ,$$

which translates in the capacitive case in the following conditions on the normal component of the electrical displacement D_n

$$D_n = \epsilon_i E_i = \epsilon_{i+1} E_{i+1}$$

The full derivation of the Green's Function can be found in [255]. Here, we shall outline the basic steps to justify the sensitivity analysis and some optimization techniques proposed in this dissertation. Figure 8.6 shows the multi-layered structure for which a Green's Function must be computed. Consider the case in which the point-charge at $\mathbf{r} = (x, y, 0)$ and the observation point at $\mathbf{r}' = (x', y', 0)$ are in the same dielectric permittivity ϵ_N . The Green's Function corresponds to an infinite series of sinusoidal functions

$$G(\mathbf{r}, \mathbf{r}') = G_0|_{z=z'=0} + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} f_{mn} C_{mn} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{m\pi x'}{a}\right) \cos\left(\frac{n\pi y}{b}\right) \cos\left(\frac{n\pi y'}{b}\right) , \quad (8.16)$$

where $C_{mn} = 0$ for $m = n = 0$, $C_{mn} = 2$ for $m = 0$ or $n = 0$, but $m \neq n$, and $C_{mn} = 4$ for all $m, n \geq 0$. Parameters a , b and d are the dimensions of substrate in x-, y- and z-direction (Figure 8.6). The term G_0 is computed as

$$G_0 = \frac{1}{ab\epsilon_N} \frac{\Gamma_N}{\beta_N} . \quad (8.17)$$

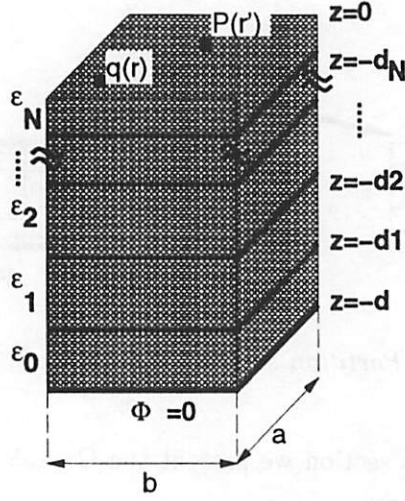


Figure 8.6: Multi-layer doping profiles (Courtesy of Ranjit Gharpurey)

The terms Γ_N and β_N are computed recursively from equations (8.18).

$$\begin{bmatrix} \beta_k \\ \Gamma_k \end{bmatrix} = \begin{bmatrix} \frac{\epsilon_{k-1}}{\epsilon_k} & 0 \\ (\frac{\epsilon_{k-1}}{\epsilon_k} - 1)d_k & 1 \end{bmatrix} \begin{bmatrix} \beta_{k-1} \\ \Gamma_{k-1} \end{bmatrix}, \quad (8.18)$$

where the recursion begins with the values $\beta_0 = 1$ and $\Gamma_0 = d$. The term f_{mn} is computed as follows

$$f_{mn} = \frac{1}{ab\gamma_{mn}\epsilon_N} \frac{\beta_N \tanh(\gamma_{mn}d) + \Gamma_N}{\beta_N + \Gamma_N \tanh(\gamma_{mn}d)}, \quad \text{with } \gamma_{mn} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}. \quad (8.19)$$

The terms β_N and Γ_N for $m \neq 0$ or $n \neq 0$ are computed recursively from equations (8.20).

$$\begin{bmatrix} \beta_k \\ \Gamma_k \end{bmatrix} = \begin{bmatrix} \frac{\epsilon_{k-1}}{\epsilon_k} \cosh^2(\theta_k) - \sinh^2(\theta_k) & (\frac{\epsilon_{k-1}}{\epsilon_k} - 1) \cosh(\theta_k) \sinh(\theta_k) \\ (1 - \frac{\epsilon_{k-1}}{\epsilon_k}) \cosh(\theta_k) \sinh(\theta_k) & \cosh^2(\theta_k) - \frac{\epsilon_{k-1}}{\epsilon_k} \sinh^2(\theta_k) \end{bmatrix} \begin{bmatrix} \beta_{k-1} \\ \Gamma_{k-1} \end{bmatrix}, \quad (8.20)$$

where $1 \leq k \leq N$, $\theta_k = \gamma_{mn}(d - d_k)$, $\beta_0 = 1$, and $\Gamma_0 = 0$.

From equation (8.12), adapted for surface contacts⁵ one can derive an expression for the average potential at contact i due to the charge on contact j

$$\bar{\Phi}_i = \frac{Q_j}{S_j S_i} \int_{S_i} \int_{S_j} G(s_j, s_i) ds_j ds_i, \quad (8.21)$$

where S_j and S_i are the surfaces of the contacts.

⁵In [255] the extension to 3-D contacts is discussed in details

The components p_{ij} of matrix \mathbf{P} are computed as the ratio of $\bar{\Phi}_i$ and Q_j , from (8.21)

$$p_{ij} = \frac{\bar{\Phi}_i}{Q_j} = \frac{1}{S_i S_j} \int_{S_i} \int_{S_j} G(s_j, s_i) ds_j ds_i, \quad (8.22)$$

Replacing (8.16) into (8.22) and integrating, one obtains an explicit formula for p_{ij} .

$$p_{ij} = \frac{\Gamma_N}{ab\epsilon_N\beta_N} + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} k_{mn} \frac{[\sin(m\pi \frac{a_2}{a}) - \sin(m\pi \frac{a_1}{a})] [\sin(m\pi \frac{a_4}{a}) - \sin(m\pi \frac{a_3}{a})]}{(a_2 - a_1)(a_4 - a_3)} \\ \times \frac{[\sin(n\pi \frac{b_2}{b}) - \sin(n\pi \frac{b_1}{b})] [\sin(n\pi \frac{b_4}{b}) - \sin(n\pi \frac{b_3}{b})]}{(b_2 - b_1)(b_4 - b_3)}, \quad (8.23)$$

where the term k_{mn} replaces the following expression

$$k_{mn} = \frac{a^2 b^2 f_{mn} C_{mn}}{m^2 n^2 \pi^4}. \quad (8.24)$$

Parameters (a_1, a_2) and (b_1, b_2) are the x- and y-coordinates of node i and (a_3, a_4) and (b_3, b_4) those of node j .

The doubly infinite series of (8.23) tends to converge slowly. The problem can be eliminated by rewriting the second term of (8.23) after proper scaling, as a cosine series

$$\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} k_{mn} \cos(m\pi \frac{a_{1,2} \pm a_{3,4}}{a}) \cos(n\pi \frac{b_{1,2} \pm b_{3,4}}{b}). \quad (8.25)$$

Equation (8.25) is a compact representation for a sum of 64 terms forming all possible combinations of signs and indices. By replacing the ratios of contact coordinates and the substrate dimensions with ratios of integers as

$$\frac{a_k}{a} = \frac{\tilde{p}_k}{\tilde{P}}; \quad \frac{b_k}{b} = \frac{\tilde{q}_k}{\tilde{Q}}, \quad (8.26)$$

and summing over finites limits \tilde{P} and \tilde{Q} , equation (8.25) becomes

$$\sum_{m=0}^{\tilde{P}-1} \sum_{n=0}^{\tilde{Q}-1} k_{mn} \cos(m\pi \frac{\tilde{p}_{1,2} \pm \tilde{q}_{3,4}}{\tilde{P}}) \cos(n\pi \frac{\tilde{q}_{1,2} \pm \tilde{q}_{3,4}}{\tilde{Q}}). \quad (8.27)$$

One can immediately see that equation (8.27) is almost identical to the two-dimensional Discrete Cosine Transform (DCT) of k_{mn}

$$K(\tilde{p}, \tilde{q}) = \sum_{m=0}^{\tilde{P}-1} \sum_{n=0}^{\tilde{Q}-1} k_{mn} \cos(m\pi \frac{\tilde{p}}{\tilde{P}}) \cos(n\pi \frac{\tilde{q}}{\tilde{Q}}). \quad (8.28)$$

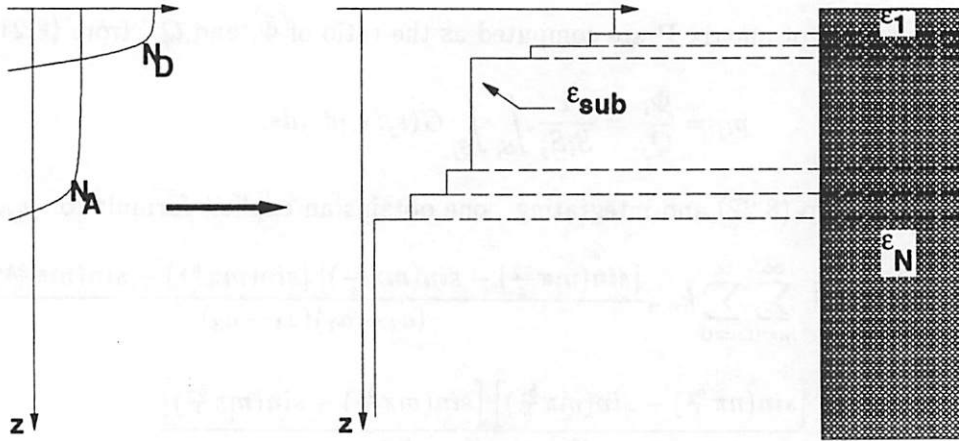


Figure 8.7: Discretization of non-abrupt doping profiles

In fact one can easily show that equations (8.27) and (8.28) can be interchanged [255]. Hence, the computation of p_{ij} ultimately requires only a simple DCT. Several efficient techniques exist for efficient computation of the DCT, e.g. FFT-based techniques only require a computation complexity $O(\tilde{P}\tilde{Q} \log_2 \tilde{P}\tilde{Q})$. Note that the value of k_{mn} is solely dependent on the properties of the substrate in z -direction. Hence, for a given substrate structure the DCT needs be derived **only once**. Any modification in the relative position of one or more nodes is captured completely by the Fourier transform, thus only matrix \mathbf{P} needs be calculated and inverted. However, due to the relatively small size of \mathbf{P} , this process does not require a significant CPU time. Non-abrupt doping profiles can be analyzed at low CPU cost by simply discretizing in z -direction with a gradually changing value of permittivity as shown in Figure 8.7.

8.2.5 Substrate Extraction Algorithm

The complete method for substrate analysis is described in Figure 8.8. The DCT of k_{mn} is computed for each location in a Manhattan grid covering the whole substrate surface. To generate matrix \mathbf{P} , it is necessary to compute the parameter p_{ij} for all the pairs of partition elements composing each contact. If no scheme is used for its reduction (see section 8.2.7), the size of \mathbf{P} is

$$|\mathbf{P}| = \sum_{i=1}^{N_c} \phi_i,$$

where N_c is the total number of contacts and ϕ_i the number of partitions in contact i .

```

fmn = compute_fmn (substrate description); // equation (8.19)
kmn = compute_kmn (fmn); // equation (8.24)
compute_DCT(kmn); // DCT of the whole workspace
store_DCT_values;
foreach contact
  contact_part = get_partitions (contact); // partition contact (Fig. 8.5)
  integer_representation (contact_part); // x, y → p, q coordinates (8.26)
  foreach contact_part(i) // consider all pairs of partitions
    foreach contact_part(j)
      pij = compute_pij (contact_part(i), contact_part(j));
  P = compose_matrix (pij);
  c = P-1 = invert_matrix (P);
  get_contact_resistances(c); // equations (8.10) and (8.15)

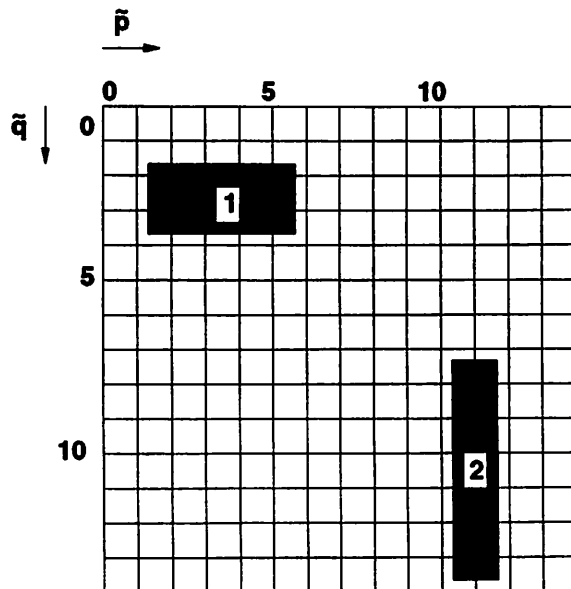
```

Figure 8.8: *Pseudo-code of the substrate resistance extraction algorithm*

The inversion of \mathbf{P} is the last step before the computation of \mathbf{R} or \mathbf{Y} , the matrices of all mutual resistances and conductances between substrate contacts. Due to the dense nature of \mathbf{P} , the inversion is the most time consuming operation of the whole algorithm. Several inversion techniques, both direct and iterative, have been implemented by us and in [255]. Among the direct methods, a LU decomposition-based algorithm of complexity $O(|\mathbf{P}|^3)$ has been used for relatively small configurations of less than approximately 1000 partitions⁶. Larger circuits required the use of various accuracy-driven simplification schemes, which are the subject of section 8.2.7.

As an illustration consider the pair of contacts depicted in Figure 8.9. Suppose that both contacts are partitioned so that each partition covers exactly a grid node. Contact 1 will cover nodes (2,2) to (2,5) in x-direction and (3,2) to (3,5) in y-direction for a total of eight nodes, while 2 will have only six partitions. Hence, in this case \mathbf{P} is a 14x14 matrix. Suppose the grid is ten times denser for a hundred fold higher resolution, moreover assume that the partitions of each contact remain unchanged. Then, the size of \mathbf{P} will not change and neither will the CPU time needed for its inversion. Only an additional overhead of logarithmic time will be necessary for the computation of the DCT, which will be computed in any case only once for the technology of this substrate.

⁶ \mathbf{P} is not sparse, hence no simplification techniques could be used.

Figure 8.9: *Discretization of the substrate surface*

8.2.6 Thermal analysis

The integrated circuit is modeled as a multi-layer, three-dimensional, rectangular structure with multiple heat sources on the surface and a heat sink on its backplane. Substrate temperature $T(x, y, z, t)$ at an arbitrary location (x, y, z) and at time t is found by solving

$$\nabla \cdot \kappa \nabla T(x, y, z, t) + Q(x, y, z, t) = \frac{1}{D} \frac{\partial T(x, y, z, t)}{\partial t}, \quad (8.29)$$

where $Q(x, y, z, t)$ is source of heat dissipation or absorption, D the thermal diffusivity (equivalent to the reciprocal of heat capacity), and κ the isotropic thermal conductivity. Assuming that the time constant for heat conduction is large compared to the speed of the circuit, we are interested in characterizing the steady-state thermal behavior of the substrate. Hence the homogeneous part of equation (8.29) reduces to

$$\nabla \cdot \kappa \nabla T(x, y, z, t) = 0,$$

which is similar to equation (8.1) for the electrical conduction problem. Hence, assuming isotropic thermal behavior, an identical solution scheme can be applied, by replacing the substrate electrical conductivity $1/\rho$ with the thermal conductivity κ . The resulting network is the equivalent of a thermal one where resistive components are the reciprocal of

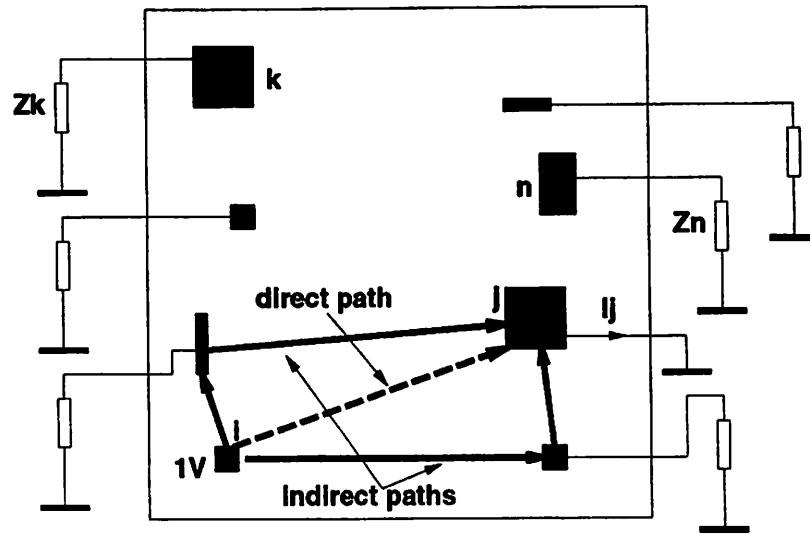


Figure 8.10: *Direct and indirect current-flow paths* (Courtesy of Ranjit Gharpurey)

thermal conductances and current sources represent heat generators. Hence the substrate temperature can be computed efficiently at each surface point by finding the corresponding voltage in the equivalent network.

8.2.7 Schemes for Efficient Solution of Large Substrate Problems

Many authors who in the past have dealt with the substrate problem, have also proposed methods for the reduction of its size, to improve the overall computation efficiency and to reduce the mesh of extracted parasitics. A classical approach consists of creating *active extraction windows* around each contact and extracting only those substrate impedances associated with contacts in close proximity, i.e. within the window. This method has a number of drawbacks. First, there is no guarantee that a given accuracy constraint can be met with a particular choice of size or shape of the window. Second, the loading on certain contacts may have an important effect on the extraction, altering the dominance of direct current paths over indirect ones in the substrate.

In [255] an alternative method was proposed. The method, implemented by us as well, is presented in this dissertation. The aim of the method is to make matrix \mathbf{P} sparse, with no accuracy reduction. Consider the scenario depicted in Figure 8.10. n contacts are laid out on the substrate surface. Each contact k is loaded with impedance Z_k . Suppose now that contact j is grounded while i is at 1 Volt and that the ratio $Y_{ij} = I_j/V_i$ is sought.

If $Z_k = 0$, $\forall k \neq i$, then the direct path dominates in the computation of Y_{ij} , thus it cannot be ignored. On the contrary, when $Z_k \neq 0$, for some $k \neq i$, then a bound $Z_k^{(bound)}$, can be derived for which the direct path can be ignored, without violating pre-determined accuracy constraints.

Let matrix \mathbf{P} be known. By equations (8.15) and (8.9) \mathbf{Y} , i.e. the matrix which relates the voltages of all contacts to the currents flowing out of them, can be easily derived. Let \mathbf{Y}_L be the vector of the load admittances at each contact, for simplicity assume that Y_L is purely resistive, i.e. $\text{Imag}\{Y_L\} = 0$. Equation (8.30) represents the effects of loading on the circuit.

$$[\mathbf{Y} + \text{diag}(\mathbf{Y}_L)] V = I . \quad (8.30)$$

V and I are the vectors of contact potentials and currents respectively. One can show that if condition

$$\left| \sum_{k=1}^N Y_{jk} V_k \right|_{k \neq i, k \neq j} > |Y_{ji}| \quad (8.31)$$

is met, component Y_{ji} can be set to zero. For each component of \mathbf{Y} set to zero a precise value can be computed for the lost accuracy of all currents I_k . Hence the process can be applied until the cumulative inaccuracies reach a pre-determined value.

The procedure is most suited for an iterative solution scheme. Figure 8.11 shows the complete extraction scheme. Each contact, the *subject*, is considered separately. First, a partition, the *subject partition* or P_{Near} is defined around the subject such that each contact belonging to it does not satisfy (8.31). Second, the substrate surface is further partitioned into larger sections or P_{Far} (See Figure 8.12) that are increasing in size geometrically⁷. Third, all contacts contained in each partition P_{Far} are replaced with a single contact, the *super-contact*, placed in the center of the partition of area equal to the sum of the areas of all the original contacts.

$$x_{sc} = \frac{x_{left} + x_{right}}{2} , \quad y_{sc} = \frac{y_{left} + y_{right}}{2} ,$$

$$A_{sc} = \sum_{k \in P_{Far}} A_k , \quad (8.32)$$

where y_{sc} and x_{sc} are the coordinates of the super-contact and A_k the contact areas; $x_{left/right}$ and $y_{left/right}$ are the left/right x- and y-coordinates of the partition boundary,

⁷The formula for the computation of the size is the following: $d_{k+1} = \alpha(d_k)$, where $\alpha = 5$ in our prototype. Since in these partitions all the contacts satisfy equation (8.31), the growth criterion does not affect the accuracy in any way and it was chosen so as to facilitate partition computations.

```

foreach contact                // consider all contacts
  select_near_contacts;         // determine partition  $P_{Near}$ 
  construct_all_partitions;     // assign all contacts to the other partitions ( $P_{Far}$ )
  foreach partition
    construct_super_contacts;   // use (8.32)
    modify_local_problem;      // use (8.33)
    iterative_solve_local_problem;

```

Figure 8.11: *Pseudo-code of the simplified substrate extraction scheme*

respectively. Fourth, a simplified equation is derived for the subject j , where the original equation is replaced as follows

$$Y_j \bullet V = I_j \rightarrow \tilde{Y}_j \bullet \tilde{V} = I_j, \quad (8.33)$$

where Y_j denotes the j th row of matrix \mathbf{Y} , \tilde{Y}_j is a vector of size $s < |Y|$ resulting from replacing the required number of contacts by super-contacts. \tilde{V} is the vector of the potentials on the remaining contacts and super-contacts.

Finally, one step of the iteration for the solution of the problem is performed and the loop continues considering the next contact. The algorithm terminates when every contact has been considered.

To obtain some bounds on the maximum attainable simplification rate, consider the following extreme cases: (1) every contact satisfies (8.31); (2) no contact satisfies (8.31). In case (1) $P_{Near} = \emptyset$, hence only super-contacts exist and the size of matrix \mathbf{Y} is reduced by one or the size of \mathbf{P} is decreased by the number of partitions internal to the subject. In case (2) matrix \mathbf{Y} is not simplified, and neither is \mathbf{P} , hence the complexity of the problem remains that of inverting \mathbf{P} .

However, this is generally not the case in real substrate problems, where complexity reduction in schematics is typically a factor of ten [255].

8.3 Switching Noise Sources

In the course of this dissertation we will often refer to low- and high-resistivity substrate. The distinction is necessary for two reasons. First, these substrate types are used

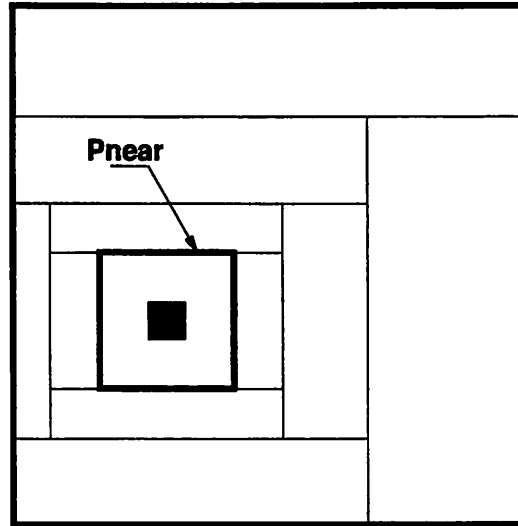


Figure 8.12: *Partitioning of substrate for the simplification algorithm*

mainly in BiCMOS and CMOS applications respectively. Second, switching noise transport mechanisms are substantially different in the two substrate types, thus resulting in different design guidelines to obtain isolation. Typical substrate implementations generally used in IC fabrication are shown in Figure 8.13. In the remainder of this section the mechanisms underlying current injection and reception are discussed.

8.3.1 Substrate Injection Mechanisms

Different types of active and passive devices in use in most IC technologies are shown in Figure 8.14. The cross-section of a bipolar npn transistor is shown in Figure 8.14a. In these devices coupling to substrate is generally capacitive through collector-to-bulk junction capacitor C_{js} . The value of C_{js} for an abrupt junction can be estimated as

$$C_{js} = \sqrt{\frac{q\epsilon}{2(\psi_{bi} + V_{cs})} \left(\frac{N_C N_S}{N_C + N_S} \right)},$$

where N_C , N_S are the collector and the substrate doping levels, ψ_{bi} the built-in junction potential, V_{cs} the collector-to-substrate bias voltage, q the electron charge and ϵ the substrate dielectric permittivity. For formulae of more complex doping profiles, see [258]. Injection can also occur through the parasitic pnp that forms, when the main device approaches the saturation region of operation and its base becomes forward biased with respect to the collector. The base of the npn acts as the emitter of the parasitic pnp, the collector as its

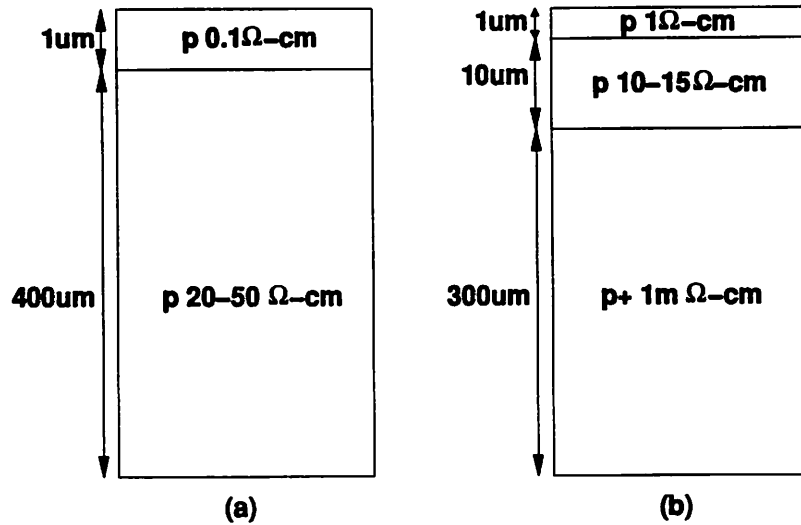


Figure 8.13: Typical IC substrates: (a) high-resistivity; (b) low-resistivity

base and the substrate as its collector. However, the gain of the pnp will be necessarily small due to the high thickness of its base.

Lateral pnp transistors inject noise mainly through the base-to-substrate capacitance (Figure 8.14b). On the contrary, in vertical pnp transistors the substrate is the collector node (Figure 8.14c). Thus, significantly higher currents can be injected in the substrate, unless low impedance draining is provided in immediate proximity of the device.

MOS transistors, shown in Figure 8.14d,e for a n-well process, can interact with substrate in a number of ways: (1) capacitively, through the source(drain)-to-substrate capacitance (modeled as C_{J0} and C_{JSW} in SPICE) and through *avalanche* effects; (2) resistively through *hot-electron* injection. Hot-electron effects are observed when the field in the depleted drain-end of a transistor becomes large enough to cause impact ionization and generate electron-hole pairs. The dependence of hot-electron induced substrate current I_{sub} on the device operating current is given by

$$I_{sub} = K_1 (V_{ds} - V_{dsat}) I_d \exp\left(-\frac{K_2}{V_{ds} - V_{dsat}}\right),$$

where I_d is the drain current, V_{ds} the drain-to-source voltage, V_{dsat} the drain-to-source voltage at saturation and K_1, K_2 are semi-empirical constants [259]. Recent experimental evidence suggests that hot-electron induced substrate currents are the dominant cause of substrate noise in NMOSFETs up to at least 100 MHz[260]. Shorter device channel are likely

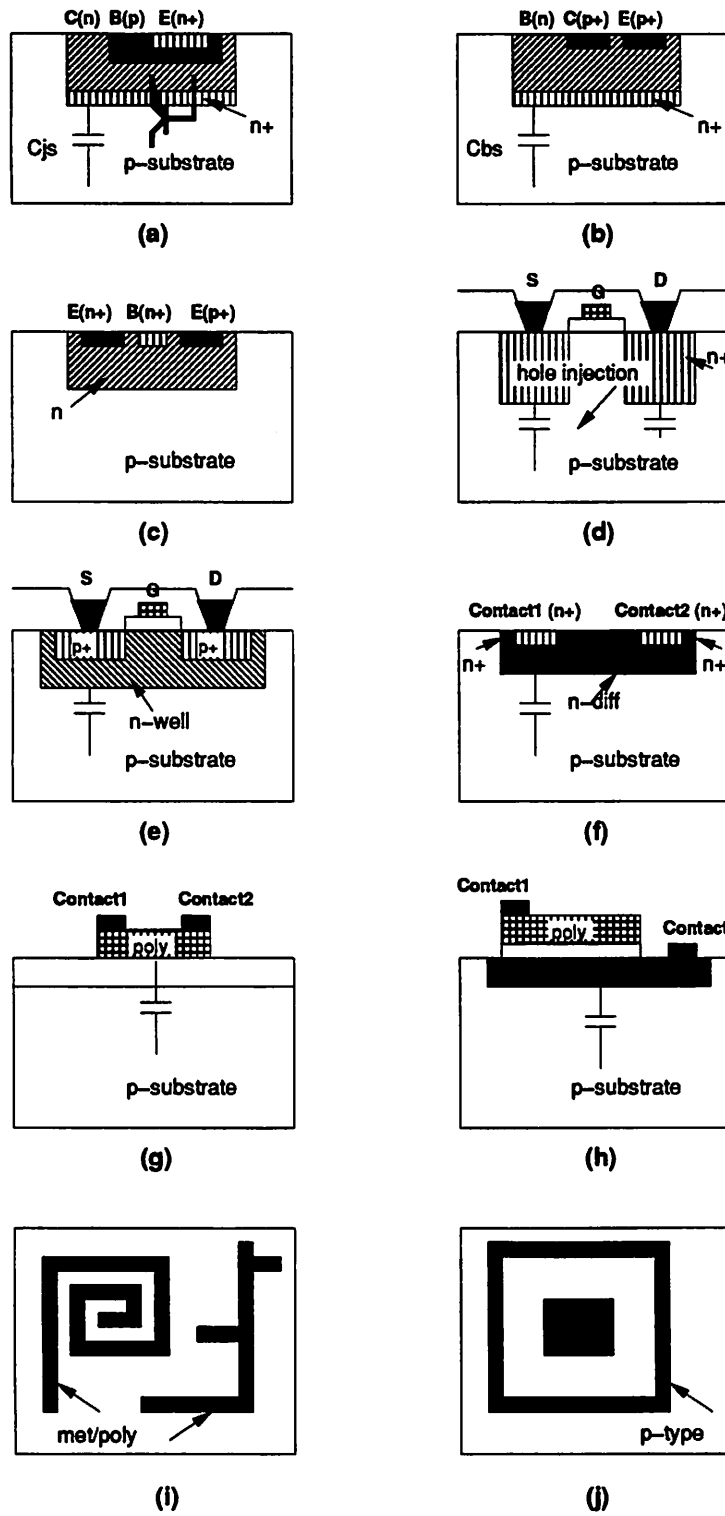


Figure 8.14: Injection and reception mechanisms (Courtesy of Ranjit Gharpurey)

to worsen the problem in the future, due to increased fields and smaller oxide thicknesses. Moreover, capacitive injection has several mitigating factors in its impact on performance: the absence of DC and even harmonic components in the power spectrum. For small-signal analysis, hot-electron induced current can be modeled as a drain-to-body transconductance g_{db} given by

$$g_{db} = \frac{\partial I_{sub}}{\partial V_D} = \frac{K_2 I_{sub}}{(V_{ds} - V_{dsat})^2} .$$

The direct effect of g_{db} , which appears in parallel to r_o , is the reduction of the transistor output impedance.

Hot-electron induced substrate currents in PMOS transistors are considerably smaller than in comparably sized NMOS transistors due to a lower hole ionization-coefficient. Substrate injection is further reduced by the fact that PMOS devices in the process shown here are built in a locally grounded well (signal wise). The quality of the grounding is crucial, in fact if the well potential is allowed to vary with respect to the substrate potential, the entire well acts as a large injector, with a large reverse-biased well-to-substrate capacitance, thus worsening the effect.

Moreover, reverse-biased pn junctions formed by all devices with substrate exhibit a steady DC leakage current. This current consists of carriers which are swept across the depletion barrier in the direction of the electric field. Electrons are injected into the n-region and holes into the p-region under the action of the field. Hence the substrate current induced by this mechanism is a majority-carrier drift current.

The passive components in typical processes are shown in Figure 8.14f,g,h,i. These components include resistors, capacitors, inductors and local diffusions. Resistors are in general either poly-type or diffused. Poly resistors have a capacitance to substrate small compared to that of diffused resistors. Assuming that one end of the resistance is connected to an AC ground, the current I injected into the substrate at low-frequencies, due to a voltage V_{in} applied at the other end is given by

$$I = \sqrt{\frac{j\omega C}{R}} \tanh\left(\frac{\sqrt{j\omega RC\ell}}{2}\right) V_{in},$$

where C is the per unit capacitance, R the resistance and ℓ the length of the resistor. Local diffusions in the substrate can be p or n-type. N-type diffusions inject noise through a reverse bias capacitance. P-type diffusions are often used as substrate taps or guard rings, to tie down the substrate to a desired potential. If designed improperly these diffusions can

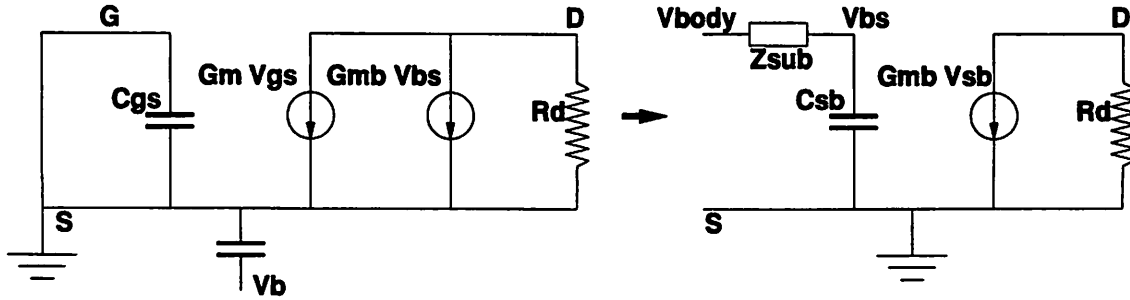


Figure 8.15: *Body Effect in MOSFETs* (Courtesy of Ranjit Gharpurey)

inject very high levels of noise into the substrate, as they act as wide ground-planes on the substrate and any voltage bounce on these diffusions is conveyed throughout their extent on the chip through a very low impedance path. Guidelines for the design of guard rings can be found in great detail in [255].

8.3.2 Substrate Reception Mechanisms

Capacitive sensing is the most common mechanism of noise reception in surface devices, as bipolar transistors, capacitors, resistors and interconnect lines. The junction with substrate in lateral pnp devices consists of the n-type base region. If the pnp device is used in a gain stage, then the base of the device must be carefully shielded, or connected to a low impedance node. Otherwise the substrate noise will be amplified by the gain of the circuit.

In addition to capacitive pickup through the source and drain depletion junctions, MOS devices also exhibit a more severe form of substrate interaction due to the *body effect*. In MOS devices threshold voltage V_t is a strong function of the substrate potential. For a uniform surface impurity concentration N_A , this dependence is given by [261]

$$V_t = V_{t0} + \frac{\sqrt{2q\epsilon N_A}}{C_{ox}} \left(\sqrt{2\Phi_f + V_{sb}} - \sqrt{2\Phi_f} \right),$$

where ϵ is the substrate dielectric permittivity, N_A the substrate doping, C_{ox} the per unit oxide capacitance, $2\Phi_f$ the surface inversion potential and V_{sb} the source-to-body potential. The effect can be represented by a linearized model parameter g_{mb} in the small signal device model [261]. Shorting gate and source of transistor in Figure 8.15, a gain stage exists

between substrate S and drain D . With suitable approximations [261] it can be shown that

$$\frac{g_{mb}}{g_m} = \frac{\sqrt{2q\epsilon N_A}}{2C_{ox}\sqrt{2\Phi_f + V_{sb}}},$$

where g_m is the small-signal transconductance of the device. Parameter g_m relates the drain current to the gate-to-source voltage. In typical processes the g_{mb}/g_m ratio varies from 0.1 to 0.3. The parasitic body-to-drain gain is thus only 14-20 dB lower than the gate-to-drain gain. This fact makes MOSFET devices especially vulnerable to substrate noise reception at low to medium frequencies. On the contrary capacitive pickup, exhibited by most other devices, becomes significant only at relatively high frequencies (above 1 MHz).

8.4 Substrate Conductivity and Technology

Computing the sensitivity of substrate coupling with respect to a number of technology parameters is useful for a number of reasons. First, it allows to evaluate the effects of slight imperfections in the fabrication process on the performance of a circuit and, ultimately, its yield. Second, it is useful for the selection of the best cost-effective technology on the basis of the class of circuits one wants to fabricate with given specifications. Finally, the technique can be used during substrate-aware optimization to help the decision process by providing a trend to the best possible improvement [31].

In section 8.2 we have shown how resistive substrate couplings can be efficiently computed using the Green's Function and the DCT. In this section we develop the theory for the computation of the sensitivity of substrate coupling with respect to the doping profiles and the geometry of the substrate structure.

The relation between circuit performance K and technology, via substrate-related parasitics⁸, is obtained using the following expression

$$\Delta K = \sum_{\ell} \frac{\partial K}{\partial T_{\ell}} \Delta T_{\ell}, \text{ with } \frac{\partial K}{\partial T_{\ell}} = \sum_{i,j} \frac{\partial K}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial T_{\ell}}, \quad (8.34)$$

where (i, j) represent a contact pair, Y_{ij} the substrate conductive coupling between i and j , and T_{ℓ} a technology parameter.

Hence, assuming $\frac{\partial K}{\partial Y_{ij}}$ exists⁹, ΔK can be easily evaluated as a linear function of technology parameters T_{ℓ} , provided that term $\frac{\partial Y_{ij}}{\partial T_{\ell}}$ has been computed. This term is

⁸The effects of technology on other parasitics have been considered in chapter 3.

⁹This term can be computed numerically in an efficient manner, during circuit simulation. See chapter 3.

generally ignored due to the extremely high complexity of its evaluation if traditional Finite Differences methods are employed. If a Green's Function-based approach is used however, an efficient method can be derived for the computation of the sensitivity with respect to any arbitrary technology parameters.

Assume that the capacitive problem has been solved and that the equivalent resistive network has been computed from the coefficient of induction matrix \mathbf{c} . Furthermore, let \mathbf{c} be scaled in such a way that the node-to-node admittance Y_{ij} and the ground admittance Y_{ii} can be computed directly using

$$Y_{ii} = \sum_{j=1}^N c_{ij}, \quad Y_{ij} = -c_{ij}. \quad (8.35)$$

Let us define \mathbf{Y} as a $N \times N$ matrix consisting of Y_{ii} on the diagonal and Y_{ij} everywhere else. Let us call $\partial \mathbf{Y} / \partial T_\ell$ the sensitivity of matrix \mathbf{Y} with respect to technology parameter T_ℓ . The components of the sensitivity matrix are terms $\partial Y_{ii} / \partial T_\ell$ on the diagonal and $\partial Y_{ij} / \partial T_\ell$ everywhere else. The terms are computed using

$$\frac{\partial Y_{ii}}{\partial T_\ell} = \sum_{j=1}^N \frac{\partial c_{ij}}{\partial T_\ell}, \quad \text{and} \quad \frac{\partial Y_{ij}}{\partial T_\ell} = -\frac{\partial c_{ij}}{\partial T_\ell}, \quad (8.36)$$

where N is the size of matrix \mathbf{c} . In order to derive $\partial c_{ij} / \partial T_\ell$, equation (8.13) is differentiated on both hand sides

$$0 = \frac{\partial \mathbf{P}}{\partial T_\ell} \mathbf{Q} + \mathbf{P} \frac{\partial \mathbf{Q}}{\partial T_\ell}, \quad (8.37)$$

where $\partial \bar{\Phi} / \partial T_\ell = 0$, by construction of the original problem. Solving (8.37) with respect to $\partial \mathbf{Q} / \partial T_\ell$ one obtains

$$\frac{\partial \mathbf{Q}}{\partial T_\ell} = -\mathbf{P}^{-1} \left(\frac{\partial \mathbf{P}}{\partial T_\ell} \mathbf{Q} \right). \quad (8.38)$$

Using the definition of c_{ij}

$$c_{ij} = \frac{Q_i}{\Phi_j} \Big|_{\Phi_k=0 \forall k \neq j}$$

and the fact that $\partial \bar{\Phi} / \partial T_\ell$ vanishes, we obtain

$$\frac{\partial c_{ij}}{\partial T_\ell} = \frac{1}{\Phi_j} \frac{\partial Q_i}{\partial T_\ell}, \quad (8.39)$$

where $\partial Q_i / \partial T_\ell$ is computed using equation (8.38).

Now, only the derivative $\partial \mathbf{P} / \partial T_\ell$, i.e. $[\partial p_{ij} / \partial T_\ell]_{i,j=1,\dots,N}$, remains to be computed. From equation (8.23), assuming zero-depth contacts and $T_\ell \neq \epsilon_N, d, a$ or b

$$\frac{\partial p_{ij}}{\partial T_\ell} = \frac{\Gamma_N \beta_N - \Gamma_N \dot{\beta}_N}{ab \epsilon_N \beta_N^2} + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} k_{mn} \frac{[\sin(m\pi \frac{a_2}{a}) - \sin(m\pi \frac{a_1}{a})] [\sin(m\pi \frac{a_4}{a}) - \sin(m\pi \frac{a_3}{a})]}{(a_2 - a_1)(a_4 - a_3)} \times$$

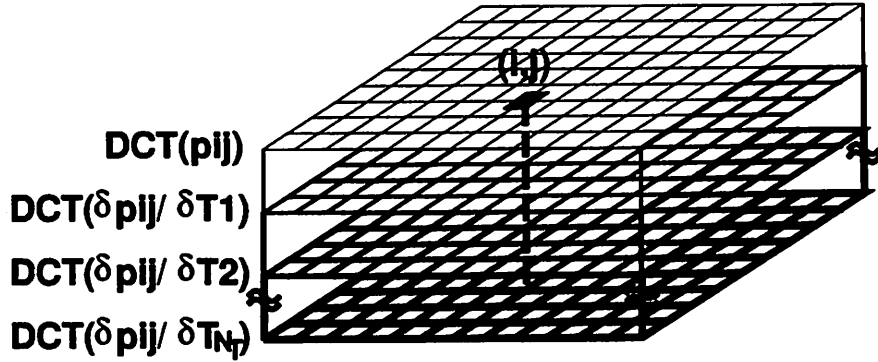


Figure 8.16: Storing one DCT for nominal parameter set and a number of DCTs for each computed sensitivity

$$\frac{[\sin(n\pi \frac{b_2}{b}) - \sin(m\pi \frac{b_1}{b})] [\sin(m\pi \frac{b_4}{b}) - \sin(m\pi \frac{b_3}{b})]}{(b_2 - b_1)(b_4 - b_3)}, \quad (8.40)$$

where $\dot{\Gamma}_N = \partial \Gamma_N / \partial T_\ell$, $\dot{\beta}_N = \partial \beta_N / \partial T_\ell$ and $\dot{k}_{mn} = \partial k_{mn} / \partial T_\ell$. Expressions for these derivatives have been derived in Appendix D.2. The extension of (8.40) for contacts with finite depth c can be found in Appendix D.2.

The first term of (8.40) can be easily calculated from the formulae in the appendix, while the second term can be efficiently computed using the DCT by replacing k_{mn} with \dot{k}_{mn} in equation (8.28). The DCT can be computed for each location in the grid and repeated for all parameters $T_\ell, \ell = 1, \dots, N_T$, where N_T is the number of technology parameters considered. To generate matrices $\partial \mathbf{c} / \partial T_\ell$ and $\partial \mathbf{P} / \partial T_\ell$, it is necessary to compute sensitivities $\partial p_{ij} / \partial T_\ell$ and $\partial c_{ij} / \partial T_\ell$ for all pairs of partition elements composing each contact. Figure 8.16 shows the data-structure used in our implementation for the storage of the various DCTs. Every sensitivity measure requires additional $N \times N$ storage, where N is the number of points in the grid of the DCT. As an example, assume $N_T = 10$, i.e. ten technology parameters T_ℓ are considered, moreover assume that a grid of 1024×1024 points is used. Then the total storage needed by our approach is 41.9 MByte, which is relatively low considering that a $1 \mu m$ resolution would be achieved on a 1×1 mm chip size.

Figure 8.17 illustrates the method used for the calculation of substrate sensitivities.

```

 $\dot{k}_{mn}$  = compute_derivative_kmn ( $k_{mn}$ );           // equation (D.9)
compute_DCT( $\dot{k}_{mn}$ );                               // DCT of the whole workspace
store_DCT_values;
foreach contact_part(i)                             // consider all pairs of partitions
  foreach contact_part(j)
     $\partial p_{ij}/\partial T_\ell$  = compute_derivative_pij (contact_part(i), contact_part(j));
 $\partial \mathbf{P}/\partial T_\ell$  = compose_matrix ( $\partial p_{ij}/\partial T_\ell$ );           // equation (8.40)
 $\partial \mathbf{c}/\partial T_\ell$  = get_c_sensitivities ( $\partial \mathbf{P}/\partial T_\ell$ );           // equation (8.39)
get_resistance_sensitivities( $\partial \mathbf{c}/\partial T_\ell$ );           // equations (8.36)

```

Figure 8.17: Pseudo-code of the substrate sensitivity extraction algorithm

8.5 Techniques for Substrate-Aware Optimization

In chapter 3 sensitivity-based constraint generation techniques have been presented for a number of discrete and high-frequency parasitics. Constraint generation has proven useful both during physical assembly and performance verification for the following reasons: (1) the existence of a constraint on a precise parasitic component can be used to guide a placer or a router to when a decision regarding a course of action is needed; (2) constraint violations can be easily detected and used to spot the causes of a system-level specification violation.

In this section we shall discuss the methods used in this dissertation for the computation of constraints associated with substrate parasitics. Moreover, we shall show how our Green's Function based analysis tools can be used for a fast and accurate evaluation of substrate parasitics and constraint enforcement during optimization.

8.5.1 Constraint Generation for Substrate Parasitic Effects

Constraint generation in a strict sense requires that parasitics be entities associated with one or more physical structures of the layout being made. In the case of switching noise the physical location and transmissions paths through the substrate may not be known before the general floorplan is performed on the chip. For this reason the constraint generation process cannot take place before the layout is, at least in part, generated, i.e. when constraints are mostly needed.

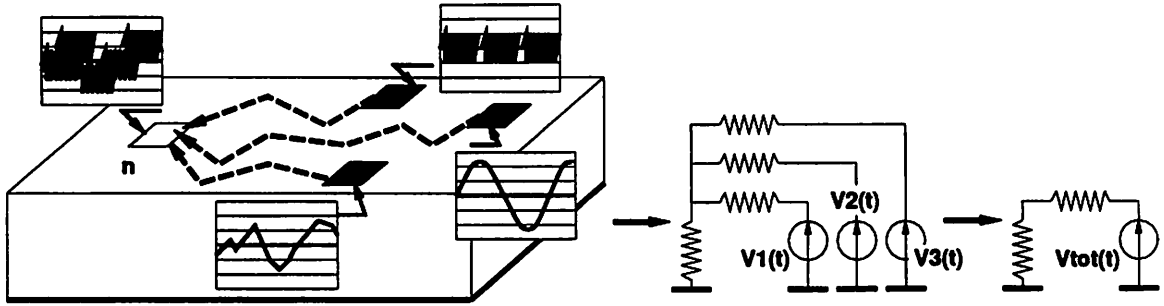


Figure 8.18: *The principle and modeling of local generators*

To address this issue we introduce the concept of *local noise generators*. A local noise generator is defined as a voltage or current source producing the equivalent of the cumulative noise contributed to by the real noise generators located in the substrate. The generator should simulate as closely as possible the waveform felt at a location n , including distortions, attenuations and group delays which transformed the original noise signal. Call n *sensing node*. See Figure 8.18. Let us call $g_n(t, \mathbf{\Pi})$ such waveform, where t is the time and $\mathbf{\Pi}$ is a vector of all the parameters relevant to it.

Let us define G_n a local noise generator producing waveform g_n . Due to the diverse nature of its parameters, $\mathbf{\Pi}$ can be split into its basic components $\mathbf{\Pi} = [\mathbf{W}^T \mathbf{G}^T T V_0]^T$. \mathbf{W} represents process-dependent and \mathbf{G} layout-related parameters, T is the temperature and V_0 the local substrate potential. One can also define vector $\Delta\mathbf{\Pi} = [\Delta\mathbf{W}^T \Delta\mathbf{G}^T \Delta T \Delta V_0]^T$ as the variation of $\mathbf{\Pi}$ from nominal.

Consider performance measure K_i , its degradation from nominal is given by the product of the i th row of $\mathbf{S}_{\mathbf{\Pi}}$ with vector $\Delta\mathbf{\Pi}$

$$\Delta K_i = (\mathbf{S}_{i,\mathbf{\Pi}})^T \Delta\mathbf{\Pi}, \quad (8.41)$$

where vector $\mathbf{S}_{i,\mathbf{\Pi}}$ is expressed as

$$\mathbf{S}_{i,\mathbf{\Pi}} = \begin{bmatrix} \mathbf{S}_{i,\mathbf{W}} \\ \mathbf{S}_{i,\mathbf{G}} \\ \mathbf{S}_{i,T} \\ \mathbf{S}_{i,V_0} \end{bmatrix}.$$

Suppose now that the exact waveform felt at n is not available, and only an estimate can be derived. Moreover, suppose that a range can be set for $\mathbf{\Pi}$

$$\mathbf{\Pi}^{(\min)} \leq \mathbf{\Pi} \leq \mathbf{\Pi}^{(\max)}, \quad (8.42)$$

where $\mathbf{\Pi}^{(\min)}$ and $\mathbf{\Pi}^{(\max)}$ are known vectors. Assuming that the sensitivity of performance K_i with respect to $\mathbf{\Pi}$ has been computed, bounds on all parameter variations $\Delta\mathbf{\Pi}^{(bound)}$ can be computed using constrained optimization. Hence, constraints are generated to bound the amount of noise at the sensing nodes *a priori*, without a precise knowledge of the structure of the layout being built. During the physical assembly of the circuit all the pre-computed constraints will be enforced separately on each component of the layout.

Let us now generalize the problem, by considering a large number of sensing nodes. From a theoretical standpoint at each receptor a different waveform could be felt. However, since the size of the analog section of a mixed-signal circuit is small compared to the distance to the noise sources, it is assumed that all the substrate nodes are reached by an identical waveform at different times.

Suppose M sensing nodes exist, each of them connected to a local generator $G_m(t - \tau_m, \mathbf{\Pi}_m)$, with $m = 1, \dots, M$, where τ_m is the propagation delay between nodes. Due to the highly non-linear dependence of performance on phase, an additive linearization around a nominal value could inaccurately model the parasitic effects of substrate.

The problem can be effectively addressed by deriving a set of worst-case sensitivities as described in chapter 3. Call $\mathbf{\Pi}'$ the array of all design parameters for which K_i is not strongly non-linear. Hence, the total linearized worst-case variation of K_i , due to node m , is derived as

$$\Delta K_i|_m = (\bar{\mathbf{S}}_{i,\mathbf{\Pi}'_m})^T \Delta\mathbf{\Pi}'_m . \quad (8.43)$$

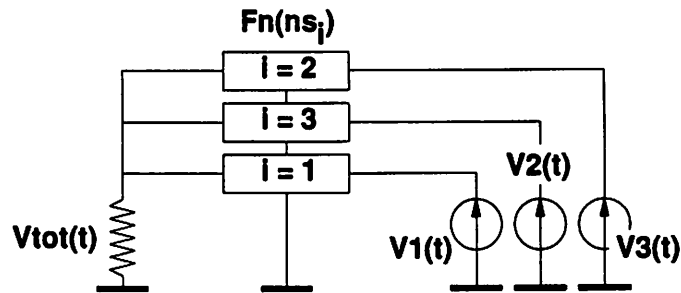
Using the same formalism of (8.41) and considering all the sensing nodes m in the circuit, we can define the matrices

$$[\bar{\mathbf{S}}_{i,\mathbf{\Pi}'}] = \begin{pmatrix} (\bar{\mathbf{S}}_{i,\mathbf{\Pi}'_0})^T \\ \vdots \\ (\bar{\mathbf{S}}_{i,\mathbf{\Pi}'_{M-1}})^T \end{pmatrix} \quad \text{and} \quad [\Delta\mathbf{\Pi}'] = \begin{pmatrix} (\Delta\mathbf{\Pi}'_0)^T \\ \vdots \\ (\Delta\mathbf{\Pi}'_{M-1})^T \end{pmatrix} . \quad (8.44)$$

Thus, the degradation of performance K_i is expressed as

$$\Delta K_i = \text{trace} \left(\bar{\mathbf{S}}_{i,\mathbf{\Pi}'} \Delta\mathbf{\Pi}' \right) . \quad (8.45)$$

Equation (8.45) models the contributions of all sensing nodes onto performance K_i . Bounds on the parameters associated with each sensing node $\Delta\mathbf{\Pi}'_n^{(bound)}$ can be computed using constrained optimization provided that conservative upper- and lower-bounds on the realization of $\mathbf{\Pi}$ are also available for each sensing node n .

Figure 8.19: *Constraint check*

The use of worst-case sensitivity matrix $\bar{S}_{i,\Pi}$ has the advantage of reducing the parameter space of Π . Secondly, non-linear behavior in a certain range of performance can be accurately modeled.

Due to the mechanism of noise modeling obtained using local generators, constraints on noise parameters can be derived independently of a particular IC process. Hence the constraint generation is required only once for a given circuit. During physical assembly, process-dependent substrate extraction, in combination with estimates of the sources of switching noise, are used to enforce the bounds. Furthermore, the effect of substrate noise can be evaluated locally, without taking into consideration neither the exact floorplan nor the actual position of the noise sources¹⁰. Once the substrate has been extracted, a transfer function $F_n(ns_i)$ can be computed relating each noise source ns_i to receptor n . Assuming that approximations or exact waveforms are known for each noise source, waveform $g_n(t, \Pi_n)$ and the corresponding parameter Π_n can be easily evaluated for each node n . Thus a simple check can be performed to verify that constraints $\Delta\Pi_n^{(bound)}$ and hence the original specifications have been met. See Figure 8.19.

8.5.2 Substrate Transport Evaluation in Iterative Algorithms

In chapters 4, 5 and 6 several tools used during the physical assembly of analog and mixed-signal ICs have been discussed. Each tool uses partially available parasitic information to perform a particular optimization within the workspace. The parasitics considered by the optimizer are generally localized to specific areas of the workspace. Hence, possibly global effects of parasitics can be often ignored. The algorithms used in floorplanning and

¹⁰A local noise generator can be seen as a model for an antenna.

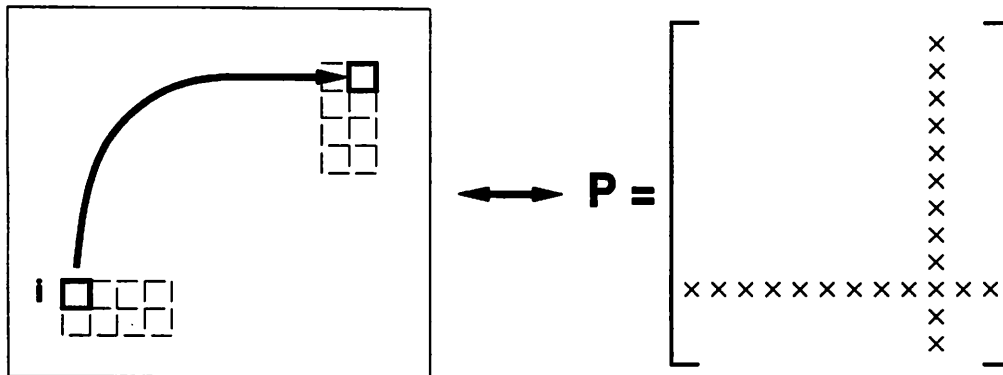


Figure 8.20: *Contact transformation and modifications in the potential matrix*

placement are based on incremental improvement techniques, consequently it is possible to derive compact and efficient ways of evaluating the degradation of performance due to parasitics while the optimization unfolds.

On the contrary, due to its “global” effects felt everywhere in the chip, substrate noise cannot be translated into a compact analytical model accounting for the entire substrate area. Hence, even if a small incremental modification is performed on the chip, the whole substrate analysis needs to be reevaluated. The traditional approach to this problem consists of using a method based on Finite Differences. To reduce the time complexity of the problem the density of the mesh that mimics the substrate bulk is drastically simplified, thus resulting in an accuracy reduction [70, 174]. In doing so however the estimation of switching noise might reach such inaccuracy levels that the insights gained applying this method might not be beneficial but misleading, thus possibly resulting in sub-optimal solutions.

The techniques proposed hereafter are based on our Green’s Function based method and are designed to allow very fast estimation of *variations* and *trends* within computationally expensive algorithms. The methods are to be published in [262].

Sherman-Morrison Update

The first technique exploits the fact that small adjustments in the position and orientation of layout elements results in a small change in the potential matrix \mathbf{P} . Figure 8.20 shows a contact (dotted lines) and its internal partitions when moving from one location to another. Consider a single partition i within the moving contact (solid line in Figure 8.20). Matrix

\mathbf{P} will change only in the locations marked by the “X” in the figure.

In order to recompute the substrate macro-model in such a case, it is unnecessary to perform the LU decomposition and the inversion steps. If matrix $\mathbf{c} = \mathbf{P}^{-1}$ or the LU decomposition of \mathbf{P} is known before the move has been performed, then it suffices to use the *Sherman-Morrison formula* for the computation of the updates.

Let \mathbf{P}' be the potential matrix associated with the new configuration. When partition i is moved to a new location, then row r and column c , i.e. the indices associated with i , will change. Let $\delta\mathbf{P}_{r.}$ and the r th row and $\delta\mathbf{P}_{.c}$ the c th column of \mathbf{P} affected by the change, then $\mathbf{P}' = \mathbf{P} + \delta\mathbf{P}_{.c} + \delta\mathbf{P}_{r.}$. For simplicity consider only the modification due to $\delta\mathbf{P}_{r.}$. Using the Sherman-Morrison formula, \mathbf{P}'^{-1} can be computed *directly* as

$$\mathbf{c}' = \mathbf{P}'^{-1} = \mathbf{c} + \delta\mathbf{c}, \text{ with } \delta\mathbf{c} = -\frac{\mathbf{c}_{.r}(\mathbf{c} \delta\mathbf{P}_{r.})}{1 + \delta\mathbf{P}_{r.} \mathbf{c}_{.r}}, \quad (8.46)$$

where $\mathbf{c}_{.r}$ is the r th column of \mathbf{c} .

The impedance and admittance networks associated with a substrate configuration are fully specified by $N_c \times N_c$ matrices $\bar{\mathbf{R}}$ and $\bar{\mathbf{Y}}$ which relate each pair of *contacts* through the corresponding impedance/admittance. $\bar{\mathbf{R}}$ and $\bar{\mathbf{Y}}$ can be derived from the coefficient of induction matrix \mathbf{c} using equations (8.9), (8.10), (8.14) and (8.15). Assuming appropriate scaling of \mathbf{c} (see Appendix C.2) one can easily show that a direct relation exists between \mathbf{c} and $\bar{\mathbf{Y}}$ through mapping $\bar{\mathbf{X}}$

$$\bar{\mathbf{Y}} = \bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}, \quad (8.47)$$

where $N \times N_c$ matrix $\bar{\mathbf{X}}$ is defined in Appendix C.2. N_c and N are the number of contacts and of contact partitions, respectively. Due to the structure of $\bar{\mathbf{X}}$, equation (8.47) only involves $(N - 1)^2$ summations, thus the computation complexity is dominated by the N^2 multiplications of the Sherman-Morrison update. The elements of matrix $\bar{\mathbf{R}}$ are computed simply using the relation $\bar{R}_{ij} = 1/\bar{Y}_{ij}$.

Sometimes it is useful to refer to another set of impedance/admittance matrices \mathbf{R} and \mathbf{Y} , which relate each pair of *contact partitions* through the corresponding impedance/admittance. The relation between \mathbf{Y} and \mathbf{c} is given by equation

$$\mathbf{Y} = \mathbf{X}^T \mathbf{c} \mathbf{X}, \quad (8.48)$$

where \mathbf{X} and \mathbf{X}^T are $N \times N$ identity matrices.

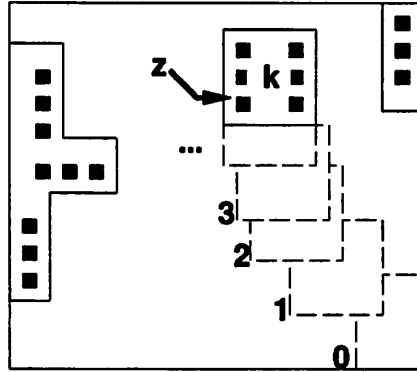


Figure 8.21: *Sensitivity of resistive macro-model from transformation of a component and its contacts*

Gradient-Based Method

The second technique is based on the concept of *sensitivity to relocation*. Suppose that a contact or a collection of contacts z is to be relocated on the substrate surface from location \mathbf{x}_0 to \mathbf{x}_k going through intermediate locations $\mathbf{x}_1, \dots, \mathbf{x}_{k-1}$. See Figure 8.21. One can easily show that

$$[\mathbf{c}]_k = [\mathbf{c}]_0 + \sum_{n=1}^k [\delta\mathbf{c}]_n ,$$

where $[\mathbf{c}]_0$ is the coefficient of induction matrix associated with location \mathbf{x}_0 , and $[\delta\mathbf{c}]_{n+1} = [\mathbf{c}]_{n+1} - [\mathbf{c}]_n$ is the $(n+1)$ th update of \mathbf{c} . The updates $[\delta\mathbf{c}]_{n+1}$ can be computed using the Sherman-Morrison formula in $O(N^2)$ time.

To further speed-up the computation one can exploit the “gradient” information of resistive and conductive networks $\bar{\mathbf{R}}$ and $\bar{\mathbf{Y}}$, contained in $[\delta\mathbf{c}]_1$. For simplicity but without loss of generality, consider the case in which each contact coincides with a single partition, i.e. $N_c = N$ or $\bar{\mathbf{R}} = \mathbf{R}$ and $\bar{\mathbf{Y}} = \mathbf{Y}$. The generalization of the concepts presented hereafter only requires the replacement of mapping \mathbf{X} with $\bar{\mathbf{X}}$ in each equation where it appears. Assume that a single contact z is relocated in direction \mathbf{v} by an amount $|\mathbf{v}| \rightarrow 0$ as in Figure 8.22. Let us define the vector $\nabla_{\mathbf{v}}\mathbf{Y}$ to be

$$\nabla_{\mathbf{v}}\mathbf{Y} = [\mathbf{A}, \mathbf{B}]^T, \quad \text{with } \mathbf{A} = \frac{\partial\mathbf{Y}}{\partial v_x}, \quad \mathbf{B} = \frac{\partial\mathbf{Y}}{\partial v_y}, \quad \text{and } \mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}.$$

The components of matrix \mathbf{A} are defined as $A_{ij} = \partial Y_{ij} / \partial v_x$, those of \mathbf{B} as $B_{ij} = \partial Y_{ij} / \partial v_y$. Recall that Y_{ij} is defined as the mutual admittance between contact partitions i and j for a given substrate configuration and that Y_{ii} is the ground admittance of i .

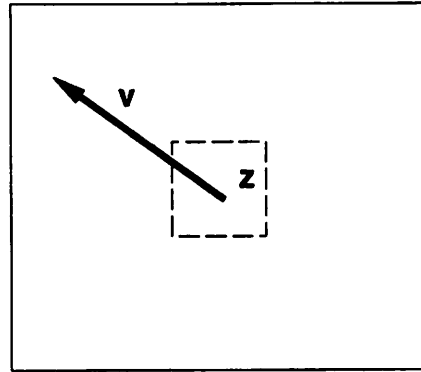


Figure 8.22: Single contact moving in direction \mathbf{v} by an infinitesimal amount

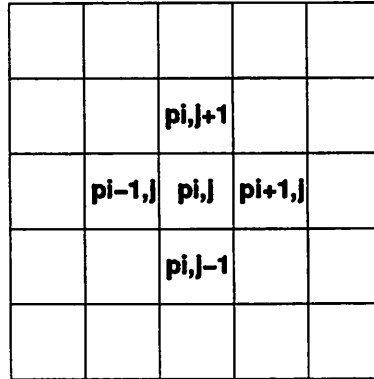


Figure 8.23: Computation of $\delta p_{i,j}$

The minimum step size in x - and y -direction corresponds a unit of the grid of the DCT. Hence, matrix $\partial \mathbf{Y} / \partial \mathbf{v}_x$ can be approximated by first computing differences $\delta p_{i,j \pm 1}$ and $\delta p_{i \pm 1,j}$ using equation (8.49). See Figure 8.23.

$$\delta p_{i,j \pm 1} = p_{i,j \pm 1} - p_{i,j}, \text{ and } \delta p_{i \pm 1,j} = p_{i \pm 1,j} - p_{i,j}. \quad (8.49)$$

Then, each component $\partial Y_{ij} / \partial v_x$ is calculated by replacing term c_{ij} with $\delta c_{i,j+1}$ in equations (8.9), (8.10), (8.14) and (8.15). Notice that term $\delta c_{i,j+1}$ is derived directly from matrix \mathbf{c} and $\delta p_{i,j+1}$ using the Sherman-Morrison formula. Moreover, the direct replacement of c_{ij} in the equations is legitimated by the fact that all manipulations are linear.

The same method is used to derive $\partial \mathbf{Y} / \partial v_y$. The time complexity of the operation is $O(N^2)$ since the Sherman-Morrison formula needs be repeated for all the contacts or partitions involved in the move.

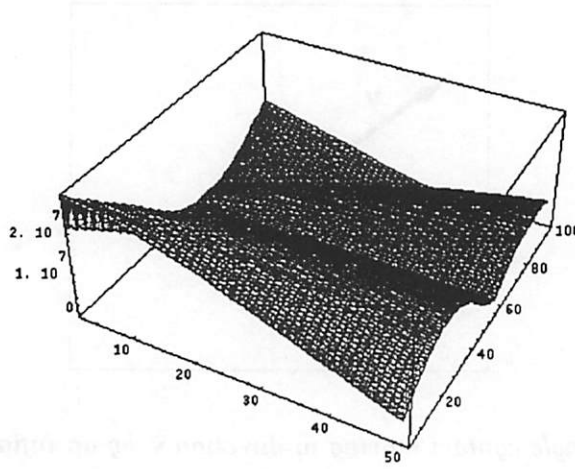


Figure 8.24: 200×200 DCT of the Green's Function for a commercial substrate

Let us assume that $\partial \mathbf{Y} / \partial v_x$ and $\partial \mathbf{Y} / \partial v_y$ have been computed at the 0th step of our incremental algorithm. Call $[\partial \mathbf{Y} / \partial v_x]_0$ and $[\partial \mathbf{Y} / \partial v_y]_0$ these matrices.

Assuming that the moving partition, contact or collection of contacts remains *close enough* to its position of step 0, then the conductance matrix at steps $1 \leq n \leq k$ can be approximated as

$$[\mathbf{Y}]_n \approx [\mathbf{Y}]_0 + \left[\frac{\partial \mathbf{Y}}{\partial v_x} \right]_0 \Delta v_x + \left[\frac{\partial \mathbf{Y}}{\partial v_y} \right]_0 \Delta v_y = [\mathbf{Y}]_0 + [\nabla_{\mathbf{v}} \mathbf{Y}^T]_0 \mathbf{v}, \quad (8.50)$$

where $\mathbf{v} = [\Delta v_x, \Delta v_y]^T$ is the vector representing the move of contact or partition z from step 0 to n .

The Green's Function and its DCT are well-behaved functions everywhere in the workspace [255]. Hence, necessarily terms $\delta p_{i,j \pm 1} < \infty$ and $\delta p_{i \pm 1,j} < \infty$. Figure 8.24 shows the plot of a 200×200 grid point DCT for a commercial substrate. No "high-frequency" components are present in the function, which makes it an ideal candidate for a highly accurate use of a gradient-based method. In fact, in our experiments the method has shown a 1% accuracy when the move occurred in the vicinity of the position at step 0, while a 10% accuracy was reached when the move was up to one tenth of the chip size.

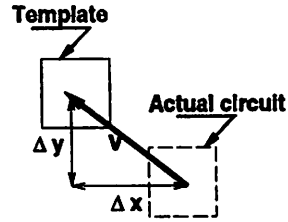


Figure 8.25: Computation of update matrix δc based on contact displacement relative to template

```

eliminate_non_crit_contacts(circuit description);
map_template(circuit description);           // use criterion (8.52)
c = c(template);
foreach contact
  modify_p_matrix;
  delta c = compute_c_update;               // use Sherman-Morrison formula
  c = c + delta c;
Y(actual) = compute_sigma(c);             // equation (8.51)

```

Figure 8.26: Pseudo-code of the template-based substrate extraction algorithm

8.5.3 Template-Based Substrate Extraction

In section 8.2 a technique was presented to speed-up the extraction process and to simplify the schematic based on the knowledge of contact loading. In this section we discuss a method for further reduction of the extraction time of large circuits that share a set of recurring contact patterns.

The technique is described in Figure 8.26. First, a set of templates with N_c or more contacts, for which an extracted schematic exists, is compared to the sample layout. Among the available ones, a template is selected and its pre-computed coefficient of induction matrix $c^{(\text{template})}$ is used to compute $c^{(\text{actual})}$, the matrix associated with the actual circuit. Each progressive update matrix δc is computed based on the displacement $\mathbf{v} = [\Delta x, \Delta y]^T$ of each contact non overlapping exactly with a corresponding contact in the template, as shown in Figure 8.25. Finally, the partial conductance matrix $\mathbf{Y}^{(\text{actual})}$ is computed directly from $c^{(\text{actual})}$ using equation (8.48)

$$\mathbf{Y}^{(\text{actual})} = \mathbf{X}^T \mathbf{c}^{(\text{actual})} \mathbf{X} . \quad (8.51)$$

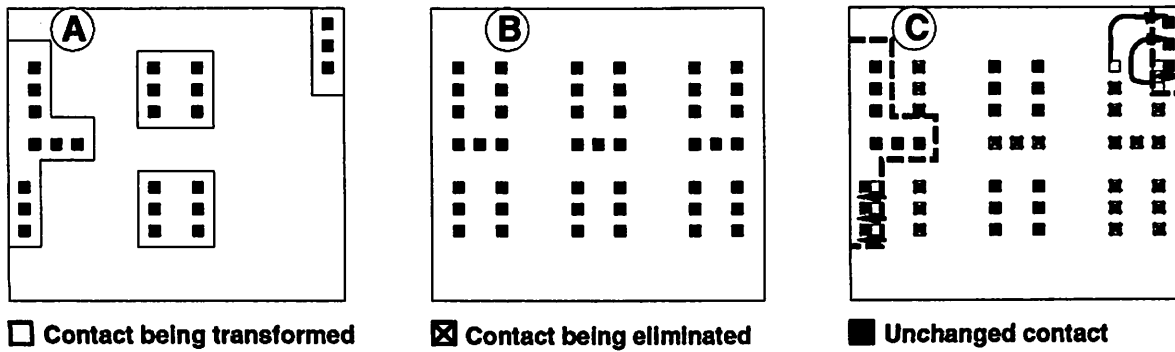


Figure 8.27: Speed-up mechanism for the extraction of large substrates

Figure 8.27A shows an example of physical layout being extracted. The template selected for this circuit is shown in Figure 8.27B. The procedure of eliminating and aligning some of the contacts of the template onto the actual circuit is shown in Figure 8.27C.

In order to derive bounds on the time complexity of the procedure, consider the following cases. First, assume the worst-case scenario, i.e. no contact exists which overlaps exactly with a contact in the template. In this case, N updates are needed for complete substrate evaluation, the resulting complexity is therefore $O(N^3)$. This case is equivalent to a full inversion of matrix \mathbf{P} , hence no improvement is achieved over the non-simplified substrate extraction.

Consider now the case in which the sample and the template are identical. In this case no computation is needed, hence the extraction complexity is zero.

The second scenario, or one as near as possible to it, is most desirable. Since, the complexity of computing an update of matrix \mathbf{c} is independent of the transformation involving the transformation, an effective criterion for selecting the template is:

$$\begin{array}{l} \text{maximize : } N_o \\ \text{all templates} \end{array} \quad (8.52)$$

where N_o is the number of contacts exactly overlapping a contact in the actual circuit layout. Consequently, assuming that $N_o < N$ contacts differ in location from corresponding contacts on the template, the complexity of the procedure could be a fraction of that needed to invert \mathbf{P} .

In real circuits however, a large number of contacts rarely overlaps to those on the template. To cope with this problem, we propose a criterion based on performance sensitiv-

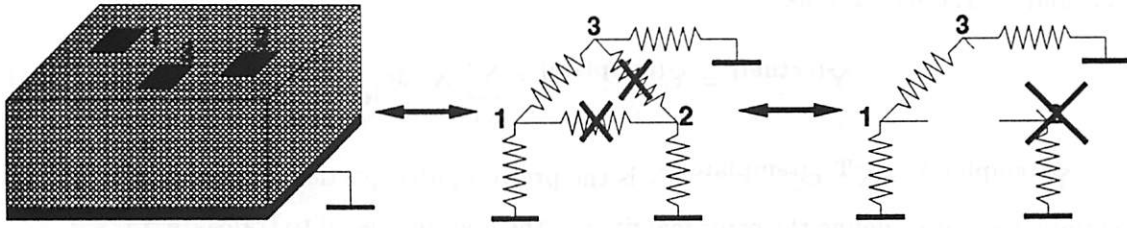


Figure 8.28: *Elimination of all non-critical conductances and contacts*

ities for the template selection and the minimization of updates needed for full extraction given pre-defined accuracy constraints. The modified template-based substrate extraction algorithm is described in Figure 8.29. For simplicity but without loss of generality, let us consider only one performance function K . Assume that the sensitivities of K with respect to all partial conductances Y_{ij} have been computed or estimated. Moreover, assume that estimates exist for the maximum values of all substrate conductances¹¹. Using equation (3.13) all non-critical conductances are eliminated from the schematic of all substrate couplings. All nodes connected to one or less conductances are also eliminated as illustrated in Figure 8.28. The resulting substrate configuration must be then compared with a set of templates and the best template must be selected. This problem is solved using optimization. A byproduct of the selection procedure is the set D of the contacts that need be extracted in all details. The displacements of the contacts in D , relative to the selected template, are identified and the updates needed for the computation of $\mathbf{c}^{(\text{actual})}$ are computed using the Sherman-Morrison formula. Partial conductance matrix $\mathbf{Y}^{(\text{actual})}$ is finally derived directly from $\mathbf{c}^{(\text{actual})}$ using (8.51).

Hereafter the template selection procedure is illustrated. Let us consider the matrix update $\delta\mathbf{c}|_i$ representing the move of contact i from its location in the template to that of the actual circuit. The coefficient of induction matrix $\mathbf{c}^{(\text{actual})}$ associated with the actual circuit is computed as

$$\mathbf{c}^{(\text{actual})} = \mathbf{c}^{(\text{template})} + \sum_{i \in D} \delta\mathbf{c}|_i, \quad (8.53)$$

where D is the set of all the contacts whose location in the template and in the actual circuit are non-identical and hence need be extracted in full detail. Combining equations

¹¹Rough estimates of the maximum/minimum value of substrate conductances can be easily computed from a simple set-up of two contacts located at chip edges or in close proximity

(8.53) and (8.51) one obtains

$$\mathbf{Y}^{(\text{actual})} = \mathbf{Y}^{(\text{template})} + \sum_{i \in D} \mathbf{X}^T \delta \mathbf{c} \Big|_i \mathbf{X} , \quad (8.54)$$

where $\mathbf{Y}^{(\text{template})} = \mathbf{X}^T \mathbf{c}^{(\text{template})} \mathbf{X}$ is the pre-computed partial conductance matrix of the template. Let us define the error matrix, i.e. the update needed to translate $\mathbf{Y}^{(\text{template})}$ into $\mathbf{Y}^{(\text{actual})}$, as

$$\boldsymbol{\epsilon} = \mathbf{Y}^{(\text{actual})} - \mathbf{Y}^{(\text{template})} = \sum_{i \in D} \boldsymbol{\epsilon} \Big|_i , \quad (8.55)$$

where $\boldsymbol{\epsilon} \Big|_i = \mathbf{X}^T \delta \mathbf{c} \Big|_i \mathbf{X}$ is the error matrix due to the displacement of contact i in the actual circuit relatively to the template¹².

Assume one could calculate $\boldsymbol{\epsilon} \Big|_i$, $\forall i \in D$ *a priori*. Using the sensitivity¹³ of performance K with respect to matrix \mathbf{Y} , one could calculate performance degradation ΔK due to the displacement of contacts in the actual circuit relatively to the template as

$$\Delta K = \mathbf{e}^T \left(\frac{\partial K}{\partial Y_{ij}} \odot \boldsymbol{\epsilon} \right) \mathbf{e} , \quad (8.56)$$

where \mathbf{e} is a $N \times 1$ unity vector such that $\mathbf{e} = [1, \dots, 1]^T$. The \odot operator is defined as following: $\mathbf{A} = \mathbf{B} \odot \mathbf{C} \Leftrightarrow a_{ij} = b_{ij} c_{ij}$. Combining (8.55) and (8.56), one obtains

$$\Delta K = \sum_{i \in D} \mathbf{e}^T \left(\frac{\partial K}{\partial Y_{ij}} \odot \boldsymbol{\epsilon} \Big|_i \right) \mathbf{e} .$$

Let us define *weighted extraction inaccuracy* A_K of an extracted schematic with respect to performance K as the relative absolute amount by which K varies if some or all parasitics are inexactly estimated. The weighted extraction inaccuracy is expressed as

$$A_K = \frac{|\Delta K| + \epsilon_p + \epsilon_r}{K_v} , \quad (8.57)$$

where ϵ_p and ϵ_r are the errors due to inaccurate parasitic and performance models, respectively, and K_v is the nominal performance value.

If $\epsilon_p + \epsilon_r \ll |\Delta K|$, equation (8.57) reduces to

$$A_K \approx \frac{|\Delta K|}{K_v} . \quad (8.58)$$

¹²Assume all the other contacts are not displaced.

¹³The sensitivity of K with respect to matrix \mathbf{Y} is a $N_c \times N_c$ matrix, whose terms in the i th row and j th column are given by the expression $\partial K / \partial Y_{ij}$.

```

eliminate_non_crit_contacts(circuit description); // equation (3.13)
map_template(circuit description); // use criterion (8.60)
c = c(template);
foreach contact ∈ D
    modify_p_matrix;
    δc = compute_c_update;
    c = c + δc;
Y(actual) = compute_sigma(c); // equation (8.51)

```

Figure 8.29: Pseudo-code of the modified template-based substrate extraction algorithm

Suppose a constraint on the weighted accuracy \bar{A}_K has been set, such that

$$A_K \leq \bar{A}_K . \quad (8.59)$$

Then, equations (8.58) and (8.59) can be used as a criterion for selecting the appropriate template

$$\begin{array}{ll}
 \text{minimize : } D & (8.60) \\
 \text{all templates} & \\
 \text{subject to:} & \\
 & A_K \leq \bar{A}_K .
 \end{array}$$

Problem (8.60) is *guaranteed* to have a solution, since a template with at least N_c contacts, all of them not overlapping with the actual circuit's contacts, exists by construction. Hence, arbitrarily small values of A_K can be achieved by simply extending D to include all the contacts i , $1 \leq i \leq N_c$. Optimization (8.60) is solved by exhaustively calculating the minimum set D needed for each template for a given inaccuracy A_K . The procedure of calculating A_K and D has a time complexity of $O(N^2)$, while the overhead of computing S_{Y_i} is generally not accounted for since the evaluation is performed beforehand during circuit synthesis. Hence, a circuit with N_c contacts and a specification on A_K (8.59) can be extracted in

$$(N_T + |D|) N^2$$

time, where N_T is the number of template circuits and $|D|$ the size of set D .

The final issue to be addressed is the efficient calculation of estimate $\epsilon|_i$. Term $\epsilon|_i$ can be computed exactly from update $\delta c|_i$ using mapping \mathbf{X} (8.54). However, a more efficient

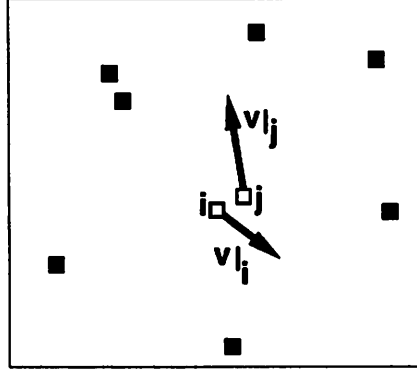


Figure 8.30: *Similar landscape and displacement of contact i and j*

computation of $\epsilon|_i$, can be obtained using the approximation of (8.50). Consider all the contacts $i \in D$, assume that the locations of i in the template and in the actual circuit are *close enough*. Then, a two-dimensional Taylor expansion for $\delta c|_i$ can be constructed as

$$\delta c|_i \approx \left. \frac{\partial c}{\partial v_x} \right|_i \Delta v_x + \left. \frac{\partial c}{\partial v_y} \right|_i \Delta v_y = \nabla_{\mathbf{v}} c|_i \mathbf{v}|_i, \quad (8.61)$$

where vector $\mathbf{v}|_i = [\Delta v_x, \Delta v_y]^T$ represents the displacement needed to bring i from the template location to the location in the actual circuit. Term $\nabla_{\mathbf{v}} c|_i = \left[\left. \frac{\partial c}{\partial v_x} \right|_i, \left. \frac{\partial c}{\partial v_y} \right|_i \right]^T$ is calculated using the Sherman-Morrison formula as in (8.50) and is valid for *small* displacements of contact i .

Assume now that there exists a contact j in the vicinity of i which is displaced by $\Delta \mathbf{v}|_j$, where $|\Delta \mathbf{v}|_j|$ is also *small*. Assuming that the surrounding objects' relative distances from i and j are similar, one can estimate the cumulative effects of the displacement of the contacts as

$$\delta c|_{i,j} \approx \nabla_{\mathbf{v}} c|_i \mathbf{v}|_i + \nabla_{\mathbf{v}} c|_i \mathbf{v}|_j, \quad (8.62)$$

where vectors $\mathbf{v}|_i$ and $\mathbf{v}|_j$ relate to the displacements of i and j as shown in Figure 8.30.

Ideally, one would like to be able to compute $c|_i$ using equation (8.62) for each contact $i = 1, \dots, N_c$. However, far contacts “see” a completely different landscape, which causes term $\delta c|_i$ to change by moving within the workspace. To improve the accuracy of (8.62), one could partition of the workspace in order to minimize the number of contacts for which a new $\nabla_{\mathbf{v}} c$ need be computed. Figure 8.31 shows such a partitioning. Notice that only one contact per partition, the *pole*, is used for the computation of $\nabla_{\mathbf{v}} c$.

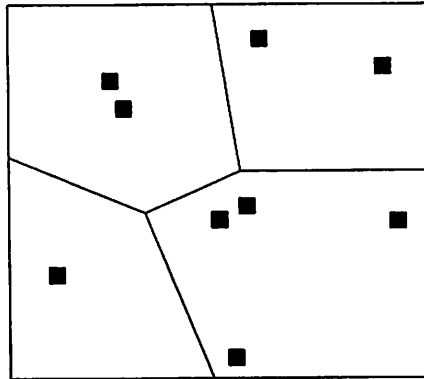


Figure 8.31: *Partitioning of substrate to minimize the number of different contacts for which $\nabla_{\mathbf{v}} \mathbf{c}$ need be computed explicitly*

The problem of minimizing the number of partitions of Figure 8.31 can be time-consuming, since it requires the estimation of each contact displacement to select the best candidates for the partitions and its poles. The complexity of this partitioning would nullify the efforts for an efficient substrate extraction. In addition, the needed parasitic estimate accuracy ϵ_p in equation (8.57) is not high. Hence, in our experiments a single contact was used to estimate $\delta \mathbf{c}_i, \forall i$ with an error of 50% or less. Moreover, this error could be modeled as term ϵ_p in equation (8.57) and hence accounted for while determining D . Figure 8.32 shows the accuracy of substrate extraction for each contact in dependence of the two-dimensional displacement from its true location.

8.5.4 Evaluating Effects of Scaling and Technology Migration

In section 8.4 the theory was developed for the sensitivity computation of substrate conductances with respect to technology parameters such as doping profiles and geometries. In this section a sensitivity-based approach is proposed for efficient evaluation of the impact of technology when a process is modified in part or completely. Figure 8.33 shows the scaling of a chip in the occurrence of re-design or technology migration. Re-design generally involves scaling in x- and y- directions, while technology migration involves a three-dimensional scaling of the design. Hereafter we propose a generalized technique that can be used for both two- and three-dimensional scaling.

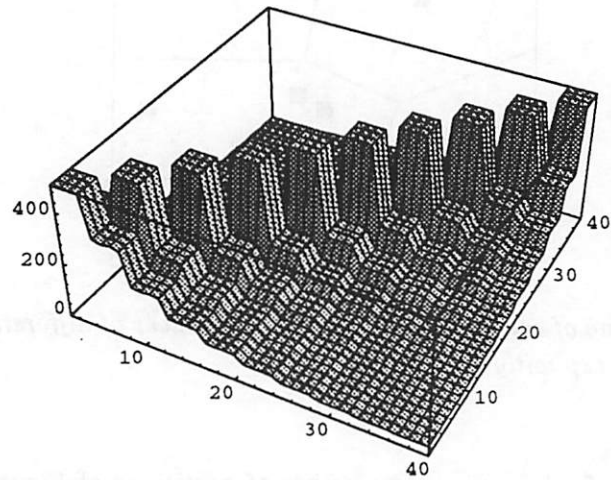


Figure 8.32: Accuracy in function of the distance of the true contact from the pre-computed contact

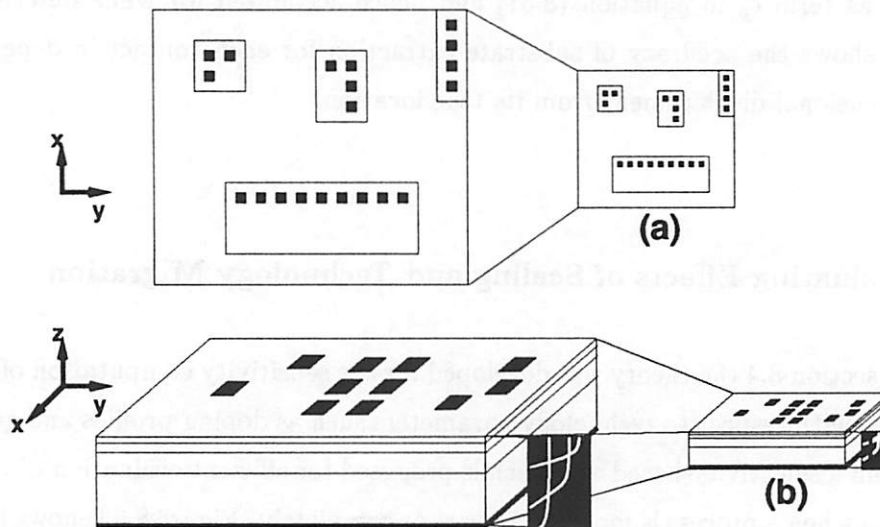


Figure 8.33: (a) Two-dimensional scaling in the event of re-design; (b) Three-dimensional scaling in technology migration

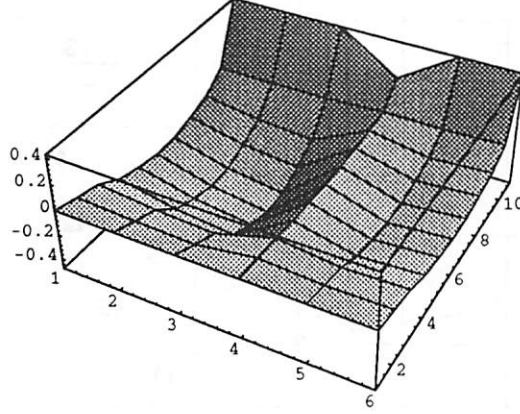


Figure 8.34: Plot of the dependence of each component of the \bar{R} matrix as a function of the contact layer depth

Deterministic Analysis

Consider first scaling in z-coordinate. Using equations (8.36), (8.38), (8.39), (8.40), and the expressions in Appendix D.2, one can efficiently compute matrix $\partial\mathbf{Y}/\partial T_\ell$. Let us define a number of technology parameters for some design \mathcal{D} , $T_\ell^{(\mathcal{D})}$, $\ell = 1, \dots, N_T$, which include layer thicknesses or profile discretizations d_k , $k = 1, \dots, N_d$ in Figure 8.7 and permittivity ϵ_k , $k = 1, \dots, N_\epsilon$. Call $\mathbf{T}^{(\mathcal{D})}$ the $N_T \times 1$ vector whose elements are the $T_\ell^{(\mathcal{D})}$ terms. As an illustration, the plot of Figure 8.34 shows the impedance of the three-contact configuration of Figure 8.28 as a function of the contact layer depth. Figure 8.35 shows a transversal cut of the plot of Figure 8.34, i.e. impedance R_{13} as a function of contact depth c . Lines (t_1, \dots, t_4) represent the sensitivities of R_{13} at several values of c as computed using the formulae in Appendix D.2.

Suppose now that admittance matrix $\mathbf{Y}^{(\mathcal{D})}$ has been calculated for a set of parameters $\mathbf{T}^{(\mathcal{D})}$. In addition, assume that an array of parameters $\mathbf{T}^{(\mathcal{D}')}$ associated with a new design \mathcal{D}' is also available. Define the $N_T \times 1$ vector $\Delta\mathbf{T}^{(\mathcal{D}\mathcal{D}')} = \mathbf{T}^{(\mathcal{D}')} - \mathbf{T}^{(\mathcal{D})}$, as the variation of technology parameters across designs \mathcal{D} and \mathcal{D}' . Admittance matrix $\mathbf{Y}^{(\mathcal{D}')}$ associated with the new design can be computed using a first order Taylor expansion as

$$\mathbf{Y}^{(\mathcal{D}')} \approx \mathbf{Y}^{(\mathcal{D})} + \sum_{\ell=1}^{N_T} \left[\frac{\partial \mathbf{Y}}{\partial T_\ell} \right]_{\mathbf{T}_\ell^{(\mathcal{D})}} \Delta T_\ell^{(\mathcal{D}\mathcal{D}')}, \quad (8.63)$$

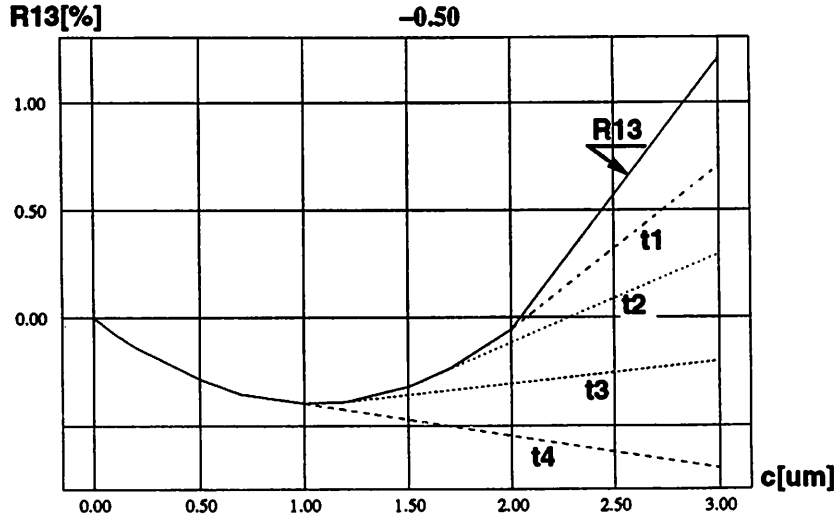


Figure 8.35: Plot of the dependence of \bar{R} as a function of the contact layer depth and related sensitivities

provided that designs \mathcal{D} and \mathcal{D}' are *close enough*, i.e. $\max_{\ell} \{|\Delta T_{\ell}^{(\mathcal{D}\mathcal{D}')}|/T_{\ell}^{(\mathcal{D})}\}$ is small.

As an illustration consider again the case of Figure 8.34. Table 8.1 lists the values of matrix \mathbf{Y} using full and sensitivity-based extraction for two configurations. All CPU times are referred to a DEC AlphaServer 2100 5/250 and relate to all computations except for the Green's Function.

Consider next scaling in (x, y) -direction. Assume that a contact i in design \mathcal{D} is located at a point $\mathbf{v}_i^{(\mathcal{D})} = [v_x, v_y]^T$, while a contact's position in design \mathcal{D}' is $\mathbf{v}_i^{(\mathcal{D}')}$. Furthermore, assume the the contact's area is not significantly changed across designs. Suppose that admittance matrix $\mathbf{Y}^{(\mathcal{D})}$ has been calculated for design \mathcal{D} and that vectors $\mathbf{v}_i^{(\mathcal{D})}$ and $\mathbf{v}_i^{(\mathcal{D}')}$ are given $\forall i = 1, \dots, N_c$. Let $\Delta \mathbf{v}_i^{(\mathcal{D}\mathcal{D}')}$ be the change in location for contact i as illustrated in Figure 8.36.

Using equation (8.50) one can approximate matrix \mathbf{Y} as follows

$$\mathbf{Y}^{(\mathcal{D}')} \approx \mathbf{Y}^{(\mathcal{D})} + \sum_{i=1}^{N_c} [\nabla_{\mathbf{v}_i} \mathbf{Y}]^T \Delta \mathbf{v}_i^{(\mathcal{D}\mathcal{D}')}, \quad (8.64)$$

Figure 8.37 shows the sensitivity of an entry in matrix \mathbf{Y} as a result of the displacement in x - and y - direction of a contact in a uniform grid of 10×10 contacts spaced by $10 \mu m$. Equations (8.63) and (8.64) can be combined so as to account for three-dimensional scaling realistically.

	method I	method II	error
R_{12}	8154.03 Ω	8098.60 Ω	0.67 %
R_{13}	1866.15 Ω	1848.10 Ω	0.96 %
R_{23}	3788.62 Ω	3743.80 Ω	1.18 %
R_{10}	893.85 Ω	893.08 Ω	0.08 %
R_{20}	460.40 Ω	458.59 Ω	0.39 %
R_{30}	690.77 Ω	688.59 Ω	0.31 %

configuration of Fig. 8.28

# contacts	method I	method II	max. error
100	73.8 sec	9.8 sec	3 %

uniform 10x10 contact grid

Table 8.1: Substrate extraction in presence of varying technology parameters using method I (full extraction) and method II (sensitivity-based extraction)

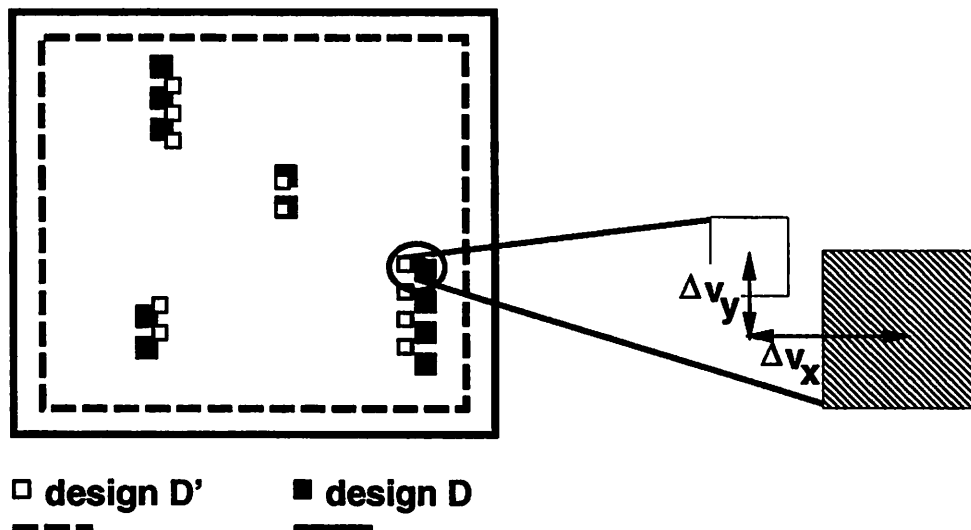


Figure 8.36: Scaling in x- and y-direction. Relocation of contacts and area scaling

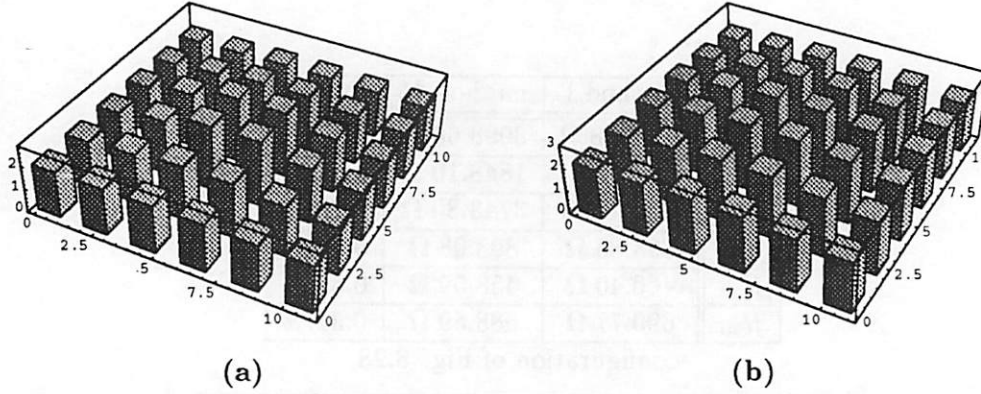


Figure 8.37: Sensitivity of entry Y_{55} in a 10×10 grid as a function of a translation in (a) x - and (b) y - direction of all the contacts in the grid

Non-Deterministic Analysis and Optimal Technology Selection

In the above discussion, we have assumed that the values of technology variations ΔT_ℓ and geometric displacements Δv_i are of a deterministic nature. Suppose on the contrary that we are given the statistical behavior of all or some technology parameters ΔT_ℓ , $\forall \ell = 1, \dots, N_T$. Assume that the terms ΔT_ℓ are random variables with mean μ_ℓ and variance σ_ℓ^2 , moreover suppose that all ΔT_ℓ are *statistically independent*.

Then, the mean $E(Y_{ij})$ and variance $\sigma^2(Y_{ij}) = E(Y_{ij}^2) - E^2(Y_{ij})$ of each entry of admittance matrix \mathbf{Y} can be computed as

$$E(Y_{ij}) \approx Y_{ij}^{(\mathcal{D})} + \sum_{\ell=1}^{N_T} \left[\frac{\partial Y_{ij}}{\partial T_\ell} \right]_{T_\ell^{(\mathcal{D})}} \mu_\ell,$$

and

$$\sigma^2(Y_{ij}) \approx \sum_{\ell=1}^{N_T} \left| \left[\frac{\partial Y_{ij}}{\partial T_\ell} \right]_{T_\ell^{(\mathcal{D})}} \right|^2 \sigma_\ell^2, \quad (8.65)$$

where $[\partial Y_{ij}/\partial T_\ell]_{T_\ell^{(\mathcal{D})}}$ is the sensitivity of entry Y_{ij} with respect to T_ℓ related to the original design \mathcal{D} .

As illustration, consider again the configuration of Figure 8.28 and the uniform 10×10 contact grid. Table 8.2 lists the mean and variance of the entries of matrix $\bar{\mathbf{R}}$ as a function of depth variance $\sigma^2(c)$, assuming $\mu(c) = 1\mu m$. The execution times for the extraction of the mean and variance of $\bar{\mathbf{R}}$ are also reported.

Our sensitivity-based method for the computation of mean and variance of \mathbf{Y} can

	$\mu(R)$	$\sigma^2(R)$	
	$\mu(c) = 1\mu m$	$\sigma^2(c) = 0.25\mu m^2$	$\sigma^2(c) = 0.1\mu m^2$
R_{12}	8066.33 Ω	0.0270 Ω^2	0.0109 Ω^2
R_{13}	1836.69 Ω	0.0040 Ω^2	0.0017 Ω^2
R_{23}	3716.29 Ω	0.0324 Ω^2	0.1296 Ω^2
R_{10}	892.11 Ω	$2.5 \times 10^{-7} \Omega^2$	$10^{-7} \Omega^2$
R_{20}	456.35 Ω	$10^{-4} \Omega^2$	$4 \times 10^{-5} \Omega^2$
R_{30}	685.69 Ω	$2.25 \times 10^{-4} \Omega^2$	$9 \times 10^{-4} \Omega^2$

configuration of Fig. 8.28

# contacts	$\mu(R)$	$\sigma^2(R)$
100	73.8 sec	16.0 sec

uniform 10x10 contact grid

Table 8.2: Mean and variance of the entries of matrix $\bar{\mathbf{R}}$ as a function of depth variance. All values are referred to a mean depth of $1\mu m$. The execution times are reported for a uniform 10x10 contact grid

also be used for the selection of a technology which is most suitable for a certain circuit and its associated performance specifications. Suppose for instance that N_p constraints on all critical substrate coupling $R_{ij}^{(bound)}$ have been computed using the techniques presented in chapter 3. Furthermore, assume that a number of technologies \mathcal{T} are available and that all relevant parameters $T_\ell^{(\mathcal{T})}$ are identified. Suppose however that for some or all technologies a number of parameters are not known precisely and only rough estimates with uncertainty exist. Assume that estimate and uncertainty can be modeled into each parameter in terms of its mean and variance. Then, by computing the mean and the variance of $\bar{\mathbf{R}}$ for a set of parameters $T_\ell^{(\mathcal{T})}$, one can derive the probability with which constraints $R_{ij}^{(bound)}$ will be met

$$P_{\mathcal{T}} \left[\mu(\bar{R}_{ij}), \sigma^2(\bar{R}_{ij}), R_{ij}^{(bound)} \right] = \text{erf} \left(\frac{R_{ij}^{(bound)} - \mu(\bar{R}_{ij})}{\sigma(\bar{R}_{ij})} \right), \quad (8.66)$$

provided that $T_\ell^{(\mathcal{T})}$ is gaussian. Notice that $\text{erf}(x)$ is defined here as the integral of a normal distribution $N(0,1)$ from minus infinity to x . The problem of selecting a technology most likely to satisfy all constraints is equivalent to solving the following problem

$$\underset{\mathcal{T}}{\text{maximize}} : \sum_{\text{all crit. constr.}} P_{\mathcal{T}} \left[\mu(\bar{R}_{ij}), \sigma^2(\bar{R}_{ij}), R_{ij}^{(bound)} \right] \quad (8.67)$$

Due the efficiency of our techniques for the calculation of means and variances, problem

	$R_{ij}^{(bound)}$	$P_{\mathcal{T}}(\mu(c)[\mu m]/\sigma^2(c)[\mu m^2])$	
		$\mathcal{T}_1: 0 / 0.1$	$\mathcal{T}_2: 1 / 0.25$
R_{12}	8066.5 Ω	0.849569	0.948270
R_{13}	1836.8 Ω	0.959005	0.996184
R_{23}	3716.4 Ω	0.729437	0.620028
R_{10}	893.0 Ω	1.0	1.0
R_{20}	457.0 Ω	1.0	1.0
R_{30}	685.8 Ω	1.0	0.999877
Total	-	5.538011	5.564359

Table 8.3: Selection of most suitable technology based on the probability of satisfying all constraints on substrate coupling resistances

(8.67) can be solved by exhaustively computing $P_{\mathcal{T}}[\mu(\bar{R}_{ij}), \sigma^2(\bar{R}_{ij}), R_{ij}^{(bound)}]$ for each technology \mathcal{T} .

As an illustration, consider the example of Figure 8.28. Suppose that all six substrate resistances R_{ij} and R_{i0} are critical and that constraints on each resistance have set as listed in Table 8.3. Clearly, technology \mathcal{T}_2 is more likely to meet the above specifications and hence it should be selected as best candidate.

Chapter 9

Experimentation

*“O frati”, dissi “che per cento milia
perigli siete giunti a l’occidente,
a questa tanto picciola vigilia*

*d’i nostri sensi ch’è del rimanente,
non vogliate negar l’esperienza,
di retro al sol, del mondo senza gente.*

*Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza”.*

Dante Alighieri, “Inferno”, Canto XXVI

9.1 Analog Benchmark Library

This section presents a number of examples of analog circuits whose physical assembly has been completed in an entirely automated fashion using the methodology and the tools presented in this dissertation. All the tools used in this section have been implemented within the OCTTOOLS framework of the University of California at Berkeley. This has allowed us to test the described algorithms, and to validate the methodological approach on a large set of test circuits. All sensitivity computation and simulations have been done using SPICE [137].

Capacitive		
Parasitic	bound	$C^{(max)}$
C_{15}	71.96fF	100fF
C_{55}	78.52fF	100fF
Resistive		
Parasitic	bound	$R^{(max)}$
R_{S_1}	1 Ω	1 Ω
R_{S_20}	7.4 Ω	50 Ω
R_{S_6}	19.9 Ω	50 Ω

Table 9.1: COMPL: *bounds on capacitive and resistive parasitics*

9.1.1 COMPL

The schematic of the circuit, reported in [142], is shown in Figure 3.4. the nominal values for its performance have been formally defined as

$$V_{off} = 0.0mV$$

$$\tau_D = 4.0ns$$

The specifications have been set to

$$|V_{off}| \leq 1.0mV$$

$$|\Delta \tau_D| \leq 3.0ns$$

Therefore

$$\mathbf{K} = \begin{bmatrix} \tau_D \\ V_{off} \\ -V_{off} \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 4.0 \text{ ns} \\ 0.0 \\ 0.0 \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 3.0ns \\ 10\mu V \\ 10\mu V \end{bmatrix}$$

Using PARCAR constraints on absolute values and mismatches were computed for all critical parasitics. Table 9.4 lists the CPU time for this operation. Table 9.1 shows some of the most critical parasitic constraints found by PARCAR. The circuit was placed, routed and compacted using the tools PUPPY-A, ROAD and SPARCS-A, respectively. ROAD [201] is a gridless area router based on the A* algorithm. PUPPY-A and SPARCS-A are described in chapters 4 and 6, respectively. Table 9.2 lists the conditions under which routing and compaction were used for this and the following circuits. The conditions under which the placer PUPPY-A operated are listed in Table 4.2. Figure 9.1a depicts the final layout obtained

Tool	Item	Conditions
ROAD	routing scheduling	by net type and size
	constraints	P , S and M enforced
SPARCS-A	first iteration	vertical
	constraints	P , S and M enforced

Table 9.2: *Conditions of operation for the routing and compaction tools used in the synthesis path. The symbols P , S and M denote parasitic, symmetry and matching constraints, respectively. The net scheduling is based on a cost function which accounts for the "difficulty" of enforcing a set of desired constraints on a given net*

	Unconstrained layout	Unconstrained compaction	Constrained compaction
V_{off}	$764\mu V$	$-13\mu V$	$8\mu V$
τ_D	9.05 ns	2.25ns	2.17ns

Table 9.3: COMPL: *performance*

enforcing the constraints in all steps of the assembly except during compaction. Figure 9.1b shows the result when constraints are enforced in all layout design phases. Table 9.3 lists the resulting performance under three different synthesis conditions: with no constraint enforcement, with constraint enforcement at all synthesis stages except compaction, and with full constraint enforcement. Table 9.4 shows the required CPU times on a DEC Station 5000/240 for unconstrained and constrained synthesis schemes.

Layout phase	Unconstrained	Constrained
Constraint generation (PARCAR)	-	0.1 sec
Placement (PUPPY-A-LDO)	182.8 sec	343.0 sec
Routing (ROAD)	51.0 sec	71.0 sec
Compaction (SPARCS-A)	4.4 sec	36.2 sec
TOTAL	238.2 sec	450.2 sec

Table 9.4: COMPL: *CPU time for each layout phase*

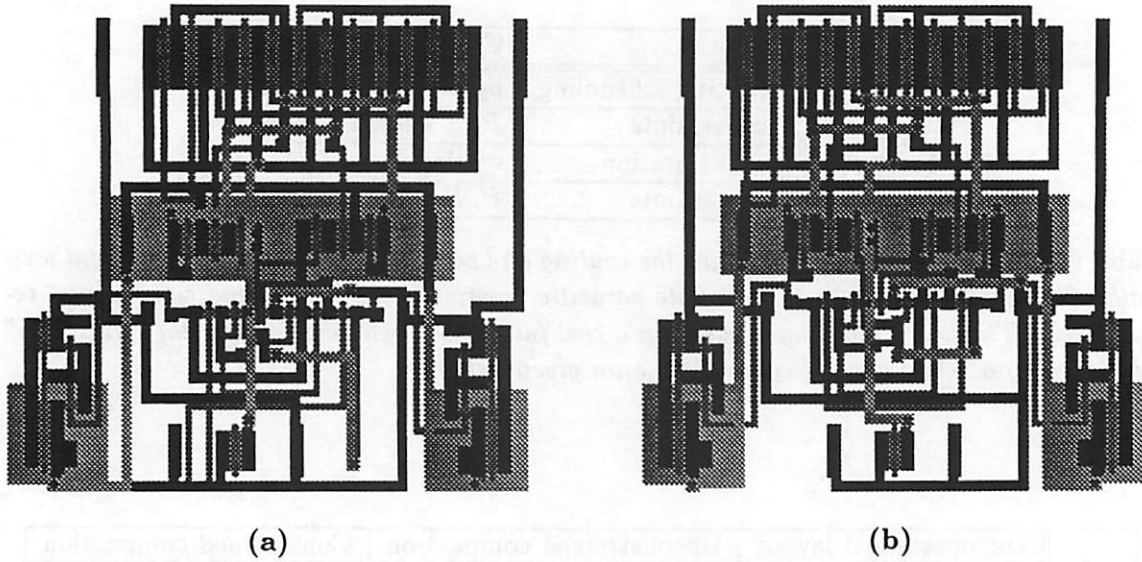


Figure 9.1: Complete layout of COMPL, (a) without enforcement, (b) with enforcement of analog constraints

9.1.2 FASTCOMP

Figure 9.2 shows the schematic of a clocked comparator named FASTCOMP. For this circuit we consider specifications on voltage offset and switching speed. The nominal values are

$$\begin{aligned} V_{off} &= 0.0mV \\ \tau_D(H \rightarrow L) &= 2.42ns \\ \tau_D(L \rightarrow H) &= 2.49ns \end{aligned}$$

The specifications have been set to

$$\begin{aligned} |V_{off}| &\leq 2.0mV \\ |\Delta \tau_D(H \rightarrow L)| &\leq 0.25ns \\ |\Delta \tau_D(L \rightarrow H)| &\leq 0.25ns \end{aligned}$$

Therefore

$$\mathbf{K} = \begin{bmatrix} V_{off} \\ -V_{off} \\ \tau_D(H \rightarrow L) \\ -\tau_D(H \rightarrow L) \\ \tau_D(L \rightarrow H) \\ -\tau_D(L \rightarrow H) \end{bmatrix} \quad \mathbf{K}(p^{(0)}) = \begin{bmatrix} 0.0 \\ 0.0 \\ 2.42ns \\ -2.42ns \\ 2.49ns \\ -2.49ns \end{bmatrix} \quad \overline{\Delta \mathbf{K}} = \begin{bmatrix} 2mV \\ 2mV \\ 0.25ns \\ 0.25ns \\ 0.25ns \\ 0.25ns \end{bmatrix}$$

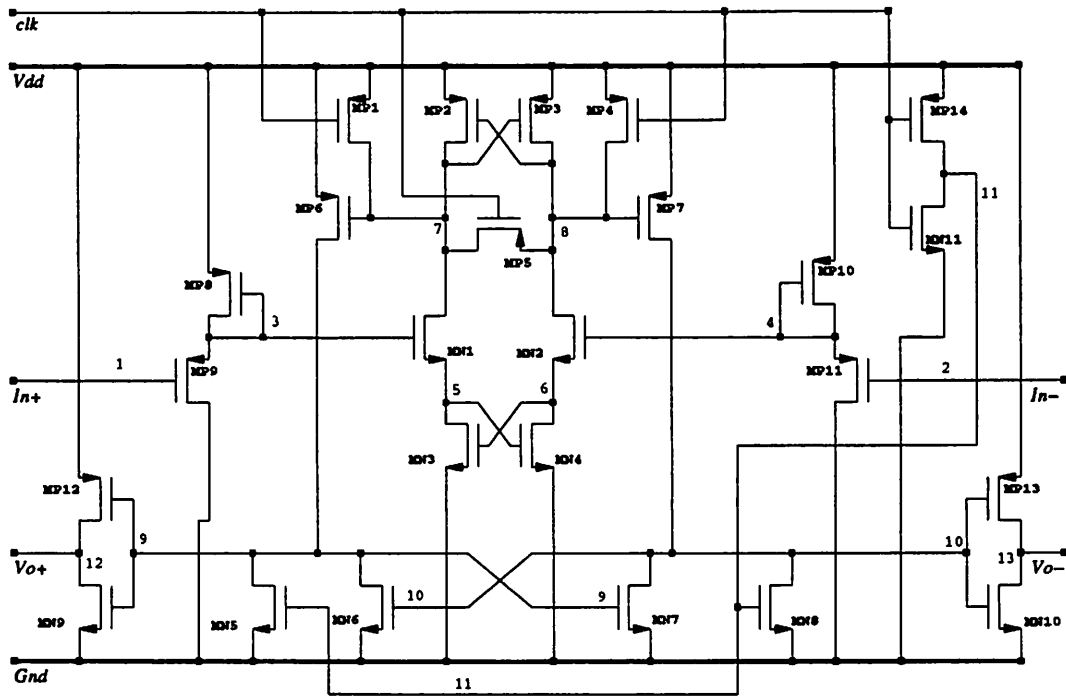


Figure 9.2: Schematic of the clocked comparator FASTCOMP

Capacitive mismatch		
Mismatch	bound	$C^{(max)}$
(C_7, C_8)	27.15fF	100fF
$(C_{7,V_{dd}}, C_{8,V_{dd}})$	27.15fF	100fF
$(C_{1,5}, C_{2,6})$	68.32fF	100fF
Resistive mismatch		
Mismatch	bound	$R^{(max)}$
(R_{S_N3}, R_{S_N4})	0.100 Ω	1 Ω
(R_{S_P2}, R_{S_P3})	7.708 Ω	50 Ω
(R_{S_P8}, R_{S_P10})	42.90 Ω	50 Ω

Table 9.5: FASTCOMP: *bounds on capacitive and resistive mismatch*

Table 9.5 shows some of the most critical parasitic constraints found by PARCAR. As expected, the main contribution to voltage offset is due to parasitic resistances responsible for source degeneration of the input pair. The input source followers (MP10-11 and MP8-9) are less critical than the high-gain pairs (MN3-4, MN1-2 and MP2-3).

Using the same settings described in Tables 4.2 and 9.2 the circuit layout was generated using PUPPY-A, ROAD and SPARCS-A. The complete layout of FASTCOMP is shown in Figure 9.3. Figure 9.4 shows two details of the layout area highlighted in Figure 9.3, respectively with and without parasitic and topological constraint enforcement. In the right-hand side example, large capacitive couplings between critical nets are clearly visible. In particular, a considerable mismatch is present between nets 7 and 8. The capacitance of nets 3,4,9,10 is large, thus slowing down the signal path. These capacitances are much smaller in the example shown in the left-hand side. Notice that relatively large cross-couplings between nets 3,4 and 9 were accepted due to their low criticality. A performance comparison of both the constrained and the unconstrained layouts is summarized in Table 9.6. Table 9.7 lists the CPU times required on a DECstation 5000/240 for each layout phase.

9.1.3 MPH

Figure 9.5 shows the schematic of a micro-power amplifier reported in [263]. This example shows how the layout methodology described in this dissertation fits also tight constraint specifications on relatively large circuits.

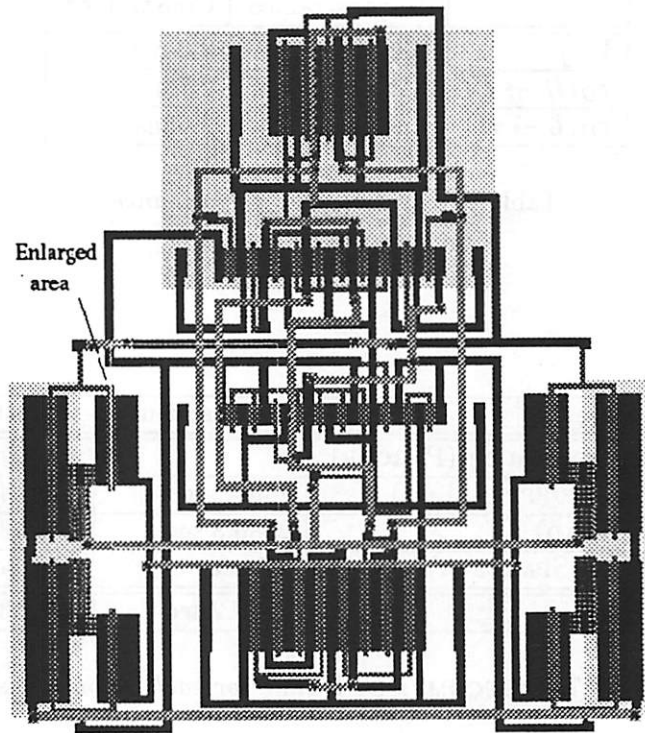


Figure 9.3: Complete layout of FASTCOMP, with enforcement of all analog constraints

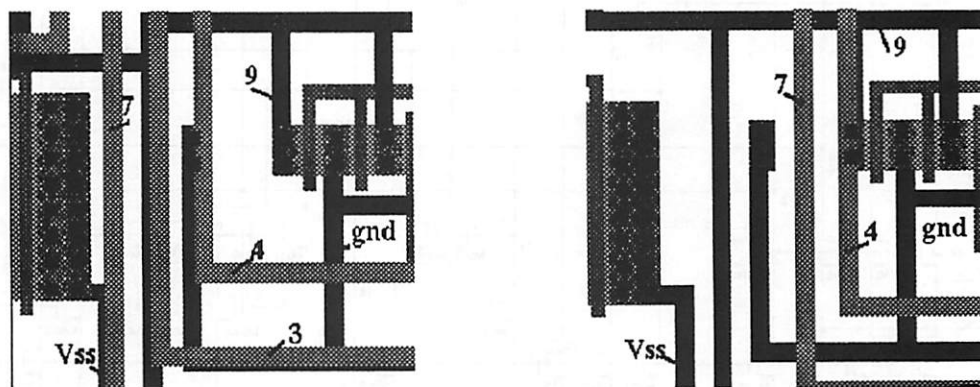


Figure 9.4: Details of the routing of FASTCOMP. Left: no parasitic constraints enforced. Right: all parasitic constraints successfully enforced

	Unconstrained	Constrained
V_{off}	8.0mV	2.0 mV
$\tau_D(H \rightarrow L)$	3.11ns	2.73ns
$\tau_D(L \rightarrow H)$	2.92ns	2.65ns

Table 9.6: FASTCOMP: performance

Layout phase	Unconstrained	Constrained
Constraint generation (PARCAR)	-	10.9 sec
Placement (PUPPY-A-LDO)	246.4 sec	1466.4 sec
Routing (ROAD)	2086.3 sec	2086.3 sec
Compaction (SPARCS-A)	2.5 sec	49.6 sec
TOTAL	2335.2 sec	3602.3 sec

Table 9.7: FASTCOMP: CPU time for each layout phase

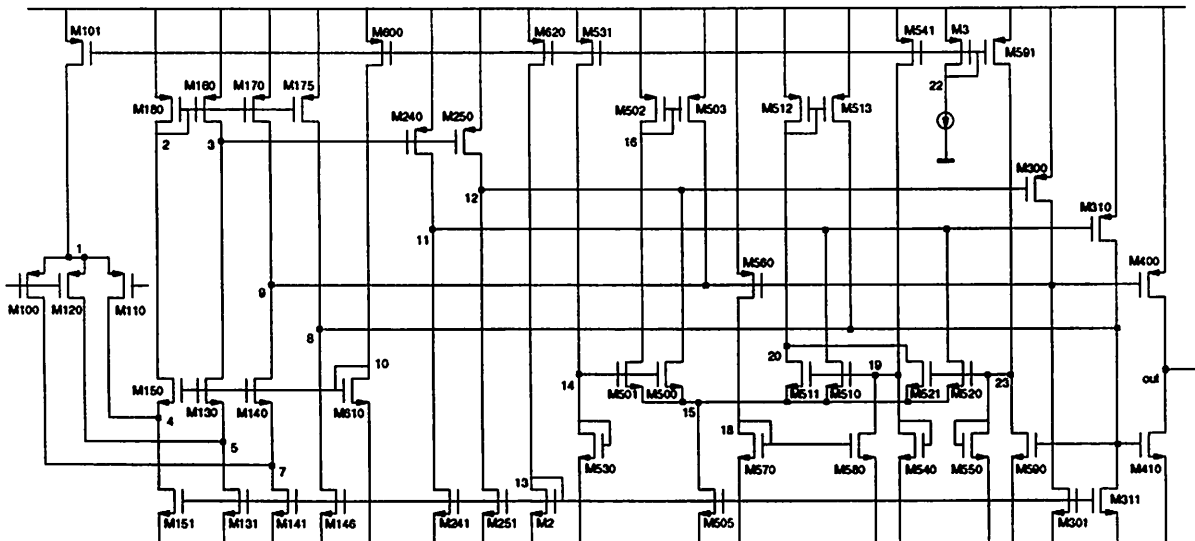


Figure 9.5: Schematic of MPH.

	Constrained	Manual design
V_{dd}	full range	full range
ω_0	7.2MHz	6.0
A_v	136dB	120dB
ϕ_M	56°	63°

Table 9.8: MPH: performance

The nominal performance values for this circuit are the following:

$$V_{dd} = 1.5V$$

$$\omega_0 = 6.0MHz$$

$$A_v = 120dB$$

$$\phi_M = 60^\circ$$

The specifications have been set to

$$|\Delta V_{dd}| \leq 150mV$$

$$\Delta\omega_0 \geq -100kHz$$

$$\Delta A_v \geq -0.1dB$$

$$|\Delta\phi_M| \leq 10^\circ$$

Therefore

$$\mathbf{K} = \begin{bmatrix} V_{dd} \\ -V_{dd} \\ -\omega_0 \\ -A_v \\ \phi_M \\ -\phi_M \end{bmatrix} \quad \mathbf{K}(\mathbf{p}^{(0)}) = \begin{bmatrix} 1.5V \\ -1.5V \\ -6.0MHz \\ -120dB \\ 60^\circ \\ -60^\circ \end{bmatrix} \quad \overline{\Delta\mathbf{K}} = \begin{bmatrix} 150mV \\ 150mV \\ 100kHz \\ 0.1dB \\ 10^\circ \\ 10^\circ \end{bmatrix}$$

The complete layout of MPH is shown in Figure 9.6.

Using the same settings described in Tables 4.2 9.2 the circuit layout was generated using PUPPY-A, ROAD and SPARCS-A. Results for this layout are reported in Table 9.8 in comparison with the data from a hand-made implementation of the same circuit, produced by an experienced designer. Table 9.9 shows the CPU times required by each phase of the design, referred to a DEC Station 5000/240.

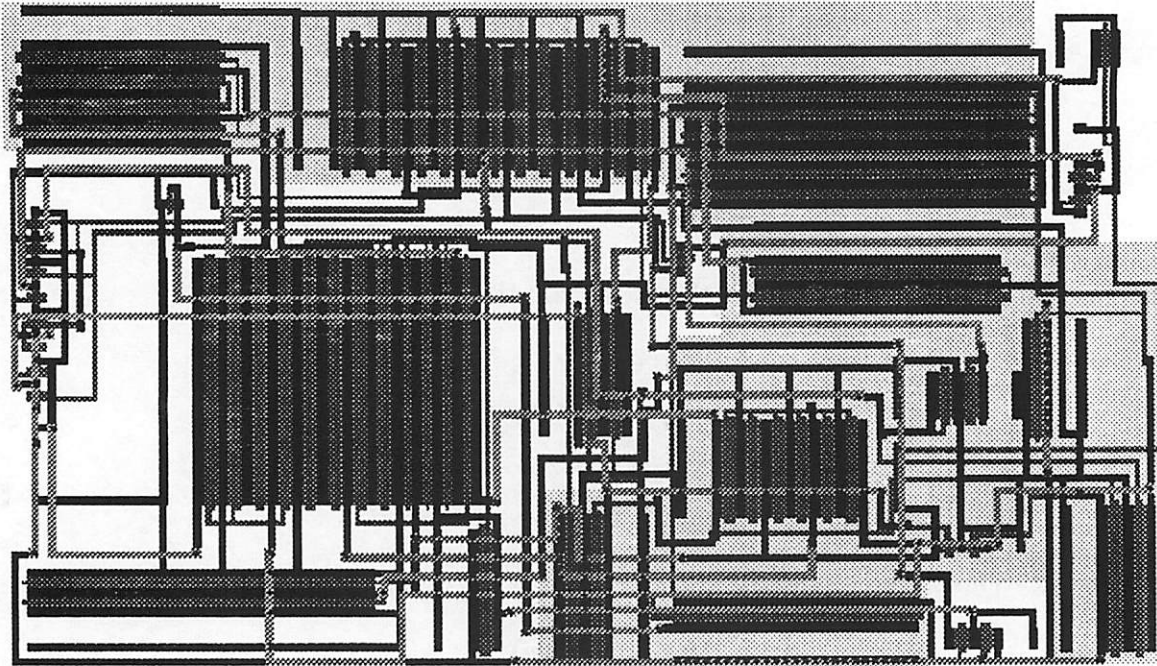


Figure 9.6: Complete layout of MPH, obtained enforcing all analog constraints

Synthesis phase	Constrained	Manual design
Constraint generation (PARCAR)	12,129 sec	-
Placement (PUPPY-A-LDO)	6096.7 sec	-
Routing (ROAD)	6625.0 sec	-
Compaction (SPARCS-A)	10.9 sec	-
Total	6.9 hrs	2 weeks

Table 9.9: MPH: CPU time for each layout phase

Name	DC gain	Offset	B.W.D.	P.M.	S.R.	Delay (H→L/L→H)
FASTCOMP	*	0.0mV	*	*	*	2.42/2.49ns
FCPHIL	35.3dB	2.0mV	28.0MHz	70deg	92V/μs	*
AB	42.2dB	54.0mV	44.6MHz	61.5deg	361V/μs	*
COMPL	*	0.0mV	*	*	*	4.0ns
OPAMP1	70.0dB	0.2mV	5.1MHz	65.0deg	*	*
OTA731	79.5dB	*	1.2MHz	62.5deg	*	*
NEWOTA	25.6dB	*	6.04MHz	65.0deg	*	*
MPH	120dB	*	6.0MHz	60.0deg	*	*

Table 9.10: Nominal performance of benchmark circuits

9.1.4 Other CMOS Benchmarks

In this section the results of extensive use of our performance-driven methodology for custom IC design are presented. The impact of our methodology on successful designs with tight specifications is also discussed. The techniques described in this dissertation have been applied to a representative sample of analog circuits, often used in larger systems. Table 9.10 shows the nominal performance of the benchmarks, *B.W.D.*, *P.M.* and *S.R.* are abbreviations for *unity gain bandwidth*, *phase margin* and *slew rate*, respectively. Parasitic and topological constraints were generated for the above benchmarks as described in chapter 3.

The Results are shown in Table 9.11. The CPU times are referred to a DEC Station 5000/240. *P*, *S* and *M* denote the number of constraints on *parasitics*, *symmetry* and *matching*, respectively. The benchmarks were assembled enforcing topological and parasitic constraints. For a comparison, experiments were conducted on the same circuits by eliminating all constraints and measuring the resulting performance.

As expected, our performance-driven methodology always ensures that all performance specifications be met. On the contrary, when no constraints are considered, large performance degradations occur. Hence, a significant portion of all specifications are violated. Table 9.12 shows the effectiveness of our methodology in enforcing performance specifications. Only in one instance the layout violated one specification. The reason for this is due to the extreme tightness of the specification and of the strong non-linearity of the circuit. Hence, the sensitivity-based performance model becomes inaccurate. Table 9.12 reports all CPU times for the synthesis of each layout as well as the percentage value of the

Name	# Devices	P	S	M	CPU (sec)
FASTCOMP	27	70	27	6	10.9
FCPHIL	26	594	15	14	2151
AB	15	211	14	14	2.6
COMPL	26	24	15	8	0.1
OPAMP1	13	79	5	4	11.9
OTA731	22	383	11	20	198.4
NEWOTA	23	83	23	8	*
MPH	28	340	16	25	12,129

Table 9.11: *Constraint generation for the given benchmark circuits*

Name	# Specs	Tightness	Met (P-D)	Met (non-P-D)	CPU (P-D)
FASTCOMP	3	10%	3	1	3602.3 sec
FCPHIL	4	1.5%	4	2	1561 sec
AB	4	0.37%	4	3	1552 sec
COMPL	2	60%	2	1	450.2 sec
OPAMP1	4	0.8%	4	2	646 sec
OTA731	3	0.16%	2	*	645 sec
NEWOTA	3	0.6%	3	2	1674 sec
MPH	3	3%	3	1	24861.6 sec

Table 9.12: *Measure of success of the performance-driven methodology*

tightest performance specification with respect to its nominal value. The number of specifications met (*met*) for the performance-driven (*P-D*) and for the non performance-driven (*P-D*) layout synthesis is also reported.

9.2 Mixed-Signal Benchmark Library

In this section two major mixed-signal systems are discussed which were designed and physically realized using our constraint-driven methodology for physical assembly. The first circuit is a display driver for color monitors (RAMDAC). The second is a $\Sigma - \Delta$ CONVERTER. Much of the work in these projects is due to Iasson Vassiliou, Alper Demir, Henry Chang and Paolo Miliozzi, whom I sincerely acknowledge.

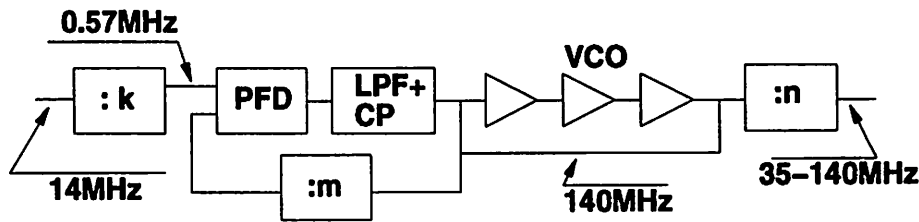


Figure 9.7: PLL schematic

9.2.1 The RAMDAC System

System Description

The RAMDAC system includes three D/A converters, a Phase Lock Loop (PLL), and digital control logic. The converters were generated using a dedicated silicon compiler based on the top-down, constraint-driven design methodology [264]. The synthesis of the entire system is described in full details in [265]. Let us now turn our attention to the PLL circuit depicted in Figure 9.7. The PLL consists of a digital section, i.e. three divide-by- n modules and a phase-frequency detector (PFD); and a number of analog components, i.e. an analog low-pass filter (LPF) and a charge pump (CP). The interface between the analog and the digital sections is represented by the voltage-controlled oscillator (VCO), which generates a digital output at a frequency proportional to the input voltage.

Typical frequencies of operation are shown in the various branches of the circuit in Figure 9.7. The architecture of the VCO, the replica bias and the basic cell, derived from [266], are depicted in Figure 9.8a,b, and c respectively. The PFD, similar to the one used in [267], is depicted in Figure 9.9. The LPF, also derived from [267], is shown in Figure 9.10a. The output frequency of the PLL F_{out} is expressed in terms of the various dividing ratios and the reference frequency F_{ref} as

$$F_{out} = F_{ref} \cdot \frac{m}{nk},$$

where n , m and k are the division factors of each divider. Figure 9.11a,b shows the schematic diagram of the programmable dividers and of the flip-flop.

Schematic and Module Generation for the PLL

An extensive survey on the effects of substrate noise in the various components of the system revealed the extreme vulnerability of the VCO to switching noise at frequencies

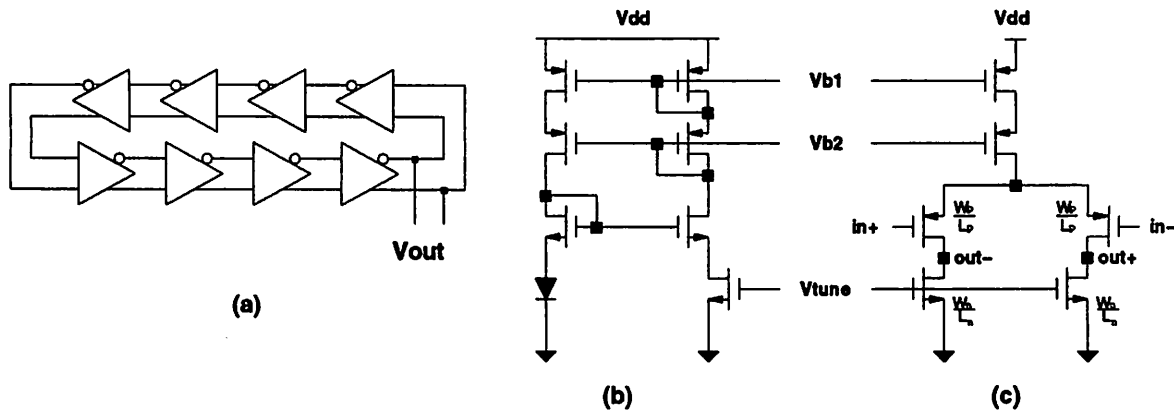


Figure 9.8: VCO block diagram and schematic of one delay cell

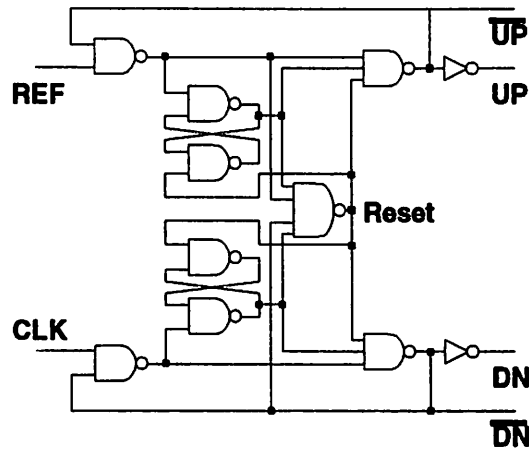


Figure 9.9: PFD schematic

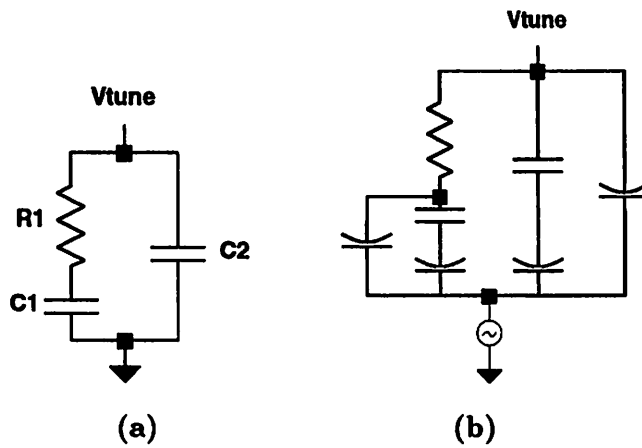


Figure 9.10: LPF schematic: (a) no substrate coupling; (b) with substrate coupling

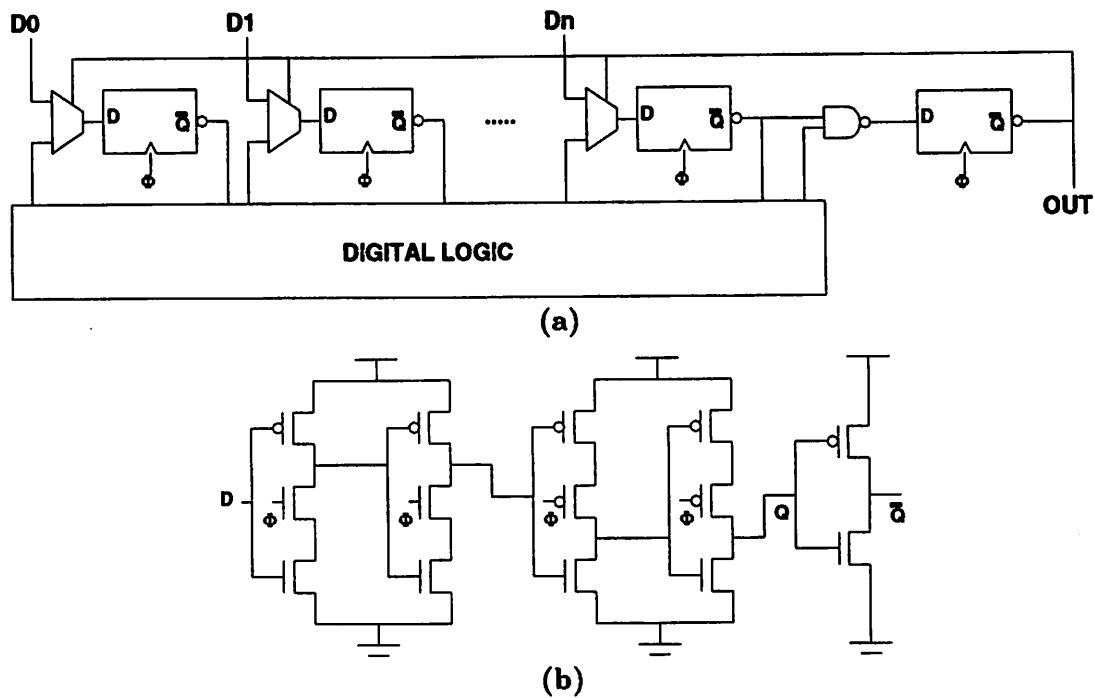


Figure 9.11: Programmable divider: (a) block diagram; (b) single-phase flip-flop

close to its operating frequency. The frequency-to-voltage characteristic can be significantly affected by process, temperature variations and layout parasitics. Due to the impossibility of controlling process and temperature variations, the PLL was optimized so as to keep performance degradation due to these non-idealities within pre-defined tolerances.

This task was accomplished using a top-down, constraint-driven approach to the design of the PLL. An exhaustive discussion of the methodology goes beyond the scopes of this dissertation. Hereafter is an outline of the main steps of the approach. First, the system is organized hierarchically in sub-partitions which represent the various levels of abstractions of the circuit. Then, each abstraction level, from the top to the bottom level of the hierarchy, is represented by a certain model which captures the behavior of the underlying system components. High-level specifications are propagated from level to level down the design hierarchy, using optimization in combination with behavioral simulation. A set of constraints on the parameters of each model is therefore generated for each abstraction model until the bottom of the hierarchy, i.e. the physical representation of the circuit, is reached. The constraints on the physical implementation are then used by the constraint-driven physical assembly.

Measure	Conditions			Specs
	PLL input frequency	Division Factor n	VCO frequency	
Stability	0.56 MHz	100	56 MHz	Yes
		250	140 MHz	Yes
Jitter $\left(\frac{\Delta T}{T}\right)$	0.56 MHz	250	140 MHz	≤ 0.007
Phase Margin	-	-	-	$\geq 45^\circ$

Table 9.13: PLL specifications

Performance	Nominal	Max. Variation
K_0	40MHz/V	4MHz/V
F_0	100MHz	10MHz
$\frac{\Delta T}{T}_{rms}$	0	0.05%
$\frac{\Delta T}{T}_{pp}$	0	1%

Table 9.14: Parameter constraints obtained for the VCO by behavioral optimization of the PLL

In the remainder of the section we will discuss issues related to the implementation of substrate-related constraints during the physical assembly of the VCO and, in general, of the entire PLL. The high-level specifications for the whole PLL are summarized in Table 9.13. The jitter $\frac{\Delta T}{T}$ is defined as the ratio between the variation from nominal of oscillation period ΔT and period T . Due to its time-variance, $\frac{\Delta T}{T}$ is generally measured in terms of its peak-to-peak or RMS value. The jitter performance of the PLL is mostly affected by the VCO jitter. The jitter in the VCO is in turn caused by thermal noise and substrate noise. Let us model the output frequency of the VCO F_{VCO} as follows

$$F_{VCO} = F_0 + K_0 \cdot \Delta V, \quad (9.1)$$

where F_0 is the VCO central frequency of operation, K_0 the frequency-to-voltage gain, and ΔV the deviation from nominal of the applied voltage in the control node.

After behavioral optimization, a set of constraints on the VCO model parameters of equation (9.1) was found. Table 9.14 lists the bounds found by the optimizer. Using the techniques described in chapters 3 and 8, one can derive a set of constraints on parasitics and on the maximum allowed switching noise at all critical locations of the VCO and of

Parasitic	Constraint	Extracted value
C_{out+}	$12.34 fF$	$11.60 fF$
C_{out-}	$12.34 fF$	$11.60 fF$
C_x	$15.84 fF$	$1.12 fF$
$V_0 _{VCO}$	$110 mV$	$109 mV$
$V_0 _{LPF}$	-	-

Table 9.15: Constraints obtained by the sensitivity analysis

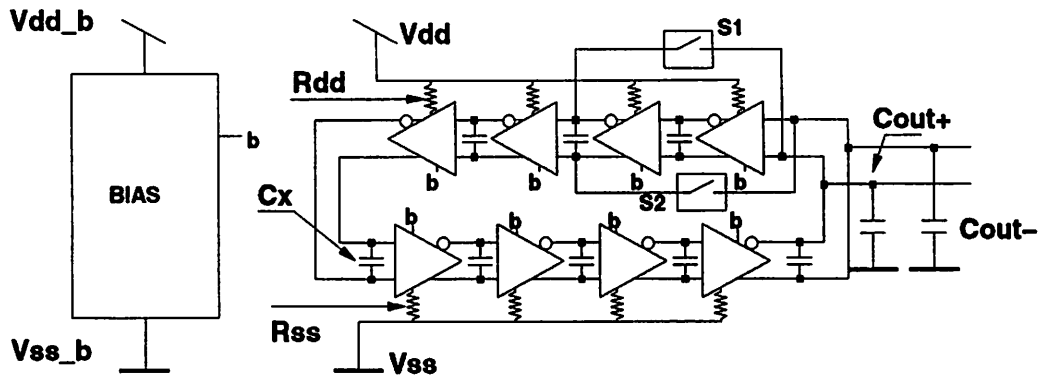
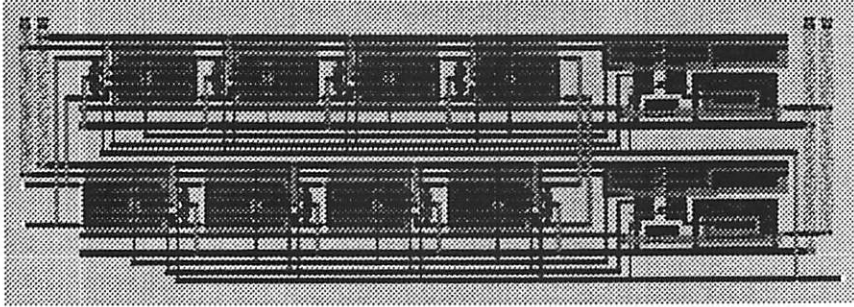


Figure 9.12: VCO architecture generated by VCOGEN

the LPF. See Table 9.15. Capacitances C_{out+} , C_{out-} and C_x are identified in Figure 9.12. Parameters $V_0|_{VCO}$ and $V_0|_{LF}$ relate to the maximum admissible peak-to-peak voltage at each critical node of the VCO and of the LPF, to insure that all performance specifications be met.

The constraints on parasitics were taken into account during the automated generation of the VCO using the module generator VCOGEN. VCOGEN is a fully parametrizable multi-architecture fixed-floorplan ring-oscillator generator, for CMOS design style in both substrate types. The module generator can enforce capacitive and resistive constraints on all the nets present in the design. The basic architecture generated by VCOGEN is illustrated in Figure 9.12. Capacitive and resistive parasitic constraints are enforced during the physical assembly by estimating the needed geometries *a priori* and by selecting the wiring combination which ensures satisfaction of the constraints. The final layout structures are selected from a set of alternatives constructed by enumerating all feasible realizations. This method is carried out efficiently due to the low number of degrees of freedom left to the

Figure 9.13: *Layout of eight-stage VCO*

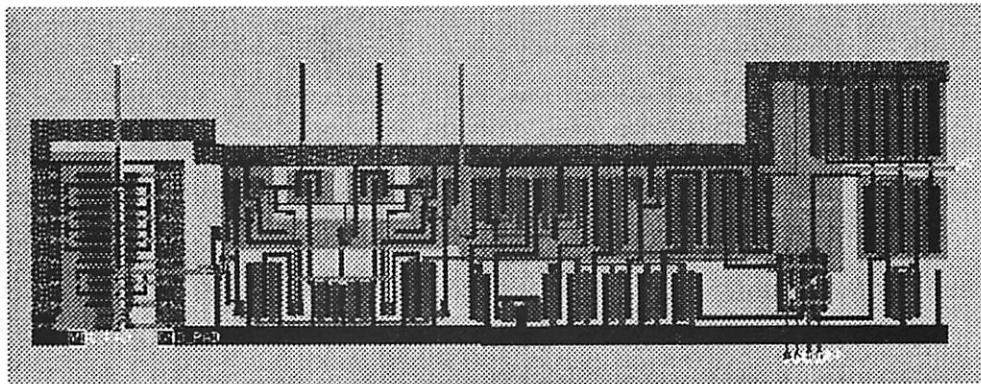
Operation	CPU time (sec)
circuit optimization	1028 †
substrate sensitivity	2545 †
interconnect sensitivity	3256 †
constraint generation	115
layout generation	43

Table 9.16: *CPU times for the design and module generation obtained on a DEC Station 5000/125 and on a DEC AlphaServer 2100 5/250 (†)*

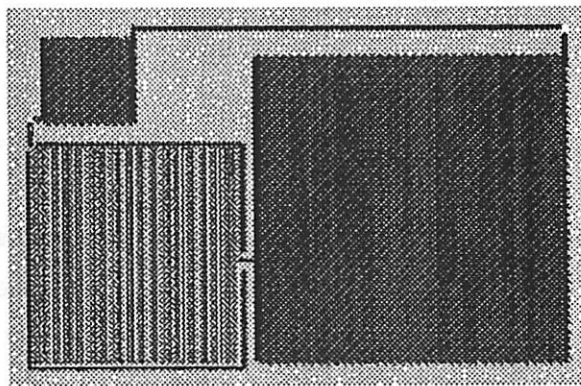
generator.

The parasitics controlled during physical assembly are depicted in Figure 9.12, while the number and position of the switches is user-defined (e.g. $S1$ and $S2$). The number of stages is also parametrized based of the results of the optimization. The replica-bias circuit, with separate supply is also generated by `VCOGEN`. See Figure 9.12. An example of the output of the module generator for a eight-stage VCO implementing the constraints of Table 9.15 is shown in Figure 9.13. The CPU times for the calculation of sensitivities, the generation of all constraints and the layout generation for the VCO are shown in Table 9.16.

A similar method was used for the design of the LPF. The LPF and the CP are shown in Figure 9.14. The PFD and the dividers were generated using standard tools for digital layout assembly.



(a)



(b)

Figure 9.14: (a) Charge pump (CP); (b) Low-pass filter (LPF)

Physical Design of the PLL

After the module generation step, the various components of the PLL were placed and routed along with the other circuits of the RAMDAC. The placement was carried out using PUPPY-A, which accounted for capacitive and resistive parasitics as well as substrate injection effects.

In the circuit there exist three major switching noise injectors, corresponding to the dividers. In order to accurately verify if the constraints on the maximum admissible noise voltage at critical nodes are violated, an accurate model is needed for the signal injected by each divider. For this reason each divider was simulated extensively using advanced models for substrate injection at the device level. In addition, the current injected via capacitive coupling by power and ground busses, connected to the supply through inductive bonding wiring, was carefully modeled. Finally, a compact macroscopic model for the cumulative switching noise of all the dividers was generated using the method outlined in chapter 4. Figure 9.15 shows the cumulative injected current and the approximate model generated for the programmable divider used in this work.

If we assume that the substrate shows a purely resistive behavior, the calculation of the peak-to-peak voltage at each node of the surface can be carried out by performing a simple DC analysis on the positive and negative peak values of the current of the injector. See Figure 9.16. The placement was performed using the techniques described in section 4.6. The original specifications imposed on the PLL (Table 9.13) were translated in a set of constraints on the maximum admissible noise voltage at critical nodes of the VCO, CP and LPF, the only critical components in our design (Table 9.15). The low-level constraints were used in the cost function of the annealing in terms of constraint violations as outlined in section 4.6.

Our efficient substrate transport analysis was used on the entire RAMDAC chip for the placement. Figure 9.17 shows the estimated values of switching noise voltage at each location in the chip during the unfolding of the annealing. Notice how the annealing attempts to reduce the switching noise amplitude at critical receptors locations in the VCO, CP and LPF¹.

Plot 9.18 shows various statistics measured during the unfolding of the annealing. Curve (a) represents the relative error on the substrate noise at receptors when the heuristic

¹For clarity, during the annealing the location of these components have been fixed at the bottom center of the chip.

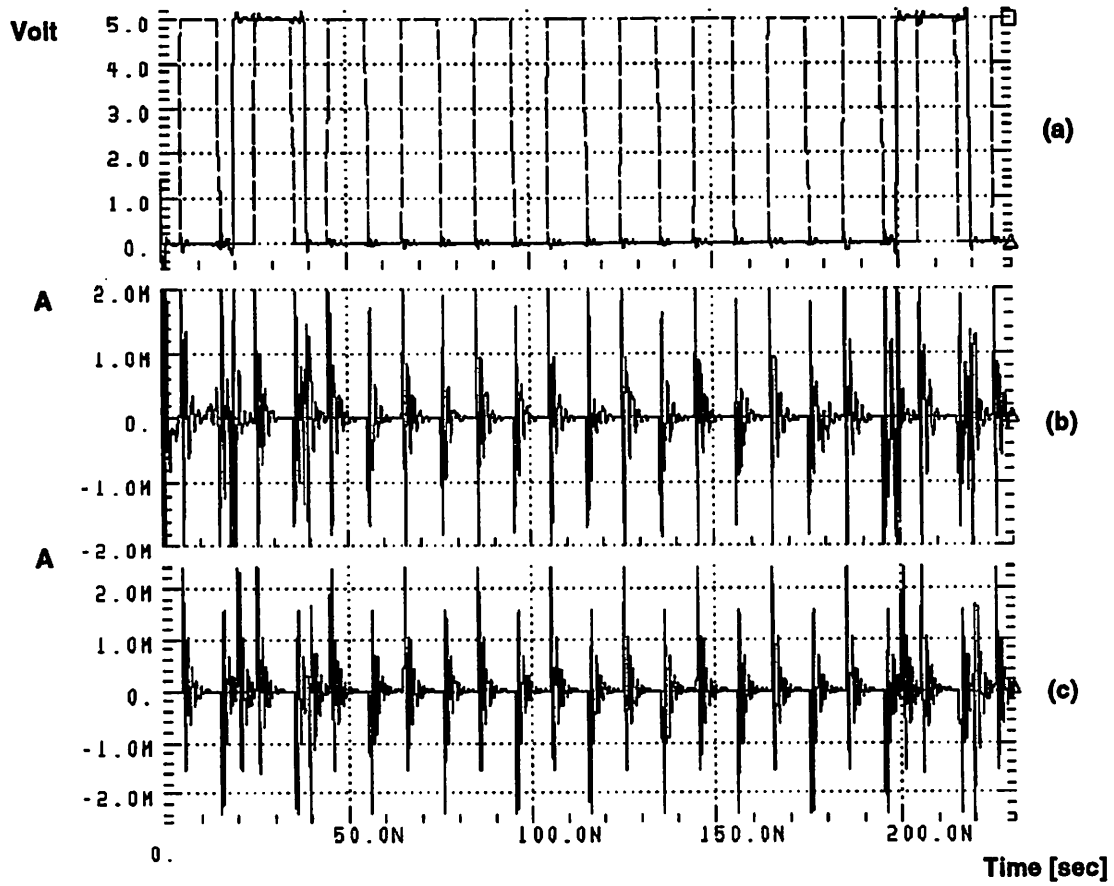


Figure 9.15: (a) Output signal of divider; (b) Injected current; (c) Model for substrate injection

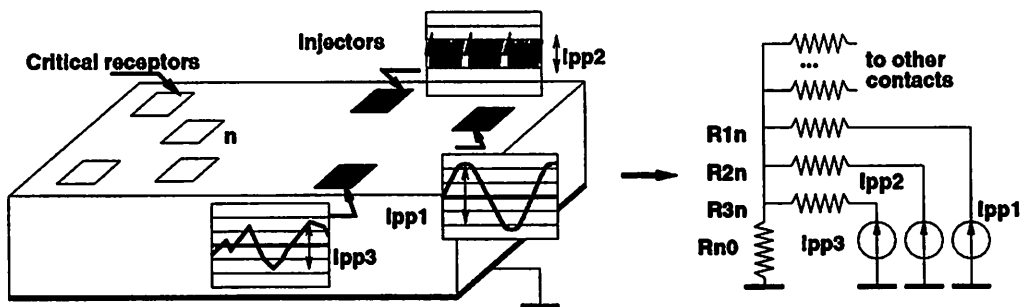


Figure 9.16: Evaluation of peak-to-peak switching noise at the receptor site

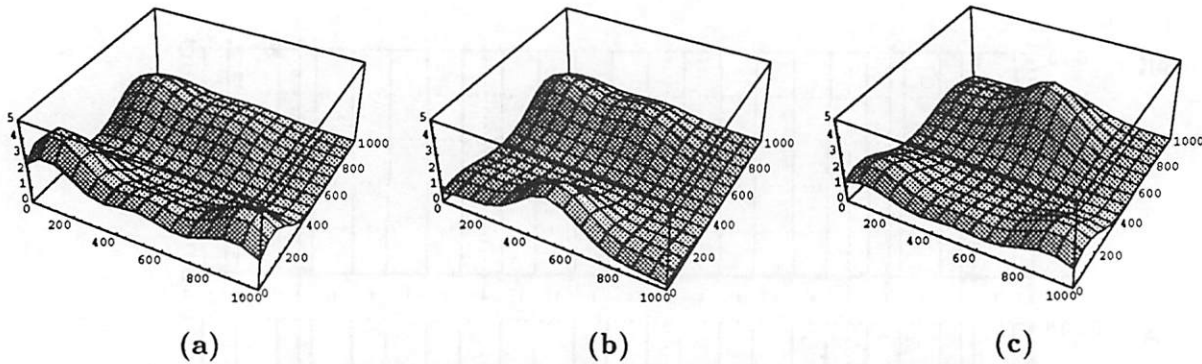


Figure 9.17: Estimated level of switching noise signal amplitude as a result of the cumulative injection of the dividers during the annealing: (a) high temperature; (b) medium temperature; (c) low temperature

Placement type	CPU time (sec)	Area (λ)	Est. Jitter
Manual	-	5637 x 6481	-
Parasitic constr.	406.74	6765 x 6528	0.1
Substrate+parasitics	885.20	7322 x 7716	0.005

Table 9.17: Placement statistics obtained on a DEC AlphaServer 2100 5/250

of Figure 4.36, based on combined use of Sherman-Morrison & gradient-based methods, is used. Curve (b) shows the error occurred using the gradient-based method only. All relative errors are obtained by comparison with an exact method, e.g. Sherman-Morrison update. Curves (c) and (d) show the behavior of the constraint violations with and without using the proposed substrate injection control techniques.

Figure 9.19 shows the final placement performed using PUPPY-A. As expected the divider n was placed at a large distance from the sensitive components of the PLL, namely the CP, VCO and LPF. On the contrary, The sensitivity of these components with respect to the switching noise produced by divider k is small, hence it can be placed consequently. For divider m the placer had to perform a trade-off between the strength of the switching noise received by it and the parasitics introduced when large interconnect capacitances are introduced.

The statistics of the placement process are summarized in Table 9.17.

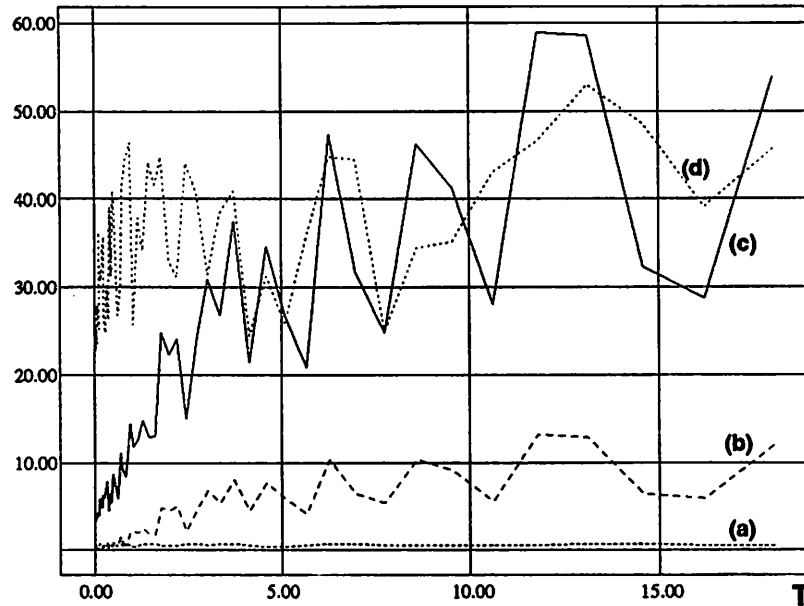


Figure 9.18: *Error in substrate injection estimation using: (a) combined heuristic; (b) gradient-based method only. Evolution of total substrate violations using: (c) combined heuristic; (d) no substrate control*

Trend Analysis

The complete chip is shown in Figure 9.20 after routing and compaction, performed with MOSAICO and SPARCS-A [268]. For the PLL, all the potential sources of switching noise are localized in the dividers, while the receptors are in the VCO, CP and LPF. Injection occurs by impact ionization through the active areas of NMOS devices (in a N-well processes) and by capacitive coupling through junctions and interconnect. Receptors are in the active areas of sensitive devices and supply lines. Table 9.18 lists the main sources and receptors of noise in the various components of the design. Using (8.34) and the sensitivity information $\partial K/\partial Y_{ij}$, all performance degradations ΔK_i can be efficiently computed for the PLL, due to small changes in the design and/or in the technology. Suppose one were interested in deriving the trend of the jitter performance if a new lightly doped substrate were to be used instead of the low-resistivity substrate for which the circuit was designed. Plot 9.21a shows the values of a sub-set of substrate matrix \mathbf{R} as a function of the doping on the epitaxial layer (Figure 8.6). Plot 9.21b shows the values of the sensitivities of one entry of \mathbf{R} at various nominal doping levels (t_1, \dots, t_3).

Suppose now we were looking at the effects of contact depth (c in Figure C.1).

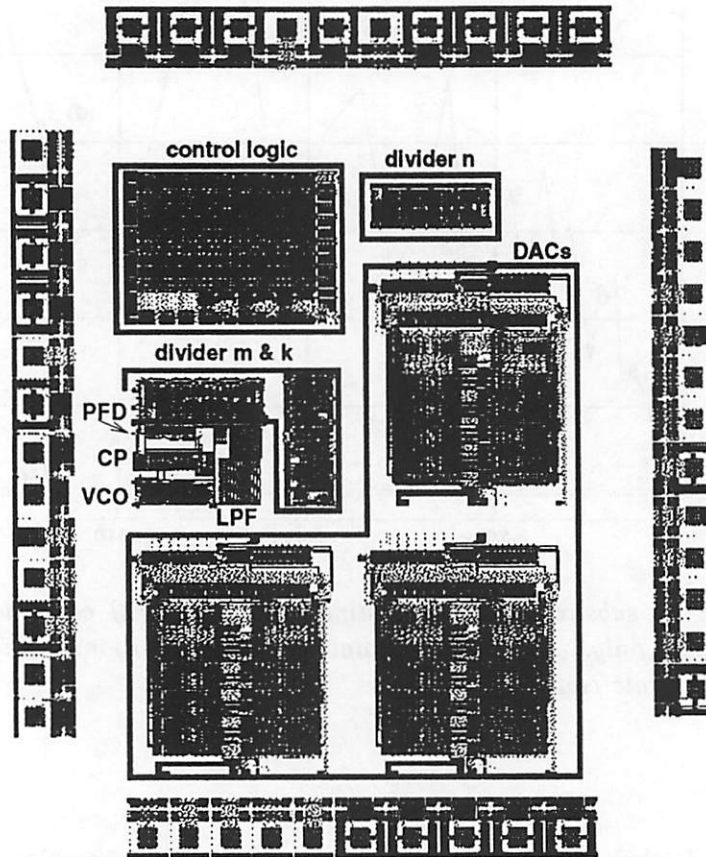


Figure 9.19: Placed PLL within the RAMDAC

Component	Number of receptors	Number of injectors
divider	-	152
PFM	-	23
VCO	85	-
LPF	5	-
CP	46	-
Total	136	175

Table 9.18: Noise injector and receptor statistics in the components of the PLL

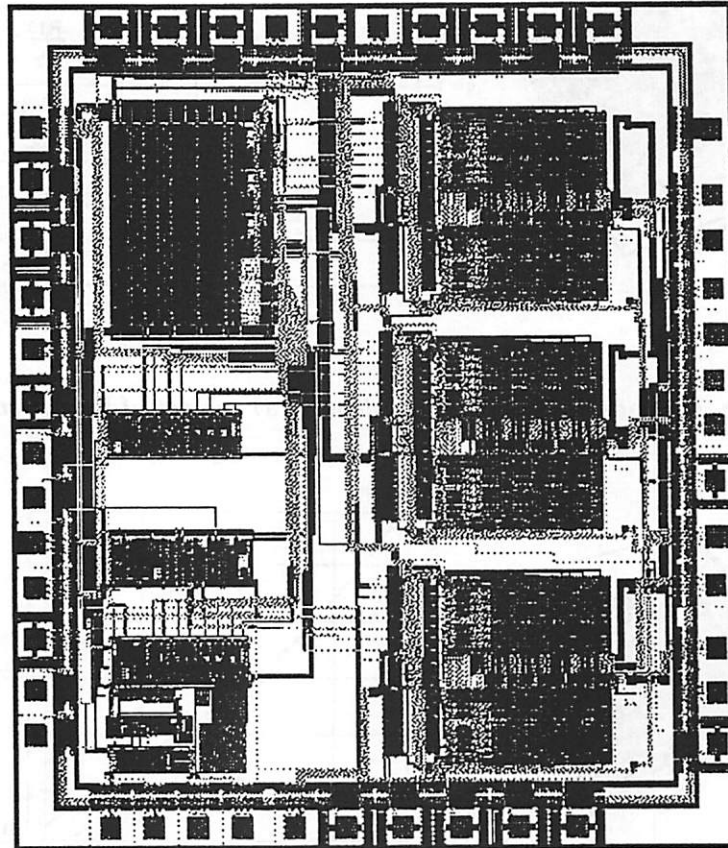


Figure 9.20: *Placed and routed PLL within the RAMDAC*

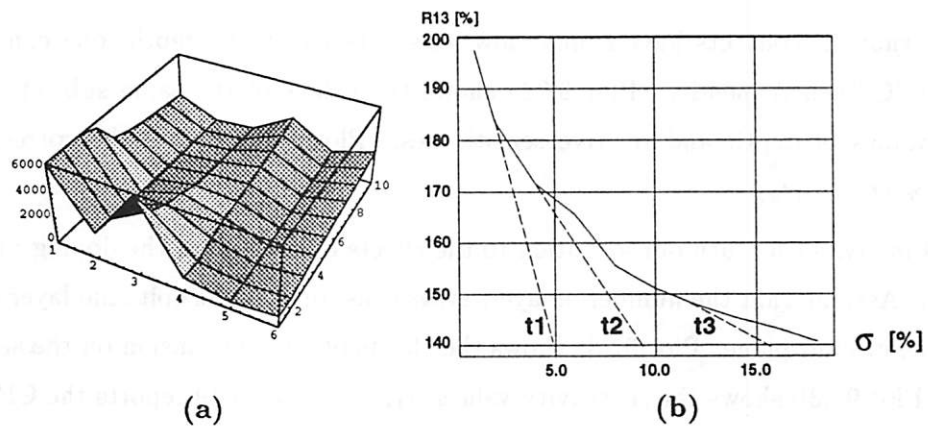


Figure 9.21: *Dependence from doping levels: (a) sub-set of R; (b) sensitivity*

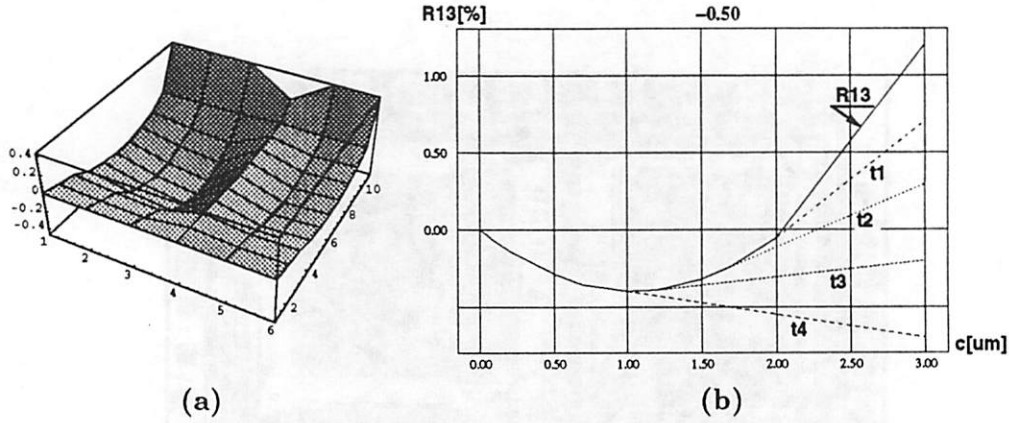


Figure 9.22: Dependence from contact depth: (a) sub-set of \mathbf{R} ; (b) sensitivity

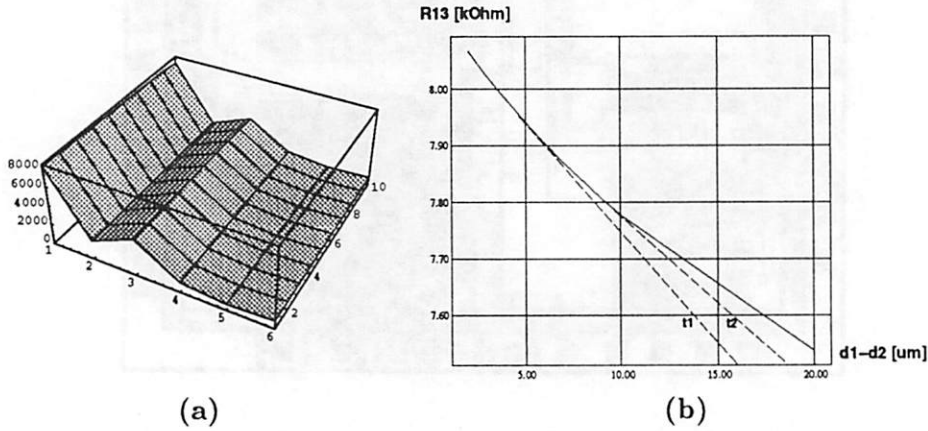


Figure 9.23: Dependence from doping profiles: (a) sub-set of \mathbf{R} ; (b) sensitivity

Assuming that all contacts have similar low-resistivity substrate depth, one can use the expression (C.2) in Appendix. Plot 9.22a shows the values of the same sub-set of \mathbf{R} for different values of depth and relative sensitivities. Plot 9.22b shows the corresponding sensitivities (t_1, \dots, t_4).

Finally, let us turn our attention to the effects of changes in the doping profiles in Figure 8.6. Assume that the number of layers stays constant but the epitaxial layer expands towards the ground-plane. Plot 9.23a shows the effects of layer expansion on the same sub-set of \mathbf{R} . Plot 9.23b shows the sensitivity values (t_1, t_2). Table 9.19 reports the CPU times for the sensitivity analysis performed in the various experiments and the estimated trend of jitter performance degradation computed using (8.34).

Experiment	CPU times (sec)	Jitter trend
Epitaxy doping	3038.88	1.34
Contact depth	2858.83	0.95
Profile change	4005.46	0.55

Table 9.19: CPU times on a DEC AlphaServer 2100 5/250 for the trend analysis for the proposed experiments on the PLL with 311 noise sources / receptors. The CPU times include DCT, parameter and sensitivity computation. For the calculation of 311 contacts the inversion of matrix \mathbf{P} was performed in 1525.0 seconds.

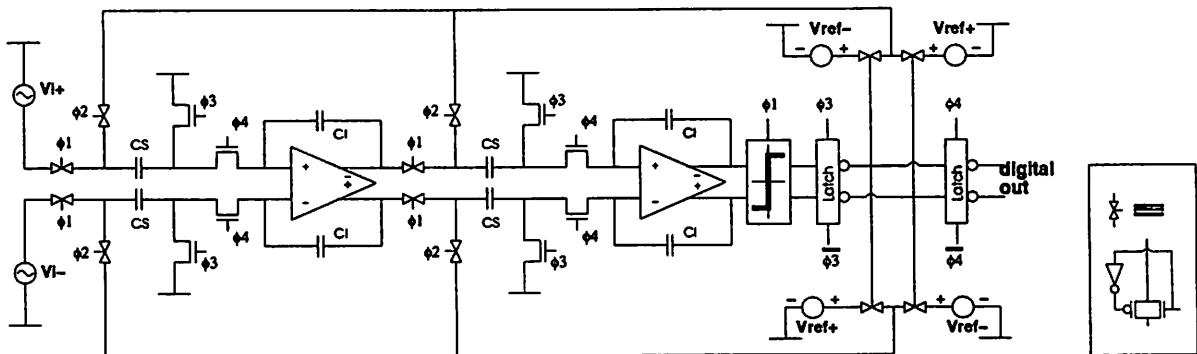


Figure 9.24: $\Sigma - \Delta$ CONVERTER architecture

9.2.2 The $\Sigma - \Delta$ CONVERTER System

System Description

The system is described as a second order $\Sigma - \Delta$ CONVERTER. The basic architecture of the $\Sigma - \Delta$ CONVERTER is shown in Figure 9.24. The system consists of two switched capacitor integrators, a comparator, and a 1 bit D/A. The integrators perform the noise-shaping function on the input signal, while the D/A in the feedback loop drives the output bit to "follow" the input signal thus giving the proper A/D conversion. The performance specifications for our example were a minimum signal-to-noise (SNR) ratio of 74 dB and a Nyquist input frequency of 250 kHz. Table 9.20 summarizes all design specifications for the chip. The technology used was the HP CMOS26B (0.8 μm minimum gate length).

Type	Specifications	Value
Performance	Minimum Signal-to-Noise (SNR) ratio	74 dB
	Nyquist Frequency (f_x)	250 kHz
Operation	Supply voltage	5 V
Technology	Design rules	scmos
	HSPICE parameters	HP CMOS26B

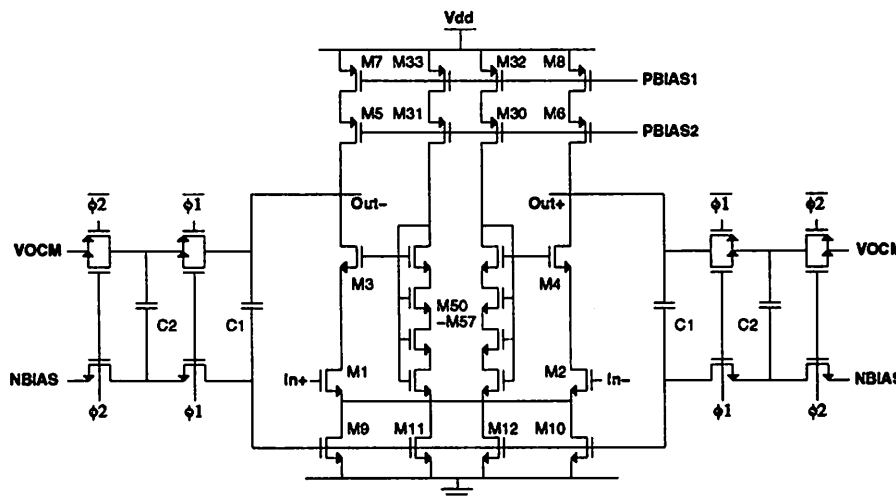
Table 9.20: $\Sigma - \Delta$ CONVERTER design specifications

Figure 9.25: Schematic of the OTA

Schematic and Module Generation for the $\Sigma - \Delta$ CONVERTER

Hereafter, a description is presented of all the components of the $\Sigma - \Delta$ CONVERTER shown in Figure 9.24. The schematic of the switched capacitor integrator is shown in Figure 9.24. A bottom plate based sampling scheme is used to minimize charge injection and for level shifting purposes. The core of the integrator circuit consists of a telescopic or unfolded cascode operational transconductance amplifier (OTA), shown in Figure 9.25. Capacitors, C_1 and C_2 form the dynamic common mode feedback (CMFB) circuitry [269]. The switches and capacitors set the output common mode voltage as well as gate voltage for M_9 through M_{12} . A replica bias circuit, shown in Figure 9.26 was used to provide reference and bias voltages for OTA and integrator. All gate voltages for the tail current devices, the current sources in the OTA, and common mode voltage are generated by the bias.

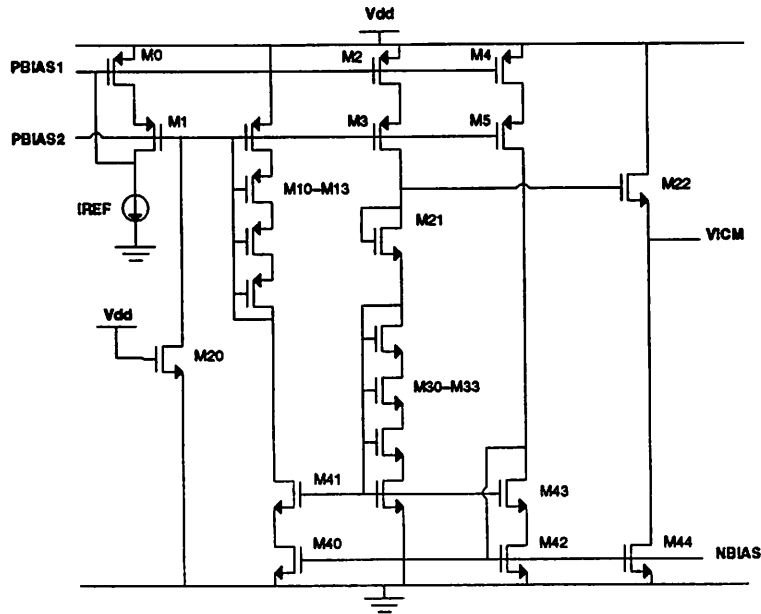


Figure 9.26: *Schematic of the bias circuitry*

Using a top-down, constraint-driven methodology, in combination with the behavioral simulator MIDAS [270], the original specifications in Table 9.20 were translated into specifications for each component of the design. The component specifications in turn, were the basis for the calculation of the sizes of each device [271].

For the modules in the integrator a set of design constraints were derived from the high level synthesis and verified using simulation on each module. Table 9.21 lists the complete set of constraints for the integrator and the actual performance measures after the design of each integrator component. The mapping procedure was performed in less than 1 CPU second.

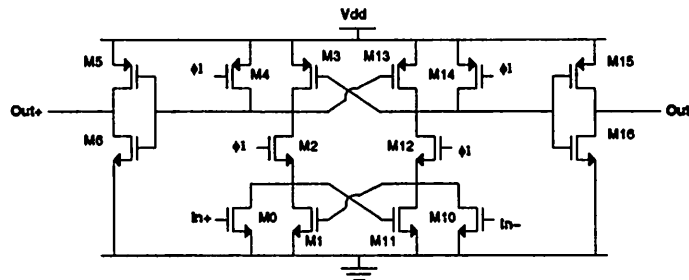
Similarly, the comparator was designed using optimization. Figure 9.27 shows the schematic of the comparator and the set of constraints is specified in Table 9.22.

The only components that have to be sized for the 1 bit D/A converter are the switches. These are selected to be the same as the integrator.

The digital blocks in this circuit, the clock generator and the two latches which follow the comparators, were generated using standard digital circuit compilers. The circuit used for the clock generator is shown in Figure 9.28. Given a single phase clock at the desired frequency at node "CLK," the appropriate phases are produced by this circuit. The "1x"

Parameter	Constraint	Simulated value
OTA offset voltage	25 mV	20 mV
OTA output range	2.0	2.0
OTA thermal noise	$20.0 \mu V_{rms}$	$1.2 \mu V_{rms}$
OTA $1/f$ noise	$1.05 mV_{rms}$	$9.9 \mu V_{rms}$
OTA open loop gain	250 (V/V)	1336 (V/V)
integrator time constant τ	1.5 ns	0.63 ns
integrator slew-rate	$1045 V/\mu s$	$1045 V/\mu s$
integrator kT/C noise	$20 \mu V_{rms}$	$20 \mu V_{rms}$
CMFB τ	2.0 ns	0.98 ns

Table 9.21: Design constraints for the integrator

Figure 9.27: Schematic of the comparator used in the $\Sigma - \Delta$ CONVERTER

Parameter	Value
comparator offset voltage	100 mV
comparator hysteresis	100 mV
comparator RMS noise	$9.35 mV_{rms}$

Table 9.22: Design constraints for the comparator

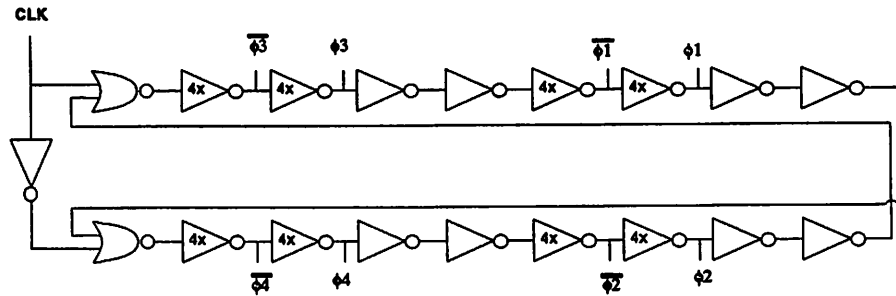


Figure 9.28: Schematic of clock generator

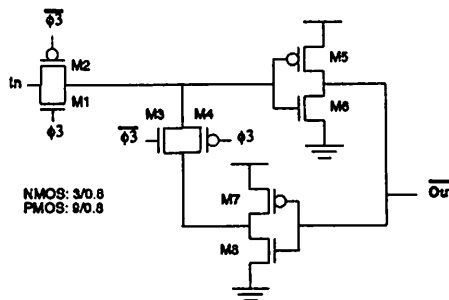


Figure 9.29: Schematic of latch

sized inverters provide the delays between ϕ_3 and ϕ_1 and between ϕ_4 and ϕ_2 . It also provides for the non-overlap time between ϕ_1 and ϕ_2 . The latch, a basic eight transistor device, is shown in Figure 9.29.

Physical Design of the $\Sigma - \Delta$ CONVERTER

Most of the layout was automatically synthesized. Issues involving why manual layout had to be used for some of the circuits will also be discussed. The initial schematic was entered in the form of a SPICE deck. The constraints on critical capacitive and resistive parasitics and mismatches were identified and generated using PARCAR. All topological constraints, symmetry and matching, were computed using MKSYM and entered in the data-base which also contains the geometrical and connectivity information about the circuit (chapter 3). The stack generator MKSTACK was then employed for the mapping of all devices with the appropriate module alternatives (chapter 4). Simultaneously, all capacitances of the integrator were created by dedicated routines available in the OCTTOOLS. The stacks were then selected and placed by PUPPY-A. See Figure 9.30.

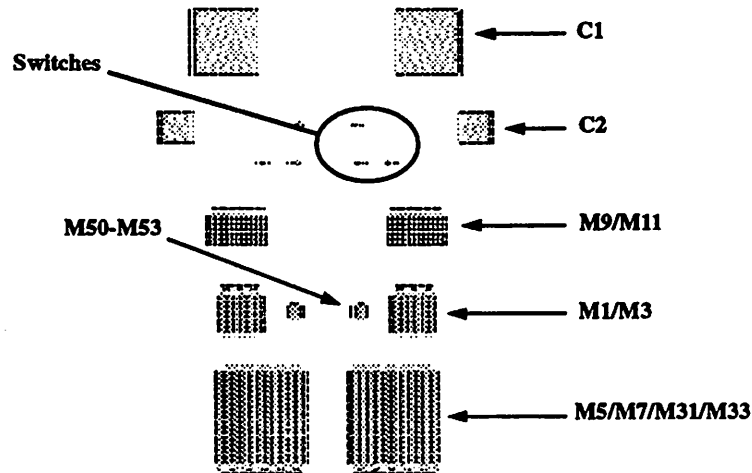


Figure 9.30: Placement by PUPPY-A of the OTA (Courtesy of H. Chang and E. Felt)

The cell was finally routed and compacted by ROAD [201] and SPARCS-A [69], respectively. Figure 9.31 shows the placed, routed and compacted result for the OTA.

The level of automation possible depends greatly on the level of organization in the circuits. The automatic tools often fail to find desirable solutions for highly organized circuits. One of such circuits is the clock generator. Its schematic is shown in Figure 9.28, and the hand layout is shown in Figure 9.32. The layout matches the schematic closely with a great deal of organization in the placement. Each inverter is placed one after the other in the chain in almost the exact manner as was drawn in the schematic. The routing is also very organized. The four power lines run horizontally across the cell so that the inverters and NOR gates can be directly tiled. The four power lines are a result of separating the substrate and well contacts from the supply and ground lines to reduce noise coupling.

The schematic of the latch is shown in Figure 9.29 while the layout is depicted in Figure 9.33. Table 9.23 lists the various circuits in the $\Sigma - \Delta$ CONVERTER and the required tools for the synthesis. The layout for the final chip is shown in Figure 9.34. The major components of the integrated circuit are labeled. At the top the clock generator is depicted. A set of shift registers for interfacing purposes is also shown on the top of the circuit. On the right one can find the bias circuitry, the latches and the D/A. The two integrators are in the lower left hand corner of the chip. The area not including the bonding pads (active area), is 1.1 mm^2 . The total area is approximately 3.1 mm^2 .

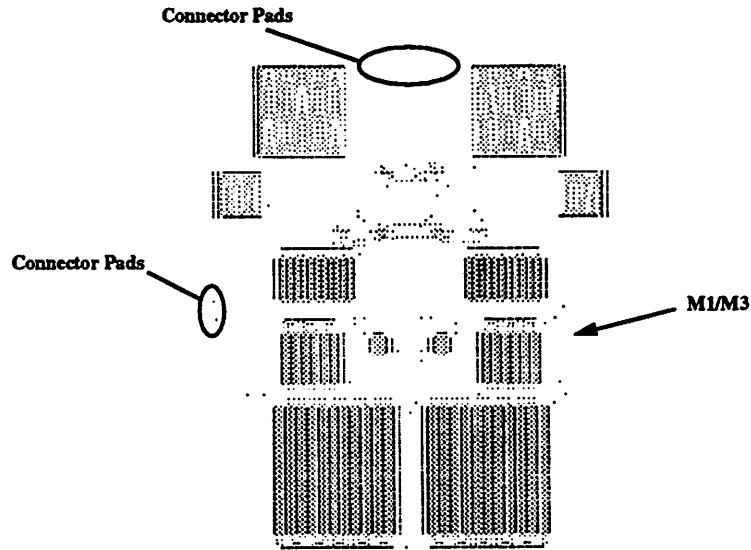


Figure 9.31: *Placed, routed, compacted OTA* (Courtesy of H. Chang and E. Felt)

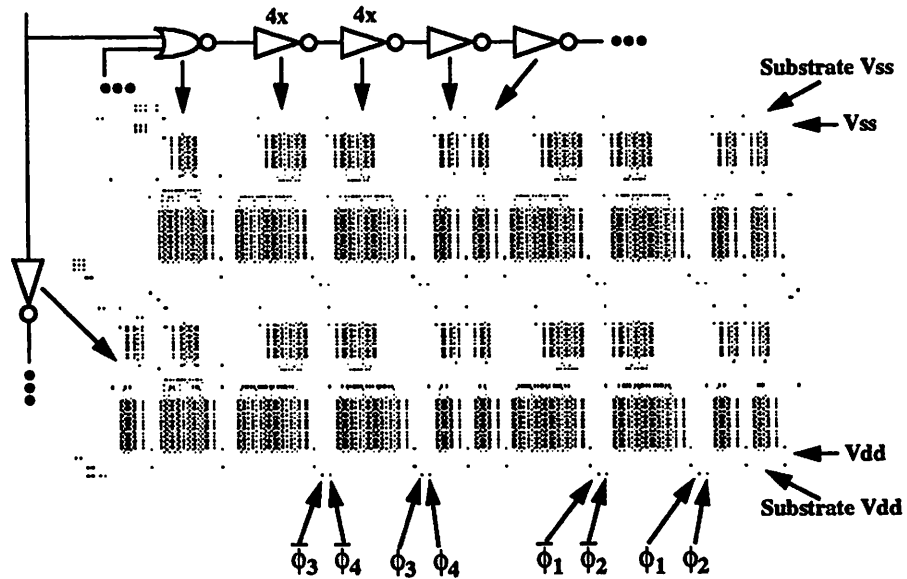


Figure 9.32: *Layout of the clock generator* (Courtesy of H. Chang and E. Felt)

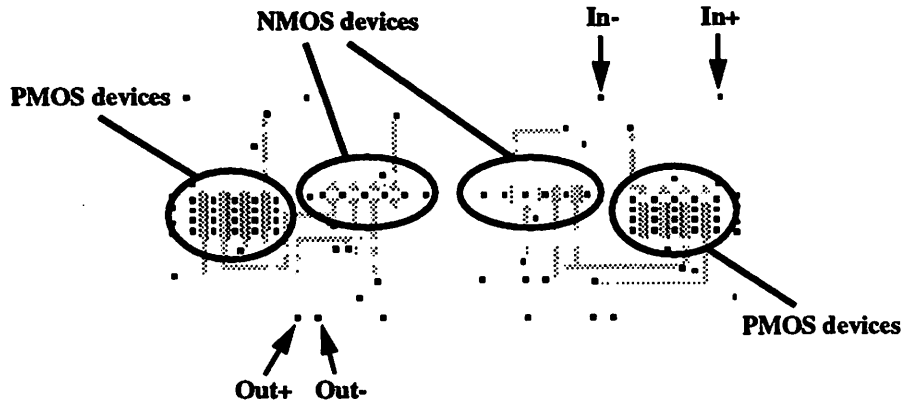


Figure 9.33: *Layout of the latch* (Courtesy of H. Chang and E. Felt)

Circuit	Element Generation	Stack Placement	Routing	Compaction	Time (days)
Bias	MKSTACK	PUPPY-A	ROAD	SPARCS-A	0.5
OTA	MKSTACK	PUPPY-A	ROAD	SPARCS-A	1.0
Integrator	MKSTACK	PUPPY-A	ROAD	SPARCS-A	0.5
Comparator	MKSTACK	PUPPY-A	hand done		0.5
Latch	MKSTACK	PUPPY-A	ROAD	SPARCS-A	0.5
D/A	MKSTACK	PUPPY-A	ROAD	SPARCS-A	0.5
Clock	hand made				0.5
Pads	dedicated module generator				0.1
Chip	n/a	n/a	MOSAICO [268]	SPARCS-A	1.5

Table 9.23: *Estimated man-time for an unexperienced tool-user to perform the layout of the $\Sigma - \Delta$ CONVERTER* (Courtesy of H. Chang and E. Felt)

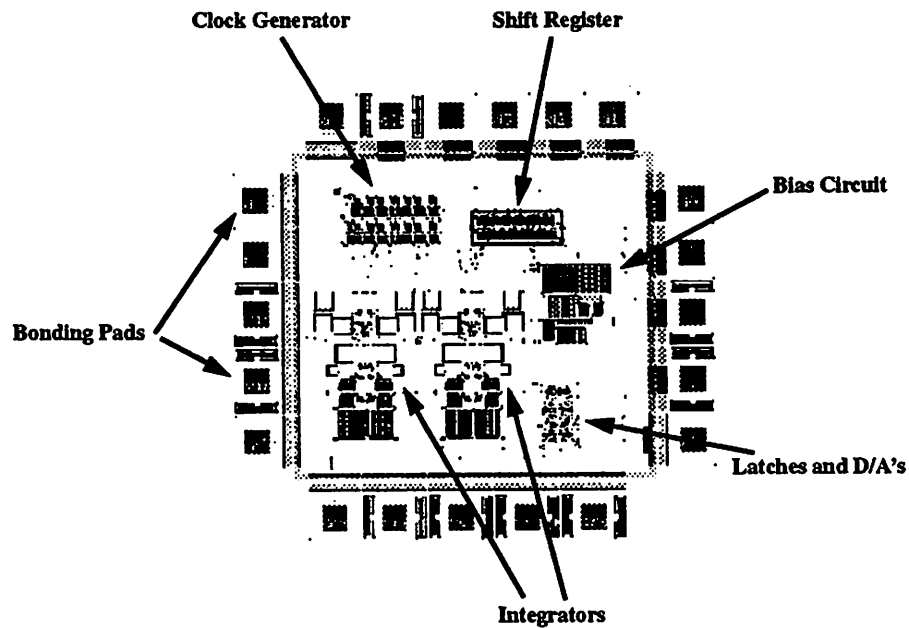


Figure 9.34: *Layout of $\Sigma - \Delta$ CONVERTER* (Courtesy of H. Chang and E. Felt)

Extraction and Verification

Hierarchical simulation was used to verify circuit performance and non-hierarchical simulations to verify circuit functionality. Using SPICE, each sub-block's performance was determined, the results were used in MIDAS's models to compute the SNR for the $\Sigma - \Delta$ CONVERTER.

In the non-hierarchical method, the entire circuit was flattened and extracted completely. Pad-to-pad SPICE transient analysis was then applied for a number of clock periods, given a fixed amplitude sine wave at the input. The higher SNR value verified the circuit's functionality. The chip was fabricated on a MOSIS $0.8 \mu\text{m}$ process. Preliminary results are shown in Figure 9.24. It is believed that these results are pessimistic, for two main reasons. First, we believe that the circuit was slowed down by a logic which was not optimized for the process used. Secondly, the SNR below specs is probably due to an insufficient MIDAS model for the integrator, which did not take several effects into account, hence preventing the optimizer from reaching a solution which could guarantee specifications.

Specifications	Value
SNR	69 dB
f_x	165 kHz

Table 9.24: $\Sigma - \Delta$ CONVERTER *experiment results*

Parasitic	Parameter	Type of model	Generation time (sec)
single models			
substrate cap.	C_1	1st ord. polynomial	414.8
cross-over cap.	C_{12}	2nd ord. polynomial	6281.7
parallel line cap.	C_{12}	2nd ord. polynomial	1888.3
char. impedance	Z_w	logarithmic	-
bend	C, L_1, L_2	lumped passive	-
via hole	L_1	1st ord. polynomial	2401.0 (HP9000/750)
correction factors			
adjacent line	ΔC_1	2nd ord. polynomial	3040.9
adjacent pad/spiral	ΔZ_w	2nd ord. polynomial	2543.7
adjacent via	ΔZ_w	2nd ord. polynomial	36 hrs (HP9000/750)

Table 9.25: *Analytical models used in synthesis for parasitic control*

9.3 RF and Microwave Benchmark Library

A number of analytical models for MMIC microstrip lines has been computed for the GaAs technology in which the chips were designed. Table 9.25 shows a summary of the models, along with the CPU time required for their generation on a DEC Station 5000/240. See Appendix E for a detailed description of the models. The via model required a large CPU time due to the high level of accuracy selected for HFSS in the 3-D field analysis. Notice however that all models have been generated for a given technology, hence no further computational effort is needed for all future designs realized in the same technology.

The routing methodology and the CAD tools described in this dissertation have been applied to a number of commercial microwave circuits. The circuits were fabricated in HP's GaAs technology and tested. See Table 9.26. The measured performance confirmed the predictions in full with a yield a fraction of a percent lower than that of hand layout. The results of the layout synthesis are shown hereafter in detail for one circuit. The discussion

Circuit	Frequency range (GHz)	# Parasitic constr.	CPU (sec)	Specs met
TWA	1-10	39	749	3/3
PCN38	20-40	72	2802	3/3
WILK	1-20	145	3450	2/2
TQ	2-7	131	2991	4/4

Table 9.26: Performance of a set of commercial RF benchmarks

Freq. range (GHz)	25-30	30-35	35-40	40-43
$ S_{11} $ (dB)	-0.15	-0.03	-0.54	-0.12
$ S_{22} $ (dB)	+1.22	-1.43	+1.59	+1.24
$ S_{21} $ (dB)	+0.54	+1.21	+1.57	+0.53

Table 9.27: Worst-case performance degradation form nominal of PCN38

and results of the synthesis of all tested benchmarks can be found in [272].

Consider the three-stage amplifier shown in Figure 9.35. The specifications are illustrated in Figure 9.36. All relevant parasitic constraints were generated by PARCAR in a total CPU time of 2454 sec on a DEC Station 5000/240. PARCAR computed bounds, the most critical of them are listed in Table 9.28, which were then used by CORAL to route the pre-placed circuit. The resulting layout is shown in Figure 9.37 and the extracted deviations from nominal performance are listed in Table 9.27. The CPU time for the routing phase

freq. range (GHz)	0-1.5	1.5-18.0	18.0-26.0
1: ΔL_{12}	<9.3%	<2.6%	<1%
2: ΔL_{56}	<8.0%	<11.5%	<10%
3: # bends in 1 and 2	2	2	2
3: # bends in 7 and 8	10	10	10
4: $\Delta Z_1, \Delta Z_2$	<1%	<3%	<2%
5: $\Delta \alpha_1, \Delta \alpha_2$	<4%	<2%	<10%

Table 9.28: Constraints on critical interconnect lines in PCN38 as computed using PARCAR. Terms of type ΔL_{xy} denote a bound to the maximum attainable length mismatch between nets x and y

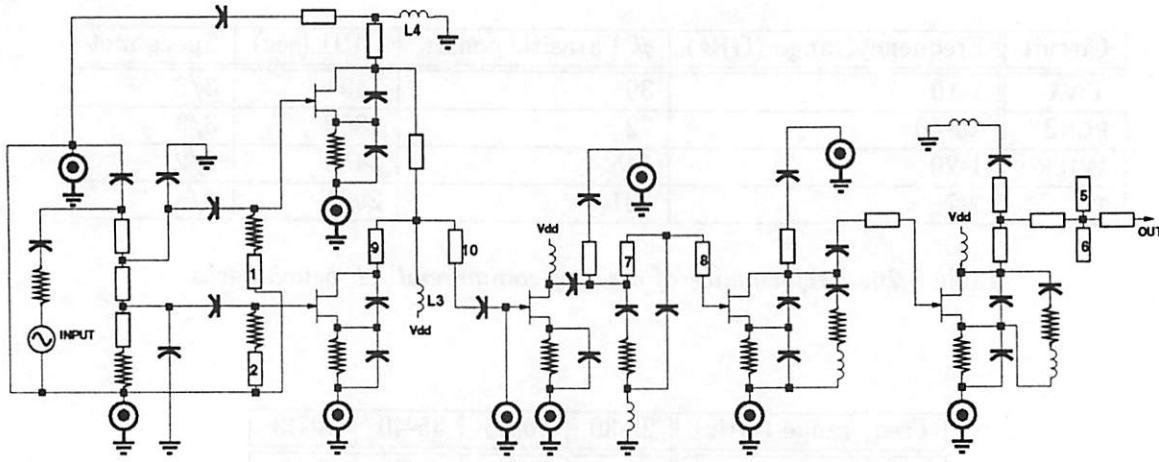


Figure 9.35: Schematic of PCN38

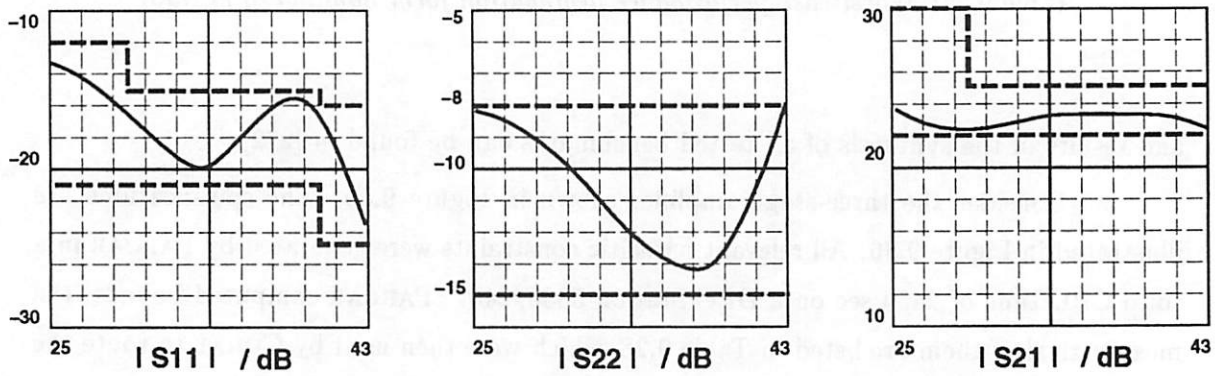


Figure 9.36: Performance of PCN38

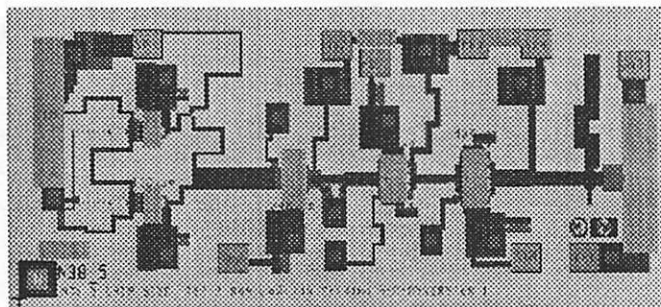


Figure 9.37: Final layout of PCN38

was 348 sec on a DEC Station 5000/240. As expected, all performance specifications were met. Figure 9.36 shows the results of $|S_{11}|$, $|S_{22}|$ and $|S_{21}|$ respectively.

Chapter 10

Conclusions

*Lo duca e io per quel cammino ascoso
intrammo a ritornar nel chiaro mondo;
e senza cura aver d'alcun riposo,
salimmo sù, el primo e io secondo,
tanto ch'i' vidi de le cose belle
che porta 'l ciel, per un pertugio tondo.*

E quindi uscimmo a riveder le stelle.

Dante Alighieri, “Inferno”, Canto XXXIV

10.1 Conclusions

The main purpose of the research described in this dissertation was the study and realization of various optimization algorithms for the problem of semi-automated analog and mixed-signal IC physical assembly. A general constraint-driven methodology was described for gluing every component of the assembly consistently with the paradigm underlying the approach. The key points of approach can be summarized as follows:

- a rigorous methodology is applied to translate high-level performance specifications into the set of constraints that the tools are able to control. The constraint generation technique guarantees that if low-level constraints are satisfied, then all high-level specifications will be met.

- At each step of the layout design the tools are able to enforce constraints on all low-level parameters of the circuit.
- Infeasibility is detected as soon as possible in the design flow. A quantitative analysis allows us to determine the causes of infeasibility and to address a re-design strategy.

The tools presented cover all the major steps of layout synthesis, namely placement, routing, compaction, module generation and extraction. The presence of a constraint-aware compactor allows a more aggressive approach in the routing phase, thus improving the success rate and the robustness of the entire synthesis. All tools have been integrated in an environment where they share data-base, constraint representation, parasitic models, and all performance analysis methods.

Moreover, the impact of each layout step on the flexibility of the entire design flow has been analyzed in detail. Several issues related to efficient parasitic estimation both in 3D and 2D settings have been addressed. Techniques for efficient and accurate substrate optimization before and after the creation of circuit libraries have also been explored and applied to a number of designs.

The ideas and algorithms presented in this dissertation have been experimented in industrial strength circuits, most of which have been fabricated and tested in modern CMOS technologies. The original constraint-driven paradigm has been applied recently to schematic design of analog and mixed-signal ICs. The technique has shown, also in this context, its advantages, namely: flexibility, robustness and effectiveness in early pinpointing errors and specification violations.

10.2 Future Work

Many areas remain open to further research. First, the continuous reduction in size of IC technologies represent a serious challenge to parasitic models in terms of both accuracy and compactness. Deep submicron technologies for example show a completely new scenario for parasitic effects.

Second, radio-frequency circuits will become a major player in the future of IC design at all levels, from communication systems to microprocessor technology. Due to the distributed parasitic effects observed at these frequencies, novel approaches to physical automation and schematic design need be adopted and appropriately analyzed. The limitation

of the algorithms to discretely valued attributes is a serious one and limits their applicability to a wider variety of problems. Removing this restriction is certainly an important area of further research.

Third, it is not clear if the partition of physical assembly into its traditional phases, placement, routing and compaction, will survive the technological revolution induced by deep submicron processes. In this dissertation we have shown the advantages of using simultaneous optimization techniques. The improvement over previous approaches is mainly the reduction of degrees of freedom in the solution resulting from multiple optimization procedures fused into one, thus making these techniques practical. Future physical assembly systems will indubitably take advantage of constraint-based techniques and even push further their development.

Finally, an open question which deserves further investigation is whether or not the existing tools and methodologies are a viable aid to IC designers as the original CAD tools were and are today. More emphasis will be probably given to enhancing and optimizing the communication and information presentation to the user. This factor has shown to be of fundamental importance in the past for the acceptance of CAD tools and will become even more important in the future in many design environments.

Appendix A

Convergence of Modified Placement Algorithms

A.1 Modification of Search Space

SA has been proved to converge to the global minimum for the class of problems to which placement belongs provided a set of conditions. In [181] conditions were set on the number of moves per annealing temperature, $t_k \rightarrow \infty$. A stronger result, proved, among others, by [192], guarantees convergence for $t_k = 1$, and $T_A(k) = a/\ln k$, with a sufficiently large.

Theorem 1 *The conditions of [181] are sufficient for the convergence of a SA based placement algorithm where the search space $\{S\}$ is modified dynamically as discussed in section 4.4 during the unfolding of the annealing.*

Proof : The annealing algorithm with the proposed modification is still a Markov Chain. Since all new states added to the search space are reachable, and the same properties apply to these states as to the others, the chain is regular and recurrent. Since, for fixed temperature, the transition rule to go from a state to the next is invariant, the chain is also homogeneous. Hence, the standard theory of homogeneous Markov Chains applies. As a result, at a given temperature all states of the chain converge to a stationary probability distribution (Theorems 3.1.4 and 3.2.1 in [181]). This implies that at each temperature a global minimum is reached, thus the convergence of the algorithm follows (Propositions 3.2.1, 3.2.2 and Theorem 3.2.2 in [181]). ■

Similar arguments can be used to prove convergence in the sense of [192], thus ensuring that the whole approach is robust.

A.2 Substrate-Aware Placement

Theorem 2 *The conditions of [181] are sufficient for the convergence of a SA based placement algorithm with the modifications discussed in section 4.6.*

Proof : Two equivalent resistive networks are used to represent the substrate in terms of its electrothermal behavior. Both networks are linear with no storage elements. Furthermore, due to finite dimensions of substrate geometries and to the inherent properties of the algorithm, each resistive component is bounded from above and below. Hence, all temperature and noise estimates are necessarily bounded. Consequently the properties of the Markov Chain underlying the annealing are not modified. Since all new states added to the search space are all reachable, and the same properties apply to these states as to the others, the chain is regular and recurrent. Since, for fixed annealing temperature, the transition rule to go from a state to the next is invariant, the chain is also homogeneous. Hence, the standard theory of homogeneous Markov Chains applies. As a result, at a given annealing temperature all states of the chain converge to a stationary probability distribution (Theorems 3.1.4 and 3.2.1 in [181]). This implies that at each annealing temperature a global minimum is reached, thus the convergence of the algorithm follows (Propositions 3.2.1, 3.2.2 and Theorem 3.2.2 in [181]). ■

Similar arguments can be used to prove convergence in the sense of [192], thus ensuring that the whole approach is robust. Convergence in the sense of [181] is also guaranteed in the case one or more cell is provided with a number of guard rings. In [175] convergence has been proven for SA in which swaps between cell implementations are allowed.

Appendix B

Compaction Roundoff Calculations

The formulation of the problem and the nature of the starting point allow us to enforce the symmetry constraints using linear programming rather than integer programming. In fact, under the following conditions the analog compaction problem is unimodular, i.e. the optimum solution contains only integer coordinates [273]:

Theorem 3 *If :*

- *The nodes of each symmetric pair are also constrained by design-rule minimum spacing requirements,*
- *All design-rule minimum spacing requirements are integers, and*
- *The leftmost node is located at an integer coordinate,*

Then there exists a solution to the LP problem where all coordinates are integer.

Proof : Let us first consider the case of a node on the critical path. Suppose that some node n_0 is on the critical path but does not converge to an integer in the optimum solution. The critical path between the leftmost node and n_0 must contain at least one *unnested* symmetry constraint with a non-integer separation between the symmetric nodes, since all other constraints are integers and any sum of integers is an integer. Call the nodes of this symmetry constraint n_{s1} and n_{s2} ¹. Since n_{s1} and n_{s2} are unnested, the only constraints

¹Unnested refers to the fact that there is no pair of symmetrically-constrained nodes on the critical path between n_{s1} and n_{s2} .

between them are minimum or maximum spacing constraints. All minimum spacing constraints are integers, so the total minimum distance between n_{s1} and n_{s2} must be an integer. If the minimum separation between the two symmetric nodes in an integer and the actual separation is not an integer distance, then either the symmetric nodes are not on the critical path or the solution is not optimal because the area could be decreased by moving the two symmetric nodes closer together. These conclusions contradict the original assumption. Thus if n_0 is on the critical path then it must converge to an integer value.

Consider now nodes not on the critical path. These nodes may or may not converge to an integer value. We can always obtain a solution with the same cost, where the non-critical nodes are at integer coordinates. ■

To adjust non-integer to integer coordinates, the nodes have to be “packed” against nodes on the critical path. Since each non-critical node is separated from critical nodes by only integer spacing constraints, it can be moved to the left by extending the linear programming objective function to be:

$$\text{minimize : } \sum_{i=1}^N (x_i - x_0) , \quad (\text{B.1})$$

where x_i is the location of node i , x_0 that of the leftmost node, and N is the number of nodes. This objective function forces every node to be as close as possible to the leftmost node, x_0 .

Another solution can be obtained by rounding off the coordinates of any nodes which do not converge to integers. There is always sufficient “slack” in the constraints so that these non-critical nodes can be rounded off to the nearest integer without violating any constraints and without increasing the area of the circuit. These propositions will be considered in turn:

Rounding off a non-critical node can never increase the area of the circuit. Rounding off never moves a node by more than 0.5 units, and if the constraints on that node were such that there was not at least 1 unit of slack on the node, then the node would have been on the critical path. This arises from the fact that all non-symmetry constraints are integer-valued.

Rounding off a non-critical node can never violate a constraint. Since all non-symmetry constraints are integer-valued, any node which is not located at an integer coordinate has at least enough slack to move to the nearest integer coordinate. If it did not, then its original non-integer coordinate would not have satisfied all of the spacing constraints. To minimize computational complexity it is desirable to keep the linear programming objective function as simple as possible, hence the round-off approach is superior to the “packing” approach for most problems. Moreover, all elements on the left may result in an “inferior” layout.

Appendix C

Green's Function Related Theory

C.1 Non-Zero Depth Contact Calculation

In order to consider contacts characterized by a non-zero depth c , equation (8.12) is re-written as follows [255]

$$p_{ij} = \frac{\bar{\Phi}_i}{Q_j} = \frac{1}{V_i V_j} \int_{-c_4}^0 \int_{b_3}^{b_4} \int_{a_3}^{a_4} \int_{-c_2}^0 \int_{b_1}^{b_2} \int_{a_1}^{a_2} G(x, y, z; x', y', z') dx dy dz dx' dy' dz', \quad (\text{C.1})$$

where the Green's Function is computed as described in section 8.2 and physical dimensions a, b, c of the contacts are illustrated in Figure C.1. After the appropriate manipulations the term p_{ij} is derived as

$$p_{ij} = \frac{1}{c_2 c_4 a b \epsilon_N \beta_N} \left(-\beta_N \left(\frac{c_s c_g^2}{2} + \frac{c_s^3}{6} \right) + c_2 c_4 \Gamma_N \right) + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} k_{mn} \frac{[\sin(m\pi \frac{a_2}{a}) - \sin(m\pi \frac{a_1}{a})] [\sin(m\pi \frac{a_4}{a}) - \sin(m\pi \frac{a_3}{a})]}{(a_2 - a_1)(a_4 - a_3)} \times \frac{[\sin(n\pi \frac{b_2}{b}) - \sin(n\pi \frac{b_1}{b})] [\sin(n\pi \frac{b_4}{b}) - \sin(n\pi \frac{b_3}{b})]}{(b_2 - b_1)(b_4 - b_3)}, \quad (\text{C.2})$$

where the term k_{mn} replaces the following expression

$$k_{mn} = C_{mn} \frac{a^2 b^2}{m^2 n^2 \pi^4} \left(f_{mn} - \frac{c_s c_g^2}{2 a b \epsilon_N c_2 c_4} \right), \quad \text{with } c_g = \max(c_2, c_4) \text{ and } c_s = \min(c_2, c_4). \quad (\text{C.3})$$

Equation (C.2) becomes (8.23) and (C.3) turns into (8.24) when c_2 and c_4 are set to zero. ■

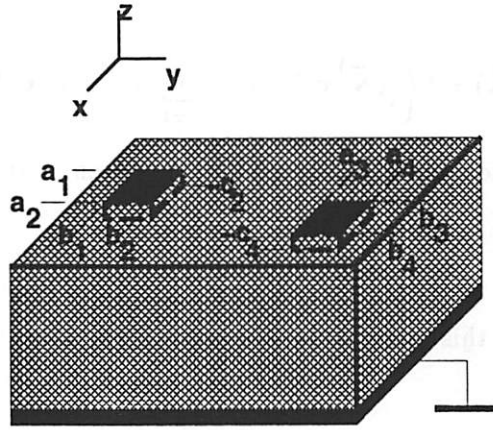


Figure C.1: Non-zero depth contacts and dimensions

C.2 Scaling Coefficient of Induction Matrix

The solution of the capacitive problem can be used to solve the resistive problem as well. Using equations (8.9), (8.10) and (8.14) one can construct the coefficient of induction matrix \mathbf{c} [255]. From \mathbf{c} , admittance matrix $\bar{\mathbf{Y}}$, relating each pairs of contacts i and j can be derived as following

$$\bar{\mathbf{Y}} = \bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}} - \Lambda(\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) + \mathbf{I} [(\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) \mathbf{e}] , \quad (\text{C.4})$$

where operator $\Lambda(\cdot)$ returns a diagonal matrix whose entries are identical the diagonal elements of the argument and \mathbf{e} is a $N_c \times 1$ unity vector, defined as $\mathbf{e} = [1, \dots, 1]^T$. Moreover, $N \times N_c$ matrix $\bar{\mathbf{X}}$ is defined as

$$\bar{\mathbf{X}} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{N_c}] ,$$

where $N \times 1$ vector \mathbf{e}_i is

$$\mathbf{e}_i = [0, \dots, 0, \underbrace{1, \dots, 1}_{\text{indices assoc. w. contact } i}, 0, \dots, 0]^T .$$

indices assoc. w. contact i

Let \mathbf{b}_i be a $N_c \times 1$ vector with all entries equal to zero except those indexed i , then $\Lambda(\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}})$ can be re-written as

$$\Lambda(\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) = \mathbf{I} \left\{ \sum_{i=1}^{N_c} [\mathbf{b}_i^T (\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) \mathbf{b}_i] \mathbf{b}_i \right\} ,$$

hence (C.4) simplifies to

$$\bar{\mathbf{Y}} = (\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) + \mathbf{I} \left\{ [(\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) \mathbf{e}] - \sum_{i=1}^{N_c} [\mathbf{b}_i^T (\bar{\mathbf{X}}^T \mathbf{c} \bar{\mathbf{X}}) \mathbf{b}_i] \mathbf{b}_i \right\}. \quad (\text{C.5})$$

By appropriate manipulations on \mathbf{c} , (C.5) can be re-written in the form of equation (8.47) as

$$\bar{\mathbf{Y}} = \bar{\mathbf{X}}^T \tilde{\mathbf{c}} \bar{\mathbf{X}}.$$

For simplicity however, in this dissertation the term $\tilde{\mathbf{c}}$ has been replaced with the \mathbf{c} notation. ■

Appendix D

Sensitivity Analysis

D.1 Canonical Representation of Performance

A generalized expression for the computation of sensitivities from a set of arbitrary performances has been derived in [31, 135]. This formulation has been used by us to represent all performances analyzed in a compact and rigorous way, thus ensuring flexibility of our design tools. For completeness the formulation has been reviewed hereafter.

Let us consider an arbitrary performance K , let

- \mathbf{x} be a vector of design parameters (e.g. capacitances, MOS channel length, etc.)
- \mathbf{y} be a vector of circuit variables (e.g. voltages, charges, etc.)
- ω be an independent variable (e.g. frequency, time)

And let

$$g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \omega), \omega) \tag{D.1}$$

be an analytical function representing the performance K .

Define the characteristic function of performance K as

$$h(\mathbf{x}, \mathbf{y}(\mathbf{x}, \omega), \omega) = 0 \tag{D.2}$$

The characteristic function h represents implicitly the operating point at which the sensitivity computations have to be performed.

The performance sensitivity of the performance K with respect to a set of design parameters

\mathbf{x} is

$$S_{\mathbf{x}}^K = \frac{\partial g}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^T \frac{\partial g}{\partial \mathbf{y}} - \left\{ \frac{\frac{\partial g}{\partial \omega} + \left(\frac{\partial g}{\partial \mathbf{y}} \right)^T \frac{\partial \mathbf{y}}{\partial \omega}}{\frac{\partial h}{\partial \omega} + \left(\frac{\partial h}{\partial \mathbf{y}} \right)^T \frac{\partial \mathbf{y}}{\partial \omega}} \right\} \left[\frac{\partial h}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^T \frac{\partial h}{\partial \mathbf{y}} \right] \quad (\text{D.3})$$

where the expression $\frac{\partial f}{\partial \mathbf{y}}$ has the same meaning of $\nabla f(\mathbf{y})$, if $f : \mathbf{R}^n \rightarrow \mathbf{R}$.

D.2 Coefficient of Potential and Technology Parameters

Let us consider first parameters Γ_N and β_N as defined in equation (8.20).

Assume $T_\ell = \epsilon_\ell$, then all Γ_k and β_k will not depend on ϵ_ℓ when $0 \leq k < \ell$, hence

$$\begin{bmatrix} \frac{\partial \beta_k}{\partial T_\ell} \\ \frac{\partial \Gamma_k}{\partial T_\ell} \end{bmatrix} = \mathbf{0}, \quad \forall 0 \leq k < \ell.$$

Consider first the case in which $k = \ell$. Equation (8.20) becomes

$$\begin{bmatrix} \frac{\partial \beta_\ell}{\partial T_\ell} \\ \frac{\partial \Gamma_\ell}{\partial T_\ell} \end{bmatrix} = \frac{\epsilon_{\ell-1}}{\epsilon_\ell^2} \begin{bmatrix} -\cosh^2(\theta_\ell) & -\cosh(\theta_\ell) \sinh(\theta_\ell) \\ \cosh(\theta_\ell) \sinh(\theta_\ell) & \sinh^2(\theta_\ell) \end{bmatrix} \begin{bmatrix} \beta_{\ell-1} \\ \Gamma_{\ell-1} \end{bmatrix}, \quad (\text{D.4})$$

where $\Gamma_{\ell-1}$ and $\beta_{\ell-1}$ are already known, while $\theta_\ell = \gamma_{mn}(d-d_\ell)$ and $\gamma_{mn} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}$.

Secondly, consider the case in which $k = \ell + 1$. Equation (8.20) becomes

$$\begin{bmatrix} \frac{\partial \beta_{\ell+1}}{\partial T_\ell} \\ \frac{\partial \Gamma_{\ell+1}}{\partial T_\ell} \end{bmatrix} = \frac{1}{\epsilon_{\ell+1}} \begin{bmatrix} \cosh^2(\theta_{\ell+1}) & \cosh(\theta_{\ell+1}) \sinh(\theta_{\ell+1}) \\ -\cosh(\theta_{\ell+1}) \sinh(\theta_{\ell+1}) & -\sinh^2(\theta_{\ell+1}) \end{bmatrix} \begin{bmatrix} \beta_\ell \\ \Gamma_\ell \end{bmatrix} + \begin{bmatrix} \frac{\epsilon_\ell}{\epsilon_{\ell+1}} \cosh^2(\theta_{\ell+1}) - \sinh^2(\theta_{\ell+1}) & (\frac{\epsilon_\ell}{\epsilon_{\ell+1}} - 1) \cosh(\theta_{\ell+1}) \sinh(\theta_{\ell+1}) \\ (1 - \frac{\epsilon_\ell}{\epsilon_{\ell+1}}) \cosh(\theta_{\ell+1}) \sinh(\theta_{\ell+1}) & \cosh^2(\theta_{\ell+1}) - \frac{\epsilon_\ell}{\epsilon_{\ell+1}} \sinh^2(\theta_{\ell+1}) \end{bmatrix} \begin{bmatrix} \frac{\partial \beta_\ell}{\partial T_\ell} \\ \frac{\partial \Gamma_\ell}{\partial T_\ell} \end{bmatrix}. \quad (\text{D.5})$$

For $\ell + 1 < k \leq N$, $\partial \beta_k / \partial T_\ell$ and $\partial \Gamma_k / \partial T_\ell$ are computed as

$$\begin{bmatrix} \frac{\partial \beta_k}{\partial T_\ell} \\ \frac{\partial \Gamma_k}{\partial T_\ell} \end{bmatrix} = \begin{bmatrix} \frac{\epsilon_{k-1}}{\epsilon_k} \cosh^2(\theta_k) - \sinh^2(\theta_k) & (\frac{\epsilon_{k-1}}{\epsilon_k} - 1) \cosh(\theta_k) \sinh(\theta_k) \\ (1 - \frac{\epsilon_{k-1}}{\epsilon_k}) \cosh(\theta_k) \sinh(\theta_k) & \cosh^2(\theta_k) - \frac{\epsilon_{k-1}}{\epsilon_k} \sinh^2(\theta_k) \end{bmatrix} \begin{bmatrix} \frac{\partial \beta_{k-1}}{\partial T_\ell} \\ \frac{\partial \Gamma_{k-1}}{\partial T_\ell} \end{bmatrix} \quad (\text{D.6})$$

recursively, where $\partial \beta_{k-1} / \partial T_\ell$ and $\partial \Gamma_{k-1} / \partial T_\ell$ are obtained directly from equation (D.5).

The recursion (D.6) ends when $\partial \beta_N / \partial T_\ell$ and $\partial \Gamma_N / \partial T_\ell$ are found. ■

Next, assume $T_\ell = d_\ell$, the layer thickness. Using a similar reasoning as before, consider first the case in which $k = \ell$. Equation (8.20) becomes

$$\begin{bmatrix} \frac{\partial \beta_\ell}{\partial T_\ell} \\ \frac{\partial \Gamma_\ell}{\partial T_\ell} \end{bmatrix} = -\gamma_{mn} \left(\frac{\epsilon_{\ell-1}}{\epsilon_\ell} - 1 \right) \begin{bmatrix} 2 \sinh(\theta_\ell) \cosh(\theta_\ell) & \cosh(2\theta_\ell) \\ -\cosh(2\theta_\ell) & -2 \sinh(\theta_\ell) \cosh(\theta_\ell) \end{bmatrix} \begin{bmatrix} \beta_{\ell-1} \\ \Gamma_{\ell-1} \end{bmatrix}, \tag{D.7}$$

where $\Gamma_{\ell-1}$ and $\beta_{\ell-1}$ are already known, while $\theta_\ell = \gamma_{mn}(d-d_\ell)$ and $\gamma_{mn} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}$. Secondly, consider again the case in which $\ell + 1 \leq k \leq N$, $\partial\beta_k/\partial T_\ell$ and $\partial\Gamma_k/\partial T_\ell$ are computed as

$$\begin{bmatrix} \frac{\partial \beta_k}{\partial T_\ell} \\ \frac{\partial \Gamma_k}{\partial T_\ell} \end{bmatrix} = \begin{bmatrix} \frac{\epsilon_{k-1}}{\epsilon_k} \cosh^2(\theta_k) - \sinh^2(\theta_k) & \left(\frac{\epsilon_{k-1}}{\epsilon_k} - 1\right) \cosh(\theta_k) \sinh(\theta_k) \\ \left(1 - \frac{\epsilon_{k-1}}{\epsilon_k}\right) \cosh(\theta_k) \sinh(\theta_k) & \cosh^2(\theta_k) - \frac{\epsilon_{k-1}}{\epsilon_k} \sinh^2(\theta_k) \end{bmatrix} \begin{bmatrix} \frac{\partial \beta_{k-1}}{\partial T_\ell} \\ \frac{\partial \Gamma_{k-1}}{\partial T_\ell} \end{bmatrix} \tag{D.8}$$

recursively, where $\partial\beta_{k-1}/\partial T_\ell$ and $\partial\Gamma_{k-1}/\partial T_\ell$ are obtained directly from equation (D.7). The recursion (D.8) ends when $\partial\beta_N/\partial T_\ell$ and $\partial\Gamma_N/\partial T_\ell$ are obtained. ■

Consider now the sensitivity of the term k_{mn} with respect to parameter T_ℓ . k_{mn} is defined in equations (8.24) and (8.19); after full expansion of its terms, it becomes

$$k_{mn} = \frac{ab C_{mn}}{m^2 n^2 \pi^4 \gamma_{mn} \epsilon_N} \frac{\beta_N \tanh(\gamma_{mn} d) + \Gamma_N}{\beta_N + \Gamma_N \tanh(\gamma_{mn} d)}, \text{ with } \gamma_{mn} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}.$$

Hence, assuming T_ℓ is either a doping level, which results in different ϵ_ℓ , or a layer thickness d_ℓ , the sensitivity of k_{mn} with respect to T_ℓ , $\forall 0 \leq \ell < N$ is computed as

$$\frac{\partial k_{mn}}{\partial T_\ell} = \frac{ab C_{mn}}{m^2 n^2 \pi^4 \gamma_{mn} \epsilon_N} \times \frac{[\dot{\beta}_N \tanh(\gamma_{mn} d) + \dot{\Gamma}_N][\beta_N + \Gamma_N \tanh(\gamma_{mn} d)] - [\beta_N \tanh(\gamma_{mn} d) + \Gamma_N][\dot{\beta}_N + \dot{\Gamma}_N \tanh(\gamma_{mn} d)]}{[\beta_N + \Gamma_N \tanh(\gamma_{mn} d)]^2}, \tag{D.9}$$

where the terms $\dot{\Gamma}_N = \partial\Gamma_N/\partial T_\ell$ and $\dot{\beta}_N = \partial\beta_N/\partial T_\ell$ are computed from equations (D.6) and (D.8). Similarly, using (D.6), (D.8) and, slightly modified, (D.9), expressions can be easily derived for $T_\ell = d$ or ϵ_N . ■

Finally, consider the sensitivity of term p_{ij} with respect to contact depth c . Expressions for term p_{ij} in presence of zero and non-zero depth are shown in equation (8.23)

and (C.2), respectively. Assume that all contacts have identical depth c , then sensitivity $\partial p_{ij}/\partial c$ is computed as follows

$$\begin{aligned} \frac{\partial p_{ij}}{\partial c} = & -\frac{2}{3} \frac{\beta_N}{ab\epsilon_N\beta_N} + \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{\partial k_{mn}}{\partial c} \frac{[\sin(m\pi \frac{a_2}{a}) - \sin(m\pi \frac{a_1}{a})] [\sin(m\pi \frac{a_4}{a}) - \sin(m\pi \frac{a_3}{a})]}{(a_2 - a_1)(a_4 - a_3)} \\ & \times \frac{[\sin(n\pi \frac{b_2}{b}) - \sin(n\pi \frac{b_1}{b})] [\sin(n\pi \frac{b_4}{b}) - \sin(n\pi \frac{b_3}{b})]}{(b_2 - b_1)(b_4 - b_3)}, \end{aligned} \quad (\text{D.10})$$

where the term $\partial k_{mn}/\partial c$ is computed as

$$\frac{\partial k_{mn}}{\partial c} = \frac{\partial k_{mn}}{\partial c} = -C_{mn} \frac{a^2 b^2}{m^2 n^2 \pi^4} \frac{1}{2ab\epsilon_N}, \quad (\text{D.11})$$

where C_{mn} is defined in section 8.2. Due to the linearity of the DCT, it is possible to compute the sensitivity of the coefficient of potential by simply calculating k_{mn} and by performing the DCT on it. Several DCTs related to a variety of different depth can be stored and used for the efficient calculation of the effects of technology on a particular circuit. ■

Appendix E

RF Parasitic Models

E.1 Closed Form Expressions for Microstrip Lines

For a uniform, isotropic, non-magnetic substrate with permittivity ϵ_r , the characteristic impedance Z_o of the microstrip line depicted in Figure 7.6, is computed as follows [274, 275]

$$Z_o = \frac{\eta}{2\pi\sqrt{\epsilon_{re}}} \ln\left(\frac{8h}{w_e} + \frac{w_e}{4h}\right), \quad w/h \leq 1$$

$$Z_o = \frac{\eta}{\sqrt{\epsilon_{re}}} \left[\frac{w_e}{h} + 1.393 + \frac{2}{3} \ln\left(\frac{w_e}{h} + \frac{13}{9}\right) \right]^{-1}, \quad w/h \geq 1$$

where $\eta = 120\pi$. ϵ_{re} and $\frac{w_e}{h}$ are computed as follows

$$\epsilon_{re} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + 10 \frac{h}{w}\right)^{-\frac{1}{2}} - \frac{(\epsilon_r - 1)t/h}{4.6\sqrt{w/h}},$$

$$\frac{w_e}{h} = \frac{w}{h} + \frac{5t}{4\pi h} (1 + \ln \frac{4\pi w}{t}), \quad w/h \geq 1/2\pi$$

$$\frac{w_e}{h} = \frac{w}{h} + \frac{5t}{4\pi h} (1 + \ln \frac{2h}{t}), \quad w/h \leq 1/2\pi$$

where t is the conductor thickness. The loss α is the sum of conductor loss α_c and dielectric loss α_d , with

$$\alpha_c = 1.38A \frac{R_s}{hZ_o} \frac{32 - (w_e/h)^2}{32 + (w_e/h)^2} \text{ dB/m}, \quad w/h \leq 1$$

$$\alpha_c = 6.1 \times 10^{-5} A \frac{R_s Z_o \epsilon_{re}}{h} \left(w_e/h + \frac{\frac{2}{3} w_e/h}{w_e/h + 13/9} \right) \text{ dB/m}, \quad w/h \geq 1,$$

where the surface resistance R_s is the conductor resistance at frequencies for which the penetration depth is comparable or greater than the thickness of the conductor [276, Chp.4.4]. This has always been the case in our MMIC benchmarks. Otherwise the for-

mula $R_s = \sqrt{\pi f \mu_o \rho}$ should be used. A is computed as follows.

$$A = 1 + \frac{h}{w_e} \left(1 + \frac{1}{\pi} \ln \frac{2h}{t}\right), \quad w/h \geq 1/2\pi,$$

$$A = 1 + \frac{h}{w_e} \left(1 + \frac{1}{\pi} \ln \frac{4\pi}{t}\right), \quad w/h \leq 1/2\pi.$$

The dielectric loss α_d is computed as

$$\alpha_d = 27.3 \frac{\epsilon_r}{\epsilon_r - 1} \frac{\epsilon_{re} - 1}{\sqrt{\epsilon_{re}}} \frac{\tan \delta}{\lambda_0} \text{ dB/m},$$

where $\tan \delta$ is the loss tangent of the dielectric and λ_0 is the wavelength of the signal.

These models can be extended to higher frequencies by replacing the effective permittivity ϵ_{re} by its frequency-dependent value

$$\epsilon_{re}(f) = \epsilon_r - \frac{\epsilon_r - \epsilon_{re}}{1 + G(f/f_p)^2},$$

where

$$f_p = \frac{Z_o}{2\mu_o h}.$$

An G is a material dependent linear function of Z_o .

For the coupled lines shown in Figure 7.7 the even and odd capacitances are estimated as follows [240]

$$C_e = C_p + C_f + C'_f, \quad C_o = C_p + C_f + C_{ga} + C_{gd},$$

where for $0.2 \leq w/h \leq 2$, $0.05 \leq s/h \leq 2$ and $\epsilon_r \geq 1$

$$C_p = \epsilon_o \epsilon_r \frac{w}{h}, \quad 2C_f = \frac{\sqrt{\epsilon_{re}}}{c Z_o} - C_p,$$

[240] gives empirical expressions for C'_f , C_{ga} and C_{gd} .

Simpler analytical models can be found by using the solution of Laplace's equation in two dimensions to fit a simple function in the widths w_1 , w_2 of the conductors and their spacing s [193]. Hence, substrate capacitance C_{0i} , $i = 1, 2$ and coupling capacitance C_{12} are computed as follows

$$C_{0i} = k_0 + k_1 w_i, \quad i = 1, 2, \quad C_{12} = \mathcal{P}(1/s) + \mathcal{P}(1/(s + w_1)) + \mathcal{P}(1/(s + w_2)),$$

where \mathcal{P} is a polynomial of second order and has been reported in [193]. The models account for the parallel plate and fringing component of the capacitance. Since some of the field is diverted from ground to the other plate, the substrate capacitances C_{0i} , $i = 1, 2$ need be corrected by the terms

$$C_{1c} = \mathcal{P}(1/(s + s_0)) + \mathcal{P}(1/(s + s_0 + w_2)), \quad C_{2c} = \mathcal{P}(1/(s + s_0)) + \mathcal{P}(1/(s + s_0 + w_1)),$$

where s_0 is a constant and \mathcal{P} corresponds to the usual polynomial. These formulae have been used to obtain relatively accurate estimates of parasitic capacitances in CORAL.

E.2 Microstrip Line Discontinuities

The discrete component realization of circuits depicted in Figure 7.8 are computed as follows [240].

Configuration (a)

$$C_0 = w \exp(2.2036 \sum_{i=1}^5 K_\epsilon \log(\frac{w}{h})^{i-1}) \text{ pF} ,$$

where the coefficients K_ϵ are a monotonic function of ϵ_r . The stripline augmentation L is approximated by the expression $\frac{c Z_0 C_0}{\sqrt{\epsilon_{re}}}$, c being the wave velocity in empty space.

Configuration (b)

L_0 is generally computed with *ad hoc* simulations of the via used in the layout. In our implementation results from simulations performed with the package HFSS [277] have been used.

Configuration (c)

$$C_0 = w \frac{(14 \epsilon_r + 12.5)w/h - (1.83 \epsilon_r + 2.25)}{\sqrt{w/h}} , \quad w/h < 1 \text{ pF} ,$$

$$C_0 = w [(9.5 \epsilon_r + 1.25)w/h - (5.2 \epsilon_r + 7.0)] \text{ pF} ,$$

$$L_1 = 100 h [4\sqrt{w/h} - 4.21] \text{ nH} .$$

E.3 3-D Discontinuities

Consider a microstrip line of width w in presence of a via structure located at a distance s and size w_2 . The analytical model for the deviation ΔZ_0 from the nominal characteristic impedance Z_0 is

$$\Delta Z_0 = \frac{k_3}{s} + \frac{k_4}{s^2} + \frac{k_5}{s + w_2} + \frac{k_6}{(s + w_2)^2} .$$

The coefficients are obtained by fitting the model to the data from numerical simulations obtained through the package HFSS on a large number of geometries. Models based on the same formula have been obtained for pairs of a transmission line and spiral interconnection and short circuit trough a via of the same type.

Appendix F

Superconducting Models

All structures described in what follows refer, for the physical design parameters, to Figure F.1. All physical dimensions are in μm , the inductances are expressed in pHenry.

F.1 Single Line

From equation (7.18) one can derive the following parameters.

Top layer:

$$L_B = 0.0225 + \frac{0.912}{w} - \frac{0.556}{w^2}$$

Bottom layer:

$$L_T = 0.0046 + \frac{0.567}{w} - \frac{0.264}{w^2}$$

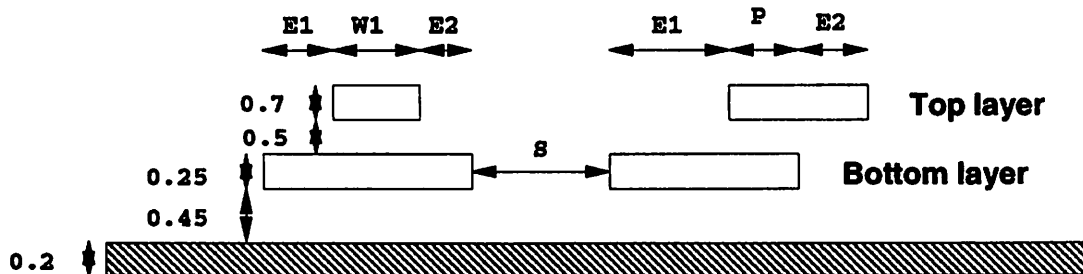


Figure F.1: *General configuration*

F.2 Coplanar Lines

From equations (7.19), (7.20) and (7.21), one derives the following parameters.

Self-inductance (including correction term):

$$L = 0.00443595 + \frac{-0.00072}{s+1} + \frac{0.605801}{w_1} + \frac{0.00111347}{s+1^2} + \frac{-0.39095}{w_1^2}$$

Mutual inductance:

$$M = 0.0003634 - \frac{0.00218}{s} - \frac{0.001558}{w_1} - \frac{0.001558}{w_2} - \frac{0.003135}{w_1 w_2} + \frac{0.0312024}{w_1 s} + \frac{0.0312024}{w_2 s}$$

F.3 Non-Overlapping Lines

From equations (7.5.4) and (7.22), one derives the following parameters.

Top layer:

$$L_T = 0.0009 + \frac{0.657}{w_1} - \frac{0.0005}{s} - \frac{0.494}{w_1^2}$$

Bottom Layer:

$$L_B = 0.0208 + \frac{1.068}{w_2} - \frac{0.0006}{s} - \frac{0.0135}{w_1} - \frac{0.974}{w_2^2}$$

Mutual inductance:

$$M = 0.00009 + \frac{0.00045}{s} - \frac{0.00106}{w_1} + \frac{0.00069}{w_2} + \frac{0.0665}{w_1 s} + \frac{-0.0154}{w_1 w_2} + \frac{0.0533}{w_2 s} - \frac{0.0066}{s^2}$$

The offset terms are negligible in the (non-)overlapping case due to the extremely small size of the oxide which separates the conductors.

F.4 Overlapping Lines

From equations (7.23), (7.24) and (7.25), one derives the following parameters.

Top layer:

$$L_T = -0.00185 + \frac{0.00473}{e_2} + \frac{0.00052}{e_1} + \frac{0.51}{w_1} - \frac{0.286}{w_1^2}$$

Bottom layer:

$$L_B = 0.00226 + \frac{0.844}{w_2} + \frac{0.0043}{e_1} - \frac{0.00039}{e_2} + \frac{0.00245}{p} - \frac{0.22}{e_2 p} + \frac{0.0102}{e_2^2} + \frac{0.0130}{p^2}$$

Mutual inductance:

$$M = -0.00077 + \frac{0.134}{w_1} + \frac{0.0534}{e_2} + \frac{0.0103}{e_1} - \frac{0.10505}{e_1 p} + \frac{0.00180}{e_2 p} + \frac{0.00943}{e_1 w_1} - \frac{0.0483}{e_2^2} + \frac{0.170337}{e_2 w_1} + \frac{0.150506}{w_1^2}$$

Appendix G

Software Availability

The tools described in this dissertation are available under the terms of the **Industrial Liaison Program (ILP)** of the *University of California at Berkeley*.

The tools PARCAR, PUPPY-A and SPARCS-A are part of the OCTTOOLS package available through ILP via ftp. The tools ROAD, LDO, ART, ESTPAR, SUBRES, VCOGEN, INDMOD, CAPMOD and MKSTACK can be obtained directly via world wide web at the following site
`ftp://ic.eecs.berkeley.edu/pub/Analog_Group/`

Please read the file README in this directory for details. The software obtained through ILP is free but subject to a handling fee. The code obtained through anonymous ftp is completely free and only subject to the rules of the *Research Agreement*.

For questions regarding the process of obtaining the OCTTOOLS software or for any additional questions, please contact the following address.

Industrial Liaison Program
Software Distribution
479 Cory Hall
University of California at Berkeley
Berkeley, CA 94720
(510) 643-6687
(510) 642-2845 (fax)
ilp.software@hera.berkeley.edu

Bibliography

- [1] U. Choudhury and A. L. Sangiovanni-Vincentelli, "Use of Performance Sensitivities in Routing of Analog Circuits", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 348–351, May 1990.
- [2] E. Charbon, E. Malavasi and A. L. Sangiovanni-Vincentelli, "Generalized Constraint Generation for Analog Circuit Design", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 408–414, November 1993.
- [3] R. S. Graham, "Relay Computer for Network Analysis", *Bell Labs Rec.*, vol. 31, pp. 152–157, April 1953.
- [4] A. F. Malmberg, F. L. Cornell and F. N. Hofer, "NET1 Network Analysis Program", Tech. rep. la-3119, 7090, Los Alamos Scientific Lab, Los Alamos, NM, August 1964.
- [5] H. W. Mathers, S. R. Sedore and J. R. Seuts, "Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation (SCEPTRE)", Ibm file 66-928-611, IBM Space Guidance Center, Oswego, NY, February 1967.
- [6] E. D. Johnson, C. T. Kleiner, L. R. McMurray, E. L. Steele and F. A. Vassallo, "Transient Radiation Analysis by Computer Program (TRAC)", Technical report, Rockwell Corp., Anaheim, CA, June 1968.
- [7] H. Shichman, "Computation of DC Solutions for Bipolar Transistor Networks", *IEEE Trans. on Circuit Theory*, vol. CT-16, pp. 460–466, 1969.
- [8] C. W. Gear, "Numerical Integration of Stiff Ordinary Differential Equations", rep. 221, University of Illinois, Urbana, 1967.

- [9] W. J. McCalla and Jr. W. G. Howard, "BIAS-3: A Program for the non-Linear DC Analysis of Bypolar Transistor Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-6, n. 1, pp. 14-19, February 1971.
- [10] G. D. Hachtel, R. K. Brayton and F. G. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design", *IEEE Trans. on Circuit Theory*, vol. CT-18, pp. 101-113, January 1971.
- [11] Various Authors, "ASTAP - Advanced Statistical Analysis Program", IBM Program Product Document SH20-1118-0, IBM Data Processing Division, White Plains, NY, 1973.
- [12] L. Nagel and R. A. Rohrer, "Computer Analysis of Nonlinear Circuits Excluding Radiation (CANCER)", *IEEE Journal of Solid State Circuits*, vol. SC-6, pp. 166-182, August 1971.
- [13] T. E. Idleman, F. S. Jenkins, W. J. McCalla and D. O. Pederson, "SLIC: A Simulator for Linear Integrated Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-6, pp. 188-204, August 1971.
- [14] L. Nagel, *SPICE2: A computer Program to Simulate Semiconductor Circuits*, PhD thesis, University of California at Berkeley, May 1975.
- [15] G. Hachtel and A. L. Sangiovanni-Vincentelli, "A Survey of Third-Generation Simulation Techniques", *Proc. of the IEEE*, vol. 69, n. 10, pp. 1264-1280, October 1981.
- [16] B. R. Chawla, H. K. Gummel and P. Kozak, "MOTIS - An MOS Timing Simulator", *IEEE Trans. on Circuits and Systems*, vol. CAS-22, pp. 901-909, December 1975.
- [17] H. DeMan, G. Arnout and P. Reyneart, "Mixed-Mode Circuit Simulation Techniques and Their Implementation in DIANA", in *Computer Design Aids for VLSI Circuits*, pp. 113-174, P. Antognetti and D. O. Pederson and H. DeMan Eds., Sijthoff & Noordhoff (The Netherlands), 1980.
- [18] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Relaxation-Based Electrical Simulation", *IEEE Trans. on Computer Aided Design*, vol. CAD-3, n. 4, pp. 308-330, October 1984.

- [19] K. A. Sakallah and S. W. Director, "An Event-Driven Approach for Mixed Gate and Circuit Level Simulation", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 1194–1197, May 1982.
- [20] A. L. Sangiovanni-Vincentelli, "Circuit Simulation", in *Computer Design Aids for VLSI Circuits*, pp. 19–112, P. Antognetti and D. O. Pederson and H. DeMan Eds., Sijthoff & Noordhoff (The Netherlands), 1980.
- [21] D. O. Pederson, "A Historical Review of Circuit Simulation", *IEEE Trans. on Circuits and Systems*, vol. CAS-31, pp. 103–111, January 1984.
- [22] A. E. Ruehli, "Circuit Analysis, Simulation and Design", in *Layout Design and Verification*, T. Ohtsuki Ed., North Holland, 1986.
- [23] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis", *IEEE Trans. on Computer Aided Design*, vol. CAD-9, n. 4, pp. 352–366, April 1990.
- [24] R. Brayton, G. Hachtel and A. L. Sangiovanni-Vincentelli, "A Survey of Optimization Techniques for Integrated-Circuit Design", *Proc. of the IEEE*, vol. 69, n. 10, pp. 1334–1364, 1981.
- [25] R. I. Dowell, *Automated Biasing of Integrated Circuits*, PhD thesis, University of California at Berkeley, April 1972.
- [26] W. J. McCalla, *Computer-Aided-Design of Integrated Bandpass Amplifiers*, PhD thesis, University of California at Berkeley, June 1972.
- [27] G. D. Hachtel, M. R. Lightner and H. J. Kelly, "Application of the Optimization Program AOP to the Design of Memory Circuits", *IEEE Trans. on Circuits and Systems*, vol. CAS-22, pp. 496–503, June 1975.
- [28] J. Cullum, "An Algorithm for Minimizing a Differentiable Function that Uses only Function Values", in *Techniques of Optimization*, pp. 117–127, A. V. Balakrishnan Ed., Academic Press, New York, NY, 1972.
- [29] G. Hachtel, T. R. Scott and R. P. Zug, "An Interactive Linear Programming Approach to Model Parameter Fitting and Worst Case Circuit Design", *IEEE Trans. on Circuits and Systems*, vol. CAS-27, pp. 871–881, October 1980.

- [30] W. T. Nye, *DELIGHT: An Interactive System for Optimization-Based Engineering Design*, PhD thesis, University of California at Berkeley, May 1983.
- [31] W. Nye, D. C. Riley, A. L. Sangiovanni-Vincentelli and A. L. Tits, "DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits", *IEEE Trans. on Computer Aided Design*, vol. CAD-7, n. 4, pp. 501–519, April 1988.
- [32] J. M. Shyu and A. L. Sangiovanni-Vincentelli, "ECSTASY: A new Environment for IC Design Optimization", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 484–487, November 1988.
- [33] J. B. Shyu, G. C. Temes and F. Krummenacher, "Random Errors Effects in Matched MOS capacitors and Current Sources", *IEEE Journal of Solid State Circuits*, vol. SC-19, pp. 948–955, 1984.
- [34] C. H. Stapper, "Modeling of Defects in Integrated Circuit Photolithographic Patterns", *IBM Journal of Research and Development*, vol. 28, n. 4, pp. 461–475, July 1984.
- [35] E. T. Lewis, "An Analysis of Interconnect Line Capacitance and Coupling for VLSI Circuits", *Solid-State Electronics*, vol. 27, pp. 741–749, 1984.
- [36] T. A. Johnson, R. W. Knepper, V. Marcello and W. Wang, "Chip Substrate Resistance Modeling Technique for Integrated Circuit Design", *IEEE Trans. on Computer Aided Design*, vol. CAD-3, pp. 126–134, 1984.
- [37] B. J. Hosticka, "Performance Comparison of Analog and Digital Circuits", *Proc. of the IEEE*, vol. 73, n. 1, pp. 25–29, 1985.
- [38] P. E. Allen and E. R. Macaluso, "AIDE2: An Automated Analog IC Design System", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 498–501, May 1985.
- [39] P. E. Allen and H. Nevarez-Lozano, "Automated Design of MOS Op Amps", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 1286–1289, 1983.
- [40] S. K. Hong and P. E. Allen, "Performance Driven Analog Layout Compiler", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 835–838, 1990.
- [41] D. E. Knuth, "Fundamental Algorithms", in *The Art of Computer Programming*, volume 1, Addison Wesley, Reading, MA, 1973.

- [42] H. Shin and A. L. Sangiovanni-Vincentelli, "A Detailed Router Based on Incremental Routing Modifications: Mighty", *IEEE Trans. on Computer Aided Design*, vol. CAD-6, n. 6, pp. 942-955, November 1987.
- [43] R. J. Bowman and D. J. Lane, "A Knowledge-Based System for Analog Integrated Circuit Design", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 210-212, 1985.
- [44] A. L. Ressler, *A Circuit Grammar for Operational Amplifier Design*, PhD thesis, Massachusetts Institute on Technology, 1984.
- [45] T. Tanaka, "Parsing Electronic Circuits in a Logic Grammar", *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, n. 2, pp. 225-39, April 1993.
- [46] U. Gatti, V. Liberali, F. Maloberti and A. Scianna, "An Expert System for the Design of Analog Building Blocks", in *Proc. 10th International Workshop on Expert Systems and their Applications, Avignon*, pp. 75-86, May 1990.
- [47] J. Trontelj, L. Trontelj, A. Pleteršek, A. Vodopivec and G. Shenton, "Synthesis and Layout Compilation Automation of Mixed Analog-Digital ASICs", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 816-819, May 1990.
- [48] F. El-Turky and E. E. Perry, "BLADES: An A.I. Approach to Analog Circuit Design", *IEEE Trans. on Computer Aided Design*, vol. CAD-8, n. 6, pp. 680-692, June 1989.
- [49] A. E. Dunlop, G. F. Gross, C. D. Kimble, M. Y. Luong, K. J. Stern and E. J. Swanson, "Features in LTX2 for Analog Layout", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 21-24, 1985.
- [50] C. D. Kimble, A. E. Dunlop, G. F. Gross, V. L. Hein, M. Y. Luong, K. J. Stern and E. J. Swanson, "Autorouted Analog VLSI", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 72-78, May 1985.
- [51] T. Ohtsuki, *Layout Design and Verification*, T. Ohtsuki Ed., North Holland, 1986.
- [52] A. L. Sangiovanni-Vincentelli, "Automatic Layout of Integrated Circuits", in *Design Systems for VLSI Circuits*, pp. 113-195, De Micheli et al. Eds., Martinus Nijhoff, 1987.
- [53] B. W. Kernighan and S. Lin, "An Efficient Heuristics Procedure for Partitioning Graphs", *Bell System Technical Journal*, vol. 49, n. 2, pp. 291-307, 1970.

- [54] A. E. Dunlop and B. W. Kernighan, "A Procedure for Placement of Standard-Cell VLSI Circuits", *IEEE Trans. on Computer Aided Design*, vol. CAD-4, n. 1, pp. 92–98, January 1985.
- [55] A. E. Dunlop, V. D. Agrawal, D. N. Deutsch, M. F. Jukl, P. Kozak and M. Wiesel, "Chip Layout Optimization Using Critical Path Weighting", in *Proc. IEEE/ACM Design Automation Conference*, pp. 133–136, June 1984.
- [56] M. G. R. Degrauwe, et al., "IDAC: An Interactive Design Tool for Analog CMOS Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-22, n. 6, pp. 1106–1116, December 1987.
- [57] J. Rijmenants, J. B. Litsios, T. R. Schwarz and M. G. R. Degrauwe, "ILAC: An Automated Layout Tool for Analog CMOS Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-24, n. 2, pp. 417–425, April 1989.
- [58] R. H. J. M. Otten, "Automatic Floorplan Design", in *Proc. IEEE/ACM Design Automation Conference*, pp. 261–267, June 1982.
- [59] H. Y. Koh, C. H. Séquin and P. R. Gray, "OPASYN: A Compiler for CMOS Operational Amplifiers", *IEEE Trans. on Computer Aided Design*, vol. CAD-9, n. 2, pp. 113–125, February 1990.
- [60] H. Y. Koh, C. H. Séquin and P. R. Gray, "Automatic Layout Generation for CMOS Operational Amplifiers", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 548–551, November 1988.
- [61] H. Onodera, H. Kanbara and K. Tamaru, "Operational-Amplifier Compilation with Performance Optimization", *IEEE Journal of Solid State Circuits*, vol. SC-25, n. 2, pp. 466–473, April 1990.
- [62] R. Harjani, R. A. Rutenbar and L. R. Carley, "OASYS: A Framework for Analog Circuit Synthesis", *IEEE Trans. on Computer Aided Design*, vol. CAD-8, n. 12, pp. 1247–1266, December 1989.
- [63] J. M. Cohn, D. J. Garrod, R. A. Rutenbar and L. R. Carley, "New Algorithms for Placement and Routing of Custom Analog Cells in ACACIA", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 2761–2765, May 1990.

- [64] E. Berkcan, M. d'Abreu and W. Laughton, "Analog Compilation Based on Successive Decompositions", in *Proc. IEEE/ACM Design Automation Conference*, pp. 369–375, June 1988.
- [65] E. Ochotta, L. R. Carley and R. A. Rutenbar, "Analog Circuit Synthesis for Large, Realistic Cells: Designing a Pipelined A/D Converter with ASTRX/OBLX", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 365–368, May 1994.
- [66] S. Kirkpatrick, C. Gelatt and M. Vecchi, "Optimization by Simulated Annealing", *Science*, vol. 220, n. 4598, pp. 671–680, May 1983.
- [67] J. M. Cohn, D. J. Garrod, R. A. Rutenbar and L. R. Carley, "KOAN/ANAGRAM II: New Tools for Device-Level Analog Placement and Routing", *IEEE Journal of Solid State Circuits*, vol. SC-26, n. 3, pp. 330–342, March 1991.
- [68] R. Okuda, T. Sato, H. Onodera and K. Tamaru, "An Efficient Algorithm for Layout Compaction Problem with Symmetry Constraints", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 148–151, November 1989.
- [69] E. Felt, E. Malavasi, E. Charbon, R. Totaro and A. L. Sangiovanni-Vincentelli, "Performance-Driven Compaction for Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 1731–1735, May 1993.
- [70] B. R. Stanasic, N. K. Verghese, D. J. Allstot, R. A. Rutenbar and L. R. Carley, "Addressing Substrate Coupling in Mixed-Mode ICs: Simulation and Power Distribution Synthesis", *IEEE Journal of Solid State Circuits*, vol. SC-29, n. 3, pp. 226–237, March 1994.
- [71] S. Mitra, S. Nag, R. A. Rutenbar and L. R. Carley, "System-level Routing of Mixed-Signal ASICs in WREN", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 394–399, November 1992.
- [72] U. Choudhury and A. L. Sangiovanni-Vincentelli, "Constraint-Based Channel Routing for Analog and Mixed-Analog Digital Circuits", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 198–201, November 1990.
- [73] H. Yaghtiel, A. L. Sangiovanni-Vincentelli and P. R. Gray, "A Methodology for Automated Layout of Switched-Capacitor Filters", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 444–447, 1986.

- [74] L. Trontelj, J. Trontelj, T. Slivnik Jr., R. Sosic and D. Lucas, "Analog Silicon Compiler for Switched Capacitor Filters", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 506–509, 1987.
- [75] G. Winner, A. Nguyen and C. Slemaker, "Analog Macrocell Assembler", *VLSI Systems Design*, vol. 8, n. 5, pp. 68–71, May 1987.
- [76] G. V. Eaton, D. G. Nairn, W. M. Snelgrove and A. S. Sedra, "SICOMP: A Silicon Compiler for SC Filters", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 321–324, 1987.
- [77] R. P. Sigg, A. Kaelin, A. Muralt, W. C. Black Jr. and G. S. Moschytz, "An SC Filter Compiler: Fully Automated Filter Synthesizer and Mask Generator for a CMOS Gate-Array-Type Filter Chip", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 510–513, 1987.
- [78] J. Assael, P. Senn and M. S. Tawfik, "A Switched-Capacitor Filter Silicon Compiler", *IEEE Journal of Solid State Circuits*, vol. SC-23, n. 1, pp. 166–174, February 1988.
- [79] M. Negahban and D. Gaiski, "Silicon Compilation of Switched-Capacitor Networks", in *Proc. EDAC*, pp. 164–168, 1990.
- [80] O. Buset, M. Declercq, F. Rahali and P. Vaucher, "Fast Prototyping of Semi-Custom Bipolar Analog ASICs", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 561–564, May 1991.
- [81] D. G. Maeding, et al., "Combining Analog and Digital Standard Cells", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 491–494, May 1985.
- [82] R. S. Gyurcsik and J.-C. Jeen, "A Generalized Approach to Routing Mixed Analog and Digital Signal Nets in a Channel", *IEEE Journal of Solid State Circuits*, vol. SC-24, n. 2, pp. 436–442, April 1989.
- [83] Z.-M. Lin, "DAVE: an Automatic Mixed Analog/Digital IC Layout Compiler", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 541–544, May 1991.
- [84] W. J. Helms and B. E. Byrkett, "Compiler Generation of A to D Converters", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 161–164, May 1987.

- [85] E. Berkcan, "MxSICO: a Silicon Compiler for Mixed Analog Digital Circuits", in *Proc. IEEE International Conference on Computer Design*, pp. 33–36, September 1990.
- [86] A. Barlow, K. Takasuka, Y. Nambu, T. Adachi and J. I. Konno, "An Integrated Switched-Capacitor Filter Design System", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 451–455, May 1989.
- [87] G. Jusuf, P. R. Gray and A. L. Sangiovanni-Vincentelli, "CADICS - Cyclic Analog-To-Digital Converter Synthesis", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 286–289, November 1990.
- [88] J. Vital, N. C. Horta, N. S. Silva and J. E. Franca, "CATALYST: A Highly Flexible CAD Tool for Architecture-Level Design and Analysis of Data Converters", in *Proc. European Design Automation Conference*, pp. 472–477, September 1993.
- [89] P. E. Allen, "A Tutorial - Computed Aided Design of Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 608–616, 1986.
- [90] E. Berkcan and F. Yassa, "Towards Mixed Analog/Digital Design Automation: A Review", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 809–815, May 1990.
- [91] G. Gielen and J. E. da Franca, "Computer-Aided Design Tools for Data Converters - Overview", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2140–2143, May 1992.
- [92] M. Kayal, S. Piguet, M. Declercq and B. Hochet, "SALIM: A Layout Generator Tool for Analog ICs", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 751–754, May 1988.
- [93] S. Piguet, F. Rahali, M. Kayal, E. Zysman and M. Declercq, "A new Routing Method for Full Custom Analog IC's", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 2771–2774, May 1990.
- [94] P. Harvey J, M. I. Elmasry and B. Leung, "STAIC: An Interactive Framework for Synthesizing CMOS and BiCMOS Analog Circuits", *IEEE Trans. on Computer Aided Design*, vol. CAD-11, n. 11, pp. 1402–1417, November 1992.

- [95] P. A. D. Powell and M. I. Elmasry, "The ICEWATER Language and Interpreter", in *Proc. IEEE/ACM Design Automation Conference*, pp. 98–102, June 1984.
- [96] B. J. Sheu, A. H. Fung and Y. N. Lai, "A Knowledge-Based Approach to Analog Integrated Circuit Design", *IEEE Trans. on Circuits and Systems*, vol. CAS-35, n. 2, pp. 256–258, February 1988.
- [97] A. H. Fung, D. J. Chen, Y. N. Lai and B. J. Sheu, "Knowledge-Based Analog Circuit Synthesis with Flexible Architecture", in *Proc. IEEE International Conference on Computer Design*, pp. 48–51, October 1988.
- [98] D. J. Chen, J. C. Lee and B. J. Sheu, "SLAM: A Smart Analog Module Generator for Mixed Analog-Digital VLSI Design", in *Proc. IEEE International Conference on Computer Design*, pp. 24–27, 1989.
- [99] D. J. Chen and B. J. Sheu, "Automatic Custom Layout of Analog IC's using Constraint-Based Module Generation", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 551–554, May 1991.
- [100] J. Kuhn, "Analog Module Generators for Silicon Compilation", *VLSI Systems Design*, vol. 8, n. 5, pp. 74–80, May 1987.
- [101] V. M. zu Bexten, C. Moraga, R. Klinke, W. Brockherde and K.-G. Hess, "ALSYN: Flexible Rule-Based Layout Synthesis for Analog IC's", *IEEE Journal of Solid State Circuits*, vol. SC-28, n. 3, pp. 261–268, March 1993.
- [102] Z.-Q. Ning, T. Mouthaan and H. Wallinga, "SEAS: A Simulated Evolution Approach for Analog Circuit Synthesis", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 521–524, May 1991.
- [103] J. D. Conway and G. G. Schrooten, "An Automatic Layout Generator for Analog Circuits", in *Proc. EDAC*, pp. 513–519, March 1992.
- [104] U. Lauther, "A Min-Cut Placement Algorithm for General Cell Assemblies Based on a Graph Representation", in *Proc. IEEE/ACM Design Automation Conference*, pp. 1–10, 1979.
- [105] M. Mogaki, N. Katoh, Y. Chikami, N. Yamada and Y. Kobayashi, "LADIES: An Automatic Layout System for Analog LSI's", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 450–453, November 1989.

- [106] Y. Shiraishi, M. Kimura, K. Kobayashi, T. Hino, M. Seriuchi and M. Kusaoke, "A High-Packing Density Module Generator for Bipolar Analog LSIs", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 194–197, November 1990.
- [107] Y. Shiraishi, J. Sakemi and M. Kuzuwada, "A High-Packing Density Module Generator for CMOS Logic Cells", in *Proc. IEEE/ACM Design Automation Conference*, pp. 439–445, June 1988.
- [108] T. Kozawa, et al., "Combine and Top Down Block Placement Algorithm for Hierarchical Logic VLSI Layout", in *Proc. IEEE/ACM Design Automation Conference*, June 1984.
- [109] M. Itoh and H. Mori, "ALE: a Layout Generating and Editing System for Analog LSIs", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 843–846, May 1990.
- [110] C. Toumazou and C. A. Makris, "Analog IC Design: Part I - Automated Circuit Generation: New Concepts and Methods", *IEEE Trans. on Computer Aided Design*, vol. CAD-14, n. 2, pp. 218–238, February 1995.
- [111] C. A. Makris and C. Toumazou, "Analog IC Design: Part II - Automated Circuit Correction by Qualitative Reasoning", *IEEE Trans. on Computer Aided Design*, vol. CAD-14, n. 2, pp. 239–254, February 1995.
- [112] C. Makris and C. Toumazou, "Qualitative Reasoning in Analog IC Design Automation", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 831–834, May 1992.
- [113] T. Koskinen and P. Y. K. Cheung, "Modelling Behaviour and Tolerances in Analogue Cells", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 871–874, May 1991.
- [114] B. C. Williams, "Qualitative Analysis of MOS Circuits", M.S. Thesis, Massachusetts Institute on Technology, 1984.
- [115] N. Gohar, P. Cheung and C. Pun, "RACHANA: An Integrated Placement and Routing Approach to CMOS Analog Cells", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2981–2984, May 1992.

- [116] U. Choudhury and A. L. Sangiovanni-Vincentelli, "Constraint Generation for Routing Analog Circuits", in *Proc. IEEE/ACM Design Automation Conference*, pp. 561–566, June 1990.
- [117] U. Choudhury and A. L. Sangiovanni-Vincentelli, "Automatic Generation of Parasitic Constraints for Performance-Constrained Physical Design of Analog Circuits", *IEEE Trans. on Computer Aided Design*, vol. CAD-12, n. 2, pp. 208–224, February 1993.
- [118] E. Malavasi, U. Choudhury and A. L. Sangiovanni-Vincentelli, "A Routing Methodology for Analog Integrated Circuits", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 202–205, November 1990.
- [119] E. Charbon, E. Malavasi, U. Choudhury, A. Casotto and A. L. Sangiovanni-Vincentelli, "A Constraint-Driven Placement Methodology for Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 2821–2824, May 1992.
- [120] D. Harrison, P. Moore, R. L. Spickelmier and A. R. Newton, "Data Management and Graphics Editing in the Berkeley Design Environment", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 24–27, November 1986.
- [121] D. S. Harrison, A. R. Newton, R. L. Spickelmier and T. J. Barnes, "Electronic CAD Frameworks", *Proc. of the IEEE*, vol. 78, n. 2, pp. 393–417, February 1990.
- [122] G. Gad-El-Karim and R. S. Gyurcsik, "Generation of Performance Sensitivities for Analog Cell Layout", in *Proc. IEEE/ACM Design Automation Conference*, pp. 500–505, June 1991.
- [123] G. Gad-El-Karim and R. S. Gyurcsik, "Use of Performance Sensitivities in Analog Cell Layout", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2008–2011, June 1991.
- [124] G. Gad-El-Karim, *Sensitivity-Driven Placement of Analog Modules*, PhD thesis, Carnegie-Mellon University, 1992.
- [125] E. Liu, A. L. Sangiovanni-Vincentelli, G. Gielen and P. Gray, "A Behavioral Representation for Nyquist Rate A/D Converters", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 386–389, November 1991.

- [126] H. Chang, A. L. Sangiovanni-Vincentelli, F. Balarin, E. Charbon, U. Choudhury, G. Jusuf, E. Liu, E. Malavasi, R. Neff and P. Gray, "A Top-down, Constraint-Driven Design Methodology for Analog Integrated Circuits", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 841–846, May 1992.
- [127] S. W. Mehranfar, "STAT: A Schematic to Artwork Translator for Custom Analog Cells", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 3021–3024, May 1990.
- [128] S. W. Mehranfar, "A Technology-Independent Approach to Custom Analog Cell Generation", *IEEE Journal of Solid State Circuits*, vol. SC-26, n. 3, pp. 386–393, March 1991.
- [129] K. Swings and W. Sansen, "ARIADNE: a Constraint-Based Approach to Computer-Aided Synthesis and Modeling of Analog Integrated Circuits", *Analog Integrated Circuits and Signal Processing*, vol. 3, n. 3, pp. 197–215, May 1993.
- [130] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Kluwer Academic Publ., Boston, MA, 1991.
- [131] S. J. Seda, M. G. R. Degrauwe and W. Fichtner, "Lazy-Expansion Symbolic Expression Approximation in SYNAP", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 310–317, November 1992.
- [132] I. Harada, H. Kitazawa and T. Kaneko, "A Routing System for Mixed A/D Standard Cell LSI's", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 378–381, November 1990.
- [133] I. Harada, H. Kitazawa and T. Kaneko, "A Layout System for Mixed A/D Standard Cell LSI's", *IEICE Trans. on Electronics*, vol. E75-C, n. 3, pp. 322–332, March 1992.
- [134] T. Ohtsuka, H. Kunieda and M. Kaneko, "LIBRA: Automatic Performance-Driven Layout for Analog LSIs", *IEICE Trans. on Electronics*, vol. E75-C, n. 3, pp. 312–321, March 1992.
- [135] J.-M. Shyu, *Performance Optimization of Integrated Circuits*, PhD thesis, University of California at Berkeley, November 1988.

- [136] S. W. Director and R. A. Rohrer, "The Generalized Adjoint Network and Network Sensitivities", *IEEE Trans. on Circuit Theory*, vol. CT-16, pp. 318–323, August 1969.
- [137] U. Choudhury, "Sensitivity Computation in SPICE3", M.S. Thesis, University of California at Berkeley, 1988.
- [138] D. A. Hocevar, P. Yang, T. N. Trick and B. D. Epler, "Transient Sensitivity Computation for MOSFET Circuits", *IEEE Trans. on Computer Aided Design*, vol. CAD-4, n. 4, pp. 609–620, October 1985.
- [139] E. Charbon, R. Gharpurey, R. G. Meyer and A. L. Sangiovanni-Vincentelli, "Semi-Analytical Techniques for Substrate Characterization in the Design of Mixed-Signal ICs", in *Proc. IEEE International Conference on Computer Aided Design*, to appear, November 1996.
- [140] D. J. Allstot and Jr. W. C. Black, "Technological Considerations for Monolithic MOS Switched-Capacitor Filtering Systems", in *Proc. of the IEEE*, pp. 967–986, 1983.
- [141] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching Properties of MOS Transistors", *IEEE Journal of Solid State Circuits*, vol. SC-24, pp. 1433–1440, October 1989.
- [142] B. Basaran, R. A. Rutenbar and L. R. Carley, "Latchup-Aware Placement and Parasitic-Bounded Routing of Custom Analog Cells", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 415–421, November 1993.
- [143] M. Hanan and J. M. Kurtzberg, "Placement Techniques", in *Design Automation of Digital Systems*, volume 1, pp. 213–282, M. A. Breuer Ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1972.
- [144] L.-O. Donzelle, P. F. Dubois, B. Hennion, J. Parissis and P. Senn, "A Constraint Based Approach to Automatic Design of Analog Cells", in *Proc. IEEE/ACM Design Automation Conference*, pp. 506–509, June 1991.
- [145] N. R. Quinn and M. A. Breuer, "A Force Directed Component Placement Procedure for Printed Circuit Boards", *IEEE Trans. on Circuits and Systems*, vol. CAS-26, pp. 377–388, 1979.

- [146] E. Bar-Yehuda, J. A. Feldman, R. Y. Pinter and S. Wimer, "Depth-First-Search and Dynamic Programming Algorithms for Efficient CMOS Cell Generation", *IEEE Trans. on Computer Aided Design*, vol. CAD-8, n. 7, pp. 737-743, July 1989.
- [147] E. Malavasi and D. Pandini, "Optimum CMOS Stack Generation with Analog Constraints", *IEEE Trans. on Computer Aided Design*, vol. CAD-14, n. 1, pp. 107-122, January 1995.
- [148] H. Onodera and K. Tamaru, "Analog Circuit Placement-Branch-And-Bound Placement with Shape Optimization", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 1151-1156, May 1992.
- [149] M. A. Breuer, "A Class of Min-Cut Algorithms", in *Proc. IEEE/ACM Design Automation Conference*, pp. 284-290, June 1977.
- [150] L. I. Corrigan, "A Placement Capability Based on Partitioning", in *Proc. IEEE/ACM Design Automation Conference*, pp. 406-413, June 1979.
- [151] M. Sriram and S. M. Kang, "A Modified Hopfield Network for Two-Dimensional Module Placement", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 1664-1667, May 1990.
- [152] J. P. Blanks, "Near-Optimal Placement Using Quadratic Objective Function", in *Proc. IEEE/ACM Design Automation Conference*, pp. 609-615, June 1985.
- [153] C. K. Cheng and E. S. Kuh, "Module Placement Based on Resistive Network Optimization", *IEEE Trans. on Computer Aided Design*, vol. CAD-3, n. 3, pp. 218-225, July 1984.
- [154] N. R. Quinn, "The Placement Problem as Viewed from the Physics of Classical Mechanics", in *Proc. IEEE/ACM Design Automation Conference*, pp. 173-178, June 1975.
- [155] L. Sha and R. W. Dutton, "An Analytical Algorithm for Placement of Arbitrary Sized Rectangular Blocks", in *Proc. IEEE/ACM Design Automation Conference*, pp. 602-607, June 1985.
- [156] J. M. Kleinhans, G. Sigl, F. M. Johannes and K. J. Antreich, "GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization", *IEEE Trans. on Computer Aided Design*, vol. CAD-10, n. 3, pp. 356-365, March 1991.

- [157] A. Srinivasan, K. Chaudhary and E. S. Kuh, "RITUAL: A Performance Driven Placement Algorithm", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, n. 11, pp. 825-840, November 1992.
- [158] H. Vaishnav and M. Pedram, "Performance Driven Placement Algorithm for Low Power Designs", in *Proc. European Design Automation Conference*, pp. 72-77, September 1993.
- [159] T. Abthoff and F. Johannes, "Analogue Placement Using Guided Enumeration", *International Journal of Circuit Theory and Applications, Special Issue on "Analog Tools for Circuit Design"*, John Wiley & Sons, vol. 23, n. 4, pp. 453-473, July-August 1995.
- [160] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison Wesley, Reading, MA, 1989.
- [161] R. G. Garside and T. A. J. Nicholson, "Permutation Procedure for the Backboard-Wiring Problem", *Proc. of the IEEE*, vol. 115, n. 1, pp. 27-30, January 1967.
- [162] M. Hanan, P. K. Wolff Sr. and B. J. Agule, "Some Experimental Results on Placement Techniques", in *Proc. IEEE/ACM Design Automation Conference*, pp. 214-224, June 1976.
- [163] A. Iosupovici, C. King and M. A. Breuer, "A Model Interchange Placement Machine", in *Proc. IEEE/ACM Design Automation Conference*, pp. 171-174, June 1983.
- [164] J. P. Cohoon and W. D. Paris, "Genetic Placement", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 422-425, November 1986.
- [165] R. M. Kling and P. Banerjee, "ESP: A new Standard Cell Placement Package Using Simulated Evolution", in *Proc. IEEE/ACM Design Automation Conference*, pp. 60-66, September 1987.
- [166] C. M. Kyung, P. V. Kraus and D. A. Mlynski, "Diffusion - an Analytic Procedure Applied to Macro Cell Placement", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 102-105, November 1990.
- [167] J. Scheible and D. A. Mlynski, "A High Density Placement Algorithm Based on Simulated Surface Tension", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2048-2051, June 1991.

- [168] C. X. Zhang, A. Vogt and D. A. Mlynski, "Floorplan Design Using a Hierarchical Neural Learning Algorithm", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 2060–2063, June 1991.
- [169] C. Sechen and K.-W. Lee, "An Improved Simulated Annealing Algorithm for Row-Based Placement", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 478–481, 1987.
- [170] A. Casotto, F. Romeo and A. L. Sangiovanni-Vincentelli, "A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells", *IEEE Trans. on Computer Aided Design*, vol. CAD-6, n. 5, pp. 838–847, September 1987.
- [171] D. F. Wong, H. W. Leong and C. L. Liu, "PLA Folding by Simulated Annealing", *IEEE Journal of Solid State Circuits*, vol. SC-22, n. 2, pp. 208–215, April 1987.
- [172] R. R. Troutman, "Latchup in CMOS Technology", in *Layout Design and Verification*, Kluwer Academic Publ., Boston, MA, 1986.
- [173] J. M. Cohn, D. J. Garrod, R. A. Rutenbar and L. R. Carley, "Techniques for Simultaneous Placement and Routing of Custom Analog Cells in KOAN/ANAGRAM II", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 394–397, November 1991.
- [174] S. Mitra, R. A. Rutenbar, L. R. Carley and D. J. Allstot, "Substrate-Aware Mixed-Signal Macro-Cell Placement in WRIGHT", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 529–532, May 1994.
- [175] E. Charbon, E. Malavasi, D. Pandini and A. L. Sangiovanni-Vincentelli, "Imposing Tight Specifications on Analog IC's through Simultaneous Placement and Module Optimization", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 525–528, May 1994.
- [176] E. Charbon, E. Malavasi, D. Pandini and A. L. Sangiovanni-Vincentelli, "Simultaneous Placement and Module Optimization of Analog IC's", in *Proc. IEEE/ACM Design Automation Conference*, pp. 31–35, June 1994.
- [177] R. H. J. M. Otten and L. P. P. P. van Ginneken, "Floorplan Design using Simulated Annealing", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 96–98, November 1984.

- [178] D. F. Wong, H. W. Leong and C. L. Liu, "A new Algorithm for Floorplan Design", in *Proc. IEEE/ACM Design Automation Conference*, pp. 101–107, 1986.
- [179] D. W. Jepsen and C. D. Gelatt, "Macro Placement by Monte Carlo Annealing", in *Proc. IEEE International Conference on Computer Design*, pp. 495–498, November 1984.
- [180] C. Sechen and A. L. Sangiovanni-Vincentelli, "Chip-Planning, Placement and Global Routing of Macro/Custom Cell IC's using Simulated Annealing", in *Proc. IEEE/ACM Design Automation Conference*, pp. 73–80, June 1988.
- [181] F. Romeo, *Simulated Annealing: Theory and Applications to Layout Problems*, PhD thesis, University of California at Berkeley, March 1989.
- [182] S. Sriram, "An Adaptive Cooling Schedule for PUPPY", Internal report, University of California at Berkeley, 1989.
- [183] L. P. P. P. van Ginneken and R. H. J. M. Otten, "Optimal Slicing of Plane Point Placements", in *Proc. European Design Automation Conference*, pp. 322–326, September 1990.
- [184] R. H. J. M. Otten and L. P. P. P. van Ginneken, "The Complexity of Adaptive Annealing", in *Proc. IEEE International Conference on Computer Design*, pp. 404–407, September 1990.
- [185] D. Marple, M. Smulders and H. Hegen, "Tailor: A Layout System Based on Trapezoidal Corner Stitching", *IEEE Trans. on Computer Aided Design*, vol. CAD-9, n. 1, pp. 66–90, January 1990.
- [186] E. Malavasi, E. Charbon, G. Jusuf, R. Totaro and A. L. Sangiovanni-Vincentelli, "Virtual Symmetry Axes for the Layout of Analog IC's", in *Proc. 2nd ICVC, Seoul, Korea*, pp. 195–198, October 1991.
- [187] G. Sorkin, *Theory and Practice of Simulated Annealing on Special Energy Landscapes*, PhD thesis, University of California at Berkeley, 1991.
- [188] E. Malavasi, D. Pandini and V. Liberali, "Optimum Stacked Layout for Analog CMOS ICs", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 1711–1714, May 1993.

- [189] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, Cambridge, MA, 1991.
- [190] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, 1979.
- [191] S. Wimer, R. Y. Pinter and J. A. Feldman, "Optimal Chaining of CMOS Transistors in a Functional Cell", *IEEE Trans. on Computer Aided Design*, vol. CAD-6, n. 5, pp. 795–801, September 1987.
- [192] B. Hajek, "Cooling Schedules for Optimal Annealing", *Mathematics of Operations Research*, vol. 13, n. 2, pp. 311–329, May 1988.
- [193] U. Choudhury and A. L. Sangiovanni-Vincentelli, "An Analytical-Model Generator for Interconnect Capacitances", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 861–864, May 1991.
- [194] B. R. Staniscic, R. A. Rutenbar and L. R. Carley, "Mixed-Signal Noise-Decoupling via Simultaneous Power Distribution and Cell Customization in RAIL", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 533–536, May 1994.
- [195] I. L. Wemple and A. T. Yang, "Mixed-Signal Switching Noise Analysis Using Voronoi-Tessellated Substrate Macromodels", in *Proc. IEEE/ACM Design Automation Conference*, pp. 439–444, June 1995.
- [196] D. K. Su, M. Loinaz, S. Masui and B. Wooley, "Experimental Results and Modeling Techniques for Substrate Noise in Mixed-Signal Integrated Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-28, n. 4, pp. 420–430, April 1993.
- [197] K. Joardar, "A Simple Approach to Modeling Cross-Talk in Integrated Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-29, n. 10, pp. 1212–1219, October 1994.
- [198] J. Rijmenants, et al., "ILAC: An Automated Layout Tool for Analog CMOS Circuits", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 761–764, May 1988.
- [199] J.-C. Jeen, R. S. Gyurcsik and W.-T. Liu, "A Two-Layer Channel Routing Algorithm for Mixed Analog and Digital Signal Nets", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 1151–1154, May 1988.

- [200] T. Ohtsuki, "Maze-Running and Line-Search Algorithms", in *Layout Design and Verification*, ch. 3, pp. 99–131, T. Ohtsuki Ed., North Holland, 1986.
- [201] E. Malavasi and A. L. Sangiovanni-Vincentelli, "Area Routing for Analog Layout", *IEEE Trans. on Computer Aided Design*, vol. CAD-12, n. 8, pp. 1186–1197, August 1993.
- [202] E. Malavasi and A. L. Sangiovanni-Vincentelli, "Dynamic Bound Generation for Constraint-Driven Routing", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 477–480, May 1995.
- [203] C. Lee, "An Algorithm for Path Connections and Applications", *IRE Trans. Electron. Computer*, vol. EC-10, pp. 346–365, September 1961.
- [204] N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, 1971.
- [205] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison Wesley, Reading, MA, 1984.
- [206] R. C. Prim, "Shortest Connection Networks and Some Generalizations", *Bell System Tech. Journal*, vol. 36, pp. 1389–1401, 1957.
- [207] R. H. Jansen, R. G. Arnonld and I. G. Eddison, "A Comprehensive CAD Approach to Design of MMICs up to MM-Wave Frequencies", *IEEE Trans. on Microwave Theory and Techniques*, vol. MTT-36, n. 2, pp. 208–219, February 1988.
- [208] W. L. Williams R. C. Compton and B. Rutledge, "Puff, an interactive Microwave Computer Aided Design Program for Personal Computers", in *Proc. IEEE Microwave Theory and Techniques Symposium*, pp. 707–708, 1987.
- [209] R. H. Jansen, "LINMIC: A CAD Package for the Layout-Oriented Design of Single- and Multi-Layer MICs/MMICs up to mm-Wave Frequencies", *"Microwave Journal"*, pp. 151–161, February 1986.
- [210] Santa Rosa Systems Division, Santa Rosa, CA, *Hewlett-Packard Microwave and RF Design System (MDS) Manuals*, December 1992.
- [211] J. F. Zürcher, "MICROS - A CAD/CAM Program for Fast Realization of Microstrip Masks", in *Proc. IEEE Microwave Theory and Techniques Symposium*, pp. 481–484, 1985.

- [212] M. E. Goldfarb and A. Platzker, "Losses in GaAs Microstrip", in *Proc. IEEE Microwave Theory and Techniques Symposium*, pp. 563–565, 1990.
- [213] J. R. James and A. Henderson, "High-Frequency Behaviour of Microstrip Open-Circuit Terminations", *IEE Journal on Microwaves, Optics and Acoustics*, vol. 3, n. 5, pp. 205–218, Sept 1979.
- [214] G. W. Clow, "A Global Routing Algorithm for General Cells", in *Proc. IEEE/ACM Design Automation Conference*, pp. 45–51, 1984.
- [215] A. Mlynsky and C.-H. Sung, "Layout Compaction", in *Layout Design and Verification*, ch. 6, pp. 199–235, T.Ohtsuki Ed., North Holland, 1986.
- [216] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Applicable Theory in Computer Science, J. Wiley & Sons, New York, 1990.
- [217] G. Kedem and H. Watanabe, "Graph-Optimization Techniques for IC Layout and Compaction", *IEEE Trans. on Computer Aided Design*, vol. CAD-3, n. 1, pp. 12–20, January 1984.
- [218] H. Shin, A. L. Sangiovanni-Vincentelli and C. Sequin, "Two-Dimensional Compaction by Zone-Refining", in *Proc. IEEE/ACM Design Automation Conference*, pp. 115–122, June 1986.
- [219] W. Schiele, "Improved Compaction by Minimized Length of Wires", in *Proc. IEEE/ACM Design Automation Conference*, pp. 121–127, 1983.
- [220] W. H. Crocker, R. Varadarajan and C. Y. Lo, "Compaction with Performance Optimization", in *Proc. IEEE International Symposium on Circuits and Systems*, pp. 514–517, 1987.
- [221] E. Malavasi, E. Felt, E. Charbon and A. L. Sangiovanni-Vincentelli, "Symbolic Compaction with Analog Constraints", *International Journal of Circuit Theory and Applications, Special Issue on "Analog Tools for Circuit Design"*, John Wiley & Sons, vol. 23, n. 4, pp. 433–452, July-August 1995.
- [222] J. R. Burns and A. R. Newton, "SPARCS: A New Constraint-Based IC Symbolic Layout Spacer", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 534–539, 1986.

- [223] L. Rijnders, P. Six and H. J. DeMan, "Design of a Process-Tolerant Cell Library for Regular Structures Using Symbolic Layout and Hierarchical Compaction", *IEEE Journal of Solid State Circuits*, vol. SC-23, n. 3, pp. 714–721, June 1988.
- [224] J. L. Burns, *Techniques for IC Symbolic Layout and Compaction*, PhD thesis, University of California at Berkeley, November 1990.
- [225] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization. Algorithms and Complexity*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.
- [226] A. E. Ruehli and P. A. Brennan, "Efficient Capacitance Calculations for three-Dimensional Multiconductor Systems", in *IEEE Trans. on Microwave Theory and Techniques*, volume 21, pp. 76–82, February 1973.
- [227] A. E. Ruehli, "Survey of Computer-Aided Electrical Analysis of Integrated Circuit Interconnections", *IBM Journal of Research and Development*, vol. 23, n. 6, pp. 626–639, November 1979.
- [228] P. A. Brennan, N. Raver and A. E. Ruehli, "Three Dimensional Inductance Computations with Partial Element Equivalent Circuits", *IBM*, vol. 23, pp. 661–668, November 1979.
- [229] M. Kamon, M. Tsuk, C. Smithhisler and J. White, "Efficient Techniques for Inductance Extraction of Complex 3-D Geometries", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 438–442, November 1992.
- [230] M. Kamon, M. T. Tsuk and J. White, "FastHenry: A Multiple-Accelerated 3-D Inductance Extraction Program", in *Proc. IEEE/ACM Design Automation Conference*, pp. 678–683, June 1993.
- [231] M. Chou, M. Kamon, K. Nabors, J. Phillips and J. White, "3-D Extraction Techniques for Signal Integrity Analysis", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 379–382, May 1995.
- [232] E. Barke, "Line-to-Ground Capacitance Calculation for VLSI: A Comparison", *IEEE Trans. on Computer Aided Design*, vol. CAD-7, n. 2, pp. 295–298, February 1988.
- [233] K. C. Gupta, R. Garg and R. Chardha, "Computer Aided Design for Microwave Circuits", Technical report, Norwood, MA: Artech House, 1981.

- [234] U. Choudhury and A. L. Sangiovanni-Vincentelli, "Automatic Generation of Analytical Models for Interconnect Capacitances", *IEEE Trans. on Computer Aided Design*, vol. CAD-14, n. 4, pp. 470–480, April 1995.
- [235] R. W. Gregor, "On the Relationship between Topography and Transistor Matching in an Analog CMOS Technology", *IEEE Trans. on Electron Devices*, vol. ED-39, n. 2, pp. 275–282, February 1992.
- [236] K. R. Lakshmi Kumar, R. A. Hadaway and M. A. Copeland, "Characterization and Modeling of Mismatch in MOS Transistors for Precision Analog Design", *IEEE Journal of Solid State Circuits*, vol. SC-21, n. 6, pp. 1057–1066, December 1986.
- [237] E. Felt, A. Narayan and A. L. Sangiovanni-Vincentelli, "Measurements and Modeling of MOS Transistor Current Mismatch in Analog ICs", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 272–277, November 1994.
- [238] C. Guardiani, A. Tomasini, J. Benkoski, M. Quarantelli and P. Gubian, "Applying a Submicron Mismatch Model to Practical IC Design", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 297–300, May 1994.
- [239] K. Nabors and J. White, "A Fast Multipole Algorithm for Capacitance Extraction of Complex 3-D Geometries", in *Proc. IEEE Custom Integrated Circuit Conference*, May 1989.
- [240] T. Edwards, *Foundations for Microstrip Circuit Design*, Wiley, West Sussex, UK, 2nd Ed. 1992.
- [241] K. C. Gupta, R. Garg and R. Chadha, *Computer Aided Design of Microwave Circuits*, Artech House Inc., Norwood, MA, 1981.
- [242] D. M. Pozar, *Microwave Engineering*, Addison-Wesley, Boston, 1990.
- [243] P. H. Xiao, E. Charbon, T. Van Duzer and S. R. Whiteley, "INDEX: An Inductance Extractor for Superconducting Circuits", in *1992 Applied Superconductivity Conference*, August 1992.
- [244] W. Chang, "Numerical Calculation of the Inductance of a Multi-Superconductor Transmission Line System", *IEEE Trans. Mag.*, vol. MAG-17, pp. 764–766, 1981.
- [245] V. Nandakumar, *Design, Fabrication, and Testing of a Josephson Shift Register*, PhD thesis, University of California at Berkeley, 1988.

- [246] T. Van Duzer and C. W. Turner, *Principles of Superconductive Devices and Circuits*, Elsevier North Holland, New York, NY, 1981.
- [247] P. H. Xiao, E. Charbon, T. Van Duzer and S. R. Whiteley, "INDEX: An Inductance Extractor for Superconducting Circuits", *IEEE Transactions on Applied Superconductivity*, vol. 3, n. 1, pt. 4, pp. 2629–2632, March 1993.
- [248] K. Fukahori and P. R. Gray, "Computer Simulation of Integrated Circuits in the Presence of Electrothermal Interactions", *IEEE Journal of Solid State Circuits*, vol. SC-11, pp. 834–846, December 1976.
- [249] S.-S. Lee and D. J. Allstot, "Electrothermal Simulation of Integrated Circuits", *IEEE Journal of Solid State Circuits*, vol. SC-28, n. 12, pp. 1283–1293, December 1993.
- [250] N. K. Verghese, D. J. Allstot and S. Masui, "Rapid Simulation of Substrate Coupling Effects in Mixed-Mode ICs", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 1831–1834, May 1993.
- [251] T. Smedes, "Substrate Resistance Extraction for Physics-Based Layout Verification", in *IEEE/PRORISC Workshop on CSSP*, pp. 101–106, March 1993.
- [252] T. Smedes, N. P. van der Meijs and A. J. van Genderen, "Extraction of Circuit Models for Substrate Cross-talk", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 199–206, November 1995.
- [253] K. J. Kerns, I. L. Wemple and A. T. Yang, "Stable and Efficient Reduction of Substrate Model Networks Using Congruence Transforms", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 207–214, November 1995.
- [254] J. A. Meijerink and H. A. van der Vorst, "An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix", *Mathematics of Computation*, vol. 31, pp. 148–162, January 1977.
- [255] R. Gharpurey, *Modeling and Analysis of Substrate Coupling in ICs*, PhD thesis, University of California at Berkeley, May 1995.
- [256] X. Huang, V. R. Raghavan and R. A. Rohrer, "AWESim: A Program for the Efficient Analysis of Linearized Circuits", in *Proc. IEEE International Conference on Computer Aided Design*, pp. 534–537, November 1990.

- [257] J. D. Jackson, *Classical Electrodynamics*, Wiley, 1975.
- [258] S. M. Sze, *Physics of Semiconductor Devices*, Wiley, New York, 1981.
- [259] C. Hu, *VLSI Electronics: Microstructure Science*, volume 18, Academic Press, New York, 1981.
- [260] R. B. Merrill, W. M. Young and K. Brehmer, "Effect of Substrate Material on Crosstalk in Mixed Analog/Digital ICs", in *Proc. IEEE International Electron Devices Meeting*, pp. 433-436, December 1994.
- [261] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, J. Wiley & Sons, New York, 1977.
- [262] E. Charbon, P. Miliozzi, E. Malavasi and A. L. Sangiovanni-Vincentelli, "Generalized Constraint Generation in the Presence of Non-Deterministic Parasitics", in *Proc. IEEE International Conference on Computer Aided Design*, to appear, November 1996.
- [263] R. G. H. Eschauzier, R. Hogerrvorst and J. H. Huijsing, "A Programmable 1.5V CMOS Class-AB Operational Amplifier with Hybrid Nested Miller Compensation for 120dB gain and 6MHz UGF", in *Proc. IEEE International Solid-State Circuits Conference*, pp. 246-247, February 1994.
- [264] R. Neff, P. Gray and A. L. Sangiovanni-Vincentelli, "A Module Generator for High Speed CMOS Current Output Digital/Analog Converters", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 481-484, May 1995.
- [265] I. Vassiliou, H. Chang, A. Demir, E. Charbon, P. Miliozzi and A. L. Sangiovanni-Vincentelli, "A Video Driver System Designed Using a Top-Down, Constraint-Driven Methodology", in *Proc. IEEE International Conference on Computer Aided Design*, to appear, November 1996.
- [266] D. Reynolds, "A 320 MHz CMOS Triple 8b DAC with On-Chip PLL and Hardware Cursor", in *Proc. IEEE International Solid-State Circuits Conference*, pp. 50-51, February 1994.
- [267] I. A. Young, J. K. Greason and K. L. Wong, "A PLL Clock Generator with 5 to 110 MHz of Lock Range for Microprocessors", *IEEE Journal of Solid State Circuits*, vol. SC-27, n. 11, pp. 1599-1607, November 1992.

- [268] J. Burns, A. Casotto, M. Igusa, F. Marron, F. Romeo, A. L. Sangiovanni-Vincentelli, C. Sechen, H. Shin, G. Srinath and H. Yaghtiel, "MOSAICO: An integrated Macro-cell Layout System", in *VLSI '87, Vancouver, Canada*, August 1987.
- [269] R. Castello and P. Gray, "A High-Performance Micropower Switched-Capacitor Filter", *IEEE Journal of Solid State Circuits*, vol. SC-20, pp. 1122-1132, December 1985.
- [270] B. Boser, K. P. Karmann, H. Martin and B. A. Wooley, "Simulating and Testing Oversampled Analog-to-Digital Converters", *IEEE Trans. on Computer Aided Design*, vol. CAD-7, n. 6, pp. 668-674, June 1988.
- [271] H. Chang, E. Felt and A. L. Sangiovanni-Vincentelli, "Top-Down, Constraint-Driven Design Methodology Based Generation of a Second Order Σ - Δ A/D Converter", in *Proc. IEEE Custom Integrated Circuit Conference*, pp. 533-536, May 1995.
- [272] E. Charbon, G. Holmlund, A. L. Sangiovanni-Vincentelli and B. Donecker, "A Performance-Driven Router for RF and Microwave Analog Circuit Design", Memorandum UCB/ERL M94/40, UCB, May 1994.
- [273] E. Felt, E. Charbon, E. Malavasi and A. L. Sangiovanni-Vincentelli, "An Efficient Methodology for Symbolic Compaction of Analog IC's with Multiple Symmetry Constraints", in *Proc. European Design Automation Conference*, pp. 148-153, September 1992.
- [274] M. V. Schneider, "Microstrip Lines for Microwave Integrated Circuits", *Bell Syst. Tech. Journal*, vol. 48, pp. 1421-1444, May/June 1969.
- [275] M. V. Getsinger, "Microstrip Dispersion Model", *IEEE Trans. on Microwave Theory and Techniques*, vol. MTT-21, n. 1, pp. 34-39, Jan 1973.
- [276] S. Ramo, J. R. Whinnery and T. Van Duzer, *Fields and Waves in Communication Electronics*, Prentice-Hall, Englewood Cliffs, N.J., 1984.
- [277] HP-EESof Labs, Santa Rosa, CA, *HFSS Manuals*, 1990.