

Design und Analyse integrierter Schaltungen mit evolutionären Algorithmen

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften

an der

Fakultät für Elektrotechnik und Informationstechnik

der

Universität Dortmund

von

Marc Thomas

Dortmund 2001

Tag der mündlichen Prüfung:

22. August 2001

Gutachter:

Prof. Dr.-Ing. K. Goser

Prof. Dr.-Ing. U. Rückert

Danksagung

Diese Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Bauelemente der Elektrotechnik der Universität Dortmund. Für die gute Zusammenarbeit möchte ich mich bei meinen Kollegen bedanken.

Herrn Prof. Dr.-Ing. K. Goser danke ich für seine Unterstützung und unsere gemeinsame Arbeit insbesondere im Sonderforschungsbereich „Computational Intelligence“.

Ich danke Herrn Prof. Dr.-Ing. U. Rückert von der Universität Paderborn für die Übernahme des Koreferates und die damit verbundene Mühe.

Herrn Prof. Dr.-Ing. K. Schumacher und Frau C. Menke-Schumacher danke ich für die organisatorische Unterstützung im Rahmen der Lehrstuhl­tätigkeit.

Christian Burwick und Jan Dienststuhl danke ich für zahlreiche, kontroverse Diskussionen bei reichlich Kaffee und Eva für die Korrektur meiner schriftlichen Arbeit.

Ein besonderer Dank gilt meiner Familie, Danielle und Jan, und meinen Verwandten und Freunden, die mich während der Zeit dieser Arbeit unterstützt haben.

Für Danielle.

Inhaltsverzeichnis

1	Einleitung	1
2	Schaltungsdesign auf Transistorebene	5
2.1	Schaltungsoptimierung	6
2.1.1	Eigenschaften des Parameterraumes	7
2.1.2	Grenzen der Optimierung	8
2.2	Existierende Ansätze	9
3	Parameteroptimierung in der Simulation	13
3.1	Grundlagen	13
3.2	Numerische Suchverfahren	16
3.2.1	Nicht lineare Programmierung	16
3.2.2	Gradientenverfahren	17
3.3	Evolutionäre Algorithmen	19
3.3.1	Operatoren evolutionärer Algorithmen	22
3.3.2	Evolutionstrategie und Genetische Algorithmen	23
3.3.3	Evolutionstrategie	23

4	Automatisiertes Schaltungsdesign	29
4.1	Designspezifikation	29
4.1.1	Beispiel: 1-Bit Volladdierer	32
4.2	Realisierung mit skalaren Optimierungsverfahren	35
4.2.1	Iterative Gewichtsbestimmung	39
4.2.2	Bauteilmodellierung mit iterativer Gewichtsbestimmung	42
4.3	Untersuchung verschiedener Schaltungskonzepte	47
4.3.1	Boolesche und Schwellwertlogik	48
4.3.2	Das Majoritätsgatter	49
4.3.3	Spiegeladdierer	52
4.3.4	Addierer mit Majoritätsgattern	54
4.3.5	Ergebnisse	55
5	Vektoroptimierung	63
5.1	Grundlagen und Einführung	63
5.2	Klassische Ansätze zur Bestimmung der Pareto Menge	67
5.2.1	Eigenschaften der Pareto Menge	67
5.2.2	Gewichtsmethode	69
5.2.3	Beschränkungsmethode	70
5.3	Bestimmung der Pareto Menge mit evolutionären Algorithmen	71
5.3.1	Räuber-Beute Modell	72
5.3.2	Strength Pareto Evolutionary Algorithm (SPEA)	73
5.4	Beschleunigung durch Parallelisierung	76
5.4.1	Asynchrones Räuber-Beute Modell	79
5.4.2	Asynchrone Variante des Strength Pareto Evolutionary Algorithm (asyncSPEA)	79
5.5	Vergleich der Verfahren	82

6	Verhaltensanalyse integrierter Schaltungen	85
6.1	Bestimmung von Parameterabhängigkeiten	86
6.2	Handhabung nicht monotoner Zusammenhänge	88
6.3	Extraktion analytischer Zusammenhänge	91
6.4	Anwendungsbeispiele	95
6.5	Bewertung der Ergebnisse und Ausblick	105
7	Zusammenfassung	109

Tabellenverzeichnis

3.1	Aspekte der simulationsbasierten Optimierung.	15
4.1	Wahrheitstabelle des 1-Bit Volladdierers.	33
4.2	Modell der RTD als I-V-Kurve.	43
4.3	Parameterbegrenzung des Modells der RTD.	44
4.4	Wahrheitstabelle für grundlegende Boolesche Operatoren.	48
4.5	Transistorgeometrien für die verwendeten Technologien.	57
4.6	Ergebnisse des Majoritätsgatters nach Lee.	58
4.7	Ergebnisse des Spiegeladdierers.	60
5.1	Algorithmus des Strength Pareto Evolutionary Algorithm.	74
5.2	Rechenleistung verschiedener Konfigurationen.	81
6.1	Grenzwerte der Verlustleistung und der Verzögerungszeit.	96

Abbildungsverzeichnis

1.1	Ziele der verschiedenen Optimierungsansätze.	3
2.1	Y-Diagramm.	6
2.2	Begrenzte Güte der simulationsbasierten Optimierung.	8
3.1	Gradientenverfahren.	17
3.2	Schema eines evolutionären Algorithmus.	20
3.3	Beispiel zur Funktionsweise eines evolutionären Algorithmus.	21
4.1	1-Bit Volladdierer.	32
4.2	Ripple-Carry Addierer.	33
4.3	Diagramm der Zustandsübergänge.	34
4.4	Eingangssignale des Volladdierers.	35
4.5	Ausgangssignal des Volladdierers.	36
4.6	Power-Delay-Kurve eines Inverters.	38
4.7	Power-Delay-Kurve eines Inverters.	39
4.8	Erläuterung der iterativen Gewichtsbestimmung.	40
4.9	Beispiel der iterativen Gewichtsbestimmung.	41
4.10	Energiediagramm der RTD.	42
4.11	Strom-Spannungs-Kennlinie der RTD.	45
4.12	Ergebnisse der RTD-Modellierung.	46

4.13	Symbol des Schwellwertgatters.	49
4.14	Majoritätsgatter nach Lee.	50
4.15	Ein-/Ausgangssignal des Lee-Gatters.	51
4.16	Spiegelladdierer.	52
4.17	Vollladdierer mit Schwellwertgattern.	54
4.18	Vollladdierer mit Lee-Gattern.	55
4.19	Transistorlayout nach $0,35\ \mu\text{m}$ CMOS Design Rules.	56
4.20	Transistorlayout mit Weite kleiner als $1,0\ \mu\text{m}$	57
4.21	Optimierungsverlauf mit EA für das Lee-Gatter.	58
4.22	Einzellösungen des Optimierungsverlaufs.	59
5.1	Beispiel für den Zielgrößenraum einer Schaltung.	64
5.2	Erläuterung der Dominanzrelation.	65
5.3	Zwei Beispiele einer Überdeckung.	66
5.4	Gegenüberstellung: global–lokal Pareto optimale Mengen.	68
5.5	Gegenüberstellung: konvexe–nicht konvexe Pareto Menge.	68
5.6	Bestimmung der Pareto Menge mit der Gewichtsmethode.	69
5.7	Beispiel zur Anwendung der Beschränkungsmethode.	70
5.8	Beispiele für die Elementbewertung im SPEA.	75
5.9	Struktur des steady-state evolutionären Algorithmus.	78
5.10	Gegenüberstellung: synchroner–asynchroner SPEA.	80
5.11	Gegenüberstellung: SPEA–Räuber-Beute Modell–Gewichtsmethode.	82
6.1	Power-Delay-Kurve des Inverters.	87
6.2	Beispiele nicht monotoner Parameterzusammenhänge.	89
6.3	Umgeformte Parameterzusammenhänge.	90
6.4	Die verschiedenen Zahlenräume.	91

6.5	Verlustleistung-Versorgungsspannung-Kurve des Inverters (1).	92
6.6	Verlustleistung-Versorgungsspannung-Kurve des Inverters (2).	92
6.7	Überdeckung und Pareto Menge.	93
6.8	Verlustleistungs-Verzögerungszeit des Inverters mit Extraktion.	95
6.9	Verlustleistungs-Verzögerungszeit des Lee-Gatters.	96
6.10	Verlustleistungs-Verzögerungszeit des Lee-Addierers.	97
6.11	Verlustleistungs-Verzögerungszeit des Spiegeladdierers.	97
6.12	Verlustleistungs-Versorgungsspannung des Lee-Gatters.	98
6.13	Verlustleistungs-Versorgungsspannung des Lee-Addierers.	99
6.14	Verlustleistungs-Versorgungsspannung des Spiegeladdierers.	99
6.15	Transistorweiten korrekter Schaltungsdesigns (Lee-Gatter).	100
6.16	Verlustleistung-Chipfläche des Lee-Gatters (1).	101
6.17	Verlustleistung-Chipfläche des Lee-Gatters (2).	102
6.18	Verlustleistung-Chipfläche des Lee-Addierers.	103
6.19	Verlustleistung-Chipfläche des Spiegeladdierers.	104
6.20	Relative Robustheit-Versorgungsspannung des Lee-Gatters.	105

Symbolverzeichnis

In der Reihenfolge der Erwähnung:

\bar{P}	mittlere Verlustleistung
t_d	Verzögerungszeit
V_{DD}	Versorgungsspannung
I_{dl}	Dioden-Leckstrom
I_S	Sättigungssperrstrom
V_T	Temperaturspannung
k	Boltzmannkonstante
T	Temperatur
q	Elementarladung
I_{st}	Unterswellenstrom
V_{GS}	Gate-Source-Spannung
V_{th}	Schwellenspannung
P_d	dynamische Verlustleistung
α	Schalzhäufigkeit
C_{eff}	effektive Kapazität
P_q	Verlustleistung aufgrund der Querströme
β	Übertragungsfaktor eines Transistors
τ	Signal-anstiegs/abfall-Zeit
T_{clk}	Periodenlänge eines Schaltsignals
$p(t)$	momentane Verlustleistung
$i_{DD}(t)$	Gesamtstrom zwischen V_{DD} und Masse
f, \vec{f}	Bewertungsfunktion, Zielgröße

X	Parameterraum
\vec{x}	Element des Parameterraumes
\vec{x}^*	globales Optimum im Parameterraum
$h_j(\vec{x})$	Bedingung j für eine Gleichungsrestriktion
n_h	Anzahl der Gleichungsrestriktionen
$g_j(\vec{x})$	Bedingung j für eine Ungleichungsrestriktion
n_g	Anzahl der Ungleichungsrestriktionen
$\vec{x}^{(t)}$	Element \vec{x} im Iterationsschritt t
\vec{x}^+	lokales Optimum im Parameterraum
η	Schrittweite des Gradientenverfahrens
ε	beliebig kleine reellwertige Größe
$E^{(g)}$	Population im Iterationsschritt g
g	Generationsindex
$N^{(g)}$	Population der Nachkommen im Iterationsschritt g
μ	Größe der Eltern-Population
λ	Größe der Nachkommenpopulation
$\vec{\sigma}$	Strategieparameter zum Parametervektor \vec{x}
P	gleichverteilte Zufallsvariable
$N(\cdot, \cdot), N_i(\cdot, \cdot)$	normalverteilte Zufallsvariablen
I	Individuum / Element der Population
τ, τ'	Strategieparameter der Evolutionsstrategie
ε_σ	Minimum für σ_i
$p_i(\vec{x})$	Strafterm für Element \vec{x}
ν_i	Gewichtung des Strafterms $p_i()$
f'	modifizierte Zielgröße
a, b, c	Eingangsgrößen des Volladdierers
s, c'	Summe und ausgangsseitiges Carry des Volladdierers
\vec{w}	Gewichtung der Zielfunktion
m	Dimension des Gewichtsvektors
A, B	Elemente im Zielgrößenraum
\vec{w}^+, \vec{w}^-	Bewertete Gewichtungen der Zielfunktion
\vec{w}^*	optimale Gewichtung der Zielfunktion
E_F	Fermienergie
E_n	Energieniveau in Doppelbarriere

I_{RTD}	Gesamtstrom der RTD
V_W	Resonanzbreite der RTD
V_P	Peakspannung der RTD
V_N	Spannung am maximalen NDR der RTD
V_V	Valleyspannung der RTD
I_P	Peakstrom der RTD
I_V	Valleystrom der RTD
n_R	Resonanzstromparameter der RTD
n_V	thermoionischer Stromparameter der RTD
f_{clk}	Schaltfrequenz
Θ	Schwellwert
W	Transistorweite
L	Transistorlänge
$M?$	Transistorkennzeichnung
r	Robustheit
C_L	Lastkapazität
\vec{x}_P	Element der Pareto Menge
X_P	Pareto Menge
N, M	Dimensionen einer strukturierten Population
$I_{p,q}$	Individuum im Räuber-Beute Model an Position p, q
E'	Externe Population des SPEA
I'_i	Individuum der externen Population
s_i	Stärke des Individuums I'_i
N	Größe der internen Population
N'	Maximale Größe der externen Population
$d(\cdot, \cdot)$	Abstand zweier Punkte im Zielgrößenraum
k_+	Überschreitung der maximalen Populationsgröße
\overline{P}'	modifizierte Verlustleistung
U	Überdeckung der Pareto Menge
\vec{u}	Element der Überdeckung
$g(f_1, \dots, f_{n-1})$	hypothetischer Parameterzusammenhang
$h(f_1, \dots, f_{n-1})$	Schätzfunktion für $g(f_1, \dots, f_{n-1})$
d_h	Abstand zwischen Überdeckung und analytischem Schätzer h
r'	modifizierte Robustheit

Kapitel 1

Einleitung

Die Entwicklung integrierter Schaltungen folgt einer stetigen Miniaturisierung, die sich auch in den nächsten Jahren fortsetzen wird [SIA00]. Sie ermöglicht einerseits, neuartige Bauelemente in der Nano-Technologie zu realisieren, andererseits lassen sich neue Schaltungstechniken umsetzen. Gemeinsam ist beiden Entwicklungsrichtungen, dass die Grundsaltungen trotz äußerer digitaler Eigenschaften intern analog betrachtet werden müssen und sich der analoge Einfluss mit zunehmender Skalierung verstärken wird [Vee98]. So lässt sich die Laufzeit- und/oder Verlustleistungsoptimierung einer Schaltung nicht wie bei klassischer CMOS Technik auf abstrakter, wie der Gatterebene, durchführen, sondern muss im Analogbereich auf Transistorebene erfolgen. Die mit der Miniaturisierung stetig steigende Komplexität neuer Systeme beruht nicht allein auf der wachsenden Zahl integrierter Objekte, sondern ebenso auf einem erhöhten Funktionsumfang der jeweiligen Gatter. Dieser höheren Komplexität ist durch eine detailliertere Analyse Rechnung zu tragen.

Bei der Verwendung etablierter Techniken und Technologien existiert das notwendige Systemwissen, um systematisch ein Schaltungsdesign zu erstellen, das nahe am Optimum liegt. In diesem Fall sind klassische Optimierungsverfahren geeignet, um verbesserte Lösungen zu bestimmen. Für die Untersuchung neuartiger Konzepte, zu denen keinerlei spezielles Vorwissen existiert, sind jedoch globale Suchstrategien wie die evolutionären Algorithmen erforderlich. Mit diesen wird es möglich, unerwartete Lösungen bzw. überhaupt ein funktionsstüchtiges Design zu erhalten. So lassen sich durch diesen globalen Ansatz objektivierte Ergebnisse erzielen, die eine Abschätzung des vorhandenen Potenzials einer neuen Technik erlauben.

In der vorliegenden Arbeit wird das Design integrierter Digitalschaltungen auf Transistor-ebene betrachtet, deren Simulation und Beurteilung im Analogbereich erfolgt. Für die nominelle Optimierung der Schaltungen werden evolutionäre Algorithmen eingesetzt. Hierbei werden praxisrelevante Erfordernisse wie begrenzte Zeitressourcen und möglichst geringes algorithmus- und problemspezifisches Vorwissen berücksichtigt. Dies spiegelt sich in der Verwendung parallelisierter Varianten der evolutionären Algorithmen wider, für die nur wenige Parameter vom Anwender zu wählen sind.

Für die Optimierung mehrerer Größen eröffnen die evolutionären Algorithmen erstmalig die Möglichkeit, die Pareto Menge eines Optimierungsproblems effizient zu bestimmen. Geeignete Verfahren wurden in den letzten zehn bis fünfzehn Jahren entwickelt. Hier werden sie für die Mikroelektronik angewandt, um Eigenschaften und Parameterabhängigkeiten in integrierten Schaltungen untersuchen zu können. Die Frage nach Zusammenhängen wie beispielsweise zwischen minimaler Verlustleistung und maximaler Verzögerungszeit lässt sich in ein Optimierungsproblem bezüglich mehrerer Größen umformulieren. Dessen Lösung kann dann Aufschluss über den gewünschten Zusammenhang liefern. Es können sowohl direkt kontrollierbare Parameter, wie die Versorgungsspannung, als auch indirekt beeinflusste Größen, wie die beiden beispielhaft genannten, betrachtet werden.

Der Ansatz, funktionale Zusammenhänge zwischen charakteristischen Größen einer integrierter Schaltung zu bestimmen, bietet neue Einblicke in die Eigenschaften integrierter Schaltungen. Sie erlauben eine noch spezifischere Bewertung des Potenzials der angewendeten Technik und Technologie.

Die Perspektiven, für den Bereich integrierter Schaltungen durch die Eingrößenoptimierung einerseits und die Mehrgrößen- bzw. Vektoroptimierung andererseits, sind völlig verschieden voneinander (Abbildung 1.1). Die Eingrößenoptimierung führt zu skalaren Kenngrößen und einzelnen Realisierungen der Schaltung. Die Vektoroptimierung hingegen erlaubt die Bestimmung von Parameterzusammenhängen und dadurch ein besseres Systemverständnis zu gewinnen.

Die Arbeit gliedert sich in zwei Teile. Im Ersten, bestehend aus den Kapiteln Zwei bis Vier wird in die Thematik der Schaltungsoptimierung auf Transistorebene eingeführt und die Optimierung auf Basis numerischer Simulationen insbesondere mit evolutionären Algorithmen charakterisiert und diskutiert. Die auf eine Zielgröße beschränkte Optimierung wird beispielhaft auf Schwellwertgatter und auf Volladdierer sowohl in Schwellwerttechnik als auch in Boolescher CMOS Technik angewendet. Im zweiten Teil der Arbeit, der sich aus den Kapiteln fünf und sechs zusammensetzt, liegt der Fokus auf der Vektoroptimierung. Nach einer

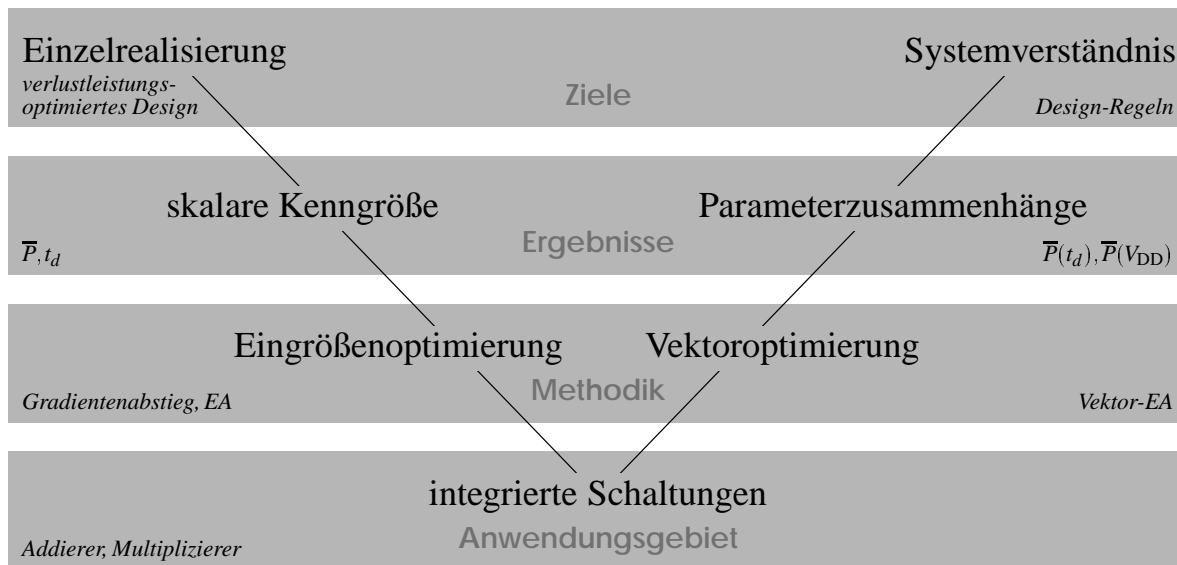


Abbildung 1.1: Ziele der verschiedenen Optimierungsansätze mit typischen Beispielen.

Einführung in deren Grundlagen werden die Verfahren auf die Bestimmung von allgemeinen Parameterzusammenhängen übertragen und diskutiert. Die Analyse integrierter Schaltungen mit den vorgestellten Verfahren wird abschließend exemplarisch demonstriert.

Kapitel 2

Schaltungsdesign auf Transistorebene

Der Entwurf integrierter Schaltungen wird heute in mehrere Abstraktionsebenen unterteilt. So können in jeder Ebene spezifische Konkretisierungen einer Problemlösung erfolgen und mittels fixierter Zwischenergebnisse verifiziert werden. Die verschiedenen Sichtweisen oder Domänen bei einem solchen Entwurf lassen sich im Y-Diagramm darstellen (Abb. 2.1) [WE85]. In jedem Abstraktionsniveau haben sich adäquate automatisierte Entwurfsmethoden etabliert. Beginnend mit einer Strukturierung des umzusetzenden Algorithmus über die Reduktion Boolescher Ausdrücke bis hin zur Abbildung auf eine gewählte Technologie, wird der Entwurf schrittweise konkretisiert. Die letztgenannte Abbildung erfolgt hierbei auf der Basis einer Standardzellenbibliothek, die für eine konkrete Technologie und vordefinierte Zeitanforderungen die Realisierung von logischen Gattern definiert. Entgegen diesem top-down Entwurf erfolgt die Erstellung einer Standardzellenbibliothek bottom-up, d.h. Design und Simulation der Schaltung auf Schaltungsebene liefern die Realisierungen und Kenngrößen einzelner Gatter oder Register.

Die Betrachtung einer Schaltung auf der Schaltkreis- oder Transistorebene steht im Blickpunkt der vorliegenden Arbeit. Die zu realisierenden Gatter seien durch ihre Anschlüsse und das geforderte Ein-/Ausgangsverhalten definiert. Der dann folgende Entwurf auf Schaltungsebene innerhalb dieser Rahmenbedingungen ist in zwei Aspekte gegliedert: Erstens ist die Struktur oder Topologie der Schaltung festzulegen, indem die logische Schaltfunktion mit einer konkreten Schaltungstechnik umgesetzt wird. Als Zweites sind die noch freien Parameter dieser Schaltungsstruktur zu wählen. Diese können geometrische Größen, wie Transistorabmessungen, aber auch Spannungen und Kapazitäten sein. Die Schaltungsstruktur sei hier bereits gegeben, sei sie manuell durch einen Experten oder mittels eines automatischen Ent-

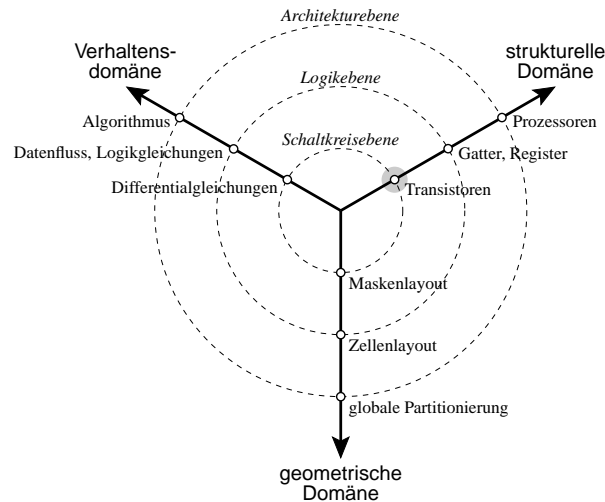


Abbildung 2.1: Y-Diagramm: Das Schaltungsdesign auf Transistorebene ist im Diagramm im grau hinterlegten Bereich lokalisiert.

wurfsverfahrens erstellt, so dass sich der Prozess des Schaltungsdesigns in dieser Arbeit auf die Parameterbestimmung in einer festen Struktur beschränkt. Dies entspricht der Motivation der Arbeit, konkrete Schaltungstechniken bewerten zu können.

2.1 Schaltungsoptimierung

Der Begriff der Optimierung wird für das Design auf Schaltungsebene mit unterschiedlichen Zielsetzungen verwendet. So lässt sich die nominelle Optimierung von der statistischen differenzieren. In der nominellen Optimierung stehen die Eigenschaften einer einzelnen Realisierung einer Schaltung im Vordergrund. In Anbetracht der nicht exakten Reproduzierbarkeit einer Schaltung und der Diskrepanz zwischen Simulation und Realität (vergleiche Kapitel 2.1.2), ist dies die Verbesserung eines eher theoretischen Schaltverhaltens. Die nominelle Optimierung wird bezüglich spezieller Größen, wie Verlustleistung und Schaltgeschwindigkeit durchgeführt. Dem gegenüber konzentriert sich die statistische Optimierung auf den Effekt der Parameterschwankungen bei der Herstellung integrierter Schaltungen. Ziel kann hierbei die Maximierung der Ausbeute (Yield) sein. Dies ist der Anteil an der gesamten Produktion von denjenigen Schaltungen, die die gegebenen Spezifikationen einhalten. Diese Arbeit konzentriert sich auf den Bereich der nominellen Optimierung zur Bestimmung des Potenzials

einer Schaltungstechnik. Aspekte der statistischen Optimierungen werden unter dem Begriff Robustheit in Kapitel 4.3.5 und 6.4 angesprochen.

Das Design integrierter Schaltungen auf Transistorebene ist durch die Verwendung eines Simulators, auf dessen Basis die Entwicklung einer Schaltung erfolgt, gekennzeichnet. Da das Verhalten einer Schaltung auf dieser Ebene nicht exakt vorhersagbar ist, dient eine numerische Simulation zur Überprüfung der gewünschten Eigenschaften. So kann z.B. durch eine Simulation im Zeitbereich das dynamische Verhalten der Schaltung kontrolliert werden. Die Bauelemente sind hierzu durch Modelle beschrieben, die den Zusammenhang zwischen Ein- und Ausgangsgrößen wiedergeben. Durch die Verwendung eines Simulators als Verbindungsglied zur Realisierung einer Schaltung, sind bei einer Optimierung verschiedenartige Anforderungen zu berücksichtigen. Diese leiten sich aus Eigenschaften des Simulators, des realen Systems und der Beziehung beider zueinander ab.

2.1.1 Eigenschaften des Parameterraumes

Für die Entwicklung einer Schaltung, die anschließend in einem Chip realisiert wird, ist es wichtig, die so genannten Design Rules des künftigen Herstellers einzuhalten. Die Design Rules beschreiben Vorgaben bezüglich Lage und Größe der Bauelemente, die erforderlich sind, um die Schaltung in der Technologielinie des Herstellers fertigen zu können. Hierzu zählen Minimalmaße für die Transistorweiten und -längen ebenso wie eine Diskretisierung. Letztere legt die minimale Variationsschrittweite der Geometrie Größen fest. Am Beispiel der $0,35\ \mu\text{m}$ Technologie in Kapitel 4.3.5 bedeutet dies eine minimale Gatelänge von $0,3\ \mu\text{m}$, die in Schritten von $0,05\ \mu\text{m}$ vergrößert werden kann. Daneben wirken sich Eigenschaften des verwendeten Transistormodells aus, dessen Gültigkeit bezüglich der beschreibenden Parameter begrenzt ist. So ist die Versorgungsspannung genügend groß zu wählen, so dass die schaltenden Transistoren sicher den Zustand der Inversion erreichen. Zudem sind die Geometrie Größen und Spannungen nach oben zu begrenzen, da andernfalls weitere Effekte auftreten, die in den Modellen keine Berücksichtigung finden (z.B. maximale Stromdichte, Kapazitäten bei sehr großen Transistoren). Im Hinblick auf die allgemeine Zielsetzung kleine, schnelle und verlustleistungsarme Schaltungen zu entwerfen, werden diese oberen Schranken auf Werte gesetzt, die weit vor dem Einsetzen solcher Effekte liegen (z.B. maximale Versorgungsspannung = $3\ \text{V}$). Diese Begrenzung des Suchraumes bewirken zudem eine Beschleunigung der Optimierung.

Zusammenfassend ergibt sich jeder Parameter als Element eines diskretisierten Intervalls. Somit ist der Parameterraum diskontinuierlich, konvex und abzählbar. Das Optimum einer Schaltung könnte also durch eine vollständige Enumeration ermittelt werden, jedoch liegt der erforderliche Zeithorizont jenseits praktikabler Grenzen. So ist bereits für das Majoritätsgatter in Kapitel 4.3.2 der Parameterraum größer als 10^{10} Elemente.

2.1.2 Grenzen der Optimierung

In den vorangegangenen Abschnitten ist bereits die Diskrepanz zwischen der Simulation einer integrierten Schaltung und deren Realisierung betont worden. Diese wird durch mehrere Einflüsse getragen (Abbildung 2.2). Zum Ersten ist bereits die Genauigkeit, mit der ein Modell einen perfekt hergestellten Transistor abbilden kann, begrenzt. Dies ergibt sich aus der Generierung des Modells anhand von Messreihen, die prinzipiell fehlerbehaftet sind. Zum Zweiten greifen hier die Schwankungen bei der Herstellung der Schaltungen, die sich sowohl über den Wafer als auch über mehrere Chargen hinweg auswirken. Der dritte Punkt ist das Fehlen jeglicher Strukturinformationen über die Lage der Transistoren zueinander sowie der Leitungsgeometrien. Erst diese würde die Berücksichtigung parasitärer Kapazitäten erlauben, die das Schaltverhalten beeinflussen. Hierzu wären jedoch vor jeder Simulation eine Platzierung der Bauelemente und das Setzen der Leitungsverbindungen (place & route) erforderlich, um die parasitären Effekte extrahieren zu können. Dieses würde den Zeitbedarf zur Bewertung der Parameter erheblich vergrößern.

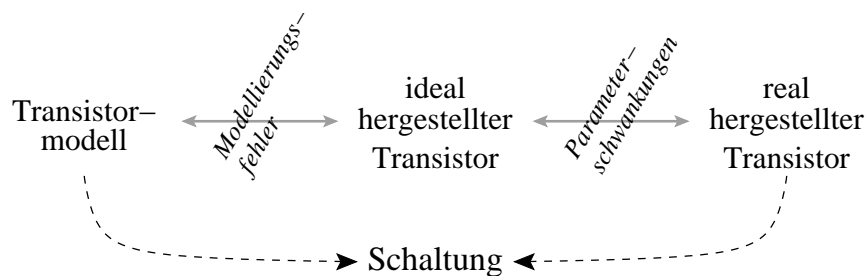


Abbildung 2.2: Begrenzende Effekte für die Güte einer auf Simulationsbasis optimierten Schaltung.

Die Differenz zwischen Simulation und Realität ist verbunden mit einer relativ starken Unsicherheit in der Bewertung der Schaltungen. Es stellt sich daher die Frage nach dem Sinn des Erreichens des absoluten Minimums innerhalb der Simulationsumgebung. So kann die

Bestimmung einer Lösung, die zu 10 % vom absoluten Minimum abweicht, als ausreichend angesehen werden. Diese Spanne bewegt sich einerseits innerhalb der Differenz zwischen Simulation und Realität. Zudem kann im Vergleich zweier Schaltungstechniken ein Unterschied innerhalb dieses Betrages nicht als signifikantes Merkmal zur Bevorzugung einer der beiden Schaltungstechniken herangezogen werden. Der Fortschritt durch die sich schnell entwickelnde Halbleitertechnologie wird bereits in wenigen Jahren um ein Vielfaches größer als diese Differenz sein [SIA00].

2.2 Existierende Ansätze

Die verschiedenen Ansätze der nominellen Schaltungsoptimierung lassen sich anhand ihrer Charakteristika differenzieren und gruppieren. Die Merkmale beziehen sich sowohl auf die Problemspezifizierung als auch auf deren Lösungsstrategie. Gemeinsam ist den meisten dargestellten Ansätzen der Verzicht auf den Anspruch der absoluten Optimalität der Lösung oder die Beschränkung auf die Verbesserung einer existierenden Lösung. Dies ist legitim in Bezug auf deren Anwendungsperspektive, nämlich die Automatisierung des industriellen Designprozesses, in dem ein hohes Maß an Vorwissen über die verwendete Schaltungstechnik vorausgesetzt werden darf. Dieses Anwendungsziel deckt sich somit nicht vollständig mit der Motivation der vorliegenden Arbeit, jedoch erlauben diese Ansätze einen Überblick über das Gebiet der nominellen Schaltungsoptimierung.

Die Verlustleistung einer CMOS Schaltung wird durch eine Reihe verschiedener Effekte hervorgerufen. Die mittlere Verlustleistung \bar{P} ergibt sich als Summe der einzelnen Anteile, die im Folgenden genannt werden. Im Ruhezustand der Schaltung, d.h. ohne transiente Vorgänge, kann einerseits statische Verlustleistung auftreten, die abhängig von der Schaltungstechnik ist. Andererseits führen Leckströme zu einem weiteren Verlustleistungsanteil, der durch die verwendete Technologie bestimmt ist. Der Dioden-Leckstrom I_{dl} (Gleichung (2.1), I_S : Sättigungssperrstrom, $V_T = \frac{kT}{q}$) tritt an den gesperrten pn-Übergängen der Transistoren auf [ML98]. Hinzu kommt der Unterschwellenstrom I_{st} (sub-threshold leakage) (Gleichung (2.2), V_{th} : Schwellenspannung), der den endlichen Widerstand des gesperrten Transistorkanals erfasst [CB95]. Während des (dynamischen) Schaltvorganges wird Leistung P_d zum Umladen der effektiven Lastkapazität C_{eff} benötigt (Gleichung (2.3), α : Schaltfrequenz) [HIG94] und Verlustleistung P_q durch den Querstrom (short-circuit) verbraucht. Der Querstrom fließt zwischen V_{DD} und Masse, während der PMOS- und der NMOS-Zweig

gleichzeitig aktiv sind (Gleichung (2.4), β : Übertragungsfaktor). Er ist neben der Versorgungsspannung, insbesondere durch das Verhältnis der Anstiegs- und Abfallzeiten τ zur Periodenlänge T_{clk} des Signals bestimmt [Vee84].

$$I_{dl} = I_S \cdot (e^{V/V_T} - 1) \quad (2.1)$$

$$I_{st} \sim e^{(V_{GS} - V_{th})/V_T} \quad (2.2)$$

$$P_d = \alpha C_{\text{eff}} V_{\text{DD}}^2 \quad (2.3)$$

$$P_q = \frac{\beta}{12} (V_{\text{DD}} - 2V_{th})^3 \frac{\tau}{T_{\text{clk}}} \quad (2.4)$$

Die Bewertung integrierter Schaltungen auf Schaltungsebene kann sowohl durch eine relativ exakte numerische Simulation der Bauelemente und deren Verknüpfung, als auch in abstrakter analytischer Form erfolgen. Letztgenanntes ist bei der Betrachtung von klassischen digitalen CMOS Schaltungen möglich [YS96, GLPS97], für die die dynamische Verlustleistung P_d dominierend ist, so dass deren alleinige Betrachtung zur Bewertung der Schaltung genügt.

Der Vorteil dieser stärkeren Abstraktion liegt in einer einfacheren und schnelleren Bewertung der Schaltung, die folglich die Betrachtung komplexerer Schaltungen gestattet. Allerdings setzt dies ein tiefes Systemwissen, wie es bei klassischer CMOS Technik existiert, voraus. Zudem führt die Vereinfachung zu einer ungenaueren Bewertung der Schaltung, die jedoch unter Berücksichtigung der Zielsetzung der Optimierung akzeptabel sein kann. Die Optimierung von Analogschaltungen setzt stattdessen die Verwendung eines numerischen Simulators voraus [BHSV81, NRSVT88, KBAK96, Vis97]. Dieser erlaubt die Berücksichtigung sämtlicher oben genannter Effekte. Für die Bestimmung der Verlustleistung wird dann in der transienten

Simulation über den Strom zwischen V_{DD} und Masse integriert und gemittelt (2.5) [Rab96]. Die Zusammenhänge in Analogschaltungen sind in der Regel zu komplex, als dass sie durch einfache Modelle ausreichend beschrieben werden könnten.

$$\bar{P} = \frac{1}{t_0} \int_0^{t_0} p(t) dt = \frac{V_{DD}}{t_0} \int_0^{t_0} i_{DD}(t) dt \quad (2.5)$$

In [GLPS97] werden die bezüglich der Verzögerungszeit kritischen Transistoren bestimmt, indem der Datenpfad analysiert und Zeitvorgaben für den Ausgang jedes Gatters definiert werden. Derart lokalisierte Randbedingungen erlauben die Konzentration der Optimierung auf den momentan zeitkritischen Transistor im kritischen Pfad. Bei der Untersuchung analoger Schaltungen ist ein solches Segmentieren nicht möglich. Hier wird das Ein-/Ausgangsverhalten der Schaltung als Ganzes definiert und eine Parameteroptimierung durchgeführt.

Zur Lösung der oben definierten Optimierungsprobleme werden unterschiedliche Ansätze verfolgt. So werden neben der linearen Programmierung [GLPS97] auch Verfahren der nicht linearen Optimierung [Vis97] eingesetzt. Das in [NRSVT88] beschriebene Verfahren ist in mehrere Phasen eingeteilt, in denen zunächst versucht wird die Randbedingungen sukzessive zu erfüllen, bevor eine Art Gradientenabstieg zu einer optimalen Schaltung führt. Ein Lösungsansatz auf der Basis evolutionärer Algorithmen zur Optimierung integrierter Schaltungen findet sich in [KBAK96]. Es werden sowohl die Struktur der Schaltung als auch die Parameter der Bauelemente bestimmt. Dies wurde für Filterstufen unter Verwendung eines numerischen Simulators durchgeführt. Weitere Arbeiten, die den Einsatz evolutionärer Algorithmen zur Optimierung von Filtern oder Operationsverstärkern exemplarisch darstellen, finden sich in [KB97, PC97, LC99]. In [SV96] ist der Ansatz des „Simulated Annealing“, der als spezieller evolutionärer Algorithmus gesehen werden kann, für die Optimierung eines Filters verwendet worden.

Gemeinsam ist sämtlichen Arbeiten zur Schaltungsoptimierung auf der Grundlage numerischer Simulationen die Konzentration auf Analogschaltungen. Bei der hier vorliegenden Thematik der Optimierung digitaler Schaltungen auf gleicher Ebene, sind jedoch Zielfunktionen mit potenziell stärkeren Nichtlinearitäten zu erwarten. Diese Annahme lässt sich aus der Tatsache motivieren, dass in Analogschaltungen Transistoren typischerweise im linearen Verstärkungsbereich betrieben werden, während für Digitalschaltungen die Transistoren als nicht lineare Schaltelemente Verwendung finden.

In anderen Bereichen, neben dem Schaltungsdesign auf Transistorebene, haben sich evolutionäre Algorithmen als alternatives Optimierungswerkzeug bereits etabliert. So existieren erfolgreiche Anwendungen für den Schaltungsentwurf auf höherer Abstraktionsebene [Lan98] ebenso wie für die Platzierung und Verdrahtung von Teilschaltungen und die automatische Generierung von Testmustern [MR99].

Kapitel 3

Parameteroptimierung in der Simulation

Die Optimierung von Systemen bzw. von Systemmodellen ist sowohl auf Strukturebene als auch bezüglich der Systemparameter möglich. Im Rahmen einer Strukturoptimierung wird der Aufbau eines Systems evaluiert, wobei Randbedingungen und Charakteristika der Struktur einschränkende Größen sind. Bei der Parameteroptimierung ist die Struktur des Systems oder dessen Modells fest gegeben, sei es durch einen menschlichen Experten, eine vorherige Optimierung oder durch die bloße Tatsache der realen Existenz des Systems. Ein solches System beinhaltet in der Regel freie reellwertige oder diskrete Parameter, die das Verhalten des Systems beeinflussen. Die Betrachtungen in dieser Arbeit beschränken sich unter Berücksichtigung der Anwendung auf diesen letzten Fall, die Parameteroptimierung.

Die Algorithmen zur Behandlung eines Optimierungsproblems werden üblicherweise als Optimierungsverfahren bezeichnet, obwohl es sich korrekterweise um zielgerichtete Suchverfahren handelt, die eine Lösung, die nahe dem Optimum sein soll, bestimmen. Daher wird dieser Vorgang auch als Meliorisierung bezeichnet. Wenn in dieser Arbeit von Optimierung und den dazu eingesetzten Methoden gesprochen wird, so sind darunter entsprechende Suchverfahren zu verstehen.

3.1 Grundlagen

Systeme aus der Anwendung, die optimiert werden sollen, sind in der Regel real existierende Systeme. Um Aussagen über den Einfluss möglicher Systemvariationen vornehmen zu

können, sind Modelle des Systems erforderlich. Direkte Eingriffe in das reale System verbieten sich zumeist aus zeitlichen, wirtschaftlichen oder sicherheitstechnischen Erwägungen. Ein Modell ist eine mehr oder weniger exakte Beschreibung eines Systems mit Hilfe von deterministischen oder stochastischen Prozessen in analytischer oder numerischer Form. Das Systemverhalten lässt sich dann durch eine Simulation unter Verwendung des Modells nachbilden. Die Güte, d.h. die Abbildungsgenauigkeit einer Simulation bzw. eines Modells, ist von verschiedenen Faktoren beeinflusst. Neben der Tiefe des Einblicks, der in das System möglich ist, sind die Erkenntnisse, die gewünscht sind, und die Zeit bzw. Komplexität, die eine Simulation erfordern darf, entscheidend [LK82].

Die Optimierung im Parameterraum lässt sich bei einer skalaren Zielgröße $f(\cdot)$ über dem Parameterraum X beschreiben als $f(\vec{x}) \rightarrow \min$ zur Bestimmung eines $\vec{x}^* \in X$ mit $f(\vec{x}^*) \leq f(\vec{x})$ für alle $\vec{x} \in X$.

Die Einbettung in eine Simulationsumgebung wirft einige Aspekte [HB98] auf, die zum Teil weniger theoretische Betrachtungen sondern mehr den praktischen Einsatz beeinflussen. Diese Aspekte, in Tabelle 3.1 aufgelistet, werden im Folgenden erläutert: Die vorliegende Optimierungslandschaft ist typischerweise multimodal und stochastisch beeinflusst, was die Wahl einer adäquaten Optimierungsstrategie erfordert (1,2). Es sind keine direkten Informationen über die partiellen Ableitungen der Zielfunktion verfügbar. Das Simulationsmodell stellt sich also als eine Art *Black Box* dar (3). Die Menge der zulässigen Lösungen, die die Problemstellung erfüllen, ist lediglich eine Teilmenge des Parameterraumes X . Sie ist definiert durch Restriktionen, die implizit im Modell definiert sind (4). Ein wichtiger Punkt bei realen Systemen ist die Empfindlichkeit, die das Optimum möglicherweise gegenüber den Eingangsgrößen aufweist (5). Dies kann die Realisierung eines gefundenen Optimums scheitern lassen und führt zum Aspekt der Robustheit. Im Anwendungsbereich des Entwurfs integrierter Schaltungen bedeutet dies Folgendes. Ein auf Simulationsbasis optimiertes Design kann eventuell, aufgrund der von Herstellungstoleranzen verursachten Parametervariationen, den geforderten Spezifikationen nicht mehr genügen. Eine große Herausforderung ist die Handhabung zeitlich varianter Systeme, bei denen das Optimum „wandert“, dieser Bereich der dynamischen Optimierung wird hier jedoch nicht weiter berücksichtigt (6). Relevant für den praktischen Einsatz einer Optimierung sind insbesondere die beiden folgenden Punkte: Die Zielfunktionsauswertung ist in der Simulation kosten- bzw. zeitintensiv (7). Dies beschränkt die Zahl der Auswertungen, um das Optimierungsergebnis nicht nur theoretisch erreichen zu können, sondern nach einem vorgegebenen Zeitraum auch zu nutzen. Zudem ist der Zeitrahmen für eine Simulation nicht notwendigerweise konstant, was die Wahl der Optimierungsmethode im Hinblick

Tabelle 3.1: Aspekte der simulationsbasierten Optimierung.

1. Zielfunktion typischerweise multimodal.
2. Zielfunktion typischerweise stochastisch beeinflusst.
3. Modell als Black Box.
4. Implizite Restriktionen begrenzen Parameterraum.
5. Eingangsempfindlichkeit des Optimums.
6. Zeitvariante Systeme.
7. Kosten-/zeitintensive Auswertung der Zielfunktion.
8. Zeitbedarf einer Zielfunktionsauswertung nicht notwendigerweise konstant.

auf Parallelisierungsmöglichkeiten beeinflusst (8). Die Varianz der Auswertungszeit ist in der ereignisorientierten Simulation oder, bezogen auf den Anwendungsbereich dieser Arbeit, in unterschiedlichen Konvergenzgeschwindigkeiten in der numerischen Simulation von SPICE begründet.

3.2 Numerische Suchverfahren

Für praxisrelevante Optimierungsprobleme stellen Suchverfahren die Möglichkeit dar, mit absehbarem Rechenaufwand eine akzeptable Näherungslösung zu ermitteln. Diese Methoden werden trotz eines gänzlich oder bezüglich der Rahmenbedingungen fehlenden Optimalitätsbeweises in der Literatur als Optimierungsverfahren bezeichnet. Das grundlegende Prinzip, das diese Verfahren gemein haben, beruht auf dem Vergleich von Funktionswerten und den Eigenschaften der so genannten Zielfunktion. Die Verfahren beschränken sich allesamt auf den Bereich der skalaren Optimierung. Ansätze zur Vektoroptimierung werden in Kapitel 5 vorgestellt. Das weit gefächerte Feld der Suchverfahren lässt sich nach vielfältigen Kriterien ordnen. So können diese in lineare Programmierung, nicht lineare Programmierung und stochastische Verfahren gruppiert werden. Unter der linearen und nicht linearen Programmierung sind dann die klassischen, deterministischen Verfahren zu verstehen, die auf Optimalitätsbedingungen unter Verwendung der partiellen Ableitungen beruhen, die typischerweise zusätzliche analytische Informationen der Zielfunktion erfordern. Unter den stochastischen Verfahren werden die evolutionären Algorithmen eingeordnet, die neben der Zielfunktion keine zusätzlichen Voraussetzungen fordern. Sie bilden den methodischen Schwerpunkt dieser Arbeit.

3.2.1 Nicht lineare Programmierung

Fragestellungen der Ressourcenplanung motivierten das Gebiet der linearen Programmierung, das sich heute auch im Themenbereich des Operations Research wiederfindet. Unter Berücksichtigung linearer Randbedingungen, die einen konvexen Körper definieren, ist eine lineare Zielfunktion zu minimieren/maximieren. Die Lösung eines solchen Problems findet sich in einer der Ecken des Körpers. Der Simplex-Algorithmus ist hierzu ein gängiges Verfahren [BSMM99].

Die Verallgemeinerung der linearen Programmierung führt zur nicht linearen Programmierung, bei der die Forderung nach Linearität entfällt. Ein Problem der nicht linearen Optimierung ist wie folgt formuliert:

$$\text{Minimiere : } f(\vec{x}) \quad \vec{x} \in X \quad (3.1)$$

unter Berücksichtigung der linearen oder nicht linearen Gleichungsrestriktionen

$$h_j(\vec{x}) = 0 \quad j = 1, \dots, n_h . \quad (3.2)$$

und der linearen oder nicht linearen Ungleichungsrestriktionen

$$g_j(\vec{x}) \geq 0 \quad j = 1, \dots, n_g. \quad (3.3)$$

Für die nicht lineare Programmierung werden typischerweise iterative Algorithmen eingesetzt, die einen Startvektor $\vec{x}^{(0)}$ in einen verbesserten Vektor $\vec{x}^{(1)}$ überführen. Die wiederholte Ausführung führt zu einer Punktfolge $\vec{x}^{(0)}, \vec{x}^{(1)}, \dots, \vec{x}^{(t)}$, die in der Anwendung endlich ist und den Lösungspunkt \vec{x}^+ liefert. Dessen Bewertung $f(\vec{x}^+)$ soll dem realen Optimum $f(\vec{x}^*)$ möglichst nahe kommen. Ein Beispiel für die nicht lineare Optimierung ist das Gradientenverfahren, das im folgenden Kapitel beschrieben wird.

3.2.2 Gradientenverfahren

Das Gradientenverfahren beruht auf der Idee, dem steilsten Abstieg einer Optimierungslandschaft (= Zielfunktionsraum) zu folgen. Dieses anschauliche Bild motiviert die Bezeichnung „Hill-Climbing“ für den Fall einer Maximum-Suche. Unter Hill-Climbing wird eine ganze Gruppe von Methoden zusammengefasst, unter denen das Gradientenverfahren nur ein Vertreter ist. Es ist ein iterativer Algorithmus, bei dem die Veränderung von $\vec{x}^{(t)}$ zu $\vec{x}^{(t+1)}$ durch einen Schritt mit einer genügend kleinen Schrittweite η in Richtung des jeweiligen negativen Gradienten der Zielfunktion f erfolgt (Abbildung 3.1).

$$\vec{x}^{(t+1)} = \vec{x}^{(t)} - \eta \cdot \text{grad } f(\vec{x}^{(t)}) \quad (3.4)$$

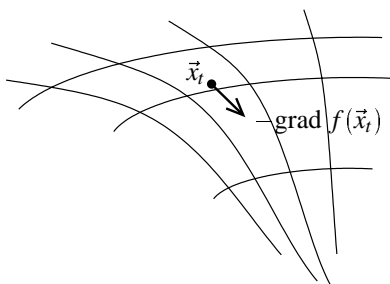


Abbildung 3.1: Gradientenverfahren zur Minimumsuche

Für die Verwendung dieses Verfahrens werden neben der einfachen Zielfunktionsauswertung auch die partiellen Ableitungen, d.h. der Gradient der Zielfunktion benötigt. Für konvexe Zielfunktionen kann diese Methode das Optimum in einem nicht restringierten Parameterraum

sicher bestimmen. Im Falle expliziter linearer Restriktionen reduziert sich der Parameterraum auf einen konvexen Körper. Verletzt ein Optimierungsschritt eine oder mehrere Restriktionen, so kann durch Projektion des Bewegungsvektors auf diese Restriktionsebene die Suche innerhalb des konvexen Parameterraumes fortgeführt werden. Dieser Ansatz wird als Verfahren der projizierten Gradienten bezeichnet [BSMM99].

Einzigster freier Parameter des Verfahrens ist die Schrittweite η , die über die gesamte Optimierung hinweg fest sein oder in jeder Iteration variieren kann. Die Vergrößerung der Schrittweite beschleunigt die Suche einer optimalen Lösung. Jedoch muss η klein genug gewählt werden, um die Stabilität der Suche zu gewährleisten, d.h. eine kontinuierliche Reduzierung des effektiven Suchraumes ($|\vec{x}^{(t_1)} - \vec{x}^{(\infty)}| < \varepsilon \Rightarrow |\vec{x}^{(t_2)} - \vec{x}^{(\infty)}| < \varepsilon, \forall t_1 < t_2$) und eine kontinuierliche Verbesserung des Suchergebnisses ($f(\vec{x}^{(t_2)}) \leq f(\vec{x}^{(t_1)}), \forall t_1 < t_2$). Andernfalls sind Verschlechterungen in der Zielfunktion und „Sprünge“ im Parameterraum zu beobachten. Somit ist die Schrittweite problemspezifisch zu wählen, um den Kompromiss zwischen Suchgeschwindigkeit und Stabilität zu gewährleisten. Für $\eta \rightarrow 0$ erreicht dieses Verfahren sicher eine lokal optimale Lösung, d.h. $f(x^+) \leq f(x^+ + \varepsilon)$, für $t \rightarrow \infty$.

Das Gradientenverfahren kann in zwei Fällen als geeignete Methode der numerischen Suche eingesetzt werden. Erstens, falls ein Problem gegeben ist, bei dem die lokal optimalen Punkte bezüglich der Zielfunktion nahe am globalen Optimum sind. Zweitens bei Vorliegen einer guten Lösung, die lediglich lokal zu verbessern ist.

Für den in dieser Arbeit vorliegenden Anwendungsbereich werden einige Eigenschaften des Gradientenabstiegs relevant, die in theoretischen Betrachtungen typischerweise nicht zu beachten sind. Die detaillierte Einführung in die Anwendung der Schaltungsoptimierung folgt in Kapitel 4, der Vorgriff auf Teilaspekte an dieser Stelle dient lediglich der Charakterisierung des Gradientenabstiegsverfahrens.

Die Optimierung von Schaltungsparametern kann allgemein als eine Minimierung einer Abbildung $f: \mathbb{R}^n \rightarrow \mathbb{R}$ beschrieben werden. Eine a priori Diskretisierung der Parameter ist nicht möglich, da dies der Nutzung des Gradienten widerspräche. Diese Parameter wirken unterschiedlich stark auf die Zielfunktion ein und sind zudem sehr verschieden skaliert. Während die Versorgungsspannung bei Zahlenwerten von 10^0 liegt, sind laterale Geometriegrößen bei 10^{-6} zu finden. Da diese Größen über ihren Wertebereich einen ähnlich hohen Einfluss auf die Zielfunktion ausüben können, ergibt sich folgende Problematik. Obwohl die Versorgungsspannung über ihren Variationsbereich die Verlustleistung stark beeinflusst, ergibt sich die partielle Ableitung im Vergleich zu Transistorgeometrien nahezu zu Null. Ein einfaches

Gradientenabstiegsverfahren würde somit in einer zeitlich begrenzten Optimierung primär die Geometriegrößen anpassen und die Versorgungsspannung fast unverändert beibehalten. Alternativ kann eine Gewichtung der Parameter und somit der Schrittweite η in den verschiedenen Dimensionen des Raumes vorgenommen werden. Dies erfordert wiederum spezifisches Wissen, um den Einfluss der einzelnen Größen abschätzen zu können.

Weiterhin ist die Optimierung digitaler Schaltungen von scharfen Grenzen im Parameterraum geprägt, die korrekt funktionierende von fehlerhaften Designs trennen. Diese Grenzen sind nicht explizit formulierbar, d.h. sie können nicht a priori bestimmt und die Optimierung entsprechend begrenzt werden. Daher ist es erforderlich, die Bewertung fehlerhafter Parametersätze mit einem ausreichend großen additiven Strafterm zu belegen, um eine Präferenz korrekter Schaltungen herbeizuführen. Die Überprüfung der Korrektheit der Schaltung ist erst im Anschluss an die numerische Simulation möglich, so dass zur Berechnung des Gradienten der Zielfunktion die direkte Gradientenmethode anzuwenden ist, d.h. der Gradient wird über den Differenzenquotienten genähert. Ansätze [BHSV81, NRSVT88, Vis97, CCH⁺98] zur Bestimmung des Gradienten intern in SPICE, z.B. mittels der Sensitivitätsanalyse, sind bei einer nachfolgenden Funktionsprüfung nicht durchführbar.

Der in der Bewertung der Schaltung verwendete Strafterm liegt typischerweise um Größenordnungen über der ursprünglichen Zielgröße. Dies führt folglich zu stark differierenden Gradienten im Bereich korrekter Schaltungen, verglichen mit Parametersätzen am Rande dieses Bereiches. Dies ist von Bedeutung für die Wahl der Schrittweite. Soll die Weite konstant sein, so zeigt sich ein Konflikt bezüglich einer zufriedenstellenden Konvergenzgeschwindigkeit im Bereich korrekter Schaltungen und einer stabilen Suche an dessen Rand. Somit ist eine Adaption der Schrittweite für eine erfolgreiche Suche erforderlich. Bei den hier durchgeführten Optimierungen ist die Schrittweite jeweils so gewählt worden, dass sie für jeden Parameter ein Hundertstel der Breite des vordefinierten Intervalls nicht überschreitet und bei mindestens einem Parameter erreicht. Dies sorgt für eine Adaption der Schrittweite, die einen akzeptablen Kompromiss zwischen Stabilität und Geschwindigkeit der Optimierung aufrecht erhält.

3.3 Evolutionäre Algorithmen

Gedankliche Anleihen bei der Natur vorzunehmen, ist seit jeher ein probates Mittel für den Menschen neue Ideen zu entwickeln. Gerade bei der anwendungsorientierten Problemlösung ist die Nachbildung von Eigenschaften und Verhaltensweisen der Natur ein erfolgreiches Re-

zept. In gleicher Weise adaptieren evolutionäre Algorithmen (EA) Mechanismen der biologischen Evolution [Sch95].

Die Reproduktion und die Selektion, die von Charles Darwin als wichtige Prinzipien der biologischen Evolution erkannt wurden, stellen die Basis für die evolutionären Algorithmen dar. Sie werden für die Parametervariation und Steuerung der Suche verwendet.

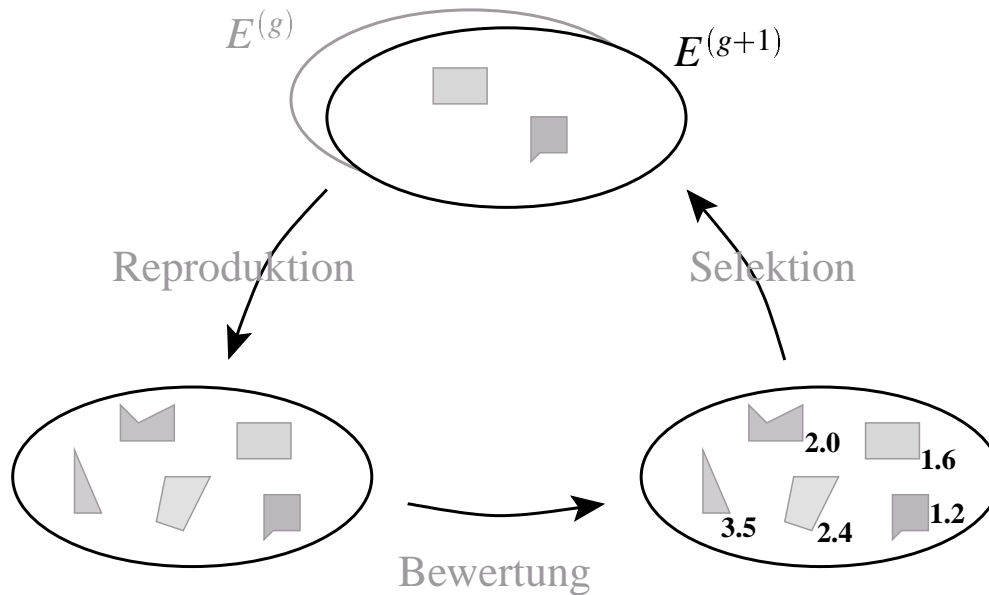


Abbildung 3.2: Schema eines evolutionären Algorithmus.

Evolutionäre Algorithmen verfolgen im Allgemeinen einen populationsbasierten Ansatz, d.h. statt eines einzigen Punktes im Parameterraum wird eine Gruppe von Punkten betrachtet. Diese Gruppe wird als Population und die einzelnen Elemente werden in Analogie zu dem biologischen Vorbild als Individuen bezeichnet. Es handelt sich um ein iteratives Verfahren, das ausgehend von einer Population $E^{(g)}$ über einen Zyklus von Reproduktion und Selektion die Nachfolpopulation $E^{(g+1)}$ bestimmt. Der Iterationsindex g bezeichnet in diesem Zusammenhang die Generation. In einem Zyklus werden aus der Population durch den Reproduktionsoperator die so genannten Nachkommen $N^{(g)}$ bestimmt. In diesem Prozess werden die Elemente der Population $E^{(g)}$ als Eltern bezeichnet. Aus den Nachkommen und je nach Variante des eingesetzten Verfahrens auch aus der Gruppe der Eltern wird durch Selektion die nachfolgende Population $E^{(g+1)}$ gebildet. Dieser Zyklus wiederholt sich bis ein vorher definiertes Abbruchkriterium erfüllt ist (vergleiche Abbildung 3.2).

Ein einfaches Beispiel zur Illustration des Vorganges ist in Abbildung 3.3 gegeben. Ein Potenzial in einer Ebene sei hier mit Hilfe einer Population der Größe Zwei zu optimieren.

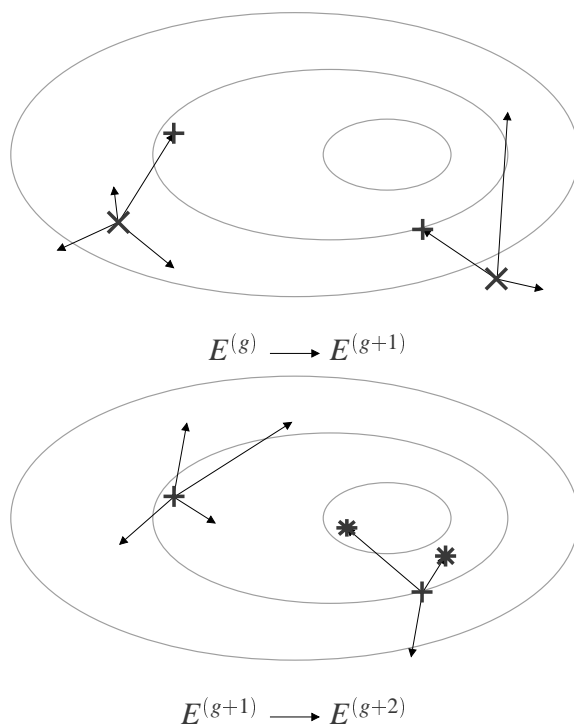


Abbildung 3.3: Beispiel für zwei Schritte einer evolutionären Optimierung; Populationsgröße Zwei, mit den Generationen g : X, $g+1$: + und $g+2$: *

Im folgenden Kapitel sollen die Operatoren, welche die Reproduktion und Selektion realisieren, konkretisiert werden. Hierzu ist vorab die Repräsentation der Individuen zu klären. Ein Element des Parameterraumes in \mathbb{R}^n kann sowohl direkt als reellwertiger Vektor als auch in Form eines Bitstrings, d.h. eines Tupels mit Elementen aus 0 und 1 gespeichert werden. Im Fall eines Bitstrings kommt der Kodierung eine entscheidende Bedeutung zu, da durch sie die Wahrscheinlichkeit für die Reproduktion der Elemente beeinflusst sein kann. Die Repräsentation der Elemente hat starken Einfluss auf die Wahl der Operatoren. In Analogie zum biologischen Vorbild der evolutionären Algorithmen wird die Repräsentation oder Kodierung der Elemente als Genotyp bezeichnet. Die Darstellung in Parameterraum wird als Ausprägung des Individuums gesehen und entsprechend Phänotyp genannt. Da der Zusammenhang zwischen Geno- und Phänotyp bei reellwertigen Repräsentationen in den meisten Fällen eine

identische Abbildung ist, kommt dieser Differenzierung bei der Verwendung von Bitstrings die größere Bedeutung zu.

3.3.1 Operatoren evolutionärer Algorithmen

Die zentralen Mechanismen der evolutionären Algorithmen sind die Reproduktion und die Selektion, die durch geeignete Operatoren zu realisieren sind.

Die Reproduktion generiert aus einer Menge von Elementen weitere, neue Elemente, so dass die Nachkommen eine gewisse Affinität zu den Eltern-Elementen aufweisen. Hierzu werden die Mutation und die Rekombination verwendet. Die Mutation ist ein unärer Operator, der ein Eltern-Element im Fall reellwertiger Kodierung mit Gauß'schem Rauschen modifiziert (siehe Seite 25). Bei der Verwendung eines Bitstrings werden per Zufall einzelne Bits verändert. Die Rekombination ist ein typischerweise binärer Operator, um Eigenschaften verschiedener Elemente zu vereinen. Bei Vektoren kann das arithmetische Mittel zweier Punkte einen Nachkommen liefern (intermediäre Rekombination). Alternativ kann jede Komponente des Vektors zufällig von einem der beiden Eltern-Elemente kopiert werden (diskrete Rekombination). Für Bitstrings wird üblicherweise ein Cross-Over Operator eingesetzt, bei dem der String an beliebigen Stellen aufgetrennt wird und die Segmente ausgetauscht werden.

Für die Selektion muss eine Bewertung der Individuen in Form der Zielfunktion vorliegen. Diese Bewertung wird auch als Fitness bezeichnet. Es existiert eine Vielzahl von Ansätzen zur Realisierung der Selektion, die die Charakteristik des evolutionären Algorithmus' als Ganzes stark beeinflussen. Beispiele sind die μ, λ -Selektion, bei der die Elemente mit der besten Bewertung in die nachfolgende Population übernommen werden. Das Turnier-Verfahren beruht auf einer zufälligen Auswahl zweier Individuen, von denen das Bessere in die nächste Generation übernommen wird. Dieser Vorgang wiederholt sich, bis die Population komplettiert ist. Die μ, λ -Selektion kann den Suchbereich stark eingrenzen und zu einer pfadorientierten Suche führen. Hierbei wird von einem hohen Selektionsdruck gesprochen, da alternative, schlechtere Lösungen sofort eliminiert werden. Dem gegenüber übt das Turnier-Verfahren einen schwächeren Selektionsdruck aus und fördert eine höhere Diversität der Suche (volumenorientierte Suche), die mit einer geringeren Konvergenzgeschwindigkeit verbunden ist [Bäc96]. In Anbetracht der vorliegenden Anwendung, wird das Turnier-Verfahren zur Selektion verwendet, um den Parameterraum für das Schaltungsdesign nicht einzuschränken.

3.3.2 Evolutionsstrategie und Genetische Algorithmen

Die Umsetzung der evolutionären Prinzipien existiert in Form der von Holland [Hol75] präsentierten genetischen Algorithmen (GA) und in Form der Evolutionsstrategien (ES), vorgestellt von Rechenberg [Rec73] und von Schwefel [Sch95]. Obwohl die Grenzen zwischen beiden Ansätzen mehr und mehr verwischen, so lassen sich doch charakteristische Aspekte beider ausmachen.

Genetische Algorithmen arbeiten auf Binärstrings als Repräsentation der Individuen. Bei der Reproduktion ist die Rekombination durch ein Cross-Over der dominante Operator. Die Mutation spielt aufgrund einer gering gewählten Wahrscheinlichkeit eine untergeordnete Rolle. Die Evolutionsstrategie hingegen verwendet reellwertige Vektoren und die Mutation als zentralen Mechanismus der Reproduktion. Der starke Einfluss der Mutation erlaubt tendenziell kleinere Populationen im Vergleich zu genetischen Algorithmen. Die ursprüngliche, von Rechenberg vorgestellte Evolutionsstrategie, betrachtet nur ein einzelnes Element, das allein durch Mutation variiert wird. Falls hierdurch eine Verbesserung eintritt, wird das neue Element übernommen. Dieses Vorgehen wird als 1+1-Evolutionsstrategie bezeichnet.

Die Evolutionsstrategien werden weiter differenziert bezüglich der Berücksichtigung der Eltern-Elemente bei der Selektion. Bei so genannten Plus-Strategien werden die Elemente der Nachfolgeneration $E^{(g+1)}$ sowohl aus der Eltern-Population $E^{(g)}$ als auch aus deren Nachkommen $N^{(g)}$ ermittelt. Bei der Komma-Strategie schlagen sich lediglich die Elemente der Nachkommen in der Nachfolgepopulation nieder, d.h. jedes Individuum existiert genau für die Zeit einer Generation. Wird mit μ die Größe der Eltern-Population und mit λ die der Nachkommen beschrieben ($\mu = |E^{(g)}|$, $\lambda = |N^{(g)}|$), so bezeichnet $(\mu+\lambda)$ -ES eine Plus-Strategie und (μ,λ) -ES eine Komma-Strategie. Die Komma-Strategie erlaubt eine größere Diversität der Suche, da Verschlechterungen der momentanen Suchposition unabhängig von dem verwendeten Selektionsmechanismus erlaubt werden. Die Plus-Strategie hingegen führt zu einer stabileren Suche, da einmal gefundene Lösungen in weiteren Generationen dauerhaft existieren können.

3.3.3 Evolutionsstrategie

In diesem Kapitel wird die Evolutionsstrategie konkretisiert, wie sie im Rahmen dieser Arbeit in den noch folgenden Anwendungen zum Einsatz kommt.

Es wird eine (μ, λ) -ES verwendet. Zur Repräsentation wird eine identische Abbildung im \mathbb{R}^n „verwendet“, so dass Geno- und Phänotyp nicht unterschieden werden müssen. Die Reproduktion erfolgt sowohl mittels der Rekombination als auch durch Mutation, deren beider Verwendung mit einer Einsatzwahrscheinlichkeit belegt werden kann. Eltern-Elemente können direkt oder nach Durchmischung durch Rekombination verwendet und anschließend mutiert oder unverändert in die Nachkommen-Population übernommen werden. Falls die Wahrscheinlichkeit für beide Operatoren kleiner Eins ist, können somit trotz Komma-Strategie Elemente für mehr als eine Generation existieren.

In den ersten Evolutionsstrategien wurde zur Mutation ein bezüglich der Population und der Dimension globaler Strategieparameter zur Bestimmung der Varianz der Gauß'schen Zufallsverteilung gewählt. Dem entgegen wird hier jedem Individuum ein Strategievektor $\vec{\sigma}$ zugeordnet, der jeweils mit kopiert und über die Reproduktionsoperatoren variiert wird. Somit ist ein Individuum als Paar $(\vec{x}, \vec{\sigma})$ aus Position \vec{x} und Strategieparameter $\vec{\sigma}$ definiert. Individuelle Strategieparameter ermöglichen, die lokale Topologie der Zielfunktion zu berücksichtigen. Diese Eigenschaft wird mit Selbstadaption [Bäc96] bezeichnet. Um die Selbstadaption nutzen zu können, ist jedoch die Verwendung einer Komma-Strategie erforderlich, da bei der Plus-Strategie fehlerhafte Strategieparameter die Anpassung der Suche über mehrere Generationen hinweg negativ beeinflussen können.

Rekombination

Die Rekombination wird für Positions- und Strategieparameter unterschiedlich durchgeführt. Die Positionen der Individuen werden durch intermediäre Rekombination, die Strategieparameter durch diskrete Rekombination zusammengefasst. Die beiden Eltern-Elemente werden zufällig aus der Eltern-Population $E^{(g)}$ gewählt. Die Ortsvektoren der Eltern-Elemente seien mit \vec{x}_p und \vec{x}_q bezeichnet und deren Strategieparameter mit $\vec{\sigma}_p$ und $\vec{\sigma}_q$. P sei eine gleichverteilte Zufallsvariable in $[0; 1[$.

$$x_i = \frac{x_{p_i} + x_{q_i}}{2} \quad (3.5)$$

$$\sigma_i = \begin{cases} P < 0.5 : & \sigma_{p_i} \\ P \geq 0.5 : & \sigma_{q_i} \end{cases} \quad (3.6)$$

Mutation

Für die Mutation wird ein Ansatz mit individueller Schrittweitenadaption verwendet. Der Einsatz des Strategievektors $\vec{\sigma}$ erlaubt die Skalierung der Schrittweite für jedes Individuum in jeder Dimension.

Der Ausdruck $N(0,1)$ bezeichne die Ziehung einer Zufallszahl gemäß einer Normalverteilung mit Erwartungswert 0 und Varianz 1. Mit $N_i(0,1)$ ist dann ein erneutes Ziehen für jeden Wert für i bestimmt. Sei ein Individuum I mit $I = (\vec{x}, \vec{\sigma})$ gegeben, so ergibt sich durch die Mutation ein Individuum I' mit $I' = (\vec{x}', \vec{\sigma}')$ wie folgt [Bäc96]:

$$\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \quad (3.7)$$

$$x'_i = x_i + N(0, \sigma'_i) \quad (3.8)$$

Die Strategieparameter werden mittels eines multiplikativen, logarithmisch normal verteilten Zufallsprozesses variiert. Anschließend dienen die mutierten Werte zur Justierung des additiven, normal verteilten Prozesses, mit dem die Elementposition verändert wird. Der globale Faktor $\exp(\tau' \cdot N(0,1))$ hat einen ganzheitlichen Einfluss auf den Strategievektor und erhält die Dimension des Suchraumes. Das heißt es wird verhindert, dass der Algorithmus den Suchraum auf einem Unterraum eingrenzt. Der lokale Faktor $\exp(\tau \cdot N_i(0,1))$ erlaubt die individuelle Änderung der Schrittweite σ_i . Die Faktoren sind wie von Schwefel [Bäc96] vorgeschlagen gewählt (3.9, 3.10) und sind durch die Dimension n des Suchraumes bestimmt.

$$\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1} \quad (3.9)$$

$$\tau' = \left(\sqrt{2n}\right)^{-1} \quad (3.10)$$

Um ein Fortschreiten der Suche zu gewährleisten, muss die Schrittweite σ_i größer als Null sein. Daher wird eine minimale Schwelle ε_σ definiert, die nicht unterschritten werden darf. Dies ist in jedem Variationsschritt zu überprüfen.

$$\sigma'_i < \varepsilon_\sigma \Rightarrow \sigma'_i := \varepsilon_\sigma \quad (3.11)$$

Diese Punkte zusammengefasst ermöglichen, für eine Evolutionsstrategie die Strategieparameter intern, ohne exogene Informationen, zu bestimmen. Einzige Ausnahme stellt die minimale Schwelle ε_σ dar, die jedoch nicht problemspezifisch, sondern abhängig von der Implementierung, genauer der internen Zahlendarstellung, zu wählen ist.

Selektion

Im Rahmen der Selektion werden nur die neu generierten Nachkommen berücksichtigt. Wie bereits erwähnt, funktioniert der Mechanismus der Selbst-Adaption nur mit einer Komma-Strategie, in der Fehlinformationen schnell entfernt werden. Für die Wahl der Parameter der μ, λ -ES sind folgende Punkte zu beachten:

Das Verhältnis λ/μ ist bestimmend für den Selektionsdruck. Für eine ausreichende Diversität der Suche muss dieses Verhältnis genügend groß sein. Für den erforderlichen Informationserhalt und ein Funktionieren der Selbstadaption hat sich nach Schwefel ein Wert von $\lambda/\mu \approx 7$ bestimmt. Auf der anderen Seite muss μ auch deutlich größer als Eins sein [Bäc96]. Gebräuchliche Werte sind $\mu = 15$ und $\lambda = 100$.

Initialisierung

Die Startwerte für eine Evolutionsstrategie werden im Hinblick auf die Rahmenbedingungen des Optimierungsproblems gewählt. In Bezug zu der vorliegenden Anwendung der Schaltungsoptimierung lassen sich zwei Szenarien unterscheiden. Im ersten Fall sei bereits eine gute Lösung des Problems bekannt, für die nur eine Verbesserung in begrenzter räumlicher Nähe erwartet wird. Dann kann diese Lösung verwendet werden, um die μ Elemente für die Startpopulation jeweils durch Mutation zu generieren. Im zweiten Fall sei kein Problemwissen vorhanden. Die Anfangspopulation wird dann durch ziehen gleichverteilter Zufallswerte bestimmt. In beiden Fällen werden die Strategieparameter σ_i auf $1/4$ der Intervallbreite in der Dimension i gesetzt.

Restriktionen

Die in dieser Arbeit betrachtete Optimierungsproblematik erfordert die Berücksichtigung sowohl expliziter als auch impliziter Restriktionen. Elemente oder Individuen, die die Restriktionen erfüllen, werden hier als zulässige Lösungen bezeichnet. Explizite Restriktionen treten in Form von Intervallgrenzen für die Parameter auf. Deren Einhaltung wird bereits bei der Generierung neuer Elemente kontrolliert und führt zu einer Wiederholung des Vorganges, bis ein vorläufig zulässiges Element erzeugt wird. Implizite Restriktionen berücksichtigen Vorgänge und Ergebnisse der Simulation, sind also im Nachhinein zu bestimmen. In der vorliegenden Implementierung können diese Randbedingungen sowohl in Form expliziter algebraischer

Ausdrücke als auch als Rückmeldungen des Simulators gegeben sein. Im Falle der Schaltungsoptimierung können dies ein Schaltvorgang nach einer definierten Latenzzeit oder ein Konvergenzproblem im Simulator für den konkreten Parametersatz sein.

Randbedingungen eines Optimierungsproblems [SC95] können als absolute oder harte Restriktionen (hard constraints) beschrieben sein, deren Erfüllung zwingend erforderlich ist. Diese Elemente werden, in Bezug auf die Zielfunktion, mit der schlechtest möglichen Bewertung belegt und fallen im folgenden Selektionsschritt heraus. Dass heißt die Intensität der Verletzung der Restriktion wird nicht berücksichtigt und mehrere nicht zulässige Lösungen sind nicht mehr vergleichbar. Alternativ können Randbedingungen als weiche Restriktionen (soft constraints), deren Verletzung die Bewertung des simulierten Prozesses an sich nicht außer Kraft setzt, definiert sein. In diesem Fall wird der Grad der Verletzung in Form eines Strafterms (punishment) zur Zielfunktion hinzugefügt. Diese modifizierte Zielfunktion ist dann von der Form $f'(\vec{x}) = f(\vec{x}) + p(\vec{x})$. Somit existiert eine Ordnung auch auf den (bezüglich der soft constraints) nicht zulässigen Lösungen des Optimierungsproblems, so dass solche Lösungen vergleichbar werden und eine gerichtete Suche zur Erfüllung der Restriktion ermöglicht wird. Ist der Zahlenwert des Strafterms groß gegenüber dem der Zielfunktion und ist die Menge der zulässigen Lösungen nicht leer, so wird das Optimum in dieser Menge liegen. Ein Suchverfahren kann dann trotz weicher Restriktionen eine zulässige Lösung bestimmen.

In der Anwendung zur Schaltungsoptimierung werden hier, wenn möglich, weiche Restriktionen verwendet. Insbesondere im Digitalschaltungsdesign ist die Zahl der Lösungen mit einer korrekten Schaltfunktion klein gegenüber dem gesamten Parameterraum. Die Verwendung absoluter Restriktionen könnte dann den evolutionären Algorithmus in eine ungerichtete Zufallssuche verwandeln. Bei sämtlichen in dieser Arbeit vorgestellten Beispielen führt die Nutzung von weichen Grenzen zu einer zulässigen Lösung, die alle Kriterien erfüllt. Als Strafterm wird ein linearer additiver Term zur Einbeziehung des Verletzungsgrades einer Restriktion verwendet.

$$f'(\vec{x}) = f(\vec{x}) + \sum_i \nu_i p_i(\vec{x}) \quad (3.12)$$

Mit $p_i(\cdot)$ sei der Grad, zu dem die Restriktion i nicht erfüllt ist, beschrieben, ν_i ist dann der jeweilige Skalierungsfaktor. Für den evolutionären Algorithmus wirksam ist dann die modifizierte Zielfunktion f' .

Kapitel 4

Automatisiertes Schaltungsdesign

Der Entwurf integrierter Schaltungen auf Transistorebene beginnt mit der Strukturierung, d.h. mit der Festlegung der Bauteile und deren Verbindungen. Im zweiten Schritt sind diese Bauteile zu dimensionieren, um die gewünschte Funktion gemäß gegebener Spezifikationen zu erfüllen.

Der Begriff Schaltungsdesign beschreibt die Strukturierung einer Schaltung sowie die Dimensionierung dieser Schaltungsstruktur. In dieser Arbeit ist das Schaltungsdesign auf den zweiten Teil, die Parametrisierung reduziert. Erst diese Konkretisierung einer Schaltung liefert ein unmittelbar bewertbares Modell. Die Bestimmung einer Schaltung, die irgendwie gearteten, vordefinierten Anforderungen genügt, ist der Prozess des Schaltungsdesigns. Die Automatisierung dieses Prozesses, auf der Basis gegebener Randbedingungen und Ziele, ohne weitere Eingriffe eines Benutzers wird in diesem Kapitel vorgestellt. Der Nutzen im Rahmen der Schaltungstechnik und Technologieanalyse wird präsentiert und an Beispielen erläutert.

4.1 Designspezifikation

Für die Automatisierung einer Schaltungsoptimierung ist eine detaillierte Zieldefinition erforderlich. Insbesondere die Beschreibung der Nebenbedingungen ist von enormer Bedeutung. Im Digitalschaltungsdesign sind neben der Erfüllung einer gegebenen logischen Funktion mögliche Vorgaben zu Schaltverhalten, Stabilität und Robustheit zu berücksichtigen. In Bezug auf das Schaltverhalten (Timing) können minimale und maximale Latenzzeiten relevant sein. Stabilitätsanforderungen fließen in Vorgaben zur Eindeutigkeit der Zustandsübergänge

ein, die kurzzeitige falsche bzw. undefinierte Zustände durch Hazards zulassen oder verbieten. Im Punkt Robustheit sind die I/O-Spezifikationen zu beachten, da z.B. ein Low-Potenzial nicht nur bei 0 Volt vorliegt, sondern in einem gewissen Intervall definiert ist.

Für eine Schaltungsanalyse auf Transistorebene ist die Betrachtung im Zeitbereich, in einer transienten Simulation, erforderlich, d.h. Testmuster einer Schaltung stellen Schaltvorgänge am Eingang zu aufeinander folgenden Zeitpunkten dar. Die Anforderungen an Timing und Stabilität, sind während der Umschaltvorgänge zu untersuchen. Die Kontrolle der logischen Funktion fokussiert sich auf den quasi statischen Bereich, in dem Einschwingvorgänge nach dem Umschalten abgeschlossen sind.

In der Timing-Analyse, ist in den meisten Fällen die Einhaltung einer maximalen Reaktionszeit zu prüfen. Ist der kritische Pfad einer Schaltung bekannt, so kann die Untersuchung auf die davon betroffenen Ausgangsgrößen beschränkt werden. Da die Verzögerungszeit (Delay) von den Eingangsgrößen und deren Dynamik abhängt, muss das Eingangssignal als Testmuster der Schaltung systematisch entwickelt werden. Dies wird im folgenden Abschnitt zur Überprüfung der logischen Funktion erörtert.

Der Entwurf der Testmuster bezüglich der logischen Funktion erfolgt rein auf der Logikebene. Zur Verifikation einer Schaltungsfunktion gemäß einer gegebenen Wahrheitstabelle, ist die Korrektheit der Schaltung für jeden Eingangszustand zu zeigen. Zudem sind sämtliche Zustandsübergänge (Transienten) zur Erfassung der Schaltungsdynamik einzubeziehen und im Falle getakteter Schaltungen die Erhaltung des Zustandes zu überprüfen. Die Menge der Testfälle wächst quadratisch mit der Zahl der Zustände an (Zustände: n , Zustandsübergänge: $1 + \dots + (n-1) = \frac{n^2-n}{2}$). Durch die Berücksichtigung von Symmetrien bezüglich der Eingangszustände und des Schaltverhaltens lässt sich jedoch der Zustandsraum und somit die Zahl der erforderlichen Testmuster reduzieren. Trotzdem ist bei komplexeren Schaltungen eine vollständige Verifikation der logischen Funktion auf Transistorebene beim heutigen Stand der Technik nicht in vertretbarem Zeitrahmen durchführbar. Hier ist Expertenwissen über das System bzw. Modell erforderlich, um den Parameterraum so einzuschränken, dass die Erfüllung der Funktion garantiert ist oder aus Tests eines Teilsystems abgeleitet werden kann.

Die Ausgänge einer Schaltung können bei gewissen Zustandsübergängen auf der logischen Ebene konstant sein. Laufzeitunterschiede innerhalb der Schaltung können jedoch zu kurzfristigen Signaländerungen an diesen Ausgängen, zu so genannten Hazards führen. Stabilitätsanforderungen zur Unterdrückung dieser Schwankungen sind dann nicht im quasi statischen Bereich, sondern zur Zeit des Schaltvorganges abzutesten. Dies erfolgt wie bei der logischen Verifikation durch Definition einer Schwelle.

Für eine Analyse der Robustheit einer Schaltung sind die Eingangssignale als analoge Werte zu betrachten. Dass heißt Betriebsspannung und Masse werden die High- und Low-Potenziale nicht mehr vorausgesetzt. Die Spezifikation einer Schaltungstechnik definiert zur Repräsentation von Booleschen Werten keine punktuellen Spannungen sondern jeweils Spannungsintervalle (voltage noise margin). Eine Schaltung, die dieser Spezifikation genügen soll, muss somit die Anforderungen an Timing, Stabilität und die logische Funktion für beliebige Spannungen in den gegebenen Intervallen erfüllen. Ist der Einfluss eines Eingangssignals auf die betrachtete Größe monoton, so genügt eine Analyse an den Intervallgrenzen des Signals für eine Verifikation. Ist die Monotonie nicht gegeben, ist im Allgemeinen keine Verifikation möglich. Erst spezifisches Systemwissen über die Wirkung der Parameter auf die Zielgröße erlaubt die Berücksichtigung der Robustheit [AGW94]. Für den allgemeinen Fall kann beispielsweise die Einhaltung der Spezifikation durch zufällige Wahl der Eingangssignalschwankung validiert werden. Sollen statt einer schlichten Prüfung der Empfindlichkeit gegenüber Störungen am Eingang auch Parameterschwankungen der Technologie berücksichtigt werden, ist nur noch eine Validierung der Schaltung möglich. Diese Parameter können einerseits Schwankungen der Geometrien, verursacht durch Abbildungsungenauigkeiten bei der Maskenherstellung und der Belichtung, sein. Andererseits sind Dotierungen und somit die Schwellenspannungen der Transistoren nur mit begrenzter Genauigkeit herstellbar. Dieser Einfluss, der in der klassischen CMOS Technik nahezu vernachlässigbar ist, gewinnt mit zunehmender Miniaturisierung und der Verwendung von Nano-Bauelementen an Bedeutung. Derartige Streuungseffekte werden in Zukunft, ähnlich der Matchinganalyse [OGS99] im Analogbereich, auch für digitale Schaltungen zu berücksichtigen sein.

Robustheitsanalysen sind zeitintensiv, da die Zahl der Eingangszustände exponentiell mit der Zahl n der Eingänge wächst (2^n) und für jeden Zustand Schwankungen an jedem Eingang betrachtet werden müssen. Somit wächst bereits der Aufwand für die Validierung oder Verifikation eines Zustandes exponentiell mit der Zahl der Eingänge. Sei oben erwähnte Monotoniebedingung gegeben, so dass eine Verifikation möglich ist, so sind hierzu 2^{2n} Simulationen erforderlich. Bei einer Schaltung mit 10 Eingangssignalen wären bereits über eine Million Simulationen zur Bewertung eines Schaltungsdesigns nötig.

Neben der Überprüfung der Spezifikation werden in der Regel weitere Größen, wie die Verlustleistung zur Bewertung des Schaltungsdesigns, aus der Simulation bestimmt. Diese Größen sind stark von der Wahl der Eingangsmuster abhängig. Sind über die Umgebung, in die das System integriert werden soll, keine spezifischen Informationen verfügbar, so kann von einer Gleichverteilung der Eingangszustände und Zustandsübergänge ausgegangen werden, um

keine falschen Präferenzen zu entwickeln. Sind weitere Informationen zu den Zustandswahrscheinlichkeiten bekannt, könnten diese bei der Generierung der Testmuster berücksichtigt werden. Eine exakte Umsetzung einer gegebenen Wahrscheinlichkeitsverteilung wird jedoch in den meisten Fällen nicht möglich sein.

4.1.1 Beispiel: 1-Bit Volladdierer

Im Folgenden sollen die verschiedenen Aspekte der Designspezifikation anhand eines Beispiels erläutert werden. Als Beispiel dient ein 1-Bit Volladdierer, der einerseits einfach genug ist, um in seiner Gesamtheit erfasst zu werden. Andererseits ist er ausreichend komplex, um sämtliche Punkte des Designs aufzuzeigen und in späteren Betrachtungen wieder Berücksichtigung zu finden [TBG00].

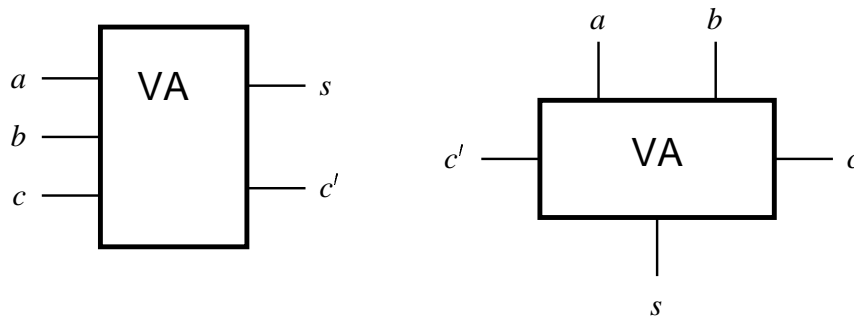


Abbildung 4.1: Symbol des 1-Bit Volladdierers in zwei Varianten.

Ein Volladdierer (Abbildung 4.1) ist eine Boolesche Abbildung $f: \{0,1\}^3 \rightarrow \{0,1\}^2$. Die Parameter seien mit a , b und c bezeichnet und repräsentieren drei Bits, die additiv zu verknüpfen sind. Die Eingangsgröße c bezeichne hierbei das Übertragsbit (Carry-Bit) des nächst niederwertigen Bits. Das Ergebnis dieser Funktion wird durch das Summen-Bit s und den Übertrag c' charakterisiert. Die Gleichungen (4.1, 4.2) und die zugehörige Wahrheitstabelle 4.1 konkretisieren die Boolesche Abbildung. Die Unterscheidung von a und b einerseits und c andererseits ist nicht aus der logischen Funktion heraus begründet, sie ist jedoch zweckmäßig für die weiteren Betrachtungen.

$$s = a \oplus b \oplus c \quad (4.1)$$

$$c' = ab + ac + bc \quad (4.2)$$

Tabelle 4.1: Wahrheitstabelle des 1-Bit Volladdierers.

a	b	c	c'	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

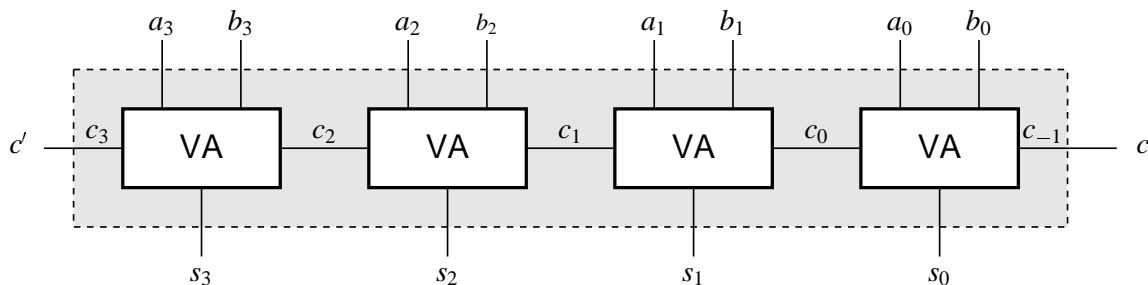


Abbildung 4.2: Ripple-Carry Addierer.

Die Eingangsgrößen a und b können als gleichwertig angesehen werden, da die Zustände $(a, b, c) = (0, 1, \cdot)$ und $(1, 0, \cdot)$ identisch in Bezug auf die Schaltfunktion sind und die beiden Größen semantisch äquivalent sind. Die Eingangsgröße c ist auf der logischen Ebene gleichwertig zu den beiden ersten, jedoch besteht ein semantischer Unterschied. Bei der Realisierung einer mehrstufigen Addition in Form eines Ripple-Carry Addierers (Abbildung 4.2) wird das Carry-Bit von Stufe zu Stufe gereicht, während a und b sofort vorliegen. Damit ist die Propagierung der Signaländerung vom eingangsseitigen Carry c zum ausgangsseitigen c' die kritische Größe im Design.

Unter Ausnutzung der Symmetrie zwischen a und b reduziert sich der Zustandsraum von acht auf die sechs Zustände $(0, 0, 0)$, $(0, 0, 1)$, $(1, 0, 0)$, $(1, 0, 1)$, $(1, 1, 0)$ und $(1, 1, 1)$ (vergleiche Abbildung 4.3). Damit verringert sich die Zahl der Zustandsübergänge von 56 auf 30 für einen statischen Addierer und von 64 auf 36 für ein dynamisches System. Dieses recht ein-

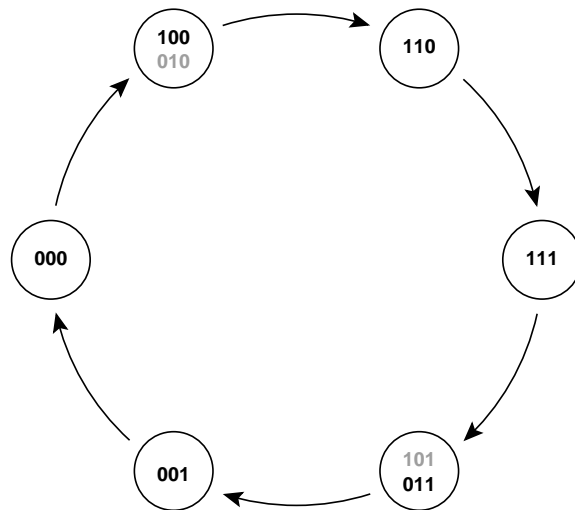


Abbildung 4.3: Zustandsübergänge zur Überprüfung der logischen Funktion eines Volladdierers. Die Bezeichnung der Zustände gibt die Eingangsgrößen a , b und c , in dieser Reihenfolge, wieder. Anfangs- und Endzustand ist '000'.

fache Beispiel zeigt bereits die Grenzen einer Schaltungsanalyse auf. Zum Einen ist die Zahl der erforderlichen Testmuster und damit die Simulationszeit für eine Schaltungsrealisierung bereits recht groß. Zum Anderen sind die Eingangszustände im zeitlichen Verlauf so zu durchlaufen, dass jeder Zustandsübergang genau einmal (oder n -mal) durchlaufen wird. Andernfalls wird eine nicht gleichverteilte Gewichtung in Bezug auf die Verlustleistung bewirkt. Dieses ist bereits in diesem Beispiel nicht zu gewährleisten. Allerdings stellt sich die Frage, ob bzw. inwieweit der Einfluss veränderter Eingangsmuster den stets vorhandenen Fehler zwischen Simulation und realer Schaltung signifikant überschreitet.

Im Rahmen dieser Betrachtung wird aus Gründen der Effizienz und Anschaulichkeit ein Eingangsmuster verwendet, das sämtliche Zustände und sechs Zustandsübergänge abdeckt. Die Zustandsfolge $(0, 0, 0) - (1, 0, 0) - (1, 1, 0) - (1, 1, 1) - (0, 1, 1) - (0, 0, 1) - (0, 0, 0)$ beschränkt sich bei jedem Übergang auf die Änderung eines Bits. Das Eingangssignal ist in Abbildung 4.4 dargestellt. Die hier verwendeten Zeitkonstanten und Spannungsniveaus sind beispielhaft gewählt und würden sich im konkreten Anwendungsfall aus externen Spezifikationen ableiten. Die Zustandsänderungen am Eingang erfolgen in einem zeitlichen Abstand von $5 ns$. Die maximale Verzögerungszeit, bis ein gültiges Ausgangssignal vorliegt sei mit $2 ns$ gegeben und die maximale Abweichung vom definierten Signal mit $20\% \cdot V_{DD}$. Im Falle der beispielhaft

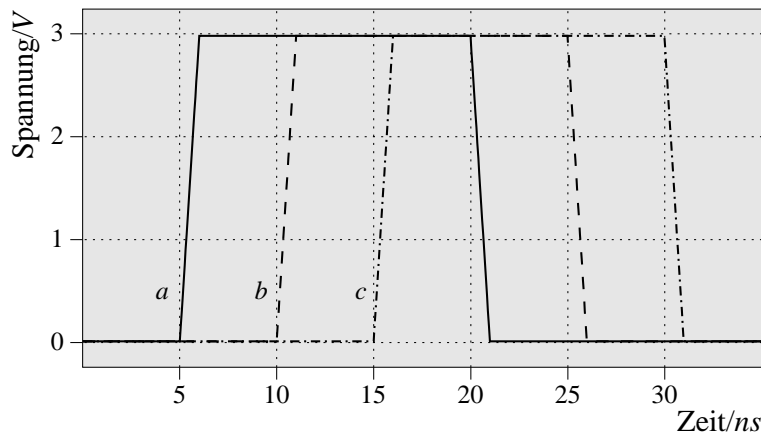


Abbildung 4.4: Eingangssignale des Volladdierers.

verwendeten 3 V Versorgungsspannung entspricht dies kleiner $0,6\text{ V}$ für ein Low-Signal und größer $2,4\text{ V}$ für ein High-Signal.

Die Anforderungen an die logische Funktion und die Verzögerungszeit lassen sich durch Definition unterer bzw. oberer Grenzen beginnend mit 2 ns nach dem Umschalten des Eingangs für die gesamte Dauer eines konstanten Eingangssignals beschreiben. Dies bedeutet bei einem Wechsel in den Zustand $(1, 0, 0)$ zum Zeitpunkt t , dass das Summensignal von $t + 2\text{ ns}$ bis $t + 5\text{ ns}$ größer als $2,4\text{ V}$ und das Carry kleiner als $0,6\text{ V}$ sein muss. Diese Grenzen beschreiben für die Ausgangsgrößen eine Art Schlauch im Zeitbereich, in dem das Signal verlaufen muss. Abbildung 4.5 zeigt diesen Schlauch für die Summe und den Übertrag basierend auf den Eingangssignalen aus Abbildung 4.4.

Zur Gewährleistung der Stabilität am Ausgang wird für den Übertrag die Signalgrenze durchgängig, auch im 2 ns -Bereich nach einem Zustandswechsel definiert, sofern sich der Sollwert nicht ändert.

4.2 Realisierung mit skalaren Optimierungsverfahren

Die Parametrisierung eines Schaltungsdesigns ist ein Problem der Mehrgrößenoptimierung. Typische Zielgrößen sind die Chipfläche, die Verzögerungszeit, die Verlustleistung und die Robustheit einer integrierten Schaltung. Die Optimierung bezüglich mehrerer Kriterien führt

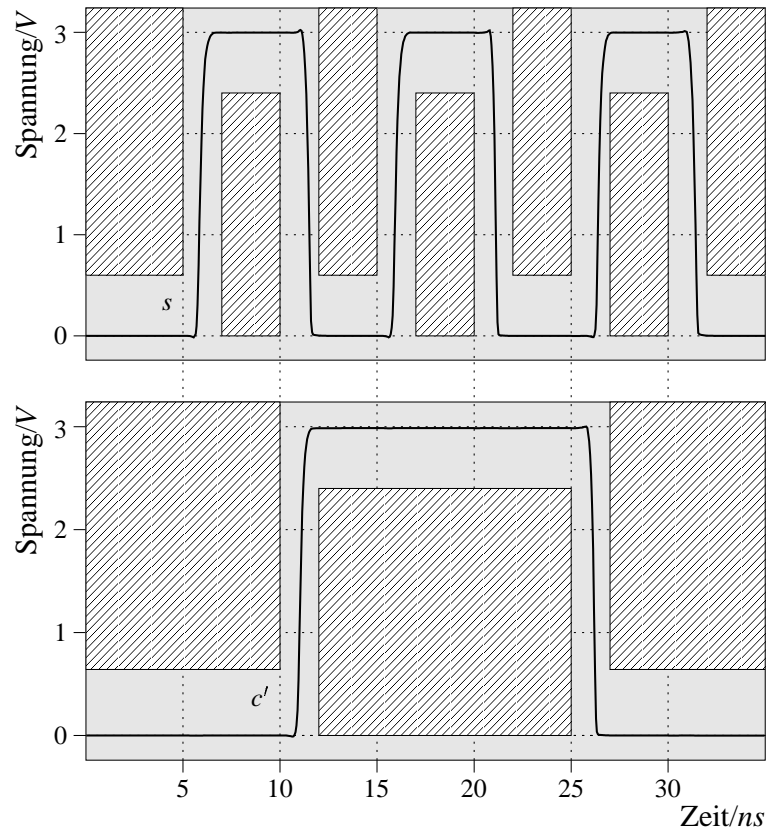


Abbildung 4.5: Ausgangssignal des Volladdierers mit definiertem Schlauch.

nicht zu einer singulären Lösung, sondern zu einer Lösungsmenge. Diese so genannte Pareto Menge umfasst sämtliche optimalen, nicht dominierten Lösungen. Eine detaillierte Darstellung der Mehrgrößenoptimierung findet sich in Kapitel 6. Die Behandlung des Schaltungsdesigns als skalares Optimierungsproblem mit einem eventuell künstlich konstruierten Kriterium, begründet sich in zwei Punkten. Erstens der Tatsache, dass kaum Verfahren zur Mehrgrößenoptimierung existieren. Klassische Verfahren betrachten typischerweise eine Lösung und suchen eine Verbesserung. Dies kann bestenfalls zu einer beliebigen Lösung des Optimierungsproblems führen. Populationsbasierte Ansätze, wie die evolutionären Algorithmen, bieten die Chance, eine Aussage über die Pareto Menge zu erhalten, falls die Lösungsmenge eine gute Überdeckung dieser liefert. Diese Ansätze zeigen jedoch eine Entwicklung der letzten Jahre und haben noch keine Verbreitung in der Praxis gefunden [Sch85, GR87, FF95, SD95, LRS98, ZT99, Zit99, vVL99]. Der zweite Punkt zur Beschränkung auf eine

Eingrößenoptimierung ist durch die Anwendung gegeben. Für eine Systemanalyse kann eine Mehrgrößenoptimierung mit einer Beschreibung der Pareto Menge als Lösung vorteilhaft sein, wie in Kapitel 6 dargestellt ist. Ist jedoch das Ziel einer Optimierung die anschließende Realisierung, in diesem Fall einer Schaltung, ist eine Reduzierung auf eine singuläre Lösung erforderlich.

Die Zahl der zu optimierenden Zielgrößen kann mittels der drei folgenden Ansätze reduziert werden:

Einführung zusätzlicher Restriktionen

Das Ersetzen einer Zielgröße durch eine Restriktion verzichtet auf die Verbesserungsmöglichkeit in dieser Dimension zugunsten der übrigen. Auf diese Weise kann die Zahl der Zielgrößen auf diejenige, die von dem Anwender als die Wichtigste identifiziert worden ist, reduziert werden. Bei gegensätzlichen Zielgrößen wird die Variationsmöglichkeit für die ersetzten Zielgrößen in der Regel bis zur Restriktion hin ausgenutzt. Dies sei an einem Beispiel (siehe Abbildung 4.6) erläutert, in dem Verlustleistung und Verzögerungszeit zu minimieren seien. Das Ersetzen der Verzögerungszeit durch eine obere Schranke, z.B. durch Spezifikationsvorgaben, wird die Minimierung der Verlustleistung unter vollständiger Ausnutzung der Verzögerungszeit erlauben (Punkt A). Dem gegenüber wird das Ersetzen der Verlustleistung durch eine Maximum-Restriktion zu einer schnelleren Schaltung führen, die das Verlustleistungspotential ausschöpfen wird (Punkt B).

Gewichtsmethode

Mit einer Abbildung $f(\vec{x}, \vec{w}): \mathbb{R}^n \rightarrow \mathbb{R}$ lässt sich die n-dimensionale Zielgröße eines zu optimierenden Systems auf eine skalare Zielgröße zurückführen. Ist f monoton in x_i , $\forall i$ für den gegebenen Gewichtsvektor $\vec{w} \in \mathbb{R}^m$, so ist die Minimierung von f über die ursprünglichen Parameter ein Element der Pareto Menge bezüglich zu minimierender x_i . Der Gewichtsvektor \vec{w} dient dann der Beeinflussung des erhaltenen Elementes. Als Beispiel sei die häufig eingesetzte, gewichtete Summe erläutert. Die Abbildung lautet $f(\vec{x}, \vec{w}) = \sum_i w_i x_i$, wobei die w_i jeweils den Einfluss der Zielgröße x_i auf die resultierende Zielgröße f festlegen. \vec{w} kann als Richtungsvektor interpretiert werden, der in Richtung schlechterer Lösungen verweist. Orthogonal hierzu liegen Hyperebenen gleichwertiger Lösungen (Vergleiche Abbildung 4.7). Die

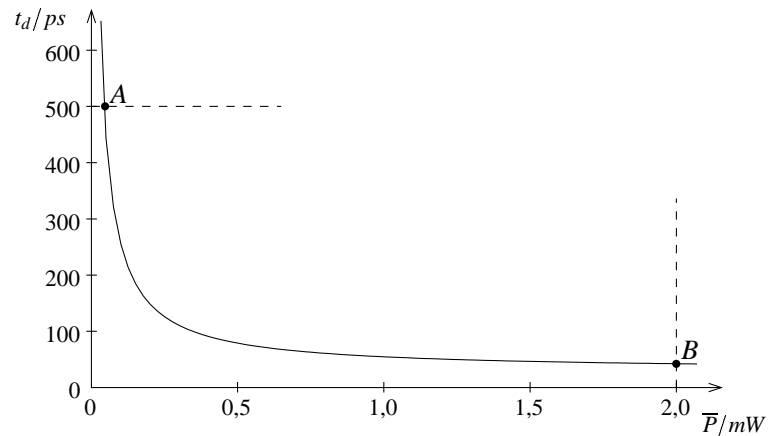


Abbildung 4.6: Verzögerungszeit-Verlustleistungs-Charakteristik eines Inverters. Die durchgezogene Linie zeigt die Pareto Menge, die gestrichelten Linien die angenommenen Spezifikationsgrenzen von $t_d = 500ps$ und $\bar{P} = 2mW$.

Schwierigkeit bei diesem Ansatz liegt in der Bestimmung geeigneter Gewichte, falls detaillierte Kenntnisse über die Struktur der Pareto Menge fehlen.

MinMax-Problem

Zur Vermeidung der Problematik einer geeigneten Gewichtswahl bei dem vorherigen Ansatz kann ein MinMax-Problem formuliert werden. Hierbei ist das Maximum der Zielgrößen zu minimieren. Das bewirkt, dass die Optimierung stets die momentan schlechteste Zielgröße zu verbessern sucht. Allerdings ist auch in diesem Ansatz eine implizite Skalierung enthalten, da in realen Problemstellungen normalerweise nicht alle Zielgrößen gleichartig sind. Um das Beispiel des Inverters beizubehalten: Der Zahlenwert der Verlustleistung liegt bei 10^{-4} , der der Verzögerungszeit bei 10^{-10} . Die Formulierung als MinMax-Problem kommt somit einem Ersatz der Zielgröße Verzögerungszeit durch eine gestellte Restriktion gleich. Für eine adäquate Anwendung des MinMax-Ansatzes ist eine Skalierung der Zielgrößen auf Zahlenwerte gleicher Größenordnung erforderlich, so dass ebenfalls Vorwissen über das vorliegende System vorhanden sein muss.

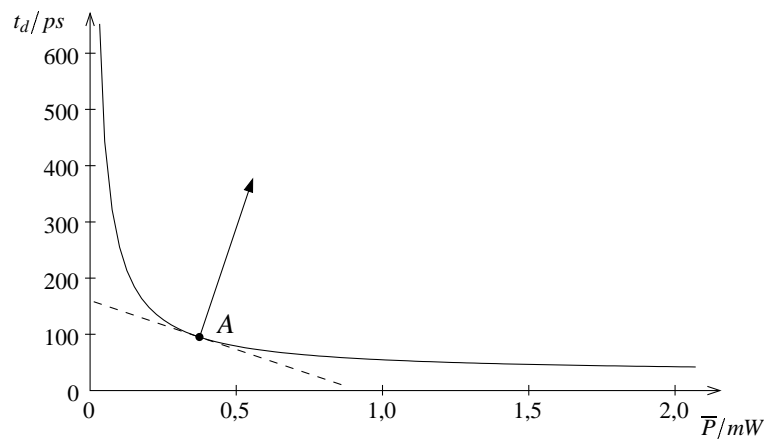


Abbildung 4.7: Verzögerungszeit-Verlustleistungs-Charakteristik eines Inverters. Die durchgezogene Linie zeigt die Pareto Menge, der Punkt A ist eine Lösung des Optimierungsproblems mit der gestrichelten Linie gleichwertiger Lösungspunkte bezüglich der eingezeichneten Gewichtung.

4.2.1 Iterative Gewichtsbestimmung

Die Problematik bei der Anwendung einer zusammenfassenden Abbildung, wie der gewichteten Summe, besteht in der Wahl geeigneter Gewichte. Dem Anwender ist in der Regel das Systemverhalten bezüglich der Zielgrößencharakteristik nicht bekannt. Um eine für den Anwender optimale Lösung zu erhalten, wäre somit eine Mehrgrößenoptimierung durchzuführen, die eine Beschreibung in Form von Einzellösungen, der Pareto Menge, liefert. Aus diesen Elementen ist dann die individuell optimale Lösung zu wählen. Das kann im zwei- und dreidimensionalen Fall visuell erfolgen, während ab vier Dimensionen Einzelvergleiche erforderlich werden.

Die Zahl der erforderlichen Elemente zur Beschreibung der Pareto Menge wächst polynomiell in der Zielgrößendimension und die Zahl der notwendigen Vergleiche nimmt exponentiell mit der Zahl der Elemente zu, so dass eine nicht vorstrukturierte Suche in zeitlicher Hinsicht nicht durchführbar ist. Eine Möglichkeit zu einer gerichteten Suche eines geeigneten Gewichtsvektors, der wiederum die gewünschte Lösung für das Optimierungsproblem liefert, besteht in einem iterativen Ansatz [TG00]. Hierbei wird der Informationsgewinn aus dem Vergleich zweier Realisierungen dazu verwendet, den Gewichtsraum einzuschränken.

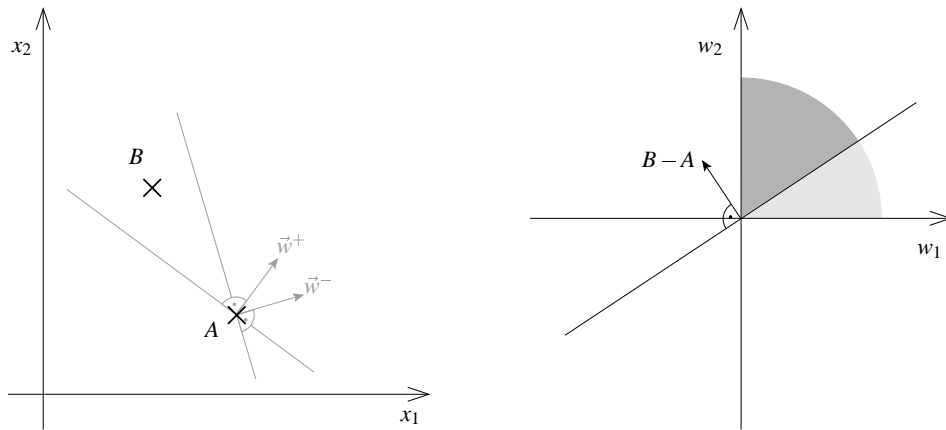


Abbildung 4.8: In der linken Abbildung sind zwei Elemente A und B des Zielgrößenraumes dargestellt. Für den Fall $A \prec_w B$ stellt \vec{w}^+ eine mögliche Gewichtung dar, während \vec{w}^- die Prämisse verletzt. Die zu den Gewichtsvektoren orthogonalen Geraden bezeichnen die Menge der zu A gleichwertigen Elemente bezüglich der jeweiligen Ordnung. In der rechten Abbildung ist der Gewichtsraum dargestellt. Der dunkel schattierte Bereich kennzeichnet die Einschränkung aufgrund der Zusatzinformation $A \prec_w B$.

Seien \vec{w}_1, \vec{w}_2 zwei beliebig gewählte Gewichtungen für die Zielgrößen (Abbildung 4.8). Da nur Minimierung betrachtet wird gilt: $w_i \geq 0$. Die aus einer Optimierung mit \vec{w}_1 bzw. \vec{w}_2 erhaltenen Realisierungen A bzw. B sind Elemente im Zielgrößenraum. Wird vom Anwender A als das bessere der beiden Elemente identifiziert ($A \prec_w B$), so lässt sich schließen, dass für die gesuchte Gewichtung \vec{w}^* gelten muss $\vec{w}^* A < \vec{w}^* B$. Unter Berücksichtigung dieser Bedingung $\vec{w}^* (B - A) > 0$ werden wiederum beliebige Gewichte gewählt, um mit dem ersten Schritt fortzufahren. Statt in einem Schritt nur zwei Gewichte zu verwenden, kann ebenso eine beliebige Menge von Gewichten gewählt werden, die es dem Anwender erlaubt, Paare von Elementen zu bestimmen, für die er eine Ordnung leicht angeben kann.

Der Ablauf des Verfahrens wird im Folgenden anhand eines Inverters erläutert. Im ersten Schritt seien drei Gewichtungen zufällig bestimmt, die die Lösungen in Abbildung 4.9(a) liefern. Von diesen sei die eingekreiste die beste, so dass deren Präferenz gegenüber den beiden übrigen Lösungen zur Eingrenzung des Gewichtsraumes führen, wie in Abbildung 4.9(d)

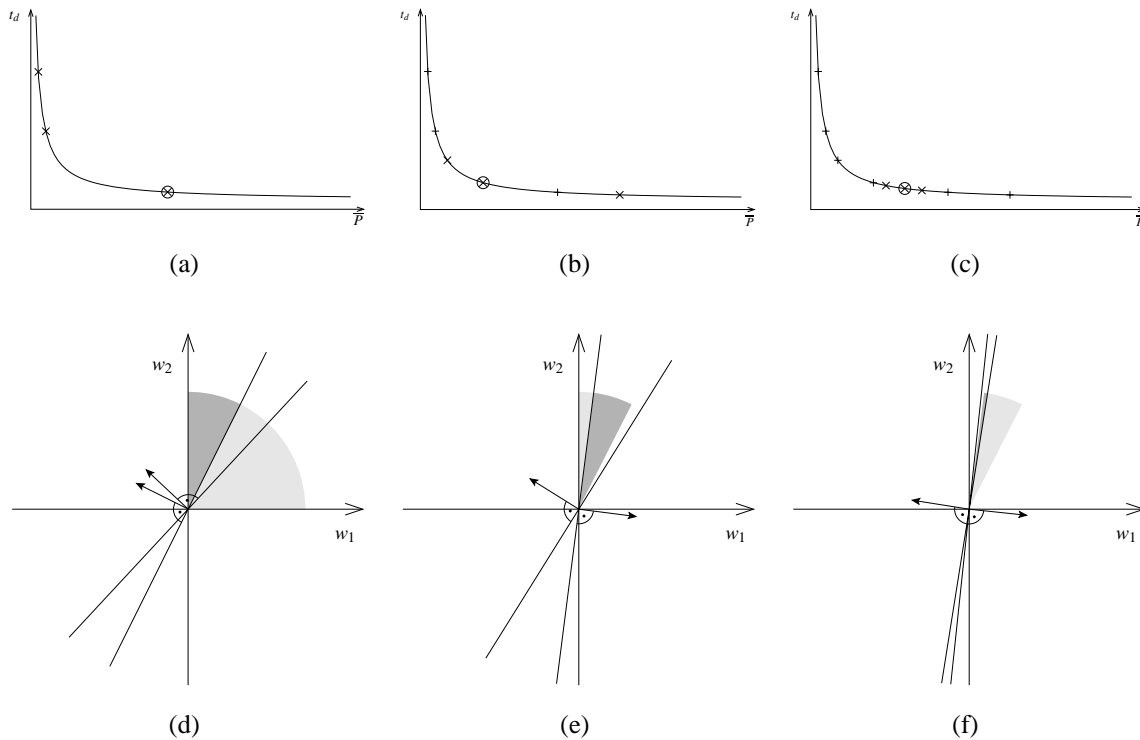


Abbildung 4.9: Drei Schritte der iterativen Gewichtsbestimmung. In der oberen Reihe ist die Darstellung im Zielgrößenraum mit den aktuellen (x) und den vorherigen (+) Resultaten. In der unteren Reihe ist der Gewichtsraum gegeben, in dem der zulässige Bereich grau schattiert ist.

gezeigt. Aus diesem reduzierten Gewichtsraum werden im zweiten Schritt wiederum drei Gewichtungen zufällig ermittelt. Die eingekreiste Lösung in Abbildung 4.9(b) sei wiederum die vom Anwender als beste ermittelte. Dies liefert den Gewichtsraum gemäß Abbildung 4.9(e).

Ebenso wird für den dritten Schritt (Abbildung 4.9(c), 4.9(f)) verfahren, bei dem die Lösungen schon dicht beieinander liegen. Auf diese Weise lässt sich iterativ der Gewichtsraum eingrenzen und eine anwenderbezogene Lösung für ein gegebenes System ermitteln. Der Vorteil dieses Ansatzes besteht in dem geringen Vergleichsaufwand für den Anwender. Die Leistungsfähigkeit der iterativen Gewichtsbestimmung zeigt sich in der folgenden Anwendung, der Modellierung eines Nano-Bauelementes.

4.2.2 Bauteilmodellierung mit iterativer Gewichtsbestimmung

Neben dem Schaltungsdesign ist für die Erarbeitung innovativer Schaltungstechniken die Berücksichtigung neuartiger Bauelemente erforderlich. Die in diesem Zusammenhang bedeutsame Nano-Technologie könnte die Möglichkeit bieten, den fortschreitenden Miniaturisierungsprozess auch in den nächsten Jahrzehnten aufrecht zu erhalten [GPKR97]. Die Modellierung dieser Nano-Bauelemente ist zum Teil erschwert durch eine höhere Komplexität der I/O-Charakteristik im Vergleich zu klassischen Halbleiterbauelementen und wird hier am Beispiel der Resonanztunneldiode betrachtet [TPG99].

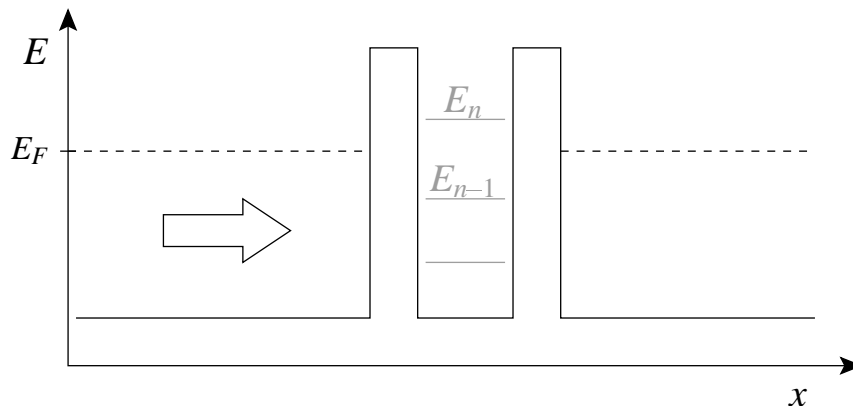


Abbildung 4.10: Ortsabhängiger Energieverlauf einer Resonanztunneldiode mit diskreten Energieniveaus zwischen den beiden Barrieren.

Die Resonanztunneldiode (RTD) besteht aus einer Halbleiterheterostruktur, die zwei eng beieinander liegende Tunnelbarrieren realisiert (Abbildung 4.10). Durch den geringen, im Nanometerbereich liegenden Abstand der beiden Barrieren, bilden sich zwischen diesen diskrete Energieniveaus aus. Durch Anlegen einer äußeren Spannung lassen sich diese Niveaus in Bezug zum äußeren Energieniveau ändern und somit die Tunnelwahrscheinlichkeit variieren. Für große Spannungen dominiert der Strom durch die Elektronen, die klassisch über die Barrieren hinweg driften, so dass die Charakteristik des Bauteils der Kennlinie einer Diode gleicht. Das hier betrachtete Modell [BBvdW⁺98, PGBP98] zur Beschreibung des Verhaltens der Resonanztunneldiode ist in Tabelle 4.2 gegeben. Zur Anpassung des Modells an eine gemessene RTD, sind die acht freien Parameter zu bestimmen (V_W : Resonanzbreite, V_P : Peakspannung, V_N : Spannung am maximalen negativen differentiellen Widerstand (NDR), V_V : Valleyspannung, I_P : Peakstrom, I_V : Valleystrom, n_R : Resonanzstromparameter, n_V : thermoionischer

$$I_{\text{RTD}}(V) = I_{\text{res}}(V) + I_{\text{res}}(-V) + I_{\text{th}}(V) \quad (4.3)$$

$$I_{\text{res}}(V) = \frac{I_P}{f} \left(1 + \frac{2}{\pi} \arctan \left(\frac{V_N - V}{V_W} \cdot \frac{n_R k T}{q (V_N - V_P)} \right) \cdot \ln \left[1 + \exp \left(\frac{q (V - V_P)}{n_R k T} \right) \right] \right) \quad (4.4)$$

$$I_{\text{th}}(V) = I_V \cdot \frac{\sinh \left(\frac{q V}{n_V k T} \right)}{\sinh \left(\frac{q V_V}{n_V k T} \right)} \quad (4.5)$$

$$f = 1 - \sqrt{2 V_W / \pi (V_N - V_P)} \quad (4.6)$$

Tabelle 4.2: Modell der Resonanztunnelodiode als Strom-Spannungs-Zusammenhang $I_{\text{RTD}}(V)$.

Stromparameter). Für eine automatisierte Bestimmung dieser Parameter unter Berücksichtigung des physikalischen Hintergrundes werden diese nicht beliebig in \mathbb{R} oder \mathbb{R}^+ variiert. Dies würde zu einem rein mathematischem Fitting der Funktion führen, das zumeist einen nicht interpretierbaren und physikalisch absurden Parametersatz liefert. Daher wird hier die Variabilität der Parameter durch absolute bzw. relative Intervalle, deren Grenzen von einem Experten definiert worden sind, eingeschränkt (vergleiche Tabelle 4.3).

Die Schwierigkeiten bei der Modellierung eines realen und gemessenen Bauteils ergeben sich aus Einschränkungen der Messpraxis. Die RTD findet ihre Anwendung u.a. in schnellen Digitalschaltungen ($f_{\text{clk}} \gg 1 \text{ GHz}$). In diesem Frequenzbereich sind die Schaltungssignale durch die verfügbare Messeinrichtungen typischerweise nicht mehr vollständig zu erfassen. Wichtige Einflussgrößen sind die Kapazitäten und Induktivitäten der Messspitzen zur Abnahme der Signale vom Chip und die Leitungslängen. Dies ist in Abbildung 4.11 nachzuvollziehen, in der eine Messkurve der RTD dem obigen Modell gegenübergestellt ist. Der Verlauf in dem abfallenden Abschnitt der Kurve (Abbildung 4.11, Bereich II) ist auf die Messeinrichtung zurückzuführen und spiegelt nicht das Verhalten der RTD innerhalb einer Schaltung wider. Dieser Umstand bewirkt, dass ein einfaches Abstandsmaß zwischen Messkurve und Modell nicht zur Bewertung des Modells geeignet ist. Daher werden verschiedene Teilaspekte der

$10 \text{ mV} \leq V_W \leq 120 \text{ mV}$
$300 \text{ mV} \leq V_P \leq 600 \text{ mV}$
$100 \text{ mV} \leq V_N \leq 250 \text{ mV}$
$400 \text{ mV} \leq V_V$
$1 \text{ mA} \leq I_P \leq 6 \text{ mA}$
$I_P/20 \leq I_V \leq I_P/5$
$1 \leq n_R$
$1 \leq n_V$

Tabelle 4.3: Eingrenzung der Parameter des Modells der Resonanztunnelodiode.

Kurve als Indikatoren verwendet, um die Approximationsgüte des betrachteten Modells beurteilen zu können.

Die fünf eingesetzten Indikatoren bestimmen sich aus partiellen Abstandsmaßen bzw. aus charakteristischen Größen der Kurven. Im Einzelnen sind dies die Folgenden: Für die ansteigenden Bereiche der Messkurve wird der mittlere quadratische Fehler berücksichtigt. Dies erfolgt separat für den ersten Anstieg (Abbildung 4.11, Bereich I), den linken Teil des zweiten Anstiegs (Bereich III) und den Rest der Kurve (Bereich IV). Für den Bereich III wird zudem der mittlere quadratische Fehler zwischen den jeweiligen Gradienten ermittelt. Der fünfte Punkt betrifft das lokale Maximum des Strom-Spannung-Zusammenhangs (Abbildung 4.11, Kreis). Es wird hierbei der Euklidische Abstand zwischen den Extrema von Mess- und Modellkurve verwendet.

Die Zusammenfassung der fünf einzelnen Kriterien erfolgt durch eine lineare Gewichtung dieser. Für die erforderliche Wahl der Gewichtung stellt sich die Problematik des vorherigen Kapitels, da die gegenseitige Abwägung der verschiedenen Ziele nicht intuitiv durchführbar ist. Daher wird an dieser Stelle die iterative Gewichtsbestimmung angewendet. Anhand des Spannungs-Strom-Verlaufs und der physikalischen Parameter ist ein Experte in der Lage, eine subjektive Bewertung und Ordnung verschiedener Lösungen vorzunehmen. Allerdings zeigt sich, dass es schwer fällt, eine Ordnung für zwei sehr schlechte Lösungen anzugeben, die in unterschiedlichen Aspekten versagen. Aus diesem Grund werden hier fünf Lösungen zugleich bestimmt (Abbildung 4.12(a)), aus denen der Experte Paare herausgreift und de-

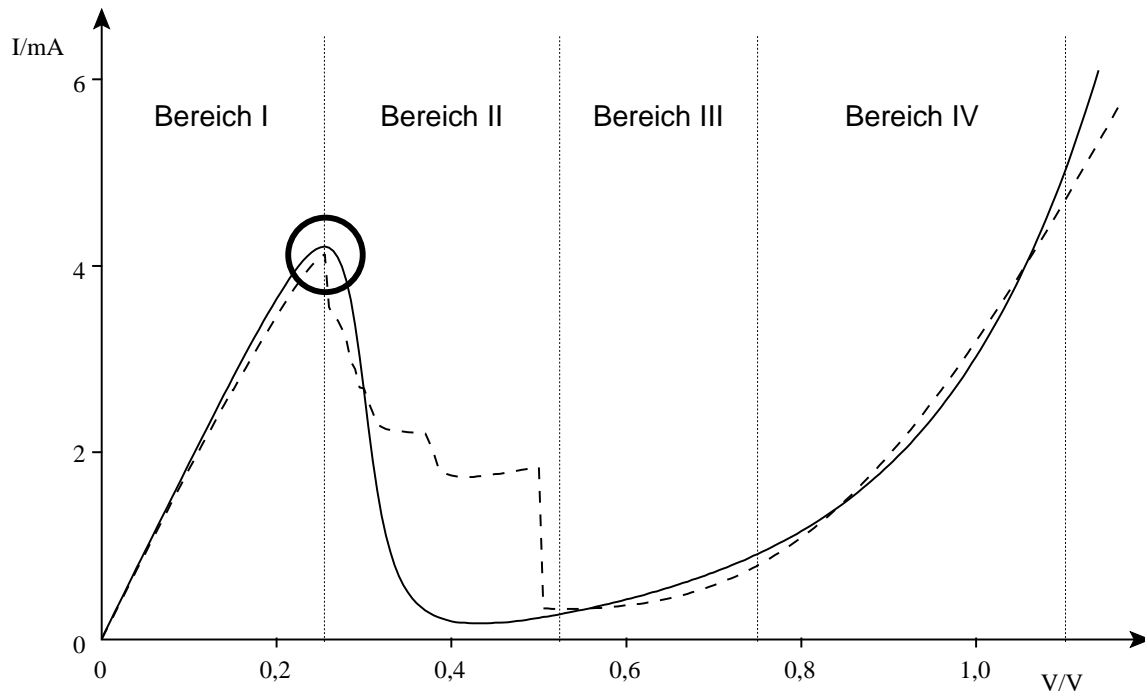
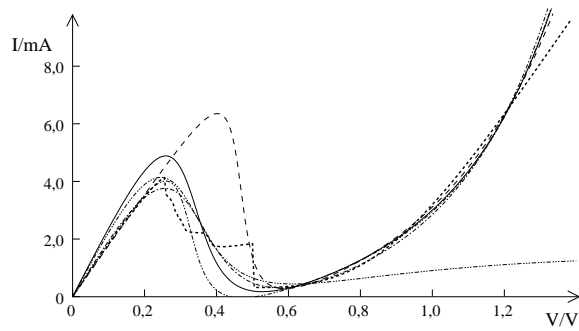


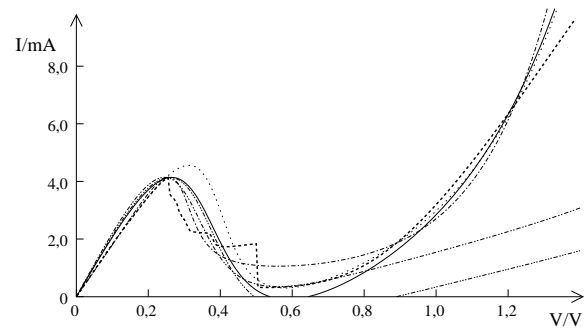
Abbildung 4.11: Spannung-Strom-Kennlinie mit der Messkurve (gestrichelte Linie) einer RTD (Typ DU595 [Pac01]) und zugehöriger Modellierung (durchgezogene Linie) mit Hilfe evolutionärer Algorithmen. Die Bereiche I, III und IV beschreiben den Wirkungsabschnitt der verschiedenen Bewertungskriterien.

ren Ordnung festlegt. Im nächsten Schritt werden unter Berücksichtigung der gewonnenen Information weitere fünf Lösungen (Abbildung 4.12(b)) generiert und so fort. In diesem Anwendungsbeispiel ist bereits im vierten Durchlauf die verbliebene Variationsbandbreite sehr gering (Abbildung 4.12(d)) und im Anschluss daran die Gewichtung der Kriterien nahezu eindeutig.

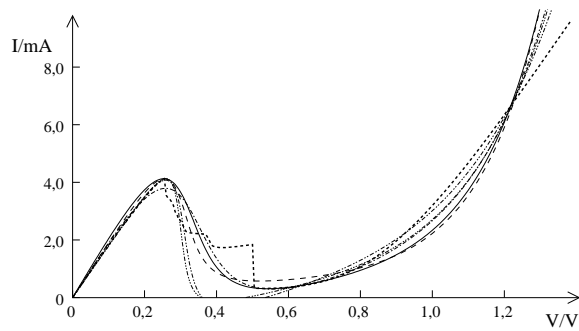
Das ermittelte Modell ist geeignet zur Simulation integrierter Schaltungen mit Resonanztunneldioden. Die hierbei bestimmte Gewichtung erlaubt zudem, die Parameter des Modells für weitere Messkurven von Resonanztunneldioden des selben Typs vollautomatisch anzupassen und so beispielsweise den Technologieprozess zu überwachen.



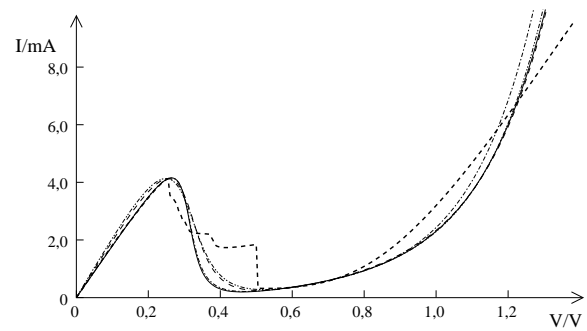
(a) Lösungen 1-5



(b) Lösungen 6-10



(c) Lösungen 11-15



(d) Lösungen 16-20

Abbildung 4.12: Ergebnisse der RTD Modellierung im Laufe der iterativen Gewichtsbestimmung. Es ist in vier Schritten (a)-(d), in denen jeweils fünf Lösungen bestimmt werden, der Fortschritt des Verfahrens zu verfolgen.

4.3 Untersuchung verschiedener Schaltungskonzepte

Die rechnergestützte Optimierung integrierter Schaltungen in der industriellen Entwicklung nahe am fertigen Produkt dient der Ausnutzung vorhandener Technologiereserven sowie der Kostenreduktion durch geringeren Flächenbedarf. Jedoch ist dies nicht die Domäne globaler Optimierungsverfahren, wie der evolutionären Algorithmen. In diesem Bereich existiert über die verwendete Technologie und Schaltungstechnik ein hohes Maß an Systemwissen. Dies ergibt sich aus der Erfahrung mehrfacher Verwendung der existierenden Technik und einer teilweisen Beschränkung z.B. auf eine feste Spannung. Das Systemwissen erlaubt die systematische Erstellung eines guten, im Sinne von nahe am Optimum liegenden Schaltungsdesigns. Für eine weitere Verbesserung genügt somit eine lokal orientierte Suche.

Sinn und Zweck von Optimierungsverfahren in der Forschung ist die Evaluation des Potentials neuer Schaltungstechniken oder Technologien. Hierzu ist für die Bestimmung des Optimums eine globale Suche, frei von der Forderung nach Vorwissen notwendig. Evolutionäre Algorithmen können aus dieser Perspektive zu einer Bewertung und vergleichenden Gegenüberstellung unterschiedlicher Schaltungskonzepte dienen.

Bei der Untersuchung neuartiger Schaltungskonzepte oder Technologien ist es wichtig, objektivierte Kenngrößen zu deren Charakterisierung zu erhalten. Diese können neben der Verlustleistung, dem Flächenbedarf und dem Power-Delay-Produkt auch die Robustheit sein. Als problematisch bei der Durchführung einer Bewertung, die stets in Relation zu existierenden und etablierten Alternativen zu setzen ist, erweist sich der menschliche Experte. Da der Vorgang des Schaltungsdesigns stark von Erfahrungen geprägt ist, ist davon auszugehen, dass das Potenzial der Technik-Technologie-Kombination, für die ein breiteres Vorwissen existiert, deutlich besser ausgeschöpft wird. Das kann dazu führen, dass neuartige Schaltungstechniken zu früh verworfen und nicht weiter verfolgt werden. Die Einbeziehung einer global orientierten Suchmethode erlaubt, die Kenngrößen durch Ausnutzen des existierenden Optimierungspotentials zu objektivieren. Ein geeignetes Instrument hierzu ist ein evolutionärer Algorithmus mit einer genügenden Diversität (vergleiche Kapitel 3.3).

Die Einsetzbarkeit dieses Ansatzes zur Untersuchung integrierter Schaltungen wird im nachfolgenden Beispiel demonstriert. Für einen 1-Bit Volladdierer werden zwei konkurrierende Ansätze auf logischer Ebene einander gegenübergestellt. Diese Ansätze spiegeln sich in verschiedenen Schaltungstechniken wider, die basierend auf einer existierenden Technologie bewertet werden.

4.3.1 Boolesche und Schwellwertlogik

Die Beschreibung logischer Netzwerke mit einem theoretischen Fundament wurde aus der *Switching Theory* entwickelt. Der verbreitete Ansatz der Booleschen Logik [Sha38, in [Mur71]] basiert auf den grundlegenden Konzepten von *UND*, *ODER* und *NICHT*. Die logischen Werte 0 und 1 entsprechen den Zuständen *falsch* und *wahr* und die logischen Operationen decken sich auf semantischer Ebene mit den sprachlichen Konzepten. Es wird hier die übliche Notation für die Verknüpfungen verwendet ($ab = a \text{ UND } b$; $a + b = a \text{ ODER } b$; $\bar{a} = \text{NICHT } a$; Tabelle 4.4);

a	b	ab	$a + b$	\bar{a}
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Tabelle 4.4: Wahrheitstabelle für die grundlegenden logischen Operatoren UND, ODER und NICHT

Neben der Booleschen Logik existiert der Ansatz der Schwellwertlogik, bei der die Eingangsgrößen durch eine Linearkombination zusammengefasst und gegen einen Schwellwert getestet werden. Die binären Eingangsgrößen x_i werden abhängig von Gewichten $w_i \in \mathbb{Z}$ und einem Schwellwert Θ über eine Funktion $f_{\vec{w},\Theta}(\vec{x})$ gemäß Gleichung (4.7) abgebildet. Eine alternative Schreibweise ist in Gleichung (4.8) gegeben, wobei $\text{sgn}(\cdot)$ die Sprungfunktion bezeichnet.

$$f_{\vec{w},\Theta}(\vec{x}) = \begin{cases} 1, & \text{wenn } \vec{w}\vec{x} \geq \Theta \\ 0, & \text{sonst} \end{cases} \quad (4.7)$$

$$f_{\vec{w},\Theta}(\vec{x}) = \text{sgn} \left(\sum_i w_i x_i - \Theta \right) \quad (4.8)$$

Die Schwellwertlogik lässt sich auf reellwertige Größen verallgemeinern mit $x_i \in [0;1]$ und $w_i, \Theta \in \mathbb{R}$. Aus der Sprungfunktion wird in diesem Fall eine sigmoide Abbildung $f: \mathbb{R} \rightarrow [0;1]$. Ein solch verallgemeinertes System zeigt Parallelen zu einem neuronalen Netz wie das Perzeptron [HKP91]. Im hier betrachteten Bereich der Digitalschaltungen ist jedoch nur der diskrete Fall zu beachten.

Die Boolesche Logik kann als Spezialfall der Schwellwertlogik dargestellt werden, d.h. jeder Boolesche Ausdruck lässt sich mit Hilfe der Schwellwertlogik beschreiben. Dies ist von Interesse, da erstere die Basis für die aktuellen Schaltungstechniken bildet und sich ein neuer Ansatz stets dem Vergleich mit existierenden Lösungen stellen muss.

Die Realisierung der obigen Beispiele und Grundkonzepte der Booleschen Logik lassen sich in der Schwellwertlogik wie folgt formulieren. *UND*: $\vec{x} = (x_1, x_2)^T$, $\vec{w} = (1, 1)^T$ und $\Theta = 2$. *ODER*: $\vec{x} = (x_1, x_2)^T$, $\vec{w} = (1, 1)^T$ und $\Theta = 1$. *NICHT*: $\vec{x} = (x_1)$, $w_1 = -1$ und $\Theta = 0$.

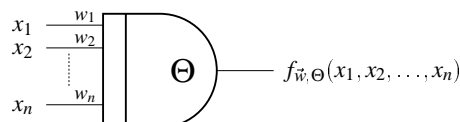


Abbildung 4.13: Symbol des Schwellwertgatters.

Ein weiterer Spezialfall der Schwellwertlogik oder Schwellwertgatter (Abbildung 4.13) ist die Majoritätslogik oder Majoritätsgatter (Majority-Gate). Ein Majoritätsgatter ist ein Schwellwertelement mit einer ungeraden Anzahl n an Eingängen, einer identischen Gewichtung $w_i = +1$ und einem Schwellwert gemäß $\Theta = \frac{n+1}{2}$. Für ein solches Gatter liegen eine Reihe schaltungstechnischer Realisierungen vor. Das Majoritätsgatter zusammen mit einer Negation ist logisch äquivalent zum Schwellwertgatter [Mur71].

4.3.2 Das Majoritätsgatter

Das hier betrachtete Majoritätsgatter nach Lee [LJ92], in vereinfachter Form auch von Johnson [Joh88] vorgestellt, ist ein statisches Gatter (vergleiche Abbildung 4.14), bei dem je Eingangssignal ein CMOS Inverter verwendet wird, die ausgangsseitig in einem Knoten zusammengeschaltet sind. An diesem Knoten stellt sich ein Spannungspotenzial ein, das die gewichtete Summe widerspiegelt. Der Schwellwert des Gatters wird mit einem Ausgangsinverter eingestellt. Das Lee-Gatter ist insofern kein vollständiges Majoritätsgatter, als dass keine negativen Gewichtungen realisiert werden können. Als Ausweg bietet sich die Möglichkeit an, dem Eingang einen klassischen CMOS Inverter vorzuschalten [BTG00]. Da abgesehen von Ausnahmefällen sowohl ein p-Kanal- als auch ein n-Kanal-Zweig durchschaltet, wird durch das Lee-Gatter statische Verlustleistung verursacht, die insbesondere bei geringer Schaltaktivität (switching activity) nachteilig ist.

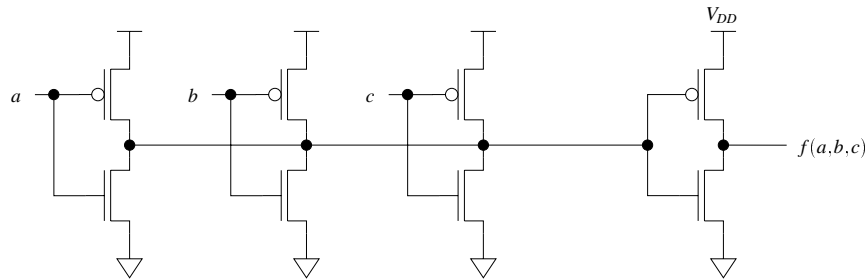


Abbildung 4.14: Majoritätsgatter nach Lee mit drei Eingängen a , b und c .

Das Lee-Gatter ist nach außen ein digitales Schaltelement. Die interne Struktur mit dem Potenzialknoten weist es jedoch als analoges Bauelement aus, so dass das Verhältnis der p-Kanal- zu n-Kanal-Transistoren nicht nur über die Symmetrie der Schaltvorgänge entscheidet, sondern auch über die Erfüllung der Schaltfunktion an sich (Abbildung 4.15). Mit dem Lee-Gatter liegt ein Majoritätsgatter vor, das sowohl in den Signalpegeln als auch in der verwendeten Technologie kompatibel zur statischen CMOS Realisierung der Booleschen Logik ist.

Für eine allgemeine Verwendung des Gatters sind die Eingangssignale bezüglich Änderungswahrscheinlichkeiten und Laufzeitanforderungen als gleichwertig anzusehen. Daher sind die eingangsseitigen p-Kanal- bzw. n-Kanal-Transistoren gleich zu dimensionieren. Die Ausgangsstufe kann hiervon unabhängig zur Realisierung des Schwellwertes bestimmt werden. Um einen möglichst großen Signal-Störspannungsabstand zu erreichen, wird der Schwellwert nicht wie im theoretischen Ansatz auf $\frac{n+1}{2}$, sondern auf $\frac{n}{2}$ eingestellt.

Wie erwähnt ist das Ausgangswiderstandsverhältnis vom p-Kanal- zum n-Kanal-Zweig für die Funktion entscheidend. Der Ausgangswiderstand wiederum ist über die Transistorgeometrie durch das W/L -Verhältnis beeinflussbar. Die notwendige Beschränkung auf einen diskreten Suchraum, bedingt durch die gegebenen Designregeln (siehe Abschnitt 4.3.5), macht somit die Einbeziehung der Transistorlängen in die Optimierung erforderlich. Dass heißt es ist nicht möglich, die Transistorlängen z.B. auf Minimaldesign a priori festzulegen und somit den Parameterraum zu verkleinern, um das gesamte Optimierungspotenzial des Lee-Gatters auszuschöpfen. Für das Design eines Majoritätsgatters nach Lee sind folglich acht Geometrieparameter zu berücksichtigen, namentlich die Weiten und Längen für die p-Kanal- und n-Kanal-Transistoren auf der Eingangsseite und gleiches für die Ausgangsseite.

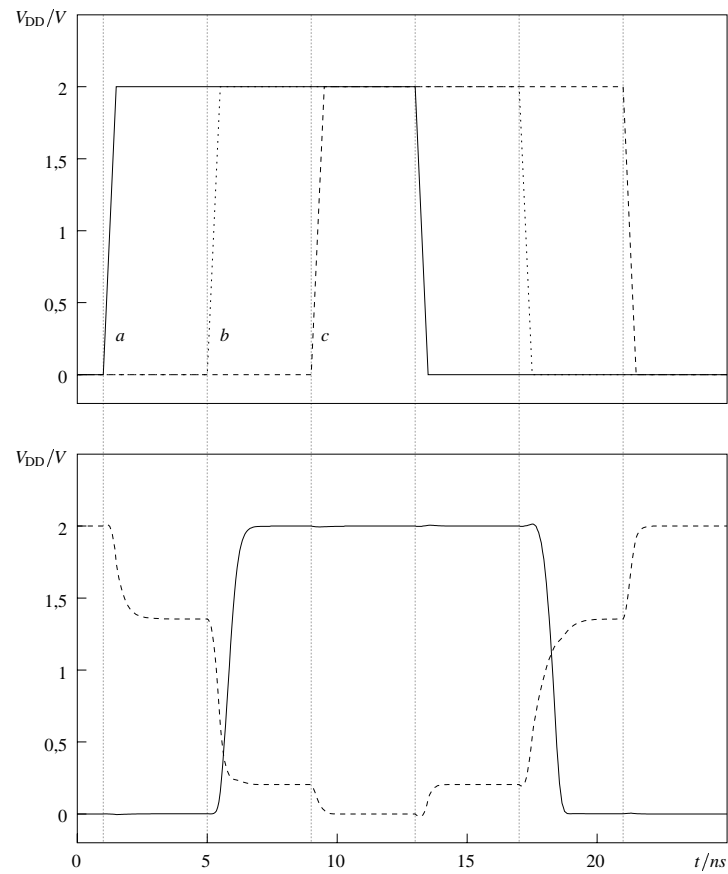


Abbildung 4.15: Ein- Ausgangssignal des Lee-Gatters: Oben sind die vorgegebenen Eingangssignale a , b und c dargestellt. Darunter finden sich die Spannungen am Ausgang (durchgezogene Linie) und am Potenzialknoten (gestrichelte Linie).

4.3.3 Spiegelladdierer

Für die Realisierung des Volladdierers in Boolescher Logik wird hier statische CMOS Technik eingesetzt. Die als Spiegelladdierer [WE85, Rab96, KL99] bezeichnete Schaltung verwirklicht die Funktion des Addierers mit einem Minimum von 24 Transistoren [Rab96] (ohne Ausgangsinverter). Deren symmetrische Anordnung erlaubt zudem ein Design mit identischen steigenden und fallenden Flanken. Varianten mit einer geringeren Zahl an Transistoren basieren auf differierenden Schaltungstechniken, wie der der Transfer-Elemente [WE85]. Der Spiegelladdierer leitet sich von einer Abwandlung der Booleschen Gleichungen für Summe und Carry (4.1, 4.2) ab. Mit den Gleichungen (4.9) und (4.10) ergibt sich die Schaltung in Abbildung 4.16. In der linken Hälfte der Abbildung findet sich die Bildung des negierten Carry \bar{c}' , in der rechten Hälfte die der negierten Summe \bar{s} . Die Ausgangsinverter sind für den alleinstehenden Volladdierer erforderlich und werden hier mit eingerechnet. Bei Einbindung in einen n-Bit Addierer oder geeigneten Folgeschaltungen kann eventuell auf diese Inverter verzichtet werden.

$$c' = ab + ac + bc \quad (4.9)$$

$$s = abc + \bar{c}'(a + b + c) \quad (4.10)$$

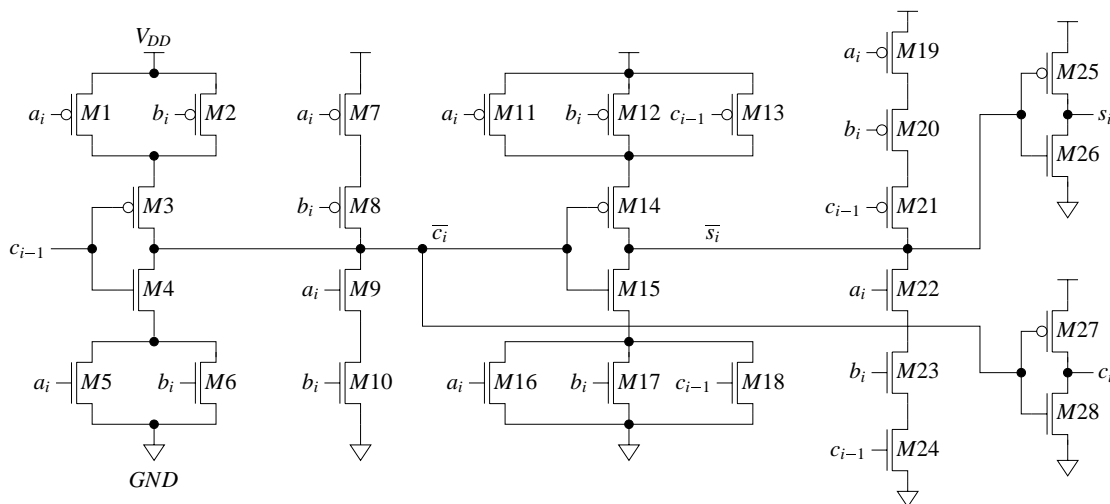


Abbildung 4.16: Spiegelladdierer.

Für die 28 Transistoren des Spiegeladdierers sind unter Berücksichtigung der Ausgangsinverter Weiten und Längen zu bestimmen. Ist die Versorgungsspannung, die zudem die Logikpegel definiert, ebenfalls variierbar, ergeben sich 57 wählbare Parameter. Im statischen CMOS Design wird die Schaltgeschwindigkeit nur über das W/L -Verhältnis und die Spannung beeinflusst. Die absoluten Transistorgeometrien sind im Bereich verlustleistungsarmer Logikschaltungen, wie in dieser Anwendung vorliegend, nicht entscheidend. Somit können die Transistorlängen dem minimalen Designmaß entsprechend gewählt werden, so dass sich die Zahl der Parameter auf 29 reduziert.

Bei den allgemeinen Betrachtungen zu dem Beispiel des Volladdierers in Kapitel 4.1.1 wurde bereits die Gleichwertigkeit der Eingangssignale a und b begründet. Dies zeigt sich in der Struktur des Spiegeladdierers bestätigt. Diese beiden Signale können beim Spiegeladdierer ohne Beeinflussung der Schaltfunktion gegeneinander ausgetauscht werden. Dies gilt neben der logischen Funktion auch für das Schaltverhalten im Zeitbereich.

Die Begründung für die Kopplung korrespondierender Transistorparameter deckt sich mit der in Kapitel 4.1 (Seite 31) beschriebenen. Die Zustandshäufigkeit im Testmuster, die nicht mit der (angenommenen) Zustandswahrscheinlichkeit des realen Systems übereinstimmen muss, wird die Auswertung bezüglich der Verlustleistung prägen. Folglich würde eine vollständig freie Suche eine auf das Testmuster optimierte Schaltung liefern. Durch eine geeignete Verknüpfung der freien Parameter lässt sich dieser Effekt unterbinden.

Die Gruppierung der Transistorparameter erschließt sich aus der Gruppierung der Transistoren zur Erfüllung der logischen Teilfunktionen. Hieraus ergeben sich die folgenden Gruppen zur Bestimmung der Transistorweite: (M1-M2), (M3), (M4), (M5-M6), (M7-M8), (M9-M10), (M11-M12-M13), (M14), (M15), (M16-M17-M18), (M19-M20-M21), (M22-M23-M24), (M25-M27), (M26-M28). Unter Hinzunahme der Versorgungsspannung ergeben sich somit 15 Designparameter. Die Berücksichtigung der gegebenen Designregeln für die verwendete Technologie wird in Abschnitt 4.3.5 diskutiert.

4.3.4 Addierer mit Majoritätsgattern

Der Realisierung eines Volladdierers in Boolescher Logik wird hier ein Addierer mit Majoritätsgattern gegenübergestellt. Die kompakteste Variante erfordert drei Gatter für die Generierung des Summen- und des Carry-Bits. Ausgangspunkt ist eine Realisierung mit Schwellwertgattern, die mit den Gleichungen (4.11) und (4.12) beschrieben ist. Abbildung 4.17 veranschaulicht diese in grafischer Form.

$$c' = \text{sgn}(a + b + c - 2) \quad (4.11)$$

$$s = \text{sgn}(a + b + c - 2c' - 1) \quad (4.12)$$

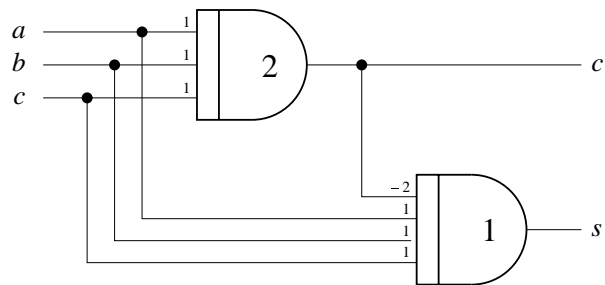


Abbildung 4.17: Volladdierer mit Schwellwertgattern.

Das Schwellwertgatter zur Bestimmung des Carries ist ein Majoritätsgatter und direkt durch ein Lee-Gatter umsetzbar. Für die Ermittlung der Summe wird das negativ gewichtete Carry-Signal durch die Verwendung seiner Negation \bar{c}' ersetzt. Dadurch kann anstelle des zweiten Schwellwertgatters ein Lee-Gatter eingesetzt werden. Das Ergebnis der Umformung zeigen die Gleichungen (4.13), (4.14) und die Abbildung 4.18. Für das erste Lee-Gatter werden 8 Transistoren benötigt. Der Inverter ist aus 2 Transistoren aufgebaut und das zweite Lee-Gatter aus 10 Transistoren. Auf den ersten Blick würden sich für das zweite Lee-Gatter 12 Transistoren ergeben, jedoch lassen sich durch den verdoppelten Eingang \bar{c}' die entsprechenden Transistoren zusammenfassen und durch einen p-Kanal- und einen n-Kanal-Transistor doppelter Weite ersetzen. Somit lässt sich die Addition mit 20 Transistoren durchführen.

$$c' = \text{sgn}(a + b + c - 2) \quad (4.13)$$

$$s = \text{sgn}(a + b + c + \bar{c}' + \bar{c}' - 3) \quad (4.14)$$

Für die beiden Lee-Gatter sind jeweils acht Geometrieparameter zu beachten. Hinzu kommen die Versorgungsspannung und die Parameter für den Inverter. Letzterer kann in klassischer

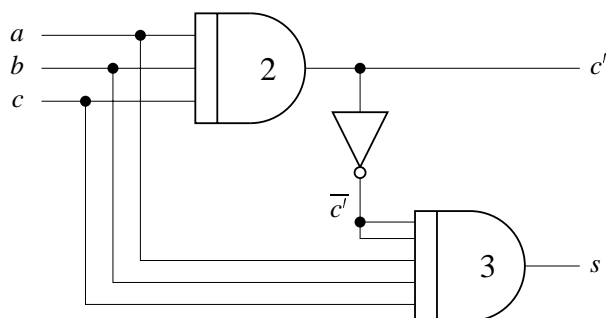


Abbildung 4.18: Volladdierer mit Lee-Gattern.

CMOS Technik mit minimalen Längen entworfen werden, so dass nur die Transistorweiten zu wählen sind. Zusammengefasst ergeben sich 19 Designparameter für den Volladdierer mit Majoritätsgattern.

4.3.5 Ergebnisse

Für die Analyse der oben genannten Schaltungen dienen die Daten einer industriellen Technologielinie als Basis für einen Vergleich. Neben einer $0,35 \mu\text{m}$ CMOS Technologie werden ebenfalls $0,6 \mu\text{m}$ - und $0,8 \mu\text{m}$ -Prozesse berücksichtigt. Für jeden dieser Prozesse liegen die erforderlichen Transistormodelle für deren Simulation mit SPICE vor.

Bereits in Kapitel 2.1.1 wurde auf die Prinzipien der Einschränkungen des Parameterraumes hingewiesen. An dieser Stelle sind nun die Randbedingungen für die vorliegenden Technologieprozesse zu bestimmen. Die zulässigen Transistorgeometrien, die hier gefragt sind, lassen sich aus den Design Rules des Prozesses ableiten. Ein minimaler Transistor in der verwendeten $0,35 \mu\text{m}$ -Technologie ist in Abbildung 4.19 gegeben. Das Design ist bestimmt durch folgende Größen: Zunächst sind die minimale Breite des Polysiliziums für das Gate (POLY1) und der Größe des Kontaktloches (CONT) zu beachten. Hinzu kommen ein Mindestabstand des Kontaktloches zum Gate und die minimale Überlappung des Diffusionsgebietes (DIFF) und der Metallisierungsebene (MET1). Zusammengefasst ergibt sich eine minimale Transistorweite von $1,0 \mu\text{m}$ und eine minimale Transistorlänge von $0,3 \mu\text{m}$.

Gegenüber der hier bestimmten Weite von $1,0 \mu\text{m}$ erlauben die Design Rules eine Transistorweite bis hinunter zu $0,6 \mu\text{m}$. Um dieses Maß zu realisieren, ist jedoch ein nicht konvexes Diffusionsgebiet erforderlich, wie in Abbildung 4.20 dargestellt. Dies führt erstens zu

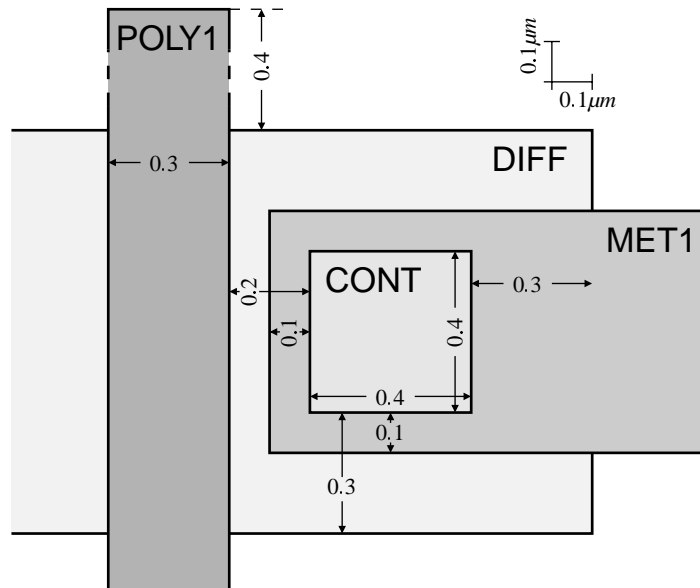
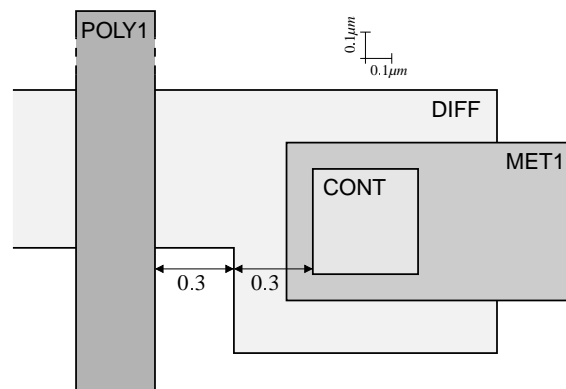


Abbildung 4.19: Transistorlayout nach $0,35 \mu\text{m}$ CMOS Design Rules (Angaben in μm).

einem nicht zu vernachlässigenden Mehr an Flächenbedarf. Bei der vorliegenden $0,35 \mu\text{m}$ -Technologie liegt der minimale Bedarf an Chipfläche bei Transistorweiten $< 1,0 \mu\text{m}$ um 48 % über dem bei einer Weite von $1,0 \mu\text{m}$. Der zweite und entscheidende Punkt ist der Einfluss parasitärer Effekte. Da die Kontaktfläche von der Metalllage zum Diffusionsgebiet ein Mindestmaß behält, skaliert die Diffusionskapazität, die linear von dieser Fläche abhängt, für Transistorweiten unter $1,0 \mu\text{m}$ nicht mit. Die fehlende Skalierung führt zu einer überproportionalen Fehler in der numerischen Simulation, da die Diffusionskapazität im SPICE Modell nicht explizit berücksichtigt wird.

Zusammen mit den minimalen Transistorgeometrien sind die Diskretisierungen der Designmaße zu beachten. Diese sind sowohl lateral als auch vertikal mit $0,05 \mu\text{m}$ vorgegeben. Entsprechend der $0,35 \mu\text{m}$ -Technologie lassen sich die Werte für die $0,6 \mu\text{m}$ - und $0,8 \mu\text{m}$ -Prozesse ableiten. Die Maße sind in Tabelle 4.5 zusammengefasst.

Im Folgenden werden Kenngrößen für das Lee-Gatter, den Spiegeladdierer und den Volladdierer mit Lee-Gattern durch Optimierung bezüglich verschiedener Kriterien bestimmt. Die Kriterien sind namentlich die durchschnittliche Verlustleistung \bar{P} , die Verzögerungszeit t_d , das Power-Delay-Produkt $\bar{P} \cdot t_d$ und für das einfache Lee-Gatter die Robustheit der Schaltung. Die Robustheit sei an dieser Stelle als relative Robustheit r definiert. Sie bestimmt die

Abbildung 4.20: Transistorlayout mit Weite kleiner als $1,0 \mu m$.

Technologie	minimale Weite	minimale Länge	Schrittweite
0,35	1,0	0,3	0,05
0,6	1,4	0,6	0,05
0,8	2,0	0,8	0,05

Tabelle 4.5: Transistorgeometrien für die verwendeten Technologien (in μm)

maximale Schwankung des Eingangssignals, normiert auf die Versorgungsspannung, bei der die Spezifikationen noch eingehalten werden ($r = \frac{\Delta V}{V_{DD}}$).

Da hier die Logikfunktionen in statischer Schaltungstechnik realisiert sind, kann nicht von einem Takt in seiner unmittelbaren Bedeutung gesprochen werden. Das Äquivalent hierzu stellt die Taktfrequenz im Sinne von Änderungsfrequenz am Eingang dar, mit der die Schaltung betrieben wird. Deren Berücksichtigung bei einer Gegenüberstellung verschiedener Schaltungstechniken ist wegen der großen Diskrepanz zwischen statischer und dynamischer Verlustleistung im Falle der statischen CMOS Technik von Bedeutung. Die Simulationsergebnisse in diesem Kapitel beziehen sich auf eine beliebig gewählte Taktfrequenz von $250 MHz$, entsprechend dem beispielhaft gewählten Eingangssignal in Kapitel 4.1.1.

Die Spezifikationen für das Majoritätsgatter nach Lee seien hier der Einfachheit halber identisch für alle drei Technologien und unabhängig von der Zahl der Eingänge gegeben. Die maximale Verlustleistung sei zu $500 \mu W$ und die maximale Verzögerungszeit auf $2 ns$ beschränkt. Der Ausgang ist mit der Lastkapazität $C_L = 50 fF$ belastet, identisch zu dem folgenden Addiererbeispiel. Die Ergebnisse der Optimierung sind in Tabelle 4.6 aufgelistet. Anhand

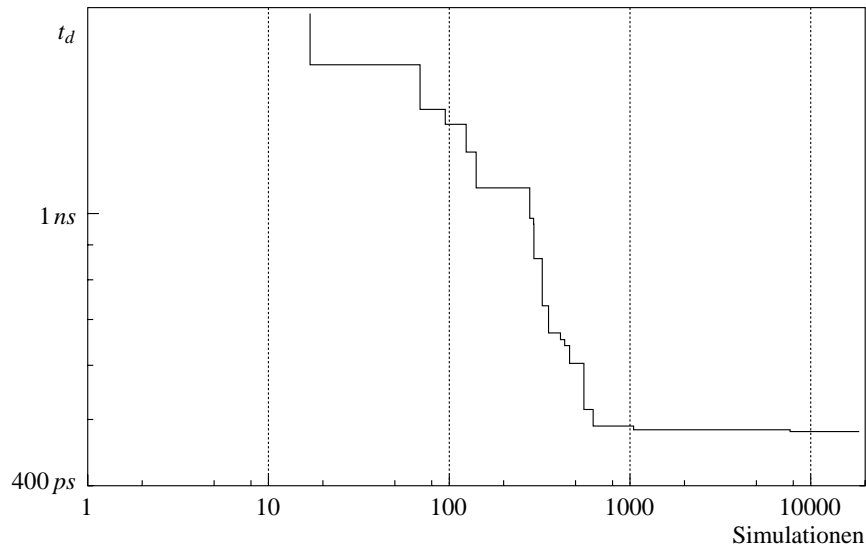


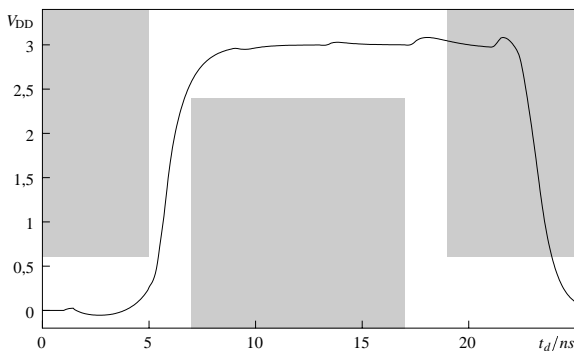
Abbildung 4.21: Delayoptimierung des Lee-Gatters: Optimierungsverlauf als Verzögerungszeit der bis dato besten Lösung in Abhängigkeit von der Zahl der durchgeführten Simulationen. Zu den markierten Positionen finden sich die optimalen Lösungen in Abbildung 4.22 .

der Simulationsergebnisse lässt sich die Skalierung der Verlustleistung mit der verwendeten Technologie ablesen.

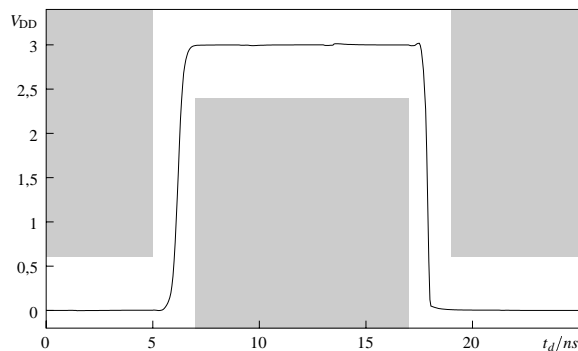
Technologie	Verlustleistung	Delay	Power-Delay-Produkt	Robustheit
0,35 μm	46 μW	753 ps	91 fJ	9,7 %
0,6 μm	74 μW	963 ps	148 fJ	8,6 %
0,8 μm	205 μW	1476 ps	410 fJ	5,9 %

Tabelle 4.6: Ergebnisse des Majoritätsgatters nach Lee.

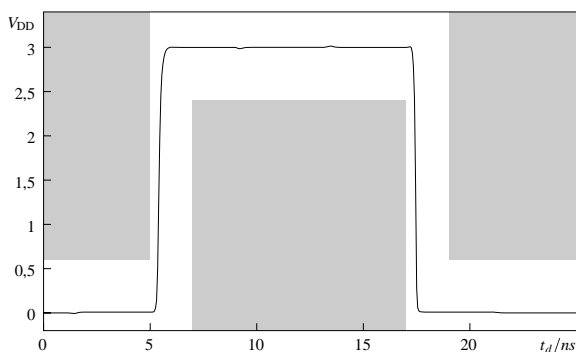
Für die Optimierung der Verzögerungszeit ist ein Optimierungslauf in Abbildung 4.21 dargestellt. Das Delay der bislang schnellsten Schaltung ist in Abhängigkeit von der Zahl der SPICE-Simulationen dargestellt. Die zugehörigen Lösungen im Zeitbereich nach 10, 100, 1000 und 10000 Simulationen sind in der Abbildung 4.22 festgehalten. Es ist gut zu beobachten, wie sich das Ausgangssignal im zulässigen Bereich zu immer früheren Zeitpunkten verschiebt, wobei die Differenz zwischen 4.22(c) und 4.22(d) nur noch marginal ist.



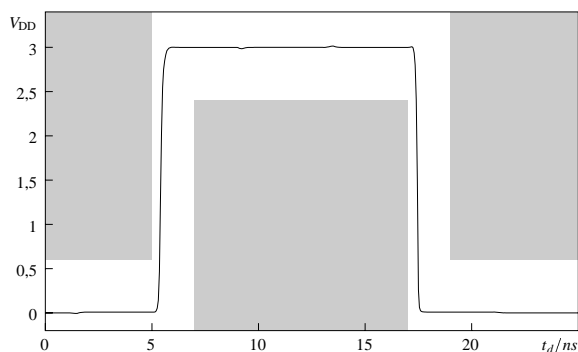
(a) Beste Lösung nach 10 Simulationen. Die logische Funktion wird nicht korrekt erfüllt.



(b) Beste Lösung nach 100 Simulationen, mit einer korrekten logischen Funktion ($t_d = 1,3 ns$).



(c) Beste Lösung nach 1.000 Simulationen ($t_d = 488 ps$).



(d) Beste Lösung nach 10.000 Simulationen ($t_d = 480 ps$).

Abbildung 4.22: Momentaufnahmen des Ausgangssignals der Lösungen für das Lee-Gatter aus dem Optimierungsverlauf (Abb. 4.21).

Auffällig an den erhaltenen Ergebnissen für die verlustleistungsoptimierte Schaltung ist das W/L -Verhältnis der eingangsseitigen n-Kanal-Transistoren, das kleiner als 1 ist. Ein solches, eher unerwartetes Ergebnis zeigt, wie wichtig der Erhalt des gesamten Variationsspektrums für die Optimierung ist.

Im Falle der Volladdierer gestaltet sich die Zielsetzung bzw. Spezifikation schwieriger. Dies ergibt sich aus der unterschiedlichen Semantik der Ein- und Ausgänge. Obwohl alle drei Eingänge logisch gleichwertig sind, kommt dem Carry, wie schon bei der Darstellung des Volladdierers in Kapitel 4.1.1 erwähnt, eine gesonderte Bedeutung zu. Ebenso hängen die zeitlichen Anforderungen an die Ausgänge stark von den Einsatzbedingungen ab. So liegt bei der Realisierung eines Ripple-Carry-Addierers wie in Abbildung 4.2 der kritische Pfad jeweils zwischen eingangsseitigem und ausgangsseitigem Carry. Der Einfluss der beiden übrigen Eingänge und die Summe sind bezüglich zeitlicher Anforderungen unkritisch und können somit immer minimal bzw. verlustleistungsoptimiert entworfen werden. Wird hingegen der Volladdierer in einem Feldmultiplizierer [Pir96] eingesetzt, können die zeitlichen Anforderungen an die beiden Ausgänge nicht so einfach formuliert werden. Für diesen Fall wird hier die Delayforderung für Summe und Carry gleichgesetzt, um die Beschränkung auf eine Zielgröße bei der zeitlichen Optimierung beizubehalten. Die Verlustleistung sei auf $\bar{P} = 500 \mu W$ begrenzt. Für ein Carry optimiertes Design ist die maximale Verzögerung mit $t_{d,C} = 2ns$ und für einen Volladdierer in einem Feldmultiplizierer mit $t_{d,CS} = 2ns$ gegeben. Die Ergebnisse der Optimierung für beide Anwendungsbereiche hinsichtlich Verlustleistung, Verzögerungszeit und Power-Delay Produkt sind in der Tabelle 4.7 aufgelistet.

Für den Volladdierer mit Lee-Gattern ist mit den geforderten Grenzen für Verlustleistung und Verzögerungszeit keine Realisierung in der $0,6 \mu m$ - oder der $0,8 \mu m$ -Technologie möglich. Nur für das Carry optimierte Design in $0,35 \mu m$ -Technologie ergeben sich Lösungen für $\bar{P} = 281 \mu W$, $t_d = 1074 ps$ und $\bar{P} \cdot t_d = 448 fJ$. Unter Verzicht auf die Verlustleistungsbeschränkung lassen sich für die Carry und Summen optimierende Spezifikation fol-

Technologie	Carry			Summe und Carry		
	\bar{P}	t_d	$\bar{P} \cdot t_d$	\bar{P}	t_d	$\bar{P} \cdot t_d$
$0,35 \mu m$	$56 \mu W$	$387 ps$	$30 fJ$	$74 \mu W$	$752 ps$	$126 fJ$
$0,6 \mu m$	$71 \mu W$	$734 ps$	$125 fJ$	$191 \mu W$	$1201 ps$	$359 fJ$
$0,8 \mu m$	$102 \mu W$	$906 ps$	$187 fJ$	$333 \mu W$	$1609 ps$	$587 fJ$

Tabelle 4.7: Ergebnisse des Spiegeladdierers.

gende Verlustleistungswerte bestimmen: $\bar{P}_{0,35 \mu m} = 1,14 \text{ mW}$, $\bar{P}_{0,60 \mu m} = 3,27 \text{ mW}$ und $\bar{P}_{0,80 \mu m} = 6,74 \text{ mW}$.

Das Gradientenabstiegsverfahren wird hier alternativ zu den evolutionären Algorithmen für die Verlustleistungsoptimierung angewendet. Mit Blick auf die Design-Rules zeigt sich jedoch die Problematik, dass das Gradientenverfahren nicht auf einem diskreten Suchraum, sondern im \mathbb{R}^n definiert ist. Die Optimierung wird daher im \mathbb{R}^n (durch Intervalle begrenzt) durchgeführt und die erhaltene Lösung ist anschließend wieder zu diskretisieren. Da das Ergebnis der Optimierung typischerweise am Rand des Bereiches korrekt funktionierender Schaltungen zu finden sein wird, kann bei der Diskretisierung nicht beliebig auf- oder abgerundet werden. Dieser Vorgang zur Anpassung an die Design-Rules wird bei klassischer CMOS Technik vergleichsweise unkritisch sein. Bei dem Lee-Gatter jedoch sind die Verhältnisse der Transistorweiten entscheidend für die logische Schaltfunktion. Somit ist es möglich, dass keine Diskretisierung unter Erhaltung der korrekten Schaltfunktion vorgenommen werden kann.

Unbeachtet dessen beziehen sich die folgenden Optimierungsergebnisse auf den reellwertigen Parameterraum. Für den Inverter liefert das Gradientenabstiegsverfahren eine gleichwertige Lösung zum evolutionären Algorithmus mit $\bar{P} = 15 \mu W$. Bei dem Spiegeladdierer (Optimierung bezüglich Carry und Summe) ist das Ergebnis stark von dem Startwert beeinflusst. Für verschiedene Zwischenergebnisse (jeweils einer korrekt funktionierenden Schaltung) des evolutionären Algorithmus' ist das Gradientenverfahren angewendet worden. Die geringste erreichte Verlustleistung von $\bar{P} = 55 \mu W$, verbessert das Resultat der evolutionären Algorithmen um 25%. Zufällige, nicht korrekte Schaltungen als Startwerte führten hingegen zu einem Wert von $\bar{P} = 113 \mu W$, 52% über der EA-Lösung.

Die Optimierungen für das Lee-Gatter und den Lee-Addierer sind wiederum, ausgehend von Zwischenlösungen des evolutionären Algorithmus mit korrekter logischer Schaltfunktion, mehrfach durchgeführt worden. Für das Lee-Gatter wurden $55 \mu W$ für die Verlustleistung erreicht. Dies sind 19% mehr als mit dem evolutionären Ansatz. Die minimale Verlustleistung des Addierers wurde zu $1,3 \text{ mW}$ bestimmt, 14% mehr gegenüber der mit EA ermittelten Lösung. Für beide konnte jedoch, bei Verwendung einer zufälligen Schaltung mit nicht korrekter logischer Funktion als Startwert, keine funktionierende Schaltung durch das Gradientenabstiegsverfahren generiert werden.

Kapitel 5

Vektoroptimierung

Nicht triviale Systeme in realen Umgebungen sind durch eine Vielzahl sowohl von Freiheitsgraden als auch von Anforderungen gekennzeichnet. Der Wunsch nach einer Verbesserung solcher Systeme führt zu Optimierungsproblemen unter Berücksichtigung mehrerer Kriterien. Deren Reduzierung auf eine einzige Zielgröße ist zumeist zur Vereinfachung der Problematik künstlich herbeigeführt. Solch ein Vorgehen wird der ursprünglichen Aufgabenstellung jedoch nicht in jedem Falle ausreichend gerecht.

5.1 Grundlagen und Einführung

Bei der Vektoroptimierung oder Mehrgrößenoptimierung (multi objective optimization) werden sämtliche Kriterien berücksichtigt. Es werden mehrere, möglicherweise konkurrierende Zielfunktionen zeitgleich betrachtet und optimiert. Hierbei wird selten eine einzelne perfekte Lösung zu finden sein, die sämtliche Kriterien optimal befriedigt. In der Regel wird eine Menge von Alternativen existieren, die ohne nähere Kenntnis über die Relevanz der Kriterien als gleichwertig anzusehen sind. Dieser Umstand besteht bereits bei zwei konkurrierenden Kriterien. In Abbildung 5.1 ist als Beispiel der Zielgrößenraum einer Schaltung in Bezug auf Verlustleistung und Verzögerungszeit dargestellt. Der grau schattierte Bereich bezeichnet mögliche Realisierungen dieser Schaltung. Da beide Zielgrößen zu minimieren sind, beschreibt die durchgezogene Linie die optimalen Lösungen der Schaltung. Für diese kann kein Kriterium verbessert werden, ohne das jeweils andere zu verschlechtern. Diese Lösungsmenge wird

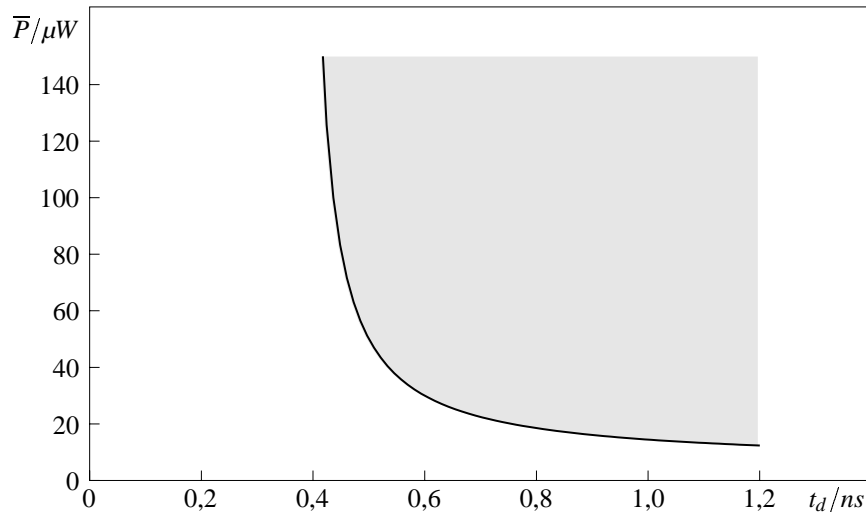


Abbildung 5.1: Beispiel für den Zielgrößenraum einer Schaltung.

als Pareto Menge bezeichnet. Ziel der Vektoroptimierung im Hinblick auf die Anwendung in dieser Arbeit ist die Bestimmung der Pareto Menge.

Zunächst seien an dieser Stelle einige Begriffe geklärt, die zum Verständnis und der Interpretation der Ergebnisse der Vektoroptimierung erforderlich sind. Die relative Position zweier Elemente oder Realisierungen A und B im Zielgrößenraum gibt Aufschluss darüber, welches der beiden im Hinblick auf das Optimierungsziel zu bevorzugen ist. Stellt sich A in sämtlichen Kriterien als besser im Vergleich zu B heraus, so ist A zu wählen. Ist B bezüglich jeder Zielgröße besser als A , so wähle man B . In diesen Fällen wird davon gesprochen, dass das bessere Element das andere dominiert. Ist jedoch die Rangfolge nicht bei jeder Zielgröße identisch, d.h. zum Beispiel bezüglich des ersten Kriteriums sei A besser und bezüglich des zweiten B , so ist die Entscheidung indifferent und die Beziehung zwischen beiden ist nicht dominant (Abbildung 5.2).

Spezifischer ist die formale Beschreibung der Dominanz und deren Negation wie in Definition 1 und 2. Wie in den Kapiteln zuvor wird hier ebenfalls nur der Fall Minimierung betrachtet. Mit diesen Definitionen folgt, dass ein Element nicht zu sich selbst dominant ist ($x \not\prec x$). Ferner ist aus einer nicht dominanten Beziehung keine Folgerung zur Dominanz der umgekehrten Beziehung möglich ($x_A \not\prec x_B \not\Rightarrow x_B \prec x_A$ und $x_A \not\prec x_B \not\Rightarrow x_B \not\prec x_A$). Die Dominanz sei sowohl im Parameterraum als auch im Zielgrößenraum definiert ($\vec{x}_A \prec \vec{x}_B \Leftrightarrow \vec{f}(\vec{x}_A) \prec \vec{f}(\vec{x}_B)$).

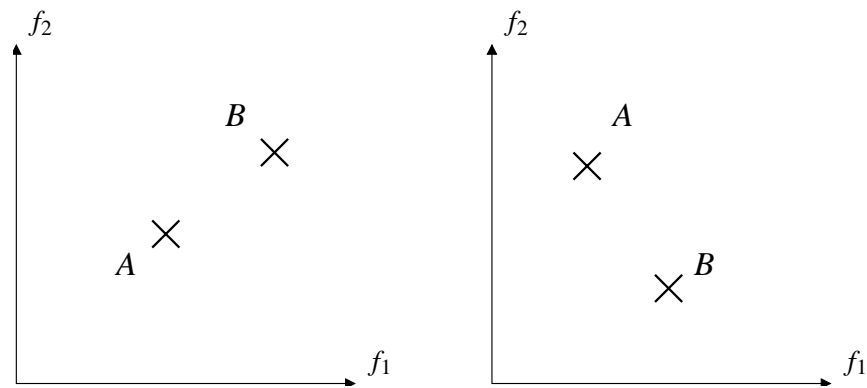


Abbildung 5.2: Relative Position zweier Elemente im zweidimensionalen Zielgrößenraum. Im linken Fall dominiert A die Lösung B ($A \prec B$). Im zweiten Fall rechts finden sich zwei nicht dominierte Elemente ($A \not\prec B$ und $B \not\prec A$).

Definition 1 (Dominanz)

Ein Punkt x_A heißt dominant gegenüber einem Punkt x_B ($x_A \prec x_B$)
 gdw.

$$\forall i \quad f_i(x_A) \leq f_i(x_B) \quad (5.1)$$

$$\wedge \exists i \quad f_i(x_A) < f_i(x_B) . \quad (5.2)$$

Definition 2

Zwei Punkte x_A und x_B heißen zueinander nicht dominant
 gdw. $x_A \not\prec x_B \quad \wedge \quad x_B \not\prec x_A$.

Für jede endliche Teilmenge der möglichen Realisierungen lässt sich somit die Menge der nicht dominierten Lösungen durch einen paarweisen Vergleich bestimmen. Wird die Menge aller möglichen Realisierungen betrachtet, so ist die Teilmenge der nicht dominierten Elemente gleich der Pareto Menge.

Definition 3 (Pareto optimal)

Ein Punkt \vec{x}_P ist Pareto optimal über einer Menge X
 gdw. $\forall \vec{x} \in X$ gilt $\vec{x} \not\prec \vec{x}_P$.

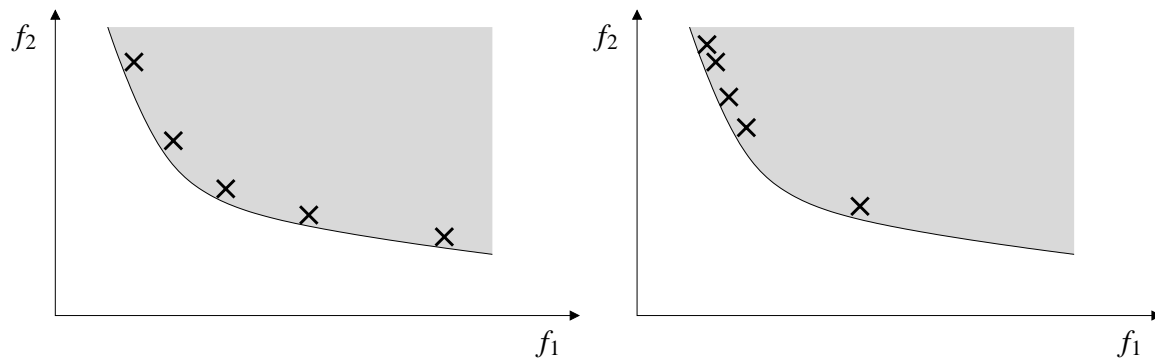


Abbildung 5.3: Zwei Beispiele einer Überdeckung. Links eine gleichmäßig verteilte Überdeckung, die die Charakteristik der Pareto Menge nachbildet. Rechts hingegen Konzentration auf einen Teilbereich.

Die Lösungsmenge eines Optimierungslaufes ist immer schlechter, bestenfalls gleich der Elemente der Pareto Menge, d.h. die Lösungsmenge stellt allenfalls eine Annäherung an die Pareto Menge dar. In der Literatur findet sich in diesem Zusammenhang der Begriff der Überdeckung (Coverage) [Zit99]. Ziel einer Mehrgrößenoptimierung ist also die Bestimmung einer ‚möglichst guten‘ Überdeckung der Pareto Menge. Unter ‚möglichst gut‘ ist einerseits zu verstehen, dass die Elemente der Lösungsmenge nahe den Elementen der Pareto Menge sind. Andererseits bedeutet dies, dass die Gesamtheit der Pareto Menge widergespiegelt wird und nicht nur eine Teilmenge derer, damit aus der Überdeckung die Charakteristik der Pareto Menge ersichtlich wird. Dies wird in Abbildung 5.3 veranschaulicht.

Klassische Optimierungsverfahren bestimmen eine singuläre Lösung, d.h. in obigem Sprachgebrauch, dass sie eine einelementige Überdeckung der Pareto Menge liefern. Eine solche Lösung kann natürlich in Bezug auf die Möglichkeit die Charakteristik der Pareto Menge zu erkennen keine ‚gute‘ Überdeckung sein. Populationsbasierte Optimierungsverfahren, die statt einer singulären Lösung eine Menge von Alternativen ermitteln, bieten dem gegenüber die Chance, eine Überdeckung der Pareto Menge in einem einzigen Optimierungslauf bestimmen zu können. In dieser Arbeit werden evolutionäre Algorithmen als populationsbasiertes Verfahren verwendet, um die Pareto Menge bzw. eine Überdeckung derer zu bestimmen.

5.2 Klassische Ansätze zur Bestimmung der Pareto Menge

Die klassischen Optimierungsverfahren sind typischerweise nur auf eine einzige Zielgröße anzuwenden. Deren Erweiterung auf eine Mehrgrößenoptimierung ist nur schwer möglich [FF95] und wird in der Praxis durch eine Neuformulierung der Problemstellung umgangen, wie in Kapitel 4.2 dargestellt. Die genannten Techniken sind unter der dortigen Zielsetzung, der Bestimmung einer einzelnen Realisierung des Systems anwendbar. Bezüglich der hier vorliegenden Fragestellung stellen diese Ansätze jedoch keine Lösung dar. Im folgenden Kapitel werden Eigenschaften der gesuchten Pareto Menge diskutiert, auf die in den darauf folgenden Lösungsansätzen Bezug genommen wird.

5.2.1 Eigenschaften der Pareto Menge

Wie oben erwähnt, ist die Pareto Menge durch die Menge der nicht dominierten Elemente bezüglich der Gesamtheit aller Realisierungen gegeben. Entgegen dieser globalen Anwendung der Dominanzrelation kann eine lokale Dominanz eingesetzt werden [Deb98]. Hierzu werden nur die Elemente innerhalb einer ε -Umgebung betrachtet, so dass sich Definition 4 für lokal Pareto optimale Elemente ergibt.

Definition 4 (lokal Pareto optimal)

Ein Punkt \vec{x}_P ist lokal Pareto optimal über einer Menge X

gdw. $\forall \vec{x} \in X$ mit $|\vec{f}(\vec{x}) - \vec{f}(\vec{x}_P)| < \varepsilon$ gilt $\vec{x} \not\prec \vec{x}_P$.

Die Anwendung der lokalen Form erlauben einen noch spezifischeren Blick in den Zielgrößenraum. Für ein weiteres Beispiel sei der Zielgrößenraum in Abbildung 5.4 gegeben, in dem im ersten Fall die (globale) Pareto Menge gekennzeichnet ist, während im zweiten Fall die Menge der lokal Pareto optimalen Elemente markiert ist.

Eine wichtiges Charakteristikum bei der mehrkriteriellen Optimierung liefert die Form der Pareto Menge. Eine Pareto Menge wird als konvex bezeichnet, wenn folgendes gilt: Jeder Punkt im Zielgrößenraum auf der Strecke zwischen zwei Pareto optimalen Punkten ist selbst Pareto optimal. Andernfalls existiert ein weiteres Element, das diesen Punkt dominiert (vergleiche Definition 5). Daraus folgt, dass ein konvexer Lösungsraum im Zielgrößenraum zu einer konvexen Pareto Menge führt. Beispiele für eine konvexe und eine nicht konvexe Menge sind in Abbildung 5.5 illustriert.

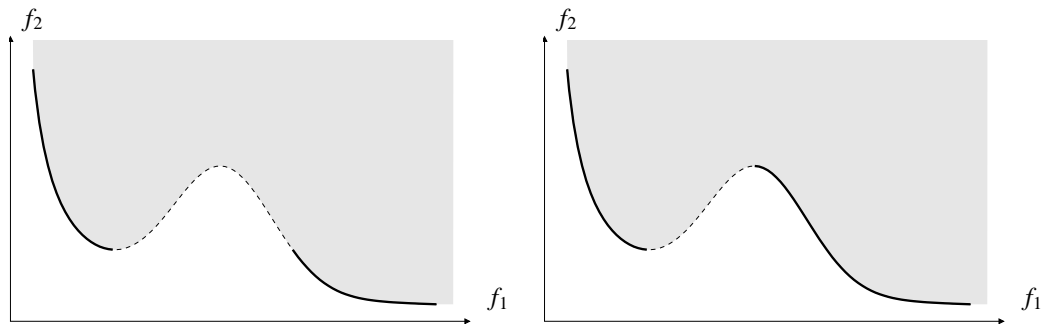


Abbildung 5.4: Beispiel lokal Pareto optimaler Punkte. Der Bereich möglicher Lösungen im Zielgrößenraum ist grau schattiert und dessen Rand gestrichelt dargestellt. Auf der linken Seite ist die Pareto Menge markiert, während rechts die lokal Pareto optimalen Punkte hervorgehoben sind (durchgezogene Linie).

Definition 5 (konvexe Pareto Menge)

Eine Pareto Menge X_P heißt konvex

gdw. $\forall \vec{x}_1, \vec{x}_2 \in X_P$ und $\lambda \in [0; 1]$ gilt:

$$\exists \vec{x} \in X_P \quad \text{mit} \quad \vec{f}(\vec{x}) = \lambda \vec{f}(\vec{x}_1) + (1 - \lambda) \vec{f}(\vec{x}_2) \quad (5.3)$$

oder

$$\exists \vec{x} \in X \quad \text{mit} \quad \vec{f}(\vec{x}) \prec \lambda \vec{f}(\vec{x}_1) + (1 - \lambda) \vec{f}(\vec{x}_2) \quad (5.4)$$

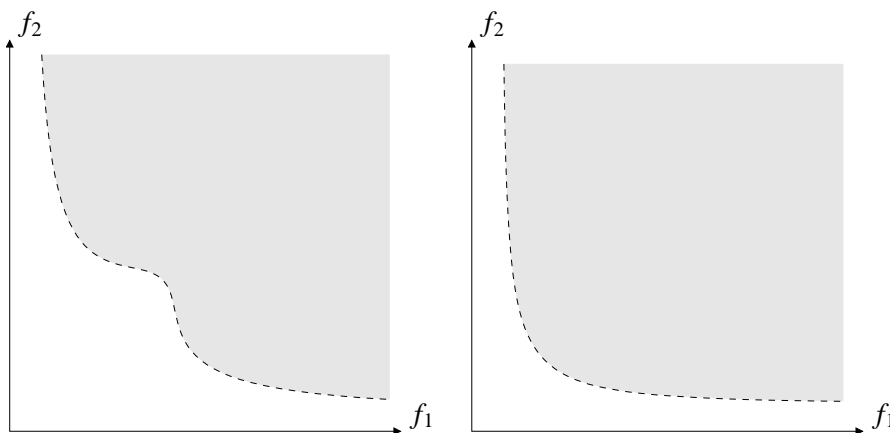


Abbildung 5.5: Darstellung einer nicht konvexen Pareto Menge auf der linken Seite gegenüber einer konvexen rechts.

Als Letztes sei hier eine Eigenschaft der Pareto Menge erwähnt, die von van Veldhuizen und Lamont dokumentiert worden ist [vVL99]. Die Dimension der Pareto Menge wächst nicht mit der Zahl der Zielfunktionen, sondern ist auf die Dimension Zwei begrenzt. Diese Eigenschaft findet in Kapitel 5.3.1 wieder Berücksichtigung.

5.2.2 Gewichtsmethode

Die Aufgabenstellung der Mehrgrößenoptimierung lässt sich, wie in Kapitel 4.2 bereits gezeigt, durch Einführen einer linearen Gewichtung der einzelnen Komponenten in ein skalares Optimierungsproblem überführen. Die Zielfunktion $f_{\vec{w}}(\vec{x}) = \sum_i w_i f_i(\vec{x})$ ist über den Vektor \vec{w} einzustellen, und die Lösung des neu formulierten Optimierungsproblems ist ein Element der Pareto Menge. Durch mehrfache Optimierungsläufe mit unterschiedlichen Richtungen für den Gewichtsvektor lässt sich somit eine Überdeckung der Pareto Menge mit einem skalaren Optimierungsverfahren bestimmen. Nachteilig für dieses Verfahren ist die beschränkte Anwendbarkeit, da es nicht geeignet ist, nicht konvexe Pareto Mengen zu bestimmen [SD95, Zit99, Deb00].

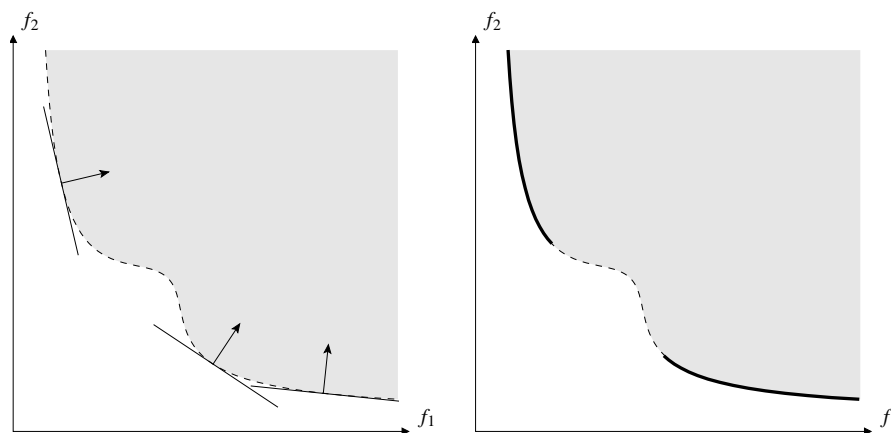


Abbildung 5.6: Bestimmung der Pareto Menge mit der Gewichtsmethode. Links sind Lösungspunkte für unterschiedliche Gewichte zu sehen, die zusammengefasst die rechts dargestellte Überdeckung liefern.

5.2.3 Beschränkungsmethode

Für die Rückführung des Mehrgrößenoptimierungsproblems auf eine Problemstellung bezüglich einer einzigen Größe werden bei der Beschränkungsmethode zusätzliche Restriktionen eingeführt. Dieses Vorgehen wurde in Kapitel 4.2 vorgestellt. Die Pareto Menge kann durch mehrfache Optimierungsläufe mit variierten Restriktionen bestimmt werden. Für n Zielgrößen werden $n - 1$ Restriktionen der Form $f_i(\cdot) < \varepsilon_i$ eingesetzt, die für unterschiedliche ε_i meist zu verschiedenen Elementen der Pareto Menge führen und so eine Überdeckung derer ergibt. In Abbildung 5.7 findet sich eine grafische Darstellung für verschiedene ε und den daraus folgenden Lösungen. Dieser Ansatz ist zur Bestimmung einer nicht konvexen Pareto Menge geeignet, weist jedoch ein Problem durch die erforderliche Wahl der ε_i auf. Es ist zu beachten, dass für ungünstige Werte die Lösungsmenge leer sein kann. Somit ist also Vorwissen über die Problemstruktur für die Anwendung der Beschränkungsmethode erforderlich [Zit99, Deb00].

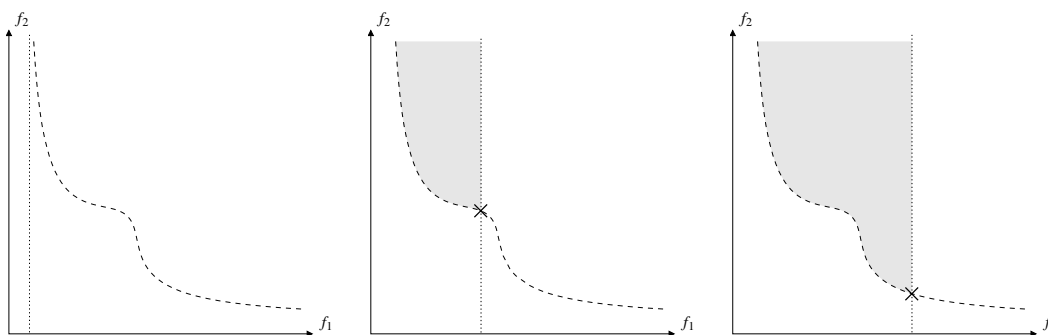


Abbildung 5.7: Beispiel zur Anwendung der Beschränkungsmethode für verschiedene ε . Die gepunktete Linie gibt die Maximumgrenze für f_1 wieder, so dass nur f_2 zu minimieren bleibt. Im Bild links ist die Lösungsmenge leer und in den beiden weiteren das Optimum mit \times gekennzeichnet.

Eine Variante der Beschränkungsmethode ergibt sich durch Ersetzen der Maximumbeschränkungen mittels Gleichungsrestriktionen $f_i(\cdot) = \varepsilon_i$. Hierdurch ist statt der Pareto Menge oder der lokal Pareto optimalen Elemente die Bestimmung des gesamten funktionalen Zusammenhangs zwischen mehreren Parametern möglich. Im Falle nicht bijektiver Abbildungen ist allerdings die Wahl der freien bzw. zu optimierenden Zielgröße nicht mehr beliebig. Entscheidender ist jedoch die Tatsache, dass nach der Reduktion des Suchraumes die Optimierung typischerweise schwerer durchzuführen ist, als dies im ursprünglichen Parameterraum der Fall war [Deb00].

5.3 Bestimmung der Pareto Menge mit evolutionären Algorithmen

Die evolutionären Algorithmen als populationsbasiertes Optimierungsverfahren erlauben, in einem einzigen Suchlauf eine Menge von Lösungsalternativen zu bestimmen. Diese kann eine geeignete Überdeckung der Pareto Menge sein. Schaffer [Sch85, in [FF95]] zeigte einen ersten Ansatz, eine Menge nicht dominierter Lösungen zu ermitteln. Das Verfahren (VEGA=vector evaluated genetic algorithm) neigt jedoch zur Bildung einiger weniger Insellösungen. Dies ist ein Effekt, der jedem einfachen evolutionären Algorithmus anhaftet, da eine Population endlicher Größe generell dazu tendiert, einzelne Lösungen zu bevorzugen [FF95]. Verantwortlich hierfür sind statistische Fehler, die die Funktion der Operatoren des EA stören [Zit99]. Zur Vermeidung dieses Effektes und Erhaltung der Diversität der Suche, um eine möglichst gleichverteilte Überdeckung der Pareto Menge zu fördern, hat sich die so genannte Nischenbildung (niche induction techniques) etabliert. Eine Nische ist als räumlich eng begrenzter Bereich zu verstehen, in dem sich mehrere Lösungen wiederfinden. Goldberg [GR87, Gol89] hat diese Technik in Form der Ressourcenteilung (fitness sharing) eingeführt. Die Bewertung eines Individuums, die dessen Reproduktionswahrscheinlichkeit bestimmt, wird in Abhängigkeit der benachbarten Individuen festgelegt. Die Idee hierbei ist, eine annähernd gleiche Reproduktionswahrscheinlichkeit in unterschiedlichen Nischen zu erreichen. Diese sei unabhängig von der Zahl der Individuen, die die Nische bilden.

Für die Definition einer Nische ist ein Distanzmaß erforderlich, das die Nähe zweier Elemente zueinander erfasst. Im Falle genetischer Algorithmen kann dieses Maß bezüglich des Genotyps oder des Phänotyps angewendet werden. In den in dieser Arbeit verwendeten Evolutionsstrategien ist die Abbildung zwischen beiden identisch und somit diese Unterscheidung nicht erforderlich. Des Weiteren kann die Nischenbildung im Parameterraum [SD95] oder im Zielgrößenraum [FF95] erfolgen. Im ersten Fall wird die Diversität im Parameterraum unterstützt, während im zweiten eine gleichmäßige Überdeckung der Pareto Menge im Vordergrund steht. Problematisch an diesem Ansatz der Ressourcenteilung ist die exogene Wahl des Distanzmaßes, die entsprechendes Vorwissen über das zu optimierende System erfordert.

Der in dieser Arbeit angewendete evolutionäre Algorithmus, SPEA (strength pareto evolutionary algorithm) kann ohne derartige Systemkenntnisse eingesetzt werden. Die Nischenbildung leitet sich allein aus der relativen Position der Individuen im Zielgrößenraum ab. Dieses Verfahren wird in Kapitel 5.3.2 im Detail dargestellt.

Ein völlig anderer Ansatz die Bildung von Nischen zu fördern, ist die Abkehr von einer global vernetzten Population. In einer strukturierten Population existiert eine Nachbarschaftsbeziehung zwischen den Individuen, die zu einem lokalen Informationsaustausch führt. Die Begrenzung des Informationsflusses mindert die Möglichkeit eines einzelnen Individuums, die Population zu dominieren. Das Räuber-Beute Modell basiert auf einer solchen strukturierten Population und wird im Folgenden genauer betrachtet.

5.3.1 Räuber-Beute Modell

In der Vektoroptimierung sind mehrere konkurrierende Zielgrößen zu verbessern bzw. zu optimieren. Der vorliegende Ansatz entlehnt sich aus einem Bild der Natur, in dem Beutetiere von unterschiedlichen Räubern gejagt werden und sich unterschiedlich optimal auf die vorliegenden Jagdstrategien einstellen. Aus diesem Szenario ist das Räuber-Beute Modell [LRS98] abgeleitet. Der Beute entsprechen die einzelnen Individuen, hier Realisierungen einer Schaltung. Die Räuber spiegeln sich in dem Selektionsmechanismus wider, der unterschiedliche Zielfunktionen, d.h. verschiedene Komponenten der ursprünglichen Zielfunktion \vec{f} berücksichtigt. Bezüglich dieser skalaren Kriterien existiert dann jeweils wieder eine Ordnung. Die Individuen sind in einer Vierer-Nachbarschaft geordnet. Das so aufgespannte Netz ist zu einem Torus strukturiert. Die Räuber wandern über diese Populationsstruktur und wenden ihren Selektionsoperator lokal an, d.h. es wird nur das besuchte Element und dessen direkten Nachbarn betrachtet. Ist die Zahl der Räuber klein gegenüber der Zahl der Individuen, werden sich die Selektionsoperatoren nur in Ausnahmefällen gegenseitig stören. Der hier vorgestellte Algorithmus ist eine mögliche Umsetzung des allgemeinen von Laumanns, Rudolph und Schwefel vorgestellten Konzeptes.

Die Individuen seien in einer $N \times M$ -Matrix gegeben. Die Nachbarelemente des Individuums $I_{p,q}$ ergeben sich dann zu $I_{p-1,q}$, $I_{p+1,q}$, $I_{p,q-1}$ und $I_{p,q+1}$. Für die toroide Struktur sind die Elemente $I_{0,q}$ bzw. $I_{p,0}$ mit $I_{N,q}$ bzw. $I_{p,M}$ gleichzusetzen. Ebenso sind $I_{N+1,q}$ und $I_{p,M+1}$ durch $I_{1,q}$ und $I_{p,1}$ definiert. Jedem Räuber ist eine Position auf dieser Struktur, d.h. ein Individuum, und zudem ein Selektionskriterium zugeordnet. Wandert der Räuber einen Schritt so bedeutet dies, dass sich die Bezugsposition zu einem der direkten Nachbarn des Individuums ändert. Die Reproduktion und die Selektion wirken nur lokal auf das Individuum der aktuellen Position und sind nur durch dessen direkten Nachbarn beeinflusst. Die Reproduktion beschränkt sich bei diesem Algorithmus auf die Mutation, da sich die Rekombination bereits in ersten Untersuchungen [LRS98] als ungeeignet erwiesen hat.

Für die Selektion wird das Individuum an der aktuellen Position mit seinen Nachbarn verglichen. Ist das aktuelle Individuum das Schlechteste dieser fünf, so wird es ersetzt. Hierzu wird ein zufällig gewähltes Nachbarelement kopiert und anschließend mutiert.

Bezogen auf die Population als Ganzes wird zunächst von allen Räubern die Selektion vorgenommen. Als Zweites werden sämtliche eliminierten Individuen durch Reproduktion ersetzt und abschließend wandern die Räuber einen Schritt in eine zufällige Richtung weiter. Diese drei Schritte wiederholen sich bis zum Erreichen eines geeigneten Abbruchkriteriums. Für das Räuber-Beute Modell wird keine Selbstadaption eingesetzt. Stattdessen wird eine globale Mutationsweite verwendet, die jedoch Vorwissen über das System erfordert.

Die Netzstruktur der Individuen legt sich im Laufe der Optimierung als Überdeckung über die Pareto Menge. Im Idealfall repräsentieren benachbarte Elemente der Struktur nahe beieinander liegende Punkte im Zielgrößenraum. Da die Pareto Menge maximal von der Dimension Zwei ist (vergleiche Kapitel 5.2.1), ist die vorliegende Netzstruktur ausreichend, um eine geeignete Überdeckung der Pareto Menge erreichen zu können.

Das Netz der Individuen ist hier zu 30×30 festgelegt, auf der 200 Räuber verwendet werden. Letztere teilen sich gleichmäßig auf die verschiedenen Zielgrößen auf. Diese Werte entsprechen den in der Literatur [LRS98] verwendeten.

5.3.2 Strength Pareto Evolutionary Algorithm (SPEA)

Aus den Erfahrungen und Eigenschaften älterer Varianten evolutionärer Algorithmen für die Vektoroptimierung ist ein Ansatz entwickelt worden, der von Zitzler und Thiele [Zit99] vorgestellt wurde. Der Strength Pareto Evolutionary Algorithm (SPEA) soll wie das Räuber-Beute Modell eine komplette Überdeckung der Pareto Menge in einem Optimierungslauf bestimmen. Die charakterisierenden Eigenschaften des Ansatzes vereinen Merkmale bestehender evolutionärer Algorithmen mit neuen Techniken.

Die gefundenen, nicht dominierten Lösungen werden beim SPEA separat gespeichert und als externe Population E' bezeichnet. Die ursprüngliche Population E wird zum Zwecke der eindeutigen Identifizierung intern genannt. Die Zahl der nicht dominierten Elemente wird mit Hilfe eines Clustering-Verfahrens auf eine feste Größe begrenzt. Die Bewertung der Elemente basiert auf der Dominanzrelation. Berücksichtigt wird jedoch nur die Relation der Elemente der internen Population zu denen der externen Population. Die Beziehung der internen Ele-

Tabelle 5.1: Algorithmus des Strength Pareto Evolutionary Algorithm.

1. Generieren einer Anfangspopulation $E^{(0)}$ und einer leeren Menge nicht dominierter Elemente $E'^{(0)}$ (externe Population).
2. Kopieren der in $E^{(g)}$ nicht dominierten Elemente in die externe Population E' .
3. Reduzieren der Menge E' auf nicht dominierte Elemente.
4. Verkleinern der externen Population durch Clustering, falls $|E'|$ ein gegebenes Maximum N' überschreitet.
5. Bestimmung der Bewertung der Elemente in $E^{(g)}$ und $E'^{(g)}$.
6. Selektieren der Elemente aus $E^{(g)} \cup E'^{(g)}$ die durch Reproduktion die Population $E^{(g+1)}$ bilden.
7. Stoppen, falls maximale Zahl an Generationen erreicht. Ansonsten mit Schritt 2 für Generation $g + 1$ fortfahren.

mente untereinander ist ohne Bedeutung. Die letzte wichtige Eigenschaft des Verfahrens ist die Nutzung einer Nischentechnik, die ohne ein vorher definiertes Distanzmaß auskommt.

Der Algorithmus ist in Tabelle 5.1 in sieben Schritten beschrieben. Zu beachten ist die zweifache Bewertung der Elemente in diesem Ansatz. Zum Ersten die Bewertung durch Funktionsauswertung, die die Position im Zielgrößenraum bestimmt. Zum Zweiten die Bewertung aufgrund der relativen Position der Elemente im Zielgrößenraum zueinander, aus der sich die Reproduktionswahrscheinlichkeit ableitet. Einige Punkte des Algorithmus bedürfen einer genaueren Spezifizierung, die im Folgenden dargestellt wird.

Für die Initialisierung der Population wird jeder Parameter der Individuen gemäß einer gleichverteilte Zufallszahl zufällig bestimmt, so dass eine gleichverteilte Population zu Beginn des Algorithmus vorliegt. Die Bewertung der Elemente im Zielgrößenraum ist in zwei Schritte unterteilt. Im ersten Schritt wird für die Elemente der externen Population die so genannte Stärke (strength) bestimmt. Die Stärke s_i eines Elementes $I'_i \in E'$ der externen Population

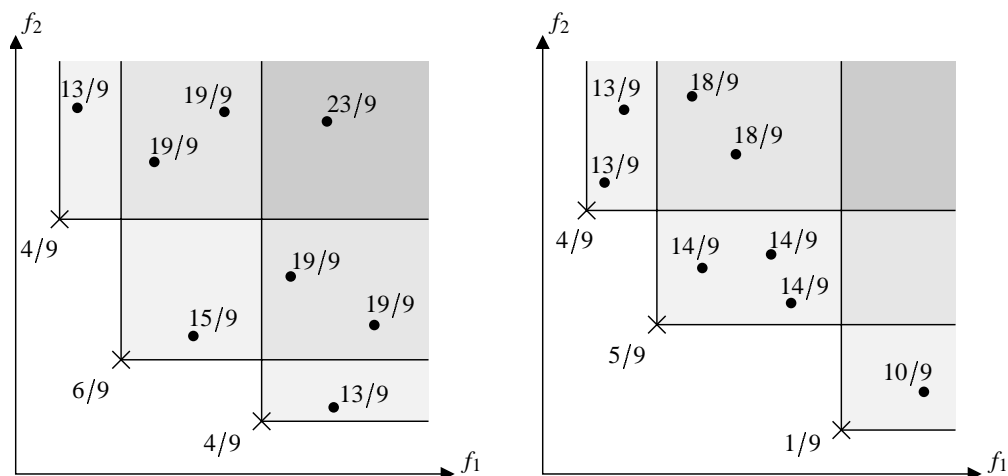


Abbildung 5.8: Zwei Beispiele für die Bewertung der Elemente im Rahmen des SPEA-Ansatzes. Zu der markierten Position der Elemente der internen (\bullet) und externen (\times) Population ist deren jeweilige Bewertung/Stärke notiert.

ergibt sich aus der Anzahl n der Elemente der internen Population, die von diesem Element I'_i dominiert werden. Sei N die Größe der internen Population, so ist $s_i = \frac{n}{N+1}$. Im zweiten Schritt wird aus den Stärken der externen Population die Bewertung der internen Population abgeleitet. Die Stärken sämtlicher Elemente, die ein Individuum $I_j \in E$ dominieren, werden aufaddiert. Zu dieser Summe wird der Wert Eins addiert ($1 + \sum_{i, i < j} s_i$), so dass die Bewertung bzw. Fitness eines Elementes der internen Population immer größer, d.h. schlechter als die eines Elementes der externen Population ist. Eine geringere Fitness führt zu einer höheren Reproduktionswahrscheinlichkeit. Für die nächste Generation werden zwei Elemente aus der Vereinigung der internen und externen Population zufällig ausgewählt. Dasjenige mit der geringeren Fitness wird durch Mutation variiert und in die Nachfolgepopulation übernommen (binary tournament).

Beispiele für Bewertungsszenarien sind in Abbildung 5.8 dargestellt, die Individuen in den hellgrauen Bereichen sind von einem Element der externen Population dominiert. Die in den mittelgrauen von zwei Elementen und die in den dunkelgrauen von Dreien. Es ist zu sehen, dass einsamere Elemente nahe der Pareto Menge eine geringere Bewertung, also höhere Reproduktionswahrscheinlichkeit aufweisen. Daraus folgt, dass spärlich besetzte Bereiche der Überdeckung durch diesen Mechanismus stärker gefördert werden.

Die Zahl der nicht dominierten Elemente wird für eine ewige Suche im Allgemeinen gegen unendlich streben. Für eine angemessene Beurteilung einer Überdeckung der Pareto Menge muss deren Größe jedoch auf eine endliche Anzahl begrenzt sein. Des Weiteren reduziert eine zu große Mächtigkeit der externen Population den Selektionsdruck und somit die Geschwindigkeit der Suche. Zusammen mit der Abhängigkeit des Nischenmechanismus von einer Gleichverteilung der nicht dominierten Elemente erklärt dies die Notwendigkeit, die Größe der externen Population zu begrenzen. Erreicht wird dies durch die Anwendung eines Clustering-Verfahrens zur Reduzierung der Menge. Da typischerweise eine geringe Reduzierung vorgenommen wird, d.h. von $N' + k$ auf N' mit $k_+ \ll N'$, kann ein einfacher Clustering-Algorithmus wie folgt angesetzt werden.

Voraussetzung für die Anwendung eines Clustering-Verfahrens ist die Existenz eines Abstandsmaßes für die Elemente. Da der SPEA die Nischenbildung im Zielgrößenraum berücksichtigen soll, um eine gute Überdeckung der Pareto Menge zu erreichen, ist auch das Abstandsmaß in diesem zu definieren. Da die einzelnen Kriterien der Optimierung reellwertige Größen sind, wird an dieser Stelle die Euklidische Norm verwendet.

$$\text{Abstand: } d(\vec{x}_1, \vec{x}_2) = \left\| \vec{f}(\vec{x}_1) - \vec{f}(\vec{x}_2) \right\|_2 \quad (5.5)$$

Die Zahl der Elemente ist mit $N' + k_+$, ($k_+ > 0$) gegeben. Dann werden die zwei Elemente \vec{x}_q und \vec{x}_p mit dem geringsten Abstand $d(\vec{x}_p, \vec{x}_q) = \min_{i,j} d(\vec{x}_i, \vec{x}_j)$ durch ein beliebiges dieser beiden ersetzt. Dieser Vorgang wiederholt sich, bis die Zahl der Elemente auf N' reduziert ist.

Der im Original mit Repräsentationen und Operatoren genetischer Algorithmen realisierte Ansatz ist in dieser Arbeit auf der Basis der Evolutionsstrategie mit einer reellwertigen Repräsentation umgesetzt worden. Die Verwendung der Selbstadaption wäre wünschenswert um einen Algorithmus zu erhalten, der vom Anwender keine algorithmusspezifischen Informationen erfordert. Im Rahmen der Arbeit hat sich jedoch die Aussage bestätigt, dass die Selbstadaption für die Vektoroptimierung nicht geeignet ist [LRS98].

5.4 Beschleunigung durch Parallelisierung

Zur Beschleunigung eines Optimierungsprozesses bietet sich der Weg der Parallelisierung an, sofern nicht mehrere Schritte des Prozesses abhängig voneinander sind, also nicht einer das Ergebnis des vorherigen erfordert. Diese Abhängigkeit ist zumeist bei klassischen Optimierungsverfahren gegeben, die auf einer einzelnen Lösung und deren schrittweiser Verbesserung

basieren, wie dies bei dem Gradientenverfahren der Fall ist. Ein populationsbasierter Ansatz wie die evolutionären Algorithmen hingegen besitzt eine inhärente Parallelisierungsoption. Die Zielfunktionsauswertungen innerhalb einer Generation sind voneinander unabhängig, so dass eine zeitgleiche Auswertung leicht realisierbar und gut skalierbar ist. Diese Parallelisierungsstrategie ist effizient, falls, wie im vorliegenden Anwendungsfall, die Zielfunktionsauswertung sehr zeitintensiv gegenüber den Operatoren des Optimierungsalgorithmus ist.

Die Skalierbarkeit eines Standard-EA ist jedoch beschränkt auf einen Parallelisierungsgrad, der klein gegenüber der Nachkommenspopulationsgröße ist. Andernfalls wird der Einfluss des Synchronisierungseffektes immer stärker, da für den Selektionsschritt alle parallelen Recheneinheiten zu synchronisieren sind, d.h. sämtliche Zielfunktionsauswertungen verfügbar sein müssen. Stellt die Populationsgröße nicht ein ganzzahliges Vielfaches der Zahl der Recheneinheiten dar, so bleibt die reale Skalierung hinter den theoretischen Werten zurück. Als Beispiel sei für eine Populationsgröße n die Zahl der Recheneinheiten $n-1$. Im ersten Schritt sind alle $n-1$ Einheiten aktiv, aber im zweiten lediglich eine, da keine weiteren Berechnungen zur gleichen Zeit nötig bzw. möglich sind. Damit beträgt die reale Beschleunigung $n/2$ gegenüber der theoretischen Grenze $n-1$.

Zwei weitere Aspekte, die die Verwendung von synchronen evolutionären Algorithmen, wie sie hier genannt werden, einschränkt, liegen in der Realisierung der Optimierung und in dem vorliegenden Anwendungsgebiet begründet. Erstens ist bei der Abbildung der Optimierung auf ein heterogenes Rechnernetz, wie es für diese Arbeit zur Verfügung stand, der unterschiedliche Zeitbedarf der einzelnen Recheneinheiten zur Zielfunktionsauswertung zu beachten. So kann sich sogar der Verzicht auf zusätzliche Rechenleistung im Hinblick auf die Gesamtperformance als sinnvoll erweisen (siehe Tabelle 5.2). Der zweite Punkt ergibt sich durch die Verwendung eines numerischen Simulators wie SPICE, dessen Konvergenzgeschwindigkeit von den ermittelten Startwerten und der jeweiligen Schaltungsdimensionierung abhängt und somit nicht konstant ist (Vergleiche Punkt 8 in Tabelle 3.1.) .

Die Alternative besteht im Einsatz asynchroner evolutionärer Algorithmen, die auf eine Generationsbildung verzichten. Der für die Eingrößenoptimierung verwendete Ansatz des steady-state EA [BBNH95] erlaubt die maximale Auslastung der verfügbaren Rechenkapazität, d.h. eine optimale Skalierbarkeit. Der steady-state EA basiert auf dem Master-Slave-Prinzip (Abbildung 5.9). Der Master-Prozess repräsentiert den evolutionären Algorithmus an sich, von dem aus in den Slave-Prozessen die Zielfunktionsauswertungen initiiert werden. Ist die Zielfunktionsauswertung in einem Prozess abgeschlossen, so wird deren Ergebnis dem Master gemeldet. Ist das neue Element besser als das schlechteste der μ Vorhandenen, so wird dieses

ersetzt, andernfalls wird das neue Element verworfen. Aus der momentanen Population wird ein weiteres Element generiert, das mit der freien Recheneinheit ausgewertet wird.

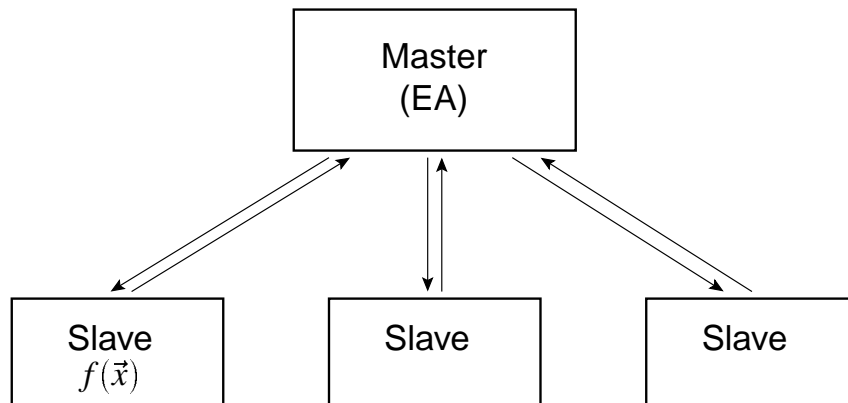


Abbildung 5.9: Struktur des steady-state evolutionären Algorithmus. In den Slave-Prozessen erfolgen die Zielfunktionsberechnungen, deren Ergebnisse für die Optimierung im Master-Prozess verwendet werden.

Der steady-state EA entspricht einer Plus-Strategie und wird in der Literatur mit einer $\mu + 1$ -Strategie gleichgesetzt, für die die Selbstadaption der Optimierung als nicht geeignet eingestuft wird [HB98]. Die Annahme einer $\mu + 1$ -Strategie ist insofern korrekt, als dass jeweils ein einzelnes Element in die Population integriert wird. Allerdings wird hierbei die zusätzliche Information in den $n - 1$ weiteren Recheneinheiten unterschlagen, so dass der steady-state EA als eigenständige Form zu betrachten ist. Für den Bereich des Designs integrierter Schaltungen wird hier die Selbstadaption erfolgreich eingesetzt (vergleiche Kapitel 3.3.3).

Der Einfluss einer asynchronen Variante auf die Konvergenzgeschwindigkeit kann in dieser Arbeit, aufgrund der erforderlichen theoretischen Betrachtungen, nicht beantwortet werden. Es ist aber nicht auszuschließen, dass sich die reale Skalierbarkeit des Algorithmus vermindert. Auf der anderen Seite ist für den Einsatz in einer nicht hundertprozentig verlässlichen Rechnernetz Umgebung zu beachten, dass der Ausfall eines Rechners bzw. Slave-Prozesses den gesamten Optimierungsvorgang stoppt. Dem gegenüber kann ein asynchroner Algorithmus weiter fortgesetzt werden. Eine solche verlässliche Umgebung kann in einem üblichen heterogenen Netz nicht vorausgesetzt werden, so dass dieser Aspekt bei Simulationszeiten von mehreren Tagen von nicht unerheblicher Relevanz ist [SM95].

5.4.1 Asynchrones Räuber-Beute Modell

Das Räuber-Beute Modell ist durch das Zusammenspiel einer aktiven und einer passiven Seite gekennzeichnet. Der passive Part ist durch die strukturierte Population von Individuen bestimmt, während der aktive durch die Menge von Räubern beschrieben ist, die ihre Position auf dieser Struktur und deren Elemente ändern. Die Betrachtung der Räuber als voneinander unabhängige Akteure bietet eine asynchrone Parallelisierung an, wie sie im steady-state EA vorzufinden ist. Die Aktionen der Räuber erfolgen nicht quasi gleichzeitig, sondern werden stattdessen sukzessive durchgeführt. Der Algorithmus lässt sich somit optimal parallelisieren, solange die Zahl der Recheneinheiten die Zahl der Räuber nicht übersteigt.

Ist die Zahl der parallelen Prozesse sehr klein gegenüber der Populationsgröße, verhindert die asynchrone Variante einen nachteiligen Aspekt des originalen Räuber-Beute Modells. Für die in der Literatur vorzufindende Konstellation von 900 Individuen und 200 Räubern ist die Wahrscheinlichkeit, dass sämtliche Räuber an verschiedenen Positionen platziert sind, nahezu gleich Null ($< 10^{-10}$). Bei einem synchronen Ansatz existieren somit immer Individuen, die mehrfach mutiert sind und deren Fitness neu berechnet wird. Von diesen kann jedoch nur eine Realisierung in die nächste Generation übernommen werden. Welche dies ist, wird allein durch die Implementierung des Algorithmus bestimmt. Die Tatsache, dass Ergebnisse einer zeitaufwendigen Zielfunktionsauswertung, wie sie in dieser Arbeit vorliegen, nicht berücksichtigt werden, widerspricht einer effizienten Nutzung der Rechenleistung. Für eine typischerweise vorliegende, kleine Zahl von Recheneinheiten gegenüber der Populationsgröße kann bei dem asynchronen Räuber-Beute Modell der Fall einer Mehrfachberechnung hingegen vernachlässigt werden. Alternativ sind für beide Varianten des Algorithmus Maßnahmen umsetzbar, die eine mehrfache Zielfunktionsauswertung grundsätzlich verhindern. Diese werden jedoch wiederum in das Verhalten des Algorithmus eingreifen. Ergebnisse bezüglich der Anwendung des asynchronen Räuber-Beute Modells sind in Kapitel 5.5 dargestellt.

5.4.2 Asynchrone Variante des Strength Pareto Evolutionary Algorithm (asyncSPEA)

Die im Rahmen dieser Arbeit entwickelte, asynchrone Variante des SPEA folgt dem Prinzip des steady-state Algorithmus und setzt dieses für die Mehrgrößenoptimierung um. Die Generationsbildung wird hierbei zugunsten einer kontinuierlichen Population ersetzt, in die stetig neue Individuen integriert werden. Die interne Population wird hierzu in einer Queue

oder einem FIFO (first-in-first-out) gespeichert. Meldet ein Slave-Prozess das Ergebnis einer Zielfunktionsberechnung, so wird das neue Element in die Population aufgenommen und das erste (älteste) entfernt. Es existiert somit eine zeitliche Ordnung der Elemente. Die folgenden Schritte werden wie im synchronen SPEA durchgeführt, jedoch dadurch vereinfacht, dass nur ein einziges, neues Element hinzukommt. Falls das neue Element nicht durch Weitere dominiert ist, wird es in die externe Population übernommen und diejenigen Elemente, die von dem neuen dominiert werden, werden aus der externen Population entfernt. Das anschließende Clustering beschränkt sich auf die Elimination eines einzelnen Elementes, von $N'+1$ auf N' . Zur Generierung eines neuen Elementes wird Eines aus der Vereinigung beider Populationen selektiert (ebenfalls binary tournament, vergleiche Seite 75) und variiert. Dieses wird anschließend dem freien Slave-Prozess zur Berechnung übertragen.

Die Initialisierung des asyncSPEA erfolgt analog zu den Schritten 1 bis 5 in Tabelle 5.1. An dieser Stelle findet sich die einzige Synchronisierung der Prozesse, da eine konsistente Populationsstruktur zum Start des Algorithmus vorliegen muss. Anschließend ist für jeden Slave-Prozess ein Element zu generieren, dessen Berechnung dann gestartet wird.

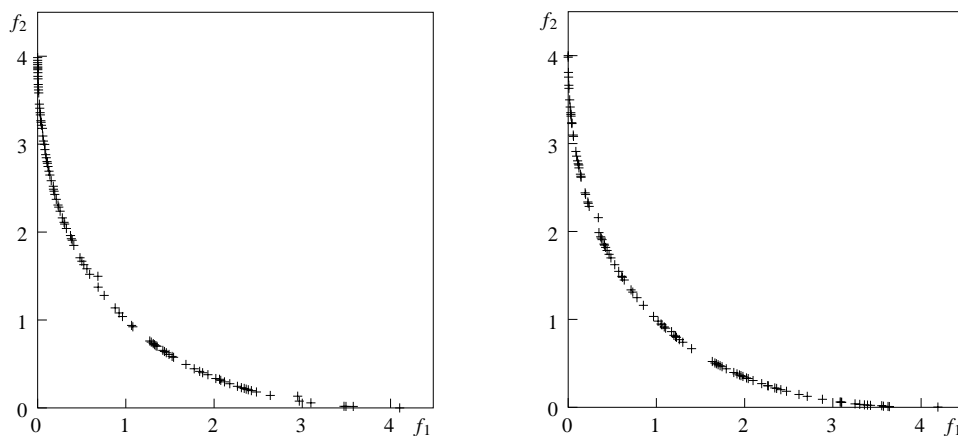


Abbildung 5.10: Die Gegenüberstellung der Ergebnisse des originalen SPEA (links) und des asyncSPEA (rechts) auf dem Testproblem von Schaffer nach 1000 Zielfunktionsauswertungen zeigt die Gleichwertigkeit der Lösungen beider Verfahren.

Zur Überprüfung der Gleichwertigkeit der hier vorgestellten asynchronen Variante zu dem originalen SPEA sollen beide Verfahren anhand des Testproblems nach Schaffer [Sch85] verglichen werden. Es sind zwei Zielfunktionen f_1 und f_2 zu minimieren, die wie folgt bestimmt sind: $f_1(\vec{x}) = x_1^2 + x_2^2$ und $f_2(\vec{x}) = (x_1 - 2)^2 + x_2^2$. Die Pareto Menge lässt sich analy-

	SPEA	asyncSPEA
sieben Rechner	2,8	3,2
alle Rechner	2,6	4,3

Tabelle 5.2: Rechenleistung im Rechnernetz des Lehrstuhls für Bauelemente der Elektrotechnik. Die Angaben bezeichnen die mittlere Anzahl der durchgeführten Simulationen je Sekunde am Beispiel des Lee-Gatters.

tisch zu $\left\{ \left(\frac{x^2}{(x-2)^2} \right) \mid x \in [0; 1] \right\}$ bestimmen und wird durch beide Verfahren gut approximiert und charakterisiert. Die Ergebnisse sind in Abbildung 5.10 dargestellt. Ein visueller Vergleich der gefundenen Überdeckung der Pareto Menge zeigt, dass das asyncSPEA zu dem SPEA gleichwertig ist.

Der zeitliche Vorteil bei der Untersuchung integrierter Schaltungen durch die Verwendung dieses asynchronen Algorithmus lässt sich durch eine direkte Gegenüberstellung des asyncSPEA zu dem SPEA abschätzen. Die Geschwindigkeit der Verfahren sei hierzu über die Zahl der durchgeführten SPICE-Simulationen je Zeiteinheit definiert. Die im Rechnernetz des Lehrstuhls für Bauelemente der Elektrotechnik ermittelten Ergebnisse sind in Tabelle 5.2 aufgelistet. In den ersten Optimierungsläufen ist eine Gruppe von sieben Rechnern vom Typ SUN Ultra mit untereinander vergleichbarer Leistungsfähigkeit eingesetzt worden. Für die weiteren Ergebnisse ist die gesamte verfügbare Rechenleistung einbezogen worden, die auch Recheneinheiten stark unterschiedlicher Leistungsfähigkeit einschließt.

Bei der Optimierung des Lee-Gatters erzielt das asyncSPEA, unter Verwendung sieben gleichwertiger Rechner, gegenüber dem SPEA eine Verbesserung um 14 %. Durch Hinzufügen zusätzlicher Rechenleistung zur Beschleunigung der Verfahren stellt sich für das asyncSPEA ein Gewinn von 34 % ein. Für das synchrone SPEA hingegen ist eine Verminderung um 7 % festzustellen. Das bedeutet somit, dass die zusätzlichen Recheneinheiten für das SPEA nicht nur unnützlich sind, sondern durch den Synchronisierungseffekt die Leistung des Gesamtsystems negativ beeinflussen. Insgesamt kann durch den asyncSPEA eine Beschleunigung auf 154 % der optimalen Geschwindigkeit des SPEA erreicht werden, ohne für den asyncSPEA die Struktur des Rechnernetzes beachten zu müssen.

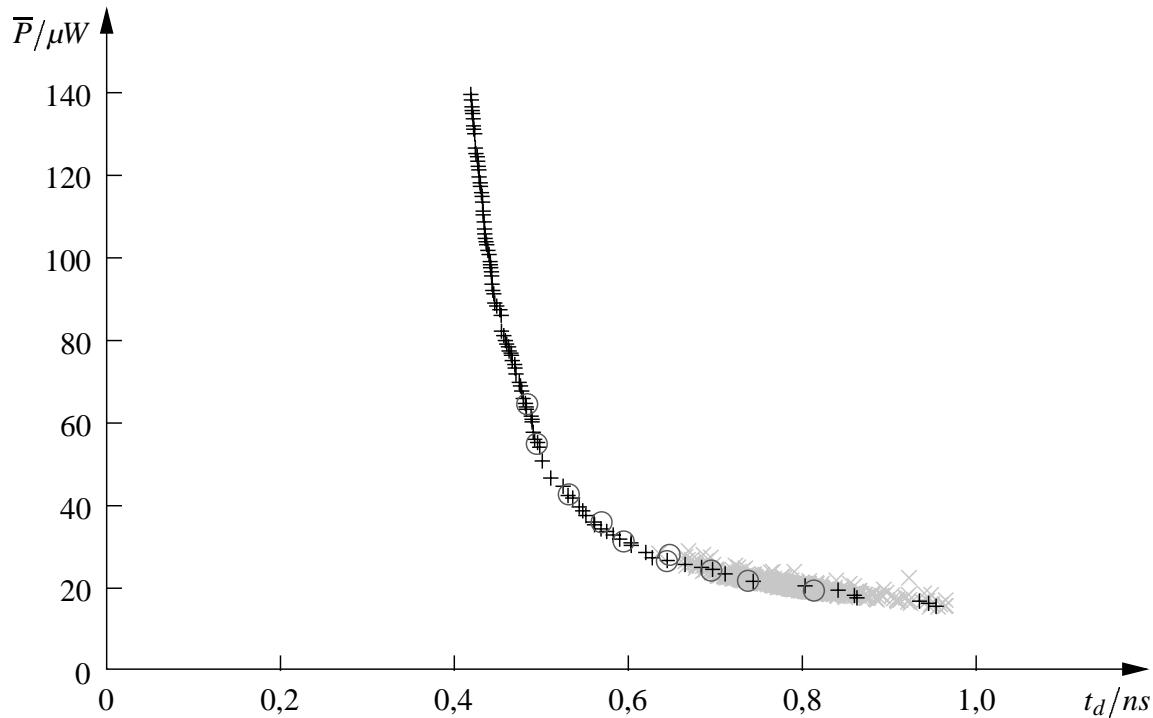


Abbildung 5.11: Gegenüberstellung der verschiedenen Lösungsmengen für den CMOS Inverter. Räuber-Beute Modell \times , asyncSPEA $+$ und Gewichtsmethode \circ .

5.5 Vergleich der Verfahren

Für eine geeignete Wahl des Optimierungsverfahrens ist eine Gegenüberstellung der erzielbaren Resultate bezogen auf den relevanten Anwendungsbereich vorzunehmen. Hierzu werden das asynchrone Räuber-Beute Modell und der asyncSPEA Ansatz zur Optimierung der Verlustleistung und der Verzögerungszeit eines CMOS Inverters angewendet. Beide Verfahren werden auf 10.000 Simulationen bzw. Zielfunktionsauswertungen begrenzt. Als dritte Alternative werden 10 Optimierungsläufe à 1.000 Simulationen nach der Gewichtsmethode durchgeführt. Die Ergebnisse der drei Ansätze sind in Abbildung 5.11 dargestellt.

Bereits in diesem einfachen Beispiel der Vektoroptimierung zeigt sich, dass die Gewichtsmethode nur mit ausreichendem Vorwissen eine verwertbare Lösung der gesamten Optimierungsaufgabe liefert. Der Zahlenwert der Verlustleistung liegt bei 10^{-5} und der der Verzögerungszeit bei 10^{-10} . Werden 10 oder auch 100 Optimierungen mit Gewichtungen, die von $(1, 0)$ bis $(0, 1)$ gleichverteilt sind, durchgeführt, so ergeben sich jeweils nahezu identische

Lösungen. Die Gewichtungen $(1, 0)$ und $(0, 1)$ seien hierbei ausgenommen. Die einzelnen Lösungen in Abbildung 5.11 sind für äquidistant verteilte Gewichtungen von $(10^{-5}, 10^{-9})$ bis $(10^{-4}, 10^{-10})$ bestimmt worden.

Ein visueller Vergleich der ermittelten Lösungsmengen zeigt die gleichmäßigere Approximation der Pareto Menge durch die Vektoroptimierungsverfahren gegenüber der Gewichtsmethode mit einer einfachen Evolutionsstrategie. Das Räuber-Beute Modell erreicht eine gute Näherung der Pareto Menge, weist jedoch Schwächen in der Diversität der resultierenden Lösungsmenge. Das Ergebnis ist somit zur Bestimmung der Charakteristik der Pareto Menge nur bedingt geeignet. Dem gegenüber liefert das asyncSPEA ein Ergebnis, das sowohl bezüglich der Approximation als auch in der Diversität der Überdeckung geeignet ist, die Charakteristik der Pareto Menge visuell zu erfassen. Aufgrund dieser Eigenschaften wird das asyncSPEA für die weiteren Betrachtungen in dieser Arbeit eingesetzt.

Kapitel 6

Verhaltensanalyse integrierter Schaltungen

Der Entwurf integrierter Schaltungen bietet mit der Vielzahl freier Parameter eine große Variationsbandbreite. Um diese für den Anwender auf ein relevantes Designspektrum zu reduzieren, ist es erforderlich, prinzipielle Zusammenhänge zwischen den Parametern einer Schaltung zu erkennen und zu nutzen. Parameter können in diesem Zusammenhang sowohl direkt steuerbare Designparameter der Schaltung, beispielsweise die Versorgungsspannung, als auch Ergebnisgrößen wie die Verlustleistung der Schaltung sein. Von Interesse hierbei sind nicht Variationen einzelner Parameter mit ansonsten fixen Bedingungen, sondern deren Variation unter gleichzeitiger Ausnutzung des gesamten verbleibenden Optimierungspotenzials.

Erkenntnisse über Abhängigkeiten der Parameter können sich aus dem physikalischen Hintergrund ableiten. Zum Beispiel die Festlegung auf minimale Längen der Transistoren in digitaler CMOS Technik für Logikschaltungen, da das W/L -Verhältnis indirekt die Schaltgeschwindigkeit bestimmt und eine Vergrößerung der Länge nur zu höherer Verlustleistung führt. Des Weiteren ist ein Erkenntnisgewinn durch die bloße Beobachtung des Verhaltens der Schaltung und Bestimmung von Regelmäßigkeiten möglich. Diese Erkenntnis wird sich in der Regel nur auf eine Teilmenge des Parameterraumes beziehen, so dass sie nur bedingt vertrauenswürdig ist.

Das hier vorgestellte Verfahren erlaubt eine rechnergestützte Bestimmung von Parameterabhängigkeiten auf der Basis des Verhaltens der betrachteten Schaltung. Eine gestellte Hypothese wird auf ein Optimierungsproblem mehrerer Zielgrößen zurückgeführt, dessen Lösung

die Beurteilung der Beziehungen zwischen den beteiligten Parametern erlaubt. Durch die Verwendung globaler Suchalgorithmen zur Optimierung ist dieser Ansatz geeignet, Hypothesen über Regelmäßigkeiten zu validieren und das Vertrauen in dieses Ergebnis zu stärken.

6.1 Bestimmung von Parameterabhängigkeiten

Die hier erarbeitete Methode zur Bestimmung von Parameterabhängigkeiten wird an dieser Stelle anhand eines Beispiels zur Veranschaulichung der einzelnen Schritte dargestellt. Es wird der Zusammenhang zwischen der minimal möglichen Verlustleistung und der minimalen Verzögerungszeit eines Inverters betrachtet. In diesem Kontext ist unter minimal die Berücksichtigung eines jeweils optimierten Designs zu verstehen. Diesen Punkt vorausgesetzt, geht eine Verringerung der Verzögerungszeit einer Schaltung mit einer Steigerung der Verlustleistung einher. Wie in Gleichung (2.3) deutlich wird, ist in CMOS Schaltungen die Verlustleistung \overline{P} proportional zur effektiven Kapazität C_{eff} , die bei kompakten Grundschaltungen mit kurzen Leitungsverbindungen im Wesentlichen von der Gatekapazität bestimmt wird. Letztere ist bei einem Schaltvorgang umzuladen, so dass sich zusammen $\overline{P} \sim \frac{1}{t_d}$ ergibt. Für diese beiden Größen ist somit eine Abhängigkeit gemäß einer Hyperbel zu erwarten. Zur Überprüfung dieses Zusammenhangs wird ein Problem der Vektroptimierung bezüglich der Verlustleistung und der Verzögerungszeit formuliert. Beide Größen sind hierbei zu minimieren. Die Lösung dieses Optimierungsproblems, d.h. eine Überdeckung der Pareto Menge, ist in Abbildung 6.1 dargestellt. Der Zusammenhang zwischen beiden Größen lässt sich visuell als Hyperbel identifizieren und die Annahme $\overline{P} \sim \frac{1}{t_d}$ bestätigen.

In diesem Beispiel sind zwei Größen, die keine direkt kontrollierbaren Parameter des Inverters sind, einander gegenübergestellt. Ebenso kann jedoch auch ein direkter Parameter, wie z.B. die Transistorweite, berücksichtigt werden, so dass unter anderem die Abhängigkeit der Verlustleistung von den Designparametern untersucht werden kann.

Die erfolgreiche Anwendung dieses Ansatzes setzt die Betrachtung einer bestimmten Form der Abhängigkeit voraus. Das eingesetzte Optimierungsverfahren ist hier eingeschränkt auf die Minimierung jeder einzelnen Komponente. Dessen Verwendung erlaubt daher die Betrachtung streng monoton fallender Parameterzusammenhänge. Ein Blick auf die Definition 6 zeigt, dass alle Punkte eines streng monoton fallenden Zusammenhangs zueinander nicht dominant im Sinne der Definition 2 (Seite 65) sind. Damit ist der Parameterzusammenhang in diesem Fall identisch mit der Pareto Menge.

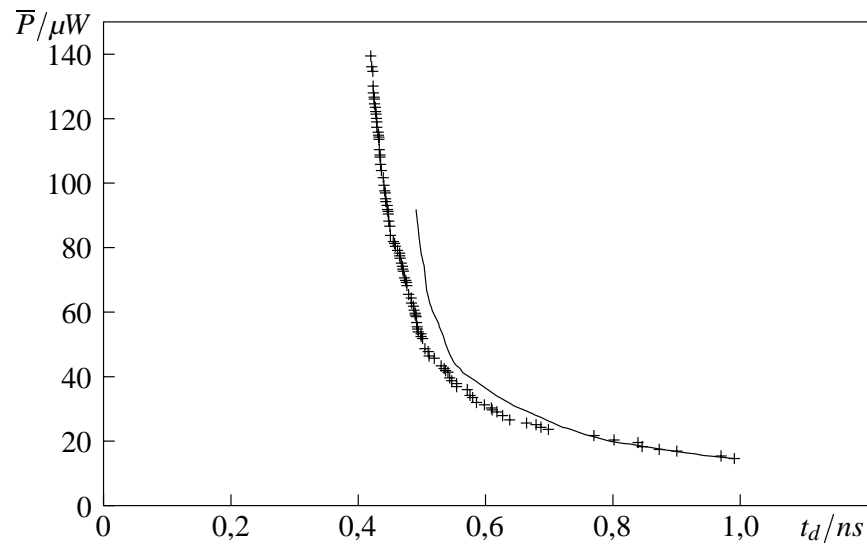


Abbildung 6.1: Verlustleistungs-Verzögerungszeit-Zusammenhang eines Inverters. Die Lösung der Optimierung und alternativ durch Spannungsvariation (Linie) bestimmter Zusammenhang.

Definition 6 (streng monoton fallend)

Eine Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}$ heißt streng monoton fallend, falls für jedes $\vec{x}' \neq \vec{x}$ mit $x'_i \geq x_i, \forall i$ gilt $f(\vec{x}') < f(\vec{x})$.

Beliebige streng monotone Parameterabhängigkeiten lassen sich durch eine lineare Abbildung in eine streng monoton fallende Funktion umformen und sind somit ebenfalls handhabbar.

Liegt die Überdeckung einer zusammenhängenden ein- bzw. zweidimensionalen Menge als Lösung des Optimierungsproblems vor, so lässt sich für den untersuchten Parameterzusammenhang Folgendes schließen. Die Grundhypothese eines streng monoton fallenden Zusammenhangs ist bestätigt. Allerdings ist diese Aussage auf den von der Menge abgedeckten Wertebereich beschränkt. Ist dieser kleiner als der Definitionsbereich eines Parameters, kann keine allgemeine Schlussfolgerung gezogen werden. Liegt hingegen eine einelementige Lösung des Optimierungsproblems vor, leitet sich daraus ein Widerspruch zur gestellten Hypothese ab und es kann keine weitere Aussage zur Art der Abhängigkeit der Parameter getroffen werden.

Zur Darstellung der Diskrepanz bei der Bestimmung der Parameterabhängigkeit mittels einer einfachen Variation einzelner Parameter einerseits und unter Ausnutzung des Optimierungspotenzials andererseits sei nochmals ein CMOS Inverter betrachtet. In Abbildung 6.1 ist der

Zusammenhang zwischen Verlustleistung und Verzögerungszeit dargestellt. Die Lösung der Optimierung ist durch die einzelnen Punkte gekennzeichnet und der gefundene Zusammenhang, manuell angepasst, ist durch eine Kurve beschrieben. Die dominierende Größe in diesem Zusammenhang ist die Versorgungsspannung, so dass es sich anbietet, die Abhängigkeit von \bar{P} und t_d durch eine Spannungsvariation zu skizzieren. Hierzu wird die verlustleistungsärmste und langsamste Lösung obiger Optimierung verwendet, die typischerweise mit einer geringen Versorgungsspannung betrieben wird. Für diese wird die Spannung bis zum Maximum hochgefahren, so dass sich die gestrichelte Kurve ergibt. Bereits an diesem einfachen Beispiel lässt sich die Diskrepanz zwischen dieser vereinfachenden Näherung und einer vollständigen Optimierung nachvollziehen.

Zur umfassenderen Nutzung dieses Ansatzes zur Bestimmung von Abhängigkeiten zwischen Parametern einer Schaltung sind zwei Erweiterungen des Verfahrens erforderlich. Dies ist zum Ersten die Berücksichtigung nicht monotoner Zusammenhänge und zum Zweiten die automatische Extraktion eines funktionalen Zusammenhangs aus den ermittelten Werten, um die Parameterabhängigkeit analytisch beschreiben zu können.

6.2 Handhabung nicht monotoner Zusammenhänge

Die Bestimmung monotoner Zusammenhänge zwischen verschiedenen Parametern kann, wie zuvor beschrieben, durch eine Formulierung als Optimierungsproblem erfolgen. Liegt ein nicht monotoner Parameterzusammenhang vor, so ist dieser nicht ohne weiteres zu erfassen. Bestenfalls ließe sich ein streng monoton fallender Abschnitt der Abhängigkeit ermitteln, falls dieser nicht von einem weiteren Punkt dominiert wird. In Abbildung 6.2 sind für zwei angenommene quadratische Abhängigkeiten die resultierenden Pareto Mengen markiert. Im ersten Fall ist ein Segment des Zusammenhangs bestimmbar, während im zweiten eine einzelne Lösung alle weiteren Punkte dominiert.

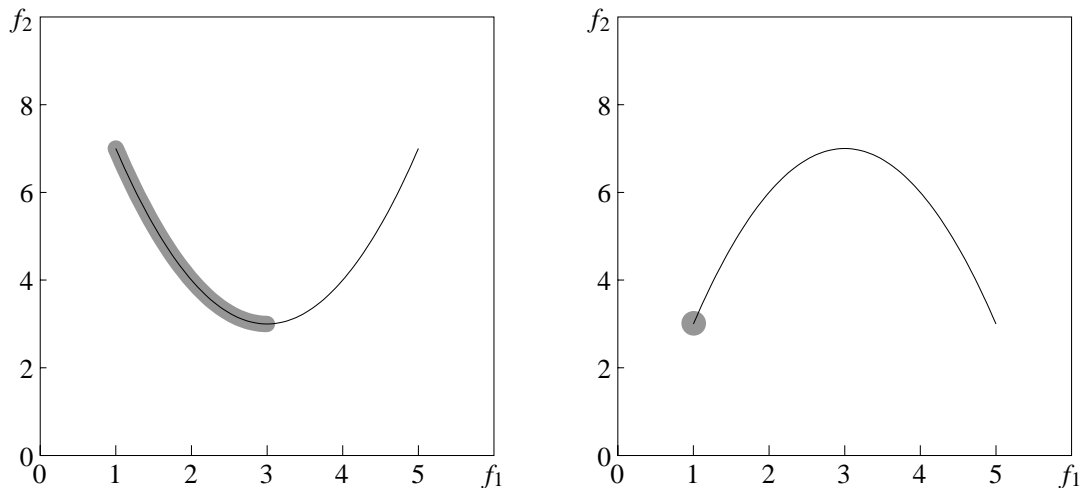


Abbildung 6.2: Zwei nicht monotone Parameterzusammenhänge. Die Pareto Menge ist links als Linie und rechts als Punkt markiert.

Zur Bestimmung eines nicht monotonen Zusammenhangs mit Hilfe der Vektoroptimierung ist folglich die Verwendung einer zusätzlichen Abbildung erforderlich. Diese muss die vorliegende Parameterabhängigkeit in einen streng monoton fallenden Zusammenhang überführen, der dann wiederum bestimmt werden kann.

Für die beiden Beispiele in Abbildung 6.2 mit einem quadratischen Zusammenhang $f_2(f_1)$ kann die Zielgröße f_2 durch eine neue Funktion $f'_2 = f_2/f_1^3$ ersetzt werden. Der resultierende Zusammenhang $f'_2(f_1)$ ist dann streng monoton fallend (d.h. $\frac{\partial f'_2}{\partial f_1} < 0$) und kann durch eine Vektoroptimierung bestimmt werden. Die Relation $f'_2(f_1)$ ist für beide Beispiele in Abbildung 6.3 dargestellt.

An dieser Stelle sind die einzelnen Zahlenräume des Verfahrens genauer zu differenzieren (Abbildung 6.4). Zur Konkretisierung des Designs der integrierten Schaltung sind die freien Designparameter festzulegen. Abhängig von diesen, ergeben sich durch die Simulation mit SPICE die Ergebnisgrößen der Schaltung. Für eine Untersuchung von Parameterabhängigkeiten sind geeignete Zielgrößen sowohl aus den Designparametern als auch aus den Ergebnisgrößen, für die Vektoroptimierung zu wählen. Diese Zielgrößen haben typischerweise eine direkte physikalische Bedeutung. Um eine Optimierung sinnvoll durchführen zu können, werden diese Zielgrößen auf synthetische Zielgrößen abgebildet, die dann die Eigenschaft eines monoton fallenden Zusammenhangs erfüllen.

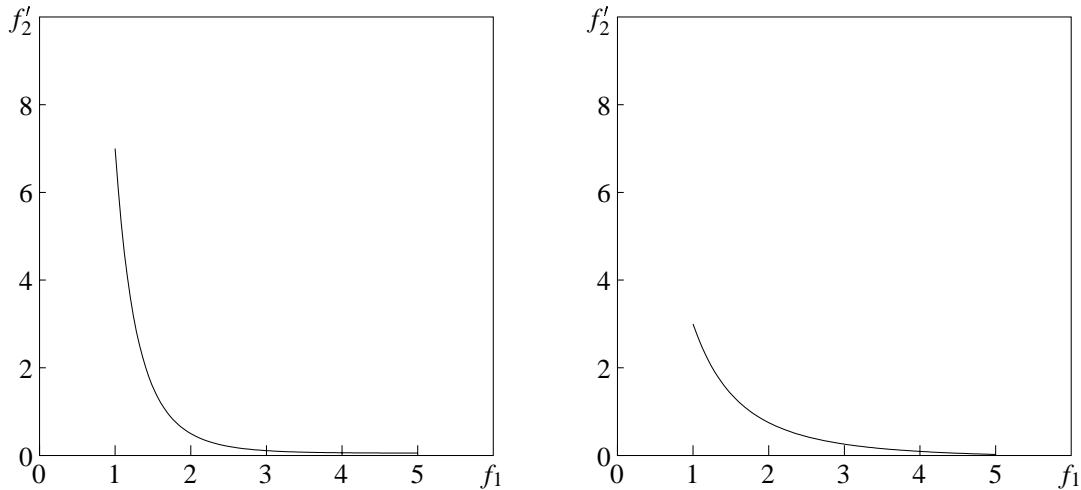


Abbildung 6.3: Die umgeformten Parameterzusammenhänge aus Abbildung 6.2, beide mit $f'_2 = \frac{f_2}{f_1}$.

Die Anwendung einer zusätzlichen Abbildung auf die physikalischen Zielgrößen erlaubt die Untersuchung beliebiger stetiger Parameterzusammenhänge. Da diese Abbildung einen eventuell positiven Gradienten eines vorliegenden Zusammenhanges in eine negative Steigung transformieren muss, ist die Stetigkeit eine hinreichende Bedingung. Eine Unstetigkeitsstelle mit einem unendlichen, positiven Gradienten kann nur noch in Sonderfällen durch komplexere Umformungen in einen streng monoton fallenden Zusammenhang umgewandelt werden.

Als Beispiel für die Untersuchung einer nicht monotonen Abhängigkeit wird die Beziehung zwischen Verlustleistung und der Versorgungsspannung an einem CMOS Inverter betrachtet. Für größere Spannungen, bei denen ausschließlich die dynamische Verlustleistung eine Rolle spielt, ist entsprechend Gleichung (2.3) ein in etwa quadratischer Zusammenhang zu erwarten. Für kleine Spannungen gegen 1 V werden sich die Einflüsse durch die Begrenzung des Designraumes, d.h. die minimalen Designmaße, bemerkbar machen. Zur Sicherstellung eines streng monotonen Zusammenhangs wird die Verlustleistung \bar{P} in eine neue Größe \bar{P}' mit $\bar{P}' = \frac{\bar{P}}{V_{DD}^4}$ überführt. Die Vektoroptimierung erfolgt somit bezüglich \bar{P}' und V_{DD} unter Beeinflussung der p-Kanal- und n-Kanal-Weiten und der Versorgungsspannung. In diesem Beispiel finden sich sowohl ein Designparameter als auch eine Ergebnisgröße der Simulationen als Zielgröße wieder. Das Resultat der Optimierung ist in Abbildung 6.5 präsentiert.

Die Übertragung aus dem synthetischen in den physikalischen Zielgrößenraum ist in Abbildung 6.6 gegeben. Hier ist ein nicht monotoner Zusammenhang zu identifizieren, bei dem sich

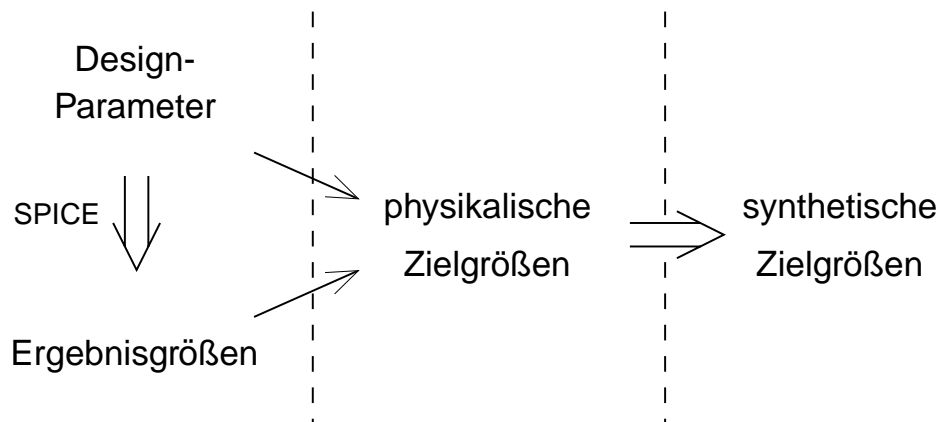


Abbildung 6.4: Zusammenspiel der verschiedenen Zahlenräume bei der Bestimmung von Parameterabhängigkeiten

die minimale Verlustleistung bei einer Versorgungsspannung von ca. $1,2\text{ V}$ einstellt. Sowohl für größere als auch für kleinere Spannungen nimmt die Verlustleistung zu. Bei kleineren Spannungen ist dies aus dem höheren Schaltstrom zur Einhaltung der maximalen Verzögerungszeit zu begründen. Für größere Spannungen folgt der Kurvenverlauf dem erwarteten quadratischen Zusammenhang.

6.3 Extraktion analytischer Zusammenhänge

Die Lösung eines mehrkriteriellen Optimierungsproblems mit einem populationsbasierten Verfahren ist eine Punktmenge, die als Überdeckung eine Approximation der Pareto Menge darstellt. Der Zusammenhang zwischen den Zielgrößen stellt sich durch diese experimentellen Daten dar, ist jedoch in dieser Form nur visuell zu erfassen und zu interpretieren. Für die Bewertung einer Parameterabhängigkeit und einen eventuellen Vergleich ist jedoch die Form eines analytischen Zusammenhangs besser geeignet.

Den zugrunde liegenden analytischen Zusammenhang aus der Überdeckung der Pareto Menge zu extrahieren, führt zu einem weiteren Optimierungsproblem, das hier wiederum mit evolutionären Algorithmen gelöst wird. Diese Extraktion ist auf die Bestimmung der Parameter für

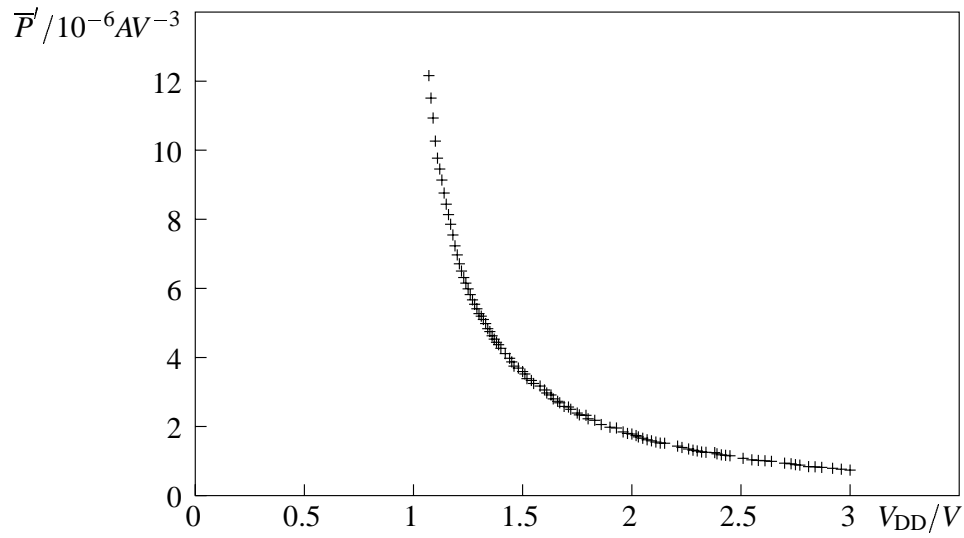


Abbildung 6.5: Der synthetische Zielgrößenraum des CMOS Inverters mit der Versorgungsspannung V_{DD} und der modifizierten Verlustleistung \bar{P}' .

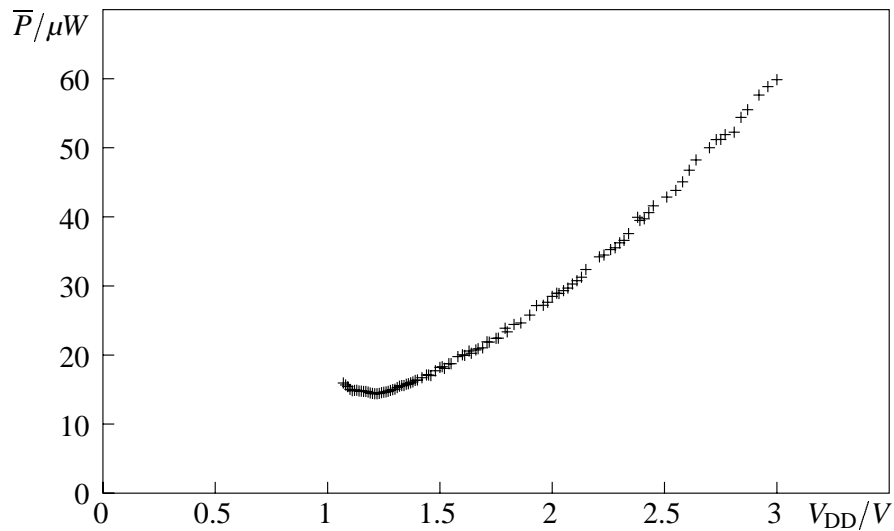


Abbildung 6.6: Der physikalische Zielgrößenraum des CMOS Inverters mit der Versorgungsspannung V_{DD} und der Verlustleistung \bar{P} .

einen gegebenen, analytischen Zusammenhang beschränkt. Das ebenfalls mögliche Evolvieren der funktionalen Struktur ist hierbei nicht sinnvoll, da sich in der Regel die Art des Zusammenhangs aus den physikalischen Gegebenheiten ableiten lassen sollte. Nur so kann ein allgemeingültiges und in den Aussagen eventuell extrapolierbares Ergebnis gewonnen werden.

Eine Überdeckung $U = \{(u_1, \dots, u_n)\}$ ist eine einseitige Approximation der Pareto-Kurve oder -Fläche. Für diese Pareto Menge gilt gemäß der Definition 3, dass es keinen Punkt gibt, der von einem Element der Überdeckung dominiert würde. Unter der Annahme, dass es einen funktionalen Zusammenhang $f_n = g(f_1, \dots, f_{n-1})$ zwischen den einzelnen Zielgrößen f_i gibt, folgt für die Elemente der Überdeckung: $u_n > g(u_1, \dots, u_{n-1})$. Gleiches muss für die Abbildung $h(\cdot)$ gelten, die den vorliegenden Zusammenhang beschreiben und somit als Schätzer für $g(\cdot)$ dienen soll (Abbildung 6.7). Die Ungleichungsbeziehung der Funktion $h(\cdot)$ zur Überdeckung ist als Kriterium hinreichend, falls $h(\cdot)$ monoton ist. Unter Berücksichtigung nicht monotoner Zusammenhänge gelten diese Aussagen im synthetischen Zielfunktionsraum. Übertragen in den Raum physikalischer Zielgrößen bleibt das Ungleichungsverhältnis zwischen Parameterzusammenhang und der Überdeckung erhalten.

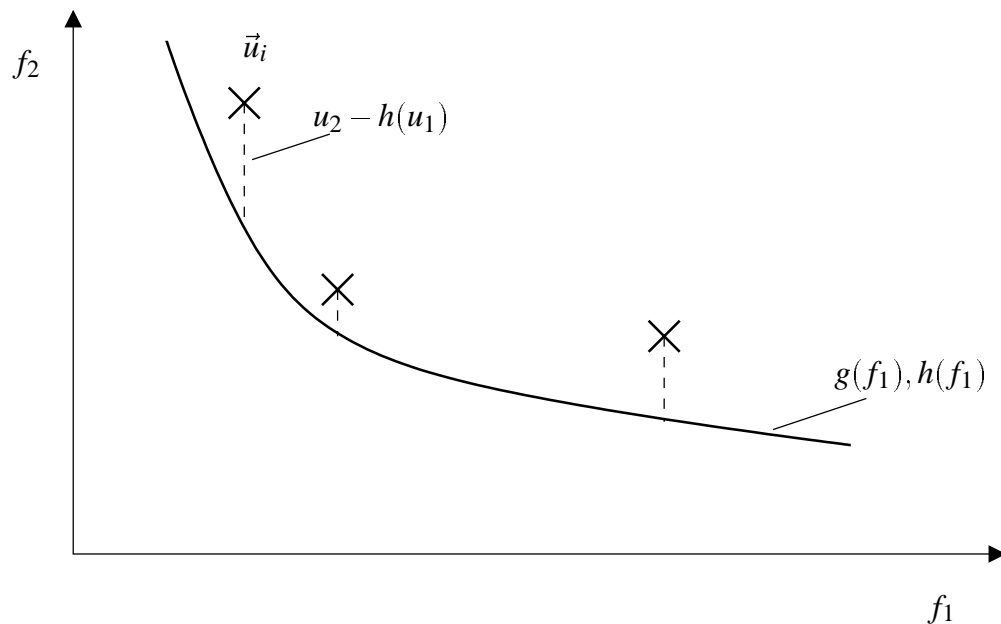


Abbildung 6.7: Darstellung einer Überdeckung und der Pareto Menge $g(\cdot)$ bzw. deren Schätzfunktion $h(\cdot)$. Der Abstand beider ist gestrichelt gekennzeichnet.

Es ist somit eine Funktion $h(\cdot)$ zu bestimmen, die für jeden Punkt \vec{u} der Überdeckung die Bedingung $u_n > h(u_1, \dots, u_{n-1})$ erfüllt und übertragen in den synthetischen Zielgrößenraum einen monoton fallenden Zusammenhang darstellt. Der Abstand von $h(\cdot)$ zur Überdeckung ist bei dieser Bestimmung zu minimieren. Hierzu sei der Abstand über die quadratische Fehlersumme nach Definition 7 gegeben. Da jede zulässige Funktion zur Approximation des Parameterzusammenhangs die Bedingung $u_n > h(u_1, \dots, u_{n-1})$ erfüllt, ist der definierte Abstand d_h immer größer Null.

Definition 7 (Abstandsmaß)

Der Abstand einer Funktion $h(\cdot)$ zu einer gegebenen Überdeckung $U = \{\vec{u}\}$ ist gegeben durch:

$$d_h = \sum_U u_n - h(u_1, \dots, u_{n-1}) \quad (6.1)$$

Die geeignete Wahl der Funktionsstruktur für die Abbildung $h(\cdot)$, z.B. als exponentieller Zusammenhang ($h(x) = a \cdot \exp(bx) + c$) oder als Hyperbel ($h(x) = a/x + c$) garantiert die Einhaltung der Monotoniebedingung im synthetischen Zielgrößenraum.

Als Beispiel für die Extraktion eines funktionalen Zusammenhanges wird die Beziehung zwischen Verlustleistung und Verzögerungszeit des CMOS Inverters betrachtet. Zwischen Beiden ist eine umgekehrt proportionale Abhängigkeit zu erwarten, so dass unter Berücksichtigung mehrerer Offsets die Funktion $h(x) = a/(x - b) + c$ anzusetzen ist. Die Parameter a , b und c werden durch eine Anpassung mittels evolutionärer Algorithmen für das Ergebnis der Optimierung bestimmt und ergeben sich zu $a = 5,3 \text{ fJ}$, $b = 5,9 \mu\text{W}$ und $c = 380 \text{ ps}$. Die Lösung der Vektoroptimierung ist in Abbildung 6.8 mit dem hier bestimmten, analytischen Zusammenhang dargestellt.

Die ermittelte Beziehung zwischen Verlustleistung und Verzögerungszeit ist in der Ungleichung (6.2) festgehalten. Durch Extrapolieren des Ergebnisses sind absolute Grenzen von $5,9 \mu\text{W}$ für die Verlustleistung und 380 ps für die Verzögerungszeit abzulesen, die eine Bewertung der Schaltung ermöglichen. Dies ist unabhängig von der Tatsache, dass diese Werte praktisch nicht erreichbar sind, da zuvor weitere Einflüsse, wie z.B. maximale Geometrieabmessungen, wirksam werden.

$$(\bar{P} - 5,9 \mu\text{W}) \cdot (t_d - 380 \text{ ps}) > 5,3 \text{ fJ} \quad (6.2)$$

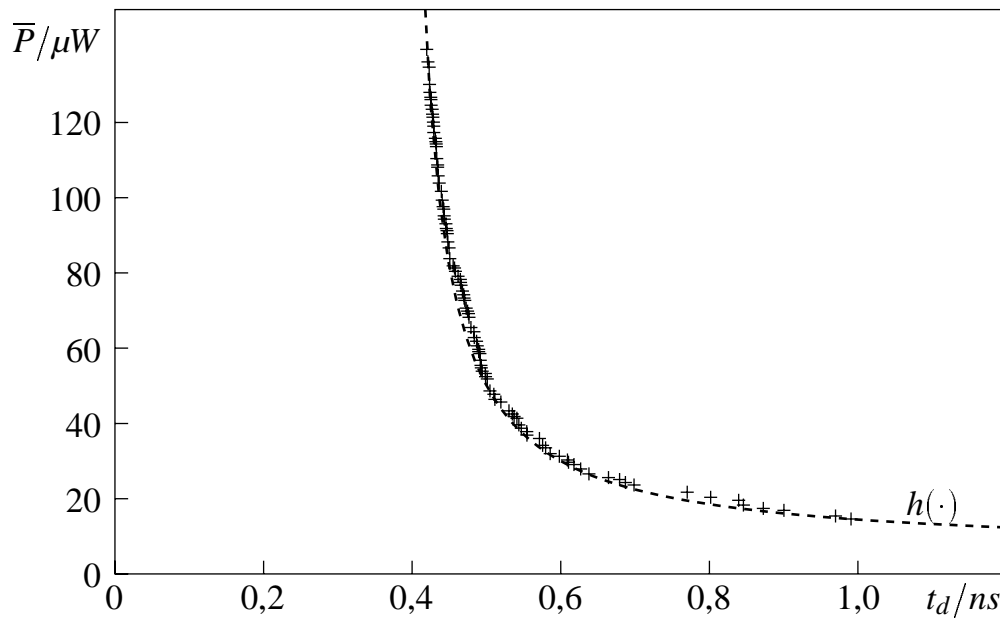


Abbildung 6.8: Verlustleistungs-Verzögerungszeit Diagramm des CMOS Inverters mit der Lösung der Vektoroptimierung und dem daraus abgeleiteten funktionalen Zusammenhang.

6.4 Anwendungsbeispiele

Die Schritte und Ergebnisse zur Durchführung der Bestimmung von Parameterzusammenhängen ist in den vorangehenden Abschnitten am Beispiel eines CMOS Inverters demonstriert worden. In diesem Kapitel wird die Verhaltensanalyse auf Grundsaltungen angewendet. Hierzu werden das Lee-Gatter und der 1-Bit Volladdierer exemplarisch betrachtet, letzterer sowohl in Schwellwerttechnik als auch in statischer CMOS Technik. Die freien Parameter der Schaltungen sind entsprechend den Angaben in Kapitel 4.3 gegeben. Da im Entwurf integrierter Schaltungen die Verlustleistung stets eine gewichtige Rolle spielt, ist deren Zusammenhang mit den weiteren Parametern zentraler Aspekt der Analyse. Es werden Abhängigkeiten der minimalen Verlustleistung sowohl von der Verzögerungszeit als auch der Versorgungsspannung und der verbrauchten Chipfläche betrachtet. Hinzu kommt die ansatzweise Berücksichtigung von Parameterschwankungen bei dem Zusammenhang von Robustheit und Versorgungsspannung.

Verlustleistung und Verzögerungszeit

Zuerst wird die Charakteristik bezüglich Verzögerungszeit und Verlustleistung betrachtet. Die Ergebnisse für die drei Anwendungsbeispiele finden sich in den Abbildungen 6.9, 6.10 und 6.11. Neben dem Optimierungsergebnis ist die extrahierte Hyperbel für diesen Zusammenhang ebenfalls gegeben. Die hieraus ableitbaren Grenzwerte für die Verlustleistung und die Verzögerungszeit sind in Tabelle 6.1 aufgelistet. Ein Vergleich der ermittelten Überdeckungen mit den extrahierten Kurven zeigt, dass die Abschätzung $\bar{P} \sim \frac{1}{t_d}$ bereits für Grundsaltungen nur bedingt geeignet ist, wohingegen sie für den Inverter zutrifft (siehe voriges Kapitel).

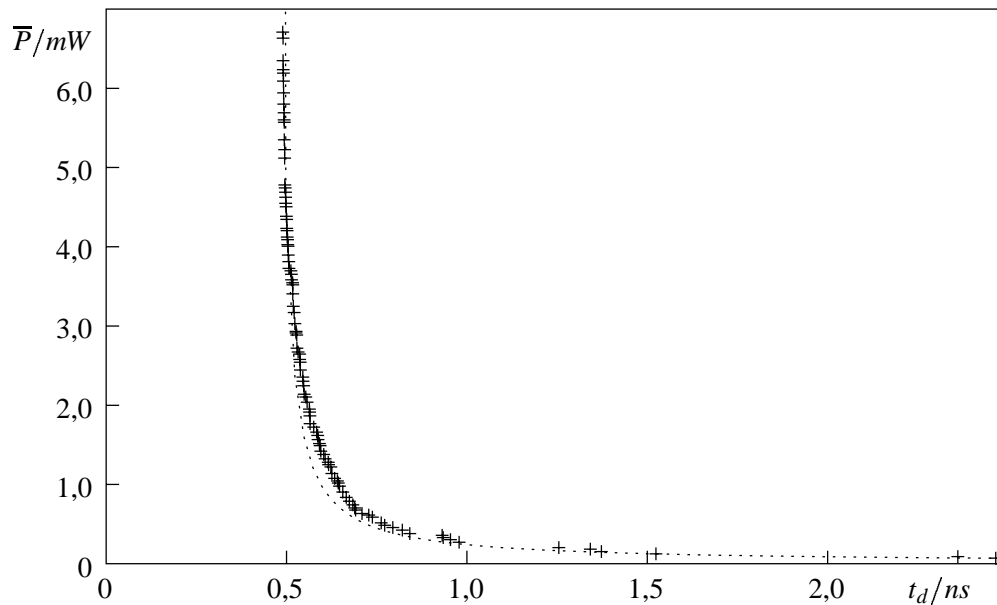


Abbildung 6.9: Verlustleistungs-Verzögerungszeit-Zusammenhang des Lee-Gatters.

Schaltung	\bar{P}	t_d
Lee-Gatter	$7,3 \mu W$	$471 ps$
Lee-Addierer	$163 \mu W$	$474 ps$
Spiegeladdierer	$7,1 \mu W$	$410 ps$

Tabelle 6.1: Grenzwerte der Verlustleistung und der Verzögerungszeit, extrahiert aus den vorhergehenden Optimierungen.

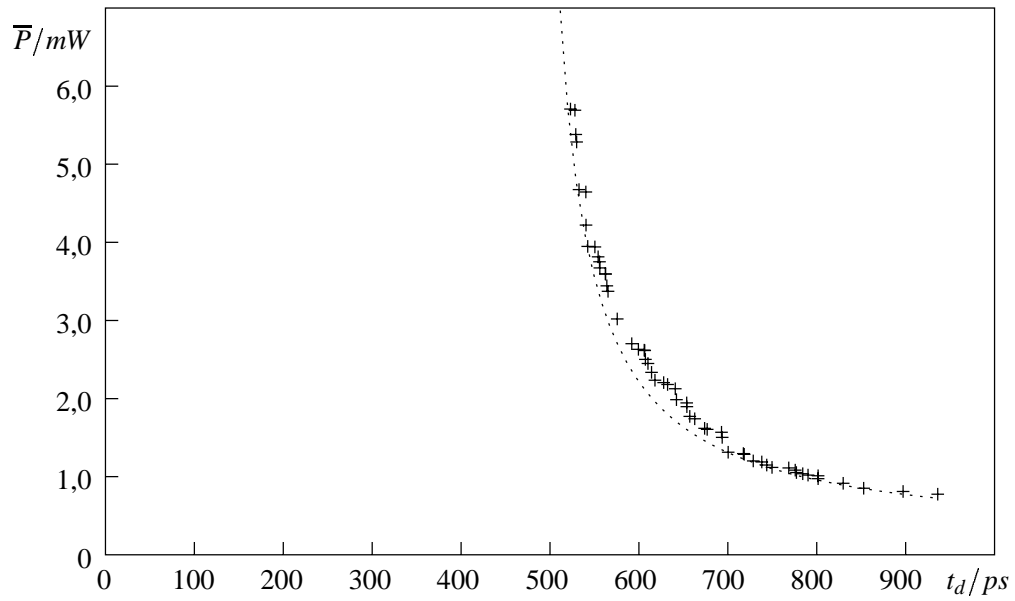


Abbildung 6.10: Verlustleistungs-Verzögerungszeit-Zusammenhang des Lee-Addierers.

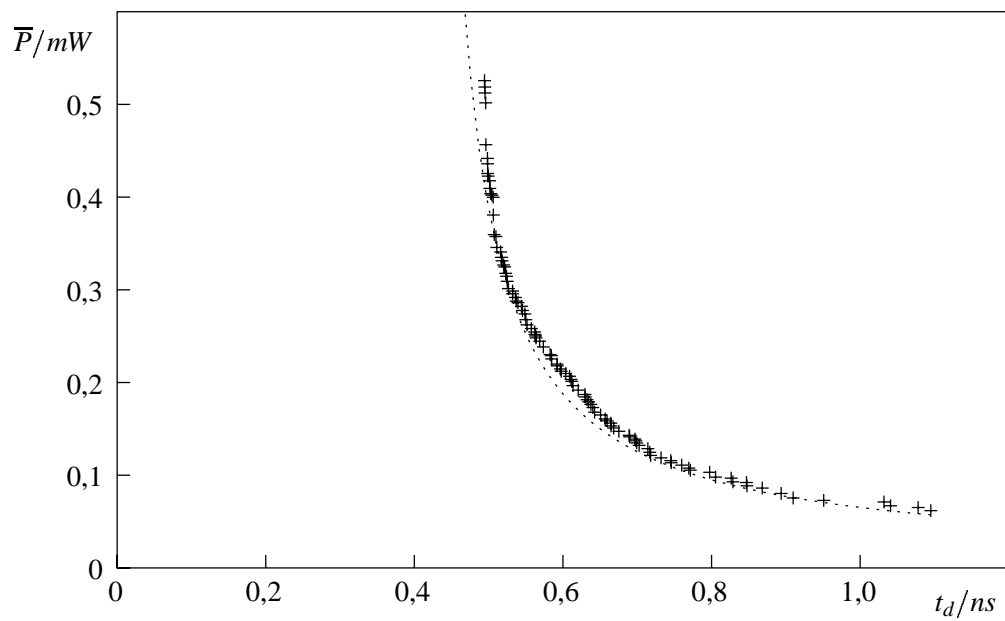


Abbildung 6.11: Verlustleistungs-Verzögerungszeit-Zusammenhang des Spiegeladdierers.

Verlustleistung und Spannung

Ein weiterer Aspekt für den Schaltungsentwurf ist die Spannungsabhängigkeit der Verlustleistung. Wie im Beispiel des CMOS Inverters in Kapitel 6.2 ist mit einem nicht monotonen Zusammenhang zu rechnen. Als synthetische Zielgröße wird hier wiederum der Quotient aus Verlustleistung und potenziertes Spannung verwendet: $\bar{P}' = \frac{\bar{P}}{V_{DD}^4}$. Die Ergebnisse der zugehörigen Optimierung sind in den Abbildungen 6.12 bis 6.14 dargestellt. In allen drei Fällen ist ein Anstieg der Verlustleistung für kleine Spannungen zu beobachten. Dieser Effekt soll am Beispiel des Lee-Gatters durch eine detaillierte Betrachtung der ermittelten Designparameter erklärt werden.

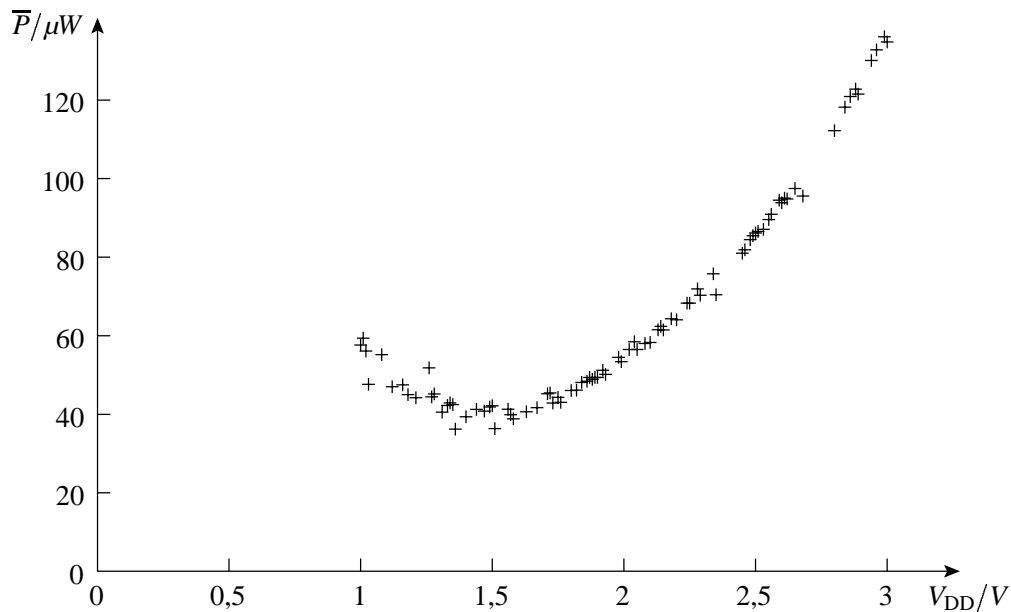


Abbildung 6.12: Verlustleistungs-Versorgungsspannung Zusammenhang des Lee-Gatters.

Die freien Größen des Lee-Gatters sind gemäß Kapitel 4.3.2 die Längen und Weiten der p-Kanal- und der n-Kanal-Transistoren, sowohl auf der Eingangs- als auch auf der Ausgangsseite. Die Versorgungsspannung ist als Zielgröße der Optimierung hierbei nicht zu berücksichtigen. In den Abbildungen 6.15(a) bis 6.15(d) sind die Transistorweiten zu sämtlichen funktionstüchtigen Realisierungen aus der Optimierung mit dem Ergebnis in Abbildung 6.12 dargestellt. Es ist festzustellen, dass die Transistorweiten $W_{p,Out}$ und $W_{n,Out}$ für kleinere Spannungen zunehmen. Somit ergibt sich ein Anstieg der Verlustleistung trotz verringerter Versorgungsspannung. Der gleiche Effekt ist bei den beiden Volladdierern zu beobachten.

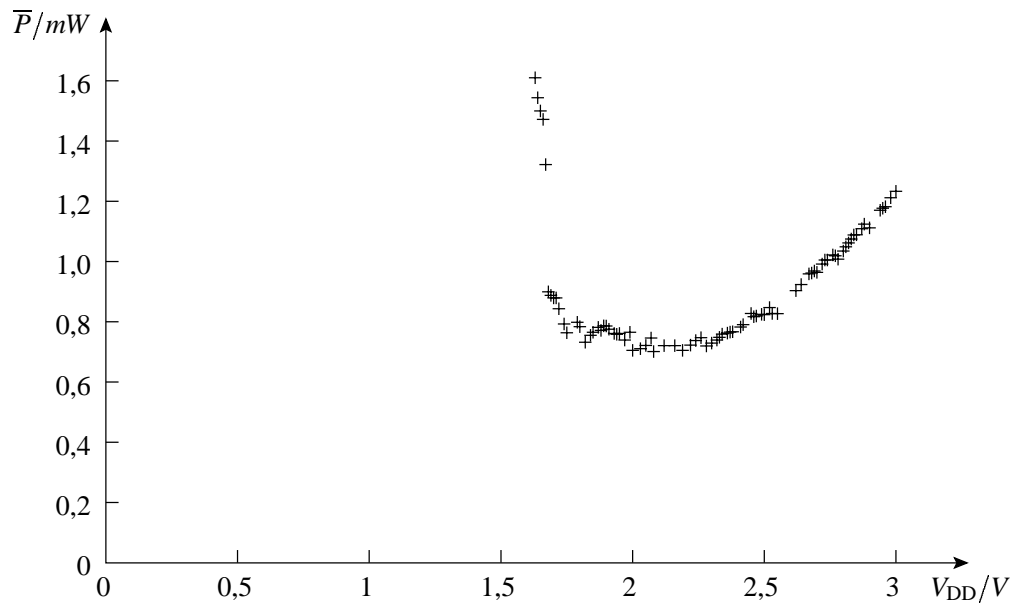


Abbildung 6.13: Verlustleistungs-Versorgungsspannung Zusammenhang des Lee-Addierers.

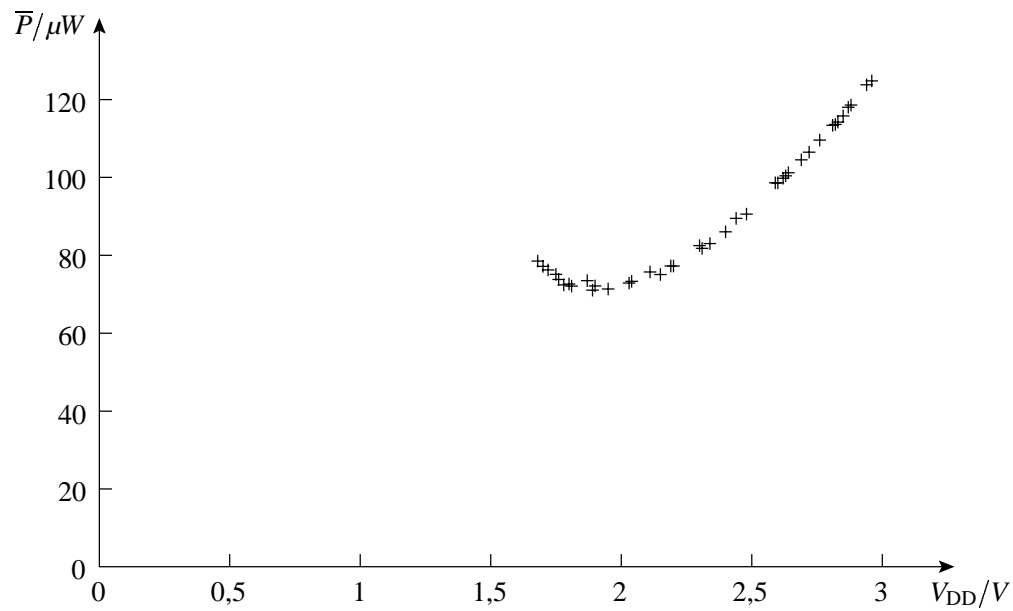


Abbildung 6.14: Verlustleistungs-Versorgungsspannung Zusammenhang des Spiegeladdierers.

Der Scheitelpunkt der Kurve liegt für das Lee-Gatter bei einer Spannung von $1,5\text{ V}$. Die demgegenüber höheren Spannungen im Falle der beiden Addierer ergeben sich aus der höheren Komplexität der geforderten Schaltfunktion (Lee-Addierer: $2,1\text{ V}$, Spiegeladdierer: $1,9\text{ V}$). So ist das Lee-Gatter bezüglich der Verzögerungszeit in erster Näherung durch zwei Inverterstufen zu charakterisieren, während dies für den Spiegeladdierer drei und für den Lee-Addierer sogar fünf sind.

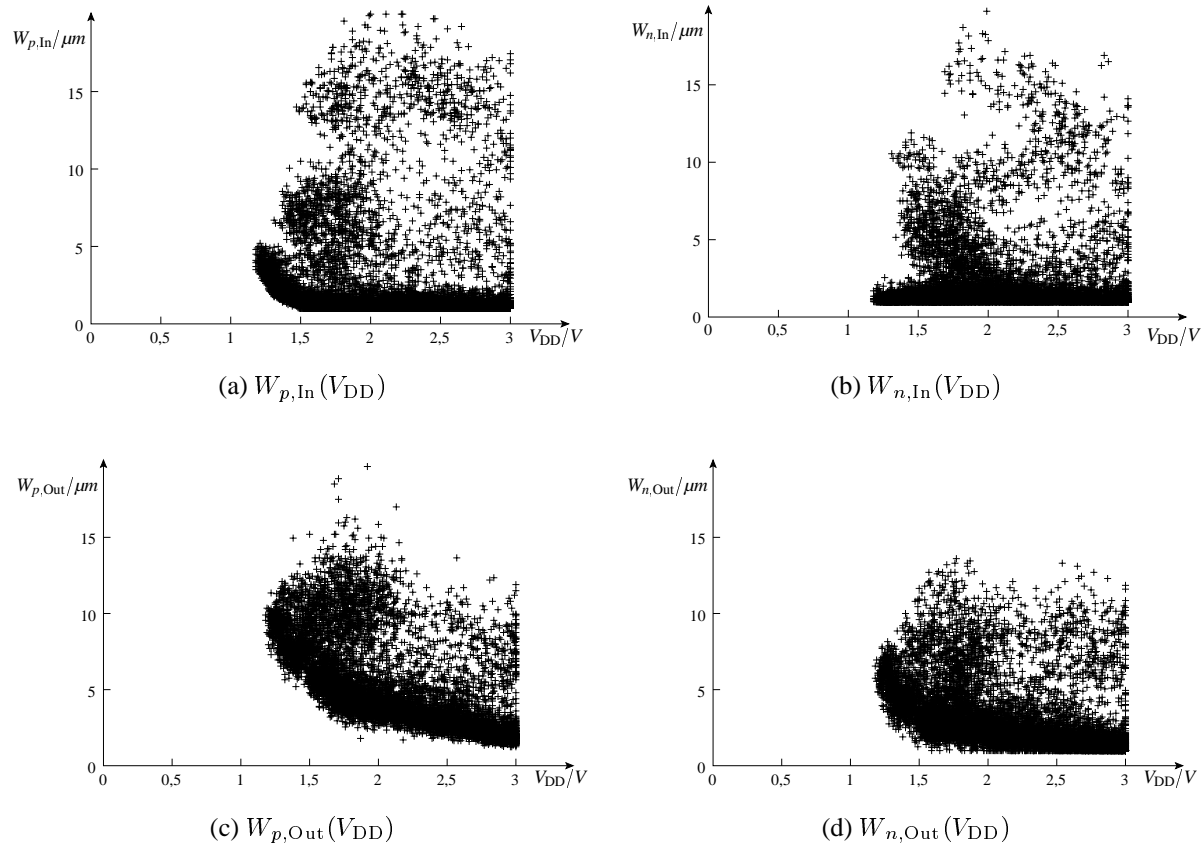


Abbildung 6.15: Transistorweiten zu sämtlichen, korrekten Schaltungsdesigns aus dem Optimierungslauf mit dem Ergebnis in Abbildung 6.12

Verlustleistung und Fläche

Als nächster Punkt sei der Einfluss der verbrauchten Chipfläche auf die Verlustleistung der Schaltung betrachtet. Einerseits wird die Verlustleistung mit der Fläche skalieren, da ein Großteil der Schaltung aktive Fläche, durch das Gate und parasitäre Kapazitäten bestimmt, sein wird. Andererseits wird für sehr kleine Designs eine Spannungserhöhung erforderlich sein, um die geforderten Zeitvorgaben einzuhalten. Die Auswirkungen dieser gegenläufigen Effekte auf die Verlustleistung des Lee-Gatters ist in Abbildung 6.16 zu beobachten. Es zeigt sich ein Minimum bei einer Fläche von $A = 9 \mu\text{m}^2$. Eine weitere Flächenreduktion bewirkt eine größere Verlustleistung.

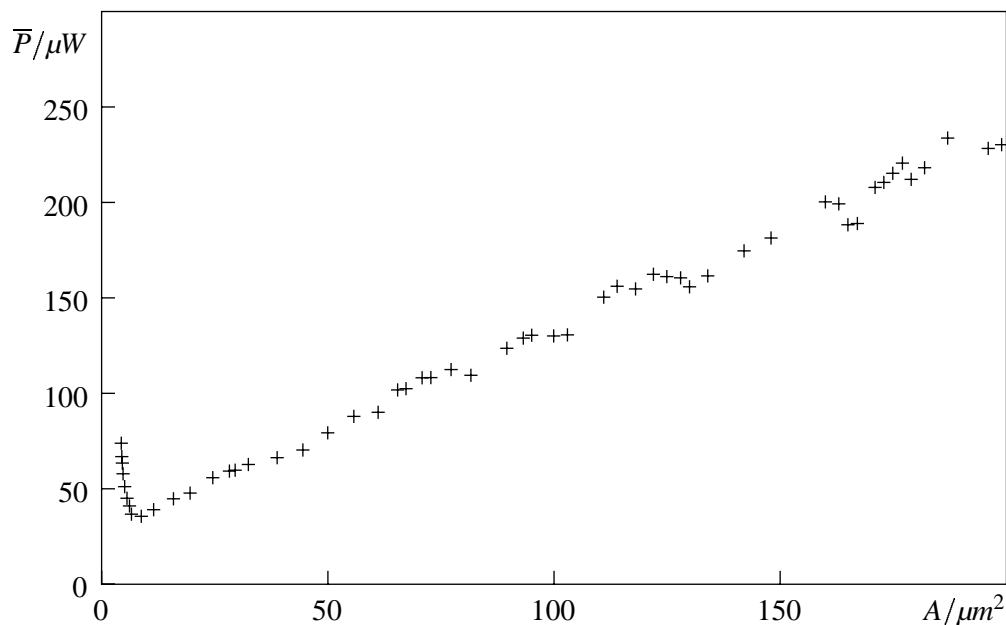


Abbildung 6.16: Verlustleistungsabhängigkeit von der verbrauchten Chipfläche für das Lee-Gatter.

Zur Übertragung der Beziehung zwischen Verlustleistung und Fläche in einen monotonen Zusammenhang genügt die Verwendung der synthetischen Größe $\bar{P}' = \frac{\bar{P}}{A^2}$. Die erforderliche Vektoroptimierung lässt sich wiederum mit dem SPEA Ansatz durchführen, zeigt jedoch eine Schwäche des originalen Verfahrens, die sich in dem verwendeten Clustering-Algorithmus begründet. Im SPEA sind zuvor entwickelte Nischentechniken der Vektoroptimierung mit evolutionären Algorithmen durch einen auf der Dominanzrelation basierten Ansatz ersetzt worden. Die Motivation hierzu liegt in der Unabhängigkeit von unterschiedlichen Skalierun-

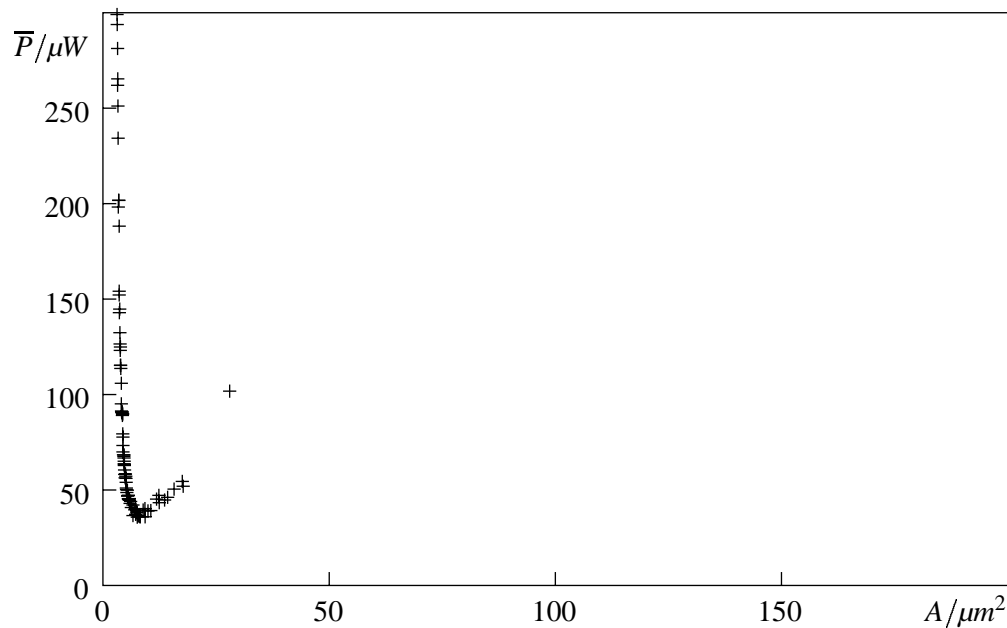


Abbildung 6.17: Verlustleistungsabhängigkeit von der verbrauchten Chipfläche für das Lee-Gatter, ermittelt über die Vektoroptimierung von \bar{P}/A^2 und A .

gen des Zielgrößenraumes. Dieser Vorteil wird durch den hier eingesetzten und erforderlichen Clustering-Algorithmus zur Begrenzung der externen Population relativiert. Durch Verwendung des Euklidischen Abstandsmaßes findet die Skalierung der Zielfunktionsgrößen Berücksichtigung und kann zu einer ungleichgewichtigen Charakterisierung bezüglich der verschiedenen Zielgrößen führen. Im vorliegenden Beispiel bedeutet dies Folgendes: Die synthetische Zielgröße \bar{P}' liegt im Bereich größer 10^{15} , während der Zahlenwert der Fläche A kleiner 10^{-9} ist (Simulationsergebnisse: $3 \cdot 10^{-12} m^2 < A < 2 \cdot 10^{-10} m^2$, $6 \cdot 10^{15} Wm^{-4} < \bar{P}' < 2 \cdot 10^{18} Wm^{-4}$). Dies führt dazu, dass aufgrund des Clustering lediglich der fallende Abschnitt der $\bar{P}(A)$ -Kurve zufriedenstellend approximiert wird, wie in Abbildung 6.17 festzustellen ist. Erst das Ersetzen der Fläche durch eine linear skalierte Größe liefert das Ergebnis in Abbildung 6.16.

Ein Ansatz zur allgemeinen Handhabung derartiger Fälle, der hier jedoch nicht umgesetzt wurde, liegt in der Berücksichtigung der Varianz der erhaltenen Lösungen in den einzelnen Dimensionen des Zielgrößenraumes. Für die vorliegende, zu reduzierende Punktmenge wird hierzu die Intervallbreite des aufgespannten Raumes in jeder Dimension erfasst. Der ausgewertete Abstand zweier Punkte bestimmt sich dann aus dem Verhältnis der Abstände in den

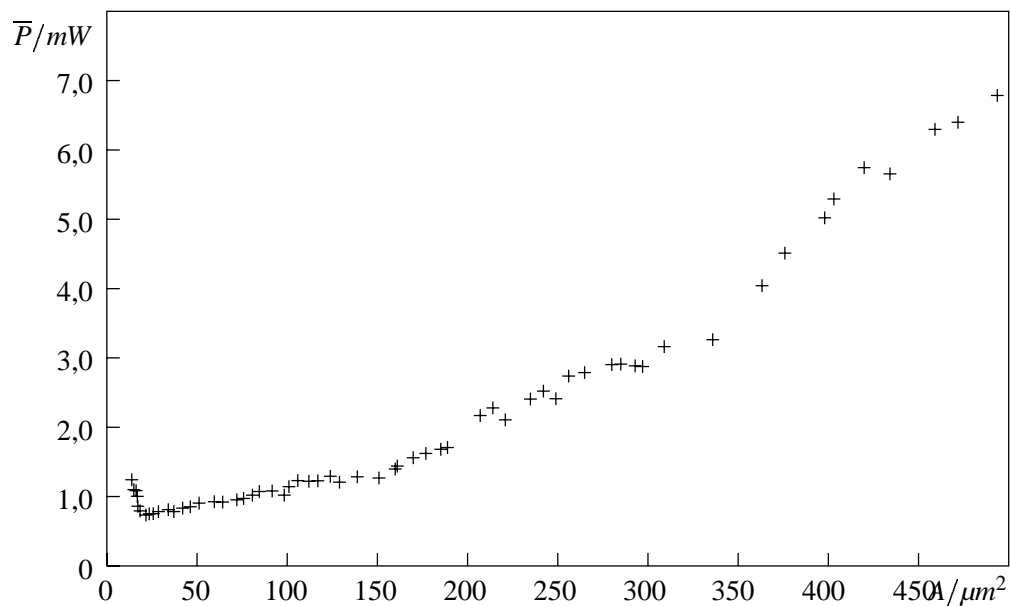


Abbildung 6.18: Verlustleistung-Chipfläche Zusammenhang des Lee-Addierers.

einzelnen Dimensionen zur Intervallbreite. Es ergibt sich somit ein gewichteter Euklidischer Raum, der sich aus den problembezogenen Daten definiert.

Wie an den vorherigen Anwendungsbeispielen der $\bar{P}(t_d)$ -Kurve und der $\bar{P}(V_{DD})$ -Kurve zu erkennen ist, wirkt sich der beschriebene und in diesem Beispiel relevante Effekt jedoch nur bei Zielgrößen aus, die stark unterschiedlich skaliert sind.

Die Verlustleistungsabhängigkeit von der verbrauchten Chipfläche für die Beispiele des Addierers mit Majoritätsgattern und den Spiegeladdierer sind in den Abbildungen 6.18 und 6.19 gegeben.

Versorgungsspannung und Robustheit

Abschließend sollen die Untersuchung von Parameterschwankungen ansatzweise betrachtet und die Durchführbarkeit einer diesbezüglichen Parameteroptimierung demonstriert werden. Unter dem Aspekt der Robustheit ist hier die (Un-) Empfindlichkeit gegenüber Schwankungen der Eingangssignale zu verstehen. Es werden also nicht die Fertigungsschwankungen berücksichtigt, die sich in Variation der Geometrieparameter oder Dotierungen bzw. Schwellenspannungen auswirken. Die Robustheit r definiert sich hier, wie in Kapitel 4.3.5 geschildert, als

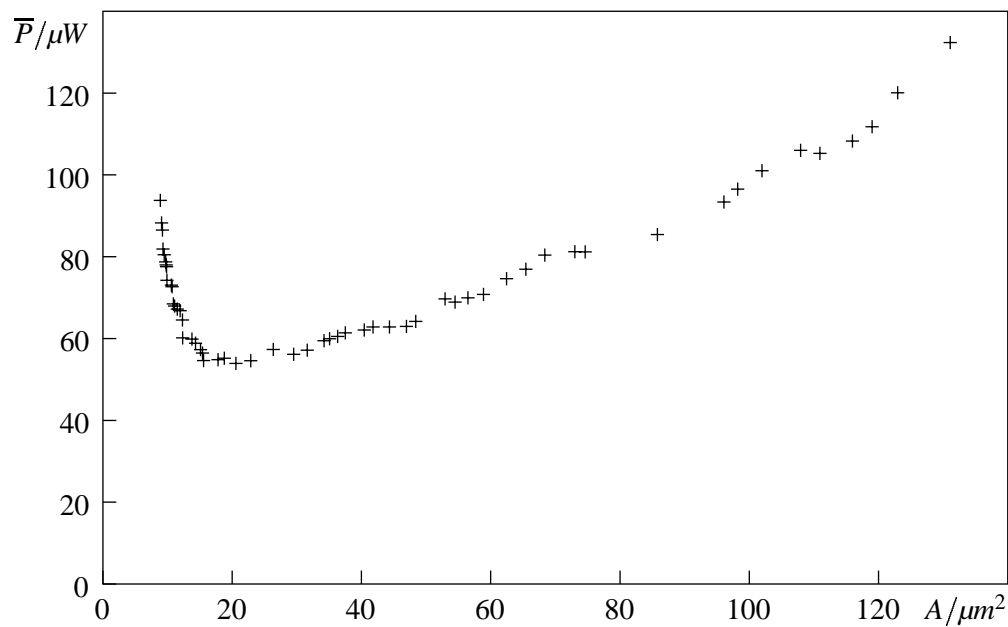


Abbildung 6.19: Verlustleistung-Chipfläche Zusammenhang des Spiegeladdierers.

relative Änderung des Eingangssignals gegenüber der Versorgungsspannung ($r = \frac{\Delta V}{V_{DD}}$). Für die Optimierung werden für eine Simulation die Robustheit vorgegeben und in deren Rahmen die drei Eingangsspannungen zufällig und gleichverteilt bestimmt. Zur Erhöhung des Vertrauensniveaus dieser Validierung wird eine Stichprobe vom Umfang 50 erstellt. Das heißt es werden 50 Simulationen durchgeführt und überprüft. Als Robustheit zur Bewertung wird anschließend nicht die Vorgabe der Zufallsvariablen, sondern die maximale Schwankung herangezogen, die innerhalb der Stichprobe aufgetreten ist. Diese Differenzierung ist notwendig, da sich die beiden Größen in einem relevanten Maße unterscheiden können und eventuell zu falschen Aussagen führen würden.

Der Zusammenhang zwischen der maximalen Robustheit und der Versorgungsspannung kann als monoton steigend angenommen werden, so dass eine einfache Negation $r'(V_{DD}) = -r(V_{DD})$ genügt, eine monoton fallende Abhängigkeit zu erhalten. Das Ergebnis der Optimierung ist in Abbildung 6.20 zu finden.

Durch die mehrfache Simulation für einen Parametersatz erhöht sich der Zeitbedarf erheblich und steigt in diesem Fall um den Faktor 30. Dadurch sinkt die Zahl der durchführbaren Zielfunktionsberechnungen spürbar und führt zu einer weniger verlässlichen Überdeckung der Pareto Menge.

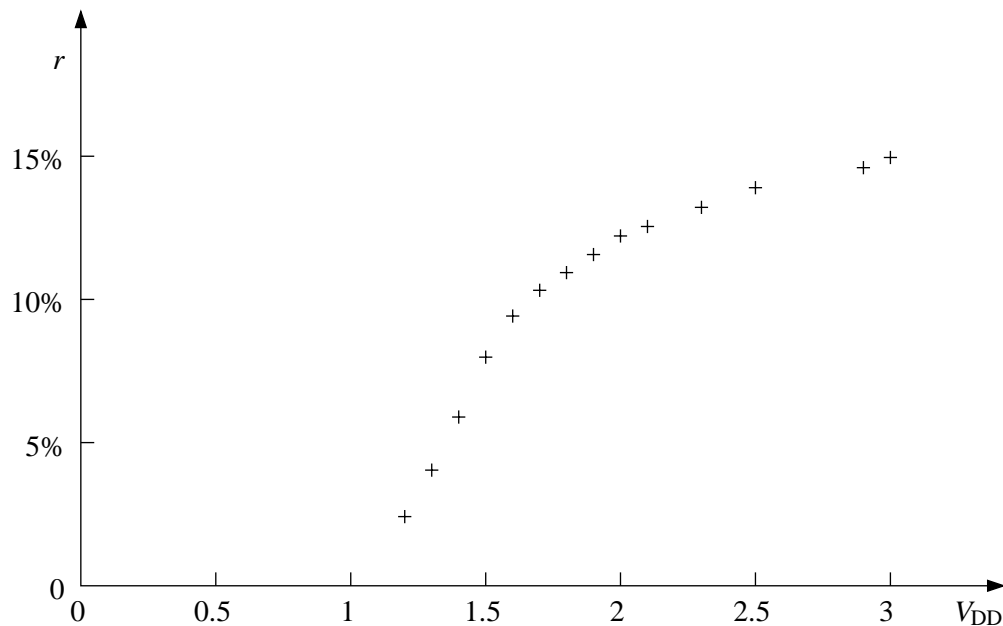


Abbildung 6.20: Zusammenhang zwischen relativer Robustheit und der Versorgungsspannung für das Lee-Gatter.

6.5 Bewertung der Ergebnisse und Ausblick

Die Summe der vorgestellten, verschiedenartigen Anwendungen der evolutionären Vektoroptimierung zur Analyse integrierter Schaltungen demonstriert die Vielseitigkeit dieses Ansatzes. Beschränkt auf die Parameteroptimierung auf Transistorebene erlaubt die beschriebene Methodik einen detaillierteren Einblick in typische Fragestellungen der Schaltungsanalyse, als dies bislang möglich gewesen ist. Zusammenhänge zwischen Verlustleistung und Verzögerungszeit sind für eine Gegenüberstellung verschiedener Schaltungstechniken oder unterschiedlicher Technologien mit dem vorgestellten Ansatz nicht mehr auf eine einzelne Größe, wie das Power-Delay-Produkt, oder auf die Variation eines einzelnen Parameters beschränkt. Vielmehr ist ein Überblick über das gesamte Spektrum dieses Zusammenhangs möglich. Neben diesen ‚klassischen‘ Aspekten eröffnet sich durch eine effiziente Bestimmung von Parameterzusammenhängen ein Feld neuartiger Fragestellungen. Es bietet sich die Chance, die Wechselwirkung zwischen verschiedenen Parametern zu untersuchen und aus den gewonnenen Beobachtungsergebnissen ein tieferes Verständnis für die internen Vorgänge der Schaltung zu entwickeln.

Trotz aller Möglichkeiten, die mit diesem Ansatz verbunden sind, ist diese Methodik weit von einer automatisierten Analyse und Auswertung gegebener Schaltungen entfernt. Die weiter bestehende Herausforderung stellt sich in der Generierung geeigneter Hypothesen (über Parameterzusammenhänge) zur Gewinnung neuer schaltungsrelevanter Information. Das Beispiel der $\overline{P}(V_{DD})$ -Kurve betrachtend, genügt es nicht über den nicht monotonen Zusammenhang informiert zu sein. Erst das Wissen um die mit der Spannung in etwa quadratisch wachsende Verlustleistung erlaubt die Wahl einer adäquaten Transformation in einen monotonen Zusammenhang. Je besser die Grundhypothese zutrifft, desto effektiver kann die Bestimmung des Parameterzusammenhanges mit Hilfe der evolutionären Vektoroptimierung erfolgen.

Der wesentliche Kritikpunkt an den evolutionären Algorithmen (im Allgemeinen) bezüglich der Schaltungsoptimierung liegt im Rechenaufwand, d.h. der Zahl der erforderlichen Schaltungssimulationen. In dem mit dieser Arbeit vorliegenden experimentellen System erfolgt die Optimierung der hier betrachteten Grundsaltungen mit 200.000 Simulationen in rund 20 Stunden auf dem Rechnernetz des Lehrstuhls für Bauelemente der Elektrotechnik an der Universität Dortmund. Da die Optimierungssoftware und der Schaltungssimulator SPICE getrennt voneinander existieren und lediglich über das Dateisystem kommunizieren können, ist das Zusammenspiel allerdings vergleichsweise ineffizient. Hinzu kommt weiterer Overhead für Lizenzmanagement der SPICE-Software über das Netzwerk. Versuche zu mehrfachen Simulationen von identischen Schaltungen zeigen, dass durch eine effiziente Kopplung beider Seiten die 200.000 Simulationen auf identischer Hardware in ca. zehn Stunden durchführbar sind. Auf Basis der SPECfp- und SPECint-Benchmarkwerte der vorliegenden und aktueller Rechnersysteme lässt sich der Zeitaufwand bei gleicher Parallelität auf unter zwei Stunden abschätzen. Eine weitere Beschleunigung durch stärkere Parallelisierung ist durch die Populationsgröße beschränkt, würde an dieser Stelle jedoch eine weitere zeitliche Reduktion um den Faktor drei ermöglichen.

Die Steigerung verfügbarer Rechenleistung der vergangenen Jahre betrachtend und gemäß „Moore’s Law“ hochgerechnet, ergibt sich für die kommenden Jahre eine Laufzeit im einstelligen Minutenbereich. Auf der anderen Seite sind die betrachteten, einfachen Anwendungsbeispiele nur exemplarisch für größere Systeme zu sehen. Dem Performancegewinn in der Optimierung steht der Aufwandszuwachs in Verbindung mit komplexeren Addierern oder Multiplizierern entgegen. Diese erfordern einerseits ein Mehr an Simulationszeit zur Überprüfung der logischen Funktion aufgrund der komplexeren Struktur. Andererseits wird die Optimierung wegen der größeren Zahl an freien Parametern schwieriger und zwingt zu längeren Laufzeiten in Bezug auf die Zahl der Simulationen.

Für die Betrachtung großer Systeme wären dann hybride Ansätze, die z.B. eine Analyse des kritischen Pfades mit einer, in der Schaltungsstruktur lokalen, evolutionären Suche kombinieren. Derart spezifischere Suchstrategien gehen jedoch mit einer Einschränkung des Suchraumes einher.

Neben den in den Beispielen betrachteten Spannungs- und Geometrievariationen sind ebenso Optimierungen bezüglich der Technologieparameter möglich. In dieser Richtung lassen sich für gegebene Schaltungskonzepte Anforderungen an die zugrunde liegende Technologie ableiten. Somit kann ein intensiverer Austausch zwischen Schaltungsdesign und Technologie im Forschungsstadium unterstützt werden.

Im Hinblick auf die Zielsetzung dieser Arbeit, das Design und die Analyse integrierter Schaltungen mit Perspektive auf neuartige Schaltungstechniken und Technologien, für die grundlegende Aspekte untersucht und Grundsaltungen betrachtet werden, stellen die hier vorgestellten evolutionsbasierten Methoden ein sinnvolles und ergänzendes Hilfsmittel für die Forschung und Entwicklung dar.

Kapitel 7

Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde die nominelle Optimierung integrierter Digital-schaltungen auf Transistorebene zur Untersuchung neuartiger Schaltungstechniken betrach-tet. Hierbei fanden die verschiedenen Aspekte des Entwurfs, wie die logische Funktion, das Timing, die Stabilität und Robustheit, Berücksichtigung. Für die globale Optimierung schal-tungsspezifischer Größen konnten evolutionäre Algorithmen erfolgreich eingesetzt werden. Die hier verwendete Variante einer Evolutionsstrategie wurde entwickelt hinsichtlich der Ei-genschaften Parallelität, Robustheit gegenüber Störungen der Simulationsumgebung und mög-lichst geringen Kenntnissen bezüglich des Optimierungsverfahrens. Mit der bislang erreich-ten Effizienz erlaubt das Verfahren die Betrachtung digitaler Grundsaltungen. Allerdings zeigen sich auch deutliche Grenzen bei der Komplexität handhabbarer Schaltungen auf. Der Ansatz entspricht somit jedoch dem fokussierten Forschungsbereich zur Untersuchung neuer Schaltungstechniken mit geringem spezifischen Vorwissen. Für industrielle Anwendungen zur Optimierung größerer Schaltungskomplexe wie beispielsweise Multiplizierer wird es weiter-hin erforderlich sein, spezifisches Wissen über die verwendete Technologie in den gesamten Entwurfsprozess und die Optimierungsalgorithmen einfließen zu lassen.

Die zeitgleiche Berücksichtigung mehrerer Kriterien führt zum Thema Vektoroptimierung. Hier konnte gezeigt werden, dass dieser Ansatz zur Schaltungsoptimierung mit evolutio-nären Algorithmen effizient durchzuführen ist. Die hier vorgestellte Anwendung der Vektor-optimierung zur Bestimmung von Parameterzusammenhängen eröffnet die Möglichkeit einer Verhaltensanalyse integrierter Schaltungen. Es wurde gezeigt, wie trotz der immanenten Be-schränkungen der Vektoroptimierung nahezu beliebige Charakteristiken untersucht werden können und sich aus den experimentellen Ergebnissen eine analytische Form ableiten lässt.

Die unterschiedlichen Anwendungsbeispiele in dieser Arbeit zeigen einerseits die Möglichkeiten des hier verfolgten Ansatzes, andererseits die Qualität der erzielbaren Ergebnisse. Die Voraussetzung einer geeigneten Hypothese zur erfolgreichen Bestimmung eines Parameterzusammenhangs zeigt jedoch auch, dass dieses Verfahren das physikalische Grundlagenwissen des Anwenders nicht ersetzen kann.

Eine Steigerung der Effizienz der Vektoroptimierung könnte für den Anwendungsbereich integrierter Schaltungen die Entwicklung eines hybriden Algorithmus bieten, der die Vorteile des evolutionären Algorithmus und des Gradientenabstiegs miteinander vereint. Des Weiteren bietet es sich an, Techniken der Vektoroptimierung in die Eingrößenoptimierung einfließen zu lassen. So ließe sich durch Nischenbildung im Parameterraum eine Exploration des Entwurfsraumes für verlustleistungsoptimierte Schaltungen erreichen. Mit Methoden der Regelextraktion wäre dann die Ableitung von Entwurfsregeln durchführbar, die einen systematischen Entwurf verlustleistungsarmer Schaltungen erlauben.

Literaturverzeichnis

- [AGW94] ANTREICH, K. J., H. E. GRAEB und C. U. WIESER: *Circuit Analysis and Optimization Driven by Worst-Case Distances*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 13(1), 1994.
- [BBNH95] BÄCK, T., T. BEIELSTEIN, B. NAUJOKS und J. HEISTERMANN: *Evolutionary Algorithms for the Optimization of Simulation Models Using PVM*. In: DONGARRA, J., M. GENGLER, B. TOURANCHEAU und X. VIGOUROUX (Herausgeber): *EuroPVM'95 Second European PVM Users' Group*, Seiten 277–282, Lyon, France, 14.-15. September 1995. Hermès, Paris.
- [BBvdW⁺98] BROEKAERT, T. P. E., B. BRAR, J. P. A. VAN DER WAGT, A. C. SEABAUGH, F. J. MORRIS, T. S. MOISE, E. A. BEAM und G. A. FRAZIER: *A Monolithic 4-Bit 2-Gsps Resonant Tunneling Analog-to-Digital Converter*. IEEE Journal of Solid-State Circuits, 33(9):1342–1347, 1998.
- [Bäc96] BÄCK, T.: *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, New York, 1996.
- [BHSV81] BRAYTON, R. K., G. D. HACHTEL und A. L. SANGIOVANNI-VINCENTELLI: *A Survey of Optimization Techniques for Integrated-Circuit Design*. Proceedings of the IEEE, 69(10):1334–1362, 1981.
- [BSMM99] BRONSTEIN, I. N., K. A. SEMENDJAJEW, G. MUSIOL und H. MÜHLIG: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 1999. 4. Auflage der Neubearbeitung.
- [BTDG01] BURWICK, C., M. THOMAS, J. DIENSTUHL und K. F. GOSER: *Threshold Gates in Arithmetic Circuits*. In: *The 8th IEEE International Conference on*

- Electronics, Circuits and Systems*, Band II, Seiten 909–912, 2.-5. September 2001.
- [BTG00] BURWICK, C., M. THOMAS und K. GOSER: *CMOS Threshold Logic Circuits for Arithmetic Computations*. In: *ITG Workshop Mikroelektronik für die Informationstechnik*, Seiten 121–126, Darmstadt, 20.-21. November 2000.
- [CB95] CHANDRAKASAN, A. P. und R. W. BRODERSEN (Herausgeber): *Low Power Digital CMOS Design*. Kluwer Academic Publishers, Boston, MA, USA, 1995.
- [CCH⁺98] CONN, A. R., P. K. COULMAN, R. A. HARING, G. L. MORILL, C. VISWESWARIAH und C. W. WU: *Jiffy Tune: Circuit Optimization Using Time-Domain Sensitivities*. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1292–1309, 1998.
- [Deb98] DEB, K.: *Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems*. Technischer Bericht 49/98, SFB 531, Universität Dortmund, 1998.
- [Deb00] DEB, K.: *An Overview of Multi-Objective Evolutionary Optimization.*, Februar 2000. Vortrag im CI-Kolloquium des Sonderforschungsbereichs *Computational Intelligence*.
- [FF95] FONSECA, C. M. und P. J. FLEMING: *An Overview of Evolutionary Algorithms in Multiobjective Optimization*. *Evolutionary Computation*, 3(1):1–16, 1995.
- [GLPS97] GIRARD, P., C. LANDRAULT, S. PRAVOSSOUDOVITCH und D. SEVERAC: *A non-iterative gate sizing algorithm for high reduction in power consumption*. *INTEGRATION, the VLSI journal*, 24:37–52, 1997.
- [Gol89] GOLDBERG, D.: *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, USA, 1989.
- [GPKR97] GOSER, K., C. PACHA, A. KANSTEIN und M. ROSSMANN: *Aspects of Systems and Circuits for Nanoelectronics*. *Proceedings of the IEEE*, 85(4):558–573, 1997.

- [GR87] GOLDBERG, D. und J. RICHARDSON: *Genetic Algorithms with Sharing for Multimodal Function Optimization*. In: *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Seiten 41–49, Hillsdale, NJ, USA, 1987.
- [HB98] HAMMEL, U. und T. BÄCK: *Optimierung in der Simulation: Evolutionäre Algorithmen*. Reihe Computational Intelligence, CI-38/98, 1998.
- [HIG94] HOROWITZ, M., T. INDERMAUR und R. GONZALEZ: *Low-Power Digital Design*. In: *IEEE Symposium on Low Power Electronics*, Seiten 8–11, 1994.
- [HKP91] HERTZ, J., A. KROGH und R. G. PALMER: *Introduction to the Theory of Neural Computation*. Santa Fé Institute Studies in Science of Complexity. Lecture Notes Volume I. Addison-Wesley, Redwood City, CA, USA, 1991.
- [Hol75] HOLLAND, J. H.: *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
- [Joh88] JOHNSON, M. G.: *CMOS NOR Gate for High-Speed Applications*. *IEEE Journal of Solid State Circuits*, 23:1233–1236, 1988.
- [KB97] KOZIEL, S. und M. BIALKO: *Multiobjective optimization of electronic circuits using evolutionary methods*. In: *Proceedings of European Conference on Circuit Theory and Design (ECCTD'97)*, Seiten 451–456, Budapest, Ungarn, 31. August-3. September 1997.
- [KBAK96] KOZA, J. R., F. H. BENNETT III., D. ANDRE und M. A. KEANE: *Automated WYWIWYG Design of Both the Topology and Component Values of Electrical Circuits Using Genetic Programming*. In: *Genetic Programming Conference*, Seiten 123–131, 1996.
- [KL99] KANG, S. M. und Y. LEBLEBICI: *CMOS Digital Integrated Circuits: Analysis and Design*. McGraw-Hill, 1999.
- [KTG97] KANSTEIN, A., M. THOMAS und K. GOSER: *Possibilistic Reasoning in a Computational Neural Network*. In: KARAYIANNIS, N. B. (Herausgeber): *Proc. IEEE Int'l Conf. Neural Networks*, Band 4, Seiten 2541–2545, Houston, TX, USA, 9.-12. Juni 1997. IEEE.

- [Lan98] LANDWEHR, B.: *ILP-basierte Mikrostruktur-Synthese mit komplexen Bausteinbibliotheken*. Nummer 282 in *Reihe 20: Rechnerunterstützte Verfahren*. VDI Verlag, 1998.
- [LC99] LOHN, J. D. und S. P. COLOMBANO: *A Circuit Representation Technique for Automated Circuit Design*. IEEE Transactions on Evolutionary Computation, 3(3):205–219, 1999.
- [LJ92] LEE, C. L. und C.-W. JEN: *Bit-sliced median filter design based on majority gate*. IEE Proceedings-G Circuits, Devices & Systems, 139(1):63–71, 1992.
- [LK82] LAW, A. und D. KELTON: *Simulation Modeling and Analysis*. McGraw-Hill, Boston, USA, 1982.
- [LRS98] LAUMANN, M., G. RUDOLPH und H.-P. SCHWEFEL: *A Spatial Predator-Prey Approach to Multi-Objective Optimization*. In: BÄCK, T., A. E. EIBEN, M. SCHOENAUER und H.-P. SCHWEFEL (Herausgeber): *Parallel Problem Solving From Nature - PPSN V*, Band 1498 der Reihe *Lecture Notes in Computer Science*, Seiten 241–249, Berlin, 1998. Springer.
- [ML98] MLYNEK, D. und Y. LEBLEBICI: *Design of VLSI Systems*. Internet, <http://vlsi.wpi.edu/webcourse>, 1998.
- [MR99] MAZUMDER, P. und E. M. RUDNICK (Herausgeber): *Genetic Algorithms for VLSI Design, Layout & Test Automation*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.
- [Mur71] MUROGA, S.: *Threshold Logic and its Applications*. Wiley-Interscience, New York, USA, 1971.
- [NRSVT88] NYE, W., D. C. RILEY, A. SANGIOVANNI-VINCENTELLI und A. L. TITS: *DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits*. IEEE Transactions on Computer-Aided Design, 7(4):501–519, 1988.
- [OGS99] OEHM, J., U. GRÜNEBAUM und K. SCHUMACHER: *Mismatch Effects Explained by the Spectral Model*. In: *The 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS'99)*, Band II, Seiten 1055–1058, Pafos, Zypern, 5.-8. September 1999.

- [Pac01] PACHA, C.: *Schaltungskonzepte für die Nanoelektronik mit Resonanz-Tunnelbauelementen*. Doktorarbeit, Fakultät für Elektrotechnik und Informationstechnik, Universität Dortmund, 2001.
- [PC97] PEIK, S. F. und Y. L. CHOW: *Genetic Algorithms Applied to Microwave Circuit Optimization*. In: *Proceedings of Asia Pacific Microwave Conference*, Band 2, Seiten 857–860, Hong Kong, 1997.
- [PGBP98] PACHA, C., K. GOSER, A. BRENNEMANN und W. PROST: *A Threshold Logic Full Adder Based on Resonant Tunneling Transistors*. In: *European Solid-State Circuits Conference (ESSCIRC'98)*, Seiten 427–431, Den Haag, Niederlande, 22.-24. September 1998.
- [Pir96] PIRSCH, P.: *Architekturen der digitalen Signalverarbeitung*. B. G. Teubner, Stuttgart, 1996.
- [Rab96] RABAEY, J. M.: *Digital Integrated Circuits: A Design Perspective*. Prentice-Hall, Upper Saddle River, NJ, USA, 1996.
- [Rec73] RECHENBERG, I.: *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [SC95] SMITH, A. E. und D. W. COIT: *Penalty Functions*. In: BÄCK, T., D. B. FOGEL und Z. MICHALEWICZ (Herausgeber): *Handbook of Evolutionary Computation*, Seiten C5.2: 1–13. Oxford University Press, Institute of Physics Publishing, 1995.
- [Sch85] SCHAFFER, J. D.: *Multiple objective optimization with vector evaluated genetic algorithms*. In: *Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Seiten 93–100, Hillsdale, NJ, USA, 1985.
- [Sch95] SCHWEFEL, H.-P.: *Evolution and Optimum Seeking*. John Wiley & Sons, New York, USA, 1995.
- [SD95] SRINIVAS, N. und K. DEB: *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. *Evolutionary Computation*, 2(3):221–248, 1995.

- [Sha38] SHANNON, C.: *A Symbolic Analysis of Relais and Switching Circuits*. Transactions of the AIEE, 57:713–723, 1938.
- [SIA00] SEMICONDUCTOR INDUSTRY ASSOCIATION: *International Technology Roadmap for Semiconductor*, <http://public.itrs.net>, 2000.
- [SM95] STANLEY, T. J. und T. MUDGE: *A Parallel Genetic Algorithm for Multiobjective Microprocessor Design*. In: *6th International Conference on Genetic Algorithms*, Pittsburgh, PA, USA, 15.-19. Juli 1995.
- [SV96] STARCK, V. und M. VALTONEN: *An Implementation of Simulated Annealing for Analog Circuit Optimization*. In: *Baltic Electronics Conference (BEC'96)*, Seiten 377–380, Tallin, Estonia, 7.-11. Oktober 1996.
- [TBG00] THOMAS, M., C. BURWICK und K. GOSER: *Circuit Analysis and Design using Evolutionary Algorithms*. Reihe Computational Intelligence, CI-85/00, 2000.
- [TG00] THOMAS, M. und K. GOSER: *Iterative Weight Selection to Multi-Objective Optimization*. In: *16th IMACS World Congress*, Lausanne, Schweiz, 21.-25. August 2000. Nr. 215-14.
- [TKG97] THOMAS, M., A. KANSTEIN und K. GOSER: *Rare Fault Detection by Possibilistic Reasoning*. In: REUSCH, B. (Herausgeber): *Computational Intelligence. Theory and Application*, Band 1226 der Reihe *Lecture Notes in Computer Science*, Seiten 294–298. Springer-Verlag, 1997.
- [TPG99] THOMAS, M., C. PACHA und K. GOSER: *Parameter Determination for Nano-Scale Modeling*. In: REUSCH, B. (Herausgeber): *Computational Intelligence. Theory and Application*, Band 1625 der Reihe *Lecture Notes in Computer Science*, Seiten 421–426. Springer-Verlag, 1999.
- [Vee84] VEENDRICK, H.: *Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits*. IEEE Journal of Solid-State Circuits, SC-19(4):468–473, 1984.
- [Vee98] VEENDRICK, H.: *Digital Goes Analog*. In: J. H. HUIJSING, A. H. M. VAN ROERMUND, H. GRÜNBAKER (Herausgeber): *Proc. of the 24th European Solid-State Circuits Conference (ESSCIRC)*, Seiten 44–50, The Hague, Niederlande, 22.-24. September 1998.

- [Vis97] VISWESWARIAH, C.: *Optimization Techniques for High-Performance Digital Computing*. In: *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*, Seiten 198–207, 9.-13. November 1997.
- [vVL99] VELDHUIZEN, D. A. VAN und G. B. LAMONT: *Multiobjective Evolutionary Algorithm Test Suites*. In: *Proceedings of the 1999 ACM symposium on Applied computing*, Seiten 351–357, San Antonio, TX, USA, 28. Februar-02. März 1999.
- [WE85] WESTE, N. und K. ESHRAGIAN: *Principles of CMOS VLSI Design: A Systems Perspective*. Addison-Wesley, Reading, MA, USA, 1985.
- [YS96] YUAN, J. und C. SVENSSON: *Principle of CMOS Circuit Power-Delay Optimization with Transistor Sizing*. In: *IEEE International Symposium on Circuits and Systems*, Band 1, Seiten 637–640, Atlanta, GA, USA, 12.-15. Mai 1996.
- [Zit99] ZITZLER, E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Doktorarbeit, Eidgenössische Technische Hochschule Zürich, 1999.
- [ZT99] ZITZLER, E. und L. THIELE: *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. *IEEE Transactions on Evolutionary Computation*, 3(4):257–71, 1999.