Ein Mehrzieloptimierungsansatz zur Dimensionierung Ressourceneffizienter Integrierter Schaltungen

Von der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

Dissertation

von

Dipl.-Math. Matthias W. Blesken

Erster Gutachter: Prof. Dr.-Ing. Ulrich Rückert
Zweiter Gutachter: Prof. Dr.-Ing. Klaus Meerkötter

Tag der mündlichen Prüfung: 16.11.2011

Paderborn 2012

Diss. EIM-E/279

Inhaltsverzeichnis

Einleitung

Zur Aufgabenstellung

Entwurfsraumexploration beschreibt die systematische Suche nach Entwurfsparametern einer Schaltung mit optimalen Schalteigenschaften wie Schnelligkeit, Energiesparsamkeit und Robustheit gegenüber Rauschen. Äquivalent zu optimalen Schalteigenschaften spricht man auch von optimalem Verbrauch verfügbarer Ressourcen (Zeit, Energie usw.), der durch Ressourcenmaße quantifizierbar ist. Diese Ressourcenmaße stehen häufig in Konkurrenz zueinander. Zum Beispiel erhöht sich die Verlustleistung mit steigender Schaltgeschwindigkeit. Als ressourceneffizient wird der Entwurf einer Schaltung dann bezeichnet, wenn durch die Einstellung aller durch dem Entwickler zugänglichen Parameter ein optimaler Kompromiss zwischen allen betrachteten Schalteigenschaften bzw. Ressourcen entsteht.

Der Entwurf ressourceneffizienter integrierter Schaltungen setzt einen Entwurf ressourceneffizienter Standardzellen voraus. Der Entwickler hat beim Entwurf solcher Standardzellen Freiheitsgrade in den Weiten und Längen der Transistoren, aus denen ein Grundgatter aufgebaut ist. Für die Festsetzung dieser Transistordimensionen gibt es einen klassischen Ansatz, der auf eine symmetrische Übergangskennline zwischen Ein- und Ausgangssignal abzielt. Die Schalteigenschaften des Gatters werden auf diese Weise indirekt festgelegt.

Optimale Kompromisse zwischen den Schalteigenschaften können aber nur dann gefunden werden, wenn diese in direktem Fokus der Suche stehen. Mathematisch betrachtet entspricht die Entwurfsraumexploration daher dem Lösen eines Mehrzieloptimierungsproblems (MOP), bei dem gleich mehrere Zielfunktionen simultan minimiert werden sollen. Die Lösung eines MOP ist eine Pareto-Menge, die alle optimalen Kompromisse dieser Zielfunktionen beinhaltet.

Aufgabe und Zielsetzung dieser Arbeit ist die Entwicklung eines Verfahrens zum Entwurf ressourceneffizienter Schaltungen, das - basierend auf dem Mehrzieloptimierungsansatz - Bausteine integrierter Schaltungen optimiert. Diese Bausteine sollen vor allem CMOS-Standardzellen sein, aus denen ressourceneffiziente digitale Schaltungen gefertigt werden können, außerdem Speicherelemente und einfache analoge Schaltungen.

Überblick über diese Arbeit

Diese Arbeit bringt mit der algorithmisch gesteuerten Entwurfsraumexploration integrierter Schaltungen einen Bereich der Elektrotechnik mit der Angewandten Mathematik zusammen. Entsprechend notwendig ist eine ausführliche Klärung aller weiterhin benutzten Fachbegriffe und Definitionen, die zum einen aus dem Ingenieurvokabular und andererseits aus der Mathematik stammen. Diese leitet das erste Kapitel ein. Weiterhin wird der bereits oben erwähnte klassische Ansatz zur Dimensionierung einfacher CMOS-Logikgatter ausführlicher erklärt und der Mehrzieloptimierungsansatz, der die Anwendung von Algorithmen zur Lösung erlaubt, als zielführendere Alternative vorgestellt.

Im zweiten Kapitel werden zunächst mengenorientierte Algorithmen vorgestellt, die eine Pareto-Menge durch Unterteilung des Suchraums in immer kleinere Boxen approximieren. Am Beispiel des CMOS-Inverters veranschaulicht ein Vergleich zwischen der algorithmisch bestimmten Lösung und einer Handrechnung die Arbeitsweise dieses Algorithmus. Außerdem werden Ergebnisse weiterer Standardzellen in 65 nm und 90 nm Technologien kommerziellen Entwürfen gegenüber gestellt, um die Ergebnisse der Algorithmen zu bewerten. Dokumentiert ist außerdem wie der Algorithmus parallelisiert werden kann.

Nachdem der Mehrzieloptimierungsansatz für Standardzellen betrieben bei voller Versorgungsspannung eingeführt und dessen Vorteile aufgezeigt wurden, wird dieser im dritten Kapitel zur Dimensionierung einer 57 Zellen umfassenden Bibliothek verwandt. Aus dieser können Schaltungen synthetisiert werden, die mit einer Versorgungsspannung unterhalb der Transistoren-Schwellspannung (Subschwellbereich) ressourceneffizient funktionieren. In diesem Kapitel werden auch evolutionäre Algorithmen vorgestellt, mit deren Hilfe sich schnell erste Erkenntnisse über die Lage einer Pareto-Menge im Suchraum sammeln lassen.

Die Optimierung von Speicher ist Inhalt des vierten Kapitels. Eine SRAM-Zelle wird dimensioniert, so dass sie sowohl bei voller Versorgungsspannung als auch im Subschwellbereich ressourceneffizient betrieben werden kann. Es handelt sich dabei um eine 9-Transistor-Zelle, die durch zusätzliche Auslese-Transistoren aus der klassischen 6- Transistor-Zelle entsteht, um einem möglichen Informationsverlust beim Auslesen entgegenzuwirken.

Am Beispiel eines Differenzverstärkers mit Eintaktausgang wird die Anwendbarkeit der betrachteten Algorithmentypen in Kapitel 5 untersucht. Festgestellt wird, dass analoge Schaltungen in ihren Schalteigenschaften sehr sensibel sind und bei der Entwurfsraumexploration auch die Robustheit der Entwurfspunkte bewertet werden muss. Für die Algorithmen wäre eine zusätzliche Sensitivitätsanalyse wünschenswert.

Vor der Zusammenfassung werden im letzten Kapitel die mengenorientierten mit den evolutionären Algorithmen für die Anwendung der Schaltungsoptimierung verglichen.

Verwandte Arbeiten zur Schaltungsoptimierung

Es gibt einige Arbeiten zur Optimierung digitaler Schaltungen auf Transistorebene. Diensthuhl befasste sich in seiner Dissertation [20] mit der Optimierung von Registern und Speicher. Untersucht wurden Methoden der Computational Intelligence für den Entwurf von 90 nm CMOS-Speicherelementen. Anders als dort, wo evolutionäre Mehrzieloptimierungsalgorithmen eingesetzt wurden, findet man häufiger Einzieloptimierungsansätze. Fisher et al. [23] haben Algorithmen zur Optimierung von Standardzellen nach vorgegeben Schalteigenschaften entworfen. Dabei werden Zielgrößen wie Verlustleistung, Verzögerungszeit und Fläche in mehreren Einzieloptimierungen getrennt voneinander betrachtet. Bei Borah et al. [8] wird mit einem Algorithmus die Verlustleistung unter Vorgabe einer oberen Schranke auf Zeit minimiert. Sie benutzen analytische Modelle, die durch SPICE Simulationen verifiziert wurden. Lin et al. [47] präsentierten einen heuristischen Auswahlalgorithmus, der basierend auf einer gewichteten Summe Verzögerungszeit und Fläche von gegebenen Standardzellen minimiert.

In weiteren Arbeiten zur Optimierung digitaler Schaltungen werden Optimierungsansätze auf höheren Abstraktionsebenen eingesetzt. Eikerling verfolgt in seiner Dissertation [22] einen automatenbasierten Mehrzielansatz zur Optimierung digitaler Schaltungen. Dabei werden im Entwurfsprozesses von der Hardware-Beschreibung bis zur Register-Transfer-Ebene einzelne Blöcke hierarchisch neu synthetisiert. Hoppe et al. [36] optimierten High-Speed CMOS-Logik auf Gatterebene. Sie entwarfen einen Algorithmus, der basierend auf einfachen analytischen Modellen (wellbehaved Macromodells) mehrere Gattereigenschaften gleichzeitig optimiert. Die Weiten der PMOS- und NMOS-Transistoren W_p und W_n sind dabei innerhalb einer Schaltung für alle Transistoren gleich. Außerdem ist $W_\mathrm{p}/W_\mathrm{n}$ fest. Variabel ist nur die Summe aller Weiten in einem Gatter.

Mehrzieloptimierungsverfahren für analoge Schaltungen gibt es beispielweise bei Müller et al. [54]. Rutenbar et al. [59] geben einen Überblick über Algorithmen zur Optimierung von komplexeren analogen Schaltungen für mehr als 10000 Transistoren durch hierarchische Modellierung. Graeb [31] führt die Anforderungen, die durch die besondere Sensitivität analoger Schaltungen gegeben sind, weiter aus und beschreibt Verfahren, um Dimensionierungsprobleme solcher Schaltungen zu lösen.

Einen Überblick über die Optimierungstechniken für die Anwendung der Dimensionierung integrierter Schaltungen gaben Brayton et al [9]. Sie untersuchten bereits Mehrzielansätze für den nominellen Entwurf, stellten aber auch statistische Methoden vor, die vor allem die Ausbeute einer Schaltung Störparameterschwankungen berücksichtigten.

In dieser Arbeit werden zwei Typen von Mehrzieloptimierungsalgorithmen verwandt. Das sind zum einen die deterministischen Unterteilungsalgorithmen, die den Suchraum einer Schaltung sukzessiv in immer kleiner Teile unterteilen, um dort nach optimalen Schalteigenschaften zu suchen. Zum anderen werden die Einsatzmöglichkeiten von evolutionären Algorithmen untersucht, die in vielen Ingenieursdisziplinen Anwendung finden, da sie durch relativ einfache Regeln auch einen komplexeren Suchraum beherrschen können.

Die Unterteilungstechniken wurden vorgestellt von Dellnitz, Schütze et al. [18, 60, 61]. Sie sind Teil der am Paderborner Lehrstuhl für Angewandte Mathematik entwickelten Toolbox *Global Analysis of Invariant Objects* (GAIO).

Die verwandten evolutionären Algorithmen gründen auf dem von Zitzler und Thiele eingeführten *Strength Pareto Evolutionary Algorithm* (*SPEA*). In [82] führten sie die Vorteile ihrer *Fitnessbewertung* aus. Andere Teile wurden bei Deb [15, 17] eingeführt. Bei Mostaghim [53] findet der SPEA Anwendung beim Entwurf von Antennen und zur Optimierung von chemischen Prozessen. Zur Optimierung integrierter Schaltungen nutzte Thomas [71] den SPEA am Beispiel von Addierern.

Kapitel 1

Entwurfsraumexploration

1.1 Begriffsklärungen und Definitionen

1.1.1 Entwurfsraumexploration

Der Entwurf einer integrierten Schaltung beginnt mit der Spezifikation ihres Verhaltens. Das *Verhalten* einer Schaltung beschreibt Signaländerungen der Ausgänge abhängig von den internen Zuständen und Signaländerungen an möglichen Eingängen. Die Spezifikation des Verhaltens allein lässt noch viele Freiheitsgrade, die der Entwickler während des Entwurfs festlegt. Die Menge dieser technischen Parameter heißt *Entwurfsraum*.

Der Entwurfsraum wird während des Entwurfsprozesses aus verschiedenen Blickwinkeln betrachtet und die einzelnen Parameter lassen sich unterschiedlichen Entwurfsebenen zuordnen (s. Abbildung 1.1). Die *Struktur* einer Schaltung beschreibt, welche Bauelemente verwendet werden und wie diese verbunden sind. Die *Geometrie* einer Schaltung wird bestimmt durch die Anordnung und Dimensionierung der Komponenten.

Den Sichten des Entwurfsraumes lassen sich beispielhaft einige Parameter des Entwurfsraumes zuordnen:

- Struktursicht: Anzahl der Prozessoren, Anzahl an Hardware-Beschleunigern, Menge des Speichers direkt am Prozessor, Breite der Network-on-Chip-Verbindungen, Anzahl der Verbindungen pro Network-on-Chip-Knoten, Instruktionssatz-Größe, Anzahl der Pipeline-Stufen der Prozessoren, Arbitrationsalgorithmus des Cluster-Busses, Größe der Eingangs-Puffer pro Netzwerk-Port

- Geometriesicht: Geometrie-Parameter der Transistoren wie Längen und Weiten des Gates, Gateoxiddicke, Wahl der Technologie

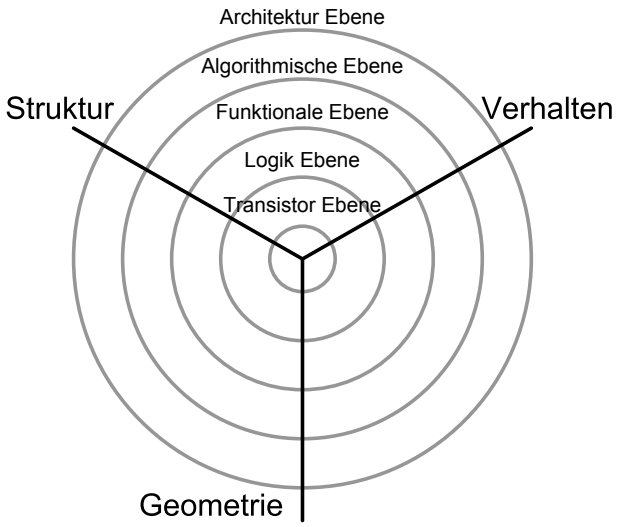


Abbildung 1.1: Y-Diagramm: Sichten des Entwurfsraums nach Gajski und Kuhn [26]

Entwurfsraumexploration beschreibt die systematische Suche nach optimalen, ressoureneffizienten (s.u.) Entwurfspunkten.

1.1.2 Ressourceneffizienz

Mit der Wahl eines Punktes im Entwurfsraum wird eine Schaltung vollständig festgelegt. Um eine Schaltung mit möglichst guten Schaltungseigenschaften zu konstruieren, wird der Entwurfsraum nach solchen Punkten durchsucht. Diese Suche nennt man *Entwurfsraumexploration*. Eine Schaltung mit optimalen Eigenschaften heißt *ressourceneffizient*. Die dem Entwerfer zur Verfügung stehenden *Ressourcen* sind Fläche des Chips, Energie, Zeit und Robustheit[1]. Um eine quantitative Bewertung der einzelnen Ressourcen zu ermöglichen, werden für diese *Ressourcenmaße* eingeführt.

Beispiele für Ressourcenmaße

- Zeit: Verzögerungszeit, Anstiegs- und Abfallszeit, Taktfrequenz
- Robustheit: Störabstand, Phasenreserve (Verstärker), statische Störabstände (SRAM)
- Fläche: Siliziumfläche, Sliceanzahl (FPGA)
- Energie: Verlust durch statische und dynamische Leckströme

Die folgend definierten Ressourcenmaße sind in der Schaltungstechnik vertraute Größen. Detailliertere Erläuterungen können in Standardliteratur wie beispielweise in [35] nachgelesen werden.

[1] Auch wenn sich vom germanistischen Standpunkt Robustheit schlecht unter den Oberbegriff Ressource fassen lässt, so soll hier zum Wohl eines vertrauten Vokabulars eine Ausnahme gemacht werden.

Schaltzeiten

Die Schaltzeiten sind ein Maß für die Schaltgeschwindigkeit eines Logikgatters. Sie sind definiert als die Länge des Zeitintervalls, das der Ausgang benötigt, um auf eine Änderung am Eingang zu reagieren (s. Abbildung 1.2).

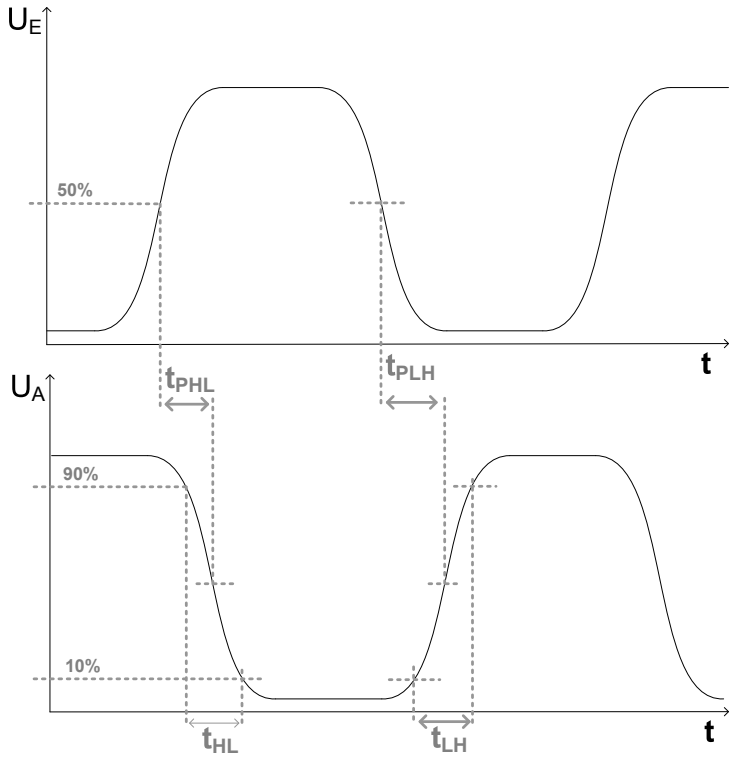


Abbildung 1.2: Spannungsverlauf des Ein- und Ausgangs (nach [41])

Die Abfallszeit wird mit t_{HL} und die Anstiegstzeit mit t_{LH} bezeichnet. Zusammenfassend wird

$$t_{\mathrm{rf}} = \max\left(t_{\mathrm{HL}}, t_{\mathrm{LH}}\right)$$

definiert. Die Verzögerungszeit wird bestimmt durch

$$t_{\mathrm{pd}} = \max\left(t_{\mathrm{PLH}}, t_{\mathrm{PHL}}\right). \tag{1.1}$$

Verlustleistung

Eine integrierte Schaltung verbraucht sowohl im Ruhezustand als auch während des Schaltvorgangs Leistung.

$$P = P_{\mathrm{stat}} + P_{\mathrm{dyn}}$$

Die statische Verlustleistung wird verursacht durch Ruheströme (Quer- und Leckströme).

$$P_{\mathrm{stat}} = P_{\mathrm{quer}} + P_{\mathrm{leck}}$$

Die dynamische Verlustleistung entsteht aufgrund von Schaltvorgängen. Sie setzt sich zusammen aus Lastströmen und Querströmen, die beim Umschalten entstehen (s. Abbildung 1.3).

$$P_{\text{dyn}} = P_{\text{last}} + P_{\text{schalt}}$$

Die dynamische Verlustenergie E_{dyn} ergibt sich dann aus

$$E_{\text{dyn}} = \int_T P_{\text{dyn}}(t)dt,$$

wobei T die Dauer des Schaltvorgangs sei.

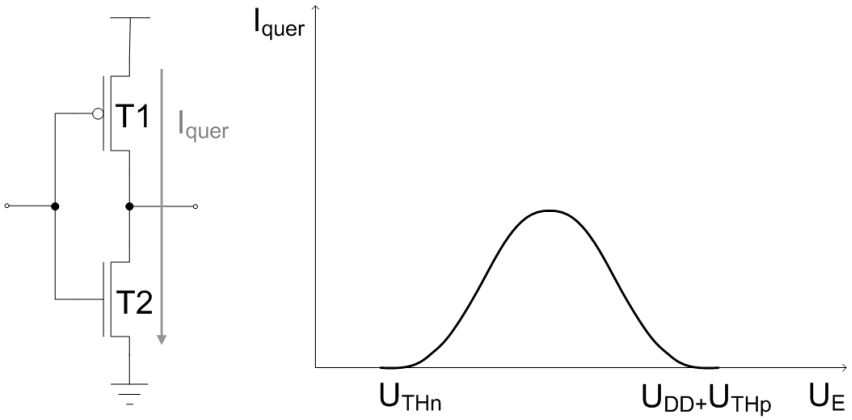


Abbildung 1.3: I_{quer} ist abhängig von der Eingangsspannung. (In Anlehnung an [41])

Für CMOS-Technologien bis zu 90nm ist die Verlustleistung

$$P_{\text{last}} = \alpha_{0\to1} \cdot f \cdot C_L \cdot U_{\text{dd}}^2, \tag{1.2}$$

zum Umladen der Kapazitäten die dominierende Verlustleistung. Die während des Schaltvorgangs verlorene Leistung P_{schalt} hat einen sehr geringen Anteil am gesamten Verlust. Die Querströme I_{quer} sind in der CMOS-Technologie zu vernachlässigen. Für Technologien unter 90nm gewinnen die Leckströme I_{leck} an Gewicht. In der 65nm Technologie fließen verhältnismäßig große Gate- und Subschwellströme. (C_{L} bezeichnet die Ausgangslast, f die Eingangsfrequenz und die Schalthäufigkeit ist mit $\alpha_{0\to1}$ bezeichnet.)

Auch am Eingang tritt während des Umschaltvorgangs ein Stromfluss auf, um die parasitäre Eingangskapazität C_{in} umzuladen. Die Eingangskapazität setzt sich aus einzelnen Kapazitäten zusammen, die an den Transitoren zwischen den Gates und ihrer Umgebung gebildet werden (s. Abbildung 1.4). Die Eingangsverlustenergie E_{in} ergibt sich aus

$$E_{\text{in}} = U_{\text{dd}} \int_T I_{\text{in}}(t)dt.$$

Sie ist außerdem eine Größe, mit der C_{in} abgeschätzt werden kann. Der Vorteil, diese nicht direkt zu berechnen, ist die Einsparung einer zusätzlichen Schaltungssimulation.

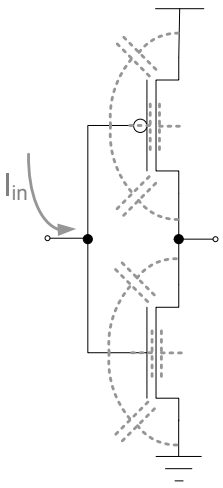


Abbildung 1.4: Beim Schaltvorgang werden durch den Eingangsstrom I_{in} Kapazitäten umgeladen. (In Anlehnung an [41])

Störabstände

Die Störabstände sind ein Maß für die Schaltzuverlässigkeit eines Logikgatters. Das Gatter ist gegenüber Störungen umso unempfindlicher je weiter U_{OH} über U_{IH} und U_{OL} unter U_{IL} liegt. Als Störabstand NM ist definiert:

$$\text{NM} = \min\left(\text{NM}_{\text{H}}, \text{NM}_{\text{L}}\right) \tag{1.3}$$

$$\begin{aligned} \text{NM}_{\text{H}} &= |U_{\text{OH}} - U_{\text{IH}}| \\ \text{NM}_{\text{L}} &= |U_{\text{IL}} - U_{\text{OL}}| \end{aligned}$$

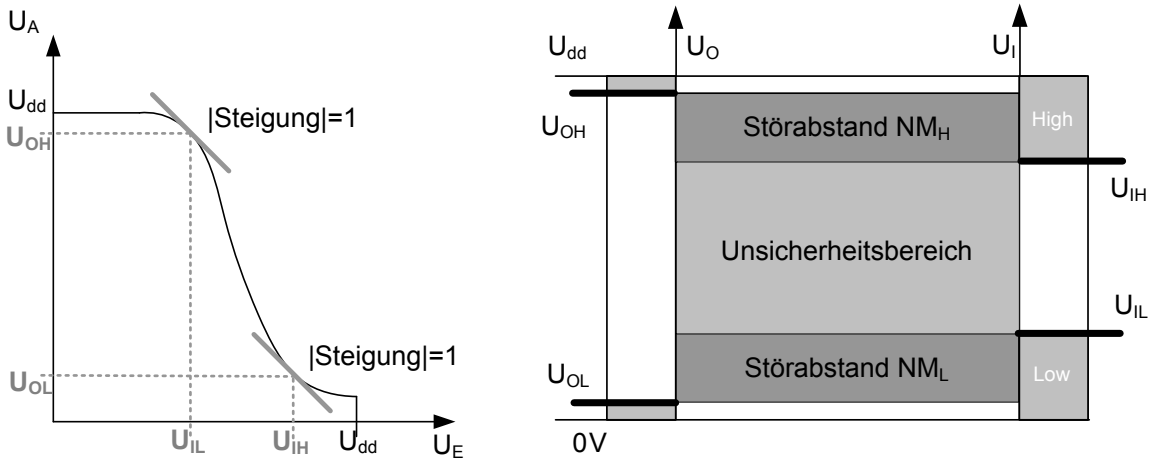


Abbildung 1.5: Störabstände (nach [41])

Die Besonderheit dieses Ressourcenmaßes besteht darin, dass ein möglichst großer Wert erzielt werden soll. Bei der Übersetzung in ein Mehrzieloptimierungsproblem wird daher $-$NM minimiert.

1.1.3 Mehrzieloptimierung / Pareto-Menge

Mit Hilfe der Ressourcenmaße läßt sich die zielgerichtete Entwurfsraumexploration als Optimierungsaufgabe formulieren. Im mathematischen Wortgebrauch werden aus den Ressourcenmaßen *Zielfunktionen*. Da es mehrere Ressourcenmaße gibt, entsteht eine *Mehrzieloptimierungsaufgabe* (1.4), deren Lösungsmenge die *Pareto-Menge* ist.

Definition Dominanz „$\prec$“ im $\mathbb{R}^m$
Für $a, b \in \mathbb{R}^m$ gilt $a \prec b$ (a dominiert b), falls $a_i \leq b_i$ für alle $i = 1..m$ und $a \neq b$.

Bemerkung Mit der Dominanz „$\prec$“ ist eine Halbordnung auf $\mathbb{R}^m$ eingeführt, mit der sich das Minimum einer vektorwertigen Funktion definieren lässt.

Definition Pareto-Menge, Pareto-Punkt und Pareto-Front
Sei $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ eine Funktion. Für das Mehrzieloptimierungsproblem

$$\min_{x \in S} f(x) \tag{1.4}$$

besteht die Pareto-Menge P aus allen Punkten $p \in S$, für die kein $x \in S$ existiert mit $f(x) \prec f(p)$:

$$P = \{p \in S | \nexists\, x \in S : f(x) \prec f(p)\}\,.$$

Jedes Element der Pareto-Menge heißt Pareto-Punkt. Mit Pareto-Front sei das Bild der Pareto-Menge f(P) bezeichnet.

Bemerkung Die Pareto-Menge hängt sowohl von der Funktion f als auch von der Menge S ab, die im Folgenden als *Suchraum* bezeichnet wird. Anstatt ausführlich $f(x)$ dominiert $f(p)$ zu schreiben wird in der Literatur vereinfacht nur x dominiert p geschrieben. Dann liegt es beim Leser zu erkennen, dass die beiden Punkte zwar Elemente des Suchraums sind, aber in ihren Funktionseigenschaften verglichen werden. Alternativ ist häufig auch „$\prec$“ schon unter Einbezug der Zielfunktionen definiert.

Definition Schwacher Pareto-Punkt
Ein Punkt $x \in \mathbb{R}^n$ heißt schwacher Pareto-Punkt bzgl. (1.4), falls kein $y \in \mathbb{R}^n$ existiert, so dass $f_i(y) < f_i(x)$ für alle $i = 1..m$.

Es können im Suchraum Punkte existieren, die keine Pareto-Punkte sind, sich aber in ihrer direkten Umgebung nicht in allen Zielfunktionen gleichzeitig verbessern lassen.

Definition Substationärer Punkt
Ein Punkt $p \in S$ heißt substationär bezüglich der Funktion $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$, falls Skalare α_1, ...,$\alpha_m \geq 0$ existieren, so dass

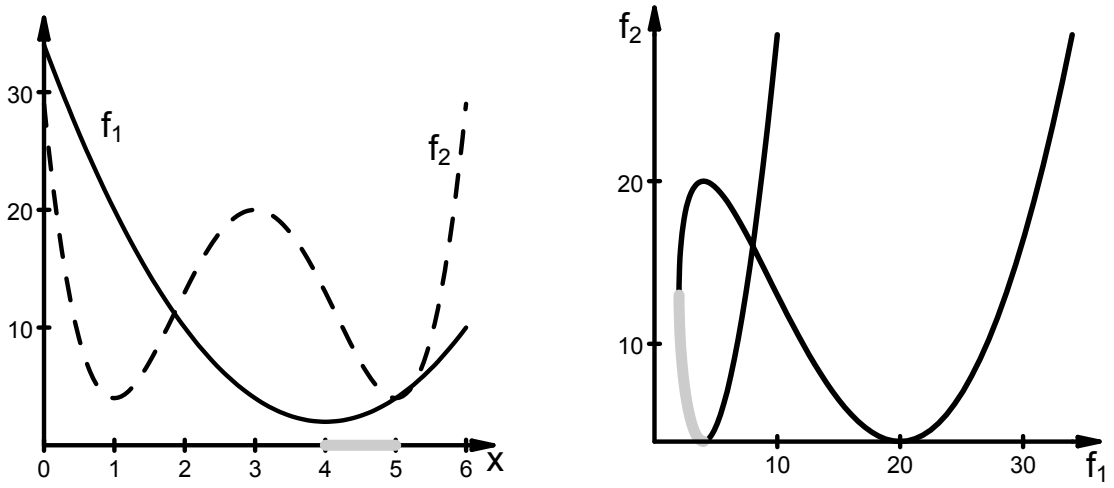
$$\textstyle\sum_{i=1}^{m} \alpha_i = 1 \quad \text{und} \quad \sum_{i=1}^{m} \alpha_i \nabla f_i(p) = 0. \tag{1.5}$$

Pareto-Punkte sind notwendigerweise substationärere Punkte (vgl. Theorem 2.4 in [18]). Die Umkehrung gilt nicht, wie das folgende Beispiel beweist.

Beispiel 1 Pareto-Menge und substationäre Punkte
Gegeben sei $f : [0, 6] \to \mathbb{R}^2$ mit

$$\begin{aligned} f_1(x) &= 2x^2 - 16x + 34 \\ f_2(x) &= x^4 - 12x^3 + 46x^2 - 60x + 29 \end{aligned}$$

(s. Abbildung 1.6). Dann ist die Pareto-Menge des Mehrzieloptimierungsproblems (1.4) das Intervall $[4, 5]$. Die Menge der substationären Punkte ist $[1, 3] \cup [4, 5]$, denn über dem Intervall $[1, 3]$ fällt f_1 während f_2 steigt und über dem Intervall $[4, 5]$ fällt f_2 während f_1 steigt.



(a) Pareto-Menge (grau)

(b) Pareto-Front (grau)

Abbildung 1.6: Pareto-Menge und ihr Bild (Pareto-Front)

Beispiel 2 Substationäre Punkte im zwei-dimensionalen Suchraum
In dieser Arbeit werden häufig Beispiele in einem zwei-dimensionalen Suchraum betrachtet. Abbildung 1.7 zeigt die Höhenlinien der Funktionen

$$\begin{aligned} f_1(x_1, x_2) &= (x_1 - 2)^2 + (x_2 - 1)^2 \\ f_2(x_1, x_2) &= x_1 + x_2 \end{aligned}$$

in hellgrau (f_1) und grau (f_2) in einem Ausschnitt des ersten Quadranten. Außerdem ist die Menge der substationären Punkte als schwarze Linie eingezeichnet.

Es ist

$$\begin{aligned} \nabla f_1(x_1, x_2) &= (2x_1 - 4, 2x_2 - 2) \\ \nabla f_2(x_1, x_2) &= (1, 1)\,. \end{aligned}$$

Nach (1.5) muss gelten

$$\begin{aligned} \alpha_1 (2x_1 - 4) + \alpha_2 &= 0 \\ \alpha_1 (2x_2 - 2) + \alpha_2 &= 0 \\ \alpha_1 + \alpha_2 &= 1, \end{aligned}$$

was eingesetzt und vereinfacht

$$\begin{aligned} \alpha_1 (2x_1 - 5) + 1 &= 0 \\ \alpha_1 (2x_2 - 3) + 1 &= 0 \end{aligned} \tag{1.6}$$

ergibt. Erweitert und subtrahiert erhält man

$$x_2 = x_1 - 1.$$

Da $0 \leq \alpha_2 = 1 - \alpha_1 \Rightarrow \alpha_1 \leq 1$ muss x_1 nach (1.6) kleiner sein als 2. Die Menge der substationären Punkte ist daher

$$\left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} | x_2 = x_1 - 1, x_1 \leq 2 \right\}.$$

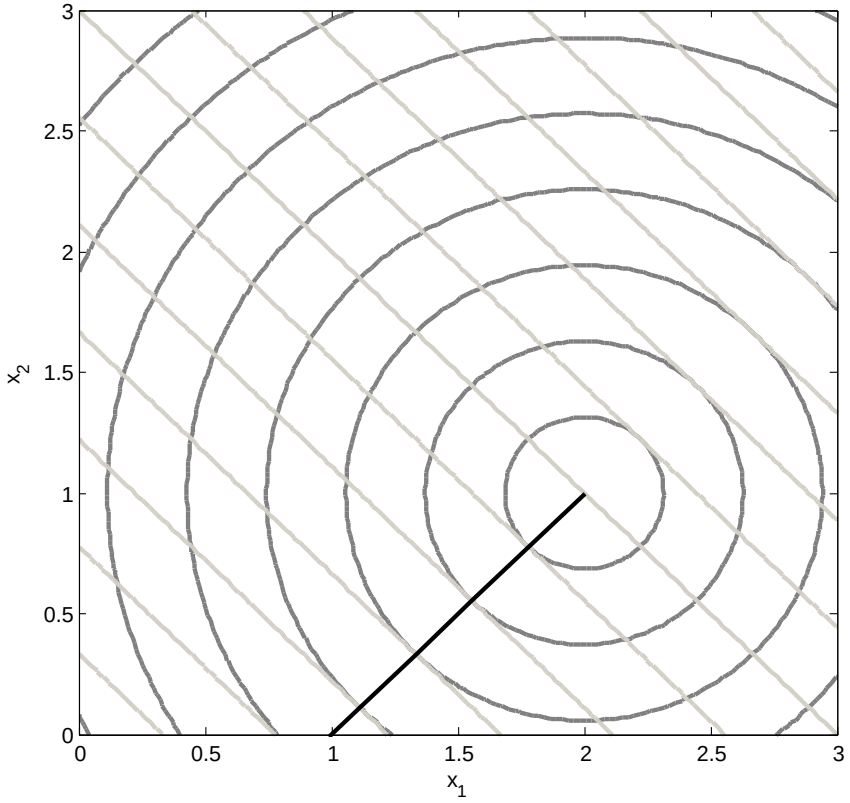


Abbildung 1.7: Höhenlinien (hellgrau und grau), Menge der substationären Punkte (schwarz)

Unter Berücksichtigung der Monotonie von f_1 und f_2 lassen sich die substationären Punkte auch grafisch aus Abbildung 1.7 ermitteln als die Punkte, an denen sich die Höhenlinien beider Funktionen berühren. Im zwei-dimensionalen Fall ist (1.5) genau dann erfüllt, wenn die Gradienten beider Funktionen in einem Winkel π zueinander stehen. Im Berührpunkt zweier Höhenlinien beträgt der Winkel entweder 0 oder π , da der Gradient einer Funktion

im rechten Winkel zu seiner Höhenlinie steht: Es sei $g : \mathbb{R} \to \mathbb{R}^n$ die Höhenlinie einer Funktion $f : \mathbb{R}^n \to \mathbb{R}$. Mit $f(g(t)) = c$ (konstant) gilt $\frac{d}{dt} f(g(t)) = 0$ und damit

$$0 = \frac{d}{dt} f(g(t)) = \nabla f(g(t)) g'(t) = \left\langle \nabla f(g(t))^T, g'(t) \right\rangle.$$

($\langle ., . \rangle$ bezeichne das euklidische Skalarprodukt des $\mathbb{R}^n$ und $\nabla f(g(t))^T$ die Transponierte zu $\nabla f(g(t))$.) Kennt man also die grobe Orientierung der Gradienten, lässt sich die Lage der substationären Punkte leicht ausmachen. Mit Hilfe der Höhenlinienwerte kann durch einen globalen Vergleich die Pareto-Menge aus den substationären Punkten extrahiert werden. In diesem Beispiel sind beide Mengen gleich.

Definition Kostenfunktion
$g : S \to \mathbb{R}$ heißt Kostenfunktion zu (1.4), falls für jede Lösung x^* von

$$\min_{x \in S} g(x) \tag{1.7}$$

gilt $x^* \in P$, wobei P die Pareto-Menge zu (1.4) sei.

Beispiel Gewichtete Summe
Es sei

$$g(x) = \sum_{i=1}^{m} \omega_i f_i(x) \tag{1.8}$$

mit $f_i : S \to \mathbb{R}$ (skalarwertige Einzelkomponente aus (1.4)) und $\omega_i > 0$ für alle $i = 1..m$. Dann ist g eine Kostenfunktion.

Beweis: Es sei g eine gewichtete Summe (1.8) eines MOP (1.4). Angenommen g wäre keine Kostenfunktion, dann existiert ein x^*, das (1.7) erfüllt, aber nicht Pareto-optimal ist. Da $x^* \in S \setminus P$ existiert ein $y^* \in P \subseteq S$ mit $f(y^*) \prec f(x^*)$. Daraus folgt $g(y^*) < g(x^*)$, was nicht sein kann, da x^* als Lösung von (1.7) angenommen wurde.

Bemerkung

1. Durch die Reduktion eines MOP auf ein skalares Optimierungsproblem (SOP) mittels einer Kostenfunktion wird die Lösung von der Pareto-Menge auf einen einzelnen Punkt reduziert. Dies vereinfacht die Suche und damit die Laufzeit des Algorithmus. Das Ergebnis vermittelt allerdings keinen Aufschluss über die Menge aller optimalen Kompromisse.

2. Die gewichtete Summe ist die verbreitetste Methode, um ein MOP auf ein SOP zu reduzieren. Eine weitere Möglichkeit ist die Verknüpfung als gewichtetes Produkt. Nach Definition gilt, dass jede Lösung eines Minimierungsproblems einer gegebenen Kostenfunktion (1.7) Pareto-optimal ist. Die Umkehrung gilt im Allgemeinen nicht. Abbildung 1.8 zeigt eine nicht konvexe Pareto-Front eines MOP mit zwei Zielfunktionen. Die Höhenlinie einer gewichteten Summe ist eine Gerade, deren Steigung durch die Gewichte ω_i bestimmt wird. Die Minimierung der Kostenfunktion lässt sich grafisch als Abstieg über die parallelen Höhenlinien verfolgen. Der Pareto-Punkt x' kann von keiner gewichteten Summe bestimmt werden[2], da $g(x^*) < g(x')$.

[2]Es wird angenommen, dass ein Suchalgorithmus immer globale Minima findet.

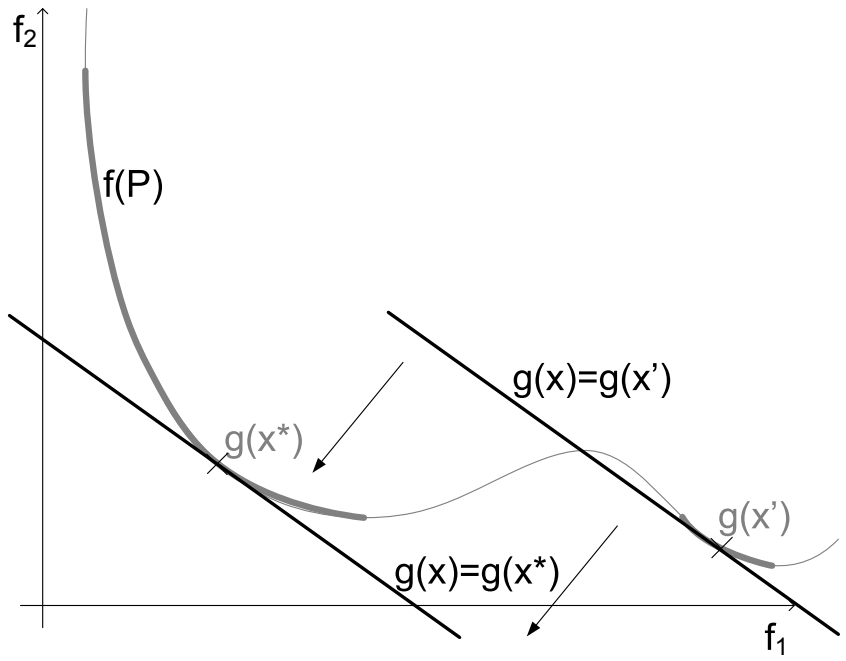


Abbildung 1.8: Optimierungsbeispiel mit zwei Zielfunktionen. Mit der gewichteten Summe g kann nur ein einzelner Pareto-Punkt x^* gefunden werden. Die genaue Lage und Form der gesamten Lösungsmenge bleibt unbestimmt. Speziell der Pareto-Punkt x' kann durch den Ansatz der gewichteten Summe auch nicht mit anderen Gewichten gefunden werden.

1.2 Der Klassische Ansatz zur Dimensionierung von CMOS-Logikgattern

Lehrbücher [2, 35] fordern die Dimensionierung der Transistoren eines CMOS-Inverters so zu wählen, dass sich eine symmetrische DC-Ausgangskennline ergibt. Dies soll erreicht werden, wenn $U_{\mathrm{IL}} = U_{\mathrm{dd}} - U_{\mathrm{IH}}$ und $U_{\mathrm{OL}} = U_{\mathrm{dd}} - U_{\mathrm{OH}}$ gilt [35]. Für einen vergleichbaren Ansatz definiert Baker [2] einen Schaltpunkt V_{SP}, für den gilt $U_{\mathrm{A}}(U_{\mathrm{E}}) = U_{\mathrm{E}}$ (s. Abbildung 1.9). Die Kennlinie ist dann symmetrisch, wenn V_{SP} bei der halben Versorgungsspannung liegt.

Wird der Kanallängeneffekt vernachlässigt, so ergibt sich aus der Gleichheit der Ströme durch N- und PMOSFET [3]

$$I_{\mathrm{DSn}} = \frac{B_n}{2}\frac{W_n}{L_n}\left(V_{\mathrm{SP}} - U_{\mathrm{THn}}\right)^2 = \frac{B_p}{2}\frac{W_p}{L_p}\left(V_{\mathrm{SP}} - U_{\mathrm{dd}} - U_{\mathrm{THp}}\right)^2 = -I_{\mathrm{DSp}}.$$

Fordert man mittels

$$V_{\mathrm{SP}} = \frac{U_{\mathrm{dd}}}{2} \tag{1.9}$$

[3] Für den Drain-Source-Strom des NMOSFETs gelte

$$I_{\mathrm{DSn}} = \left\{ \begin{array}{ll} 0 & ;U_{\mathrm{GSn}} < U_{\mathrm{THn}} \\ B_n \frac{W_n}{L_n}\left(U_{\mathrm{GSn}} - U_{\mathrm{THn}} - \frac{U_{\mathrm{DSn}}}{2}\right) U_{\mathrm{DSn}} & ;U_{\mathrm{DSn}} \leq U_{\mathrm{GSn}} - U_{\mathrm{THn}} \\ \frac{B_n}{2}\frac{W_n}{L_n}\left(U_{\mathrm{GSn}} - U_{\mathrm{THn}}\right)^2 & ;U_{\mathrm{DSn}} > U_{\mathrm{GSn}} - U_{\mathrm{THn}} \end{array} \right\}$$

mit $B_n = \mu_n c_{ox}$. Der Drain-Source-Strom des PMOSFETs I_{DSp} sei analog definiert.

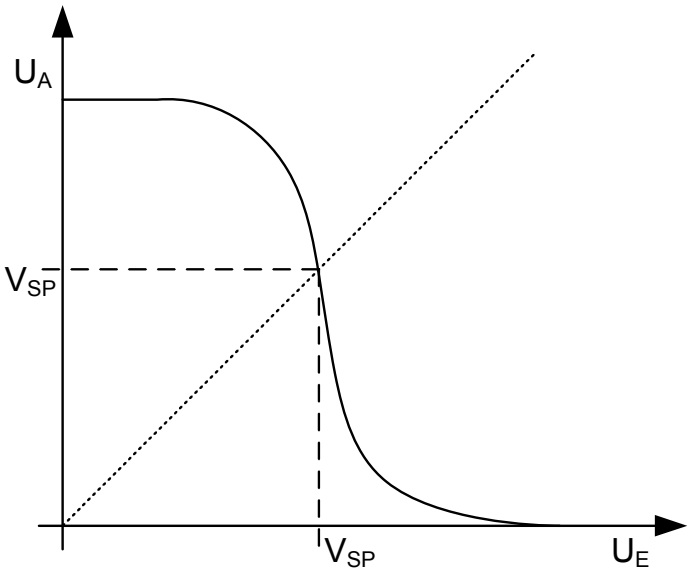


Abbildung 1.9: Ausgangskennlinie mit Schaltpunkt V_{SP} (vgl. [7])

eine symmetrische Ausgangkennlinie, so ergibt sich

$$\frac{B_n \frac{W_n}{L_n}}{B_p \frac{W_p}{L_p}} = \left(\frac{U_{\mathrm{dd}} + 2U_{\mathrm{THp}}}{U_{\mathrm{dd}} - 2U_{\mathrm{THn}}}\right)^2. \tag{1.10}$$

Um die Fläche und damit die Eingangskapazitäten der Schaltung zu minimieren, werden die Längen L_n und L_p minimal gewählt. Unter der Annahme $U_{\mathrm{THp}} \approx -U_{\mathrm{THn}}$ wird (1.10) durch 1 abgeschätzt. So ergibt sich ein Verhältnis von

$$W_p \approx \frac{B_p}{B_n} W_n = \frac{\mu_p}{\mu_n} W_n. \tag{1.11}$$

Die Weite des PMOSFET sollte also als ein Vielfaches der Weite des NMOSFETs gewählt werden, um eine symmetrische Ausgangskennline zu erhalten. Die Ladungsträgerbeweglichkeiten μ_n und μ_p bestimmen den Proportionalitätsfaktor.

Komplexere CMOS-Logikgatter können auf gleiche Weise behandelt werden. Um der obigen Rechnung folgen zu können, müssen jeweils alle NMOSFET und alle PMOSFET zusammengefasst werden, indem entweder ihre Längen (Reihenschaltung) oder ihre Weiten (Parallelschaltung) addiert werden. Diese Vereinfachung ist für jede CMOS-Logik möglich. Beispielhaft ergibt sich für ein NAND-Gatter mit k Eingängen eine Dimensionierung von

$$W_p \approx k^2 \frac{B_p}{B_n} W_n = k^2 \frac{\mu_p}{\mu_n} W_n. \tag{1.12}$$

Für kleine Strukturgrößen ist (1.12) nur noch eine sehr grobe Abschätzung, da Effekte vernachlässigt werden, die in Nanotechnologien immer stärkeren Einfluss haben. Außerdem wird hier vorausgesetzt, dass alle Eingänge synchron schalten, was mitunter nicht beabsichtigt wird. Unterstützt durch Simulationsergebnisse führt der Ansatz (1.9) allerdings

weiterhin zu einer annähernd symmetrischen Ausgangskennlinie. Eine solche verspricht große Störabstände.

Mit der oben aufgezeigten Methode muss der Entwickler einer CMOS-Logik-Schaltung nur noch eine der Transitorweiten setzen, um damit die gesamten Schaltungsmaße festzulegen. Mit dem letzten Freiheitsgrad bestimmt man die Treiberstärke des Gatters. Unter der Vorgabe verschiedener Treiberstärken kann so eine ganze Standardzellenbibliothek von Logikgattern dimensioniert werden. Der Aufbau kommerzieller Bibliotheken lässt vermuten, dass sie eben auf diese Weise entworfen und nur durch leichte Veränderungen an den Standardzellenrahmen angepasst werden. In Tabelle 1.1 ist zu erkennen, dass die Weiten der Transistoren so gewählt wurden, dass (1.9) fast erfüllt ist. Klassischerweise

	W_p [µm]	W_n [µm]	$W_p/\ W_n$	V_{SP} $[V]$
INV65_X1	0.28	0.2	1.4	0.49 U_{dd}
INV65_X2	0.55	0.39	1.41	0.49 U_{dd}
INV65_X3	0.83	0.58	1.43	0.49 U_{dd}
INV65_X4	1.1	0.78	1.41	0.49 U_{dd}

Tabelle 1.1: 65nm CMOS-Inverter einer kommerziellen Standardzellenbibliothek

wird CMOS-Logik also in drei Schritten dimensioniert:

1. Setze alle Transistorlängen auf ihr Minimalmaß
2. Wähle alle W_p als konstante Vielfache von W_n, sodass (1.9) erfüllt ist. (Eine Abschätzung ist mit (1.11) gegeben.)
3. Setze W_p oder W_n so fest, dass eine vorgegebene Treiberstärke erreicht wird.

Dieses Vorgehen reduziert die Suche auf ein Entscheidungsproblem. Es beschränkt die Sicht aber auf nur zwei Schalteigenschaften, die auch nur indirekt berücksichtigt werden. Die mehrkriterielle Optimierung betrachtet weitere Schalteigenschaften. Sie durchsucht den gesamten Entwurfsraum und liefert eine ganze Menge von alternativen Dimensionierungen, die neben dem Störabstand auch Schaltzeiten und Verlustleistungen optimiert.

1.3 Algorithmen zur Schaltungsoptimierung

Die Aufgabenstellung der Schaltungsoptimierung gliedert sich in zwei wesentliche Teilbereiche, einen schaltungstechnischen und einen algorithmischen Bereich. Im schaltungstechnischen Teil wird zunächst zu jeder Schaltung entschieden, welche Ressourcen in die Optimierung eingehen sollen. Dabei ist zu untersuchen, von welchen Parametern die Maße dieser Ressourcen abhängen und welche davon durch den Schaltungsentwickler beeinflusst werden können. Für den Simulator werden die Netzlisten der Schaltungen

festgelegt. Entsprechend der zu messenden Ressourcen ist eine Testumgebung mit den benötigten Eingangssignalen zu definieren. Erst wenn es möglich ist, die Simulationen automatisiert anzustoßen und zu vermessen, ist eine Funktion definiert, die zu einem Satz von Entwurfsparametern die zugehörigen Werte der Ressourcenmaße liefert. Mit dieser Funktion und einem vorgegebenem Suchraum wird dann ein Mehrzieloptimierungsproblem festgelegt. Im algorithmischen Teil kann dann ein Algorithmus nach optimalen Entwurfspunkten des definierten MOP suchen. Die oben eingeführten MOPs fungieren somit als Schnittstelle zwischen Schaltungsentwurf und algorithmischer Optimierung. In Abbildung 1.10 sind diese Schritte zusammengefasst.

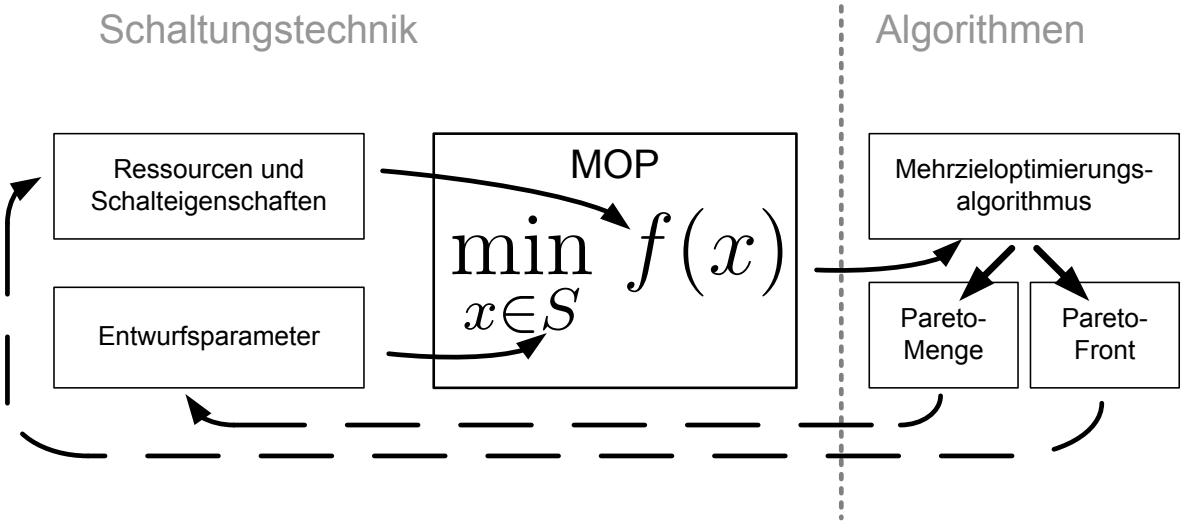


Abbildung 1.10: Die beiden Teilbereiche der Schaltungsoptimierung

Ebenfalls Aufgabe des Schaltungsentwicklers ist die Bewertung der durch die Algorithmen gelieferten Ergebnisse. Daher ist die Analyse der betrachteten Schaltungen in dieser Arbeit auch auf Basis von Handrechnungen mit einfachen Transistormodellen geschehen. Bevor in weiteren Kapiteln die Optimierung spezieller Schaltungen diskutiert wird, sollen an dieser Stelle zunächst die zwei verschieden Klassen von Algorithmen erwähnt werden, die zur Approximation der gesuchten Pareto-Mengen der Schaltungen genutzt und erweitert wurden. Eine ausführlichere Beschreibung dieser Algorithmen findet sich in den folgenden Kapiteln.

1.3.1 Evolutionäre Algorithmen

Evolutionäre Algorithmen (EA) finden in Ingenieurdisziplinen häufig Anwendungen. Sie sind nach dem Vorbild der biologischen Evolution entworfen und bestehen klassischerweise aus den Schritten *Fitnessbewertung*, *Selektion* und *Rekombination* (letztere besteht wiederum aus *Crossover* und *Mutation*), in denen aus einer Population P_t eine folgende P_{t+1} entsteht. Ihr Vorteil ist, dass sie in niedrigdimensionalen Suchräumen schon nach kurzer

Laufzeit eine gute Abschätzung der Lage der Pareto-Menge im Surchraum liefern. Nachteilig ist allerdings, dass sie viele Parameter enthalten, die problemspezifisch sind und vom Benutzer vorgegeben werden müssen. Um diese zu ermitteln, bedarf es zumeist mehrerer Testläufe und Erfahrungswerte. Außerdem kann, im Gegensatz zu den GAIO-Algorithmen (s.u.), keine Konvergenz gegen die Pareto-Menge garantiert werden.

Zitzler und Thiele [82] untersuchten 1999 verschiedene evolutionäre Mehrzieloptimierungsalgorithmen und veröffentlichten einen eigenen unter dem Namen *Strength Pareto EA* (*SPEA*). Im Rahmen dieser Arbeit ist eine Version dieses Algorithmus implementiert worden, der die Fitnessbewertung des SPEA nutzt. Für eine ausführlichere Dokumentation siehe Abschnitt 3.2.

Bei Mostaghim [53] findet der SPEA Anwendung beim Entwurf von Antennen und zur Optimierung von chemischen Prozessen. Dort werden auch Datenstrukturen für MOPs vorgestellt und ϵ-*Dominanz* definiert. Mittels der ϵ-Dominanz lässt sich die Größe des sich aufbauenden Archivs kontrollieren und es bietet sich der Vorteil eines Konvergenznachweises, wie er bei Schütze [63] aufgeführt ist. Durch Hinzunahme eines Archivierers soll die Anzahl der Vergleiche und somit die Laufzeit des EA verringert werden, während eine gleichmäßige Annäherung an die Pareto-Menge gewährleistet sein soll. Für die Anwendung der Schaltungsoptimierung machen die Funktionsauswertungen (Schaltungssimulationen) mehr als 90 % der Laufzeit aus, weshalb Archivierer in dieser Arbeit nicht berücksichtigt werden.

Zur Optimierung integrierter Schaltungen nutzt Thomas [71] den SPEA am Beispiel von Addierern. Dienstuhl [21, 20] wendet EAs zur Auswahl von Transistortypen für Flip-Flop Gatter und Speicherzellen an. Fonseca [24] stellt einen *genetischen Algorithmus* (Unterklasse der EAs) zur Mehrzieloptimierung vor.

1.3.2 GAIO - Multilevel Subdivision Techniques

Am Lehrstuhl für Angewandte Mathematik der Universität Paderborn ist eine Sammlung von Algorithmen entstanden, die die Pareto-Menge einer Mehrzieloptimierungsaufgabe approximieren [18, 62]. Diese sind in einer Software Toolbox *GAIO*[4] zusammengefasst. Für eine Variante dieser Algorithmen wird Konvergenz gegen die Menge der stationären Punkte nachweisen [18] (von der die gesuchte Pareto-Menge eine Teilmenge ist, s.o.). Im Kapitel 2 werden aus dieser Toolbox zwei Algorithmen vorgestellt.

Durch die Konvergenz dieser Klasse kann garantiert werden, dass durch Wahl der Anzahl an Schritten eine vorgegebene Approximationsgüte gewährleistet ist und die Menge der ressourceneffizienten Schaltungen somit beliebig genau bestimmt werden kann. Außerdem lassen sich durch die verwandte *Boxstruktur* die Ergebnisse übersichtlicher darstellen. Durch ihre sehr gründliche Suche benötigen sie für den schlechtest angenommenen Fall, im Gegensatz zu den EAs etwas mehr Funktionsauswertungen (vgl. Kapitel 6). Daher sind

[4]Global Analysis of Invariant Objects, s. www.math.upb.de/∼agdellnitz

gerade zur Voruntersuchung, zur Bestimmung der ungefähren Lage der Pareto-Menge, die EAs im Vorteil. Praktischerweise lohnt es sich daher zur Schaltungsoptimierung eine Kombination beider Algorithmentypen einzusetzen, um Vorteile beider Algorithmen zu nutzen.

Kapitel 2

Entwurf ressourceneffizienter Standardzellen

In diesem Kapitel wird am Beispiel einfacher CMOS-Standardzellen die Methodik der MOP-basierten Schaltungsoptimierung vorgestellt. Zunächst werden die zu betrachtenden Ressourcen und ihre Abhängigkeit von den Entwurfsparametern untersucht. Für den CMOS-Inverter kann so die ungefähre Lage der Pareto-Menge vorherbestimmt werden. Mit zwei verschiedenen Algorithmen der *Global Analysis of Invariant Objects* (GAIO) Toolbox [18, 60, 61], die im folgenden Abschnitt kurz vorgestellt werden, wird die Lösungsmenge genauer bestimmt. Alle in dieser Arbeit mit GAIO-Algorithmen erzielten Resultate sind in Kooperation mit dem Paderborner Lehrstuhl für Angewandte Mathematik entstanden. Ebenfalls festgehalten wurden Ergebnisse dieser Kooperation in [7] und [67].

2.1 GAIOs Sampling Algorithmus

Der *Sampling Algorithmus* wird von Dellnitz, Schütze und Hestermeyer in [18] vorgestellt. Der Suchraum S wird rekursiv in schrittweise feinere Boxkollektionen $\{\mathcal{B}_k\}_k$ unterteilt, die die Pareto-Menge immer genauer approximieren. Durch Stichproben wird entschieden, welche der Boxen $B \in \mathcal{B}_k$ keine Pareto-Punkte enthalten, um diese dann für die weitere Suche auszuschließen. Initialisiert durch eine Unterteilung $\mathcal{B}_0$ mit

$$\bigcup_{B \in \mathcal{B}_0} B = S$$

führt der Algorithmus bis zu einer vorgegebenen Rekursionstiefe in jedem Schritt eine *Unterteilung* und eine *Auswahl* durch.

Unterteilung

a) Bilde aus $\mathcal{B}_{k-1}$ ein neues System $\hat{\mathcal{B}}_k$ von Teilmengen, so dass

$$\bigcup_{B \in \hat{\mathcal{B}}_k} B = \bigcup_{B \in \mathcal{B}_{k-1}} B$$

und

$$\operatorname{diam}\left(\hat{\mathcal{B}}_k\right) = \theta_k \operatorname{diam}\left(\mathcal{B}_{k-1}\right),$$

wobei

$$0 < \theta_{\min} \leq \theta_k \leq \theta_{\max} < 1.$$

Auswahl

b) Für alle $B \in \hat{\mathcal{B}}_k$: Wähle eine Menge von Testpunkten $\mathcal{X}_B \subset B$

c)

$$N := \text{nicht-dominierte Punkte aus } \bigcup_{B \in \hat{\mathcal{B}}_k} \mathcal{X}_B$$

d)

$$\mathcal{B}_k := \left\{ B \in \hat{\mathcal{B}}_k : \exists y \in \mathcal{X}_B \cap N \right\}.$$

Die Notation in Punkt „d)“ entspricht der in [18]. y wird hier nicht weiter verwandt. Ausgedrückt werden soll lediglich, dass die Schnittmege nicht leer ist.

Abbildung 2.1 zeigt die ersten zwei Schritte des Unterteilungsalgorithmus am Beispiel des CMOS-Inverters entsprechend der obigen Nummerierung. Die Ergebnisse des Suchlaufs werden weiter hinten vorgestellt.

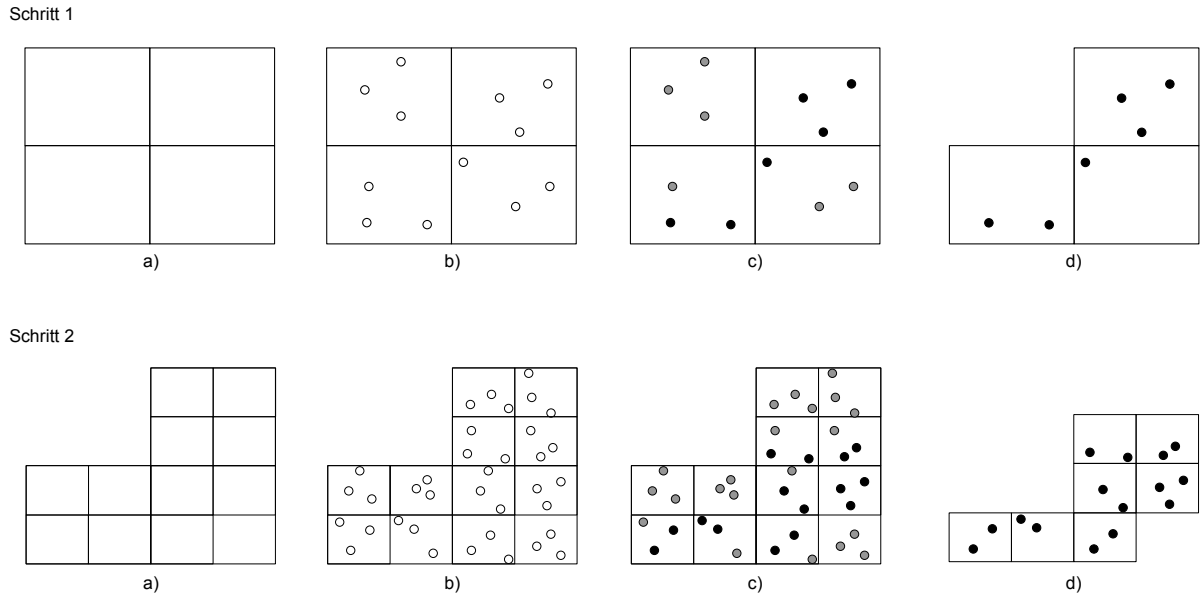


Abbildung 2.1: Die ersten Schritte von GAIOs Sampling Algorithmus. Beispiel CMOS-Inverter. Dominierte Punkte sind grau gekennzeichnet.

Einige weitere Unterteilungsschritte zur Approximation der Pareto-Menge sind in Abbildung 2.2 dargestellt. Informationen über die Wahlmöglichkeiten der Testpunkte sind zu finden in [60] und [61].

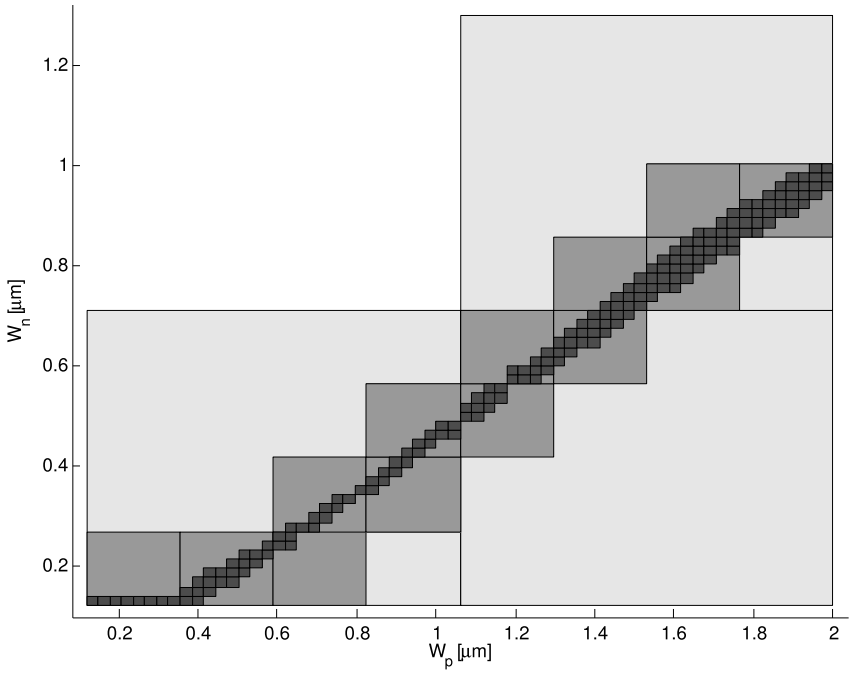


Abbildung 2.2: Einige ausgewählte Unterteilungsschritte des Sampling Algorithmus (90 nm CMOS-Inverter) [7]

Ebenso wie der Sampling Algorithmus verfeinert GAIOs Unterteilungsalgorithmus eine Folge von Umgebungen der gesuchten Pareto-Menge. Während der Suche werden bei diesem Algorithmus Punkte nicht stichprobenartig gestreut, sondern schrittweise in einem Abstiegsverfahren aus vorhergehenden berechnet:

$$x_{j+1} = x_j + h_j q(x_j), j = 0, 1, ...,$$

mit einer Schrittweite h_j. Die Richtung $q : \mathbb{R}^n \to \mathbb{R}^n$ wird bestimmt als gewichtete Summe der Gradienten von f_i, $i = 1, .., m$ des MOP (1.4)

$$q(x) = \sum_{i=1}^{m} \hat{\alpha}_i \nabla f_i(x), \tag{2.1}$$

wobei $\hat{\alpha} = (\hat{\alpha}_1, .., \hat{\alpha}_m)$ eine Lösung sei für

$$\min_{\alpha \in \mathbb{R}^m} \left\{ \left\| \sum_{i=1}^{m} \alpha_i \nabla f_i(x) \right\|_2^2 \; \alpha_i \geq 0, i = 1, ..m, \sum_{i=1}^{m} \alpha_i = 1 \right\}. \tag{2.2}$$

Für diesen Algorithmus wird in [18] Konvergenz gegen substationäre Punkte bewiesen.

Zu den Zielfunktionen der in dieser Arbeit betrachteten Grundgatter liegen keine Ableitungen vor. Approximationen für $\nabla f_i(x)$ aus (2.1) und (2.2) müssten in jedem Schritt, für jeden Punkt und für k Zielfunktionen berechnet werden. Da jede Auswertung der Zielfunktionen eine Schaltungssimulation erfordert, würde der Unterteilungsalgorithmus eine wesentlich längere Laufzeit haben als der Sampling Algorithmus. Untersuchungen

der Zielfunktionen und Probeläufe des Suchalgorithmus haben außerdem gezeigt, dass sich die Zielfunktionen gutmütig genug verhalten, um mit den Testpunkten des Sampling Algorithmus mit hoher Wahrscheinlichkeit entscheiden zu können, ob eine Box Teile der Pareto-Menge enthält oder nicht. Ein weiterer Vorteil des Sampling Algorithmus gegenüber dem Unterteilungsalgorithmus liegt in der globalen Optimierung, die nicht die Menge substationärer Punkte sondern deren Teilmenge, die Pareto-Menge, approximiert.

2.2 GAIOs Fortsetzungsalgorithmus zur Wiederherstellung

Falls während des Suchvorgangs mit dem Sampling Algorithmus Boxen entfernt werden, die Teile der Pareto-Menge enthalten, können die verloren gegangenen Boxen wiederhergestellt werden. Die Pareto-Menge des MOP (1.4) bildet - unter einfachen Glattheitsbedingungen der Zielfunktionen - lokal eine $(k-1)$-dimensionale Mannigfaltigkeit [34] und somit sind weitere Pareto-Punkte nur in der Umgebung der berechneten Boxsammlung zu erwarten. Nachdem der Sampling Algorithmus eine vorgegebene Unterteilungstiefe erreicht hat, kann zur möglicherweise notwendigen Wiederherstellung der Boxsammlung ein Fortsetzungsalgorithmus der GAIO-Toolbox verwandt werden.

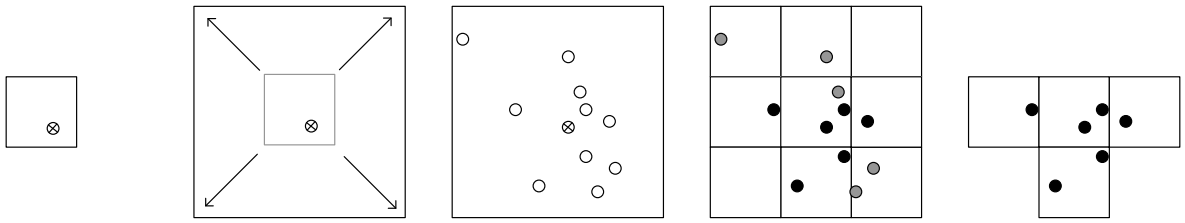


Abbildung 2.3: GAIOs Fortsetzungsalgorithmus für kontinuierlichen Parameterraum (Dominierte Punkte sind grau gekennzeichnet.)

Gegeben ist eine Menge von Startpunkten und eine Unterteilungstiefe des Suchraums. Abbildung 2.3 verdeutlicht die Funktionalität des Fortsetzungsalgorithmus: Zu jedem Startpunkt wird die Box, in der er liegt, um ein ganzes Vielfaches (hier dreifach) der Kantenlänge in alle Dimensionen vergrößert und auf die vorgegebene Boxgröße unterteilt. Genau wie im Sampling Algorithmus wird durch Testpunkte untersucht, ob die neuen Boxen Teile der Pareto-Menge enthalten.

Der Fortsetzungsalgorithmus kann aufbauend auf den durch den Sampling Algorithmus berechneten nicht-dominierten Punkten eine Boxsammlung in derselben Unterteilungstiefe gegebenenfalls vervollständigen. Er kann bei gutmütigen Zielfunktionen aber auch mit Punkten außerhalb der Pareto-Menge initialisiert werden. Abbildung 2.4 zeigt, wie sich die Pareto-Menge eines CMOS-Inverters auf diese Weise überdecken lässt. Beim Übergang vom Sampling Algorithmus zum Fortsetzungsalgorithmus können daher auch einige

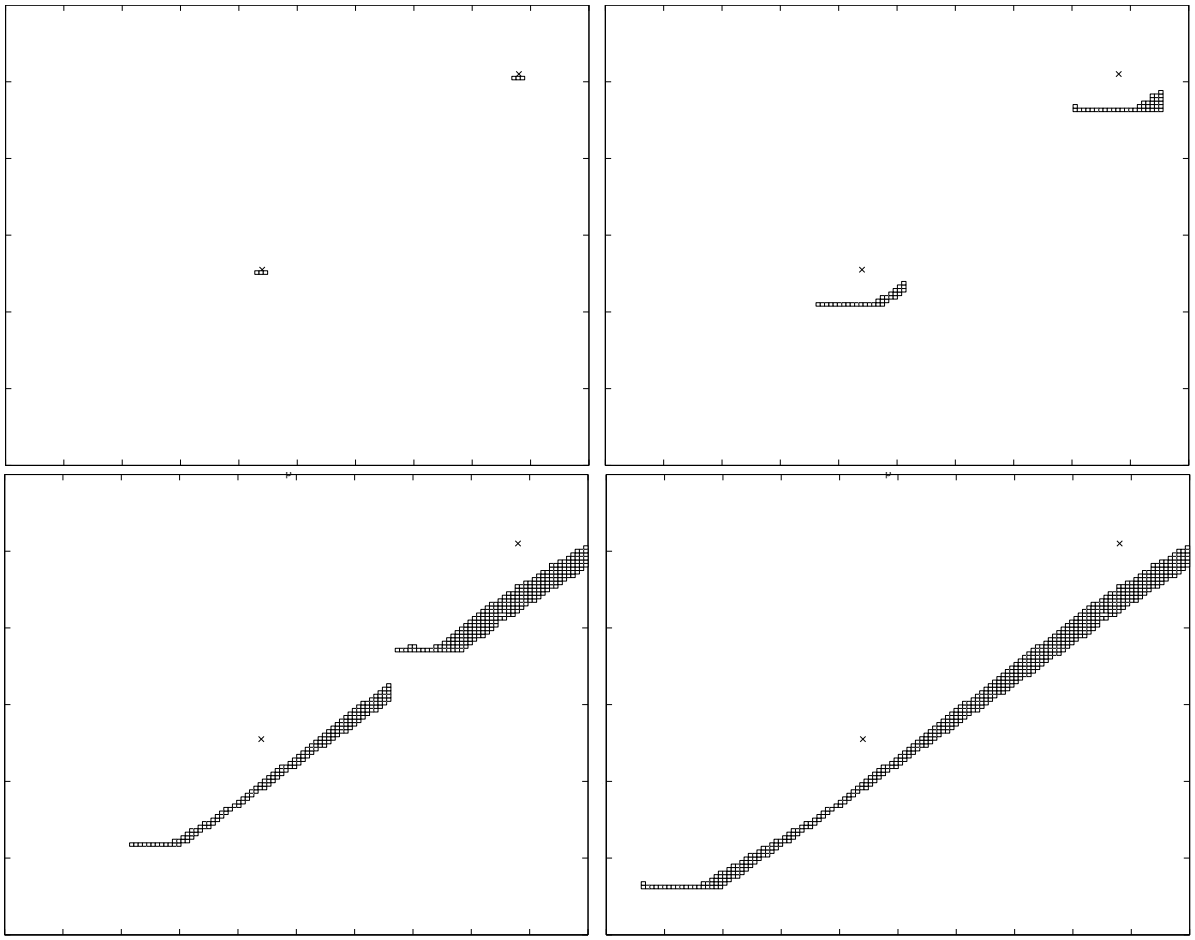


Abbildung 2.4: GAIOs Fortsetzungsalgorithmus mit zwei Startpunkten (durch x gekennzeichnet)

Unterteilungstiefen übersprungen werden. Es genügt eine Menge von Startpunkten aus einer recht groben Unterteilung, um eine wesentlich feinere Überdeckung zu erzeugen. Für dieses Beispiel wurden als Startpunkte zwei kommerzielle Inverter gewählt von denen man ausgehen konnte, dass sie annähernd optimal sind. Daher sind auch durch Handrechnungen approximierte Weitenverhältnisse gute Startwerte für die Schaltungsoptimierung mit dem Wiederherstellungsalgorithmus.

2.2.1 GAIO zur Optimierung digitaler Grundgatter

Die Algorithmen werden zur Mehrzieloptimierung digitaler Standardzellen unter Berücksichtigung der problemspezifischen Eigenschaften erweitert und angepasst. Die Aufgabe der in dieser Arbeit vorgestellten Schaltungsoptimierung besteht zum einen in der Anpassung und Erweiterung der GAIO-Algorithmen, um diese nutzbringend einsetzen zu können. Zum anderen besteht sie in der Formulierung eines sinnvollen MOPs zu einer zu optimierenden Schaltung. Wie zu einer CMOS-Schaltung ein MOP formuliert wird, ist im folgenden Abschnitt am Beispiel des CMOS-Inverters beschrieben. Ein Testpunkt pro

Box ist für die Anwendung der Schaltungsoptimierung sinnvoll, da der Fortsetzungsalgorithmus im Regelfall in einer Unterteilungstiefe gestartet wird, in der die Kantenlängen der Boxen etwa der minimalen Schrittweite der Transistorweiten und -längen entsprechen. Alle Punkte innerhalb einer Box würden daher auf dasselbe Layout gerundet werden.

Besondere Berücksichtigung muss in der Schaltungsoptimierung die sehr lange Auswertungszeit der Zielfunktionen, die eine Schaltungssimulation erfordert (1-5 Sekunden), finden. Abbildung 2.5 zeigt, wie der Fortsetzungsalgorithmus im Rahmen der Diplomarbeit [67] umgestaltet wurde, um die Anzahl der Testpunkte auf ein Minimum zu reduzieren: Während die zufällig verteilten Testpunkte in Abbildung 2.4 noch vermehrt in derselben Box liegen können, wird zur Schaltungsoptimierung jede Box nur durch einen einzelnen Testpunkt kontrolliert. Diese Vorgehensweise spart Zeit bei der Funktionsauswertung ein, da zum einen Boxen nicht mehrfach geprüft werden. Zum anderen ist sichergestellt, dass keine Box ohne Testpunkt und damit ungetestet verworfen wird, was dazu führen könnte, dass in einem weiteren Schritt dieselbe Umgebung nochmals durchsucht wird.

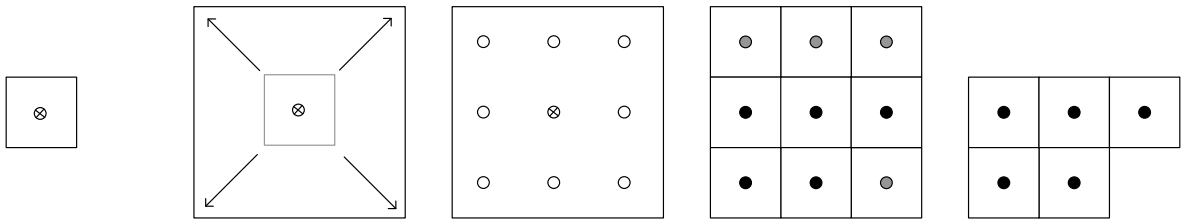


Abbildung 2.5: GAIOs Fortsetzungsalgorithmus für diskreten Parameterraum (Dominierte Punkte sind grau gekennzeichnet.)

Die Funktionsauswertungen haben den bei weitem größten Anteil an der Laufzeit der vorgestellten GAIO-Algorithmen. Dadurch, dass in einem Schritt der Algorithmen jeweils eine ganze Menge von Punkten gleichzeitig ausgewertet werden, lassen sich viele Zielfunktionswerte gleichzeitig berechnen. Abbildung 2.6 zeigt ein Server-Client-System, das speziell für die Schaltungsoptimierung als Erweiterung im Rahmen der Diplomarbeit [67] zur GAIO-Toolbox hinzugekommen ist. Der Server verteilt eine Liste von Punkten an Clients in einem Rechencluster. Jeder Punkt steht für die individuelle Konfiguration einer Schaltung, deren Simulation von jedem Client angestoßen und anschließend ausgewertet wird. Die Ergebnisse der Simulation werden dann vom Server gesammelt und geschlossen als Antwort an GAIO zurückversandt. Durch die Möglichkeit der parallelen Funktionsauswertung reduziert sich die Laufzeit des gesamten Algorithmus nahezu linear mit der Anzahl der Simulatoren.

Die GAIO-Algorithmen sind in C und C++ programmiert. Die Kommunikation mit dem verteilten System nutzt Routinen, die auch in anderen C und C++ Programmen verwandt werden können. Außerdem können Punktlisten aus Matlab heraus ausgewertet werden.

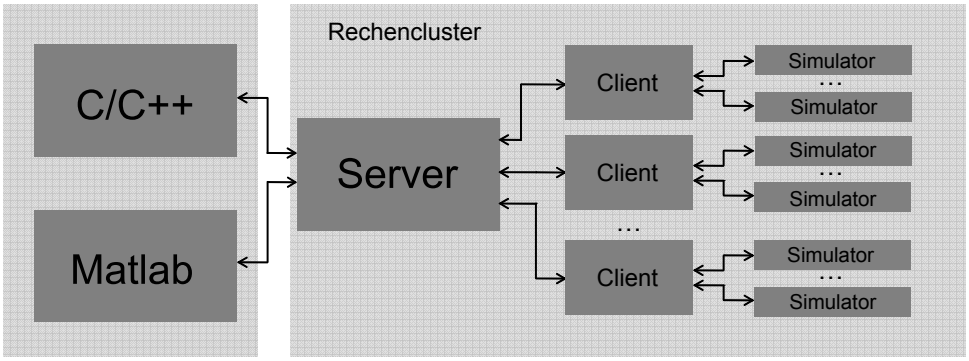


Abbildung 2.6: Verteiltes System zur parallelen Simulation von Grundgattern

2.3 Der CMOS-Inverter

Sowohl in seiner Funktion als auch in seinem Aufbau ist der Inverter das einfachste Logikgatter. Abhängigkeiten zwischen den Geometrieparametern und seinen Ressourcenmaßen sind daher einfacher auszumachen als bei anderen Gattern. In integrierten Schaltungen findet er nicht nur in seiner boolschen Funktion Anwendung, sondern auch als Verstärker oder als Basis-Baustein einer statischen Speichereinheit. Im ersten Kapitel wurde auf die klassische Art der Dimensionierung des CMOS-Inverters und die Übertragbarkeit auf komplexere Logikgatter eingegangen. Im Folgenden wird gezeigt werden wie weitreichend die Verbesserungsmöglichkeiten durch die Anwendung von Mehrzieloptimierungsalgorithmen sind. Viele grundsätzliche Eigenschaften des Inverters werden sich auf komplexere Gatter übertragen lassen.

Die Anwendung eines Mehrzieloptimierungsalgorithmus erfordert zunächst die Formulierung eines MOP (1.4). Das bedeutet, dass der Suchraum S der freien Parameter abgesteckt wird und aus den Ressourcenmaßen die Zielfunktionen definiert werden. Vor Beginn der Optimierung ist daher eine genauere Untersuchung der Schalteigenschaften sinnvoll. An den wichtigsten Eigenschaften wird in diesem Abschnitt untersucht wie gut sich ein einfaches Transistor Modell zur Vorhersage von mit BSIM Modellen berechneten Pareto-Mengen eignet.

2.3.1 Analytische Vorbetrachtung

Abbildung 2.7 zeigt einige Ressourcenmaße des CMOS-Inverters in Abhängigkeit seiner Transistorweiten. Die Längen sind für diese Vorbetrachtung stets konstant und kleinstmöglich gewählt. Um die ungefähre Lage einer Pareto-Menge vorherzusagen, ist es notwendig, die konkurrierenden Schalteigenschaften in ihrer Relation genauer zu untersuchen. Im Folgenden wird gezeigt, wie man dazu das Monotonieverhalten der einzelnen Eigenschaften anhand eines Großsignal-Transistormodells abschätzen kann.

Abbildung 2.8 zeigt das Schaltbild eines CMOS-Inverters einschließlich einer Lastkapazität. Sie entspricht dem Aufbau der im Simulator verwendeten Testumgebung und definiert die

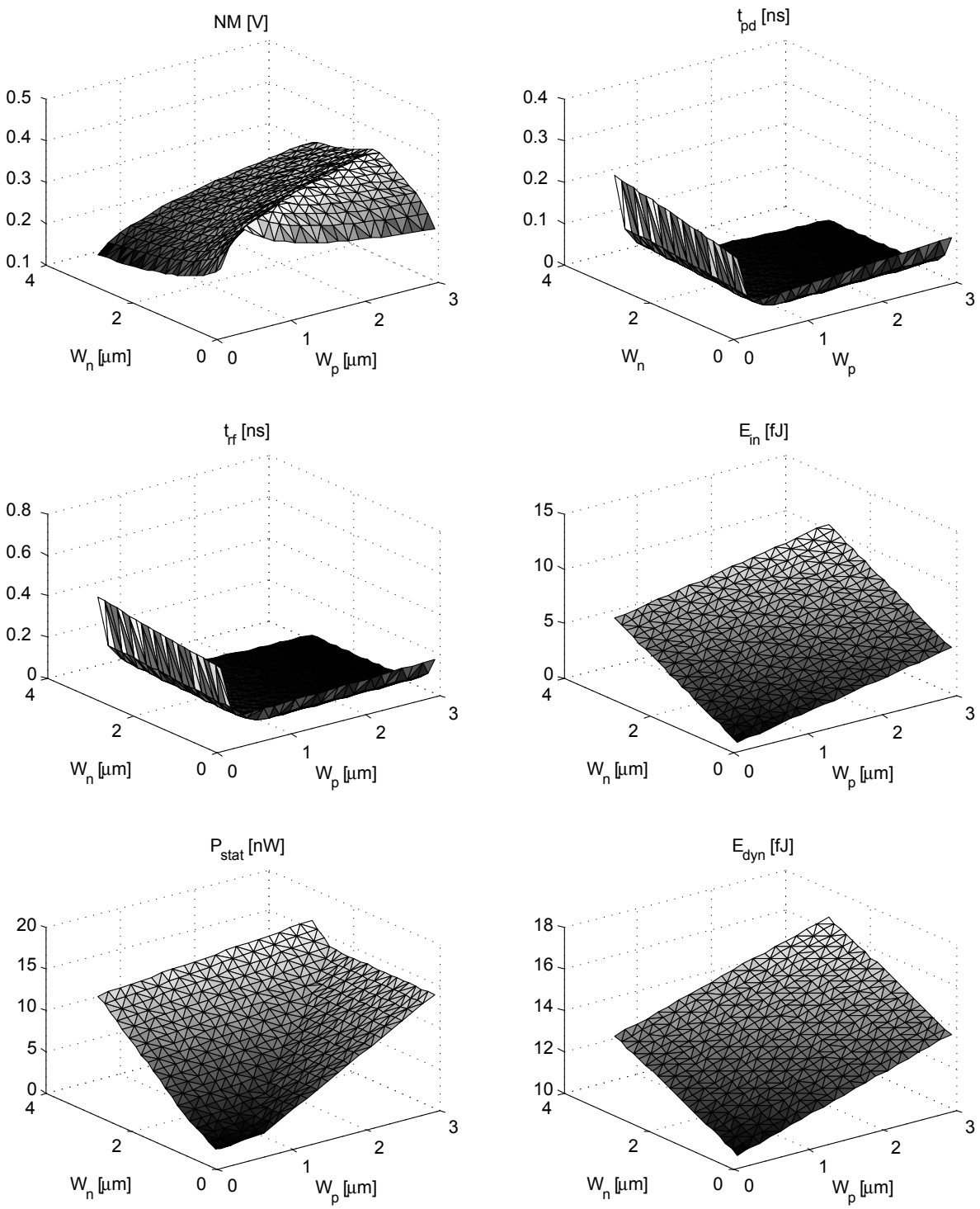


Abbildung 2.7: Schalteigenschaften des 90 nm CMOS-Inverters in Abhängigkeit der Transistorweiten. (Längen minimal, HSPICE-Simulationen)

verwandten Bezeichner.

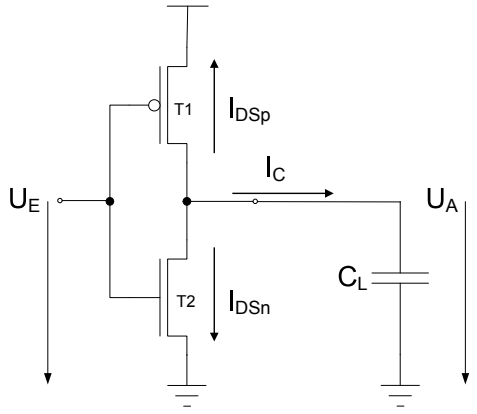


Abbildung 2.8: Schaltbild CMOS-Inverter mit Lastkapazität

Die Gleichungen für den Drain Source Strom der beiden Transistoren

$$I_{\mathrm{DSn}} = \left\{ \begin{array}{ll} 0 & ;U_{\mathrm{GSn}} < U_{\mathrm{THn}} \\ B_{\mathrm{n}} \frac{W_{\mathrm{n}}}{L_{\mathrm{n}}} \left(U_{\mathrm{GSn}} - U_{\mathrm{THn}} - \frac{U_{\mathrm{DSn}}}{2} \right) U_{\mathrm{DSn}} & ;U_{\mathrm{DSn}} \leq U_{\mathrm{GSn}} - U_{\mathrm{THn}} \\ \frac{B_{\mathrm{n}}}{2} \frac{W_{\mathrm{n}}}{L_{\mathrm{n}}} \left(U_{\mathrm{GSn}} - U_{\mathrm{THn}} \right)^2 & \\ \quad \cdot \left[1 + \lambda_{\mathrm{n}} \left(U_{\mathrm{DSn}} - \left(U_{\mathrm{GSn}} - U_{\mathrm{THn}} \right) \right) \right] & ;U_{\mathrm{DSn}} > U_{\mathrm{GSn}} - U_{\mathrm{THn}} \end{array} \right\}, \tag{2.3}$$

$$I_{\mathrm{DSp}} = \left\{ \begin{array}{ll} 0 & ;U_{\mathrm{GSp}} > U_{\mathrm{THp}} \\ B_{p} \frac{W_{\mathrm{p}}}{L_{p}} \left(U_{\mathrm{GSp}} - U_{\mathrm{THp}} - \frac{U_{\mathrm{DSp}}}{2} \right) U_{\mathrm{DSp}} & ;U_{\mathrm{DSp}} \geq U_{\mathrm{GSp}} - U_{\mathrm{THp}} \\ \frac{B_{p}}{2} \frac{W_{\mathrm{p}}}{L_{p}} \left(U_{\mathrm{GSp}} - U_{\mathrm{THp}} \right)^2 & \\ \quad \cdot \left[1 + \lambda_{\mathrm{p}} \left(U_{\mathrm{DSp}} - \left(U_{\mathrm{GSp}} - U_{\mathrm{THp}} \right) \right) \right] & ;U_{\mathrm{DSp}} < U_{\mathrm{GSp}} - U_{\mathrm{THp}} \end{array} \right\} \tag{2.4}$$

enthalten mit $\left[1 + \lambda \left(U_{DS} - \left(U_{GS} - U_{TH} \right) \right) \right]$ einen Term, der den Kanallängeneffekt in Sättigung berücksichtigt und den Übergang zum Triodenbereich stetig macht. (Vergleiche dazu [2].) Dort sind auch die (Material-)Konstanten $\lambda_{\ldots}$ und $B_{\ldots}$ beschrieben. W_{n}, W_{p}, L_{n} und L_{p} sind die Weiten und Längen der NMOS- und PMOS-Transistoren.

Zeit und Verlustleistung

Bei sprunghaftem Spannungsanstieg am Eingang von 0V auf U_{dd}, entlädt sich die Kapazität am Ausgang über den NMOS-Transistor. Bei sprunghaftem Spannungsabfall am Eingang von U_{dd} auf 0V, lädt sich die Kapazität über den PMOS-Transistor auf. Wird der jeweils andere Transistor als sperrend angenommen, so fließt während der Umschaltvorgänge ein Strom

$$I_{\mathrm{C}} = C_{\mathrm{L}} \frac{dU_A(t)}{dt} = -I_{\mathrm{DSn/p}}. \tag{2.5}$$

Die Verzögerungszeit t_{HL} für die fallende Flanke des CMOS-Inverters ergibt sich aus der Lösung $U_A(t)$ des aus (2.5) und (2.3) gebildeten Anfangswertproblems (2.6)

$$C_{\mathrm{L}}\frac{dU_A(t)}{dt} = \left\{ \begin{array}{ll} B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}\left(U_{\mathrm{dd}} - U_{\mathrm{THn}} - \frac{U_A(t)}{2}\right)U_A(t) & ;U_A(t) \leq U_{\mathrm{dd}} - U_{\mathrm{THn}} \\ \frac{B_{\mathrm{n}}}{2}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}\left(U_{\mathrm{dd}} - U_{\mathrm{THn}}\right)^2 & \\ \quad \cdot\left[1 + \lambda_{\mathrm{n}}\left(U_A(t) - (U_{\mathrm{dd}} - U_{\mathrm{THn}})\right)\right] & ;U_A(t) > U_{\mathrm{dd}} - U_{\mathrm{THn}} \end{array} \right\} \tag{2.6}$$

mit

$$U_{\mathrm{A}}(0) = 0.9U_{\mathrm{dd}}. \tag{2.7}$$

Der NMOS-Transistor befindet sich in Sättigung ($U_{\mathrm{A}}(t) > U_{\mathrm{dd}} - U_{\mathrm{THn}}$) bis zum Zeitpunkt

$$\begin{array}{rcl} t_1 & = & \frac{-2C_{\mathrm{L}}}{B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}(U_{\mathrm{dd}}-U_{\mathrm{THn}})^2}\int_{0.9U_{\mathrm{dd}}}^{U_{\mathrm{dd}}-U_{\mathrm{THn}}}\frac{dU_{\mathrm{A}}}{1+\lambda_{\mathrm{n}}(U_A-(U_{\mathrm{dd}}-U_{\mathrm{THn}}))} \\ & = & \frac{-2C_{\mathrm{L}}}{B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}(U_{\mathrm{dd}}-U_{\mathrm{THn}})^2}\frac{1}{\lambda_{\mathrm{n}}}\left[\ln\left(1+\lambda_{\mathrm{n}}\left(U_A-(U_{\mathrm{dd}}-U_{\mathrm{THn}})\right)\right)\right]_{U_A=0.9U_{\mathrm{dd}}}^{U_A=U_{\mathrm{dd}}-U_{\mathrm{THn}}} \\ & = & \frac{2C_{\mathrm{L}}}{B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}(U_{\mathrm{dd}}-U_{\mathrm{THn}})^2}\frac{1}{\lambda_{\mathrm{n}}}\ln\left(1+\lambda_{\mathrm{n}}\left(U_{\mathrm{THn}}-0.1U_{\mathrm{dd}}\right)\right) \end{array}$$

und entlädt sich im Triodenbereich ($U_{\mathrm{A}}(t) \leq U_{\mathrm{dd}} - U_{\mathrm{THn}}$) auf $U_{\mathrm{A}}(t_{\mathrm{HL}}) = 0.1U_{\mathrm{dd}}$ bei

$$\begin{array}{rcl} t_{\mathrm{HL}} & = & t_1 + \frac{-C_{\mathrm{L}}}{B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}}\int_{U_{\mathrm{dd}}-U_{\mathrm{THn}}}^{0.1U_{\mathrm{dd}}}\frac{dU_{\mathrm{A}}}{\left(U_{\mathrm{dd}}-U_{\mathrm{THn}}-\frac{U_{\mathrm{A}}}{2}\right)U_{\mathrm{A}}} \\ & = & t_1 + \frac{-C_{\mathrm{L}}}{B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}}\frac{1}{U_{\mathrm{dd}}-U_{\mathrm{THn}}}\left[\ln\left(\frac{U_{\mathrm{A}}}{U_{\mathrm{A}}-2(U_{\mathrm{dd}}-U_{\mathrm{THn}})}\right)\right]_{U_{\mathrm{A}}=U_{\mathrm{dd}}-U_{\mathrm{THn}}}^{U_{\mathrm{A}}=0.1U_{\mathrm{dd}}} \\ & = & t_1 + \frac{C_{\mathrm{L}}}{B_{\mathrm{n}}\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}(U_{\mathrm{dd}}-U_{\mathrm{THn}})}\ln\left(\frac{19U_{\mathrm{dd}}-20U_{\mathrm{THn}}}{U_{\mathrm{dd}}}\right) \end{array}$$

Zusammengefasst ergibt sich

$$t_{\mathrm{HL}} = \frac{L_{\mathrm{n}}}{W_{\mathrm{n}}} \cdot K_{\mathrm{n}} \tag{2.8}$$

mit

$$\begin{array}{r} K_{\mathrm{n}} = \frac{2C_{\mathrm{L}}}{B_{\mathrm{n}}(U_{\mathrm{dd}}-U_{\mathrm{THn}})^2}\frac{1}{\lambda_{\mathrm{n}}}\ln\left(1+\lambda_{\mathrm{n}}\left(U_{\mathrm{THn}}-0.1U_{\mathrm{dd}}\right)\right) \\ +\frac{C_{\mathrm{L}}}{B_{\mathrm{n}}(U_{\mathrm{dd}}-U_{\mathrm{THn}})}\ln\left(\frac{19U_{\mathrm{dd}}-20U_{\mathrm{THn}}}{U_{\mathrm{dd}}}\right) \end{array}.$$

Die Einflüsse der Transistormaße auf K_{n} sind gering und werden in diesem einfachen Transistormodell vernachlässigt. Analoges gilt für die Anstiegszeit t_{LH} des Aufladevorgangs über den PMOSFET. Die Verzögerungszeit hat nach (1.1) die Form

$$t_{\mathrm{rf}} = \max\left(\frac{L_{\mathrm{n}}}{W_{\mathrm{n}}} \cdot K_{\mathrm{n}}, \frac{L_p}{W_{\mathrm{p}}} \cdot K_{\mathrm{p}}\right)$$

und fällt erkennbar monoton mit steigendem W/L-Verhältnis.

Kleinere Schaltzeiten haben eine größere Verlustleistung zur Folge. Den größten Anteil an der Verlustleistung hat P_{last}. Für diese Betrachtung ist $\alpha_{0\to1}$ aus (1.2) gleich eins. Die Frequenz

$$f = \max(\frac{1}{t_{\mathrm{HL}}}, \frac{1}{t_{\mathrm{LH}}})$$

dagegen ändert sich mit der Schaltgeschwindigkeit des Gatters. Daher steigt

$$P_{\mathrm{last}} = \max\left(\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}} \cdot \frac{1}{K_{\mathrm{n}}}, \frac{W_{\mathrm{p}}}{L_p} \cdot \frac{1}{K_p}\right) \cdot C_L \cdot U_{\mathrm{dd}}^2$$

monoton mit wachsendem W/L-Verhältnis. Was hier an den Beispielen t_{rf} und P_{last} berechnet wurde, lässt sich auf die Verlustleistungsanteile P_{stat}, E_{in} und E_{dyn} und der Schaltgeschwindigkeit t_{rf} verallgemeinern. Die Abhängigkeit von W_{n} und W_{p} bei minimalen Längen ist auch in Abbildung 2.7 zu erkennen, die mithilfe eines Simulators berechnet worden sind.

Um die Punkte optimaler Kompromisse (Pareto-Punkte) zwischen P_{stat} und t_{pd} zu finden, kann man bei diesem einfachen zwei-dimensionalen Beispiel die Höhenlinien der einen Funktion im Suchraum so lange verfolgen wie die andere Funktion abnimmt (wie im Beispiel 2 aus Kapitel 1). Abbildung 2.9 zeigt die Höhenlinien von P_{stat} und t_{pd} wie sie der Simulator berechnet. Während P_{stat} zum Ursprung hin abnimmt, steigt t_{pd} an. Auf Grund des Monotonieverhaltens beider Größen lässt sich die Pareto-Menge in der Nähe der soliden schwarzen Linie erwarten.

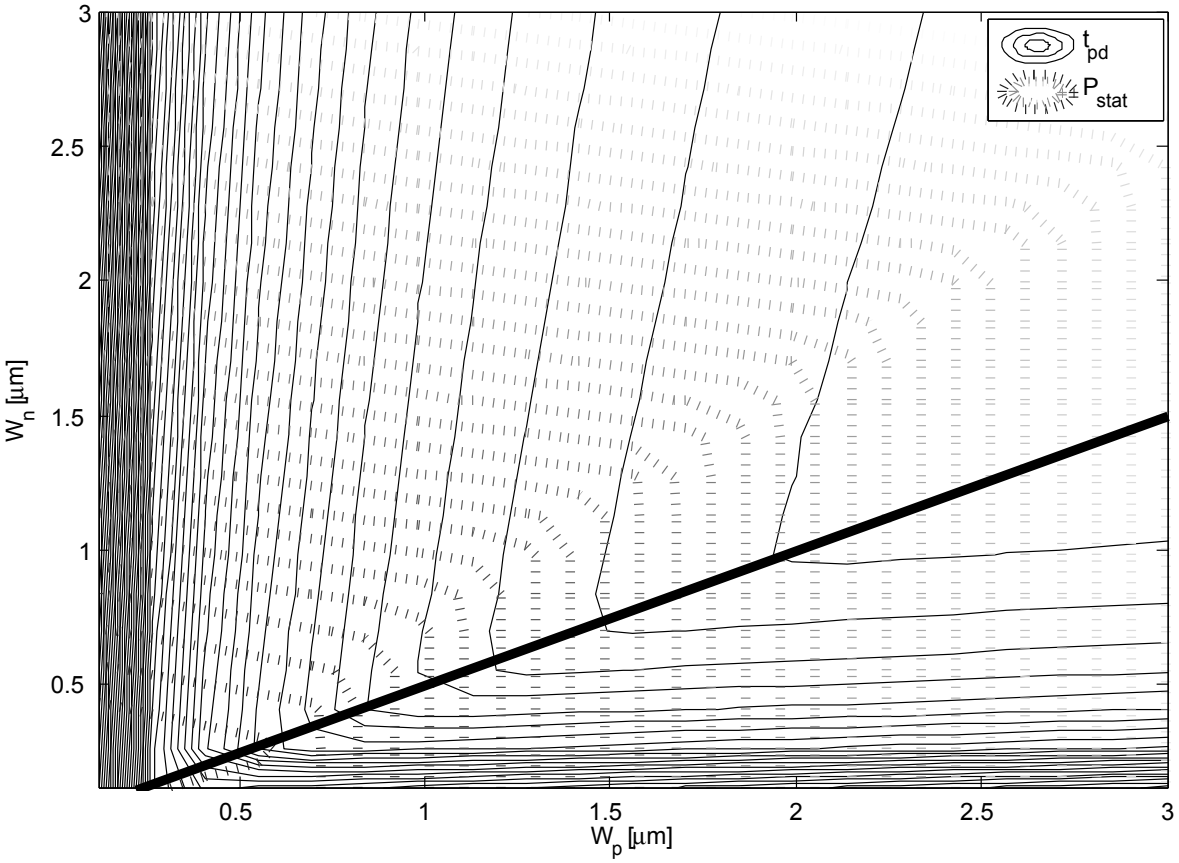


Abbildung 2.9: Höhenlinien t_{pd} und P_{stat} des CMOS-Inverters in Abhängigkeit der Transistorweiten (Längen minimal, HSPICE-Simulationen). Die solide schwarze Linie liegt in der Nähe der zu erwartenden Pareto-Menge.

Die mit der Handrechnung zu erwartende Form ist in Abbildung 2.10 dargestellt. Die genaue Lage und Breite ist von Technologieparametern abhängig, die die Konstanten

K_n und K_p beeinflussen. Beim Vergleich beider Pareto-Mengen ist zu erkennen, dass der Ansatz mit dem einfachen Transistormodell (2.3) und (2.4) zu ähnlichen Vorhersagen führt.

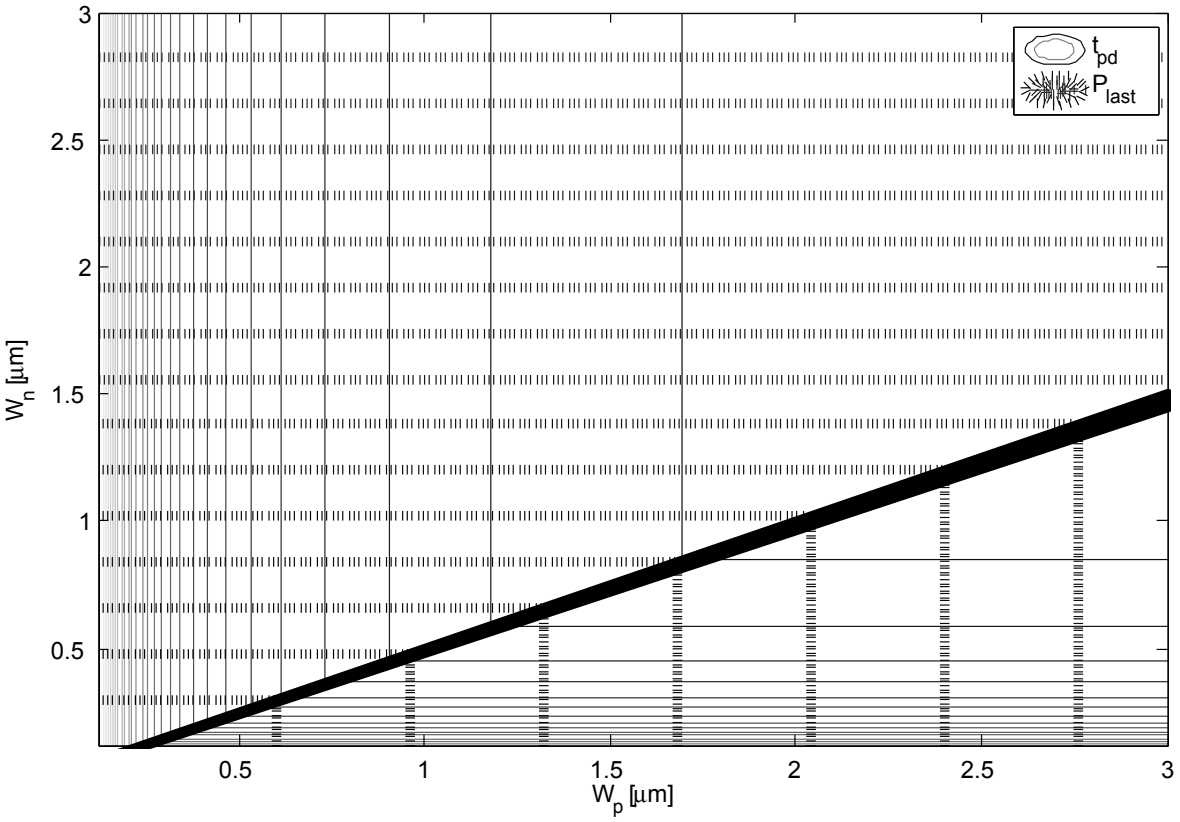


Abbildung 2.10: Höhenlinien t_{pd} und P_{last} des CMOS-Inverters in Abhängigkeit der Transistorweiten (Handrechnung). Die schwarze Fläche liegt in der Nähe der zu erwartenden Pareto-Menge.

Störabstand

Aus (2.3) und (2.4) kann die geschlossene DC Kennline des CMOS-Inverters berechnet werden. Während des Schaltvorgangs durchlaufen die Transistoren fünf Zustandskombinationen (s. Abbildung 2.11). Eine seitlich verschobene Kennlinie würde entweder $\mathrm{NM_H}$ oder $\mathrm{NM_L}$ vergrößern und den anderen Störabstand verkleinern. Soll aber NM in (1.3) maximiert werden, so müssen die Transistoren so dimensioniert werden, dass die DC-Kennlinie symmetrisch verläuft. Mit der Forderung (1.9) aus dem klassischen Ansatz findet man mit den erweiterten Gleichungen (2.3) und (2.4)

$$\frac{W_\mathrm{p}}{L_p}\Big/\frac{W_\mathrm{n}}{L_\mathrm{n}} = \frac{B_\mathrm{n}}{B_p}\,\frac{\left(\frac{U_\mathrm{dd}}{2}-U_\mathrm{THn}\right)^2}{\left(\frac{U_\mathrm{dd}}{2}+U_\mathrm{THp}\right)^2}\,\frac{1+\lambda_\mathrm{n}U_\mathrm{THn}}{1+\lambda_\mathrm{p}U_\mathrm{THp}}.$$

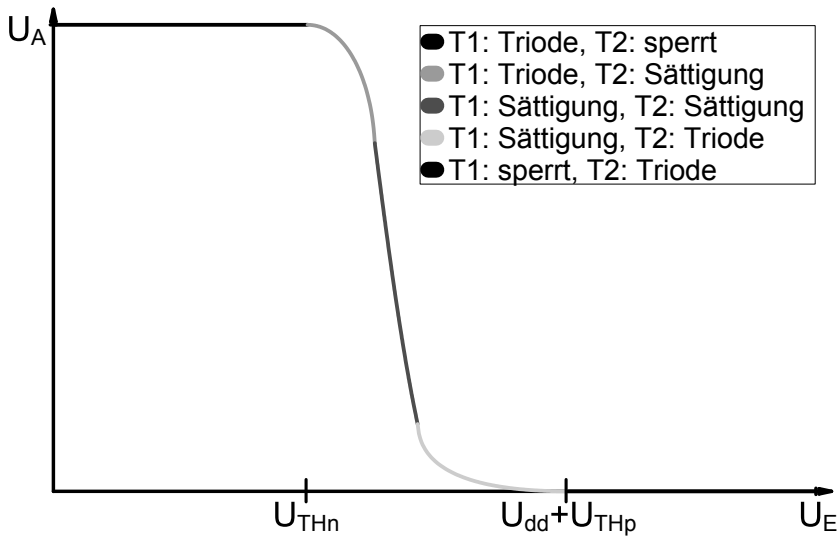


Abbildung 2.11: DC-Kennlinie des CMOS-Inverters

Da für kleinere Technologien die Annahme $U_{\mathrm{THp}} = -U_{\mathrm{THn}}$ und $\lambda_{\mathrm{p}} = -\lambda_{\mathrm{n}}$ nicht mehr hält, ist die Abschätzung (1.11) ungenau. Trotzdem lässt sich die Monotonie des Störabstands in Abbildung 2.7 erklären: Die größten Werte ergeben sich für ein festes $W_{\mathrm{p}}/W_{\mathrm{n}}$-Verhältnis. Je weiter von diesem Verhältnis abgewichen wird, desto schlechter wird der kleinere Störabstand. Daher ergibt sich ein gerader Gipfel, dessen Hänge monoton im ganzen Suchraum abfallen.

Betrachtet man außer der Zeit und der Verlustleistung auch den Störabstand, so wird sich die Pareto-Menge vergrößern, da sich die Möglichkeit Kompromisse zu machen mit einer weiteren Zielgröße erhöht. Ein Gatter mit symmetrischer DC Ausgangskennline, die gute Störabstände verspricht, hat auch eine gleichmäßige transiente Kennlinie zu erwarten und damit gute Schaltzeiten. Daher ist die Pareto-Menge weiterhin in der Nähe der Line aus Abbildung 2.9 zu erwarten, allerdings breiter. Wie viel sich die Menge im Suchraum verbreitert ist größtenteils von Effekten abhängig, die im Level1 Modell nicht modelliert sind. Simulationsunterstützung mit BSIM Modellen ist daher zur genaueren Suche unumgänglich. Die in den folgenden Kapiteln vorgestellten Ergebnisse werden zeigen, dass die mit (2.3) und (2.4) getroffenen Vorhersagen aber zumindest in der Nähe der Pareto-Mengen liegen.

Fläche und Ausbeute

Unter Ausbeute versteht man die Wahrscheinlichkeit mit der eine im Layout vorliegende Schaltung nach Fertigung unter Einfluss der statistischen Streuung in Prozessparametern noch funktioniert. Ohne auf den Fertigungsprozess und die Streuparameter näher einzugehen ist einzusehen, dass größere Transistoren gegenüber Parameterschwankungen unanfälliger sind, und damit die Ausbeute einer Schaltung monoton steigt, je mehr Fläche zur Verfügung steht. Diese beiden Ressourcen in der Mehrzieloptimierung ebenfalls zu

berücksichtigen würde die Pareto-Menge daher weiter vergrößern.

Die während der Suchraumexploration durchgeführten Simulationen sind Simulationen einer Netzliste. Während der Suche liegt dem Algorithmus kein Layout vor, auf dessen Grundlage er die Fläche der Zelle bestimmen könnte. Die automatische Generierung eines Layouts minimaler Größe für jede Schaltung ist, falls möglich, sehr zeitintensiv und damit unpraktikabel. Die Fläche könnte lediglich durch Längen und Weiten der Transistoren abschätzt werden. Für eine sinnvolle Definition eines Ressourcenmaßes $A : \mathbb{R}_+^4 \to \mathbb{R}_+$, $(W_\mathrm{p}, W_\mathrm{n}, L_p, L_\mathrm{n}) \mapsto A\,(W_\mathrm{p}, W_\mathrm{n}, L_p, L_\mathrm{n})$ müsste Monotonie gelten:

$$\hat{W}_\mathrm{p} \leq \check{W}_\mathrm{p} \wedge \hat{L}_\mathrm{p} \leq \check{L}_\mathrm{p} \wedge \hat{W}_\mathrm{n} \leq \check{W}_\mathrm{n} \wedge \hat{L}_\mathrm{n} \leq \check{L}_\mathrm{n} \Rightarrow A(\hat{W}_\mathrm{p}, \hat{L}_\mathrm{p}, \hat{W}_\mathrm{n}, \hat{L}_\mathrm{n}) \leq A(\check{W}_\mathrm{p}, \check{L}_\mathrm{p}, \check{W}_\mathrm{n}, \check{L}_\mathrm{n}).$$

Die bisher eingeführten genau messbaren Schalteigenschaften sollen aber nicht mit einer groben „Schätzfunktion“ kombiniert werden. Die Auswirkungen eines zu ungenauen Ressourcenmaßes könnten die Pareto-Menge grob verschieben. Die Fläche lässt sich aber bei der Wahl einer ressourceneffizienten Schaltung aus der Pareto-Menge berücksichtigen. Die Lage des Punktes im Suchraum, der durch die Transistordimensionen aufgespannt wird, verrät, ob es sich um eine größere oder kleinere Schaltung handelt.

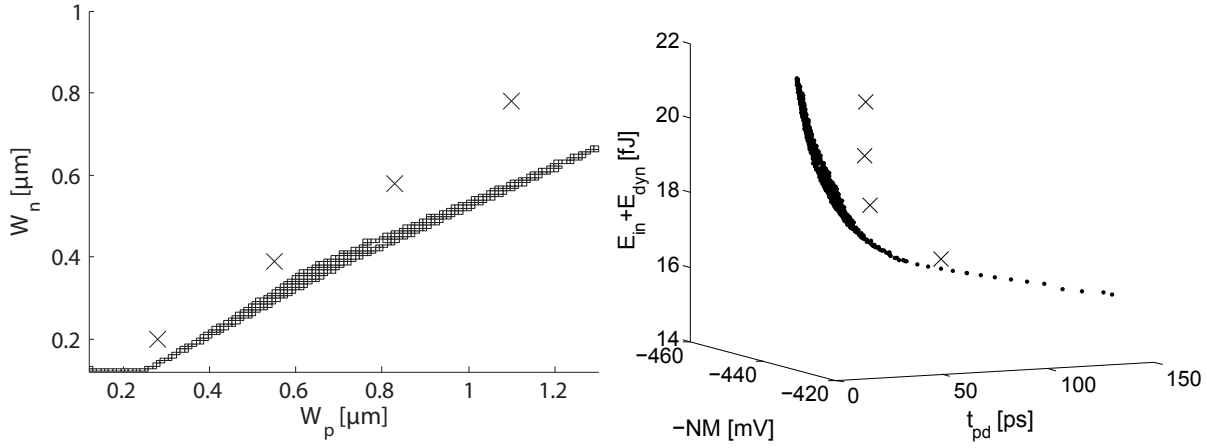
Ebenso wird die Ausbeute erst in einem weiteren Schritt zur Bewertung der einzelnen Pareto-Punkte herangezogen. Monte-Carlo-Simulationen sind - ebenso wie der Entwurf eines Layouts - während der Laufzeit zu aufwendig.

2.3.2 GAIO-Ergebnisse

Die oben vorgestellten Algorithmen des Softwaretools GAIO werden zur Mehrzieloptimierung des CMOS-Inverters angewandt. Die betrachteten Zielfunktionen sind die Ressourcenmaße Störabstand NM, Verzögerungszeit t_pd und Verlustenergie $E_\mathrm{in} + E_\mathrm{dyn}$, die sich während des Umschaltvorgangs durch den Umladevorgang der Eingangskapazität und durch den Stromfluss von Versorgungsspannung zur Masse ergibt. Die Bestimmung der Schalteigenschaften erfolgt durch Simulation der Schaltungen mit BSIM Modellen zu Transistoren einer 65nm und 90nm Technologie. Um die Ergebnisse der Optimierung bewerten zu können, werden die approximierten Pareto-Mengen mit Elementen einer kommerziellen Standardzellenbibliothek verglichen. Diese kommerziellen Schaltungen sind, wie im Abschnitt 1.2 beschrieben, mit konstanten Längen dimensioniert. Zur fairen Vergleichbarkeit sollen die variablen Parameter daher nur die Weiten W_n des NMOSFET und W_p des PMOSFET sein, die einen zwei-dimensionalen Suchraum S aufspannen. Der Vergleich mit Referenzschaltungen wird Verbesserungspotential professionell layouteter Schaltungen zeigen. Ebenso wird die Lage der Pareto-Mengen, die durch BSIM Modelle berechnet werden, die analytische Vorbetrachtung des letzten Abschnitts als gute Abschätzung verdeutlichen.

Der 65nm CMOS-Inverter

Abbildung 2.12 zeigt die approximierte Pareto-Menge und die zugehörige Pareto-Front des 65nm CMOS-Inverters. Mit Kreuzen dargestellt sind die Vergleichswerte der kommerziellen Standardzellenbibliothek. Die Lage der Pareto-Menge ist ähnlich wie in den Abbildungen 2.9 und 2.10. Lediglich ihre Form ist etwas breiter, da hier mit NM eine weitere Zielfunktion hinzu kommt, die die Menge der optimalen Kompromisse vergrößert.



(a) Suchraum (b) Bildraum

Abbildung 2.12: Pareto-Menge und -Front des 65nm CMOS-Inverters mit Referenzpunkten [7] (vgl. auch [67])

Die Grafik zeigt bereits, dass sich die Referenzschaltungen alle verbessern lassen, da sie von Punkten der Pareto-Menge dominiert werden. In Tabelle 2.3.2 werden die Ressourcenmaße der vier Schaltungen Pareto-Punkten gegenübergestellt. Die Pareto-Punkte sind zu jeder Referenzschaltung so gewählt, dass sie in etwa eine gleiche Verzögerungszeit t_{pd} haben und damit eine vergleichbare Treiberstärke. Mit PP1_INV65_X9 und PP2_INV65_X9 gibt es zu INV65_INV65_X9 zwei Pareto-Punkte, die bei gleicher Verzögerungszeit in beiden anderen Zielgrößen eine Verbesserung um rund 4% bieten. Andere Pareto-Punkte verbessern gleichzeitig den Störabstand bis zu 3% und die Verlustenergie um bis zu 2%.

Der 90nm CMOS-Inverter

In der 90nm Technologie können aus einer kommerziellen Standardzellenbibliothek zwei CMOS-Inverter als Referenzpunkte ausgewählt werden. Auch diese liegen, wie Abbildung 2.13 verdeutlicht, neben der Pareto-Menge, die mit den GAIO-Algorithmen bestimmt wurde.

Aus Tabelle 2.3.2 wird ersichtlich, dass PP1_INV90_X1 zu INV90_X1 einen Vorteil von 4,7% beim Störabstand und 3% weniger Verlustenergie bringt. Gleiches gilt für PP2_INV90_X1. INV90_X2 kann ebenfalls verbessert werden. Die dominierenden Pareto-

	W_p [µm]	W_n [µm]	t_pd [ps]	NM [mV]	$E_\mathrm{in} + E_\mathrm{dyn}$ [fJ]
INV65_X2	0,28	0,2	77,1036	436,2425	16,584
PP1_INV65_X2	0,28	0,14	76,8715	448,2422	16,2984
INV65_X4	0,55	0,39	45,5771	437,4251	18,1075
PP1_INV65_X4	0,55	0,32	45,228	446,9256	17,8326
PP2_INV65_X4	0,55	0,29	45,2378	443,7841	17,7499
INV65_X7	0,83	0,58	33,2908	431,4846	19,6167
PP1_INV65_X7	0,825	0,46	33,0596	445,6814	19,2095
PP2_INV65_X7	0,815	0,44	33,2621	444,1997	19,0912
INV65_X9	1,1	0,78	27,19	427,7998	21,1792
PP1_INV65_X9	1,075	0,57	27,1251	445,1607	20,3592
PP2_INV65_X9	1,075	0,555	27,0866	444,1381	20,3154

Tabelle 2.1: 65nm CMOS-Inverter einer kommerziellen Standardzellenbibliothek mit Pareto-Punkten

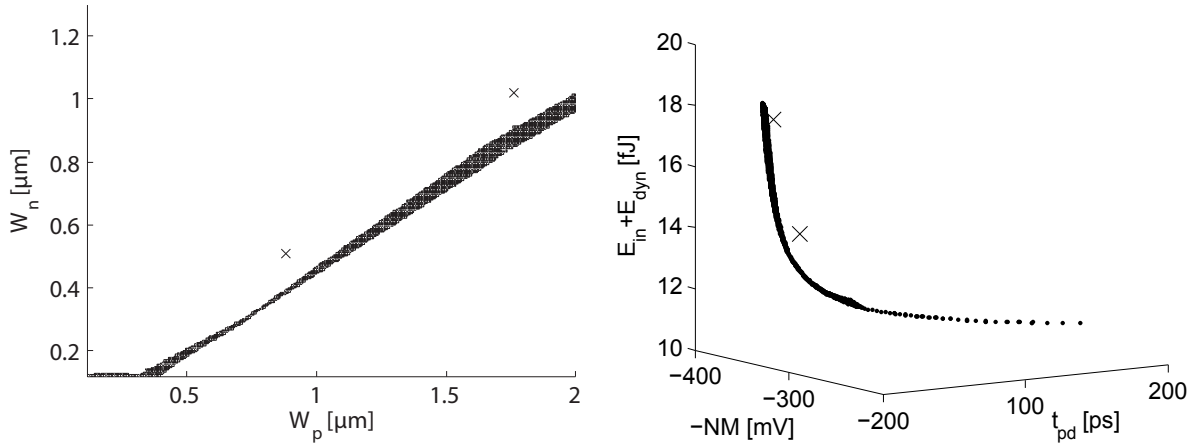
Punkte PP1_INV90_X2 und PP2_INV90_X2 liefern 2 bis 3% Gewinn in beiden Zielgrößen bei gleicher Treiberstärke.

	W_p [µm]	W_n [µm]	t_pd [ps]	NM [mV]	$E_\mathrm{in} + E_\mathrm{dyn}$ [fJ]
INV90_X1	0,88	0,51	31,7405	336,8837	14,1033
PP1_INV90_X1	0,875	0,385	31,5154	352,6163	13,6682
PP2_INV90_X1	0,875	0,38	31,4984	351,895	13,6523
INV90_X2	1,76	1,02	18,1167	343,4092	17,8497
PP1_INV90_X2	1,74	0,89	18,0708	353,462	17,3699
PP2_INV90_X2	1,725	0,845	18,0917	350,1904	17,1537

Tabelle 2.2: 90nm CMOS-Inverter einer kommerziellen Standardzellenbibliothek mit Pareto-Punkten

2.4 CMOS-Gatter mit mehreren Eingängen

Der Ansatz, die Entwurfsraumexploration als MOP zu formulieren, bietet ein hohes Maß an Flexibilität. Bei der Pareto Optimierung von NAND und NOR-Gattern mit zwei Eingängen ist es sinnvoll, die Weiten der in Reihe geschalteten Transistoren unabhängig voneinander variieren zu können. Somit erweitert sich der Suchraum um eine Dimension. Die verwandten Algorithmen müssen allerdings nicht verändert werden. In diesem Abschnitt werden zum einen wiederum Vergleiche mit den kommerziellen Standardzellenbibliotheken gezogen, zum anderen wird der Gewinn durch Erweiterung des Suchraums am 65nm NOR-Gatter demonstriert.



(a) Suchraum (b) Bildraum

Abbildung 2.13: Pareto-Menge und -Front des 90nm CMOS-Inverters mit Referenzpunkten (vgl. [67])

2.4.1 Problemstellung und Simulationsaufbau

In Abschnitt 1.2 wurde ebenfalls kurz darauf eingegangen, wie sich das Dimensionierungsproblem von CMOS-Logikgattern mit mehreren Eingängen auf das des Inverters vereinfachen lässt. Hier werden nicht nur die Längen auf Minimalmaß festgesetzt, sondern auch alle Weiten der P- bzw. NMOSFET zusammengefasst. In gleicher Weise sind auch kommerzielle CMOS-NAND und NOR-Gatter vereinfacht. Um auch hier für einen Vergleich dieselben Voraussetzungen zu haben, soll der Suchraum zunächst wie beim Inverter zwei-dimensional durch die Weiten W_n und W_p aufgespannt werden. Das Eingangssignal wird, wie in Abbildung 2.14 am Beispiel des NAND-Gatters dargestellt, auf den Eingang gegeben, dessen Transistor dem Substrateffekt unterliegt. Der Logikpegel des zweiten Eingangs wird konstant gewählt, so dass die angeschlossenen Transistoren leiten.

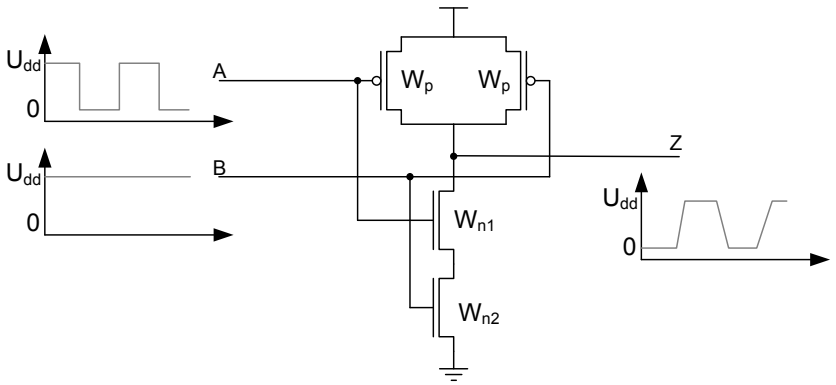


Abbildung 2.14: Simulationsaufbau des NAND-Gatters

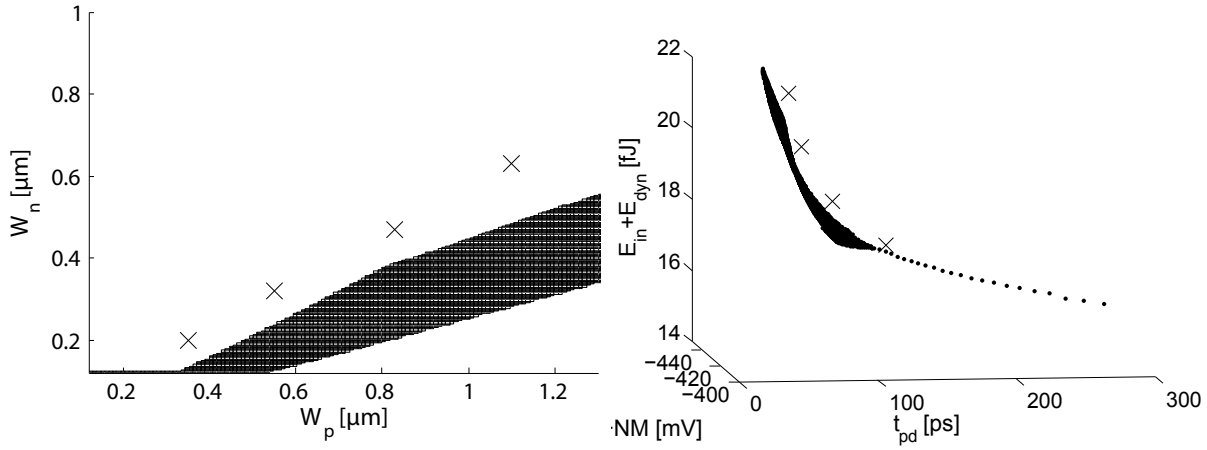
Um den Substrateffekt zu berücksichtigen, werden die Weiten der in Reihe geschalteten Transistoren entkoppelt und wie in Abbildung 2.14 benannt. Damit der Einfluss unter-

schiedlicher Transistorweiten messbar wird, wird der Simulationsaufbau erweitert und das Verhalten der Schaltung zusätzlich mit einem Signal auf Eingang B betrachtet. Aus den sich ergebenden Ressourcenmaßen beider Simulationen wird der jeweils schlechtere Wert ermittelt und als Zielfunktion gewählt. Der Bildraum wird so weiterhin durch die drei Zielgrößen aufgespannt, die schon für den Inverter betrachtet wurden.

2.4.2 GAIO-Ergebnisse

65nm CMOS-NOR-Gatter

Abbildung 2.15 verdeutlicht, dass die Referenzpunkte durch die Ergebnisse der Optimierung in allen Eigenschaften zugleich verbessert werden können. Beim Vergleich mit den Pareto-Mengen der Inverter fällt auf, dass die des NOR-Gatters wesentlich breiter ist. Das bedeutet, dass es im Suchraum des NOR-Gatters sehr viel mehr ressourceneffiziente Realisierungen gibt.



(a) Suchraum (b) Bildraum

Abbildung 2.15: Pareto-Menge und -Front des 65nm CMOS-NOR-Gatters mit Referenzpunkten [7] (vgl. auch [67])

In Tabelle 2.3 sind zu den Referenzpunkten dominierende Pareto-Punkte in Zahlenwerten dargestellt. Die Verzögerungszeit zwischen Referenz- und Pareto-Punkt weicht dabei maximal um 1% (für NOR65_X2 um 1,1 %) ab, wobei die Gatter der Pareto-Punkte stets etwas schneller schalten. PP2_NOR65_X6 verbessert die Verlustenergie von NOR65_X6 um 7,5% und der Störabstand von NOR65_X2 wird durch PP1_NOR65_X1 um 4% erhöht.

Indem die Weiten der PMOS-Transistoren unabhängig voneinander variiert werden, können bei der Suche nach optimalen NOR-Gattern weitere Verbesserungen erzielt werden. In Abbildung 2.16 ist nochmals die Pareto-Front aus Abbildung 2.15 mit den Referenzpunkten der kommerziellen Standardzellenbibliothek abgebildet. In Grau kommt durch die Exploration im erweiterten Suchraum eine Pareto-Front hinzu, die weiter unterhalb liegt.

	W_p [µm]	W_n [µm]	t_pd [ps]	NM [mV]	$E_\mathrm{in} + E_\mathrm{dyn}$ [fJ]
NOR65_X2	0,35	0,2	126,1537	429,4732	17,1132
PP1_NOR65_X1	0,35	0,13	124,7573	446,8807	16,739
NOR65_X3	0,55	0,32	89,1354	431,7035	18,3153
PP1_NOR65_X3	0,545	0,23	88,5165	445,0913	17,8965
PP2_NOR65_X3	0,535	0,155	88,4347	432,2561	17,5125
NOR65_X5	0,83	0,47	65,1734	429,9016	19,9042
PP1_NOR65_X5	0,82	0,385	64,6765	443,9693	19,4902
PP2_NOR65_X5	0,795	0,25	64,9977	430,2098	18,8214
NOR65_X6	1,1	0,63	53,1307	425,5751	21,4935
PP1_NOR65_X6	1,07	0,47	52,867	443,2806	20,7107
PP2_NOR65_X6	1,04	0,31	52,9415	425,6106	19,892

Tabelle 2.3: 65nm CMOS-NOR-Gatter (W_p1=W_p2) einer kommerziellen Standardzellenbibliothek mit Pareto-Punkten

Somit ist zu erkennen, dass nicht nur die Referenzpunkte sondern auch die Pareto-Punkte der zwei-dimensionalen Suche weiter verbessert werden können. Bei gleicher Verzögerungszeit wird im Störabstand eine Verbesserung von bis zu 6% und im Energieverbrauch um 7% zwischen den Pareto-Fronten erreicht.

90nm CMOS-NAND und NOR-Gatter

In Abbildung 2.18 liegen nur einzelne Referenzpunkte für NAND und NOR-Gatter einer 90nm Technologie außerhalb der Pareto-Menge. Selbst diese liegen nicht weit genug entfernt, um gravierend verbessert werden zu können. Allerdings lassen die breiten Pareto-Mengen erkennen, dass sich sehr unterschiedliche ressourceneffiziente Gatter dimensionieren lassen.

65nm CMOS-NAND-Gatter

Die Pareto-Menge des 65nm NAND-Gatters ist für den Fall $W_\mathrm{n1} = W_\mathrm{n2}$ ähnlich breit wie die des 90nm NAND-Gatters. Auch hier befinden sich allerdings die meisten Referenzpunkte unter den ressourceneffizienten Schaltungen. Abbildung 2.19 zeigt, dass sich im dreidimensionalen Suchraum bei weitem bessere NAND-Gatter dimensionieren lassen.

Die in Abbildung 2.17 abgebildete schwarze Pareto-Front enthält die Bilder der Referenzpunkte, die bei der zwei-dimensionalen Suche, wie oben erwähnt, nicht verbessert werden können. Der Vorteil der Dimensionierungen mit unabhängigen Weiten der NMOSFET liegt bei 5% für den Störabstand (zwischen $(W_\mathrm{p}, W_\mathrm{n1}, W_\mathrm{n2}) = (0.8529, 0.7559, 0.7559)$ und $(W_\mathrm{p}, W_\mathrm{n1}, W_\mathrm{n2}) = (1.1018, 0.6391, 0.9484)$) und einer niedrigeren Verlustenergie von bis zu 8% (zwischen $(W_\mathrm{p}, W_\mathrm{n1}, W_\mathrm{n2}) = (1.2862, 0.8591, 0.8591)$ und $(W_\mathrm{p}, W_\mathrm{n1}, W_\mathrm{n2}) = (0.9082, 0.6803, 0.9278)$) bei gleicher Treiberstärke.

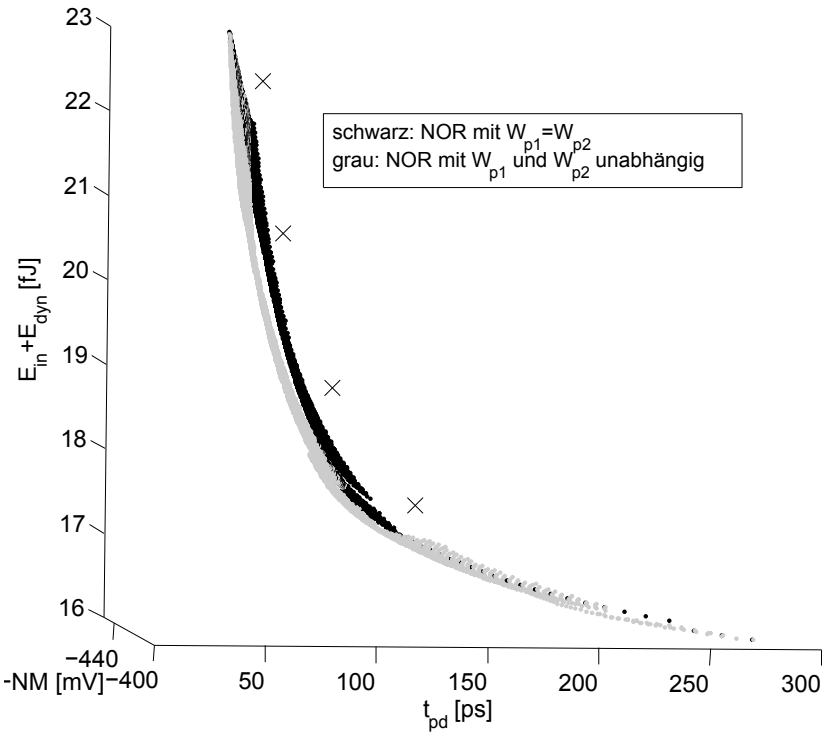


Abbildung 2.16: Pareto-Fronten des 65nm CMOS-NOR-Gatters mit den Variablen W_{n}, W_{p1} und W_{p2} (vgl. [7])

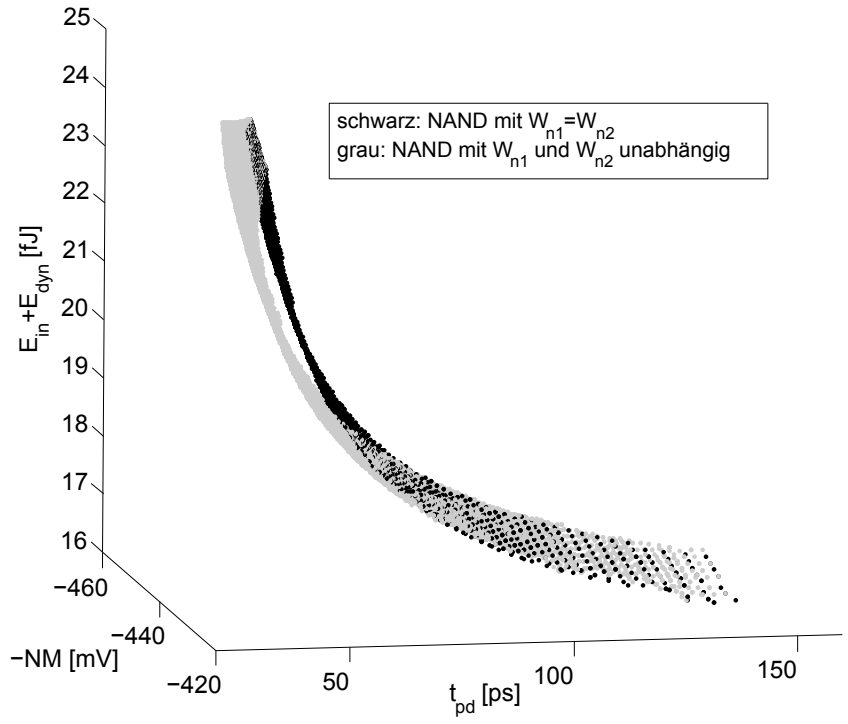
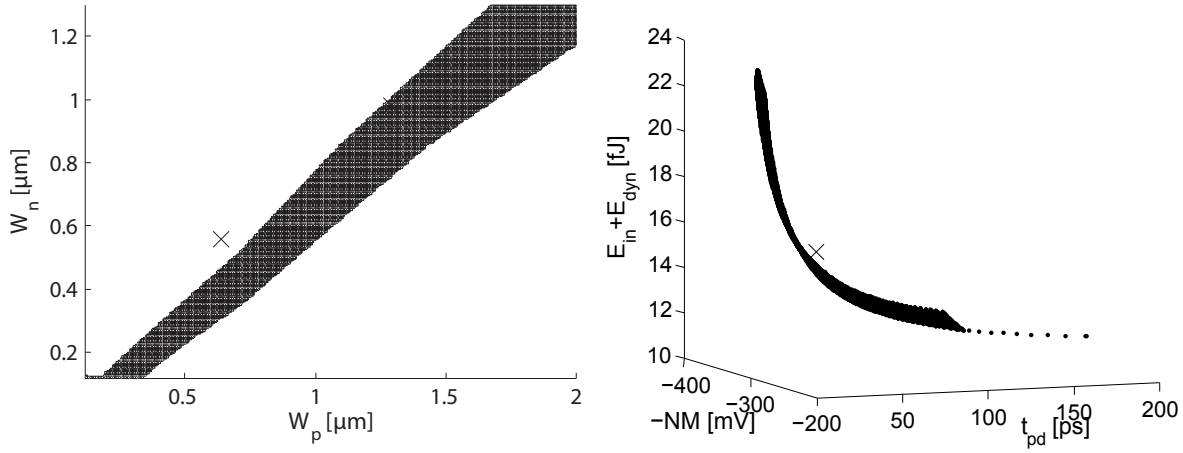
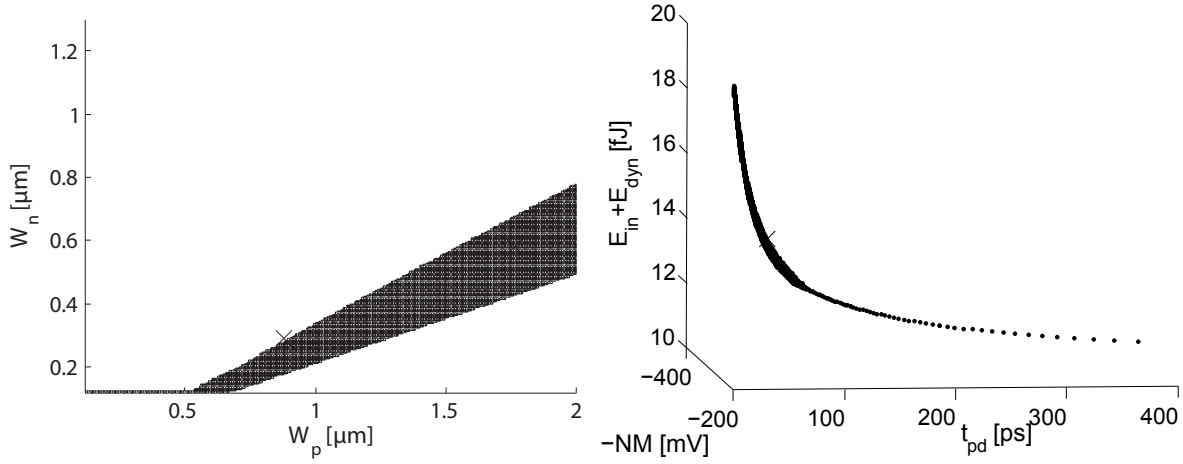


Abbildung 2.17: Pareto-Fronten des 65nm CMOS-NAND-Gatters mit den Variablen W_{p}, W_{n1} und W_{n2} (vgl. [67])

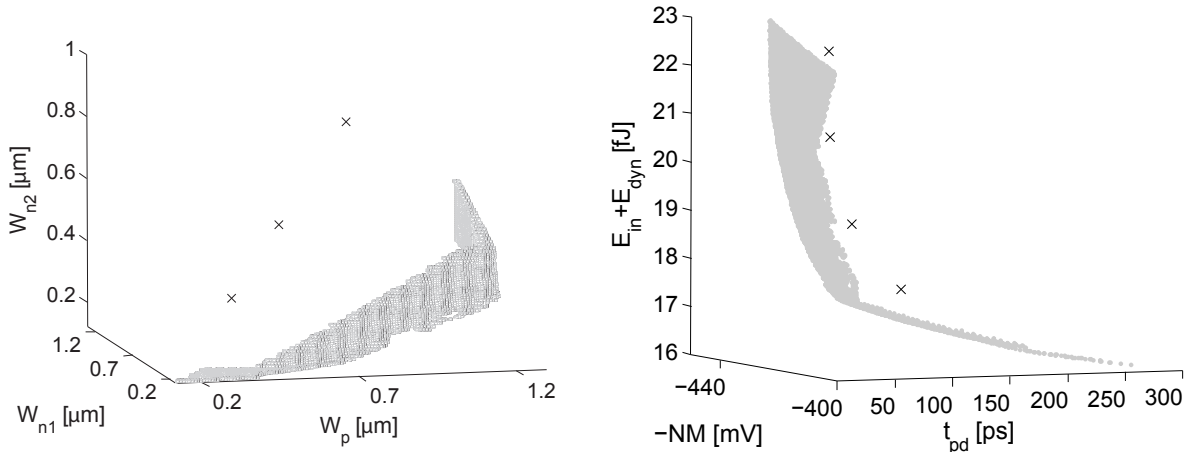


(a) 90nm CMOS-NAND-Gatter



(b) 90nm CMOS-NOR-Gatter

Abbildung 2.18: Approximierte Pareto-Mengen und Fronten mit Referenzpunkten kommerzieller Gatter (vgl. [67])



(a) Suchraum (b) Bildraum

Abbildung 2.19: Pareto-Menge und -Front des 65nm CMOS-NAND-Gatters mit Referenzpunkten (vgl. [67])

Komplexere Gatter

Mit mehr Transistoren erhöht sich sowohl die Komplexität des Suchraums als auch das Verbesserungspotential gegenüber kommerziellen Schaltungen. Der Energieverbrauch eines *And-Or-Invert*-Gatters, dessen Netzliste im folgenden Kapitel in Abbildung 3.14a gegeben ist, kann gegenüber kommerziellen Referenzschaltungen um bis zu 13 % verbessert werden, in den Störabständen können etwa 4 % erreicht werden (s. Tabelle 2.4). Ob nun alle NMOSFET und alle PMOSFET in einer zwei-dimensionalen Suche zusammengefasst werden oder ob eine vier-dimensionale Suche angestoßen wird, bringt keine gravierenden Unterschiede in den Störabständen. Bei gleichen Störabständen können aber in der vier-dimensionalen Suche durch mehr Freiheiten Transistordimensionierungen mit niedrigerem Energieverbrauch gefunden werden. Auf eine grafische Darstellung wird hier verzichtet. Die Bilder beider Suchen ähneln denen in Abbildung 2.16 dargestellten Pareto-Fronten des NOR-Gatters.

	W_1 [µm]	W_5 [µm]	t_{pd} [ps]	NM [mV]	$E_{\mathrm{in}} + E_{\mathrm{dyn}}$ [fJ]
AOI_X1	0,28	0,21	173,7803	431,8824	19,3873
PP1_AOI_X1	0,26	0,14	175,484	444,3232	18,6367
PP2_AOI_X1	0,26	0,12	173,8999	438,4917	18,4657
AOI_X3	0,55	0,39	112,0687	434,6306	22,9784
PP1_AOI_X3	0,53	0,32	112,843	442,9617	22,1886
PP2_AOI_X3	0,51	0,23	112,4175	432,3455	21,2956
AOI_X4	0,83	0,58	87,7816	428,5668	26,6774
PP1_AOI_X4	0,77	0,45	88,5593	441,9882	25,1558
PP2_AOI_X4	0,74	0,35	88,5025	430,734	24,0689
AOI_X6	1,10	0,78	75,8418	424,7445	30,3853
PP1_AOI_X6	0,99	0,55	76,5646	441,4839	27,5882
PP2_AOI_X6	0,95	0,44	76,2401	430,799	26,3588

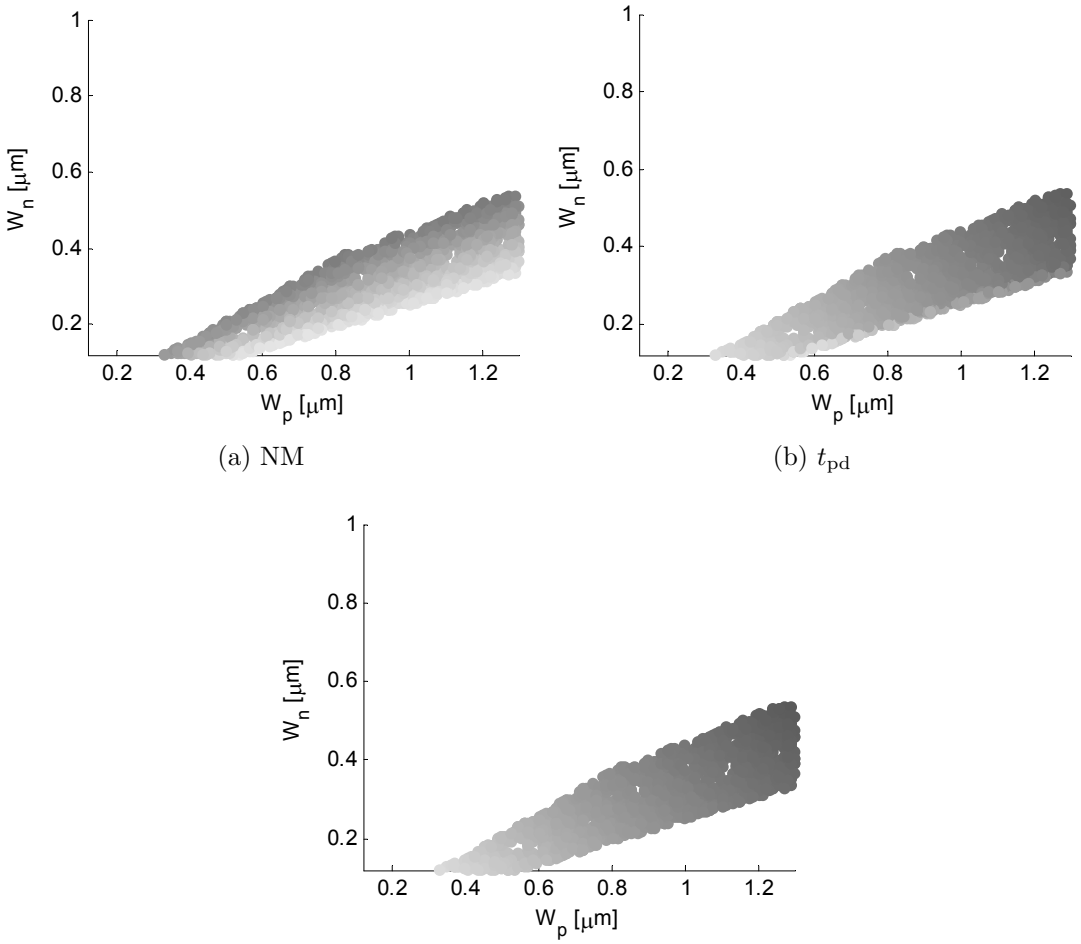
Tabelle 2.4: 65nm And-Or-Invert-Gatter einer kommerziellen Standardzellenbibliothek mit Pareto-Punkten

2.5 Entwurfszentrierung

Die bisherige Suche nach optimalen Entwurfspunkten stützte sich ausschließlich auf Simulationsergebnisse unter der Annahme typischer Umgebungsbedingungen. Beim Entwurf müssen allerdings zu erwartende störende Einflüsse berücksichtigt werden, um bei der anschließenden Fertigung eine möglichst hohe Ausbeute an funktionierenden Schaltungen zu erzielen. Sogenannte *Monte-Carlo-Simulationen* bei denen die Verschiebung der Schalteigenschaften einer Schaltung unter Variation von streuenden Umgebungsvariablen untersucht wird, können im Anschluss an die algorithmische Suchraumexploration

durchgeführt werden. Bei analogen Schaltungen ist auf Grund der hohen Sensibilität eine Erweiterung der Algorithmen um eine Robustheitsanalyse während der Suchraumexploration sinnvoll (s. dazu Kapitel 5).

Für digitale Schaltungen reicht es aus zunächst eine Suchraumexploration ohne Berücksichtigung der Sensitivität durchzuführen und einzelne Entwurfspunkte in einem darauf folgenden Schritt auf ihre Robustheit zu testen, da die Abhängigkeit der Robustheit von den freien Parametern für diese Schaltungen leichter kalkulierbar ist. Abbildung 2.20 zeigt die eingefärbte Pareto-Menge des bereits diskutierten 65nm CMOS-NOR-Gatters. Je heller die Entwurfspunkte eingefärbt sind, desto mehr schwanken die Zielgrößen t_{pd}, NM und $E_{\mathrm{in}} + E_{\mathrm{dyn}}$ mit sich ändernden Temperaturen und Versorgungsspannungen. Wie zu erwarten sind Schaltungen mit größeren Transistoren weniger störanfällig als Schaltungen mit kleineren Strukturen. Dieses Ergebnis ist auch in [5] veröffentlicht.



(a) NM

(b) t_{pd}

(c) $E_{\mathrm{in}} + E_{\mathrm{dyn}}$

Abbildung 2.20: Pareto-Menge des 65nm CMOS-NOR-Gatter eingfärbt nach der Sensitbilität der Entwurfspunkte gegenüber Temperatur- und Versorgungsspannungsschwankungen. Helle Punkte sind sensibler als dunkle. (vgl. [5])

Kapitel 3

Optimierung einer Subschwellbibliothek

Die Reduktion der Versorgungsspannung ist die effektivste Möglichkeit die Verlustleistung einer Schaltung zu verringern und diese gleichzeitig vollständig aktiv zu halten und nicht wie beim *Clock-* oder *Powergating* ganze Blöcke abzuschalten. Der Betrieb integrierter Schaltungen mit einer Versorgungsspannung unterhalb der Transistorschwellspannung ($U_{\mathrm{dd}} < U_{\mathrm{TH}}$) erlaubt eine besonders große Energieersparnis. Subschwellschaltungen können daher in Systemen mit möglichst geringer Energieaufnahme Anwendung finden, bei denen eine geringere Geschwindigkeit akzeptabel ist. Diese Anforderung kann zum einen durch ein beschränktes Energiereservoir begründet sein wie es bei batteriebetriebenen Mikro-Sensor-Knoten, *Radio-Frequency-Identification* (RFID) oder anderen portablen Geräten mit Low-Power Digital-Signal-Prozessoren (DSP) oder Mikrokontrollern der Fall ist [74]. Zum anderen liegt oftmals das Energieoptimum einer Schaltung unterhalb der Schwellspannung [74]. Abbildung 3.1 zeigt skizzenhaft die Abhängigkeit der Verlustenergie von der Versorgungsspannung um und unterhalb der Schwellspannung U_{TH}. Die dynamische Energie

$$E_{\mathrm{dyn}} = \alpha_{0\to1} \cdot C_L \cdot U_{\mathrm{dd}}^2$$

(vgl. (1.2)) fällt quadratisch mit sinkender Versorgungsspannung. Die statische Verlustenergie

$$E_{\mathrm{stat}} = P_{\mathrm{stat}} \cdot T$$

dagegen steigt mit kleiner werdendem U_{dd}, da sich die statische Verlustleistung P_{stat} zwar leicht verringert, die Schaltzeit T sich aber exponentiell erhöht. Damit hat die gesamte verbrauchte Energie

$$E_{\mathrm{ges}} = E_{\mathrm{dyn}} + E_{\mathrm{stat}}$$

ein Minimum U_{opt}. Abhängig davon wie groß bei einer konkreten Schaltung der Einfluss der Leckströme ist und abhängig von der Schaltaktivität $\alpha_{0\to1}$ einer Anwendung verschiebt sich das Energieoptimum. In [78] liegt das Energieoptimum einer Inverterkette unterhalb der Schwellspannung. Eine genauere Analyse kann in [77] gefunden werden.

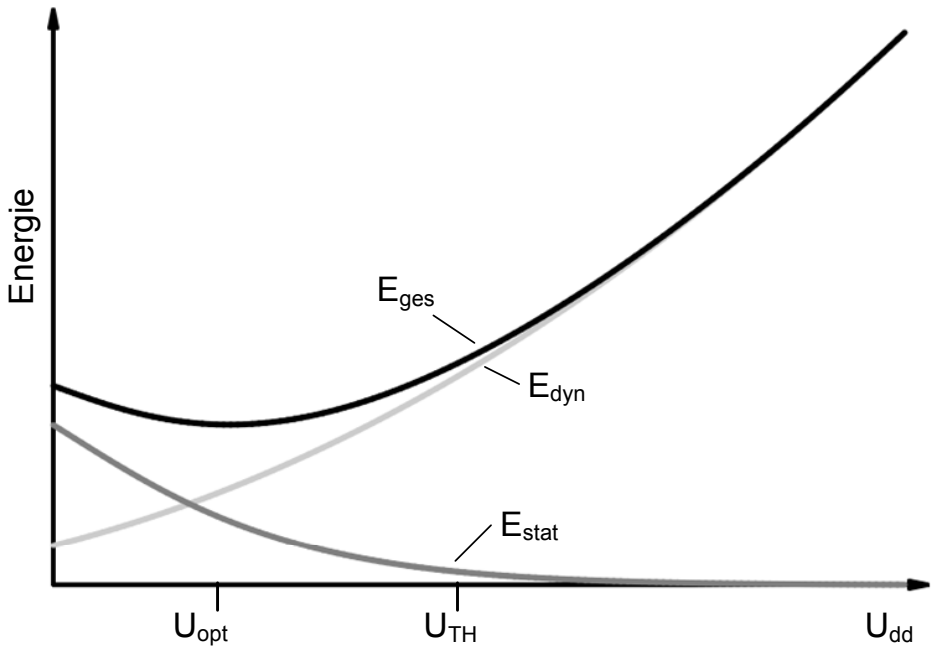


Abbildung 3.1: Illustration des Energieoptimums. Zusammenfassung von Beispielen u.a. aus [74]

3.1 Neue Herausforderungen für den Subschwellbetrieb

Erste Schritte Richtung Subschwellentwurf machten Swanson, Meindl [69] und Troutman [72] bereits Anfang der 1970er Jahre. Mittlerweile lassen sich Subschwellschaltungen auch in den klassischen standardzellenbasierten Entwurfsablauf einbinden [75, 45, 33]. Bisweilen verfügbare kommerzielle Standardzellbibliotheken sind allerdings nur für volle Versorgungsspannung ausgelegt und für den Betrieb mit sehr niedrigen Spannungen nicht geeignet, denn Abbildung 3.2 zeigt wie sich die Störabstände kommerzieller Inverter mit abnehmender Versorgungsspannung dramatisch verschlechtern. Bei $U_{\mathrm{dd}} = 300\,\mathrm{mV}$ liegt der relative Störabstand unter 30 %. Mit $U_{\mathrm{dd}} = 200\,\mathrm{mV}$ führt schon ein Rauschen von 40 mV zu einer Missinterpretation der logischen Zustände. Desweiteren zeigt Abbildung 3.3, dass eine Mehrzieloptimierung (nach NM, t_{pd} und $E_{\mathrm{in}} + E_{\mathrm{dyn}}$) für verschiedene Versorgungsspannungen zu vollständig verschiedenen Ergebnissen führt: Die für 1,2 V Pareto-optimalen Entwurfspunkte werden bei 300 mV dominiert. Erste Messergebnisse für eine 32bit-ALU in 90nm CMOS sind in [51] zu finden. Diese ist für den Betrieb im Subschwellbereich ohne die verwendeten Standardzellen algorithmisch zu optimieren. Für einen 65nm CMOS-32bit-RISC-core, der bei einer Versorgungspannung von $U_{\mathrm{dd}} = 300\,\mathrm{mV}$ betrieben werden soll, wird folgend der Entwurf einer mehrzieloptimierten Subschwellbibliothek dokumentiert. Teile dieses Kapitels sind veröffentlicht in [6].

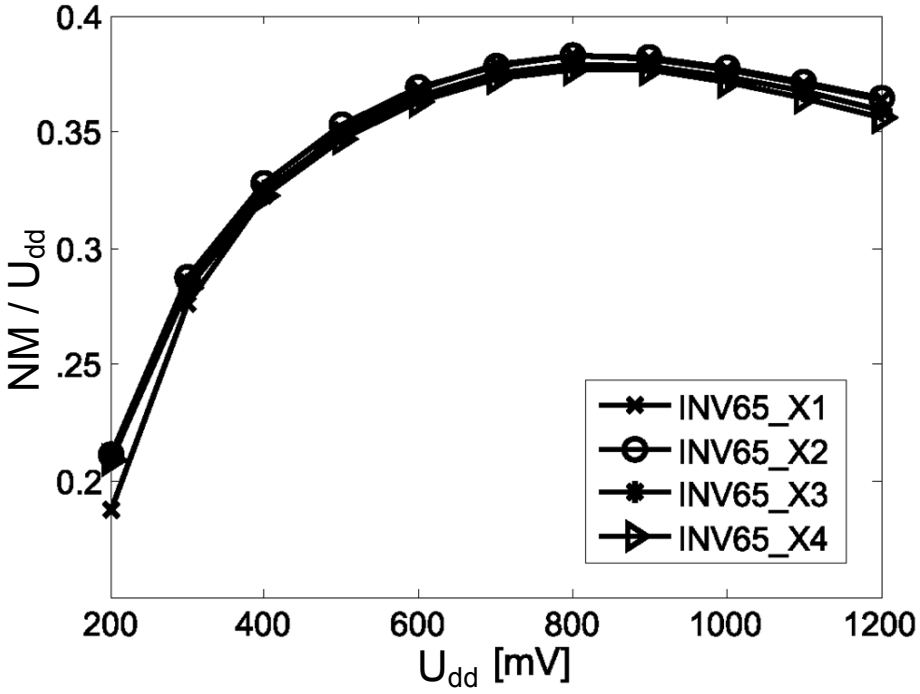


Abbildung 3.2: Relative Störabstände kommerzieller CMOS-Inverter in 65 nm von unterschiedlicher Treiberstärke (indiziert durch _X1 ... _X4) (vgl. [6])

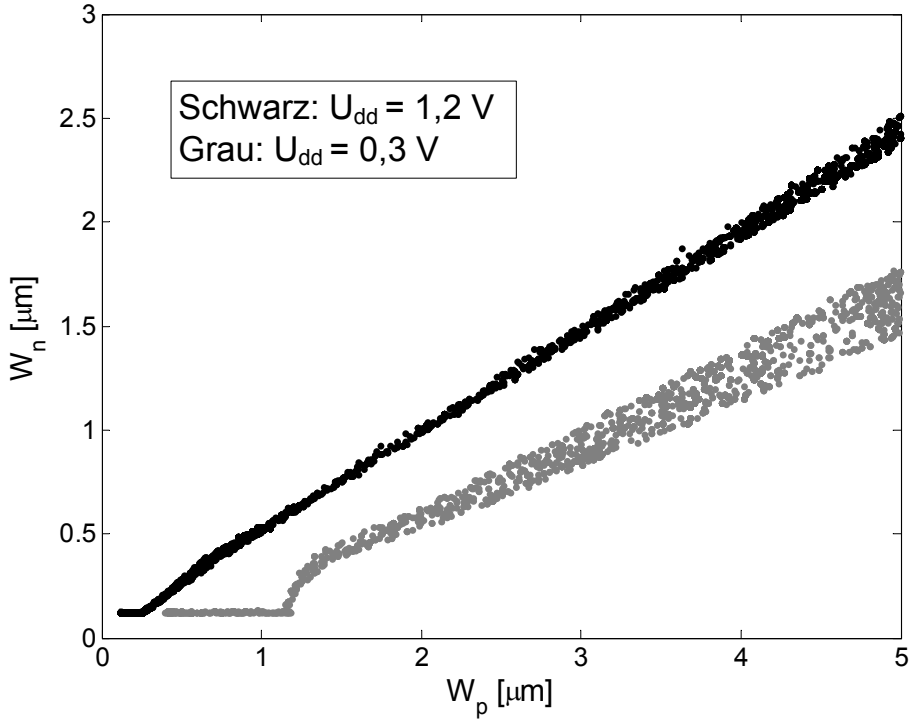


Abbildung 3.3: Pareto-Mengen des CMOS-Inverters in 65 nm bei festen Transistorlängen L_p=L_n=L_min bei $U_\mathrm{dd} = 1,2\,\mathrm{V}$ und $U_\mathrm{dd} = 0,3\,\mathrm{V}$ (vgl. [6])

3.2 Evolutionäre Algorithmen zur Mehrzieloptimierung

Um Pareto-Mengen zu approximieren, werden hier zusätzlich zu den im vorherigen Kapitel vorgestellten GAIO-Algorithmen *evolutionäre Algorithmen* (EA) eingesetzt. Ihr Vorteil ist, dass sie schon nach kurzer Laufzeit eine gute Abschätzung der Lage der Pareto-Menge im Surchraum liefern. Nachteilig ist, dass sie viele Parameter enthalten, die problemspezifisch vom Benutzer angepasst werden müssen. Um diese zu ermitteln bedarf es zumeist mehrerer Testläufe und Erfahrungswerte. Teile dieser Arbeit sind im Rahmen einer Diplomarbeit implementiert und in [12] dokumentiert worden.

Algorithmus 1 $A := Basic\_Evolutionary\_Algorithm\,(N, T, p_\mathrm{c}, p_\mathrm{m})$

01: **Initialize population:** $P_t = \emptyset, t = 0$
02: **for** $i = 1$ to N **do**
03: Find a random $j_i \in X$.
04: $P_t = P_t \cup \{j_i\}$
05: **end for**
06: **Fitness Evaluation:**
07: **for** $i = 1$ to N **do**
08: Calculate $F\,(j_i)$.
09: **end for**
10: **Selection:** $P' = \emptyset$, (P': mating pool)
11: **for** $i = 1$ to N **do**
12: Select one individual j, according to the selection method.
13: $P' := P' \cup \{j\}$
14: **end for**
15: **Recombination:**
16: **cross-over:** Choose two individuals randomly from P'. Apply cross-over operator on
17: them with a probability of p_c. Replace the (new) individuals in P'.
18: **mutation:** Choose one individual randomly from P'. Apply mutation operator on it
19: with a probability of p_m. Replace it in P'.
20: **Termination:** $P_{t+1} = P', t = t + 1$
21: **if** $t > T$ **then**
22: $A = P_t$, STOP
23: **else**
24: GOTO line 6
25: **end if**

EAs sind nach dem Vorbild der biologischen Evolution gestaltet. Sie bestehen klassischerweise aus den Schritten *Fitnessbewertung*, *Selektion* und *Rekombination* (letztere besteht wiederum aus *Crossover* und *Mutation*), in denen aus einer Population $\mathrm{P_t}$ eine folgende $\mathrm{P_{t+1}}$ entsteht. Algorithmus 1 (Pseudocode nach [53]) zeigt die wesentlichen Schritte: Bei

gegebener Populationsgröße N, Crossover- und Mutationswahrscheinlichkeiten p_c und p_m wird nach T Schritten eine Population $A = P_{T+1}$ ausgegeben.

Zitzler und Thiele [82] untersuchten 1999 verschiedene evolutionäre Mehrzieloptimierungsalgorithmen und veröffentlichten einen eigenen unter dem Namen *Strength Pareto EA* (*SPEA*). Bei diesem Ansatz wird jedem *Individuum* einer *Population* basierend auf der Dominanzrelation ein *Fitnesswert* S zugeordnet.[1] Dieser wird in einem nachfolgenden Turnier darüber entscheiden, ob ein Individuum für eine nächste Generation erhalten bleibt oder verworfen wird.

Abbildung 3.4 veranschaulicht die Ermittlung des Fitnesswerts: Für nicht-dominierte Individuen entspricht der Fitnesswert der Anzahl der Individuen, die von diesem dominiert werden, geteilt durch die Anzahl aller dominierten Individuen plus eins (hier $7 + 1 = 8$). Allen dominierten Individuen wird die Summe der Fitnesswerte der Individuen zugewiesen, von denen sie dominiert werden plus eins. Kleinere Werte sind Indikator für einen kleineren Abstand zur Pareto-Front oder größerem Abstand zu anderen Individuen. Daher werden im Turnier kleinere Fitnesswerte bevorzugt. Diese auf Pareto-Dominanz basierende Fitnesszuweisung wurde nach dem Vorbild eines genetischen Algorithmus in [29] eingeführt.

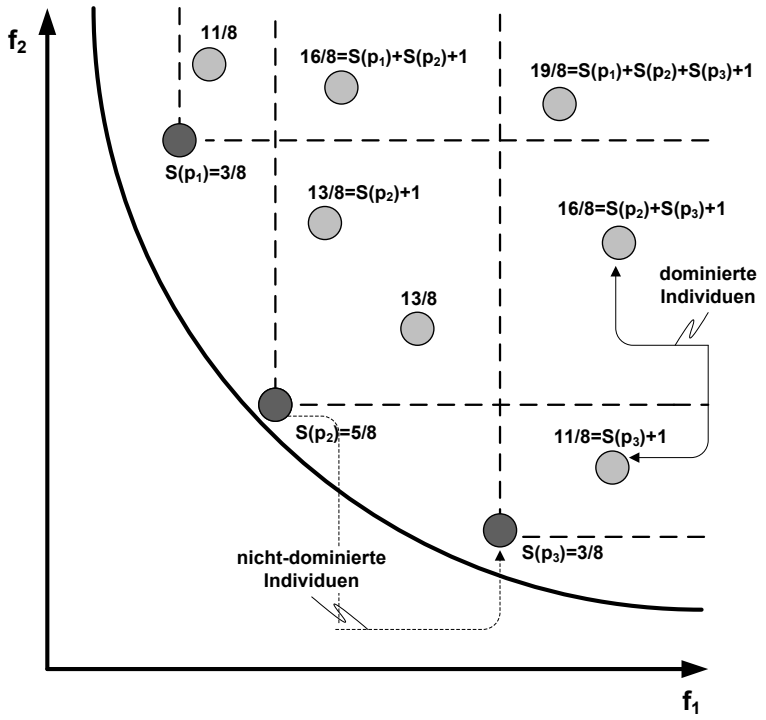


Abbildung 3.4: Beispiel für die Fitnessbewertung des SPEA

Nach Finessbewertung und Turnier folgen Crossover und Mutation. Die hier verwendeten

[1] Die Bezeichnung des Fitnesswerts mit S ist aus [82] übernommen. Wann S in dieser Arbeit den Suchraum bezeichnet ist, wird aus dem Kontext ersichtlich sein.

Operatoren sind aus einer Vielzahl möglicher Methoden gewählt worden, die sich für die Optimierung der betrachteten Standardzellen als erfolgreich erwiesen haben. Beim Crossover zwischen zwei zufällig ausgewählten Individuen (*Eltern*) p_1 und p_2 werden *Kinder*

$$\begin{aligned} c_1 &= (1-u)\,p_1 + up_2 \\ c_2 &= up_1 + (1-u)\,p_2 \end{aligned} \tag{3.1}$$

durch eine gleichverteilte Zufallsvariable $u \in [-0.5, 1.5]$ bestimmt. Die Eltern werden in der Population mit der gegebenen Crossoverwahrscheinlichkeit p_c durch die Kinder ersetzt. Der Vorteil dieser Crossovermethode ist, dass durch (3.1) zum einen Kinder mit gleicher Wahrscheinlichkeit innerhalb oder außerhalb der bisher ermittelten Population bestimmt werden, was eine große Diversität in den Lösungen hervorbringt. Zum anderen wird aber auch sicher gestellt, dass der Mittelwert zwischen den Eltern weiterhin erhalten bleibt und sich die Kinder kontrolliert über den Suchraum verteilen.

Bei der Mutation wird, wie in [17], auf jedes Individuum p einer Population ein Störwert addiert:

$$p := p + \delta \cdot \frac{b-a}{2},$$

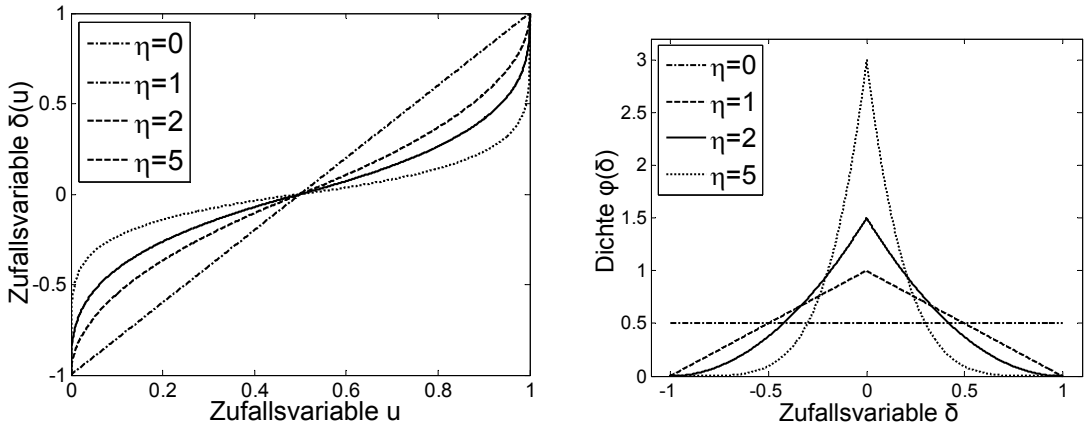
wobei die Zufallsvariable $\delta = \delta\,(u)$ bestimmt wird durch

$$\delta\,(u) = \begin{cases} (2u)^{\frac{1}{\eta+1}} - 1, & ; u \leq 0.5, \\ 1 - [2\,(1-u)]^{\frac{1}{\eta+1}}\,, & ; \text{sonst.} \end{cases}$$

a und b bezeichen den vektorwertigen unteren und oberen Rand des Suchraums. u ist wiederum eine gleichverteile Zufallsvariable, hier über dem Intervall $[0, 1]$. Somit lässt sich die Varianz von δ durch die Wahl des Skalars η regulieren. Abbildung 3.5b zeigt die Dichtefunktion

$$\varphi(\delta) = \frac{1}{2}(\eta+1)(1-|\delta|)^{\eta}.$$

zu verschiedenen Zufallsvariablen aus Abbildung 3.5a.



(a) (b)

Abbildung 3.5: Dichtefunktion der Mutation

3.3 Entwurf einzelner Gatter für den Subschwellbetrieb

Unterhalb der Schwellspannung (in schwacher Inversion) ist die Stromkennlinie der Transistoren geprägt durch Einflüsse zweiten und dritten Grades. Bei der Suche nach einer optimalen Dimensionierung sind dabei Effekte, die abhängig von den Weiten und Längen der Transistoren sind, besonders zu berücksichtigen. Abbildung 3.6 zeigt den Drain-Source-Strom I_{DS} bei $U_{\mathrm{GS}} = U_{\mathrm{DS}} = 300\,\mathrm{mV}$ des NMOSFETs. Für kleine Transistoren unterscheidet sich die Kennlinie grundlegend vom Betrieb mit voller Versorgungsspannung. Der *Reverse-Short-Channel-Effect* (RSCE) sorgt dafür, dass der Strom mit Verlängerung des Kanals vom Minimalmaß zunächt ansteigt [52, 39]. Da die Schwellspannung abhängig ist von der Transistorweite, lässt sich bei niedriger Versorgungsspannung erkennen, dass sich auch die Abhängigkeit des Stroms für kleine Weiten umkehrt. Dieser Effekt wird *Reverse-Narrow-Channel-Effect* (RNCE) genannt (s. dazu [56]).

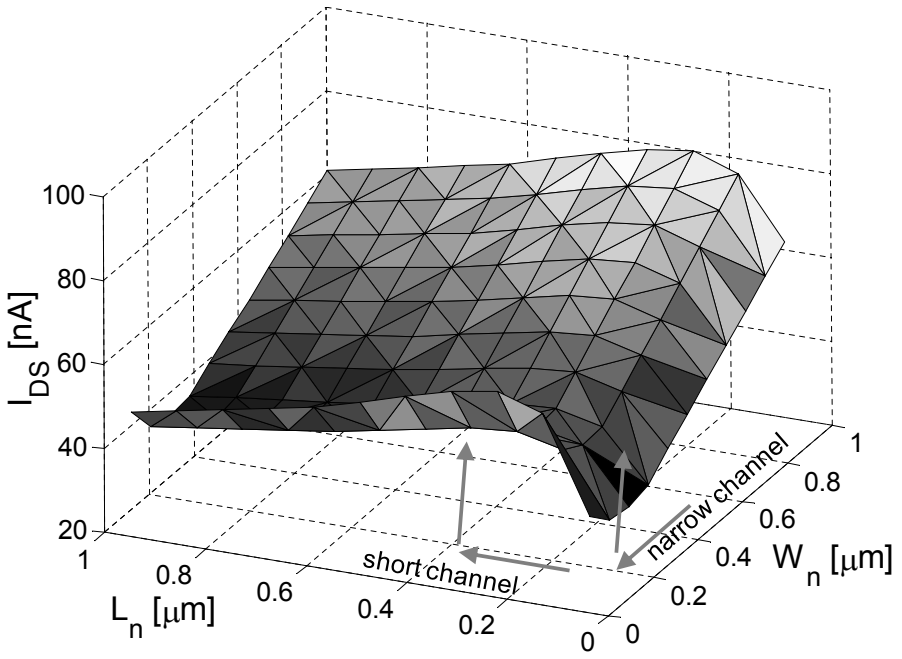
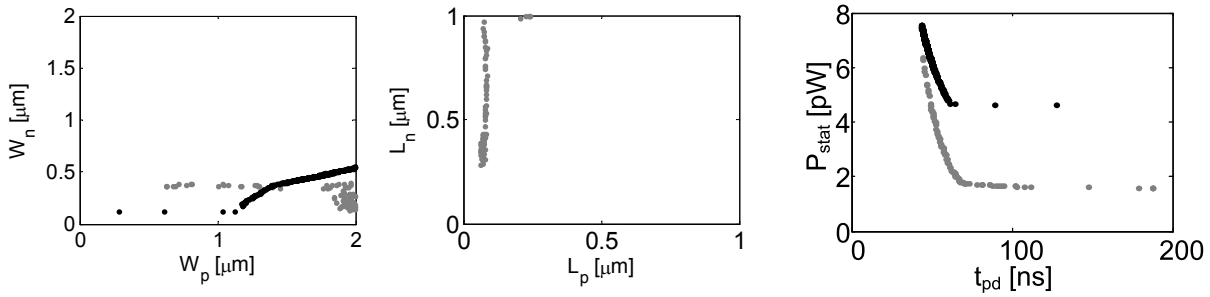


Abbildung 3.6: Drain-Source-Strom I_{DS} eines NMOSFET (65nm) betrieben mit $U_{\mathrm{dd}} = 300\,\mathrm{mV}$. Werte basieren auf HSPICE-Simulationen

Im Gegensatz zur Optimierung bei voller Versorgungsspannung (Kapitel 2) ist es nun nicht mehr sinnvoll, die Kanallängen von Anfang an auf Minimalmaß festzusetzen. Der RSCE lässt sich nutzen, um bessere Kompromisse zwischen konkurrierenden Ressourcen zu erreichen. Abbildung 3.7b zeigt Pareto-Fronten zu zwei MOPs. In beiden Fällen ist nach statischer Verlustleistung P_{stat} und Verzögerungszeit t_{pd} optimiert worden. In einem Fall wurden allerdings die Längen fixiert (schwarz), während in einer zweiten Konfiguration die Längen variabel sind (grau). Der Suchraum erweitert sich so von zwei auf vier Dimensionen, weshalb dieser in 3.7a in Weiten und Längen unterteilt abgebildet ist. Zu erkennen ist

im Bildraum, dass sich bei gleicher Verzögerungszeit die statische Verlustleistung nahezu dritteln lässt. An der Pareto-Menge sieht man, dass dieser Gewinn vor allem durch die freie Wahl der n-Kanallänge erreicht wird. Die optimale Länge des PMOSFET liegt um das Minimalmaß. Der Grund dafür ist, dass der RSCE beim PMOSFET in dieser Technologie nur sehr schwach ist.



(a) Suchraum (b) Bildraum

Abbildung 3.7: Approximation der Pareto-Menge des CMOS-Inverters. Schwarz: Längen auf Mimimum fixiert L_p=L_n=L_min. Grau: Längen variabel. (vgl. [6])

Eine Optimierung, bei der die Kanallängen der PMOSFET minimal und die der NMOSFET variabel gehalten würde, erscheint daher erfolgsversprechend zu sein. Da aber eine mit den Standardzellen gefertigte Schaltung auch bei höheren Versorgungsspannungen (*voltage scaling*) zuverlässig funktionieren soll, sollten die Längen beider Transistortypen gleich sein. Eine Näherung erster Ordnung für die Verzögerungszeit des CMOS-Inverters im Subschwellbereich ist gegeben in [74] (s. auch Anhang) durch

$$t_\mathrm{pd} = \frac{KC_\mathrm{g}U_\mathrm{dd}}{I_\mathrm{o}\exp\left(\frac{U_\mathrm{dd}-U_\mathrm{TH}}{nU_\mathrm{T}}\right)}$$

mit der Temperaturspannung U_T, dem *On*-Strom I_o, einem Subschwellfaktor $n = 1 + C_\mathrm{d}/C_\mathrm{ox}$ und einer Konstanten K. Die Standardabweichung der Schwellspannung ist nach [14] abhängig von den Transistormaßen:

$$\sigma(U_\mathrm{TH}) \sim \frac{1}{\sqrt{WL}}$$

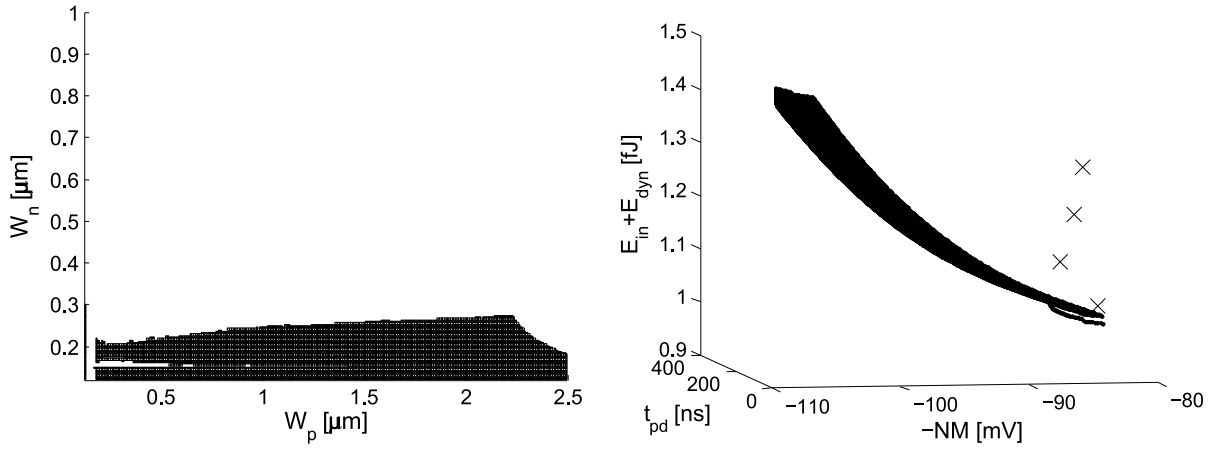
und verringert sich mit steigender Transistorgröße. Die Transistorlängen sollen daher alle gleich, aber nicht minimal gewählt sein. Für den weiteren Entwurf der Standardzellenbibliothek wird

$$L_\mathrm{p} = L_\mathrm{n} = 1.5L_\mathrm{min} = 90\,\mathrm{nm}$$

festgelegt, um einen Kompromiss zu finden. Auf diese Weise verkleinert sich auch die Größe des Suchraums, was die Laufzeit der Suchalgorithmen verbessert.

CMOS-Inverter, NAND- und NOR-Gatter

Abbildung 3.8a zeigt das Ergebnis der Mehrzieloptimierung für den CMOS-Inverter mit GAIO-Algorithmen. Die Pareto-Menge unterscheidet sich wesentlich von der im Kapitel 2 berechneten Lösung für volle Versorgungsspannung (vgl. Abbildung 2.12). Der Suchraum ist entsprechend der Anforderungen an die Standardzellen durch $W_\mathrm{p} \leq 2,5\,\mu\mathrm{m}$ und $W_\mathrm{n} \leq 1\,\mu\mathrm{m}$ abgegrenzt. Hier lässt sich keine lineare Abhängigkeit wie bei $U_\mathrm{dd} = 1,2\,\mathrm{V}$ erkennen: Die NMOSFET-Weiten aller ressourceneffizienten Inverter sind kleiner als 400 nm. Erst in einem größeren Suchraum ist ein Anstieg von W_n zu erkennen (vgl. Abbildungen 3.3, wo die Transistorlängen zunächst minimal gewählt wurden und 3.13a). Im Bildraum (Abbildung 3.8b) sind außer der Pareto-Front auch die Schalteigenschaften der kommerziellen Standardzellen aus Kapitel 2 (betrieben bei gleicher Versorgungsspannung) eingezeichnet.



(a) Suchraum (b) Bildraum

Abbildung 3.8: Pareto-Menge und -Front des 65nm CMOS-Inverters mit Referenzpunkten (Referenzpunkte nur im Bildraum eingezeichnet) (vgl. [6])

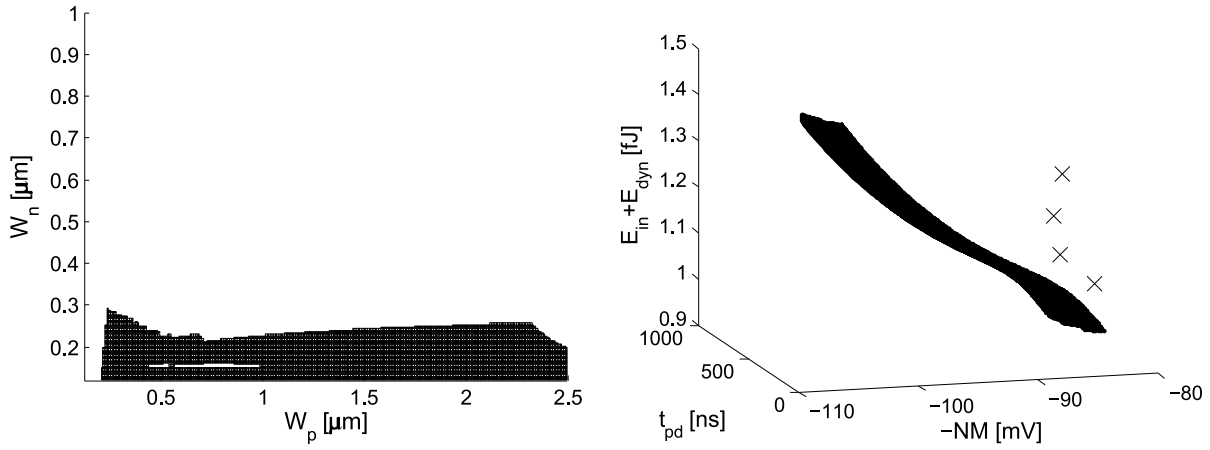
Diese Werte sind außerdem noch in Tabelle 3.1 nachzulesen. Dort sind auch Pareto-Punkte mit gleicher Treiberstärke als Vergleichsgrößen aufgeführt. Erkennbar ist vor allem eine starke Verbesserung in den Störabständen: Diese liegen zwischen 14 (PP1_INV65_X1) und 26 % (PP1_INV65_X3). Mit 300 mV Versorgungsspannung sind alle Pareto-Punkte mit über 90 mV NM relativ robust. Im Suchraum (Abbildung 3.8a) sind die Referenzpunkte nicht eingezeichnet, da sich die kommerziellen Inverter in den Transistorlängen unterscheiden. Die kleineren Transistoren führen bei gleicher Verzögerungszeit zu weniger Energieverlust, was die leicht besseren Werte der Referenzpunkte bezüglich $E_\mathrm{in} + E_\mathrm{dyn}$ in Tabelle 3.1 erklärt.

Das CMOS-NOR-Gatter liefert recht ähnliche Ergebnisse wie der Inverter. Abbildung 3.9a zeigt dessen Pareto-Menge und Abbildung 3.9b die entsprechende Pareto-Front. Auch hier sind die Referenzpunkte abgebildet und es sind ähnliche Verbesserungen abzulesen. Während die kommerziellen Gatter lediglich Störabstände von maximal 87 mV aufweisen,

	W_p [µm]	W_n [µm]	L [µm]	t_pd [ns]	$E_\mathrm{in} + E_\mathrm{dyn}$ [fJ]	NM [mV]
INV65_X1	0,28	0,2	0,06	128,953	1,0252	82,82
PP1_INV65_X1	0,665	0,23	0,09	128,4606	1,1109	94,13
INV65_X2	0,55	0,39	0,06	98,1677	1,1139	86,21
PP1_INV65_X2	1,13	0,25	0,09	97,4569	1,2045	99,98
INV65_X3	0,83	0,58	0,06	78,2787	1,2044	85,43
PP1_INV65_X3	1,685	0,265	0,09	77,7139	1,3135	104,85
PP2_INV65_X3	1,595	0,12	0,09	78,9478	1,2628	100,05
INV65_X4	1,1	0,78	0,06	66,5421	1,2948	84,9
PP1_INV65_X3	2,485	0,135	0,09	67,1836	1,4361	106,7
PP2_INV65_X3	2,115	0,12	0,09	67,0409	1,3624	103,66

Tabelle 3.1: 65nm CMOS-Inverter einer kommerziellen Standardzellenbibliothek mit Pareto-Punkten im Subschwellbereich

erreichen die Pareto-Punkte 107 mV. Der Aufbau der Testumgebung ist für Gatter mit zwei und mehr Eingängen entsprechend dem des vorherigen Kapitels.

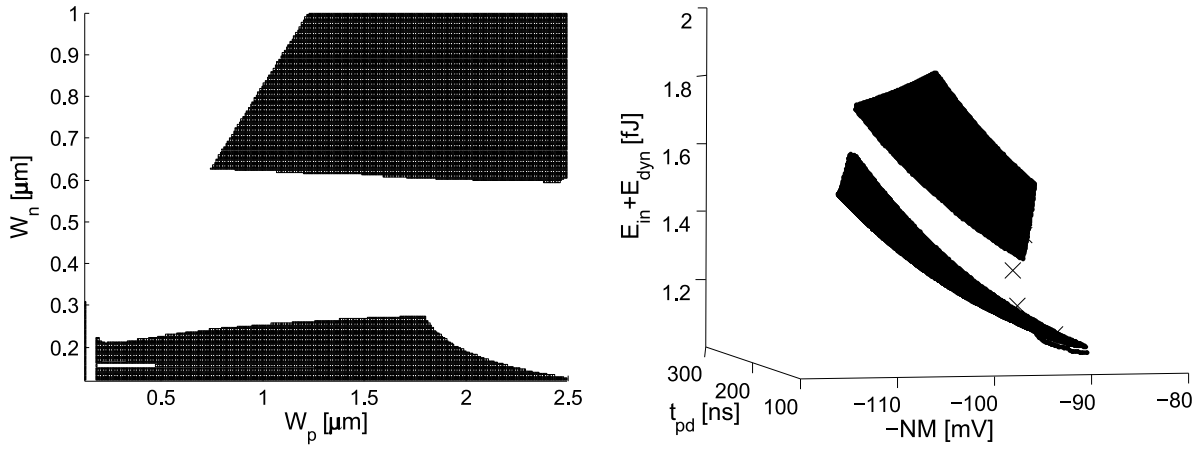


(a) Suchraum (b) Bildraum

Abbildung 3.9: Pareto-Menge und -Front des NOR-Gatters mit Referenzpunkten (Referenzpunkte nur im Bildraum eingezeichnet) (vgl. [6])

Die in Abbildung 3.10a gezeigte Pareto-Menge des NAND-Gatters unterscheidet sich stark von den Ergebissen des Inverters und des NOR, da sie disjunkt ist. Ebenso zerfällt die Pareto-Front in Abbildung 3.10b. Für den Bereich $0,3\,\mu\mathrm{m} \leq W_\mathrm{n} \leq 0,6\,\mu\mathrm{m}$ gibt es keine ressourceneffizienten Realisierungen des NAND-Gatters.

Der Grund hierfür erwächst aus dem Einfluss des RNCE der NMOSFETs auf die Verzögerungszeit t_pd. Abbildung 3.11 zeigt die Zielfunktionen des NAND-Gatters über dem Suchraum. Zu erkennen ist, dass t_pd genau in dem oben erwähnten Intervall von W_n einen Kamm bildet, während die anderen beiden Zielgrößen in ihrer Monotonie etwa wie bei voller Versorgungsspannung verlaufen. Der RNCE sorgt dafür, dass es Schaltungen mit



(a) Suchraum (b) Bildraum

Abbildung 3.10: Pareto-Menge und -Front des CMOS-NAND-Gatters mit Referenzpunkten (Referenzpunkte nur im Bildraum eingezeichnet) (vgl. [6])

gleicher Verzögerungszeit vor und hinter dem Kamm gibt, die sich aber in $E_{\mathrm{in}} + E_{\mathrm{dyn}}$ unterscheiden, da größere Transistoren mehr Verlustenergie verursachen.

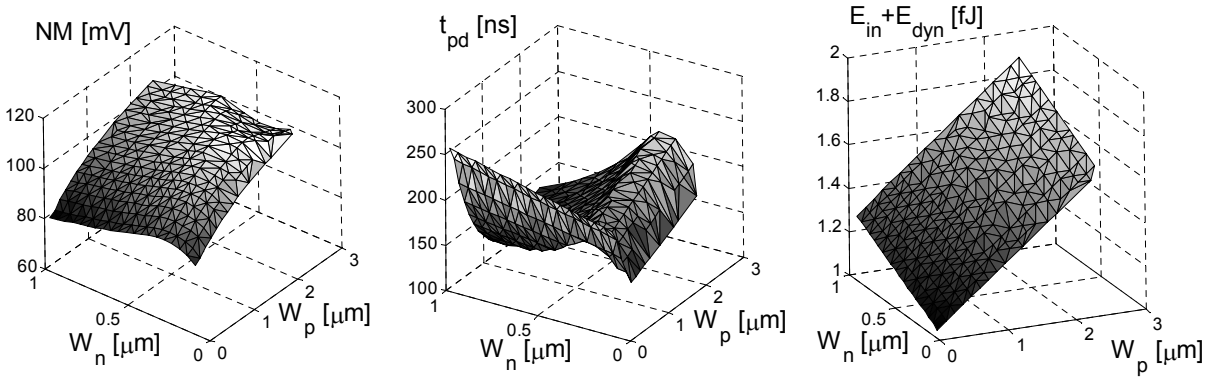


Abbildung 3.11: Zielfunktionen des CMOS-NAND-Gatters bei $U_{\mathrm{dd}} = 300\,\mathrm{mV}$

Wie die verschobene Zielfunktion t_{pd} die Form der Pareto-Menge beeinflusst, lässt sich besser noch an der Lage der Höhenlinien aller drei Zielfunktionen zueinander analysieren. In Abbildung 3.12 sind alle nicht-dominierten Punkte eines groben Rasters im Suchraum dargestellt. Sie indizieren die Lage der Pareto-Menge (vgl. Abbildung 3.10a). In verschiedenen Grautönen sind die Höhenlinien der drei Zielfunktionen eingezeichnet. Man erkennt deutlich den Kamm von t_{pd}, der etwa bei $W_{\mathrm{n}} = 0,3\,\mu\mathrm{m}$ verläuft. Die Höhenlinien der Störabstände verlaufen für $W_{\mathrm{p}} \geq 0,2\,\mu\mathrm{m}$ auch auf einem Kamm. (Dies ist ebenso bei voller Versorgungsspannung der Fall.) ebenfalls etwa bei $W_{\mathrm{n}} = 0,3\,\mu\mathrm{m}$ mit gleicher Monotonie. Somit konkurrieren diese zwei Größen kaum miteinander. Die meisten nicht-dominierten Punkte sind Kompromisse zwischen diesen und der Verlustenergie $E_{\mathrm{in}} + E_{\mathrm{dyn}}$. Zu Punkten im Suchraum mit $0,3\,\mu\mathrm{m} \leq W_{\mathrm{n}} \leq 0,6\,\mu\mathrm{m}$ existiert immer ein Punkt im Bereich $W_{\mathrm{n}} \leq 0,3\,\mu\mathrm{m}$ auf denselben Höhenlinien von NM und t_{pd}, der aber eine tiefere

Isobare von $E_{\text{in}} + E_{\text{dyn}}$ schneidet und somit Erstgenannten dominiert.

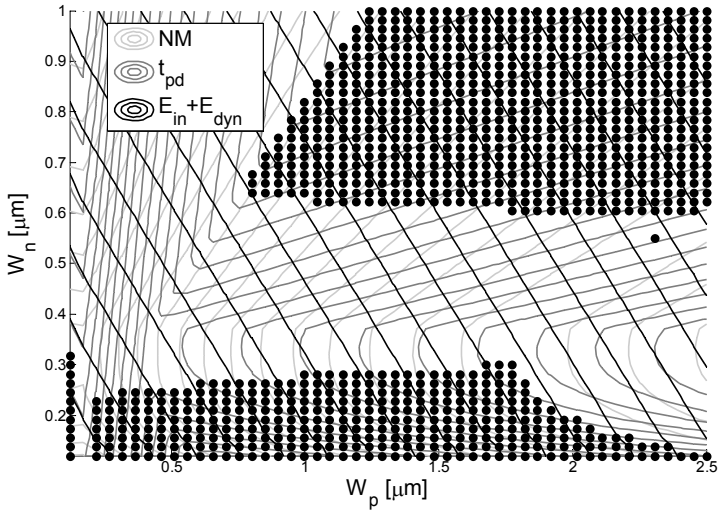
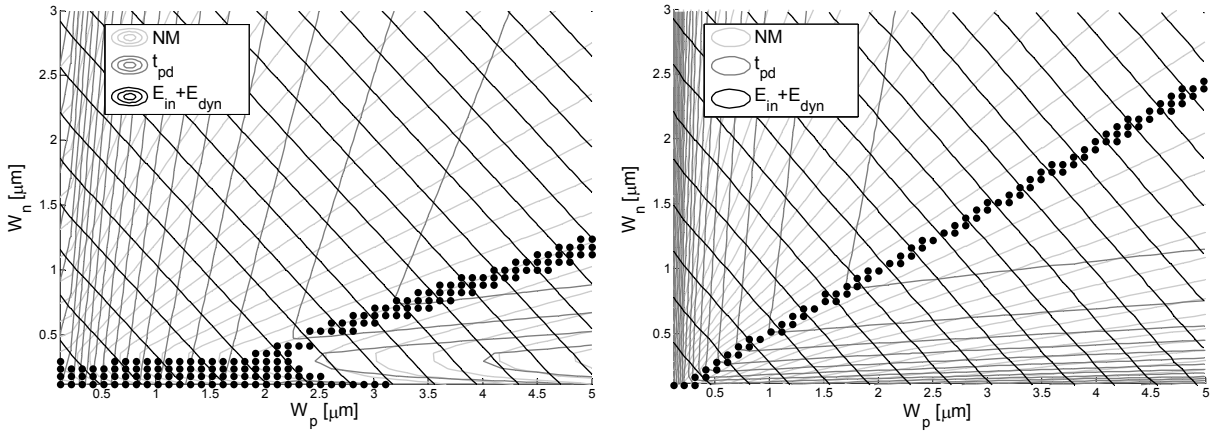


Abbildung 3.12: Höhenlinien des CMOS-NAND-Gatters bei $U_{\text{dd}} = 300\,\text{mV}$

Auch die Pareto-Menge des Inverters teilt sich in einem größeren Suchraum mit $U_{\text{dd}} = 300\,\text{mV}$. Die Lücke ist allerdings nicht allzu groß und bei dem in Abbildung 3.13a gewählten groben Raster an Testpunkten zunächst nur zu erahnen. Hier bildet sich erst ab $W_{\text{p}} \geq 2,5\,\mu\text{m}$, also kurz hinter dem zuvor untersuchten Bereich, ein Kamm in der Zielfunktion der Verzögerungszeit aus. Zum Vergleich ist in Abbildung 3.13b das Ergebnis bei $U_{\text{dd}} = 1,2\,\text{V}$ gezeigt: Die lokale Erhebung in t_{pd} ist wirklich auf Subschwelleffekte zurückzuführen. Hier wird nochmals deutlich, dass für den Entwurf von Standardzellen einer Subschwellbibliothek eine Suchraumexploration nur unterstützt durch Simulationsmodelle erfolgreich sein kann, da spezielle Effekte großen Einfluss auf die Lage und Form der Pareto-Menge nehmen.

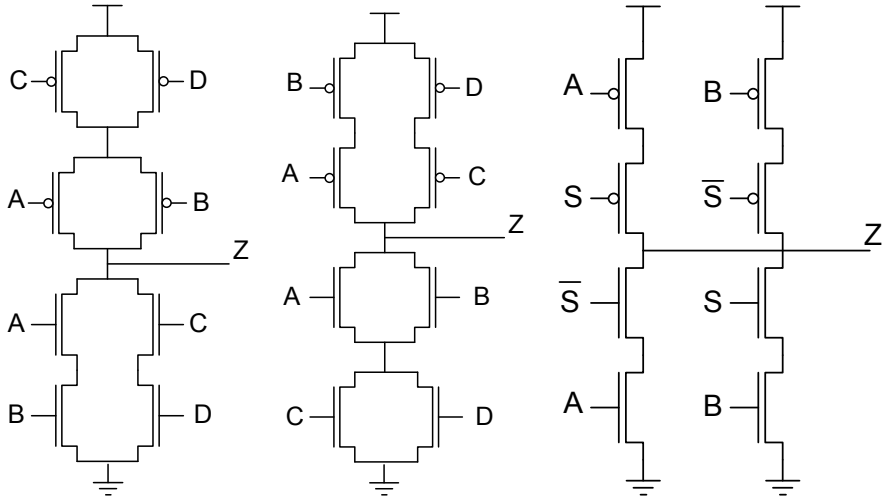


(a) $U_{\text{dd}} = 300\text{mV}$ (b) $U_{\text{dd}} = 1,2\text{V}$

Abbildung 3.13: Höhenlinien des CMOS-Inverters

Komplexere Gatter

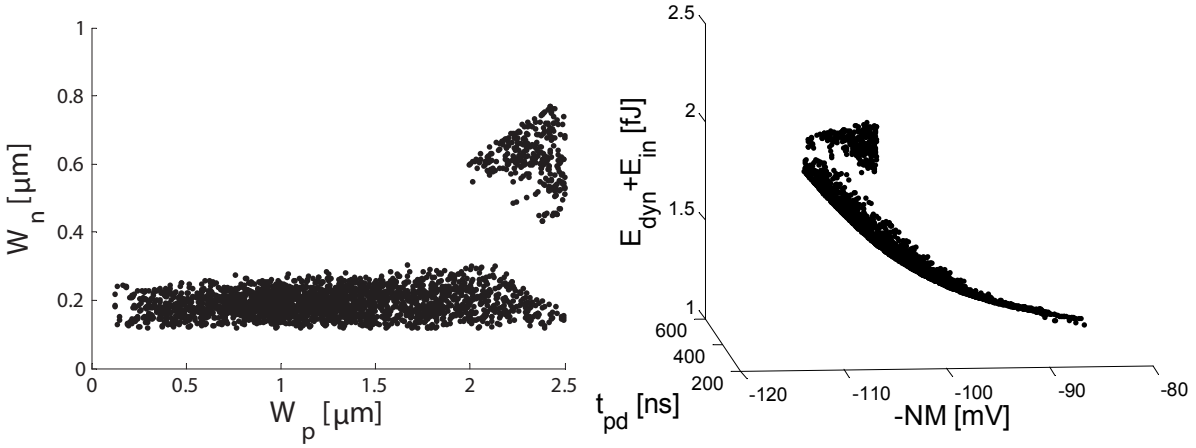
Durch Vermeidung komplizierter Zellen mit großen Stapeln von Transistoren wird Verlustleistung reduziert und Performanz verbessert [57]. Auch um im Subschwellbereich Funktionalität zu garantieren, wird in der Standardzellenbibliothek auf Gatter mit mehr als zwei Transistoren in Reihe verzichtet. Die gesamte Bibliothek umfasst 57 Elemente. Darunter befinden sich die oben beschriebenen Inverter, NAND- und NOR-Gatter, außerdem Buffer, Minority-3-Gatter, Flip-Flops und Latches. Folgend werden in aller Kürze Ergebnisse zur Transistordimensionierung dreier weiterer Zellen vorgestellt: Das *And-Or-Invert AOI* ($Z = \overline{(A \wedge B) \vee (C \wedge D)}$), das *Or-And-Invert OAI* ($Z = \overline{(A \vee B) \wedge (C \vee D)}$) und ein *Multiplexer MUX* ($Z = \overline{(A \wedge \overline{S}) \vee (B \wedge S)}$). Die Netzlisten sind in Abbildung 3.14 abgebildet.



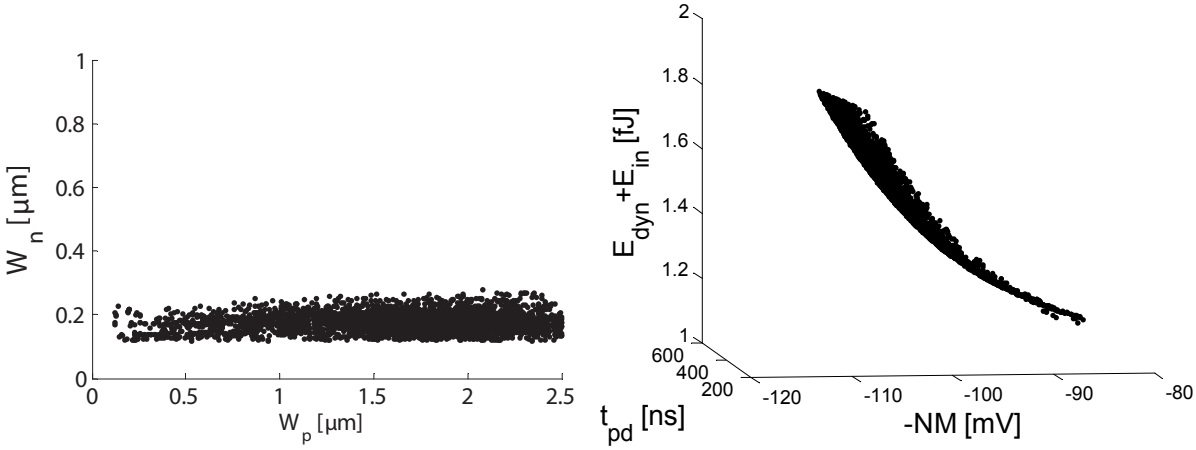
(a) And-Or-Invert AOI (b) Or-And-Invert OAI (c) Multiplexer MUX

Abbildung 3.14: Netzlisten

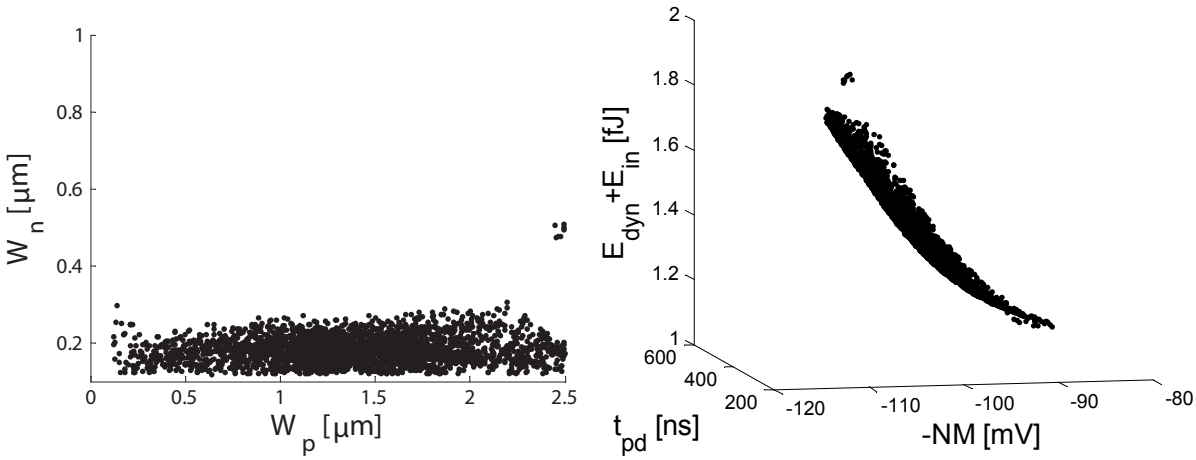
Alle Gatter wurden, wie im letzten Abschnitt, nach NM, t_{pd} und $E_{\mathrm{in}} + E_{\mathrm{dyn}}$ optimiert. Auch hier gilt für alle Kanallängen $L_{\mathrm{p}} = L_{\mathrm{n}} = 90\,\mathrm{nm}$. Abgebildet sind die Ergebnisse der Mehrzieloptimierung mit dem oben beschriebenen EA in Abbildung 3.15. Man erkennt bei allen Gattern ähnliche Eigenschaften, die auch schon bei vorherigen Gattern auffällig waren: Die Weiten der NMOSFET übersteigen erst eine Schwelle von ca. 300 nm, wenn die PMOSFETs entsprechend groß sind. Beim AOI ist das der Fall bei $W_{\mathrm{p}} \approx 2\,\mu\mathrm{m}$, beim OAI liegt diese Grenze außerhalb des gewählten Suchraums und beim Multiplexer scheint diese etwa bei $W_{\mathrm{p}} \approx 2,5\,\mu\mathrm{m}$ zu liegen, weshalb in Abbildung 3.15c schon vereinzelte nicht-dominierte Punkte mit $W_{\mathrm{n}} \geq 400\,\mathrm{nm}$ gefunden wurden.



(a) And-Or-Invert AOI



(b) Or-And-Invert OAI



(c) Multiplexer MUX

Abbildung 3.15: Approximierte Pareto-Mengen und -Fronten, berechnet mit evolutionären Algorithmen

3.4 Vergleich mit Standardzellen zur vollen Versorgungsspannung

Sowohl bei voller Versorgungsspannung als auch für den Subschwellbetrieb lassen sich in der betrachteten 65nm Technologie Inverter fertigen, deren Störabstände bis zu 37 % U_{dd} erreichen. Ein Vergleich von Abbildung 2.12 und Abbildung 3.8 lässt erkennen, dass durch unterschiedliche Wahl der Transistorweiten, relativ betrachtet, etwa gleich große Wertebereiche in den Ressourcen $E_{\mathrm{in}}+E_{\mathrm{dyn}}$ und t_{pd} abgedeckt werden: Schnelle Schaltungen sind in beiden Fällen etwa fünf- bis sechsmal schneller als langsame, während letztere ca. 50 % weniger Energie verbrauchen. Allerdings wird eine Abdeckung eines so weiten Wertebereichs im Subschwellbereich nur durch einen vergrößerten Suchraum erreicht. Die maximale Weite des PMOSFET ist hier 2,5 mal größer als beim Entwurf für $U_{\mathrm{dd}} = 1,2\,\mathrm{V}$. Optimierungsergebnisse liefern dagegen eine maximale Weite der NMOSFET von 300 nm.

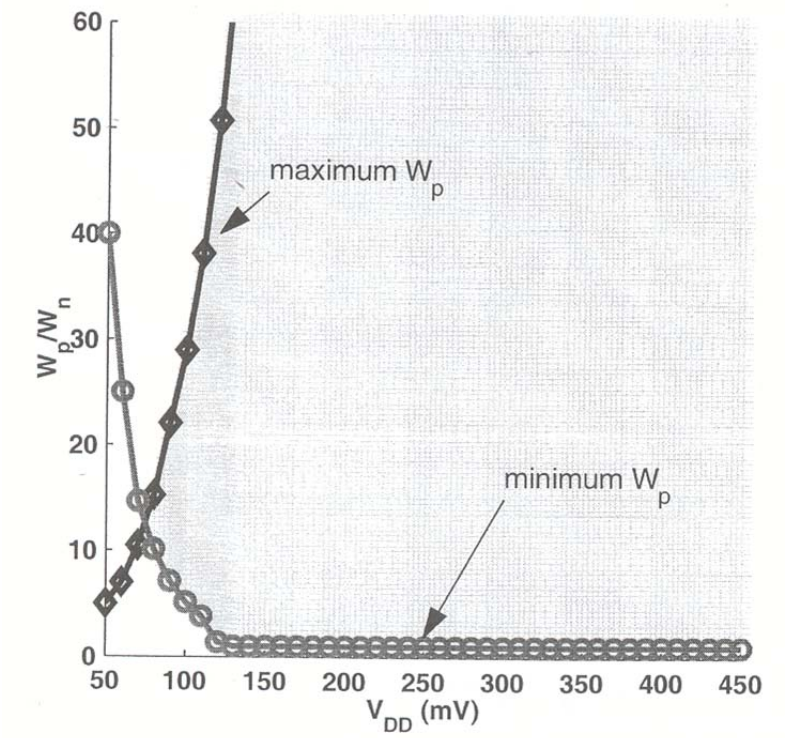


Abbildung 3.16: Simulationsergebnisse eines 0,18 µm Ringoszillartor. Minimale Versorgungsspannung (hier V_{DD}), die einen 10-90 % Ausschlag zulässt. (aus [74])

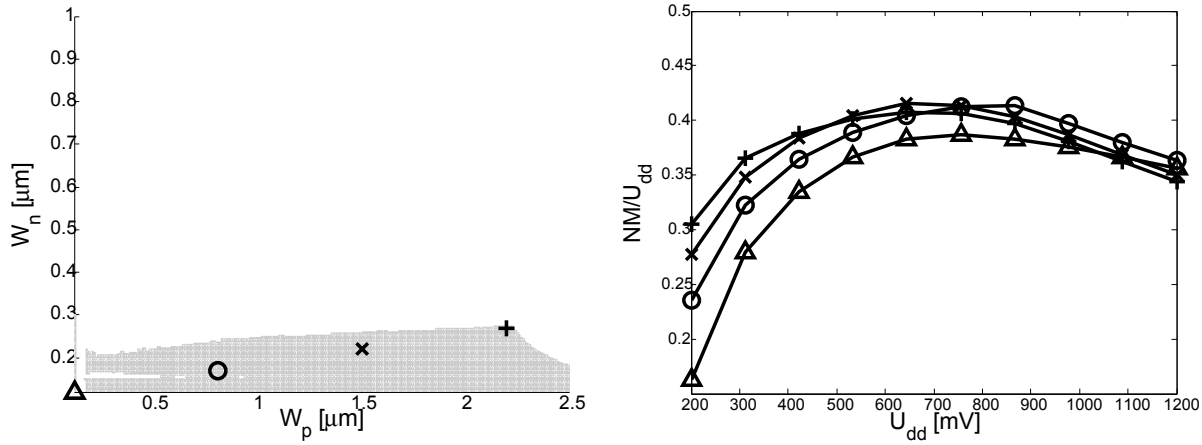
Somit ist das Verhältnis $W_{\mathrm{p}}/W_{\mathrm{n}}$ im Subschwellbereich stark verschoben gegenüber voller Versorgungsspannung. Diese Erkenntnis liefert auch eine weitere Analyse der Funktionalität von CMOS-Subschwellschaltungen. Wang, Colhoun und Chadrakasan [74] untersuchen bei einem 0,18 µm Ringoszillator bis zu welcher Versorgungsspannung am Ausgang noch ein Ausschlag von 10-90 % U_{dd} erzeugt wird. Sie stellen fest, dass dies im tiefen Subschwellbereich nur noch für ein eingeschränktes Weitenverhältnis der Fall ist. Links des

Kreuzungspunktes in Abbildung 3.16 ist entweder der NMOSFET zu schwach, um die Ausgangsspannung gegen den weiten PMOSFET auf 10 % abzusenken (maximum W_p) oder andersherum ist der PMOSFET zu schwach, um den Ausgang gegen den NMOSFET entsprechend auf 90 % anzuheben (minimum W_p). Unter Parametervariation kann sich der Kreuzungspunkt sogar von ca. 80 mV auf etwa 200 mV verschieben. Dieses Experiment verdeutlicht, dass bei niedrigen Versorgungsspannungen und kleinen Transistoren der PMOSFET wesentlich weiter als der NMOSFET sein muss.

Diese Erkenntnis deckt sich mit den Ergebnissen vom Inverter. Allerdings relativiert sich das $W_\mathrm{p}/W_\mathrm{n}$ Verhältnis für größere Transistoren. Während im Suchraum in Abbildung 3.8 noch Verhältnisse von bis zu $\frac{W_\mathrm{p}}{W_\mathrm{n}} = \frac{2{,}5}{0{,}12} \approx 21$ abzulesen sind, erkennt man im erweiterten Suchraum (Abbildung 3.13a), dass sich unter den Pareto-optimalen Invertern für $W_\mathrm{p} > 3$ ein Quotient von etwa 4,2 einstellt.

Die Bibliothek ist für den Subschwellbetrieb entworfen und optimiert. Ein mit ihr gefertigter IC soll trotzdem bei höheren Versorgungsspannungen zuverlässig funktionieren. (Aus diesem Grund wurden z.B. die Längen aller Transistoren gleich gewählt.) Messergebnisse des zu fertigenden 32bit RISC Prozessors werden erst mit der Dissertation von Lütkemeier [50] publiziert. Bisher ist eine 32bit-ALU in 90nm CMOS veröffentlicht worden (s. [51]).

Trotzdem kann am Beispiel des CMOS-Inverters, der für 300 mV U_dd dimensioniert wurde, gezeigt werden, dass dieser auch bei höheren Versorgungsspannungen gegenüber Rauschen der Logikpegel robust ist. In Abbildung 3.17b sind die relativen Störabstände der in Abbildung 3.17a gekennzeichneten Inverter dargestellt. Ein Vergleich mit den kommerziellen Standardzellen in Abbildung 3.2 zeigt deutlich höhere Werte im Subschwellbereich bei den meisten Pareto-Punkten. Auch bei $U_\mathrm{dd} = 1.2\,\mathrm{V}$ sind diese Inverter mindestens genau so robust wie die kommerziellen Referenzschaltungen.



(a) ausgewählte Inverter

(b) relative Störabstände dieser Inverter

Abbildung 3.17: Störabstände ausgewählter ressourceneffizienter CMOS-Inverter in Abhängigkeit von U_dd

Kapitel 4

Optimierung einer SRAM-Zelle für verschiedene Versorgungsspannungen

In diesem Kapitel wird der mehrkriterielle Optimierungsansatz zur Dimensionierung einer SRAM-Zelle in einer 65nm Technologie angewandt. Während das erste Ziel eine ressourceneffiziente Realisierung bei voller Versorgungsspannung $U_{\mathrm{dd}} = 1.2\,\mathrm{V}$ sein soll, wird weiterhin untersucht in wie weit Funktionalität bei sinkender Spannung garantiert werden kann. Bei einer angestrebten Versorgungsspannung von $U_{\mathrm{dd}} = 300\,\mathrm{mV}$ kann die klassische 6-Transistor-Zelle nicht mehr ausgelesen werden ohne eine Zerstörung des Speicherinhalts zu riskieren. Daher wird eine Erweiterterung auf neun Transistoren vorgestellt. Die Eigenschaften dieser Zelle werden bei beiden Versorgungsspannungen optimiert.

4.1 Optimierung der 6-Transistor-SRAM-Zelle

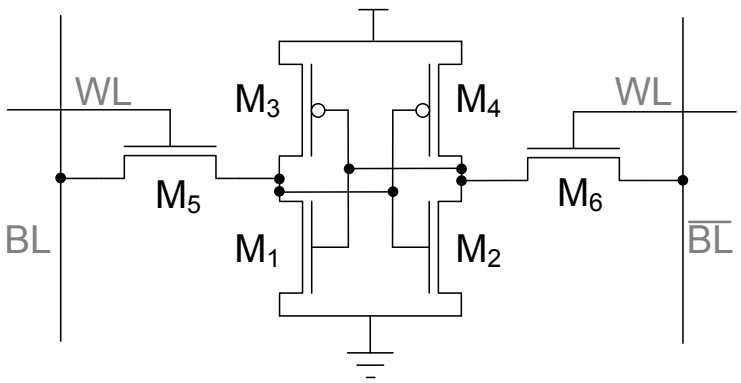


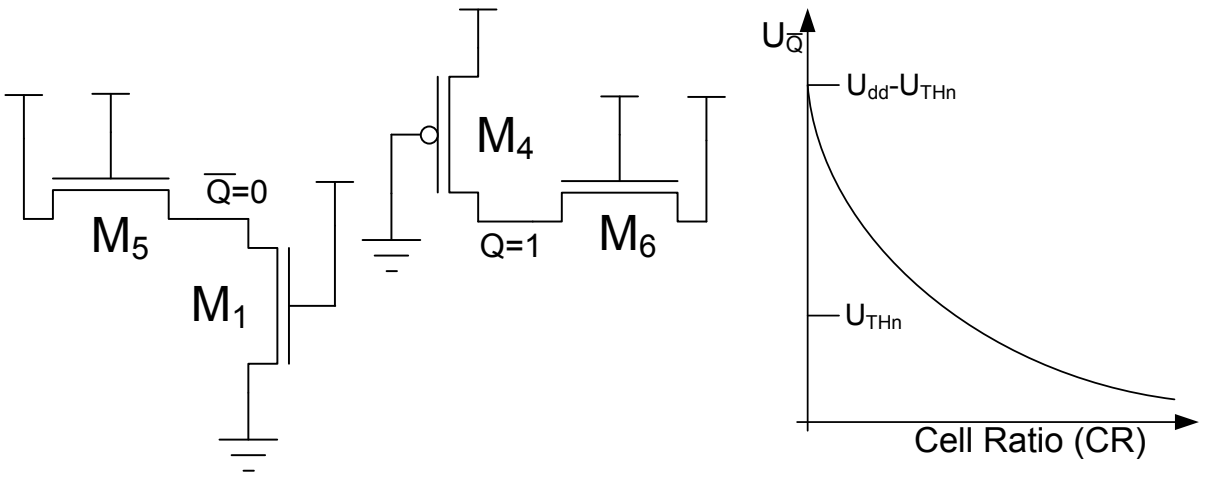
Abbildung 4.1: Netzliste der 6-Transistor-SRAM-Zelle mit Bezeichnern

Abbildung 4.1 zeigt die Netzliste einer traditionellen 6-Transistor-SRAM-Zelle. BL und $\overline{\text{BL}}$ bezeichnen die Bitleitungen, von denen ein logisches Bit und sein Inverses bei aktiver Wortleitung WL über die Transistoren M_5 und M_6 geschrieben werden. Zwei gekoppelte CMOS-Inverter (Transistoren M_1 - M_4) halten die geschriebenen Werte. Aus den Transistornamen lassen sich die Bezeichner ihrer Längen und Weiten ableiten (z.B. W_1 als Weite von M_1). Aufgrund der Symmetrie der Zelle werden im Folgenden die Transistorenpaare stets gleich dimensioniert ($W_1 = W_2$, $L_1 = L_2$, $W_3 = W_4$, ...).

Robustheit spielt beim Skalieren der Transistormaße eine zentrale Rolle. Geringe Störanfälligkeit beim Schreib- und Lesevorgang können als Zielgrößen für eine analytische Vorbetrachtung herangezogen werden, bevor anschließend durch eine simulationsbasierte algorithmische Suche optimale Kompromisse mehrerer Zielgrößen bestimmt werden.

4.1.1 Analytische Vorbetrachtung

Zu Beginn des Lesevorgangs sind beide Bitleitungen auf U_{dd} vorgeladen. Die Zelle sei vorgeladen mit $\text{Q} = U_{\text{dd}}$ und $\overline{\text{Q}} = 0$. Daher sperren der linke PMOS- M_3 und der rechte NMOS-Transistor M_2 der Zelle und es ergibt sich eine vereinfachte Netzliste, wie in Abbildung 4.2a dargestellt.



(a) Vereinfachte Netzliste

(b) Skizze des Funktionsgrafen zum Cell Ratio (CR)

Abbildung 4.2: Die 6-Transistor-SRAM-Zelle zu Lesebeginn

M_5 befindet sich in Sättigung und M_1 im Triodenbereich. Daher ergibt sich die Stomgleichung

$$\frac{B_{\text{n}}}{2}\frac{W_5}{L_5}\left(U_{\text{dd}} - U_{\overline{\text{Q}}} - U_{\text{THn}}\right)^2 = B_{\text{n}}\frac{W_1}{L_1}\left(U_{\text{dd}} - U_{\text{THn}} - \frac{U_{\overline{\text{Q}}}}{2}\right)U_{\overline{\text{Q}}}.$$

Zu Lesebeginn steigt daher die Spannung am Knoten $\overline{\mathrm{Q}}$ von 0 V auf

$$U_{\overline{\mathrm{Q}}} = U_{\mathrm{dd}} - U_{\mathrm{THn}} - \sqrt{\frac{\mathrm{CR}\left(U_{\mathrm{dd}} - U_{\mathrm{THn}}\right)^2}{1 + \mathrm{CR}}}, \tag{4.1}$$

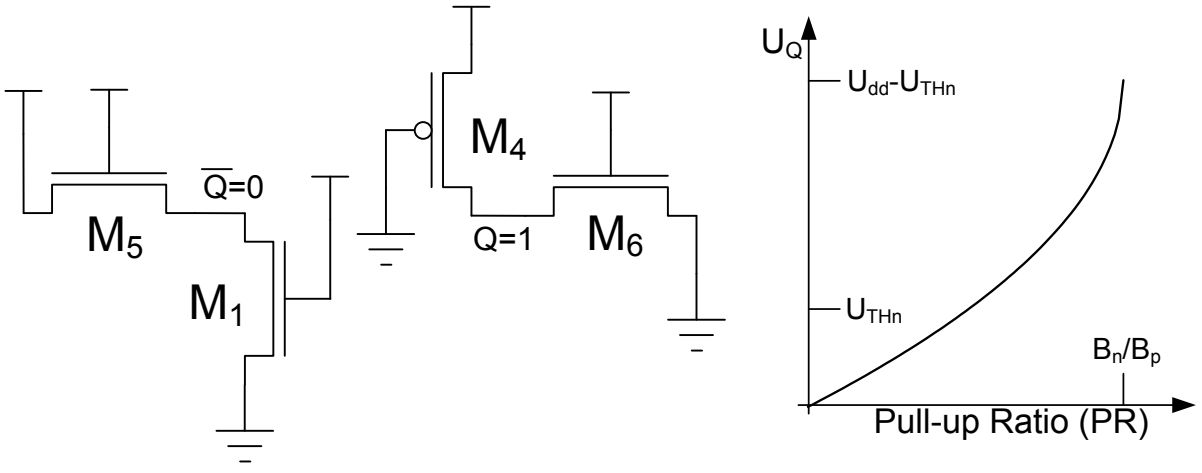
wobei CR, der *Cell Ratio* (auch bekannt als *Beta Ratio* [4]), definiert ist durch

$$\mathrm{CR} = \frac{W_1}{L_1} / \frac{W_5}{L_5}. \tag{4.2}$$

Beim Lesen soll die Spannung des Knotens der logischen Null (hier $\overline{\mathrm{Q}}$) nicht in die Nähe des Umschaltpunkts der Zelle angehoben werden. Abbildung 4.2b zeigt den Graphen von $U_{\overline{\mathrm{Q}}}$ abhängig von CR. Ein größerer Cell Ratio senkt den Spannungsanstieg und damit die Störanfälligkeit. Allerdings führt ein großer CR zu einer großen Fläche und zu längeren Schreibzeiten. Somit muss hier ein Kompromiss gefunden werden. Bleibt $U_{\overline{\mathrm{Q}}}$ unterhalb der Schwellenspnnung, sperrt der NMOSFET des rechten Inverters weiterhin und ein unbeabsichtigtes Umkippen des Zellwertes ist nicht zu befürchten. Für $U_{\overline{\mathrm{Q}}} = U_{\mathrm{THn}} \approx \frac{1}{3} U_{\mathrm{dd}}$ ergäbe sich

$$\mathrm{CR} = \frac{(U_{\mathrm{dd}} - 2U_{THn})^2}{2U_{\mathrm{dd}}U_{\mathrm{THn}} - 3U_{\mathrm{THn}}^2} \approx \frac{1}{3}. \tag{4.3}$$

Zum Schreiben werden die Bitleitungen jeweils invers zueinander vorgeladen. Wenn die Wortleitungstransistoren M_5 und M_6 geöffnet sind, wird der Zustand der Zelle durch die angelegten Spannungen überschrieben. Abbildung 4.3a zeigt die vereinfachte Netzliste zu Beginn des Schreibvorgans, bei der wiederum sperrende Transistoren ausgelassen sind.



(a) Vereinfachte Netzliste

(b) Skizze des Funktionsgrafen zum Pull-up-Ratio (PR)

Abbildung 4.3: Die 6-Transistor-SRAM-Zelle zu Schreibbeginn

Mit Beginn des Schreibvorgangs fällt die Spannung U_{Q} sprunghaft ab, so dass M_6 aus Sättigung in den Triodenbereich übergeht. Zum etwa gleichen Zeitpunkt ($U_{\mathrm{THn}} \approx -U_{\mathrm{THp}}$) wechselt M_4 vom Triodenbereich in Sättigung. Hier ergibt sich die Stromgleichung

$$B_{\mathrm{n}} \frac{W_6}{L_6} \left(U_{\mathrm{dd}} - U_{\mathrm{THn}} - \frac{U_{\mathrm{Q}}}{2}\right) U_{\mathrm{Q}} = \frac{B_{\mathrm{p}}}{2} \frac{W_4}{L_4} \left(-U_{\mathrm{dd}} - U_{\mathrm{THp}}\right)^2. \tag{4.4}$$

Der *Pull-up-Ratio* ist definiert durch

$$\mathrm{PR} = \frac{W_4}{L_4} / \frac{W_6}{L_6}, \tag{4.5}$$

und die Spannung an Q lässt sich aus (4.4) umformen zu

$$U_\mathrm{Q} = U_\mathrm{dd} - U_\mathrm{THn} - \sqrt{(U_\mathrm{dd} - U_\mathrm{THn})^2 - \mathrm{PR}\frac{B_\mathrm{p}}{B_\mathrm{n}}(U_\mathrm{dd} + U_\mathrm{THp})^2}. \tag{4.6}$$

Erstrebenswert ist eine möglichst kleine Spannung. Fällt U_Q weit unter $U_\mathrm{dd} + U_\mathrm{THp}$, müssten die zwei bisher ausgelassenen Transistoren berücksichtigt werden, die den Abfall noch beschleunigen würden. Trotzdem ist aus der Spannungskurve, wie sie in Abbildung 4.3b dargestellt wird, ableitbar, dass die SRAM-Zelle mit kleiner werdendem Pull-up-Ratio sicherer und schneller schaltet. Fällt PR unter U_THn, ist ein erfolgreiches Umkippen gesichert. Während W_5/L_5 für einen großen Cell Ratio noch klein gewählt werden sollte, würde für das Schreiben ein großer Wert bevorzugt. Für $U_\mathrm{Q} = U_\mathrm{THn} \approx -U_\mathrm{THp} \approx \frac{1}{3}U_\mathrm{dd}$ und $B_\mathrm{n}/B_\mathrm{p} \approx 1,4$ (vgl. Ratio beim 65nm CMOS-Inverter) würde sich ein Pull-up-Ratio

$$\mathrm{PR} = \frac{2U_\mathrm{dd}U_\mathrm{THn} - 3U_\mathrm{THn}^2}{\frac{B_\mathrm{p}}{B_\mathrm{n}}(U_\mathrm{dd} + U_{THp})^2} \approx 1,4 \cdot \frac{3}{4} = 1,05 \tag{4.7}$$

ergeben.

Auf der Grundlage der gemachten Vereinfachungen ließe sich die 6-Transistor-SRAM-Zelle wie folgt dimensionieren: Durch Vorgabe von Spannungen in den Gleichungen (4.1) und (4.6) werden der Cell-Ratio in (4.2) und der Pull-up-Ratio (4.5) festgelegt. Die Längen können mit Hinblick auf ein möglichst kleines Layout minimal gewählt werden. Durch Wahl einer einzigen Weite werden durch die Abhängigkeit

$$W_1 = \mathrm{CR} \cdot W_5 = \frac{\mathrm{CR}}{\mathrm{PR}} \cdot W_3 \tag{4.8}$$

alle weiteren Weiten festgelegt. Mit(4.3) und (4.7) ergibt sich aus (4.8)

$$W_1 \approx \frac{1}{3} \cdot W_5 \approx 0,3 \cdot W_3. \tag{4.9}$$

Schon bei der Optimierung der digitalen Standardzellen ließ sich die ungefähre Lage der Pareto-Menge der ressourceneffizienten Schaltungen durch ein lineares Verhältnis zwischen den Weiten der Transistoren beschreiben. Auch die Abschätzung (4.9) ist nicht allzu weit von optimalen Dimensionierungen entfernt.

4.1.2 Die Mehrzieloptimierungsaufgabe

Beim Entwurf integrierter Speicherbausteine gilt es, die Setup- und Hold-Zeiten sowie die Lesezeiten zu minimieren [20]. Daher werden auf Transistorebene als Zeitmaße die Verzögerungszeiten beim Schreiben t_write und Lesen t_read betrachtet. t_write ist definiert

als die Zeit, die vergeht während der sich die Spannung am Knoten Q oder $\overline{\mathrm{Q}}$ von $10\,\%\mathrm{U_{dd}}$ auf $90\,\%\mathrm{U_{dd}}$ ändert. Der Lesevorgang beginnt, wenn die Wortleitungstransistoren abrupt geöffnet werden und ist abgeschlossen, wenn sich die vorgeladenen Bitleitungen um mindestens $10\,\%\mathrm{U_{dd}}$ unterscheiden, so dass ein Leseverstärker das gespeicherte Bit zuverlässig bestimmen kann.

Da sich die einzelnen Speicherzellen statistisch zumeist im Halte-Zustand befinden und den gespeicherten Wert nur halten, hat die Standby Verlustleistung P_{leak}, die durch Leckströme während des Haltens eines Bits ensteht, einen erheblichen Anteil an der gesamten Verlustleistung der Schaltung. Beim Schreiben eines Bits wird durch Querstörme und Ladungsfluss von der Bitleitung auf die parasitären Kapazitäten der Transistoren eine dynamische Verlustleistung E_{write} induziert.

Die wichtigste Messgröße der Robustheit einer SRAM-Zelle wurde von Seevinck, List und Lohstroh [64] mit dem *statischen Störabstand* (engl.: static noise margin, SNM) eingeführt. Abbildung 4.4 zeigt die Schaltbilder der Simulationen für den Störabstand im Halte-Zustand $\mathrm{SNM_{hold}}$, während des Lesens $\mathrm{SNM_{read}}$ und Schreibens $\mathrm{SNM_{write}}$. Quantifiziert werden soll, wie schnell oder stark sich die SRAM-Zelle einer Störquelle widersetzen kann, um die stabilen Fixpunkte der Schmetterlingsgrafen (jeweils rechte Seite) $(1, 0)$ und $(0, 1)$ zu halten. Je weiter die Augen des Schmetterlingsgraphen geöffnet sind, desto schneller wird durch Rückkopplung der beiden Inverter der stabile Zustand wieder hergestellt. Bestimmt wird die Kantenlänge der größten Quadrate, die sich in die Augen einfügen lassen. Während bei $\mathrm{SNM_{hold}}$ und $\mathrm{SNM_{read}}$ möglichst große Werte erstrebt werden, garantieren für den Schreibvorgang möglichst kleine (und vor allem negative) $\mathrm{SNM_{write}}$-Werte ein sicheres Umkippen des Zellinhalts.

Mehrzieloptimierungsansatz bei voller Versorgungsspannung

Abbildung 4.5 zeigt die vorgestellten Schalteigenschaften der 6-Transistor-SRAM-Zelle für eine Versorgungsspannung von 1.2 V und feste Weite des Wortleitungstransistors. Es ist sofort erkennbar, dass auch hier, genau wie bei den Logik Standardzellen, einige Ressourcenmaße konkurrieren, während andere gleichzeitig optimale Werte annehmen. Folgend werden einige dieser Funktionen ausgewählt, um ein MOP zu formulieren. Ebenso wird durch Wahl der freien Parameter ein Suchraum abgesteckt.

Werden zunächst alle Transistorlängen festgesetzt auf $L_1 = L_3 = L_5 = L_{\mathrm{min}}$, so sind immer noch drei Weiten frei wählbar. Für einen ersten Ansatz soll ein zwei-dimensionales Suchproblem formuliert werden, da sich die Ergebnisse leichter nachvollziehen und grafisch darstellen lassen. Die in der analytischen Vorbetrachtung gemachten Ergebnisse der Robustheitsbetrachtung lassen erwarten, dass sich zum Verknüpfen der drei Weiten eine Linearkombination wie in (4.8) eignet. Mit den statischen Störabständen liegen drei mögliche Robustheitsmaße vor.

In Abbildung 4.6 sind die Höhenlinien des statischen Störabstands im Halte-Zustand zu verschiedenen Versorgungsspannungen abgebildet. Der $\mathrm{SNM_{hold}}$ ist nach Simulationsaufbau

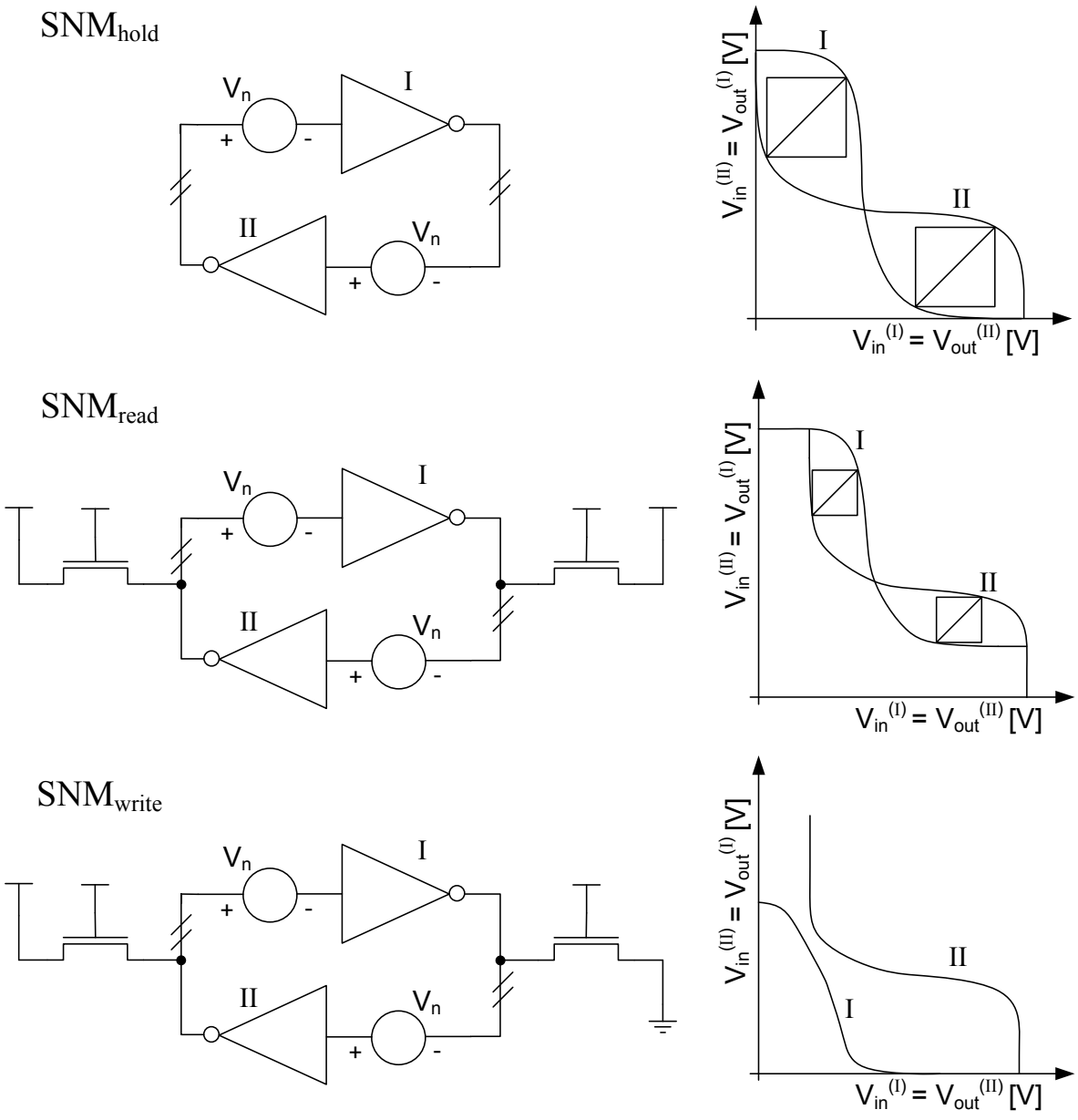


Abbildung 4.4: Definition der Störabstände beim Halten $\mathrm{SNM_{hold}}$ Lesen $\mathrm{SNM_{read}}$ und Schreiben $\mathrm{SNM_{write}}$ (vgl. [64])

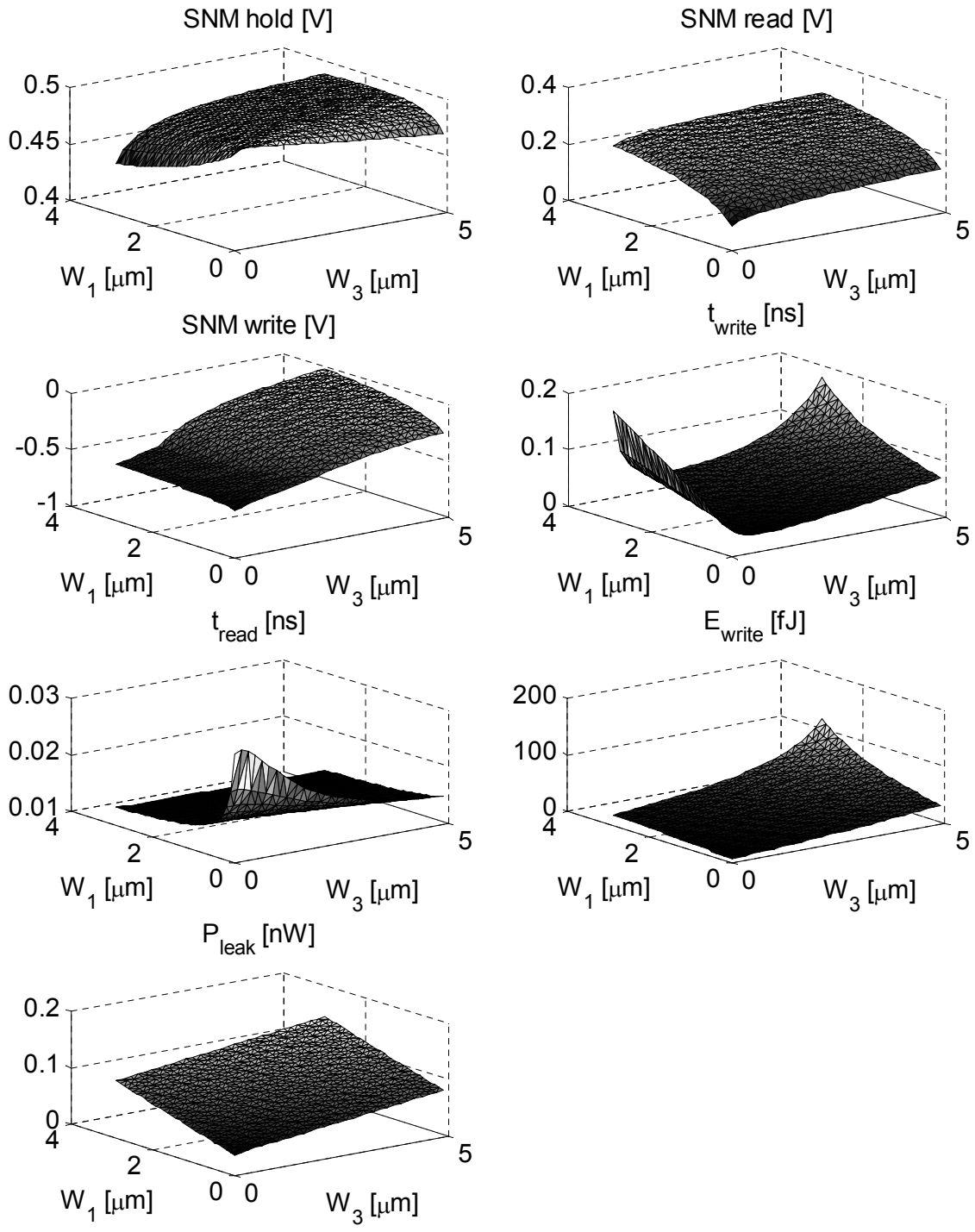


Abbildung 4.5: Schalteigenschaften der 6-Transistor-SRAM-Zelle in Abhängigkeit der Transistorweiten W_3 und W_1 bei $W_5 = 1\,\mu\text{m}$. (Längen minimal, HSPICE-Simulationen)

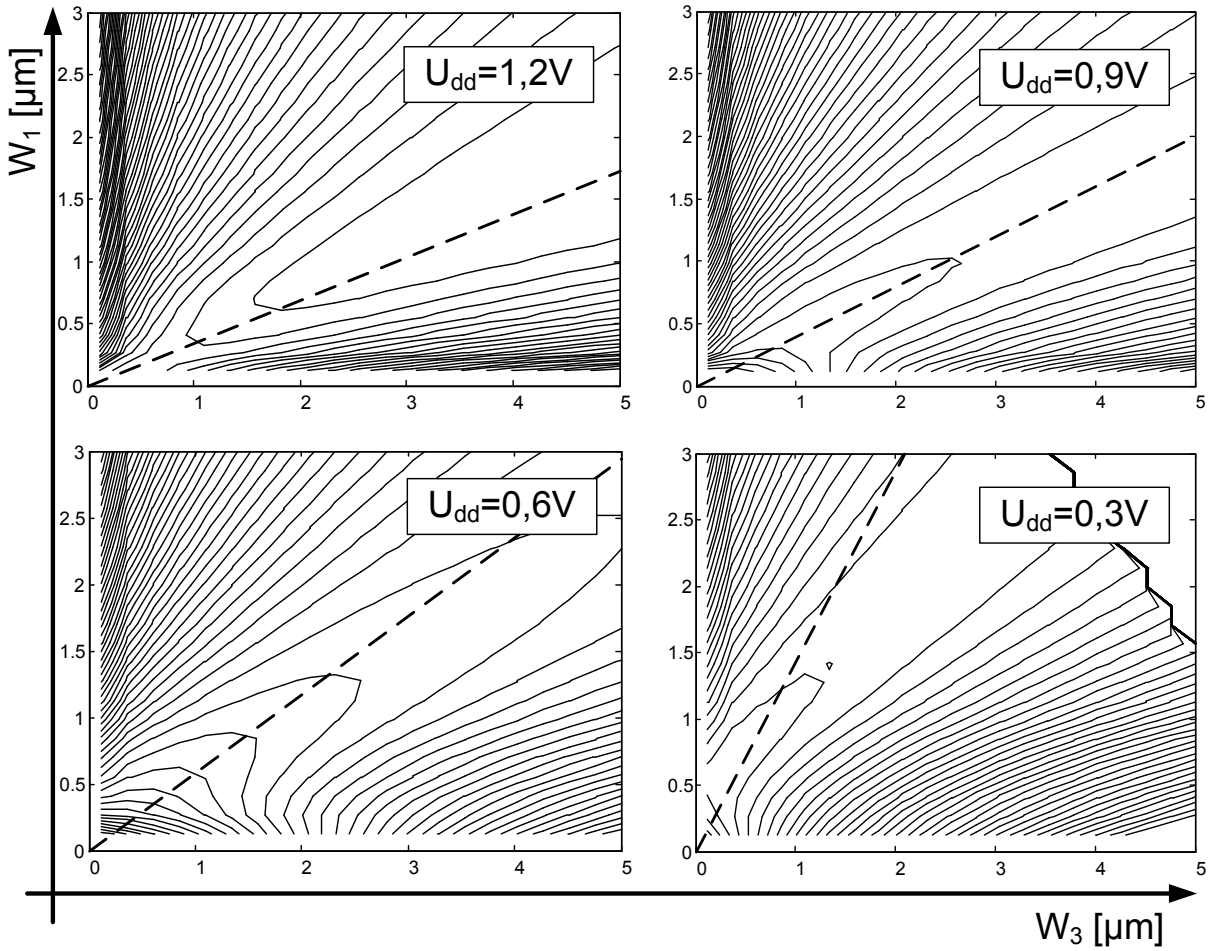


Abbildung 4.6: Höhenlinien $\mathrm{SNM_{hold}}$ bei minimalen Transistorlängen zu verschiedenen Versorgungsspannungen U_{dd}.

unabhängig von den Transistoren $\mathrm{M_5}$ und $\mathrm{M_6}$. Zu erkennen ist die Abhängigkeit von den Weiten der Transistoren der internen Inverter. Insbesondere deutet sich ein Bereich größter Werte an, durch den eine Gerade (gestrichelt dargestellt) verläuft. Die gewählte Gerade wird beschrieben durch

$$W_3 = \alpha \cdot W_1 \tag{4.10}$$

mit von der Versorgungsspannung abhängigen Werten für $\alpha = \alpha\,(U_{\mathrm{dd}})$ nach Tabelle 4.1. (Die gestrichelten Linie in Abbildung 4.6 durch den höchsten Wert bei $W_{\mathrm{p}} = 1\,\mu\mathrm{m}$ ist für $U_{\mathrm{dd}} = 0,3\,\mathrm{V}$ nur noch eine grobe Approximation des Bereiches mit den größten Hold-Störabständen.)

U_{dd}	$1,2\,\mathrm{V}$	$0,9\,\mathrm{V}$	$0,6\,\mathrm{V}$	$0,3\,\mathrm{V}$
$\alpha\,(U_{\mathrm{dd}})$	2,9	2,5	1,7	0,7

Tabelle 4.1: Aus Abbildung 4.6 ermittelte α zu (4.10).

Für $U_{\mathrm{dd}} = 1,2\,\mathrm{V}$ entspricht dies in etwa einem Verhältnis wie in (4.9).

Nachdem unter der Einschränkung (4.10) die SRAM-Zelle im Hold Zustand robust ist, sollen die Störabstände zunächst zurückgestellt werden. Nach Robustheit liegt die Zielsetzung des Entwicklers auf Geschwindigkeit und Energieeffizienz. Um die Zugriffszeiten auf den Speicherblock gering zu halten, muss sowohl schnell geschrieben als auch gelesen werden können. Während beim sehr seltenen Schreibzugriff auch selten dynamische Energie verbraucht wird, darf P_{leak}, wie oben erwähnt, nicht vernachlässigt werden. Somit ergibt sich folgendes MOP

$$\min_{W_3=2{,}9\cdot W_1, W_5} \{t_{\mathrm{write}}, t_{\mathrm{read}}, P_{\mathrm{leak}}\}\,, \tag{4.11}$$

dessen Pareto-Menge duch die oben beschriebenen evolutionären Algorithmen approxiniert wurde und in Abbildung 4.7 dargestellt ist.

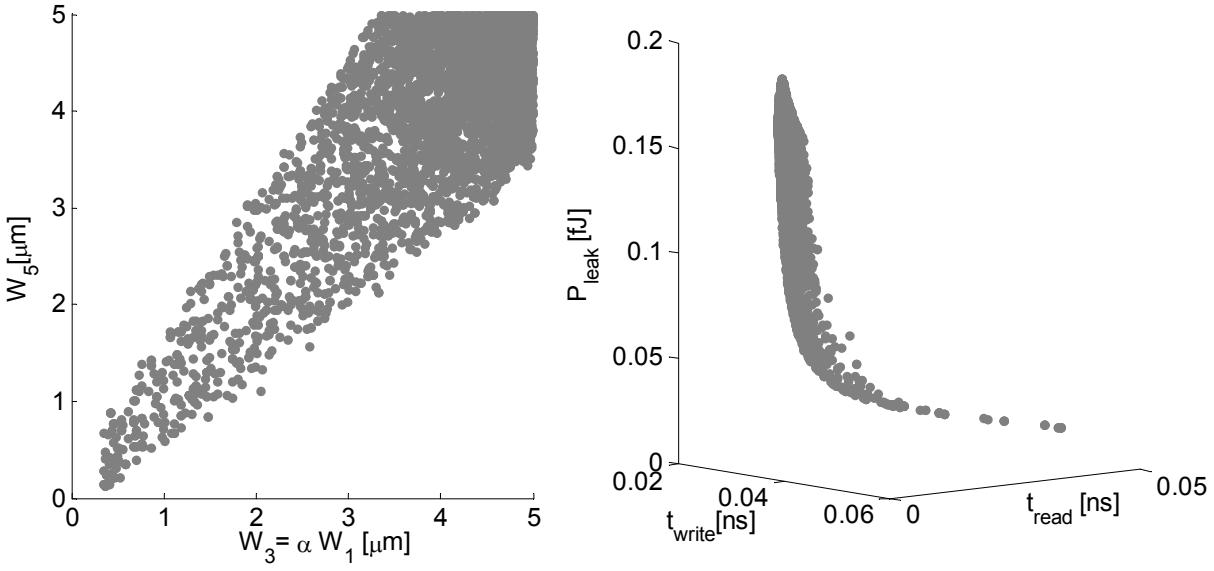


Abbildung 4.7: Pareto-Menge und Front zu (4.11)

Gut zu erkennen ist, dass sich die Gerade $W_5 = 3 \cdot 0,3 \cdot W_3 = 0,9 \cdot W_3$ innerhalb der Pareto-Menge befindet. Die zu (4.11) bestimmte Lösungsmenge befindet sich also in der Nähe der durch (4.9) zuvor berechneten Geraden. Somit sind die Lösungen des mehrkriteriellen Optimierungsansatzes für die 6-Transistor-SRAM-Zelle, wie zuvor für die Standardzellen, mit theoretischen Vorbetrachtungen konsistent. Darüber hinaus sind auch hier die Vorteile der algorithmischen Optimierung eindeutig: Es wurde systematisch der gesamte Suchraum unter Verwendung sehr genauer (BSIM4) Transistormodelle durchsucht. Die gefundene Lösung umfasst eine Vielzahl ressourceneffizienter Schaltungen.

SRAM-Entwurf für den Subschwellbereich

Über die Möglichkeiten und Probleme, die traditionelle 6-Transisitsor-Zelle auch für Versorgungsspannungen unterhalb der Schwellspannungen der Transistoren einzusetzen, gibt es bereits einige Arbeiten. Callhoun und Chandrakasan [10] haben die traditionelle 6-Transistor-SRAM-Zelle bei 300 mV Versorgungsspannung untersucht und zeigen genauer, wie sich Prozessparameterschwankungen und Temperaturschwankungen auf die Schreibzuverlässigkeit auswirken. Das Resultat ist eindeutig: Die volle Funktionalität der Zelle kann für $U_{\mathrm{dd}} = 0,3\,\mathrm{V}$ nicht gewährleistet werden. Sie stellen eine neue Architektur vor, die mit vier zusätzlichen Transistoren die erkannten Schwächen überwinden soll. Als Versorgungsspannung wählen sie 400 mV, die in der 65nm Technologie knapp unterhalb der Schwellspannung liegt.

Ebenfalls eine 10-Transistor-SRAM-Zelle entwarfen Kulkarni, Kim und Roy [44]. Diese benötigt, anders als die eben erwähnte, keine doppelten Wort- oder Bitleitungen. Sie funktioniert außerdem in einer 130nm Technologie bei nur 175 mV.

Der Ansatz (4.10) ist für $U_{\mathrm{dd}} = 0,3\,\mathrm{V}$ nicht mehr praktikabel. Hier würde das MOP

$$\min_{W_{\mathrm{p}}=0{,}7\cdot W_{\mathrm{n}},W_5} \{t_{\mathrm{write}}, t_{\mathrm{read}}, P_{\mathrm{leak}}\}, \tag{4.12}$$

eine Pareto-Menge liefern, die in Abbildung 4.8 durch ein Punktraster angedeutet ist.

Einer der Punkte steht für eine Dimensionierung $W_3 = 0,7 \cdot W_1 = 0,99\,\mu\mathrm{m}$ und $W_5 = 3,05\,\mu\mathrm{m}$. Für diesen ist in Abbildung 4.9 beispielhaft gezeigt, wie schlecht es um seine Robustheit bei einer Versorgungsspannung von $U_{\mathrm{dd}} = 300\,\mathrm{V}$ steht: Während sich die statischen Störabstände $\mathrm{SNM}_{\mathrm{hold}}$ und $\mathrm{SNM}_{\mathrm{write}}$ noch einen einigermaßen sicheren Abstand zur Null haben, lässt sich an $\mathrm{SNM}_{\mathrm{read}}$ erkennen, dass die Schaltung beim Lesen sehr störanfällig ist und für die Ecke FS (schneller NMOSFET, langsamer PMOSFET) keine Erhaltung des gespeicherten Bits garantiert werden kann.

Eine 6-Transistor-Zelle kann bei $U_{\mathrm{dd}} = 300\,\mathrm{V}$ nicht benutzt werden, ohne beim Lesen einen Informationsverlust zu riskieren. Allein durch eine andere Wahl von Transistorweiten ist dem Problem nicht nachzukommen. Im folgenden Abschnitt wird daher eine SRAM-Zelle mit neun Transistoren vorgestellt, die die Störanfälligkeit beim Lesen durch zusätzliche Transistoren überwindet.

Ein 9-Transistor-SRAM-Entwurf zum Voltage-scaling

Es sind vor allem statische Leckströme, die den größten Anteil an der Verlustleistung des SRAM-Speichers verursachen, da die Aktivität einer einzelnen Zelle sehr gering ist. Für zukünftige SoCs wird erwartet, dass der Speicher bis zu 90 % der gesamten Chipfläche belegt [55]. Die Entwicklung von Speicher, deren Funktionalität bei abgesenkter

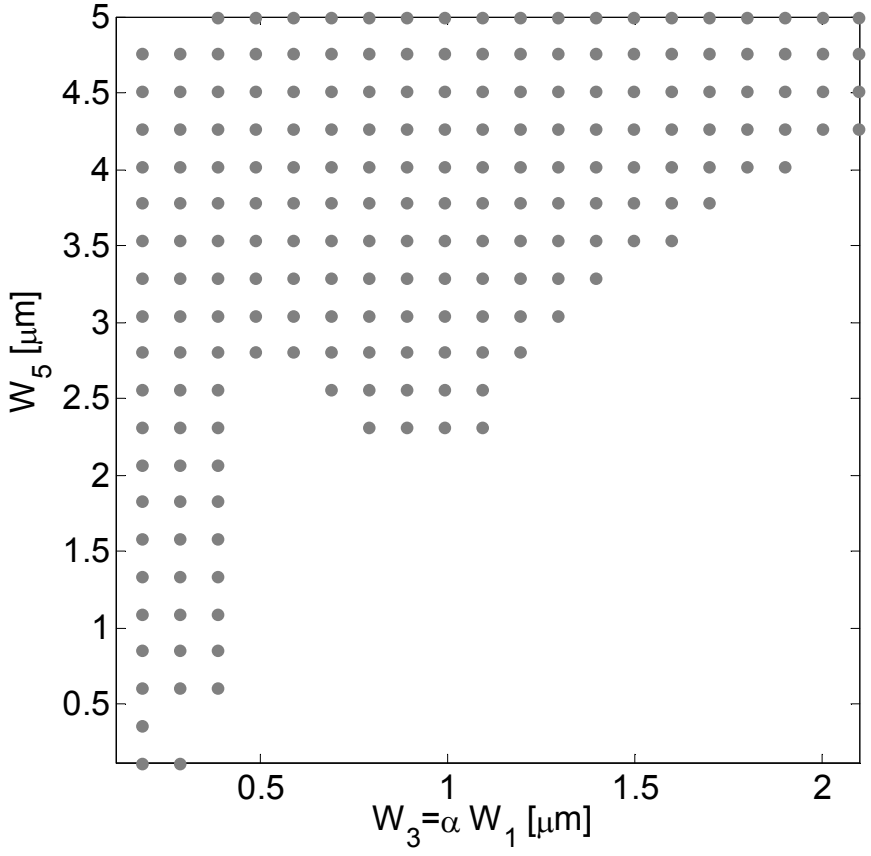


Abbildung 4.8: Approximation der Pareto-Menge zu (4.12) mittels Punkteraster

Versorgungsspannung garantiert werden kann, ist somit ein sehr effektiver Hebel, um die Gesamtverlustleistung eines IC beachtlich zu reduzieren. Die angestrebte Versorgungsspannung soll, wie bei den Standardzellen, 300 mV betragen. Die Speicherzelle soll aber nicht nur robust sondern auch ressourceneffizient sein und dies sowohl im Subschwellbereich, als auch bei voller Versorgungsspannung. Mit dieser Zielsetzung wird in diesem Abschnitt eine 9-Transistor-Zelle dimensioniert. Die vorgestellten Ergebnisse sind teilweise in [3] veröffentlicht worden.

4.1.3 Die 9-Transistor-SRAM-Zelle

Bei Versorgungsspannungen unterhalb der Schwellspannung der Transistoren lässt sich die klassische 6-Transistor-Zelle nicht mehr sicher beschreiben und auslesen (s. oben und [10]). Daher kann jene Zelle nur während des Haltens eines gespeicherten Bits in den Subschwellbetrieb versetzt werden. Dies bedeutet aber, dass die Versorgungsspannung zum Lesen oder Schreiben zunächst angehoben werden müsste. Weil dies die Zugriffsgeschwindigkeit stark verschlechtert, zusätzliche Energie verbraucht wird und eine höhere Versorgungsspannung vielleicht nicht verfügbar ist, sind eine Reihe neuer SRAM-Entwürfe für den Betrieb bei sehr niedriger Versorgungsspannung entwickelt worden [10, 44, 65, 11]. In all diesen wird während des Schreibvorgangs die Versorgungsspannung innerhalb der

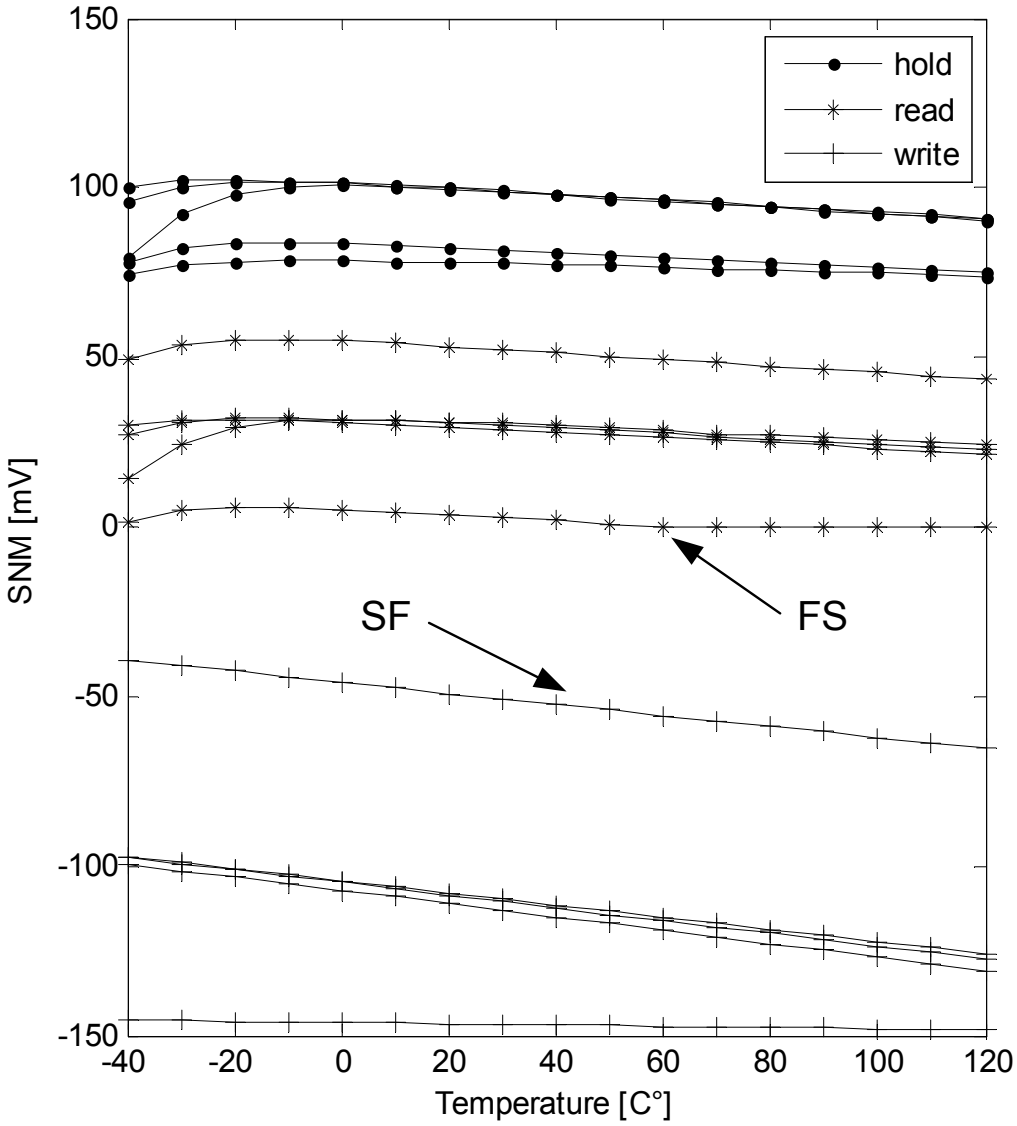


Abbildung 4.9: $U_{\mathrm{dd}} = 300\,\mathrm{V}$, $W_3 = 0,7W_1 = 0,99\,\mu\mathrm{m}$ und $W_5 = 3,05\,\mu\mathrm{m}$, für TT und Ecken SF, FS, SS und FF bei dev=2

Zelle getrennt und zum Auslesen zusätzlicher Transistoren verwandt, um ein Lesen des Zellinhalts ohne Öffnen der Wortleitungstransistoren zu ermöglichen.

Die hier dikutierte 9-Transistor-Zelle (Netzliste s. Abbildung 4.10) wurde in [48] vorgestellt, aber noch nicht für den Betrieb im Subschwellbereich genutzt. Sie hat allerdings einen entkoppelten Lesezweig (Transistoren M_7-M_9), womit sie eine wesentliche Voraussetzung erfüllt. Anders als in [48] werden hier allerdings low-V_t-Transistoren für M_7-M_9 verwendet, um die Lesezeit zu verringern. Transistoren M_1-M_6 sind, wie bei der 6-Transistor-Zelle, high-V_t-Elemente, da hier der größte Anteil der Leckströme fließt, die durch diese Wahl verringert werden. Hier werden für die Wortleitungstransistoren M_5 und M_6 als PMOSFETs gewählt, um im Layout die Anzahl der Streifen mit unterschiedlichen Schwellspannungen zu reduzieren.

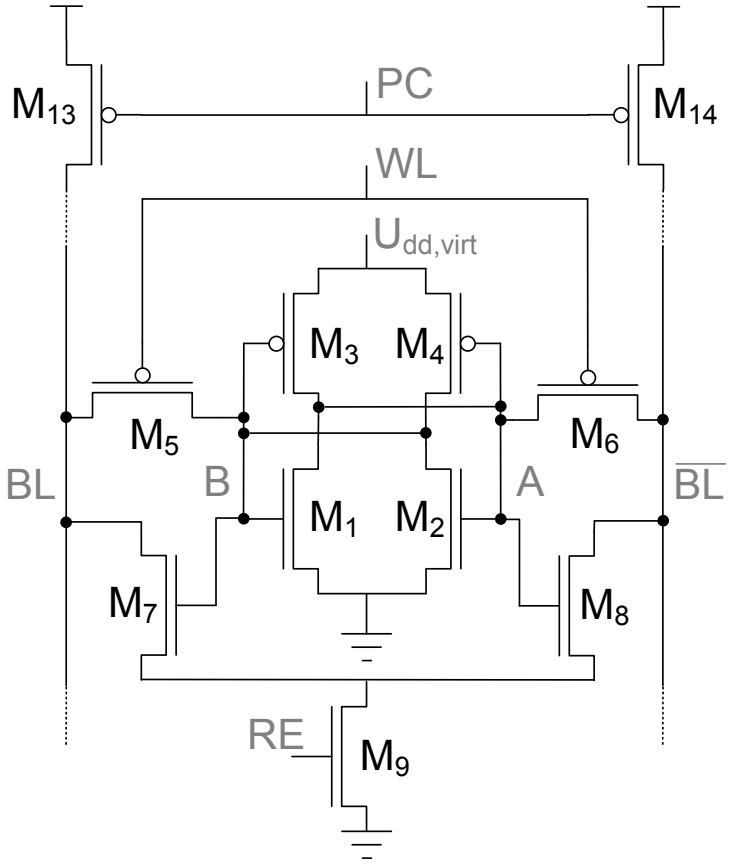


Abbildung 4.10: Netzliste der 9-Transistor-SRAM-Zelle (M_1-M_9) [3]

$U_{dd,virt}$ bezeichnet die virtuelle Versorgungsspannung, die während des Lesens abgesenkt wird. Die Spannung an diesem Knoten beeinflusst den Wert SNM_{write}, der somit Werte unterhalb von $-\frac{U_{dd}}{2}$ erreichen kann. Für die folgenden Optimierungen wird angenommen, dass $U_{dd,virt}$ auf 250 mV abfällt.Die Verzögerungszeit beim Lesen t_{read} wird bestimmt als die Summe der Verzögerung beim Vorladen der Bitleitungen BL und $\overline{BL}$ auf $0.95\,U_{dd}$ und Entladen einer Bitleitung auf $0.5\,U_{dd}$. Die Zeit t_{write} verstreicht beim Schreiben bis die Spannungen an Knoten A und B denselben Wert annehmen.

4.1.4 Suchraumexploration

Die Zelle soll für zwei Anwendungsfälle dimensioniert werden. Als Erstes wird sie ausschließlich für den Betrieb bei 300 mV untersucht. In einem zweiten Schritt wird der Anwendungsfall um die volle Versorgungsspannung erweitert. In beiden Fällen werden die Zielfunktionen SNM_{hold}, SNM_{write}, t_{write}, t_{read} und P_{leak} berücksichtigt. Wie auch bei der 6-Transistor-Zelle werden Transistoren symmetrisch ausgelegt und die Bezeichner von Kanallängen und -weiten können von der Netzliste (Abbildung 4.10) abgeleitet werden.

Durch die Erweiterung der 6-Transistor-Zelle um einen Auslesepfad, können einige der Zielfunktionen unabhängig voneinander optimiert werden (z.B. t_{write} und t_{read}). Dies ermöglicht eine getrennte und schrittweise Exploration des Suchraums, was die Komple-

xität der einzelnen Schritte verringert (die Dimension des Suchraums). Algorithmisch unterstützt wird die Suche durch die im letzten Kapitel vorgestellten Algorithmen. Auch deren Laufzeit wird durch kleinere Suchräume reduziert. Mit dem Ziel, die berechneten Ergebnisse in einfachen Plots darzustellen, werden folgend Optimierungsschritte mit zwei-dimensionalen Suchräumen (und zumeist auch zwei-dimensionalen Bildräumen) formuliert. Diese Einteilungen ergeben sich auf sehr nachvollziehbare Weise. Als eine weitere initiale Bedingung werden alle Längen zunächst auf 170 nm fixiert. Für M_1-M_4 werden so die Leckströme um über 90 % verringert. Durch Ausnutzung der Entwurfsregeln ergibt sich aber nur eine Vergrößerung der Fläche um 15 % im Vergleich zu minimalen Längen. (Dass des Weiteren bei einigen der verwendeten Transistortypen mit längeren Kanälen sogar der Reverse-Short-Channel-Effekt genutzt werden kann, wurde schon im letzten Kapitel erwähnt.)

Ein Subthreshold Entwurf für 300 mV Versorgungsspannung

Zunächst soll die Robustheit der Zelle optimiert werden. $\mathrm{SNM}_{\mathrm{hold}} - 6\sigma_{\mathrm{SNMhold}}$ sollte dazu möglichst groß und vor allem größer als Null sein. Ohne Parameterschwankungen und bei Raumtemperatur liegt $\mathrm{SNM}_{\mathrm{hold}}$ bei Werten über 110 mV und ist damit für den typischen Fall ausreichend robust. Die Abweichung $\sigma_{\mathrm{SNMhold}}$ dagegen ist sehr abhängig von Abweichungen der Transistoren von Nominalwerten ihrer Entwurfsparameter und variiert zwischen 12 mV und 4 mV. $\sigma_{\mathrm{SNMhold}}$ lässt sich aus dem Streuverhalten der Transistoren M_1 und M_3 ableiten. Eine Approximation durch

$$\sigma_{\mathrm{SNMhold}}\left(W_1, W_3\right) = \sqrt{\frac{{\sigma_n}^2}{W_1/W_{0,n}} + \frac{{\sigma_p}^2}{W_3/W_{0,p}}}, \tag{4.13}$$

wobei $\sigma_n \approx 9.9\,\mathrm{mV}$ und $\sigma_p \approx 5.2\,\mathrm{mV}$, liefert einen relativen Fehler von maximal 6 % verglichen mit einigen wenigen Monte-Carlo-Simulationen. Zur Herleitung s. Anhang.

Die Standardabweichung verringert sich für größere Transistoren. Somit werden in einem ersten Schritt die Pareto-optimalen Lösungen zum MOP

$$\min_{W_3, W_1} \; \{\sigma_{\mathrm{SNM}_{\mathrm{hold}}}, A\} \tag{4.14}$$

bestimmt. A steht für die Fläche der Zelle und kann durch geeignete Wahl von Konstanten $c_1, c_2 > 0$ beschrieben werden mit

$$A = A\left(W_1, W_3\right) = c_1 \cdot \left(W_1 + W_3 + c_2\right). \tag{4.15}$$

Die Pareto-Menge von (4.14) kann analytisch bestimmt werden, da der Suchraum zwei-dimensional ist und die Ableitungen vorliegen (s. Kapitel 1.1.3). Es ist

$$\nabla A\left(W_1, W_3\right) = c_1 \cdot (1, 1)$$

und

$$\nabla\sigma_{\text{SNMhold}}\left(W_1, W_3\right) = \frac{1}{2\sqrt{\frac{{\sigma_n}^2}{W_1/W_{0,n}} + \frac{{\sigma_p}^2}{W_3/W_{0,p}}}} \cdot \left(-\frac{\sigma_n^2}{W_1^2/W_{0,n}}, -\frac{\sigma_p^2}{W_3^2/W_{0,p}}\right).$$

Somit muss für alle substationären Punkte gelten:

$$\frac{\sigma_n^2}{W_1^2/W_{0,n}} = \frac{\sigma_p^2}{W_3^2/W_{0,p}}.$$

Da $W_{0,\text{n}} = W_{0,\text{p}}$ gewählt sei und weiterhin $W_1, W_3 > 0$, ergibt sich eine lineare Abhängigkeit

$$W_3 = \alpha \cdot W_1, \tag{4.16}$$

mit $\alpha = \sigma_p/\sigma_n \approx 0.53$.

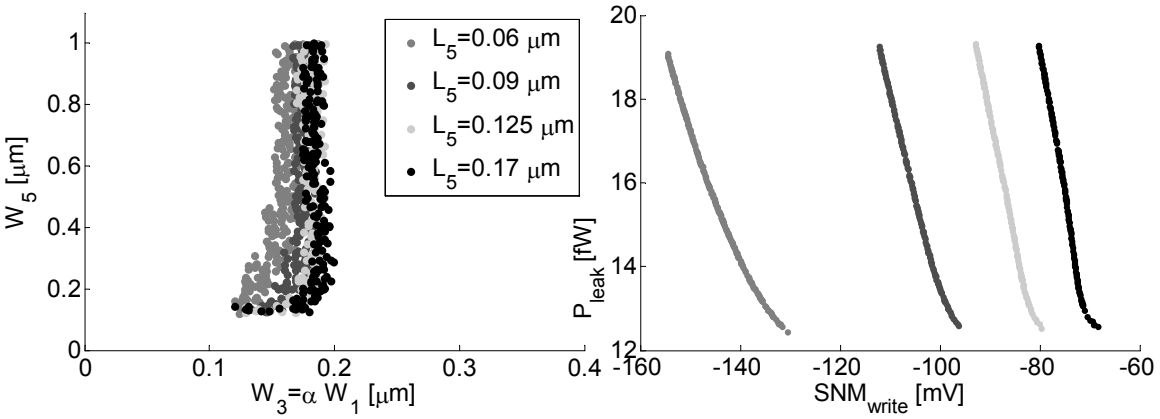


Abbildung 4.11: Pareto-Mengen und Fronten zum MOP (4.17) mit verschiedenen Längen $L_5 = L_6$ [3]

Die Größe von M_5 hat den größten Einfluss auf das Schreibverhalten der Zelle. Sowohl $\text{SNM}_{\text{write}}$ als auch t_{write} verbessern sich mit weiteren Transistoren aber erhöhen gleichzeitig die Verlustleistung. In einem weiteren Schritt werden daher optimale Werte für W_1, W_3 und W_5 durch Lösen des MOP

$$\min_{W_3=\alpha W_1, W_5} \{\text{SNM}_{\text{write}}, P_{\text{leak}}\} \tag{4.17}$$

gesucht. Abbildung 4.11 zeigt die durch den EA approximierten Pareto-Mengen zu (4.17), wobei für die Längen von M_5 verschiedene Werte gewählt wurden. Es fällt auf, dass geringere Längen für M_5 eine klare Verbesserung im Robustheitsmaß $\text{SNM}_{\text{write}}$ bewirken ohne dabei größere Leckströme zu verursachen. Damit ist die Wahl minimaler Transistorlängen für M_5 und M_6 auch im Subschwellbetrieb sinnvoll.

Im Subschwellbereich dominiert der Einfluss von M_{13} und M_{14} gegenüber M_7-M_9 die Verzögerungszeit t_{read}, da das Vorladen der Bitleitungen bei weitem länger dauert als das

Entladen. Daher liefert das MOP

$$\min_{W_7=W_9,W_{13}} \{t_{\text{read}}, P_{\text{leak}}\} \tag{4.18}$$

$W_7 = W_9 = W_{\text{min}}$ und verschiedene Werte für M_{13} bestimmen optimale Kompromisse zwischen kürzerer Lesezeit und weniger Verlustleistung. Abhängig von n_{Zellen}, der Anzahl der Zellen pro Spalte, kann die Weite W_{13} angepasst werden, damit t_{read} mit der durch M_1-M_6 bestimmten Schreibzeit übereinstimmt.

In Tabelle 4.2 ist ein Pareto-Punkt für $n_{Zellen} = 16$ gewählt, dessen Schalteigenschaften weiter unten mit bereits veröffentlichten Zellen verglichen werden.

M_1	M_3	M_5	M_7, M_9	M_{13}
0.305 / 0.17	0.16 / 0.17	0.49 / 0.06	0.15 / 0.17	10.31 / 0.17

Tabelle 4.2: Beispiel der 9-Transistor-SRAM-Zelle optimiert für $n_{Zellen} = 16$ bei 0,3 V. Schreibweise: Weite / Länge

Ein Entwurf zur Versorgungsspannungskalierung

Die Bestimmung optimaler Kompromisse für Versorgungsspannungen von sowohl 300 mV als auch 1.2 V erhöht die Zahl der zu berücksichtigenden Zielfunktionen und Parameter. Zu Beginn soll aber auch hier die Robustheit der Schaltung im Subschwellbereich für den Haltezustand optimiert werden. Daher wird die Abhängigkeit (4.16) vorausgesetzt. Bei hoher Versorgungsspannung sind beim Lesen schnelle Zugriffszeiten erstrebenswert, während bei 300 mV die Robustheit erhöht werden muss. Somit wird folgendes MOP formuliert:

$$\min_{W_3=\alpha W_1,W_5} \{t_{\text{write},1.2\,\text{V}}, \text{SNM}_{\text{write},0.3\,\text{V}}, P^*_{\text{leak}}\}\,. \tag{4.19}$$

Die Verlustleistung P^*_{leak} setzt sich aus den Verlustleistungen beider Betriebszustände zusammen:

$$P^*_{\text{leak}} = (1 - \text{AF}) \cdot P_{\text{leak},0.3\,\text{V}} + \text{AF} \cdot P_{\text{leak},1.2\,\text{V}},$$

wobei ein Aktivitätsfaktor AF eingeführt wird, der abhängig von einer späteren Anwendung ist. Beispielsweise kann AF= $\frac{1}{24}$ gewählt werden, wenn man davon ausgeht, dass der Speicher in einem Mobiltelefon verbaut wird, dessen aktive Zeit bei etwa einer Stunde pro Tag liegt.[1] Das Ergebnis der Optimierung ist in Abbildung 4.12 gezeigt.

[1]Das Monotonieverhalten von P^*_{leak} über dem Suchraum ist nicht sehr sensibel bezüglich AF, da die Verlustleistung in beiden Betriebszuständen mit weiteren Transistoren ansteigt. Somit wird das Ergebnis der Mehrzieloptimierung kaum/nicht durch die Wahl von AF beeinflusst.

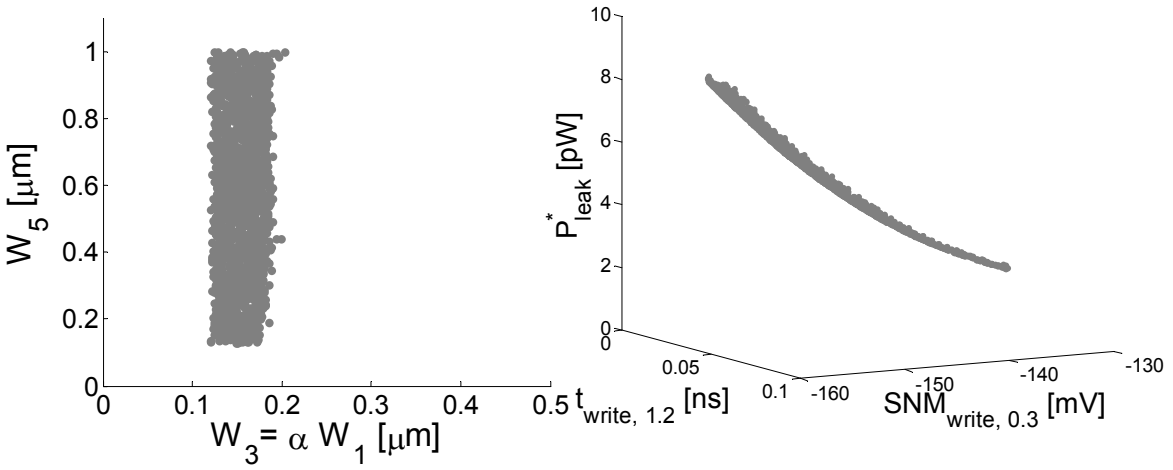


Abbildung 4.12: Pareto-Menge und -Front zu M_1 - M_6 nach (4.19) [3]

Um die Transistoren M_7 - M_9, M_{13} und M_{14} zu dimensionieren, soll die Pareto-Menge des MOP

$$\min_{W_7=W_9,W_{13}} \{t_{\mathrm{read,1,2\,V}}, t_{\mathrm{read,0,3\,V}}, P^*_{\mathrm{leak}}\}$$

bestimmt werden. Die Robustheit des Lesevorgangs wird durch die Architektur gewährleistet, was ermöglicht, anders als in (4.19), die Geschwindigkeit statt eines Robustheitsmaßes bei 300 mV zu berücksichtigen. Die approximierte Lösung ist in Abbildung 4.13 dargestellt. Alle Kompromisse zwischen $t_{\mathrm{read,0,3\,V}}$ und P^*_{leak} liefern $W_7 = W_9 = W_{\mathrm{min}}$ (s.o.). Doch bei voller Versorgungsspannung schaffen weitere Transistoren M_7 - M_9 Kompromisse zwischen $t_{\mathrm{read,1,2\,V}}$ und P^*_{leak}.

Die Pareto-Punkte erreichen Werte in $t_{\mathrm{read,1,2\,V}}$ von 0,349 ns bis 9,36 ns. Ein Schreibzugriff bei 0,3 V dauert mindestens 289 ns und P^*_{leak} reicht von 0,29 pW bis 4,57 pW.[2]

4.1.5 Vergleich zu anderen Implementationen

Die Schalteigenschaften des Beispiels aus Tabelle 4.2 sind in Tabelle 4.3 einigen bereits veröffentlichten SRAM-Zellen gegenübergestellt, die für den Betrieb im Subschwellbereich entworfen wurden. Die Standardabweichungen sind bestimmt auf Basis von Monte-Carlo-Simulationen kombiniert mit Prozessparameterschwankungen und Mismatch. Da $\mathrm{SNM_{hold}} - 6\sigma_{\mathrm{SNMhold}} \approx 32\,\mathrm{mV}$, kann eine hohe Ausbeute bei 300 mV erwartet werden. $\mathrm{SNM_{write}} = -152$ ist niedrig genug, um zuverlässiges Schreiben auch unter Störungen zu gewährleisten. Das bedeutet auch, dass eine Absenkung von $U_{\mathrm{dd,virt}}$ auf 250 mV ausreichend ist.

[2]Die Verlustleistung wird beim Schreiben durch die Leckströme innerhalb der 6-Transistor-Zelle gemessen, während beim Lesen nur Ströme von den Bitleitungen durch $M_7 - M_9$ berücksichtigt werden.

[5]Die Verlustleistung ergibt sich ausschließlich aus statischen Leckströmen. Außerdem sind keine Einflüsse umliegender Zellen berücksichtigt.

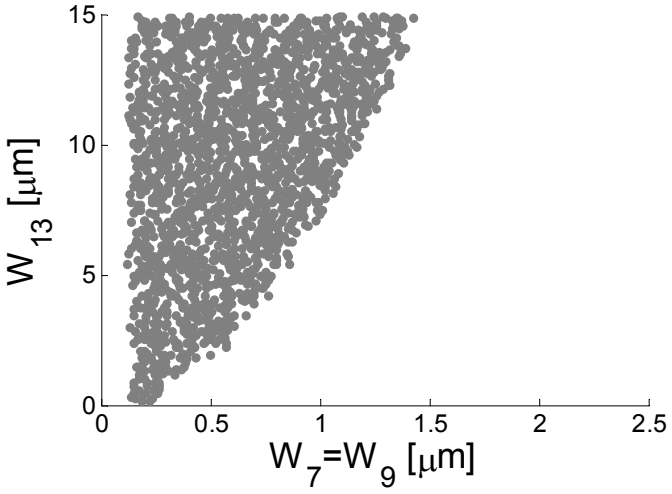


Abbildung 4.13: Pareto-Menge M_7 - M_9 und M_{13} - M_{14} für P^*_{leak} und Verzögerungszeit beim Lesen für 16 Zellen pro Spalte. [3]

	Referenzschaltungen				
Parameter	eigene	[11]	[65]	[78]	[40]
SNM_{hold} [mV]	107	≈ 90	-		$\approx$130
σ_{SNMhold}	12.5	≈ 15	-		-
$\text{SNM}_{\text{write}}$ [mV]	-152	$\approx$-135	$\approx$-100		-78
σ_{SNMwrite}	22.8	≈ 10	≈ 70		1.4
t_{read} [μs]	0.268	-	≈ 5		-
t_{write} [μs]	0.268	-	-		-
P_{leak} [pW/Zelle]	0.26[5]	≈ 25	≈ 12	≈ 37	≈ 25
T/Zelle	9	10	8	6	10
n_{Zellen}	16	256	64	256	1024
Max f [kHz]	>100	580	50	$\approx$100	190

Tabelle 4.3: Vergleich von Subschwell SRAM bei 0.3 V

Basierend auf Simulationsergebnissen verspricht die hier diskutierte 9-Transistor-SRAM-Zelle niedrige Leckströme. Die Abschätzung der maximalen Betriebsfrequenz und absoluten Verlustleistung ist ohne Berücksichtigung umliegender Zellen nur ungenau. Zu erwarten sind einige hundert kHz. Für die Optimierungsansätze sind 16 Zellen pro Spalte ($n_{Zellen} = 16$) gewählt worden. Simulationsbasierte Untersuchungen haben aber gezeigt, dass sich n_{Zellen} auf mindestens 32 erhöhen lässt.[3] In Planung sind Implementationen, die für $n_{Zellen} > 16$ optimiert und gefertigt werden.

[3]Für $n_{Zellen} = 32$ bewegt sich der Quotient $I_{\text{on}}/I_{\text{off}}$ bei 10 in einer 5σ Ecke.

Kapitel 5

Optimierung analoger Schaltungen

Analoge Blöcke wie Verstärker, Oszillatoren, Filter, Gleichrichter und Mischer sind auf fast jedem IC zu finden. Daher soll im Kontext dieser Arbeit eine Einleitung in die Optimierung analoger Schaltungen gegeben werden. Mit dem Differenzverstärker mit Stromspiegel wird ein einfaches und typisches Beispiel für eine analoge Schaltung vorgestellt. Eine Diskussion dessen Schalteigenschaften verdeutlicht, wie essentiell wichtig eine Robustheitsanalyse bei der Suchraumexploration analoger Schaltungen ist. Entwürfe von Algorithmen, die diesen neuen Ansprüchen gerecht werden sollen, werden vorgestellt. Außerdem wird auf eine alternative Optimierungsstrategie, dem *Design-Centering* zum Entwurf robuster Analogschaltungen, eingegangen.

5.1 Schalteigenschaften des Differenzverstärkers

Ein Differenzverstärker verstärkt die Differenz zweier Eingangssignale am Ausgang um den Faktor A:

$$U_{\text{out}} = A\left(U_{\text{in1}} - U_{\text{in2}}\right).$$

Abbildung 5.1 zeigt die Netzliste des Verstärkers und Abbildung 5.2 die Übergangskennlinie. Die Verstärkung ist die Ableitung im Arbeitspunkt. Hier wird $U_{\text{in1}} = U_{\text{in2}} = \frac{U_{\text{dd}}}{2}$ gewählt. Betrachtet werden differentielle Eingangssignale mit kleinen Amplituden um diesen Arbeitspunkt.

Nach [58] lässt sich die Verstärkung analytisch approximieren durch

$$A = g_{\text{m1,2}}\left(r_{\text{ds1,2}} || r_{\text{ds3,4}}\right) = g_{\text{m1,2}} \frac{1}{\frac{1}{r_{\text{ds1,2}}} + \frac{1}{r_{\text{ds3,4}}}}. \tag{5.1}$$

Die Steilheit ist definiert für einen Transistor in Sättigung im Arbeitspunkt $U_{\text{GS}} = U_{\text{GS0}}$ als

$$g_m = \frac{\partial I_{\text{DS}}}{\partial U_{\text{GS}}}\bigg|_{U_{\text{GS}}=U_{\text{GS0}}} = B\frac{W}{L}\left(U_{\text{GS0}} - U_{\text{TH}}\right), \tag{5.2}$$

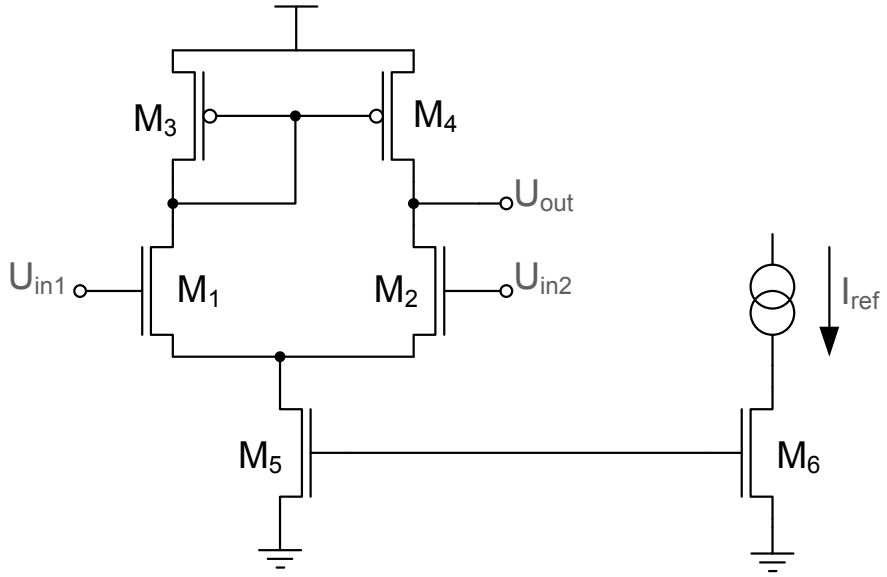


Abbildung 5.1: Netzliste des Differenzverstärkers

und für den Drain-Source-Widerstand r_{ds} gilt

$$\frac{1}{r_{\mathrm{ds}}} = \frac{\partial I_{\mathrm{DS}}}{\partial U_{\mathrm{DS}}}\bigg|_{U_{\mathrm{DS}} > U_{\mathrm{GS0}} - U_{\mathrm{TH}}} = \frac{B}{2}\frac{W}{L}\left(U_{\mathrm{GS0}} - U_{\mathrm{TH}}\right)^2 . \tag{5.3}$$

Somit lässt sich die Verstärkung zunächst unabhängig von Transistor M_5 abschätzen. Außerdem ist nach (5.3) $\frac{1}{r_{O3,4}}$ am kleinsten und somit A am größten, falls $W_3 = W_{\mathrm{min}}$. Schalteigenschaften analoger Schaltungen sind in sehr viel komplexerer Weise von Entwurfsparametern abhängig als digitale. Daher wird im Folgenden nicht versucht, alle Eigenschaften in ihren Abhängigkeiten zu den Transistorgrößen analytisch aufzuschlüsseln. Für die Bestimmung der Verstärkung unterstützen Simulationsergebnisse die Abschätzung (5.1) über dem Suchraum (s. Abbildungen 5.4 und 5.5).

Für diesen Differenzverstärker gibt es mehr steuerbare Parameter als die Transistormaße. Zum Beispiel beeinflusst ebenfalls der Referenzstrom I_{ref}, der über M_5 und M_6 gespiegelt und verstärkt wird, die Eigenschaften der Schaltung. Allerdings lässt sich der Strom auch durch das Weiten-zu-Längenverhältnis von M_5 steuern. Auch kann der Arbeitspunkt von $\frac{U_{\mathrm{dd}}}{2}$ verschieden gewählt werden.

Für Verstärker werden Stabilitätsbedingungen definiert, die quantifizieren, wie sicher sie vor einem Aufschwingen geschützt sind. Unter der Annahme, dass das Eingangssignal eines Verstärkers in einer negativen Rückkopplungsschleife durch sein Ausgangssignal verstärkt wird, sollte der Abstand zwischen -180 ° und der Phase der Schleifenverstärkung zur Frequenz, an der der Betrag der Schleifenverstärkung kleiner wird als eins, möglichst groß sein. Dieser Abstand ist die Phasenreserve (vgl. dazu [58]). Abbildung 5.3a zeigt das Bode-Diagramm der Schleifenverstärkung des Differenzverstärkers, bei der der Ausgang direkt mit dem Eingang verbunden ist.

Die Grenzfrequenz (oder 3dB-Grenzfrequenz) f_{g} ist definiert als die Frequenz, bei der die Verstärkung bereits um 3 dB abgefallen ist (s. Abbildung 5.3b). (-3 dB entspricht etwa $\frac{1}{\sqrt{2}}$.

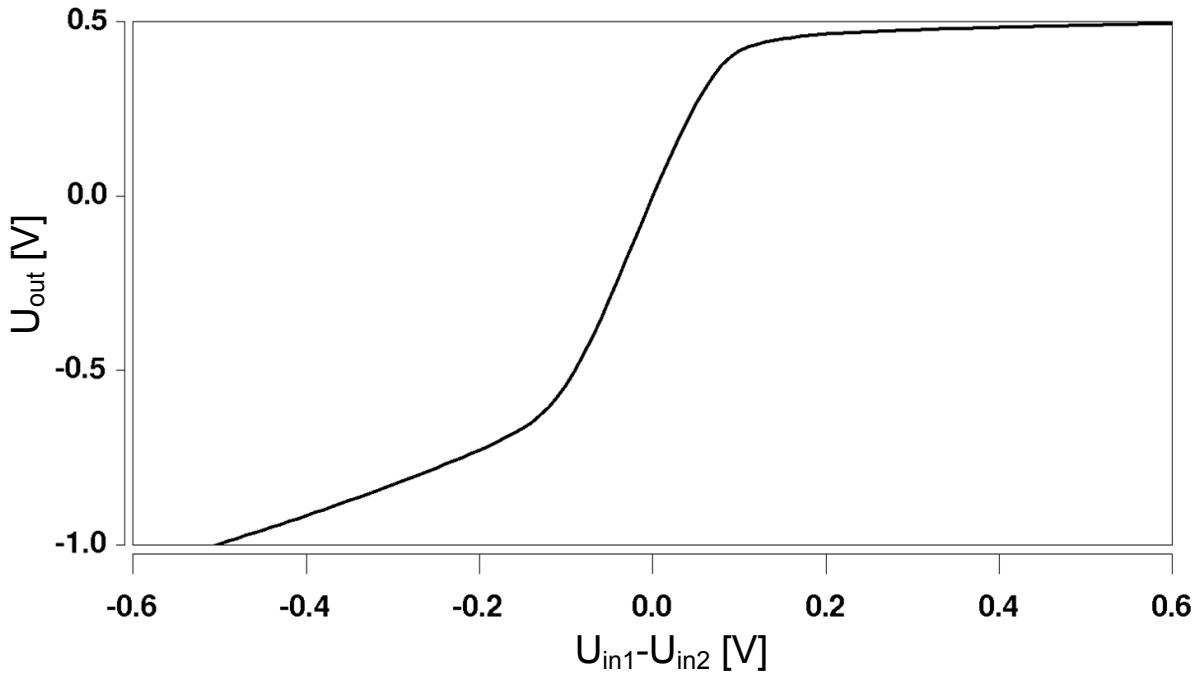


Abbildung 5.2: DC Übertragungsfunktion

Ein Abfall der verstärkten Spannungsamplitude um 3 dB ergäbe damit eine Halbierung der Maximalleistung.)

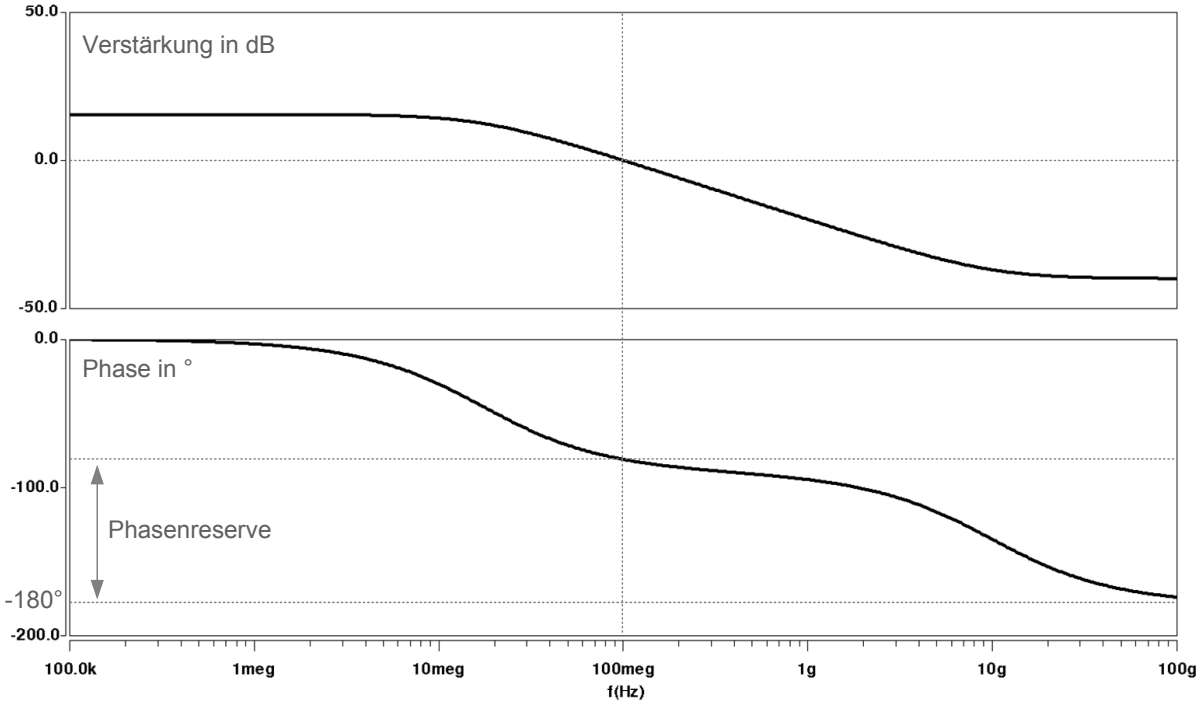
Ein Eingangssignal wird nur im Operationspunkt mit dem angegebenen Faktor A verstärkt. Je weiter sich das Signal entfernt, desto geringer ist die Verstärkung. Als weitere Zielgröße ist daher die Linearität, definiert als die maximale mögliche Abweichung des Eingangssignals, bei dem sich A nicht mehr als 1 % ändert. Bestimmt wird der Wert für niedrige Frequenzen in einer DC-Analyse.

Auch für diese Schaltung ist die Verlustleistung zu berücksichtigen. Mit P_{leak} sind zunächst nur die statischen Leckströme im Arbeitspunkt berücksichtigt. In Konkurrenz zu dieser steht wiederum die Verzögerungszeit t, die die Verzögerung des Ausgangssignals auf ein eingehendes Signal misst. Simulationsergebnisse all dieser Funktionen über zweidimensionale Suchräumen sind in Abbildungen 5.4 und 5.5 abgebildet.

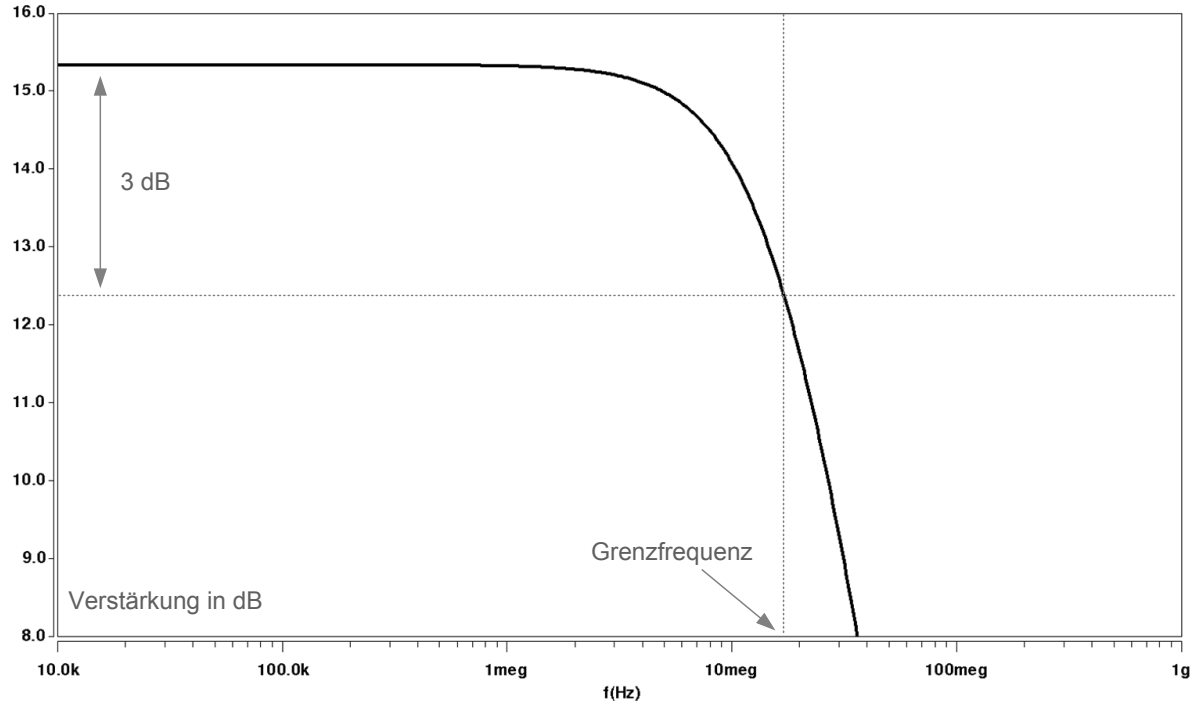
5.2 Mehrzieloptimierung analoger Schaltungen

Die im letzten Abschnitt vorgestellten Schalteigenschaften sind lediglich eine Auswahl und werden abhängig von der betrachteten Schaltung erweitert. Für die Mehrzieloptimierung bedeutet dies, dass sich die Anzahl der Zielfunktionen im Vergleich zu digitalen Logikgattern stark erhöht. Daraus resultieren zumeist MOPs, deren Pareto-Menge nahezu den gesamten Suchraum abdecken.

Bei der Optimierung analoger Schaltungen muss allerdings vor allem die Störanfälligkeit berücksichtigt werden. Entwurfspunkte müssen so gewählt sein, dass die Schalteigenschaften



(a) Schleifenverstärkung des Differenzverstärkers mit Phasenreserve



(b) Verstärkung des Differenzverstärkers mit 3 dB Grenzfrequenz

Abbildung 5.3: Bode-Diagramme

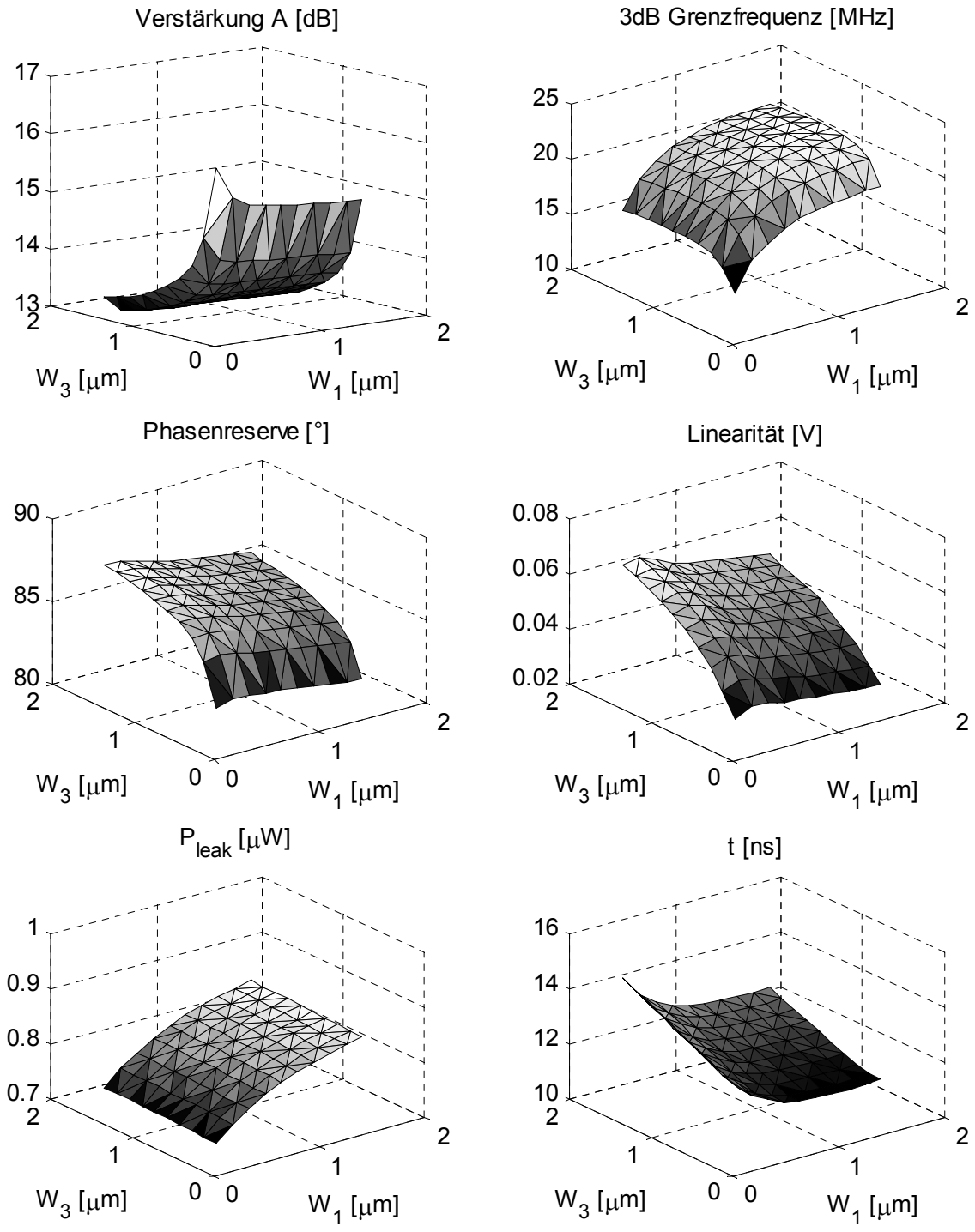


Abbildung 5.4: Zielfunktionen des Differenzverstärkers aus Abbildung 5.1 ($W_5 = W_6 = 0.2\,\mu\text{m}$)

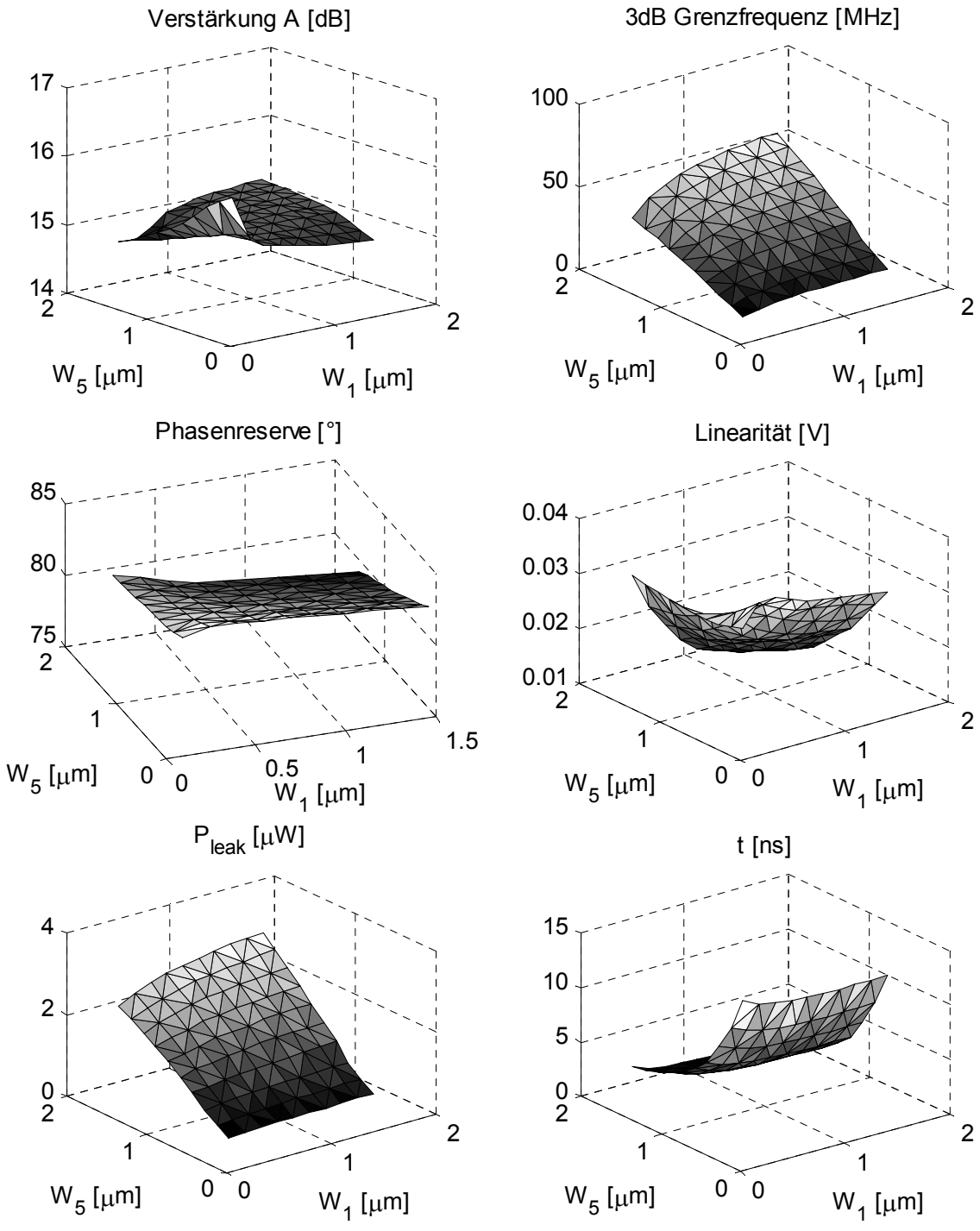


Abbildung 5.5: Zielfunktionen des Differenzverstärkers aus Abbildung 5.1 ($W_3 = 0.12\,\mu\text{m}$)

gegenüber Parameterschwankungen robust sind. In Abbildung 5.6 wird die Pareto-Menge und -Front des MOP

$$\min_{W_1,W_3,W_5} \{-f_\mathrm{g}, P_\mathrm{leak}\} \tag{5.4}$$

dargestellt. Abbildung 5.6b zeigt außerdem wie Temperatur- und Spannungsschwankungen die für den nominalen Fall ermittelten Pareto-Punkte stören. Die relative Verschiebung aller im vorherigen Abschnitt eingeführten Eigenschaften sind in Tabelle 5.1 aufgeführt. Prozessparameterschwankungen werden die Störungen noch vergrößern.

	A	f_g	Phasenr.	Lin.	P_leak	t
U_dd=1.1 V, θ=25 °C	1 %	13 %	0 %	18 %	22 %	-5 %
U_dd=1.3 V, θ=25 °C	-1 %	-12 %	0 %	-17 %	-24 %	4 %
U_dd=1.2 V, θ=-40 °C	-7 %	21 %	3 %	44 %	31 %	-19 %
U_dd=1.2 V, θ=125 °C	10 %	-26 %	-4 %	-71 %	-47 %	13 %

Tabelle 5.1: Relative Verschiebung der Schalteigenschaften der Pareto-Menge von (5.4) im Nominalfall U_dd=1.2 V, θ=25 °C

5.3 Algorithmen zur Optimierung analoger Schaltungen

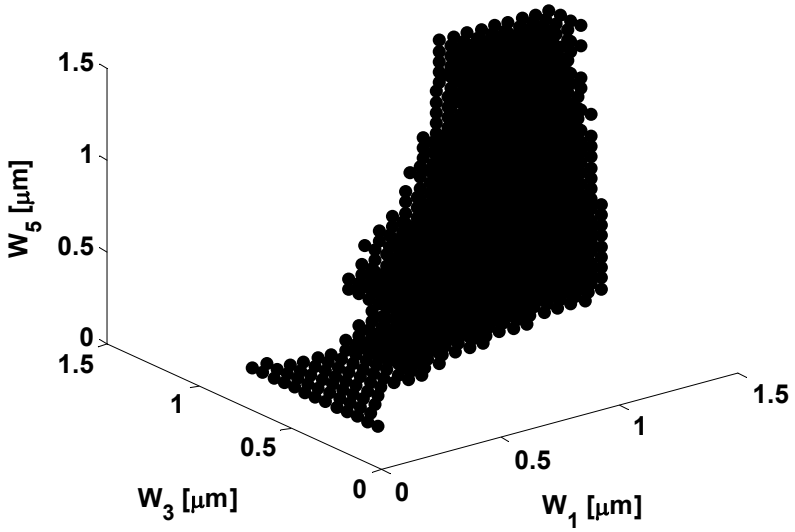
In [59] wird ein Überblick über moderne kommerzielle CAD-Werkzeuge zur Entwicklung integrierter analoger Komponenten gegeben. Die Suchraumexploration komplexerer Elemente folgt dabei ihrer inhärenten hierarchischen Struktur. Für eine reine Optimierung auf Transistorebene werden in [27] einige Werkzeuge vorgestellt, die Schaltungen mit zehn bis maximal hundert Transistoren erfolgreich bewältigen. Unter diesen Techniken basieren, aufgrund der bis dato beschränkten Rechenleistung, erst Algorithmen ab den 90er Jahren auf vollständigen numerischen Simulationen von Netzlisten.

Für die Mehrzieloptimierung analoger Schaltungen wird in [54] ein *Sequential-Quadratic-Programming* Ansatz vorgestellt, bei dem die Pareto-Front an ν gleichmäßig verteilten Punkten approximiert wird. Auf diese Weise wird das MOP auf ν ein-dimensionale Suchen reduziert, was die Laufzeiten steuerbar macht und dem Entwickler die Freiheit lässt, zwischen einer groben Approximation und niedrigen Laufzeiten abzuwägen.

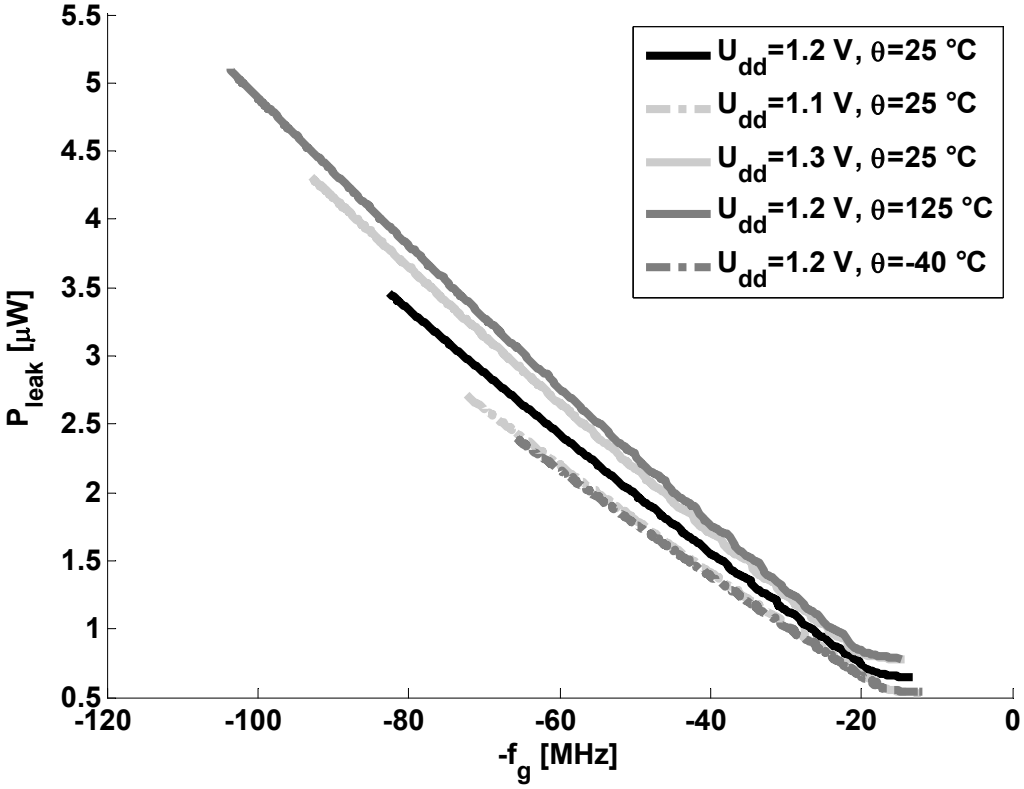
Wie oben gesehen, sollte die klassische Mehrzieloptimierungsaufgabe (1.4) für die Suche nach störungsunanfälligen Entwurfspunkten erweitert werden. Es gilt, Lösungen für das störparameterabhängige MOP

$$\min_{x \in S} f(x, \lambda) \tag{5.5}$$

zu finden, wobei λ mehrere beschränkte Parameter $\lambda_i \in \left[\lambda_i^l, \lambda_i^u\right] = \Lambda$ zusammenfasst.



(a) Pareto-Menge (Nominalfall)



(b) Verschiebung der Pareto-Front bei Schwankungen der Versorgungsspannung U_{dd} und der Temperatur θ

Abbildung 5.6: Pareto-Menge und -Front (schwarz) zu (5.4) für den Nominalfall U_{dd}=1.2 V, θ=25 °C

Solche wenig störungsanfällige Entwurfspunkte können als *robuste* Pareto-Punkte bezeichnet werden. Mit Variation von $\lambda \in \Lambda$ verschiebt sich sowohl die zugehörige Pareto-Menge P_λ als auch deren Front $f(P_\lambda)$. Um die Störanfälligkeit zu bestimmen, wird in [19] der Pfad der Pareto-Punkte verfolgt. Ein Maß für die Robustheit

$$\sigma : S \to \mathbb{R}^+ \tag{5.6}$$

kann auf verschiedene Weisen definiert werden. Zum einen kann die Länge des Pfads bestimmt werden, den ein Pareto-Punkt beschreitet, wenn sich das MOP (5.5) mit λ ändert. Alternativ könnte die maximale Abweichung dieses Punkts berechnet werden. Für den Anwendungsfall integrierter Schaltungen entfernt sich ein Entwurfspunkt mit Variation der Störparameter gewöhnlich am weitesten, wenn der Störparameter maximal ($\lambda_i = \lambda_i^l$ oder $\lambda_i = \lambda_i^u$) ausschlägt. Diese Kenntnis würde die Laufzeit des zweiten Ansatzes stark verringern. Desweiteren kann, je nach Anwendungsfall, der Pfad im Such- oder Bildraum vermessen werden.

Für die Schaltungsoptimierung, und speziell für den Entwurf analoger Bausteine, ist von zentralem Interesse, wie sensibel die Eigenschaften eines Pareto-Punkts gegenüber Parameterschwankungen sind. Ein Algorithmus, der motiviert durch die Schaltungsoptimierung am Paderborner Lehrstuhl für Angewandte Mathematik entstand, ist in [67] dokumentiert. Dieser operiert mit einer *absoluten Sensitivität*, die definiert ist durch

$$(x, \lambda') \mapsto \max_{\lambda \in \Lambda} d\left(f(x, \lambda'), f(x, \lambda)\right). \tag{5.7}$$

Als Metrik wurde

$$d(x, y) = \|s(x) - s(y)\|_2$$

gewählt, wobei mit der Funktion $s : \mathbb{R}^k \to \mathbb{R}^k$ die Zielfunktionen normiert werden sollen. Erste Test haben gezeigt, dass der Algorithmus die Pareto-Punkte erfolgreich durch das Sensitivitätsmaß (5.7) bewertet. Optimierungsergebnisse analoger Schaltungen liegen zu diesem Zeitpunkt noch nicht vor.

Mit der Sensitivitätsanalyse wächst die Anzahl der nötigen Funktionsauswertungen (Schaltungssimulationen) drastisch, was die Laufzeiten stark beeinträchtigt. Um die Auswirkungen von Schwankungen von Prozessparametern zu untersuchen, werden Alternativen zu Monte-Carlo-Simulationen entwickelt, von denen für jeden Entwurfspunkt einige 100 bis 1000 notwendig wären. An dieser Stelle sei nur auf [30] verwiesen, wo die Umgebung hoher Wahrscheinlichkeiten um einen Entwurfspunkt durch affine Abbildungen modelliert und so leicht in den Bildraum abgebildet werden kann. Für eine mögliche Erweiterung der im Rahmen dieser Arbeit untersuchten Algorithmen in Richtung Mehrzieloptimierung störanfälliger analoger Schaltungen sollten derartige Methoden untersucht werden. Die oben erwähnten ersten Tests haben gezeigt, dass vereinfachte Simulationsmodelle für höher-dimensionale Suchräume unumgänglich sind.

5.4 Optimierung der Ausbeute

Statt nach ausgewählten Schalteigenschaften zu optimieren, können für diese auch Grenzen vorgegeben werden, in denen deren Werte liegen sollen. Dann kann versucht werden, Entwurfspunkte zu finden, die nach der Fertigung mit hoher Wahrscheinlichkeit innerhalb der vorgegebenen Limitierung liegen. Die Optimierung der Ausbeute ist ein verbreiteter Ansatz. Daher wird in diesem Abschnitt kurz die Idee dieser Suchraumexploration vorgestellt. Die Suche nach einer möglichst hohen Ausbeute bedeutet, dass ein Entwurfspunkt gesucht wird, dessen Schalteigenschaften sowohl für den Nominalfall, als auch unter Parameterstöreinflüssen mit möglichst hoher Wahrscheinlichkeit innerhalb vorgegebener Grenzen liegen. Die Suche nach solchen Entwurfspunkten nennt man *Design-Centering*. Graeb liefert mit [31] einen Einblick in die verschiedenen Problemstellungen des Design-Centering analoger Schaltungen.

Im Gegensatz zu den bisherigen Optimierungen gibt es beim Design-Centering nur eine Zielfunktion, die Ausbeute y, die in diesem Fall maximiert werden soll:

$$\max_{x \in S} y(x) \tag{5.8}$$

Dabei müssen Randbedingungen

$$g\left(f_1\left(x\right), \cdots, f_{\mathrm{n}}\left(x\right)\right) \leq 0 \tag{5.9}$$

($g : \mathbb{R}^{(n+m)} \rightarrow \mathbb{R}^l$ eine mehrdimensionale Funktion) erfüllt sein. Die Funktionen f_i können die im vorherigen Abschnitt beschriebenen Schalteigenschaften sein. Möglich, und zumeist sinnvoll, sind weitere Funktionen, die beispielsweise die Einhaltung von Entwurfsregeln oder vorgegebene Betriebszustände von Transistoren garantieren.

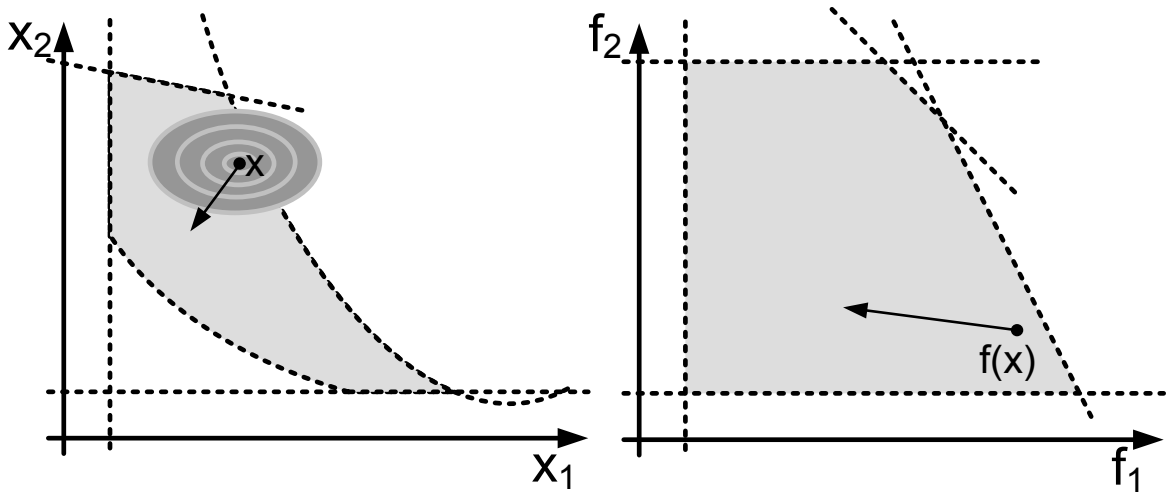


Abbildung 5.7: Suche nach der höchsten Ausbeute im Suchraum (links) unter einschränkenden Bedingungen im Bildraum (rechts) (In Anlehnung an [31])

Die Dichtefunktion der Verteilung der Streuparameter lässt sich näherungsweise als Normalverteilung $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$ mit Dichtefunktion $\phi\left(\mu, \sigma^2, t\right)$ modellieren. Die Höhenlinien

dieser *Gaußglocke* bilden konzentrische Ellipsen um den erwarteten Nominalwert μ. In Abbildung 5.7 ist ein Beispiel der Suche nach maximaler Ausbeute schematisch dargestellt. Im Suchraum (links) entspricht die hellgraue Fläche, genannt Lösungsraum L, dem Urbild der hellgrauen Fläche im Bildraum (rechts), die durch Grenzbedingungen (5.9) beschränkt wird. Die Ellipsen um den Entwurfspunkt x decken den Bereich $B(x,p)$ ab, in dem mit einer gegebener Wahrscheinlichkeit p alle gestörten Werte von $\mu = x$ liegen.[1]

Die Ausbeute ist bestimmt durch das Integral der Dichtefunktion über dem Lösungsraum

$$y(x) = \int_L \phi\left(x, \sigma^2, t\right) dt. \tag{5.10}$$

Für jede Lösung von (5.8) liegt x zentral im Lösungsraum L. Dann ist auch p maximal für $B(x,p) \subseteq L$. Es wird also das Optimierungsproblem

$$\max_{x \in S} \{p\}$$

unter der Bedingung, dass

$$B(x,p) \subseteq L$$

gelöst.

Hierin ist der Name *Design-Centering* begründet. Algorithmen zu dieser Problemstellung werden in dieser Arbeit nicht weiter untersucht. Verwiesen sei an dieser Stelle auf Algorithmen in [1, 49] und [76].

[1] Im gezeigten Beispiel sind die Haupt- und die Nebenachse der Ellipse parallel zu den Achsen des Suchraums, was aus der Unkorreliertheit der Streuparameter rührt. Dies ist ein typisches Szenario in der Schaltungsoptimierung.

Kapitel 6

Vergleich von Algorithmen zur Optimierung integrierter Schaltungen

Im Rahmen dieser Arbeit wurden ausschließlich Mehrzieloptimierungsalgorithmen betrachtet. Die verwandten Algorithmen sind zum einen die mengenorientierten, deterministischen *GAIO* Algorithmen (Kapitel 2), die die zu approximierende Pareto-Menge sukzessiv durch immer feinere Boxen einschließen. Zum anderen sind evolutionäre Algorithmen nach der Vorlage des *SPEA*, bei dem Fitnesswerte abhängig von Dominanzverhältnissen vergeben werden, zur Optimierung angepasst, erweitert und eingesetzt worden (Kapitel 3).

In diesem Kapitel werden beide Algorithmentypen und deren Anwendbarkeit zum Entwurf ressourceneffizienter integrierter Schaltungen miteinander verglichen. Außerdem werden Vorteile der algorithmischen Entwurfsraumexploration gegenüber der Exhaustionsmethode (auch bekannt als *Brute-Force*) diskutiert.

6.1 SPEA, GAIO und Exhaustionsmethode

Der *Strength-Pareto-Evolutionary-Algorithm SPEA* wurde von Zitzler und Thiele in [82] als eine Kombination und Erweiterung vorhergehender Mehrzieloptimierungs-EAs (MOEAs) eingeführt und seine Leistungsfähigkeit im Vergleich mit vier vorhergegangenen MOEAs aufgezeigt. Verglichen wurde er dort mit einem genetischen Algorithmus, bei dem die Fitnesswerte der dominierten und nicht-dominierten Individuen sortiert vergeben werden [66], einem genetischen Algorithmus, der die Selektion für jede Zielfunktion separat ausführt [25], durch einen Ansatz paralleler Einzieloptimierungen gewichteter Summen [32] und einem genetischer Algorithmus, bei dessen Selektionsverfahren die Größe der Nischen (Abstand zum nächsten Individuum) berücksichtigt wird [37] [38]. In [82] werden speziell zum Vergleich Bewertungsgrößen definiert, die die Ergebnisse evolutionärer Algorithmen miteinander vergleichbar machen. Für den Anwendungsfall des 1/0-Knapsack Problems

ist dort als Ergebnis ausführlicher Testes gezeigt, dass der SPEA mindestens 87 % und zumeist 100 % der nicht-dominierten Lösungen der anderen vier EAs abdeckt. Außerdem deckt jeder Algorithmus weniger als 5 % der Fläche unterhalb der SPEA-Pareto-Front ab. Weitere Vorteile des SPEA sind dokumentiert an verschiedenen Anwendungsbeispielen in [83] und [80]. Eine Weiterentwicklung, den *SPEA2*, gibt es in [81].

In einem Überblick über 20 Jahre MOEAs [13] wird der SPEA als erste effiziente Weiterentwicklung dieser Algorithmen benannt. Es ist vor allem die Fitnesszuweisung, die gleichzeitig eine große Diversität und gleichmäßige Abdeckung der Pareto-Front garantiert. Außerdem wird hier ϵ-Dominanz als ein neuer Trend (Zeitpunkt 2006) erwähnt, mit der die Feinheit der Approximation der Pareto-Front durch eine Population kontrolliert wird. Auf den Begriff der ϵ-Dominanz wird weiter unten eingegangen.

6.1.1 Laufzeitanalyse

Nach der in Abbildung 3.2 dargestellten Struktur hat die Laufzeit von evolutionären Algorithmen die Form

$$\mathcal{O}\left(N + T \cdot \left(\left(c_{\text{select}} + c_{\text{cross}} + c_{\text{mut}}\right) \cdot N\right)\right)$$

für Konstanten c_{select}, c_{cross} und c_{mut}, die von der Wahl der Selektion, des Mutations- und Crossoveroperators abhängig sind. Zur Erinnerung: N bezeichnet hier die Größe der Population und T die Anzahl der Schritte.

Die Laufzeit des Unterteilungsalgorithmus der GAIO Toolbox ist bestimmt durch die Dimension n des Suchraums $S \subset \mathbb{R}^n$ und der Unterteilungstiefe der Boxen l. Nach Schütze [61] kann die Anzahl der Boxen, die die $(k-1)$-dimensionale Pareto-Menge überdecken, mit $\left(\sqrt[n]{2^{k-1}}\right)^l$ abgeschätzt werden. Die gesamte Anzahl der im Algorithmus ausgewerteten Boxen ergibt sich durch Summation über die Boxen aller Unterteilungstiefen zu

$$n_{\text{Boxen}} = \sum_{i=0}^{l} 2^{\frac{(k-1)i}{n}}. \tag{6.1}$$

Für den häufig auftretenden Fall in den betrachteten Anwendungsbeispielen der Schaltungsoptimierung gilt $n = k - 1$. Hierfür lässt sich (6.1) als endliche geometrische Summe vereinfachen zu

$$n_{\text{Boxen}} = \sum_{i=0}^{l} 2^i = \frac{1 - 2^{l+1}}{1 - 2} = 2^{l+1} - 1. \tag{6.2}$$

Um eine gewünschte Approximationsgenauigkeit ϵ_i für jede Dimension $i = 1..n$ zu erreichen, müssen die Boxen in jeder Richtung entsprechend oft halbiert werden. Seien L_i die Kantenlängen der Mutterbox (=Suchraum), so muss diese in jeder Dimension l_i-mal unterteilt werden, damit gilt

$$L_i \cdot 2^{-l_i} \leq \epsilon_i, \qquad i = 1..n. \tag{6.3}$$

Die nötige Unterteilungstiefe l ist die Summe der Unterteilungen jeder Dimension:

$$l = \sum_{i=0}^{n} l_i. \tag{6.4}$$

Die Anzahl der Punkte pro Box kann für jede Unterteilungstiefe einzeln gewählt werden. Diese Testpunkte können zufällig und in Gitterform verteilt sein (vgl. [18]). Sinnvollerweise sollten für größere Boxen zwar mehr Testpunkte gewählt werden als für kleinere, aber hier kann zur Vereinfachung angenommen werden, dass jede Box gleich viele Punkte enthält und daher die Anzahl der archivierten Punkte n_{Punkte} linear mit der Anzahl der Boxen wächst. Der Dominanztest wird in Zeit $\mathcal{O}\left(\left(n_{\text{Punkte}}\right)^2\right)$ durchgeführt. Alle weiteren Operationen hängen nur linear von der Anzahl der Boxen ab (vgl. Kapitel 2). Die gesamte Laufzeit ist also $\mathcal{O}\left(2^{2l+2}\right)$ und wächst daher exponentiell mit der Unterteilungstiefe, die nach (6.4) abhängig von der Dimension des Suchraums ist.

6.1.2 Konvergenz

Die evolutionären Algorithmen sind stochastische Suchverfahren und damit Heuristiken. Heuristik (altgr. heurísko „ich finde“; heuriskein, „(auf-)finden“, „entdecken“) bezeichnet die Kunst, mit begrenztem Wissen und wenig Zeit zu guten Lösungen zu kommen (Frei übersetzt aus [28].). Sie finden somit in Ingenieurdisziplinen verbreitet Anwendung, um möglichst schnell eine Routine zur algorithmischen Entwurfsraumexploration zur Hand zu haben, die basierend auf Erfahrungswerten, selbst in anfangs unübersichtlichen Suchräumen „gute“ Lösungen findet. Aufgrund ihrer Natur ist aber ein Nachweis der Konvergenz, das heißt der Nachweis über die schrittweise Verbesserung der Approximation an eine Lösungsmenge, nur schwer oder eingeschränkt möglich, wie folgend diskutiert wird.

Das Theorem 1 beschreibt unter welchen Voraussetzungen sich die von einem EA gesammelten nicht-dominierten Punkte der Pareto-Menge annähern.

Theorem 1 Es sei ein MOP (1.4) mit stetigem f gegeben, der Suchraum $S \subset \mathbb{R}^n$ sei kompakt und P bezeichne die zugehörige Pareto-Menge in S. Weiterhin enthalte $S \setminus P$ keine schwachen Pareto-Punkte und für die Populationsfolge $\{P_t\}_t$ des EA gelte

$$\forall x \in S \text{ und } \forall \delta > 0 : \qquad P\left(\exists t \in \mathbb{N} : P_t \cap B_\delta\left(x\right) \cap S \neq \emptyset\right) = 1. \tag{6.5}$$

Dann generiert ein Archiv

$$\begin{aligned} &A_0 = \emptyset, \\ &A_{t+1} = \{x \in P_{t+1} \cup A_t : x \text{ nicht-dominiert in } P_{t+1} \cup A_t\} \end{aligned} \tag{6.6}$$

eine Folge $\{A_t\}_t$, für die gilt

$$\lim_{t \to \infty} h\left(f\left(P\right), f\left(A_t\right)\right) = 0 \text{ mit Wahrscheinlichkeit eins,} \tag{6.7}$$

wobei die Hausdorff Metrik h den Abstand zwischen zwei Mengen misst.
Beweis s. [61]

Die Bedingung (6.5) fordert, dass die vom EA generierten Individuen den Suchraum beliebig fein absuchen. Nur dann kann die Konvergenz (6.7) garantieren, dass sich das Archiv mit allen Pareto-Punkten füllt. $B_\delta(x)$ bezeichnet die Kugel um den Mittelpunkt x mit Radius δ.

Ein Algorithmus, der (6.5) erfüllt, ist allerdings ein sehr ineffizienter Algorithmus, der zudem unendlich lange laufen müsste (falls S nicht nur endlich viele Elemente enthält). Für die mathematische Konvergenzanalyse ist diese Forderung allerdings unerlässlich. In der Praxis wird man versuchen, in weiteren Schritten Punkte zu generieren, die der Pareto-Menge immer näher kommen. Diese Eigenschaft nachzuweisen ist bei heuristischen Crossover- und Mutationsstrategien unmöglich.

Der große Nachteil der EAs mit einem simplen Archivierer, wie in (6.6), ist, dass sie mit (6.7) zwar Konvergenz garantieren, aber nicht nach endlich vielen Schritten, so dass kein Abbruchkriterium, vergleichbar mit (6.3), gefordert werden kann. Um dieses Ziel zu verfolgen, werden ϵ-Archivierer eingeführt, die auf dem Konzept der ϵ-Dominanz Punkte im Archiv aufnehmen.

Definition ϵ-Dominanz „$\prec_\epsilon$“ im $\mathbb{R}^m$
Es sei $\epsilon \in \mathbb{R}^n_+$. Für $a, b \in \mathbb{R}^m$ gilt $a \prec_\epsilon b$ (a ϵ-dominiert b), falls $a - \epsilon \prec b$.

Algorithmus 2 zeigt ein Beispiel eines simplen ϵ-Archivierers nach [63].

Algorithmus 2 $A := ArchiveUpdate1_\epsilon(P, A_0)$

```
1: A := A_0
2: for all p ∈ P do
3:     if ∃a ∈ A : f(a) ≺_{ε/3} f(p) then
4:         CONTINUE                          ▷ Zeilen 6-11 nicht ausführen
5:     end if
6:     for all a ∈ A do
7:         if a ≺ p then
8:             A := A \ {a}
9:         end if
10:    end for
11:    A := A ∪ {p}
12: end for
```

Definition ϵ-approximierende Pareto-Menge
$A_\epsilon \subseteq S \subset \mathbb{R}^m$ ist eine ϵ-approximierende Pareto-Menge in S zum MOP (1.4), wenn jedes $x \in S$ ϵ-dominiert wird durch mindestens ein Element $a \in A_\epsilon$, also

$$\forall x \in S : \exists a \in A_\epsilon : f(a) \prec_\epsilon f(x).$$

Nun kann mit Theorem 2 eine Aussage formuliert werden, die ein theoretisches Abbruchkriterium abhängig von einer vorgegeben Approximationsgüte liefert.

Theorem 2 Gelten die gleichen Voraussetzungen wie in Theorem 1, dann erzeugt ein EA mit dem ϵ-Archivierer wie in Abbildung 6.1.2 eine Folge von Archiven $\{A_t\}_t$, so dass mit Wahrscheinlichkeit eins ein $t_0 \in \mathbb{N}$ existiert, so dass A_t für alle $t \geq t_0$ eine ϵ-approximierende Pareto-Menge ist.
Beweis s. [61]

Mit dem Archivierer wird also nach t_0 Schritten ein Archiv A_{t_0} erzeugt, welches eine ϵ-approximierende Pareto-Menge zum MOP ist. Der EA sollte terminieren, wenn keine weiteren Punkte dem Archiv hinzu gefügt werden können. Für den Programmierer des EA besteht damit allerdings immer noch ein Problem bei der Wahl des Abbruchkriteriums. Nur weil in einem Schritt dem Archiv keine Individuen hinzugefügt werden konnten, heißt dies nicht, dass nicht nach mehreren weiteren Schritten wieder Populationen generiert werden, bei denen dies wieder möglich ist.

Das vorrangige Ziel, das durch Einführung der ϵ-Archivierer verfolgt wird, ist es, einen Kompromiss zwischen Archivgröße und Feinheit der Approximation zu finden, der durch einfache Kriterien leicht steuerbar ist und effizient implementiert werden kann. Denn mit der Größe des anwachsenden Archivs steigt die Anzahl der für den Dominanztest durchzuführenden Vergleiche quadratisch. In den Anfängen wurde zu diesem Zweck nach der Erweiterung des Archivs in jedem Schritt ein *Clustering* durchgeführt, bei dem Individuen dichter besiedelter Stellen teilweise gelöscht werden. Dieses Clustering hätte allerdings ebenfalls quadratische Laufzeit. Mit dem ϵ-Archivierer lassen sich so zwei Schritte effektiv zusammenfassen.

Allerdings ist die Laufzeit des Dominanztests für die Optimierung integrierter Schaltungen zu vernachlässigen, da die Dauer der Funktionsauswertungen, bei denen für jeden Punkt Schaltungssimulationen durchgeführt werden müssen, fast die gesamte Laufzeit der algorithmischen Suchraumexploration ausmachen. Der Vergleich beider Algorithmentypen unter Berücksichtigung dieser Anwendungen ist Thema des nächsten Abschnitts 6.1.3.

Für den GAIO-Unterteilungsalgorithmus mit Abstiegsverfahren wird in [18] Konvergenz gegen substationäre Punkte bewiesen, die eine Obermenge der Pareto-Menge bilden. Für die hier benutzte Version, in der eine Box stellvertretend durch einzelne Testpunkte ausgewertet wird, lässt sich eine vorgegebene Approximationsgüte ϵ_i nach (6.3) durch Wahl der Unterteilungstiefe

$$l_i \geq \log_2 \left(\frac{L_i}{\epsilon_i} \right)$$

für jede Dimension $i = 1..n$ erreichen.

6.1.3 Anwendung: Optimierung integrierter Schaltungen

Ein direkter Vergleich der verwandten Mehrzieloptimierungsalgorithmen, mengen-orientierten deterministischen GAIO-Algorithmen und den populationsbasierten, stochastischen evo-

lutionären Algorithmen ist nur eingeschränkt möglich, da die Konstruktionsziele beider Algorithmentypen grundverschieden sind. Wie oben klar geworden ist, wird bei den GAIO-Methoden eine Feinheit der Approximation vorgegeben und es kann abhängig von dieser die zu erwartende Laufzeit bestimmt werden. Allerdings steigt diese Laufzeit exponentiell mit der Unterteilungstiefe. Bei den EAs werden viele Einstellungen nach Erfahrungswerten bestimmt und die Berechnugen halten nach einer vorgegebenen Anzahl von Schritten an, ohne dass dem Benutzer klar ist, wie weit seine aktuelle Lösung von der Pareto-Menge entfernt sein kann. Der Vorteil dieser Heuristik liegt in der simplen Struktur, die nach biologischem Vorbild entworfen wurde und mittlerweile mehr als zwanzig Jahre erfolgreich in vielen Disziplinen der Ingieurwissenschaften Anwendung gefunden hat.

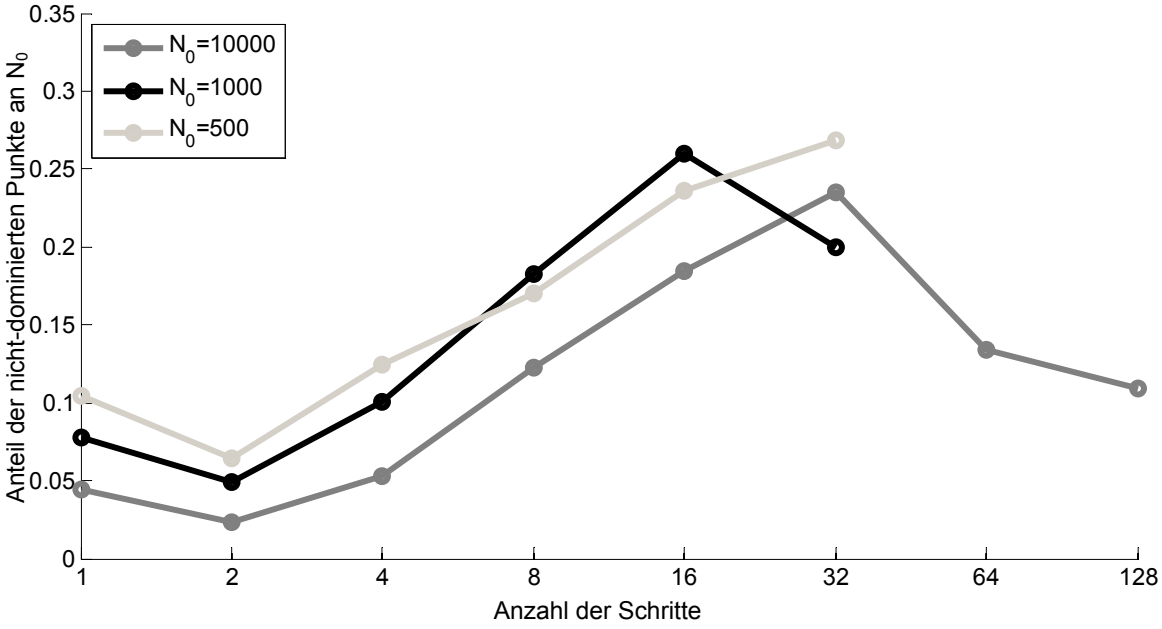
Nur unter Betrachtung konkreter Beispiele besteht die Möglichkeit, Vor- und Nachteile beider Algorithmentypen zu diskutieren. In diesem Abschnitt wird der Versuch unternommen, dies für die Optimierung eines Inverters und eines NAND-Gatters zu tun. Als schwierig stellt sich die Suche nach einem geeigneten Maß für die Qualtität des Optimierungsergebnisses heraus. Schon alleine zum Vergleich verschiedener EAs untereinander gibt es keine Einigkeit über das aussagekräftigste Verfahren sondern nur viele verschiedene Vorschläge [16, 42, 43, 46, 70, 73, 82, 83]. Da sich die zwei-dimensionalen Suchräume beider Beispiele grafisch darstellen lassen, kann der Vergleich visuell bewerkstelligt werden.

Zentrale Anforderung an die Algorithmen zur Schaltungsoptimierung ist die Zahl der Funktionsauswertungen, die die Laufzeit immer dominieren, da jedesmal eine Schaltungssimulation durchgeführt werden muss. Für die in dieser Arbeit betrachteten Zellen beträgt die Simulationszeit ca. 1-5 Sekunden. Aus diesem Grund gilt es zum Vergleich der verwandten Algorithmen, die Anzahl der Funktionsauswertungen der Approximation der Pareto-Mengen gegenüberzustellen.

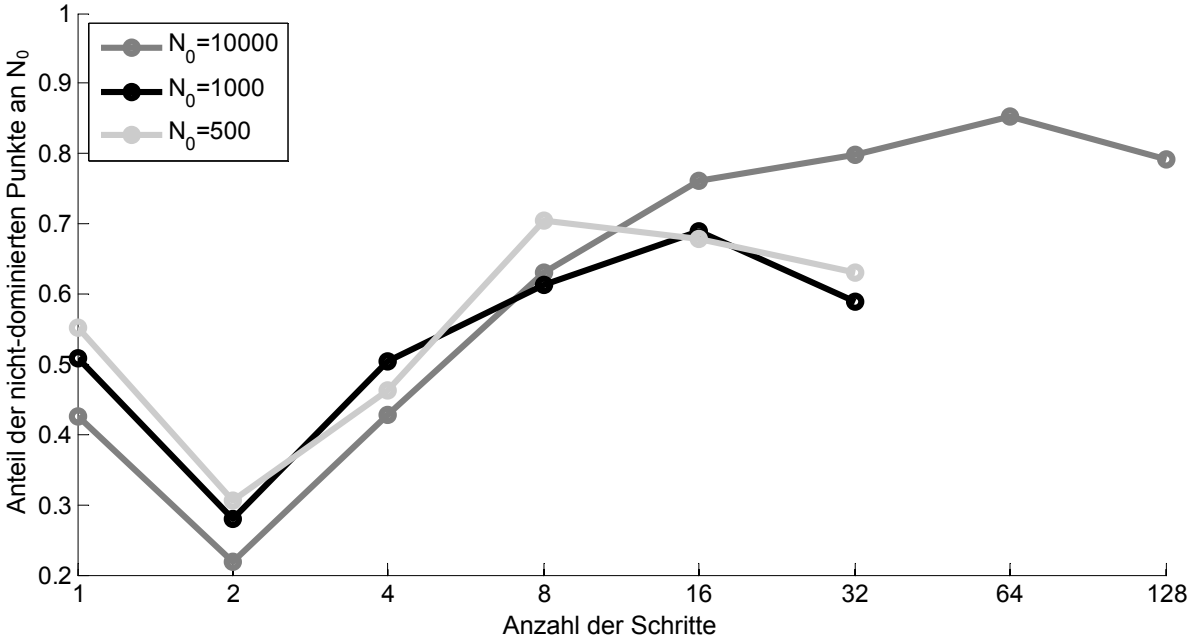
In Abbildung 6.1a ist die Ausbeute des EA bei der Optimierung des CMOS-Inverters in 65 nm abhängig von der Anzahl der Schritte T, die auf eine fest vorgegebene Anzahl von Funktionsauswertungen N_0 verteilt wurden, dargestellt.[1] Die Größe der Populationen ist gewählt als $N = N_0/T$ unter der Annahme, dass immer alle Individuen durch Crossover und Mutation neu erzeugt werden. Als Ausbeute sei der Anteil der gefundenen nichtdominierten Punkte an N_0 bezeichnet. Nach denselben Vorgaben ist in Abbildung 6.1b die Ausbeute bei der Optimierung des 65 nm CMOS-NAND-Gatters bei $U_{\mathrm{dd}} = 300\,\mathrm{mV}$ dargestellt. Selbstverständlich wurde in diesem Versuchsaufbau kein Clustering angewandt.

Erkennen lässt sich in beiden Abbildungen, dass es eine optimale Verteilung der Funktionsauswertungen pro Schritt zu geben scheint, dort wo die einzelnen Kurven ihren höchsten Wert annehmen. Während die Aufteilung der Funktionsauswertungen auf eine kleine Anzahl von Schritten wenig Vorteil gegenüber einem einzelnen Schritt bringt, lässt sich die Ausbeute beim Inverter fast vervierfachen und beim NAND-Gatter immerhin mehr als verdoppeln. Für mehr Schritte fällt die Ausbeute auf Grund der kleiner werdenden Populationsgrößen wieder. Die Messwerte sind Ergebnisse stochastischer Prozesse. Sie variieren nicht nur mit der Zufälligkeit der Operatoren sondern auch mit den Crossover-

[1] Hier wird die Initialisierung als erster Schritt gezählt.



(a) Inverter



(b) NAND

Abbildung 6.1: Anteil der nicht-dominierten Punkte an N_0 Funktionsauswertungen.

und Mutationswahrscheinlichkeiten p_c und p_m. Hier sind $p_c = 0.8$ und $p_m = 0.1$. Außerdem sind die Ergebnisse stark abhängig vom Anwendungsbeispiel. Hat die Pareto-Menge einen größeren Anteil am Suchraum, wie es beim NAND-Gatter bei 300 mV der Fall ist, so ist die Ausbeute erwartungsgemäß höher.

Insgesamt lässt sich aus den beiden Beispielen ableiten, dass es zur „guten“ Ausbeute von N_0 Funktionsauswertungen zwar Richtgrößen für die Anzahl der Schritte zu geben scheint, diese aber mit N_0 und dem Anwendungsbeispiel variieren. Sicherlich werden andere Crossover- und Mutationswahrscheinlichkeiten ebenfalls andere Ausbeuten liefern. All diese Faktoren sind im Vorhinein schlecht abzuschätzen und optimale Einstellungen sind nur durch etliche Tests zu ermitteln. Da dies für jedes Beispiel neu nötig wäre, muss der Benutzer bei der Wahl der freien Parameter des Algorithmus leider auf die Erfahrungen vorheriger Optimierungen vertrauen.

Für eine erste grobe Abschätzung können einige zufällig oder gitterförmig ausgewählte Punkte im Suchraum ausgewertet und gegenseitig auf Dominanz überprüft werden. Dies ist ein sogenannter *Brute-Force* Ansatz. In den Abbildungen 3.11 und 3.13b wurde er verwendet, um die Lage der Pareto-Menge im Suchraum graphisch anzudeuten, ohne dabei eine feinere Suche durchzuführen. Die Exhaustionsmethode entspricht der Initialisierung des EA. Die zu erwartende Ausbeute an den Testpunkten lässt sich in Abbildung 6.1 bei einem Schritt ablesen. Zu erkennen ist dort, dass der EA nach einigen Schritten fast immer besser ist, als die Exhaustionsmethode. Anzumerken ist außerdem, dass nicht-dominierte Punkte nicht gleich Pareto-Punkte sind. Durch mehrere Schritte, in denen sich die Populationen auf die Pareto-Menge zubewegen, erhöht sich die Wahrscheinlichkeit, dass gefundene Punkte Pareto-optimal sind.

Unterteilungstiefe l	0	1	2	3	4	5	6	7	8	9	10	11
Punkte pro Box	200	100	50	25	12	12	6	6	3	3	1	1
Anzahl der Boxen	1	2	3	5	6	10	13	22	29	47	68	117

Tabelle 6.1: Punkte pro Box in jeder Unterteilungstiefe zur Approximation des CMOS-Inverters

In Tabelle 6.1 ist aufgeführt, mit wie vielen Punkten pro Box der GAIO-Algorithmus in der jeweiligen Unterteilungstiefe l zur Optimierung des CMOS-Inverters aufgerufen wurde. Nach zwölf Unterteilungen (Unterteilungstiefe $l = 11$, je sechs in jeder Dimension) hat GAIO

$$1\cdot200+2\cdot100+3\cdot50+5\cdot25+6\cdot12+10\cdot12+13\cdot6+22\cdot6+29\cdot3+47\cdot3+68\cdot1+117\cdot1 = 1490$$

Funktionsauswertungen durchgeführt. Entsprechend sind zum Vergleich Ergebnisse des EA für $N_0 = 1490$ verteilt auf $T = 1, 2, 4, 8, 16, 32$ Schritte in Abbildung 6.2 dargestellt. Grau hinterlegt sind die Ergebnisse der Boxapproximationen.

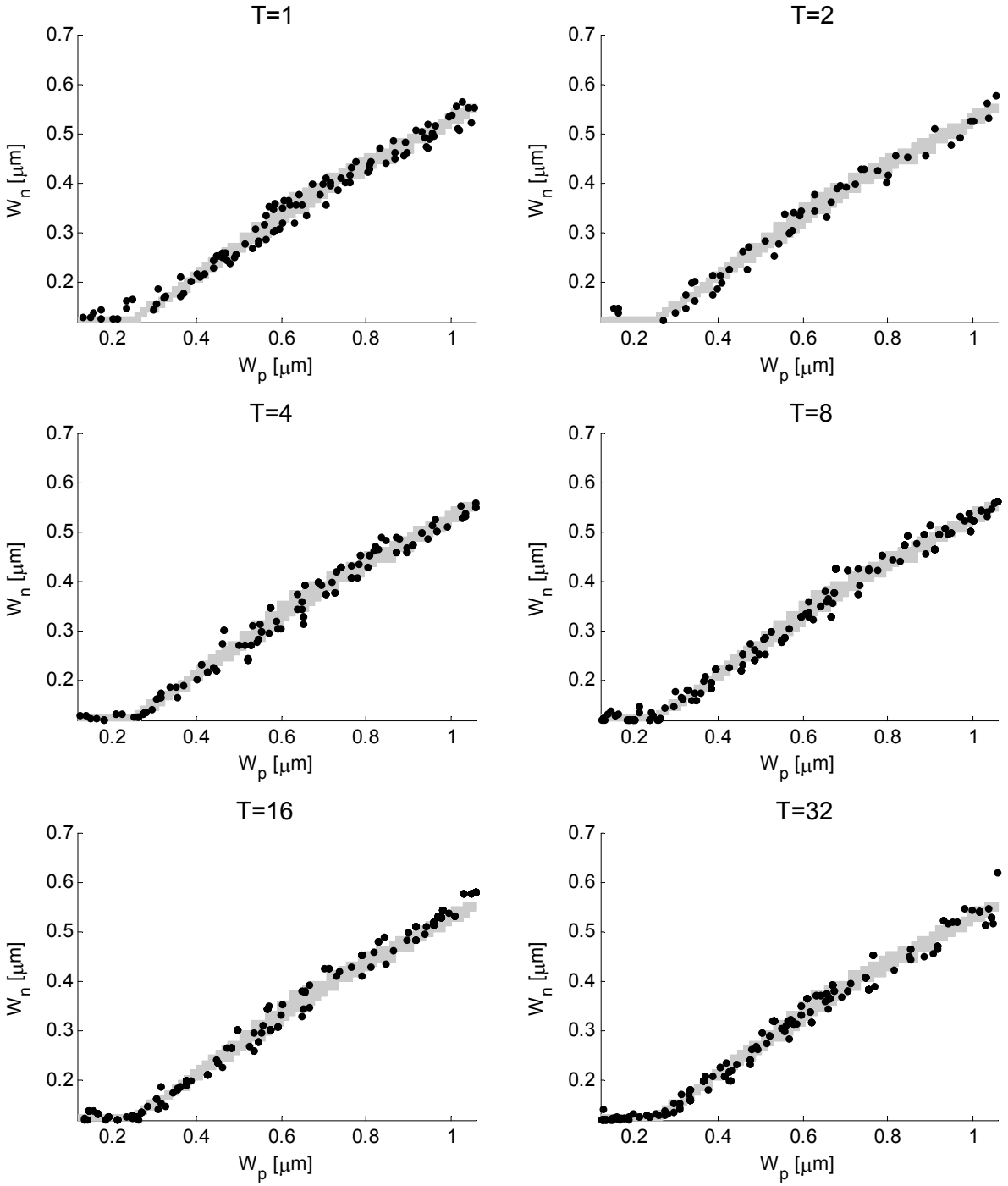


Abbildung 6.2: Approximation der Pareto-Menge des 65 nm CMOS-Inverters durch den evolutionären Algorithmus mit etwa 1490 Funktionsauswertungen verteilt auf $T = 1, 2, 4, 8, 16, 32$ Schritte. Zur Referenz ist in Hellgrau das Ergebnis des GAIO-Algorithmus, der dieselbe Anzahl an Funktionsauswertungen benötigt hat.

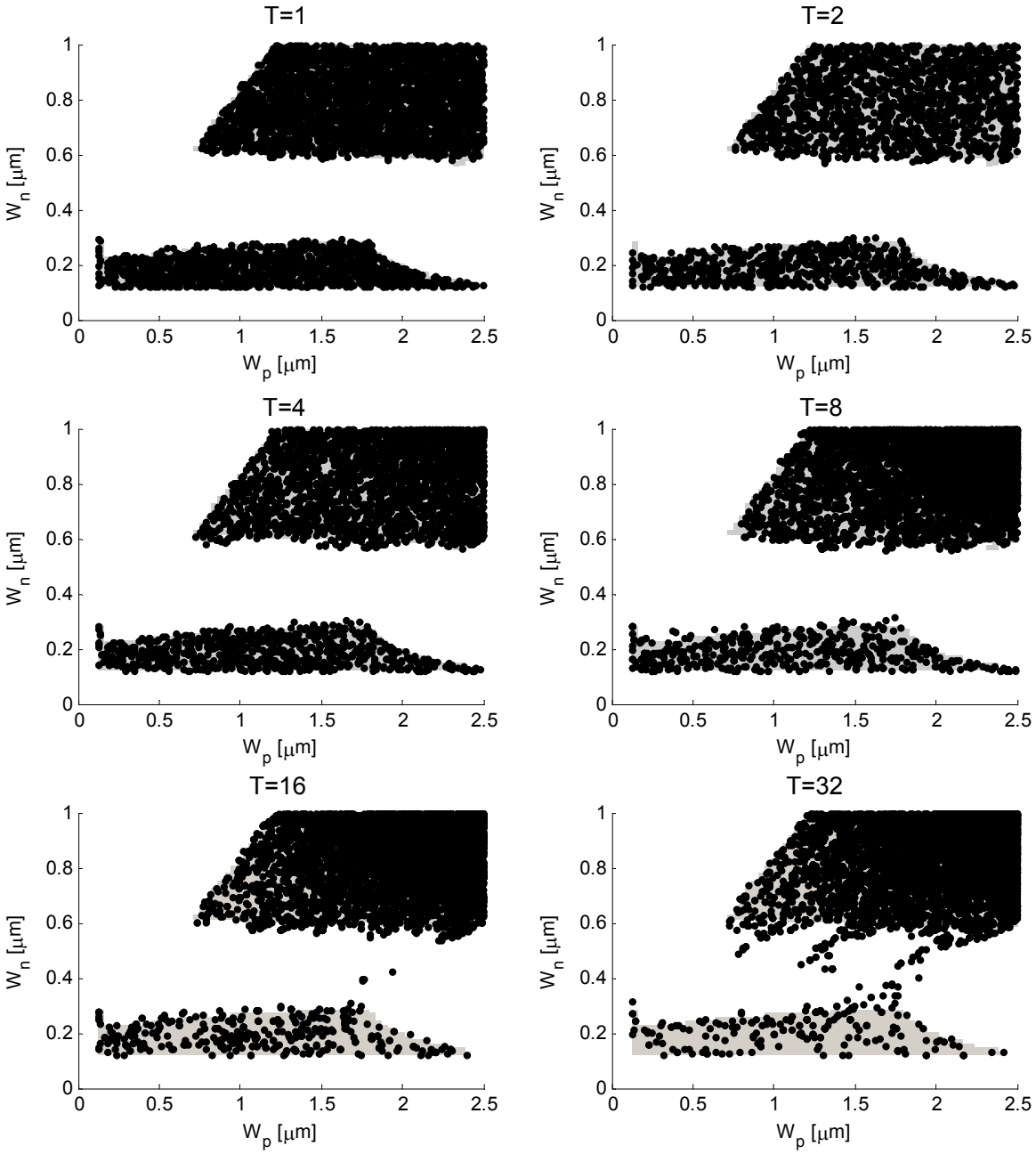


Abbildung 6.3: Approximation der Pareto-Menge des 65 nm CMOS-NAND-Gatter betrieben bei 300 mV durch den evolutionären Algorithmus mit etwa 10000 Funktionsauswertungen verteilt auf $T = 1, 2, 4, 8, 16, 32$ Schritte. Zur Referenz ist in Hellgrau das Ergebnis des GAIO-Algorithmus hinterlegt, der ähnlich viele Funktionsauswertungen benötigt hat.

Man erkennt, dass die Pareto-Menge durch die Punktwolken wesentlich ungenauer dargestellt wird als durch die Boxen. GAIO liefert für alle Fälle bessere Approximationen. Bei Vergleich von T=1 mit T=8 fällt auf, dass für den EA eine höhere Ausbeute an nicht-dominierten Punkten pro Funktionsauswertung nicht auch ein besseres Bild der Approximation der Pareto-Menge bedeutet. Für T=16 ist vor allem der Bereich kleiner Transistoren wesentlich besser abgedeckt als beim Brute-Force-Ansatz. Vergleiche mit mehr Funktionsauswertungen haben gezeigt, dass $N_0 = 1490$ für dieses Beispiel noch zu gering ist. Der EA benötigt größere Populationen, um durch Crossover und Mutation eine bessere Approximation der Pareto-Menge zu erzeugen. GAIO liefert also ein wesentlich besseres Ergebnis als der EA.

Am Beispiel des NAND-Gatters bei 300 mV kann man in Abbildung 6.1b weitere Eigenschaften beider Algorithmen erkennen. Hier benötigt der GAIO-Algorithmus etwa 10000 Funktionsauswertungen, da durch die große Pareto-Menge wenige Boxen aus der Boxsammlung entfernt werden. Es ist auch die große Fläche, die die Trefferquote des Brute-Force erhöht. Für T=1 ist die Approximation außerdem besser als für alle anderen T. Dies liegt daran, dass das Crossover immer wieder Individuen in der Lücke der beiden Teilmengen generiert. Bei T=32 ist klar zu sehen, dass der EA für dieses Beispiel schlecht geeignet ist, da der untere Teil kaum abgedeckt ist, während der obige ausfranst. Aus diesem Beispiel ist abzulesen, dass dieser GAIO-Algorithmus zuverlässiger approximiert, da er sich nicht an der Disjunktheit der Lösungsmenge stört. Man beachte, dass dies nicht für alle GAIO-Algorithmen gilt. Der Algorithmus, der mit einem Abstiegsverfahren arbeitet, hätte eine Obermenge bestimmt, die sowohl beide Teile der Pareto-Menge als auch die Lücke enthalten.

Für die beiden oberen Beispiele wurde festgestellt, dass die Algorithmen bei gleicher Anzahl an Funktionsauswertungen, d.h. gleicher Laufzeitdauer, eine genauere Approximation der Pareto-Menge liefern. Gestartet wurden die evolutionären Algorithmen, nachdem bekannt war, wie lange die GAIO-Algorithmen gelaufen sind. Während man beim EA die Anzahl der Funktionsauswertungen im Vorhinein unabhängig vom Anwendungsbeispiel kennt, müsste man beim Boxalgorithmus vom schlechtesten Fall ausgehen, bei dem keine Boxen verloren gingen. Folgt man den Werten aus Tabelle 6.1, wären für die beschriebene Situation nach sechs Unterteilungen schon

$$2^0 \cdot 200 + 2^1 \cdot 100 + 2^2 \cdot 50 + 2^3 \cdot 25 + 2^4 \cdot 12 + 2^5 \cdot 12 = 1376$$

Punkte ausgewertet. Mit einer Unterteilung mehr wären bereits $1376 + 2^6 \cdot 6 = 1760$ Schaltungen simuliert. Abbildung 6.4 zeigt nochmals die Ergebnisse des EA aus Abbildung 6.2. Diesmal sind allerdings die Boxen der Unterteilungstiefe fünf hinterlegt. Nun ist ein Vorteil der evolutionären Algorithmen klar erkennar. Im Nachhinein stellt man allerdings fest, dass GAIO nur etwa 60% der erwarteten Zeit benötigt hat.

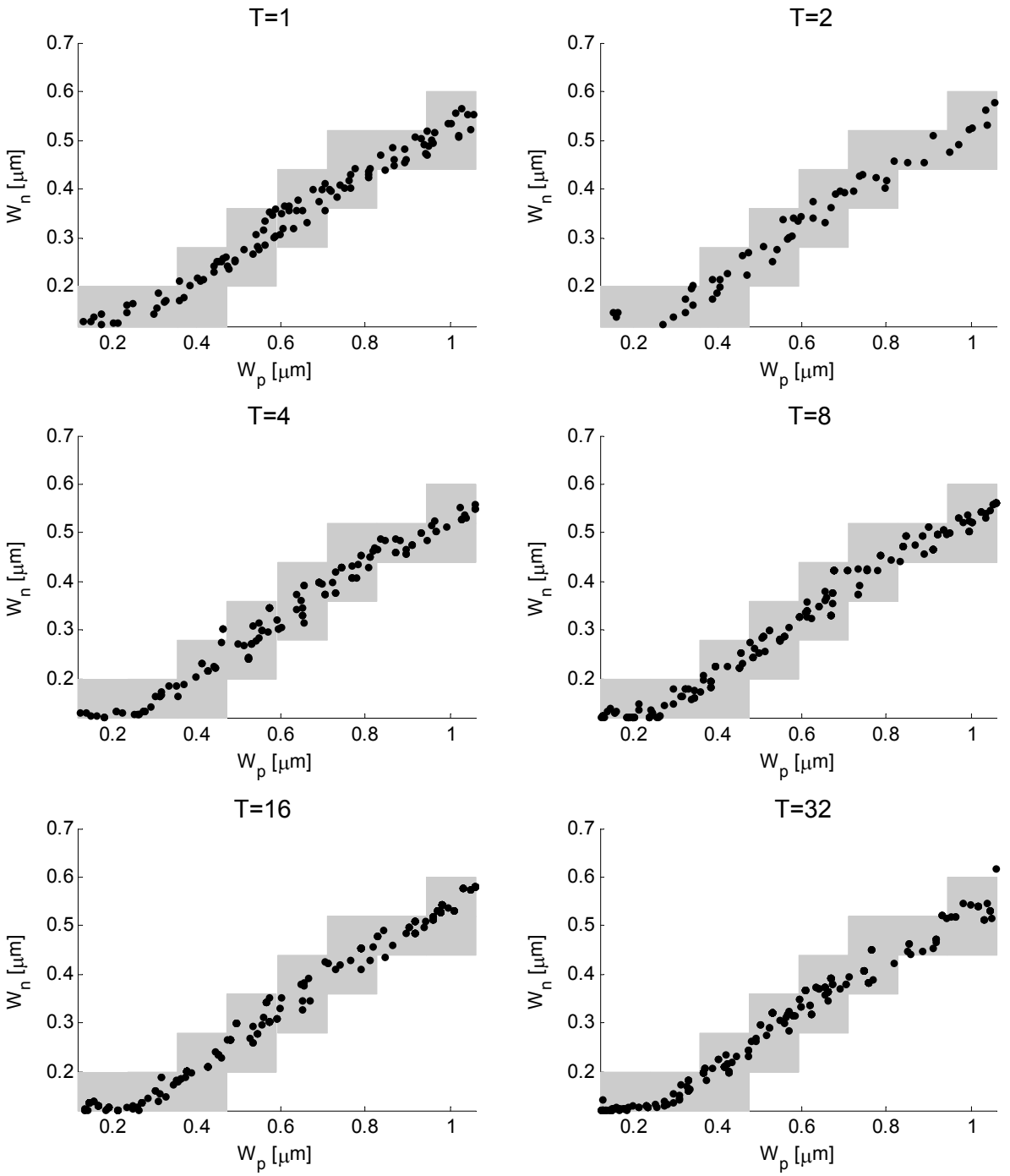


Abbildung 6.4: Approximation der Pareto-Menge des 65 nm CMOS-Inverters durch den evolutionären Algorithmus mit etwa 1490 Funktionsauswertungen verteilt auf $T = 1, 2, 4, 8, 16, 32$ Schritte. Zur Referenz ist in Hellgrau das Ergebnis des GAIO-Algorithmus hinterlegt, der im schlechtesten Fall mit 1376 ähnlich viele Funktionsauswertungen benötigt hätte.

Die evolutionären Algorithmen sind, wie oben schon erwähnt, eine Heuristik, die mit einfachen Mitteln einen ersten Überblick über eine Lösung schaffen möchte. Die Entwicklung der EAs ist vor allem aus Anwendungsfällen motiviert, bei denen der Suchraum sehr komplex und schwer überschaubar ist [79]. Vorteile in Geschwindigkeit und Approximationsgüte gegenüber den GAIO-Algorithmen können also erst entstehen, wenn die Dimension des Suchraums größer wird, denn die Anzahl der Boxen erhöht sich dann exponentiell. Für mehr als drei-dimensionale Beispiele ist es allerdings noch schwieriger, aussagekräftige Vergleiche zu machen, da sich die Suchräume nicht mehr grafisch darstellen lassen. Die Suchräume der Mehrzieloptimierungsprobleme, die im Rahmen dieser Arbeit von den betrachteten Schaltungen abgeleitet wurden, sind nicht nur, aber auch, zum Zweck der Anschaulichkeit entsprechend klein gehalten. Allerdings wäre aufgrund der guten Parallelisierbarkeit eine längere Laufzeit der GAIO-Algorithmen durchaus akzeptabel. Durch die nachgewiesene Konvergenz ist sichergestellt, dass die Algorithmen die Menge aller ressourceneffizienten Schaltungen beliebig genau approximieren. Da auch kleine Verbesserungen in den Zielgrößen jeder einzelnen Standardzelle die Schalteigenschaften eines auf ihnen basierenden Prozessors erwartungsgemäß linear verbessern, ist hier der Kompromiss zwischen Laufzeit und Approximationsgüte zugunsten der Approximationsgüte sinnvoll.

Zusammenfassung

Beim Entwurf integrierter Schaltungen stehen dem Entwickler viele Ressourcen und freie Parameter zur Verfügung, mit denen er den Ressourcenverbrauch beeinflussen und steuern kann. Die Aufgabe des Entwicklers besteht darin, den Verbrauch der einen Ressource gegen den der anderen abzuwägen und mittels der freien Parameter einen optimalen Kompromiss herbeizuführen. Optimal ist ein Kompromiss dann, wenn von keiner Ressource weniger verbraucht werden kann, ohne den Anteil der anderen zu erhöhen. Einen solchen Entwurfspunkt im Suchraum nennt man ressourceneffizient. Die Suche nach solchen Punkten wird als Entwurfsraumexploration bezeichnet. Mathematisch entspricht die Suchraumexploration dem Lösen eines Mehrzieloptimierungsproblems. Es ergibt sich daher auf natürliche Weise, durch Definition von Ressourcenmaßen, aus einer integrierten Schaltung ein Mehrzieloptimierungsproblem, dessen Lösung (Pareto-Menge) sich algorithmisch approximieren lässt. Im Rahmen dieser Arbeit wurde dieser Mehrzieloptimierungsansatz verfolgt, um die Entwurfsraumexploration integrierter Schaltungen durch mathematische Mehrzieloptimierungsalgorithmen zu unterstützen.

Für den Anwendungsfall der Schaltungsoptimierung weiterentwickelt wurden die mengenorientierten Algorithmen der *Global Analysis of Invariant Objects*(GAIO) Toolbox. Um den Algorithmus zu unterstützen, wurde ein Server-Client-System entwickelt, dass es ermöglicht, viele Schaltungen parallel zu simulieren und auszuwerten. Die Laufzeit wird somit fast linear mit der Anzahl der angeschlossenen Clients verringert. Die Ergebnisse einer Optimierung von Standardzellen unterstützen den Entwickler mit einer Approximation der gesamten Menge an ressourceneffizienten Dimensionierungen jeder Zelle bezüglich Geschwindigkeit, Energieverbrauch und Robustheit gegenüber Rauschen. Vergleiche mit Schalteigenschaften kommerzieller Gatter in 65 nm und 90 nm Technologien zeigen, dass mit Einsatz der Algorithmen sogar Verbesserungen möglich sind.

Nachdem der Mehrzieloptimierungsansatz für Standardzellen, betrieben bei voller Versorgungsspannung, eingeführt und dessen Vorteile aufgezeigt wurden, konnten die Algorithmen zur Dimensionierung einer 57 Zellen umfassenden Bibliothek verwendet werden. Mit diesen werden Schaltungen synthetisiert, die mit einer Versorgungsspannung unterhalb der Transistoren-Schwellspannung (Subschwellbereich) ressourceneffizient funktionieren.

Außerdem wurde ein evolutionärer Algorithmus (EA) vorgestellt, mit dessen Hilfe sich schnell erste Erkenntnisse über die Lage einer Pareto-Menge im Suchraum sammeln lassen. Alle bestimmten Schaltungen sind optimal bezüglich derselben Zielfunktionen wie oben. Bei

Vergleichen mit kommerziellen Standardzellen, die aber nur für volle Versorgungsspannung ausgelegt sind, werden große Vorteile, vor allem in den Störabständen, offensichtlich.

Eine SRAM-Zelle wurde dimensioniert, so dass sie sowohl bei voller Versorgungsspannung als auch im Subschwellbereich ressourceneffizient betrieben werden kann. Es handelt sich dabei um eine 9-Transistor-Zelle, die durch zusätzliche Auslese-Transistoren aus der klassischen 6-Transistor-Zelle entsteht, um einem möglichen Informationsverlust beim Auslesen entgegenzuwirken. Vergleiche mit ähnlichen Speichern haben gezeigt, dass weniger Leckströhme bei ähnlichen Verzögerungszeiten fließen.

Bei der Untersuchung eines Differenzverstärkers mit Eintaktausgang wurde festgestellt, dass analoge Schaltungen in ihren Schalteigenschaften sehr sensibel auf Parameterschwankungen reagieren und daher bei der Entwurfsraumexploration auch die Robustheit der Entwurfspunkte bewertet werden muss. Die Algorithmen sollten deshalb eine zusätzliche Sensitivitätsanalyse durchführen. Erste Erweiterungen der GAIO-Algorithmen zum Bestimmen robuster Pareto-Punkte sind am Paderborner Lehrstuhl für Angewandte Mathematik erfolgt.

Ein Vergleich der mengenorientierten mit den evolutionären Algorithmen für die Anwendung der Schaltungsoptimierung hat gezeigt, dass die Ergebnisse der GAIO-Algorithmen bei vergleichbarer Laufzeit genauer und übersichtlicher sind. Der Einsatz evolutionärer Algorithmen ist vorteilhaft gegenüber der simplen Exhaustionsmethode.

Mit der Entwicklung eines Verfahrens zum Entwurf ressourceneffizienter Schaltungen, das basierend auf dem Mehrzieloptimierungsansatz Bausteine integrierter Schaltungen wie CMOS-Standardzellen, Speicherelemente und einfache analoge Schaltungen optimiert, wurde die Zielsetzung dieser Arbeit erfüllt. Desweiteren konnten der Ansatz und die Algorithmen erfolgreich eingesetzt werden, um eine Standardzellbibliothek für den Subschwellbereich zu entwerfen. Auch die Dimensionierung einer SRAM-Zelle, die ressourceneffizient bei 300 mV funktioniert, unterstützt den Entwurf eines 32bit-RISC-Prozessors.

Anhang

Transistormodell im Subschwellbereich

Das EKV-Modell des MOS-Transistors wird in [74] hergeleitet und ausführlich kommentiert. Für die Stromgleichung gilt

$$I_{\mathrm{D}} = I_{\mathrm{spec}} \exp \frac{V_{\mathrm{G}} - V_{\mathrm{T0}}}{nU_{\mathrm{T}}} \left(\exp \frac{-V_{\mathrm{S}}}{U_{\mathrm{T}}} - \exp \frac{-V_{\mathrm{D}}}{U_{\mathrm{T}}} \right) \tag{1}$$

mit dem Transistor-spezifischen Strom

$$I_{\mathrm{spec}} = 2n\mu C_{\mathrm{ox}} \frac{W}{L} U_{\mathrm{T}}^2$$

Die Spannungen sind in [74] als Differenz zur Substratspannung angegeben. V_{T0} bezeichnet die Schwellspannung des Transistors während U_{T} für die Temperaturspannung steht. Der letzte Faktor von (1) kann für Darin-Source-Spannungen nahe unterhalb der Schwellspannung vernachlässigt werden. Mit der Abschätzungsformel zur Verzögerungszeit durch

$$t_{\mathrm{pd}} = \frac{KCU_{\mathrm{dd}}}{I}$$

ergibt sich in diesem Betriebsbereich die Verzögerungszeit, wie in Kapitel 3

$$t_{\mathrm{pd}} = \frac{KC_{\mathrm{g}}U_{\mathrm{dd}}}{I_{\mathrm{o}} \exp\left(\frac{U_{\mathrm{dd}} - U_{\mathrm{TH}}}{nU_{\mathrm{T}}}\right)}.$$

Die Bezeichner in [74] variieren leider von Kapitel zu Kapitel. I_{spec} und I_{o} bezeichnen denselben Strom und C_{g} und C_{ox} dieselbe Kapazität.

Modell zur Standardabweichung des $\mathrm{SNM_{hold}}$ der 9-Transistor-SRAM-Zelle

Zur Herleitung der Formel (4.13) für die Standardabweichung

$$\sigma_{\mathrm{SNMhold}}\left(W_1, W_3\right) = \sqrt{\frac{{\sigma_\mathrm{n}}^2}{W_1/W_{0,n}} + \frac{{\sigma_\mathrm{p}}^2}{W_3/W_{0,p}}},$$

wird davon ausgegangen, dass der statische Störabstand affin-linear von den Schwellspannungen der Transitoren abhängt. Es gelte

$$\mathrm{SNM_{hold}} = c_\mathrm{p} V_{\mathrm{th,p}} + c_\mathrm{n} V_{\mathrm{th,n}} + c \tag{2}$$

mit Konstanten c, c_p und c_n. Die Schwellspannungen unterliegen Störeinflüssen und werden daher als normalverteilte Zufallsvariablen betrachtet. Mit der Aufteilung

$$\begin{aligned} V_{\mathrm{th,n}} &= E[V_{\mathrm{th,n}}] + \Delta V_{\mathrm{th,n}} \\ V_{\mathrm{th,p}} &= E[V_{\mathrm{th,p}}] + \Delta V_{\mathrm{th,p}} \end{aligned}$$

lässt sich (2) zu

$$\mathrm{SNM_{hold}} = c_\mathrm{p} \Delta V_{\mathrm{th,p}} + c_\mathrm{n} \Delta V_{\mathrm{th,n}} + E[\mathrm{SNM_{hold}}] \tag{3}$$

umformen.

Die Standardabweichung der Schwellspannung ist nach Pelgrom [68] durch

$$\sigma(V_\mathrm{th}) = \frac{\sqrt[4]{4q^3 \varepsilon_\mathrm{Si} \phi_B}}{2} \frac{t_\mathrm{ox}}{\varepsilon_\mathrm{ox}} \frac{\sqrt[4]{N}}{\sqrt{W_\mathrm{eff} L_\mathrm{eff}}} = \frac{k}{\sqrt{W_\mathrm{eff} L_\mathrm{eff}}}$$

gegeben. k ist dabei eine Konstante, die vom Transistortyp abhängt.

Es wird vorausgesetzt, dass die Zufallsvariablen $\Delta V_{\mathrm{th,p}}$ und $\Delta V_{\mathrm{th,n}}$ in (3) statistisch unabhängig voneinander sind und beide als Erwartungswert Null haben. Die Standardabweichung für $\mathrm{SNM_{hold}}$ ergibt sich damit aus der gewichteten Summe der Standardabweichungen von $\Delta V_{\mathrm{th,p}}$ und $\Delta V_{\mathrm{th,n}}$:

$$\sigma^2_{\mathrm{SNMhold}} = c_\mathrm{p}^2 (\sigma(V_{\mathrm{th,p}}))^2 + c_\mathrm{n}^2 (\sigma(V_{\mathrm{th,n}}))^2 = c_\mathrm{p}^2 k_\mathrm{p}^2 / \left(W_\mathrm{p} L_\mathrm{p}\right) + c_\mathrm{n}^2 k_\mathrm{n}^2 / \left(W_\mathrm{n} L_\mathrm{n}\right) \tag{4}$$

Werte für die Varianzen beider Transistoren wurden angesetzt durch

$$\begin{aligned} \sigma_\mathrm{n}^2 &= \frac{R_\mathrm{n}^2 k_\mathrm{n}^2}{W_{0,n} L_{0,n}} \\ \sigma_\mathrm{p}^2 &= \frac{R_\mathrm{p}^2 k_\mathrm{p}^2}{W_{0,p} L_{0,p}}. \end{aligned} \tag{5}$$

Die Konstanten in (5) wurden durch Monte-Carlo-Simulationen bestimmt. Mit einem relativen Fehler von maximal 6 % ergaben sich $\sigma_\mathrm{n} \approx 9.9\,\mathrm{mV}$ und $\sigma_\mathrm{p} \approx 5.2\,\mathrm{mV}$. Nach Einsetzen von (5) und Umformen von (4) ergibt sich

$$\sigma_{\mathrm{SNMhold}} = \sqrt{\frac{{\sigma_\mathrm{n}}^2}{W_1 L_1 / \left(W_{0,n} L_{0,n}\right)} + \frac{{\sigma_\mathrm{p}}^2}{W_3 L_3 / \left(W_{0,p} L_{0,p}\right)}} \tag{6}$$

Da die Längen $L_1 = L_{0,\mathrm{n}}$ und $L_3 = L_{0,\mathrm{p}}$ konstant und gleich sind, ergibt sich (4.13) aus (6).

Literaturverzeichnis

[1] H.L. Abdel-Malek and A.K.S.O. Hassan. The ellipsoidal technique for design centering and region approximation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(8):1006–1014, 1991.

[2] R.J. Baker. *CMOS: circuit design, layout, and simulation*. Wiley-IEEE Press, 2007.

[3] H. K O. Berge, M. Blesken, S. Aunet, and U. Rückert. Design of 9T SRAM for dynamic voltage supplies by a multiobjective optimization approach. *Electronics, Circuits, and Systems (ICECS), 17th IEEE International Conference on*, pages 319 – 322, 2010.

[4] A. Bhavnagarwala, X. Tang, and J.D. Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE Journal of Solid State Circuits*, 36:658–665, 2001.

[5] M. Blesken, A. Chebil, U. Rückert, X. Esquivel, and O. Schütze. Integrated Circuit Optimization by Means of Evolutionary Multi-Objective Optimization. *Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 807–812, 2011.

[6] M. Blesken, S. Lütkemeier, and U. Rückert. Multiobjective optimization for transistor sizing sub-threshold CMOS logic standard cells. *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1480–1483, 2010.

[7] M. Blesken, U. Rückert, D. Steenken, K. Witting, and M. Dellnitz. Multiobjective Optimization for Transistor Sizing of CMOS Logic Standard Cells Using Set-Oriented Numerical Techniques. In *Proceedings of 27th Norchip Conference*, pages 1 – 4, 2009.

[8] M. Borah, R.M. Owens, and M.J. Irwin. Transistor sizing for low power CMOS circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(6):665–671, 1996.

[9] R.K. Brayton, G.D. Hachtel, and A.L. Sangiovanni-Vincentelli. A survey of optimization techniques for integrated-circuit design. *Proceedings of the IEEE*, 69(10):1334–1362, 1981.

[10] B.H. Calhoun and A. Chandrakasan. A 256-kb 65-nm sub-threshold SRAM design for ultra-low-voltage operation. *IEEE journal of solid-state circuits*, 42:680–688, 2007.

[11] I.J. Chang, J.J. Kim, S.P. Park, and K. Roy. A 32 kb 10T Sub-Threshold SRAM Array With Bit-Interleaving and Differential Read Scheme in 90 nm CMOS. *Solid-State Circuits, IEEE Journal of*, 44(2):650 –658, feb. 2009.

[12] A. Chebil. Optimization of Sub-Threshold Digital Cells by Evolutionary Algorithms. Diplomarbeit, Université de Monastir, 2009.

[13] C.A.C. Coello, S. de Computación, and C.S.P. Zacatenco. Twenty years of evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, 2006.

[14] J.A. Croon, S. Decoutere, W. Sansen, and H.E. Maes. Physical modeling and prediction of the matching properties of MOSFETs. In *Solid-State Device Research conference, 2004. ESSDERC 2004. Proceeding of the 34th European*, pages 193–196, Sept. 2004.

[15] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 2:115–148, 1995.

[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*, pages 849–858. Springer, 2000.

[17] K. Deb and M. Goyal. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

[18] M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of optimization theory and applications*, 124(1):113–136, 2005.

[19] M. Dellnitz and K. Witting. Computation of robust Pareto points. *International Journal of Computing Science and Mathematics*, 2(3):243–266, 2009.

[20] J. Dienstuhl. *Optimierung und Modellierung von sub-100 nm CMOS-Speicherelementen mit Methoden der Computational Intelligence*. Theophano, 2006.

[21] J. Dienstuhl and K. Goser. Methoden der Computational Intelligence fur die Optimierung und Charakterisierung von CMOS Standardzellen am Beispiel eines Flip-Flop Speicherelementes. *VDE/VDI Fachbericht der 7. GMM/ITG-Diskussionssitzung ANALOG*, pages 141–146, 2003.

[22] H.-J. Eikerling. *Optimierung digitaler Schaltungen durch Partitionierung und Resynthese*. Shaker, 1997.

[23] C. Fisher, R. Blankenship, J. Jensen, T. Rossman, and K. Svilich. Optimization of standard cell libraries for low power, high speed, or minimal area designs. *Proceedings of the IEEE 1996*, pages 493–496, 1996.

[24] C.M. Fonseca. *Multiobjective genetic algorithms with application to control engineering problems*. Citeseer, 1995.

[25] M.P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 141–153. L. Erlbaum Associates Inc., 1985.

[26] D.D. Gajski and R.H. Kuhn. Introduction: New VLSI Tools. *IEEE Computer*, 12:11–14, 1983.

[27] G.E. Georges, G.G.E. Gielen, and R.A. Rutenbar. Computer-Aided Design Of Analog And Mixed-Signal Integrated Circuits. *Proc. of The IEEE*, 88(12):1823–1852, 2000.

[28] G. Gigerenzer and P.M. Todd. the ABC research group. *Simple heuristics that make us smart*, 15, 1999.

[29] D.E. Goldberg. *Genetic Algorithms in Search and Optimization, and Machine Learning*. Addison-Wesley, 1989.

[30] D. Grabowski, M. Olbrich, and E. Barke. AC-Analyse analoger Schaltungen mit affiner Arithmetik. *GMM-Fachbericht-ANALOG'08*, 2008.

[31] H.E. Graeb. *Analog design centering and sizing*. Springer, 2007.

[32] P. Hajela and C.Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural and Multidisciplinary Optimization*, 4(2):99–107, 1992.

[33] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw. Performance and variability optimization strategies in a sub-200mv, 3.5pj/inst, 11nw subthreshold processor. In *VLSI Circuits, 2007 IEEE Symposium on*, pages 152–153, June 2007.

[34] C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.

[35] K. Hoffmann. *Systemintegration: Vom Transistor zur großintegrierten Schaltung*. Oldenbourg Wissenschaftsverlag, 2006.

[36] B. Hoppe, G. Neuendorf, D. Schmitt-Landsiedel, and W. Specks. Optimization of high-speed CMOS logic circuits with analytical models for signal delay, chip area, and dynamic power dissipation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(3):236–247, 1990.

[37] J. Horn, N. Nafpliotis, and D.E. Goldberg. Multiobjective optimization using the niched pareto genetic algorithm. *Urbana*, 51:61801–2996, 1993.

[38] J. Horn, N. Nafpliotis, and D.E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87. Citeseer, 1994.

[39] T.H. Kim, J. Keane, H. Eom, and C.H. Kim. Utilizing reverse short-channel effect for optimal subthreshold circuit design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(7):821–829, 2007.

[40] T.H. Kim, J. Liu, J. Keane, and C.H. Kim. A 0.2 v, 480 kb subthreshold sram with 1 k cells per bitline for ultra-low-voltage computing. *Solid-State Circuits, IEEE Journal of*, 43(2):518 –529, feb. 2008.

[41] H. Klar. *Integrierte digitale Schaltungen MOS/BICMOS*. Springer Verlag, 1996.

[42] J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *Proc. of 2002 Congress on Evolutionary Computation*, volume 711, page 716, 2002.

[43] J. Knowles, D. Corne, and M. Oates. On the assessment of multiobjective approaches to the adaptive distributed database management problem. In *Parallel Problem Solving from Nature PPSN VI*, pages 869–878. Springer, 2000.

[44] J.P. Kulkarni, K. Kim, and K. Roy. A 160 mV Robust Schmitt Trigger Based Subthreshold SRAM. *IEEE Journal of Solid-State Circuits*, 42:2303–2313, 2007.

[45] J. Kwong, Y.K. Ramadass, N. Verma, and A. Chandrakasan. A 65 nm Sub-Vt Microcontroller With Integrated SRAM and Switched Capacitor DC-DC Converter. *IEEE Journal of Solid-State Circuits*, 44(1):115–126, January 2009.

[46] G.B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. 2000.

[47] S. Lin, M. Marek-Sadowska, and E.S. Kuh. Delay and area optimization in standard-cell design. pages 349–352, 1991.

[48] Z. Liu and V. Kursun. Characterization of a novel nine-transistor sram cell. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(4):488 –492, april 2008.

[49] K.K. Low and S.W. Director. A New Methodology for the Design Centering of 1C Fabrication Processes. *IEEE Transactions on Computer-Aided Design*, 10(7):895, 1991.

[50] S. Lütkemeier. *Ressourceneffziente Digitalschaltungen für den Subschwell-Betrieb*. Universität Paderborn, Deutschland, 2012.

[51] S. Lütkemeier, T. Kaulmann, and U. Rückert. A Sub-200mV 32bit ALU with 0.45pJ/instruction in 90nm CMOS. *Proc. Semiconductor Conf. Dresden*, April 2009.

[52] M. Miura-Mattausch, M. Suetake, H.J. Mattausch, S. Kumashiro, N. Shigyo, S. Odanaka, and N. Nakayama. Physical modeling of the reverse-short-channel effect for circuitsimulation. *IEEE Transactions on Electron Devices*, 48(10):2449–2452, 2001.

[53] S. Mostaghim. *Multi-Objective Evolutionary Algorithms: Data Structures, Convergence, and Diversity*. Shaker, 2005.

[54] D. Mueller, H. Graeb, and U. Schlichtmann. Trade-off design of analog circuits using goal attainment and Wave Front sequential quadratic programming. pages 75–80, 2007.

[55] Y. Nakagome, M. Horiguchi, T. Kawahara, and K. Itoh. Review and future prospects of low-voltage RAM circuits. *IBM Journal of Research and Development*, 47(5/6):525–552, 2003.

[56] A. Ono, R. Ueno, and I. Sakai. TED control technology for suppression of reverse narrow channeleffect in 0.1 μm MOS devices. In *Electron Devices Meeting, 1997. IEDM'97. Technical Digest., International*, pages 227–230, 1997.

[57] C. Piguet, J.M. Masgonty, S. Cserveny, C. Arm, and P.D. Pfister. Low-power low-voltage library cells and memories. In *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on*, volume 3, 2001.

[58] B. Razavi. *Design of analog CMOS integrated circuits*. McGraw-Hill, Inc. New York, NY, USA, 2000.

[59] R.A. Rutenbar, G.G.E. Gielen, and J. Roychowdhury. Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs. *Proceedings of the IEEE*, 95(3):640–669, 2007.

[60] O. Schütze. A new data structure for the nondominance problem in multi-objective optimization. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization (EMO 03)*, volume 2 of *Springer*, pages 509–518, 2003.

[61] O. Schütze. *Set Oriented Methods for Global Optimization*. University of Paderborn, Germany, 2004.

[62] O. Schütze, A. Dell'Aere, and M. Dellnitz. On Continuation Methods for the Numerical Treatment of Multi-Objective Optimization Problems. In J. Branke, K. Deb, K. Miettinen, and Steuer, R. E. (Eds.), editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, 2005. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany.

[63] O. Schütze, M. Laumanns, C.A. Coello Coello, M. Dellnitz, and E. Talbi. Convergence of stochastic search algorithms to finite size Pareto set approximations. *Journal of Global Optimization*, 41(4):559–577, 2008.

[64] E. Seevinck, FJ List, and J. Lohstroh. Static-noise margin analysis of MOS SRAM cells. *IEEE Journal of Solid State Circuits*, 22(5):748–754, 1987.

[65] M.E. Sinangil, N. Verma, and A. Chandrakasan. A reconfigurable 8t ultra-dynamic voltage scalable (u-dvs) sram in 65 nm cmos. *Solid-State Circuits, IEEE Journal of*, 44(11):3163 –3173, nov. 2009.

[66] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[67] D. Steenken. Multiobjective Optimization and Sensitivity Evaluation of IC base components using set-oriented methods. Diplomarbeit am Lehrstuhl für Angewandte Mathematik, Universität Paderborn., 2009.

[68] P.A. Stolk, F.P. Widdershoven, and D.B.M. Klaassen. Modeling statistical dopant fluctuations in MOS transistors. *Electron Devices, IEEE Transactions on*, 45(9):1960–1971, 2002.

[69] R.M. Swanson and J.D. Meindl. Ion-implanted complementary MOS transistors in low-voltage circuits. *IEEE Journal of Solid-State Circuits*, 7(2):146–153, 1972.

[70] K.C. Tan, T.H. Lee, and E.F. Khor. Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. *Artificial intelligence review*, 17(4):251–290, 2002.

[71] M. Thomas. Design und Analyse integrierter Schaltungen mit evolutionären Algorithmen. 2001.

[72] R. Troutman. Subthreshold design considerations for insulated-gate field-effect transistors. In *1973 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, volume 16, 1973.

[73] D.A. Van Veldhuizen. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. 1999.

[74] A. Wang, B.H. Calhoun, and A. Chandrakasan. *Sub-threshold Design for Ultra Low-Power Systems (Series on Integrated Circuits and Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[75] A. Wang and A. Chandrakasan. A 180-mV subthreshold FFT processor using a minimum energy design methodology. *IEEE Journal of Solid-State Circuits*, 40(1):310–319, 2005.

[76] J.M. Wojciechowski and J. Vlach. Ellipsoidal method for design centering and yield estimation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(10):1570–1579, 1993.

[77] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner. Theoretical and practical limits of dynamic voltage scaling. *Proc. IEEE / ACM Design Automation Conf.*, pages 868–873, 2004.

[78] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester. A Variation-Tolerant Sub-200 mV 6-T Subthreshold SRAM. *IEEE Journal of Solid-State Circuits*, 43:2338–2348, 2008.

[79] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. TIK-Schriftenreihe Nr.30, 1999.

[80] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[81] E. Zitzler, M. Laumanns, L. Thiele, et al. SPEA2: Improving the strength Pareto evolutionary algorithm. In *EUROGEN*, pages 95–100. Citeseer, 2001.

[82] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[83] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.