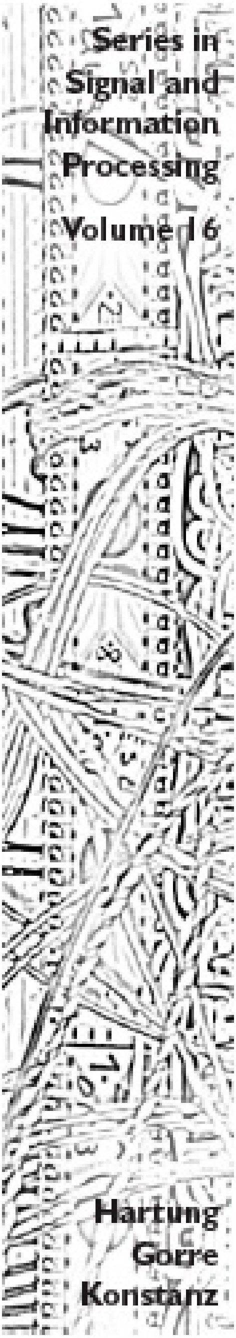


Series in
Signal and
Information
Processing
Volume 16

Diss. ETH No. 16536

On Analog Decoders and Digitally Corrected Converters

A dissertation submitted to the
Swiss Federal Institute of Technology, Zürich
for the degree of
Doctor of Technical Sciences

presented by

Matthias U. Frey

dipl. El.-Ing. ETH
born on December 31, 1975
citizen of Laufen-Stadt (BL), Switzerland

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Prof. Dr. Vincent Gaudet, co-examiner
Prof. Dr. Nikolai Nefedov, co-examiner

April 2006

Hartung
Gorre
Konstanz

Diss. ETH No. 16536

On Analog Decoders and Digitally Corrected Converters

A dissertation submitted to the
Swiss Federal Institute of Technology, Zürich
for the degree of
Doctor of Technical Sciences

presented by

Matthias U. Frey

dipl. El.-Ing. ETH
born on December 31, 1975
citizen of Laufen-Stadt (BL), Switzerland

accepted on the recommendation of
Prof. Dr. Hans-Andrea Loeliger, examiner
Prof. Dr. Vincent Gaudet, co-examiner
Prof. Dr. Nikolai Nefedov, co-examiner

Hartung-Gorre Verlag, Konstanz, April 2006

Series in Signal and Information Processing **Vol. 16**

Editor: Hans-Andrea Loeliger

Bibliographic Information published by Die Deutsche Bibliothek

Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the internet at http://dnb.ddb.de.

First Edition 2006

HARTUNG-GORRE VERLAG KONSTANZ

ISSN 1616-671X
ISBN 3-86628-074-2

There's a light at the end of the tunnel,
it's probably a burning car.

Phil Duke/Osmosis

Acknowledgments

First of all I would like to thank my advisor Prof. Dr. Hans-Andrea Loeliger for giving me the opportunity to conduct research at the Signal and Information Processing Laboratory (ISI). He was always available and interested in discussions and greatly shaped this thesis.

I profited from many enriching discussions with Prof. Dr. Nikolai Nefedov, before and after he agreed to be my co-advisor. Thank you for sharing your broad knowledge! Many thanks also to Prof. Dr. Vincent Gaudet, my second co-advisor. His valuable comments led to a substantial improvement of my thesis.

I am greatly indebted to three former ISI-members. Through his enthusiasm during discussions, Jossy Sayir sparked my interest in FFTs, coding theory, and other ISI-related topics. Markus Helfenstein and Felix Lustenberger were highly inspirational as predecessors and very obliging before and during my years at ISI.

Without the help of many people, this thesis would not have been possible: I would like to thank Patrick Merkli—not only for sharing the office with me for nearly three years and hardly ever complaining about the noisy music, but also for the enjoyable collaboration and the joint presentations and chip-designs, and for all the vitamin-drinks before tape-out! Patrik Strebel and Thomas Schärer greatly contributed to this thesis by constructing all the measurement devices and the discrete hardware. Daniel Furrer trusted me to supervise his Master's thesis and did a great job in designing a "big analog decoder." Christine Fleischmann was kind enough to search for spelling mistakes and grammatical flaws in my thesis.

For funding of the chip area, I would like to thank IBM. I am also grateful to Dr. Martin Schmatz, Zurich Research Laboratory, IBM Research, Zurich, Switzerland for his support, and to Mr. Martin Lanz, Laboratory for Electromagnetic Fields and Microwave Electronics, for bonding and packaging our chips in no time at all.

Many people at ISI added to the congenial atmosphere: I am very thankful to "Oberassistent" Stefan Moser who gave the sometimes chaotic environs some structure. He and Daniel Hösli formed a perfect "helpdesk" to answer all my computer- and LaTeX-related questions. I very much enjoyed the lunch-break discussions on "Japan or China" with Qun Gao, on "religion and probabilities" with Sascha Korl and Markus Hofbauer, on "analog vs. digital photography" with Maja Ostojic, on "music and integrals" with Natalia Miliou, on "sports and apples" with Stephan Tinguely and Tobias Koch, on "judo vs. hand-ball" with Michèle Wigger, on "fit or power" with Jonas Biveroni, and on "chaos and curry" with Murthy Devarakonda.

There is nothing like joining a judo training after a long day in front of the computer: a deep bow to my judo friends and training partners in Ebikon, Uster, Bern, Magglingen, Köln and Marseille. A cheery wave goes to my "Japanese friends", Hélène, Kei, Manuel, Pavel, Roberto, Satomi, and Susanti; they're scattered all over the world, but still close. A very special "thank you!" to Bettina Steinauer for her telephone-support on non-technical matters.

Last but not least, I would like to express my warmest gratitude to my parents, to my sisters, and to Reto and Philipp for the constant support and interest in all my studies, and for reminding me not to work too hard.

Abstract

In recent years, the demand for efficient and reliable communication networks has greatly increased. To satisfy this need, powerful error correcting codes were introduced. The (iterative) algorithms used for decoding such modern codes are computationally very demanding and need great computing power to deliver real-time results. Mobile users, however, ask for low-power electronics; the combination of both demands led to an increased interest in analog communication circuits, e.g., in analog decoders for error correcting codes.

The first part of this thesis discusses various implementations of analog decoders. An analog decoder can be understood as a code-representing (factor) graph mapped on analog silicon, whereas the decoding algorithm (e.g., the sum-product algorithm) corresponds to the settling behavior of the analog circuit. The performance gain of analog decoders compared to digital implementations in terms of speed or power-consumption is believed to be at least a factor of 100.

The following implementations of such analog decoders are discussed: Hamming decoders built out of two generations of discrete softgates, an integrated Hamming decoder and an integrated Reed-Muller decoder are presented. An extensive collection of measured error-rate curves of all decoders under various operating conditions prove their full functionality and demonstrate their behavior under transistor mismatch.

Furthermore, a novel circuit to compute the soft symbols for a PAM or QAM signal is presented. This simple transistor network blends in nicely with analog decoders—its outputs are currents proportional to the symbol-likelihoods.

The second part of this thesis is devoted to analog-to-digital and digital-to-analog converters with minimal-sized elements. Digital data processing is pervasive, and the need for fast, high-resolution and low-power converters persistent. Highly accurate converters usually require large elements to achieve the desired minimal mismatch; large elements however demand high currents for high speed. This trade-off can be circumvented by using small-sized, yet imprecise, elements and then adding digital post-correction circuitry.

It can be shown, that the effective resolution of a digitally-corrected analog-to-digital converter only weakly depends on the comparator mismatch. Therefore, such converters can be built with minimal-sized elements and a digital post-correction. This was confirmed by measurements on an integrated flash analog-to-digital converter containing 256 low-precision comparators and achieving an effective resolution of nearly 7 bits.

A similar statement holds for current-steering digital-to-analog converters with almost minimal-sized current sources: for a converter containing 12 low-precision current sources and digital pre-correction, an effective resolution of more than 10 bits was achieved—virtually irrespective of the mismatch.

The thesis is concluded by a summary of the obtained results and a collection of research proposals in the field of analog decoding and for digitally-corrected converters.

Keywords: Factor graphs, message-passing algorithms, sum-product algorithm, analog non-linear transistor circuits, translinear circuits, analog iterative decoders, soft symbol detection, digital data transmission, analog-to-digital converter, digital-to-analog converter, calibration, post-correction, imprecise elements.

Kurzfassung

In den letzten Jahren ist die Nachfrage nach effizienten und zuverlässigen Kommunikationsnetzwerken stark gestiegen. Leistungsfähige Fehler korrigierende Codes wurden eingeführt, um diesem Bedarf nachzukommen. Die (iterativen) Algorithmen, welche zur Decodierung dieser modernen Codes verwendet werden, sind äusserst rechenintensiv. Eine hohe Rechenleistung wird benötigt, um die decodierten Daten in Echtzeit bereitstellen zu können. Die meist mobilen Anwender hingegen fordern leistungsverbrauchsarme Elektronik. Die Kombination beider Anforderungen - hohe Rechenleistung bei kleinem Leistungsverbrauch - führte zu einem vermehrten Interesse an analoger Kommunikationselektronik; auch das Interesse an analoger Decodierung von Fehler korrigierenden Codes ist gestiegen.

Der erste Teil dieser Dissertation bespricht verschiedene Implementierungen analoger Decoder. Ein analoger Decoder kann als Übersetzung eines code-darstellenden (Faktor-)Graphen in analoge Elektronik verstanden werden. Der auf dem (Faktor-)Graph ausgeführte Dekodier-Algorithmus (zum Beispiel der Summe-Produkt-Algorithmus) entspricht dann dem Einschwingvorgang in der analogen Schaltung. Es wird angenommen, dass die Rechenleistungssteigerung bei konstantem Leistungsverbrauch (oder die Verminderung des Leistungsverbauchs bei konstanter Rechenleistung) von analogen Decodern gegenüber einem digitalen Decoder etwa einem Faktor von 100 entspricht.

Die folgenden Implementierungen solch analoger Decoder werden besprochen: Hamming Decoder, welche diskret aus zwei Generationen von “softgates”[1] aufgebaut sind, ein integrierter Hamming Decoder, sowie ein

---

[1] “Softgates” sind bidirektionale Gatter, welche mit Wahrscheinlichkeiten statt mit

integrierter Reed-Muller Decoder werden vorgestellt. Eine umfangreiche Sammlung von Fehlerwahrscheinlichkeitskurven aller Decoder, gemessen unter verschiedenen Betriebsbedingungen, belegen die Funktionalität aller Schaltungen. Die Kurven zeigen auch das Verhalten der Decoder gegenüber nicht-idealen Transistoren auf.

Eine neuartige Schaltung zur Berechnung der Wahrscheinlichkeiten der Symbole bei der PAM- (pulse amplitude modulation) oder QAM- (quadrature amplitude modulation) Übertragung wurde eingeführt. Diese einfache Transistor-Schaltung verträgt sich gut mit den bestehenden Schaltungen der analogen Decodierung: Die Ausgänge der Schaltung sind Ströme, welche proportional zu den gesuchten "Symbol-Wahrscheinlichkeiten" sind.

Der zweite Teil dieser Arbeit konzentriert sich auf Analog/Digital- und Digital/Analog-Wandler mit unpräzisen Elementen. Die digitale Datenverarbeitung ist omnipräsent und der Bedarf an schnellen, leistungsarmen und hochauflösenden Analog/Digital- und Digital/Analog-Wandlern ist anhaltend hoch. Sehr präzise Wandler benötigen üblicherweise grosse Bauteile, um die geforderten kleinen Toleranzen zu erzielen; grosse Bauteile bedingen aber auch grosse Ströme, um hohe Geschwindigkeiten zu erreichen. Dieser Konflikt kann durch das Verwenden von kleinen, unpräzisen Bauteilen und der Einführung einer digitalen Korrektur umgangen werden.

Es kann gezeigt werden, dass die effektive Auflösung eines digital-korrigierten Analog/Digital-Wandlers nur leicht von der Grösse der Ungenauigkeit der Komparatoren abhängt. Deshalb können solche Wandler mit ungenauen Bauteilen minimaler Grösse und anschliessender digitaler Korrektur gebaut werden. Diese Beobachtung konnte mit Messungen an einem integrierten Analog/Digital-Wandler bestätigt werden: Mit 256 ungenauen Komparatoren und einer digitalen Korrektur konnte eine effektive Auflösung von fast 7 bit erreicht werden.

Eine ähnliche Beobachtung wurde für Digital/Analog-Wandler mit kleinsten (und ungenauen) gesteuerten Stromquellen gemacht: Ein Wandler mit 12 kleinen (unpräzisen) Stromquellen und digitaler Korrektur erreichte eine effektive Auflösung von mehr als 10 bit und dies fast unabhängig von der Genauigkeit der Stromquellen.

---

binären Werten rechnen.

Diese Arbeit schliesst mit einer Zusammenfassung aller erzielten Resultate sowie mit einer Sammlung von Forschungsanregungen in den Gebieten der analogen Decodierung und der digital-korrigierten Wandler.

Stichworte: Faktorgraph, Message-Passing Algorithmus, Summe-Produkt-Algorithmus, analoge nicht-lineare Transistor-Schaltung, translineare Schaltung, analoge iterative Decoder, “soft symbol” Detektion, digitale Datenübertragung, Analog/Digital-Wandler, Digital/Analog-Wandler, Kalibrierung, Post-Korrektur, ungenaue Elemente.

Contents

Abstract **vii**

Kurzfassung **ix**

1 Introduction **1**
1.1 Motivation 1
1.2 Contributions 3
1.3 Outline 4

2 Analog Decoder Examples **7**
2.1 Introductory Remarks 7
2.1.1 Factor Graphs and Analog Decoders 8
2.1.2 The Channel Model 10
2.1.3 Transistor Circuit Notation 11
2.1.4 Chapter Overview and Some Definitions 13
2.2 Softgates, Building Blocks of Analog Decoders 15
2.2.1 First-Generation Softgates 15
2.2.2 Second-Generation Softgates 30
2.3 The (8,4,4) Extended Hamming Decoder Chip 45
2.3.1 Measurements 47
2.3.2 Conclusions 55
2.4 (16,5,8) Reed-Muller Decoder Chip 59
2.4.1 The Serial-to-Parallel Input Interface 60
2.4.2 Error Rate Measurements 65
2.4.3 Conclusions 68
2.5 An Analog Coprocessor for Decoding a Turbo Code . . . 69
2.5.1 Turbo Encoder and Sliding Window Decoder . . . 71
2.5.2 Description of the Circuit 74
2.5.3 Preliminary Conclusions 77

2.6 Concluding Remarks 79

3 Soft Symbol Detection 83
3.1 Introductory Remarks 83
3.2 PAM, QAM and Probabilities 84
3.3 PAM Soft Symbol Detector Circuit 87
3.3.1 Example: Soft Symbol Detection Circuit for 4-PAM 91
3.4 QAM Soft Symbol Detector Circuit 95
3.4.1 Example: Soft Symbol Detection Circuit for 16-QAM 96
3.4.2 Example: Soft Symbol Detection Circuit for 12-QAM 98
3.5 Applications for Trellis-Coded Modulation and Iterative Receivers 98
3.5.1 From Symbol Metrics to Label Metrics 98
3.5.2 Iterative SISO Modules 100
3.6 Generalized Scaling and Damping 103
3.7 Conclusions and Outlook 104

4 A/D-Conversion with Imprecise Elements 107
4.1 Introductory Remarks 107
4.2 Introduction to A/D-Conversion 109
4.3 The Principle 112
4.4 Design Example 115
4.5 Measurements 117
4.5.1 The Comparators 117
4.5.2 The Complete ADC 118
4.6 Conclusions and Outlook 123

5 D/A-Conversion with Imprecise Elements 125
5.1 Introductory Remarks 125
5.2 Introduction to D/A-Conversion 126
5.3 Introduction to Current-Steering Digital-to-Analog Converters 130
5.4 Simulations of Digital-to-Analog Converters with Imprecise Current Sources 132
5.4.1 Variable Input Interval 132
5.4.2 Binary-Scaled Current Sources with Mismatch . . 133
5.4.3 Unit Current Sources with Mismatch 135
5.4.4 Geometrically Scaled Current Sources with Mismatch 138
5.4.5 Arithmetically Scaled Current Sources with Mismatch 140

5.5 ΔΣ-Converters with Corrected ADCs and DACs 143
5.5.1 ΔΣ-Converter with a non-ideal ADC 149
5.5.2 ΔΣ-Converters with a non-ideal ADC and a non-ideal DAC 152
5.6 Conclusions, Outlook 153

6 Conclusion and Outlook 155
6.1 Conclusions 155
6.2 Outlook 156

A Computing with Analog Transistors 159
A.1 Multiplication and Division Circuits for Three Input Variables 160
A.2 Squaring and Square Root Extracting Circuit 164

B Limits on the Effective Resolution of Flash ADCs 167

Bibliography 171

About the Author 179

Index 179

Chapter 1

Introduction

> "I don't know where this will lead us,
> but I think it will be a place both
> wonderful and strange."
> – SA Dale Cooper

1.1 Motivation

Long before digital processors were invented, scientists had the vision to compute with analog electronic circuits, to build analog systems that could make calculations with high precision. The input to analog computers are variables represented by physical quantities (e.g., voltages or currents); the solution to the problem is then given by the equilibrium state of the system after it has settled. An analog computer can also be understood as operating on an analogous system that can be solved more easily, but which is similar to the real system and its solution is valid for both: a physical system (that needs to be computed) is represented by an alternative system (an analogy of the system) that can easily be observed and measured.

In an analog electronic computer, all operations are performed in parallel, and data is represented by voltages, currents or charges—a very compact but not necessarily robust form of data. A single capacitor, for example, can represent one continuous variable. Storage for continuous variables can never be robust: a small perturbation can change the stored value to another possible value. (Therefore, storage is something inherently digital.)

With the beginning of the "digital era", these analog computational circuits were partly forgotten. The now pervasive digital computation is inherently different: in general, digital computers operate on data one step at a time, and the digital computer represents the data with an extensive, but very robust, representation called binary. A single capacitor in a digital (electronic) computer can only store one bit (capacitor charged or capacitor uncharged). Digital computers do not rely on the underlying physical system, and can be implemented on electronic circuits, optical devices or any other imaginable system with two states.

Digital computers show many advantages over analog computers (e.g., increased robustness, ease of data storage). However, for applications where very low power consumption or very high frequencies are required, analog circuits show advantages. In some cases, analog signal processing is much "cheaper" than digital signal processing; "cheap" in terms of power consumption and chip area.

In recent years, the demand for efficient and reliable communication networks has greatly increased. To satisfy this demand, powerful error correcting codes—such as turbo codes or low-density parity check codes—are applied to the digital data. Such modern codes are decoded by the sum-product algorithm or some variation thereof, iterative algorithms that operate by passing "messages" along the edges of a factor graph of the code. These algorithms are computationally very demanding and need great computational power to deliver real-time results.

The first part of this thesis deals with chip implementations of analog decoders. An analog decoder can be understood as an "analogy" of the code-representing factor graph. In such analog decoders, the iterations of the standard (discrete-time) decoding algorithm are replaced by the natural (continuous-time) settling behavior of the transistor network. Analog decoders seem to consume substantially less power (for a given speed) than their digital counterparts do. They are therefore suited for modern low-power applications, and for some applications (very high speed or very low power), they may mark the only feasible solution.

Error correcting codes used in modern communication systems support various standards with different block-sizes and require varying interleavers. The decoder needs to support all these standards, and, therefore, has to be programmable. Designing reprogrammable analog networks is very demanding. Instead, only some parts of the decoder can be implemented in analog electronics, the re-programmability can then be obtained with a digital processor. This mixed-mode approach requires low-power (and high-speed) analog-to-digital (ADC) and digital-to-analog (DAC) converters.

The second part of the thesis discusses digitally-corrected analog-to-digital and digital-to-analog converters. With the aid of digital post-correction, a high effective resolution can be achieved, even while using minimal-sized low precision elements.

1.2 Contributions

The main contributions of this thesis are as follows:

- In order to deepen the understanding of analog decoders, extensive measurement results of various decoders are presented. In a first step, “breadboard” analog decoders (built out of discrete soft-gate chips) for decoding an (8,4,4) extended Hamming code were measured. The robustness of analog decoders in general, and their behavior under significant transistor mismatch could be shown.

- Two analog decoders, one for an (8,4,4) extended Hamming code and one for a (16,5,8) Reed Muller code, were integrated in a 0.25 μm- and a 0.18 μm-CMOS process, respectively. Both ASICs were measured under various operating conditions and proved to be fully functional. Many measurements of these decoders corroborate the feasibility of implementing message-passing algorithms in analog electronic circuits but they also show that (small) analog decoders are not immune against transistor mismatch. Especially mismatch in the input interface needs to be minimized. A mixed-signal decoder chip, for decoding a turbo-code, was implemented, but has not been measured yet.

- A new basic transistor circuit, a “soft symbol detection” circuit, was introduced. This circuit is a simple analog network that fits in nicely with the aforementioned analog decoders. It computes (fundamentally exact) the likelihoods of a baseband PAM or QAM signal and can easily be extended to be used in the context of trellis coded modulation or with iterative receivers.

- Simple observations on digitally corrected flash analog-to-digital converters were made. Simulation and measurement results endorse the observation that, above a certain comparator mismatch, the effective resolution of an ADC is irrespective of the mismatch approximately 1 bit below the maximum effective resolution.

- Simulations of digitally corrected digital-to-analog converters with minimal-sized current sources are presented that achieve a high effective resolution on a small chip area. With only N current sources—again (nearly) irrespective of the mismatch—an effective resolution of $N-2$ bits can be achieved.

1.3 Outline

The present thesis is structured in two main parts: the first part deals with analog decoders, whereas the second part describes the corrected analog-to-digital and digital-to-analog converters.

In Chapter 2, after a brief introduction to analog decoders and factor graphs, simulation results obtained from discrete Hamming decoders are presented. The circuit structure of two generations of softgates, the building blocks of the discrete analog decoders, are discussed. Measurements of two analog decoder ASICs, for an (8,4,4) extended Hamming code and for a (16,5,8) Reed Muller code, demonstrate their full functionality. The latest design, a mixed signal coprocessor for decoding turbo codes, is presented. A novel analog circuit that can compute likelihoods of baseband PAM and QAM signals is introduced in Chapter 3. Several examples of soft symbol detection circuits for various setups are presented. Furthermore, a generalized current-weighting circuit that is related to the soft symbol detection circuit is described. Both circuits blend in nicely with the analog decoders.

The second main part begins with Chapter 4. A flash analog-to-digital converter with low-precision elements was implemented in a

0.18 μm-BiCMOS process. The measurement results of this chip and simulation results are presented. A digital correction scheme is introduced that leaves the effective resolution of the ADC nearly independent of the mismatch of its elements. A similar approach is taken in Chapter 5 for digital-to-analog converters: nearly irrespective of the current-source mismatch, a digitally-corrected digital-to-analog converter can achieve a high effective resolution. This is corroborated by high-level simulation results.

Chapter 6 concludes the thesis by pointing out the main contributions and by summarizing the chapters. An outlook containing possible directions for future research is given.

Appendix A specifies a collection of analog computational circuits; some of them are probably well-known, others may be more original.

Appendix B derives a formula on the limit of the average resolution of a flash ADC incorporating comparators with large mismatch.

Chapter 2

Analog Decoder Examples

2.1 Introductory Remarks

Following a suggestion by Wiberg et al. [62], Hagenauer et al. [27, 28] and Loeliger et al. [41] independently proposed to decode error correcting codes by novel analog electronic networks in 1998; these networks are a direct translation of the factor graph[1] [35] of such codes into networks of electronic “softgates” or “probability gates.” It is hoped, and partly corroborated by measurements, that such analog decoders consume substantially less power (for a given speed) than a digital implementation of the corresponding iterative decoding algorithm.

Since 1998, much effort has been spent towards turning these ideas into working chips. The first fully functional analog decoder of this type was built in 1998 by Lustenberger et al. using discrete transistors [45]. Lustenberger et al. then designed and manufactured first, a decoder for a four-state tail biting (18,9,5) convolutional code and later, a decoder for a (44,22,8) low density code, both in 0.8 μm BiCMOS technology [45, 46].

---

[1] A very brief description of factor graphs (in the context of analog decoders) will be given in Subsection 2.1.1. A good introduction to factor graphs and the sum-product algorithm is given in [40].

Unfortunately, both of these chips turned out to have some problems with the interface circuits (outside the actual decoder).

Hagenauer et al. demonstrated a perfectly working decoder for a two-state (16,8,3) tail-biting convolutional code in 0.25 μm BiCMOS technology. Winstead et al. [65] have fabricated a decoder of the (8,4,4) Hamming code in 0.5 μm CMOS technology; that decoder, too, was reported to have some minor design error. A second version of this decoder, now enhanced with a serial-to-parallel interface, was published in [64]. Also in 2004, Winstead reported a decoder for a (16,11) Hamming code, as well as a decoder for a $(16,11)^2$ product code in his thesis [66]. In 2003, Gaudet et al. presented an analog decoder with a programmable interleaver for a (48,16) turbo code [22, 23]. Recently, the successful implementation of a chip for decoding a (132,40) turbo code (according to the UMTS standard [14]) was reported by Vogrig et al. in [60]. A more detailed history of analog decoders is given in [48, 66].

2.1.1 Factor Graphs and Analog Decoders

A natural way of looking at error correcting codes is by means of graphical models (“Tanner graphs,” “factor graphs”). The decoding of such codes may be viewed as applying one of various versions of the general iterative “sum-product algorithm” to such graphical models [35, 62]. For the description of our decoders, we use Forney-style factor graphs as in [40].

As an example, a Forney-style factor graph (called “normal graph” in [17]) for the (8,4,4) extended Hamming code is shown in Figure 2.1.

The nodes represent either equality constraints or parity checks. The bits $u_0 \ldots u_3$ in Figure 2.1 will be called “information bits” and the bits $x_4 \ldots x_7$ will be called “parity bits.” The corresponding equations are given in (2.1)–(2.4), the parity check matrix in (2.5).

$$x_4 = u_0 + u_1 + u_2 \tag{2.1}$$

$$x_5 = u_0 + u_1 + u_3 \tag{2.2}$$

$$x_6 = u_0 + u_2 + u_3 \tag{2.3}$$

$$x_7 = u_1 + u_2 + u_3 \tag{2.4}$$

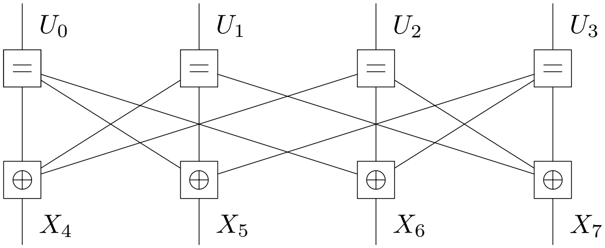


Figure 2.1: A Forney-style factor graph for the (8,4,4) extended Hamming code.

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

For further details, we refer to [17].

Note that the graph of Figure 2.1 is far from unique. In general, a huge variety of different graphs exists for the same code, and each such graph yields a different decoder. In the following subsections, the Hamming decoder examples were all derived from the factor graph shown in Figure 2.1 (except for the redundant Hamming decoder).

Any factor graph (or Forney-style factor graph, as, e.g., in Figure 2.1) may be interpreted as a blueprint for a decoder of the corresponding code. In this interpretation, each node of the graph may be viewed as a "processor." The decoder operates by passing "messages" along the edges of the graph; the messages are iteratively (re-)computed by the nodes/processors according to the rules of the sum-product algorithm [17, 35].

As observed in [29, 43], a factor graph such as shown in Figure 2.1 can be mapped directly into simple analog transistor circuits. In the resulting analog decoder, the iterations of the standard (discrete-time) decoding algorithm are replaced by the natural (continuous-time) settling behavior of the transistor network. The nodes ("processors") are translated into "softgates", which are wired according to the topology of the graph. Other than logic gates which operate on finite fields, e.g., binary numbers, these softgates perform computations with probabilities. The circuits rely on the exponential characteristics of either bipolar junction transistors or CMOS transistors operating in the subthreshold region; within the validity of the exponential model, the probability calculations are exact. For details see [43, 45].

Such an analog decoder can then be understood as a code-representing graph mapped on silicon, and the settling behavior of the circuit corresponds to the message-passing algorithm (e.g., the sum-product algorithm) on the factor graph.

2.1.2 The Channel Model

For all the following measurements and simulations the additive white Gaussian noise (AWGN) channel was used as channel model. This is a discrete-time, memoryless channel with output Y_i at time i. Y_i is the sum of the input X_i and the i.i.d. Gaussian noise Z_i with variance σ^2_{noise} [8]:

$$Y_i = X_i + Z_i, \qquad Z_i \sim \mathcal{N}(0, \sigma^2_{\text{noise}}) \tag{2.6}$$

The setup for transmitting a binary antipodal signal is shown in Figure 2.2.

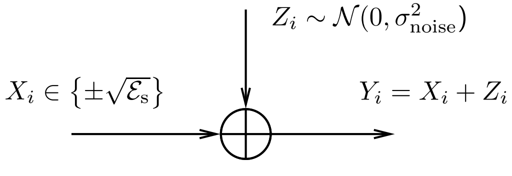


Figure 2.2: The AWGN-channel model.

In this thesis we will use the following definition of the signal-to-noise ratio (SNR):

Definition 2.1 (Signal-to-noise ratio, SNR). *The "signal-to-noise ratio" in dB is defined as*

$$\text{SNR}_{\text{dB}} = 10 \log_{10}\left(\frac{\mathcal{E}_\text{s}}{2\sigma^2_{\text{noise}} R}\right), \tag{2.7}$$

where $\mathcal{E}_\mathrm{s}$ *is the energy per transmitted symbol* X_i*,* σ^2_noise *is the noise variance of the AWGN channel and* R *is the rate of the code.*

Analog decoders require the likelihoods $P(Y_i|X_i = +\sqrt{\mathcal{E}_\mathrm{s}})$ and $P(Y_i|X_i = -\sqrt{\mathcal{E}_\mathrm{s}})$ as inputs. In order to compute such likelihoods from the channel output Y_i, the SNR of the channel is either known or an assumption thereof can be made. This is illustrated in Figure 2.3, where $\mathrm{SNR}_\mathrm{assumed}$ is the assumed SNR of the channel[2].

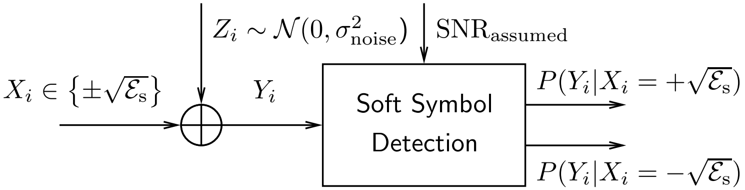


Figure 2.3: The soft symbol detector computes the likelihoods from the AWGN channel output and with the assumed channel SNR.

2.1.3 Transistor Circuit Notation

The main element for all the circuits presented in this thesis is certainly the transistor. For analog circuits the transistors are usually operated in the active region, and can therefore ideally be seen as voltage-controlled current sources. Here, both bipolar and MOS transistors can be used; the latter, however, must operate in weak inversion (also called subthreshold region) for the required exponential behavior.

The simple MOS transistor model for transistors operating in weak inversion and in the active region, leads to the following drain current, I_D:

$$I_\mathrm{D} = I_\mathrm{D0} e^{\frac{V_\mathrm{GS}}{nU_\mathrm{T}}}, \tag{2.8}$$

where I_D is the drain-current, I_D0 is a temperature- and technology-dependent constant, V_GS is the gate-source voltage, n the slope-factor of a MOS transistor and $U_\mathrm{T} = kT/q$ is the thermal voltage [32,47]. (k is the

---

[2] The parameter $\mathrm{SNR}_\mathrm{assumed}$ can be set to optimize the performance of the analog decoder; we observed that $\mathrm{SNR}_\mathrm{assumed} = 4$ dB yields good results.

Boltzmann constant, T the absolute temperature and q is the elementary charge.)

For bipolar transistors in the forward active mode, the collector current I_{C} is modeled as

$$I_{\mathrm{C}} = I_{\mathrm{S}} e^{\frac{V_{\mathrm{BE}}}{n U_{\mathrm{T}}}}, \tag{2.9}$$

where the saturation current I_{S} is a function of technology parameters, the transistor geometry and the absolute temperature T [26, 47]. The factor n is the "emission coefficient" and usually takes on values close to $n = 1$.

The definition of the terminals and the corresponding voltages, for both bipolar and MOS transistors, are given in Figure 2.4.

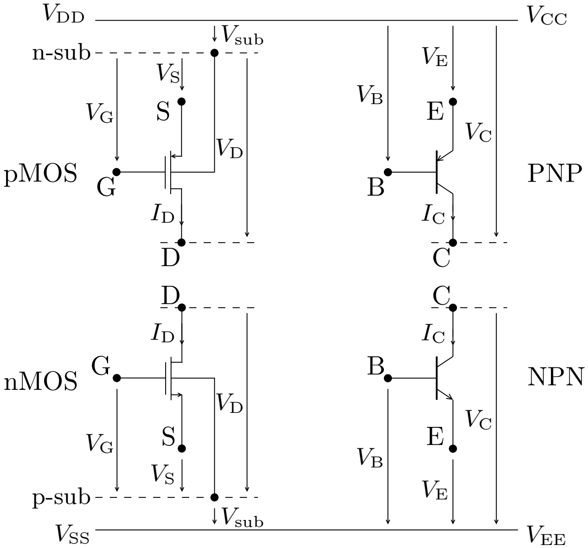


Figure 2.4: Voltage and terminal definitions for the MOS transistor (left) and the bipolar transistor (right).

Although the MOS transistors are 4-terminal devices, we often draw only the three main terminals: gate, drain and source. The substrate-terminal is omitted if it is connected to a supply-voltage, "n-sub" to V_{DD} or "p-sub" to V_{SS}.

2.1.4 Chapter Overview and Some Definitions

The experience with analog decoders is still very limited and solid experimental measurements of bit error rates, decoding speed, or the influence of transistor mismatch are scarce. In this chapter, after a brief introduction, a vast collection of measurements from different analog decoders is presented.

In Section 2.2 two series of softgates—building blocks of analog decoders—are introduced. Statistical measurements for such softgate chips, as well as bit-error rate (BER) curves for a "discrete" (8,4,4) extended Hamming decoder (built out of such softgates) are presented.

A full-custom analog CMOS chip for decoding the (8,4,4) extended Hamming code is presented in Section 2.3. Measured error rate curves for different operating conditions are shown and prove the chip's full functionality.

In Section 2.4 a more advanced analog decoder chip for a (16,5,8) Reed-Muller code, incorporating some "special features" (i.e., a nonstandard message-passing algorithm and a serial-to-parallel current input interface) are presented. Again, many BER curves demonstrate the chip's behavior and functionality.

The structure of an analog coprocessor for decoding a turbo code (according to the UMTS-standard) that has not yet been measured, is introduced in Section 2.5.

The chapter is then concluded in Section 2.6 with an outlook and some comments on analog decoders.

We will now introduce some definitions that will be used in the following sections.

Definition 2.2 (Sum current (I_{sum})). *After every softgate the output currents are rescaled (see Definition 2.4), so that the sum of all currents corresponding to one bit stays constant* $\sum_i I_{\text{out}}(i) = I_{\text{sum}}$. *As the currents are proportional to probabilities (or likelihoods), and probabilities sum to 1, the sum of all currents,* I_{sum}, *corresponds to a probability of 1.*

Remark 2.3 (Internal and external sum current (I_{sum}, $I_{\text{sum}}^{\text{pad}}$)). *The core of an analog decoder may run on very low current levels (e.g.,* $I_{\text{sum}} = 100$ nA*), lower than can be measured reliably. Therefore, the* *decoder chip may contain scaling-circuits that increase the current level from* I_{sum} *to* $I_{\text{sum}}^{\text{pad}}$*, from the decoder core to the pad of the chip;* $I_{\text{sum}}^{\text{pad}}$ *is the sum current seen at the outside of the chip.*

Definition 2.4 (Output scaling, input scaling). *In order to keep the current-level at a constant level throughout the decoder, either input scaling or output scaling is introduced: Before (for input scaling) or after (for output scaling) every softgate (or "node processor"), the sum of all currents corresponding to one variable are scaled to a fixed sum current* I_{sum} *(see Definition 2.2).*

Remark 2.5 (Bias point of analog decoders). *The underlying circuits of analog decoders are translinear, a term coined by Gilbert [24]. If second order effects are neglected, the computations carried out by translinear circuits are fundamentally exact and rely heavily on transistor physics. Unlike linearized circuits that operate around an operating point correctly, translinear circuits have a very wide dynamic range. For an analog decoder to operate properly, the underlying translinear circuits need to function correctly for the whole current range from 0 to* I_{sum}*. Therefore, analog decoders cannot be designed to work around a bias point, but they need to be designed to work properly with a nominal sum current* I_{sum}*.*

Remark 2.6 (Translinear circuits and transistors with exponential characteristics). *Translinear circuits rely on the exponential behavior of transistors. Either bipolar transistors or CMOS transistors in weak inversion show the desired exponential characteristic. All the analog decoder chips presented in this chapter operate with CMOS transistors in weak inversion. Although weakly inverted transistors tend to be slow, they do have the advantage of consuming very little power and are able to operate with very low supply voltages.*

Definition 2.7 (Decoding time). *The decoder's decoding time,* T_{dec}*, is defined as the time between the release of the decoder reset and the time the output is sampled.*

Definition 2.8 (Settling time). *The settling time* T_{settling}*, used as a measure for the speed of an analog decoder, is defined as the minimal decoding time the decoder needs for attaining the minimal error rate up to plot accuracy*[3]*.*

---

[3] We remark that, in the literature, "settling time" is sometimes not well defined and comparisons thereof can be misleading.

Definition 2.9 (Power consumption P_{tot}, P_{core}). *P_{tot} is the total power (measured) and P_{core} is the power dissipation of the decoder core without the interface. If nothing else is stated, nominal conditions are assumed.*

Definition 2.10 (Energy per decoded info bit). *The "energy per decoded info bit" is defined as $P_{\text{tot}} \cdot T_{\text{dec}}/(\text{number of information bits})$.*

2.2 Softgates, Building Blocks of Analog Decoders

In order to gain some deeper understanding of analog decoders and their behavior towards transistor mismatch, the basic building blocks of analog decoders were fabricated: two generations of approximately 150 fully bidirectional softgates—soft equality constraint gates (softEQU) as well as soft parity check gates (softXOR)—were implemented in a semi-custom process and packaged. Statistical measurements of the softgates were undertaken. Furthermore, these softgates were used to build various realizations of a discrete (8,4,4) Hamming decoder which were then characterized under diverse operating conditions.

The measurement results were obtained in a collaborative effort: Felix Lustenberger was responsible for the schematics of the first-generation of softgates; Patrik Strebel was in charge of the schematics of the second-generation softgates as well as of all softgate's chip-layouts. All measurements were conducted by Patrik Strebel, who also designed and manufactured the testing equipment which contains 16 12-bit D/A- and A/D-converters.

2.2.1 First-Generation Softgates

A collection of over 140 softXOR gates and just as many softEQU gates were produced, each as an individual ASIC in its own package.

Each gate has three bidirectional ports. For each of its three output ports, the softXOR gate computes the output according to (2.10).

$$\begin{bmatrix} p_{\text{out}}(0) \\ p_{\text{out}}(1) \end{bmatrix} = \begin{bmatrix} p_{\text{in1}}(0)\, p_{\text{in2}}(0) + p_{\text{in1}}(1)\, p_{\text{in2}}(1) \\ p_{\text{in1}}(0)\, p_{\text{in2}}(1) + p_{\text{in1}}(1)\, p_{\text{in2}}(0) \end{bmatrix} \tag{2.10}$$

The softEQU gate computes the three output ports according to (2.11), where the scale factor γ is implicitly defined by the condition $p_{\text{out}}(0) + p_{\text{out}}(1) = 1$.

$$\begin{bmatrix} p_{\text{out}}(0) \\ p_{\text{out}}(1) \end{bmatrix} = \gamma \begin{bmatrix} p_{\text{in1}}(0)\, p_{\text{in2}}(0) \\ p_{\text{in1}}(1)\, p_{\text{in2}}(1) \end{bmatrix} \tag{2.11}$$

As further described in [43], probabilities are represented by currents. All the transistors operate in subthreshold mode, nominally at a current-level of $I_{\text{sum}} = I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$ (see Definition 2.2).

The Circuit

The first-generation softgate circuit was implemented in a 0.8 μm CMOS process (Philips C175SC) using the inexpensive semi-custom mixed-signal array “MD300” provided by Microdul AG, Zürich. Special care was taken to minimize the mismatch of corresponding transistors in current mirrors and differential pairs: 24 unit-transistors were connected in a parallel and interleaved manner ($W_{\text{N,P}} = 24 \times 6\ \mu\text{m} = 144\mu\text{m}$, $L_{\text{N,P}} = 5.6\mu\text{m}$) to form a common-centroid layout [32]. The nMOS- and pMOS-transistors were sized equally. In order to keep all the transistors in weak inversion, the nominal sum current was chosen to be $I_{\text{sum}} = 1\mu\text{A}$. The nominal supply voltage was fixed to $V_{\text{dd}} = 5$ V. The complete schematics of the softEQU and the softXOR are given in Figures 2.5 and 2.6, respectively. The circuit topologies of the two probability gates are further described in [45].

This collection of softgates now allows us to carry out a large variety of measurements. First we can examine the characteristic of a single softgate. We then have the possibility to build decoders for different codes. For each code there exists a large family of suitable decoder structures (which we call realizations). Once we have wired one such realization on the breadboard, the impact of transistor mismatch can be analyzed by plugging different sets of ASICs into the sockets. In addition, the influence of some other conditions (e.g. supply voltage, decoding time) on the decoder's performance can be studied.

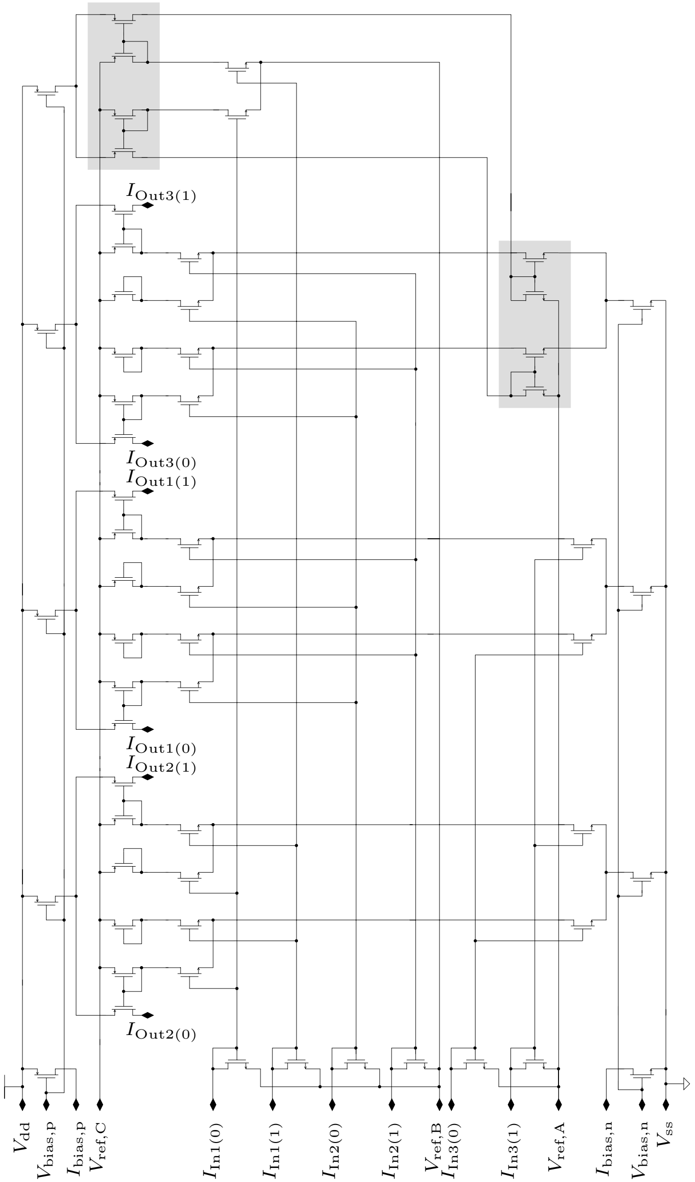


Figure 2.5: Schematic of the first-generation softEQU. The grey boxes mark the unbalance between the three outputs.

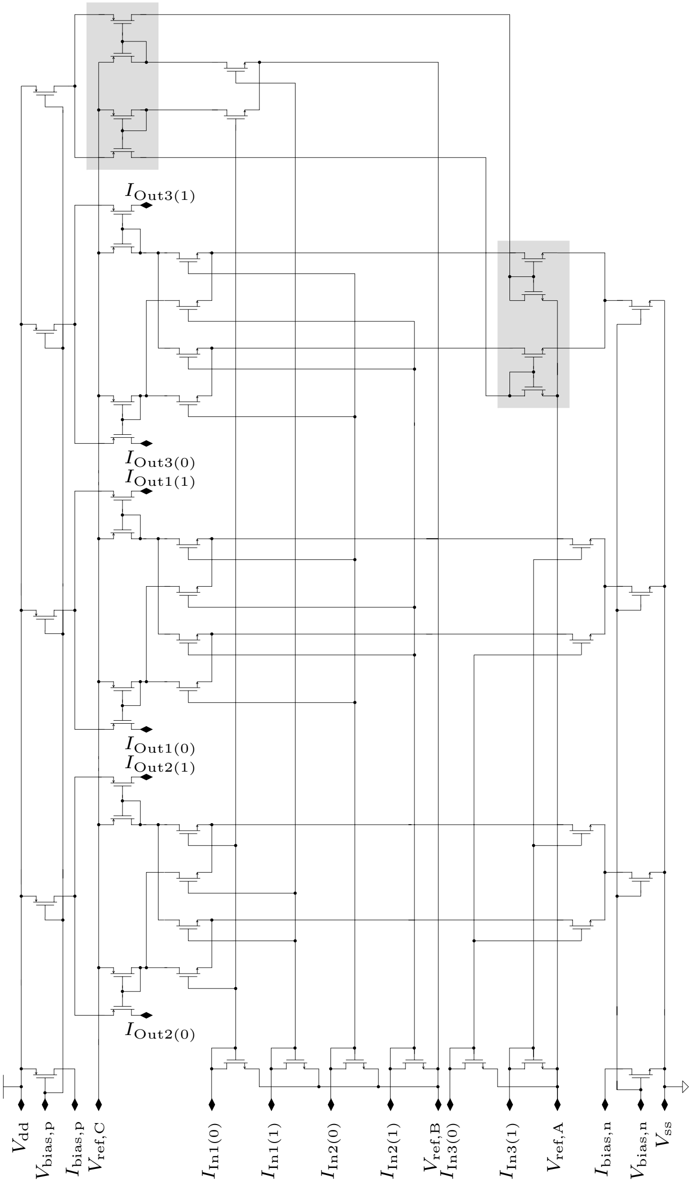


Figure 2.6: Schematic of the first-generation softXOR. The grey boxes mark the unbalance between the three outputs.

Softgate Measurements

All measurements were carried out on our lab's own measuring device, which contains 16 12-bit D/A- and A/D-converters. The softgates' characteristics were verified first: 50 bidirectional softEQU gates as well as 50 bidirectional softXOR gates were measured and simulated. (One fully bidirectional softgate contains three unidirectional softgates.) The results for nominal operating conditions are shown in Figure 2.7 for the softEQU gates and in Figure 2.8 for the softXOR gates. The solid lines represent the ideal characteristics of the softgates, whereas the dashed lines show the 10 and 90 percentiles for the 3×50 measured characteristics of the two types of gates (i.e., 80 percent of the measurements lie between the dashed lines). The varying parameter for all the softgate measurements is the log-likelihood ratio of the second input: $\mathrm{LLR}_{\mathrm{in}2} = \log_{10}[p_{\mathrm{in},2}(0)/p_{\mathrm{in},2}(1)]$.

The variations in the softgates' characteristics are primarily due to transistor mismatch (and to some degree also due to asymmetries among the circuits); some deviation from the ideal characteristic is also due to the deviations of the real transistors from our exponential model. The flattening of the curves for large log-likelihood ratios of the output of the softEQU gate, as seen in Figure 2.7, is an artifact caused by the resolution limit of the measurement equipment.

A fully bidirectional softgate for three variables contains three unidirectional softgates. Concerning "output 3", an unbalance can be found in both softgates: "input 1" is mirrored and scaled one more time before being passed to the current multiplier that generates "output 3." These additional current-scalers and current-mirrors are marked in Figures 2.5 and 2.6 with grey boxes.

This unbalance can be seen in the measurements of the outputs: if the 3 outputs are considered separately, the one with the additional current-scaling transistors shows systematically worse behavior than the other two outputs. Figures 2.9(c) and 2.10(c) result from the third output and exhibit a worse behavior than the figures corresponding to outputs 1 and 2. It might be interesting to note that, while all softgate outputs tend to be closer to the neutral message than the ideal output suggests (i.e., $|\mathrm{LLR}_{\mathrm{out,\ measured}}| < |\mathrm{LLR}_{\mathrm{out,\ ideal}}|$), "output 3" of the softXOR gate seems to be different, i.e., $|\mathrm{LLR}_{\mathrm{out,\ measured}}| > |\mathrm{LLR}_{\mathrm{out,\ ideal}}|$; we do not know where this effect comes from.

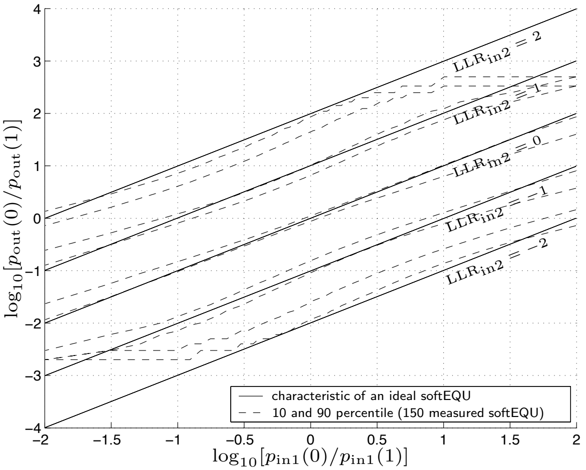


Figure 2.7: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the softEQU gate operating in weak inversion. The varying parameter is the log-likelihood ratio of the second input: $\text{LLR}_{\text{in2}} = \log_{10}[p_{\text{in2}}(0)/p_{\text{in2}}(1)]$.

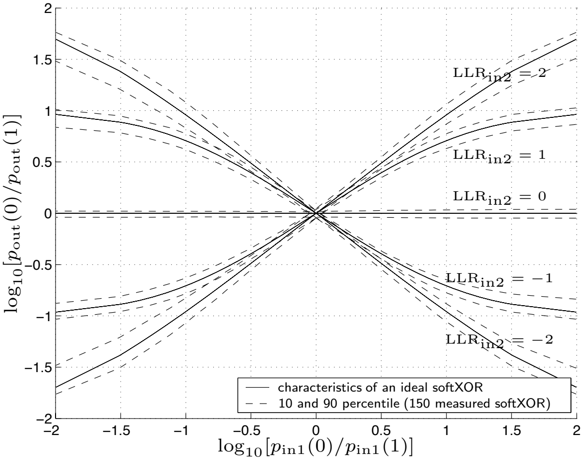
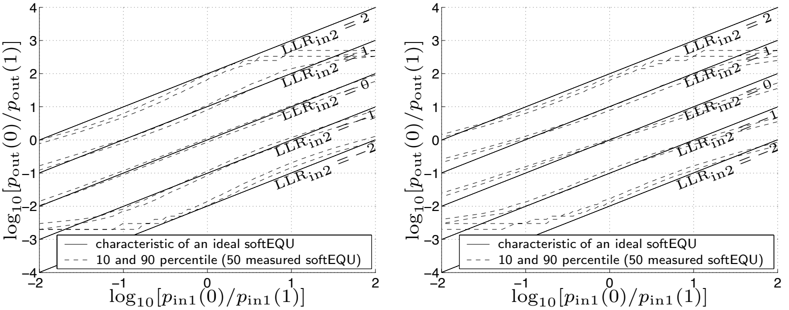
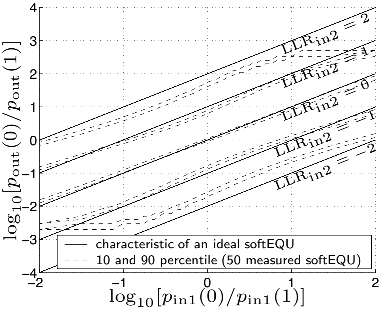


Figure 2.8: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the softXOR gate operating in weak inversion. The varying parameter is the log-likelihood ratio of the second input: $\text{LLR}_{\text{in2}} = \log_{10}[p_{\text{in2}}(0)/p_{\text{in2}}(1)]$.

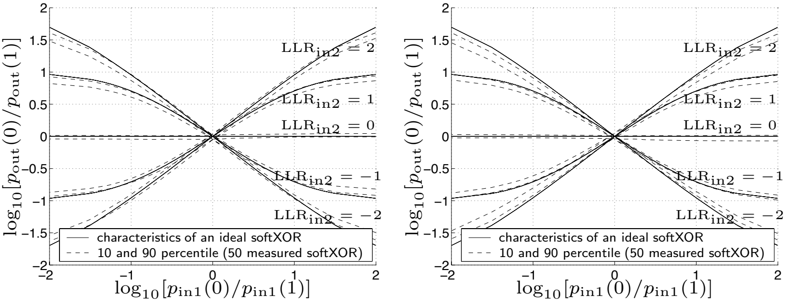


(a) Output 1 **(b)** Output 2

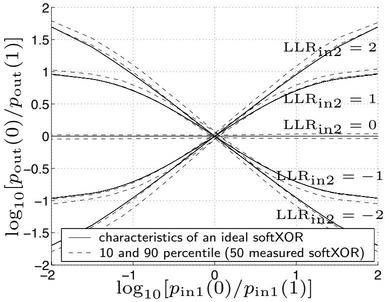


(c) Output 3

Figure 2.9: Ideal and measured (10 and 90 percentile, 3×50 measurements) characteristics of the softEQU gate operating in weak inversion. The varying parameter is the log-likelihood ratio of the second input: $\text{LLR}_{\text{in2}} = \log_{10}[p_{\text{in2}}(0)/p_{\text{in2}}(1)]$. The unbalance among the different outputs can be observed.



(a) Output 1 **(b)** Output 2



(c) Output 3

Figure 2.10: Ideal and measured (10 and 90 percentile, 3 × 50 measurements) characteristics of the softXOR gate operating in weak inversion. The varying parameter is the log-likelihood ratio of the second input: $\text{LLR}_{\text{in2}} = \log_{10}[p_{\text{in2}}(0)/p_{\text{in2}}(1)]$. The unbalance among the different outputs can be observed.

By increasing the sum current (here from $I_{\text{sum}} = 1\ \mu\text{A}$ to $I_{\text{sum}} = 30\ \mu\text{A}$), the softgate's transistors leave the subthreshold region and, therefore, move away from the exponential characteristics. As will be seen later, the analog decoder's settling speed may improve by increasing the sum current. The behavior of the discrete softgates in strong inversion is illustrated by the statistics presented in Figures 2.11 and 2.12.

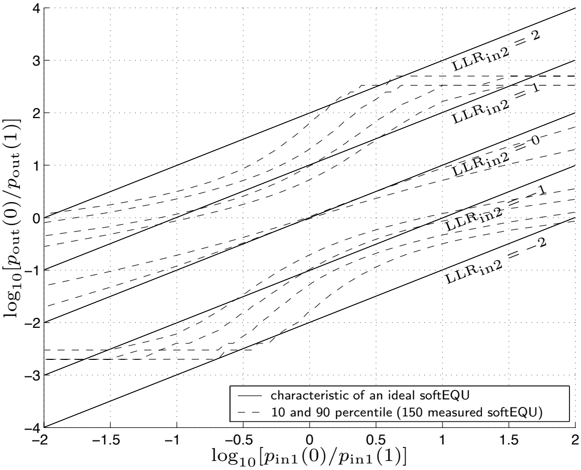


Figure 2.11: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the softEQU gate operating in strong inversion. The varying parameter is the log-likelihood ratio of the second input.

We observe that the softEQU gate deviates more from the ideal characteristics than the softXOR gate does. (Again, the flattening of the softEQU's characteristic results from the limited resolution of the measurement device—with a sum current in the μA-range and for an output of $\log_{10}[p_{\text{in2}}(0)/p_{\text{in2}}(1)] = 4$, the measurement device would need to measure currents in the sub-nA range correctly.)

After having measured the single softgates, decoders can be built and measured with these building blocks. Measurements for an (8,4,4) extended Hamming decoder are presented in the following subsection.

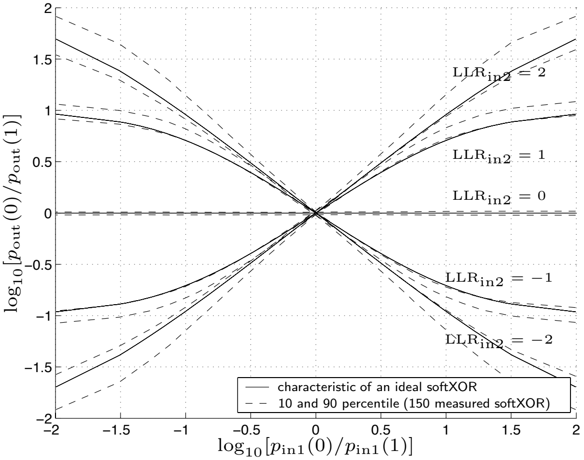


Figure 2.12: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the softXOR gate operating in strong inversion. The varying parameter is the log-likelihood ratio of the second input.

Discrete (8,4,4) Extended Hamming Decoder

The previously discussed softgates were used to build discrete Hamming decoders on breadboard level as shown in Figure 2.13. From that photograph, it might seem doubtful whether such a decoder can work at all; in fact, also despite the low current levels, quite robust behavior was observed, as will be detailed below.

For each code there exists a large family of suitable decoder structures (which we call realizations). However, for all the following measurements only the realization described by the factor graph shown in Figure 2.1 was considered.

After having wired this realization on the breadboard, the impact of transistor mismatch can be analyzed by plugging different sets of softgate chips (all stemming from the same wafer) into the sockets. In addition, the influence of some other conditions (e.g., supply voltage, decoding time, increased sum current) on the decoder's performance can be studied.

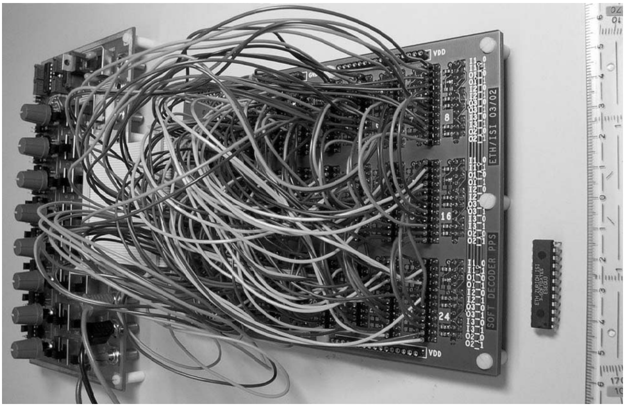


Figure 2.13: Breadboard-level decoder with a softgate chip of the first-generation.

In order to determine the performance of the decoder, the error rate curves for the additive white Gaussian noise (AWGN) channel was measured. The transmission over the noisy channel was simulated in software, and the resulting "noisy" codewords were applied in parallel to the decoder; after a fixed and preset time the A/D-converters then measured the decoder's outputs. The plots in this section were obtained with a decoding time of 50 ms (see Definition 2.7).

The effect of transistor mismatch was studied by plugging in 11 different chip sets (all stemming from the same wafer) into the sockets of the decoder shown in Figure 2.13.

For the AWGN channel, 10 error rate curves were measured up to an SNR of 6 dB; one measurement (which nearly took 2 weeks!) was measured up to an SNR of 8 dB. For the following measurements the channel SNR assumed by the receiver was the same as the actual SNR of the channel. The results of these measurements are summarized in Figure 2.14, which shows both the frame error rate (including errors both in the information bits and in the parity bits) and the average information-bit error rate. Also shown in Figure 2.14 are the corresponding curves for an "ideal" analog decoder with ideal softgates according to (2.10) and (2.11). (Since the graph of Figure 2.1 has loops, that ideal analog decoder is not a MAP decoder of the code.) Somewhat surprisingly, the measured performance curves of different chip sets are almost identical; the influence of transistor mismatch, which appears to be quite significant in Figures 2.7 and 2.8, is hardly noticeable in the overall decoder performance.

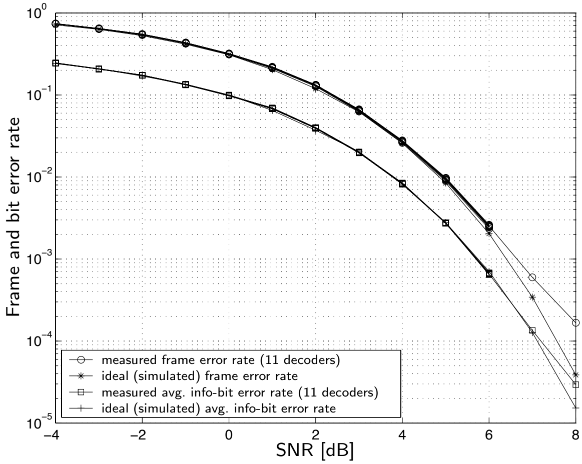


Figure 2.14: Measured error rate curves of 11 versions (each with a different set of chips) of the decoder of Figure 2.1 for nominal conditions ($I_{\mathrm{sum}}^{\mathrm{pad}} = 1\ \mu$A). Ideal frame error rate and ideal average information-bit error rate for such a decoder.

In order to determine the speed of the decoder, characterized by the settling time T_{settling} (see Definition 2.8) for nominal conditions, bit error rate curves for varying decoding times $T_{\mathrm{dec}} = 1, 2, 5, 10, 20, 50$ ms were measured. These measurements are presented in Figure 2.15; best performance is achieved for a decoding time of $T_{\mathrm{dec}} = 50$ ms and no performance gain is obtained by further increasing the decoding time. Applying Definition 2.8, the settling time amounts to $T_{\mathrm{settling}} = 50$ ms.

By increasing the sum current, the transistors move away from the slow subthreshold region and faster decoders can be obtained. However, as the transistors leave the exponential region the BER curves may show worse performance. Both can be observed in Figures 2.16 and 2.17: for a decoder operated at a sum current of $I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu$A (instead of $I_{\mathrm{sum}}^{\mathrm{pad}} = 1\ \mu$A for nominal conditions) the decoding time is decreased from 100 ms to 0.1 ms and the BER curves reveal no further loss of

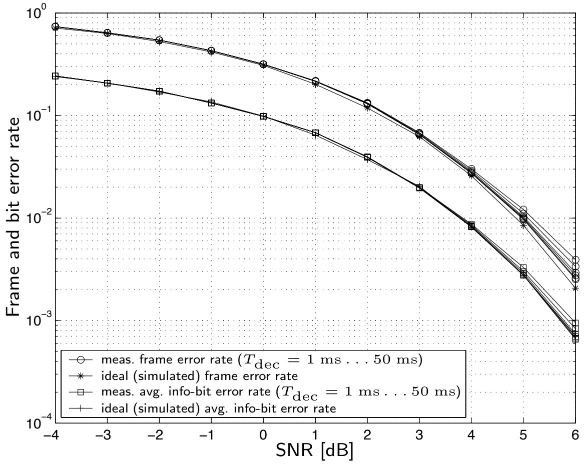


Figure 2.15: Measured error rate for varying decoding times and nominal conditions ($T_{\mathrm{dec}} = 1, 2, 5, 10, 20, 50$ ms, $I_{\mathrm{sum}}^{\mathrm{pad}} = 1$ μA) of the decoder of Figure 2.1 for the AWGN channel. Ideal error rates for such a decoder.

performance[4]. However, increasing the sum current leads to a worse performance (as can be seen in Figure 2.17) where the same decoder is measured for various sum currents, for $I_{\mathrm{sum}}^{\mathrm{pad}} = 1$ μA . . . 100 μA. Best performance is obtained with $I_{\mathrm{sum}}^{\mathrm{pad}} = 1$ μA, worst with $I_{\mathrm{sum}}^{\mathrm{pad}} = 100$ μA. The worsened BER curves for higher sum currents seem plausible by observing the characteristics of the softgates at higher current levels, as presented in Figures 2.11 and 2.12.

The softgate chips were designed for a nominal supply voltage of $V_{\mathrm{dd}} = 5$ V; out of curiosity, and because transistors operating in the subthreshold region only need a small drain-source voltage to stay saturated, the supply voltage was lowered [13]. The measurements are presented in Figure 2.18: nearly the same bit error rate performance was achieved by the decoder with any supply-voltage between $V_{\mathrm{dd}} = 5$ V and $V_{\mathrm{dd}} = 1.2$ V; only at supply voltages below 1.2 V the decoder's error correcting capabilities decreased.

---

[4] As the analog decoder is not a maximum-likelihood (ML) decoder, an inherent loss of performance compared to the ML decoder is inevitable.

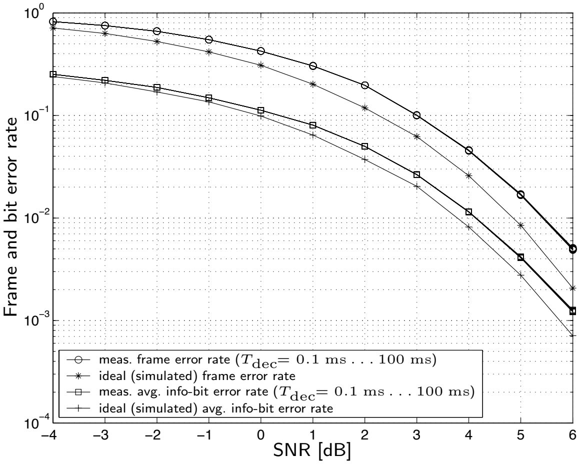


Figure 2.16: Measured and ideal error rates for varying decoding times and an increased sum current ($T_{\mathrm{dec}} = 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\,\mathrm{ms}$, $I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu\mathrm{A}$) of the decoder of Figure 2.1 for the AWGN channel.

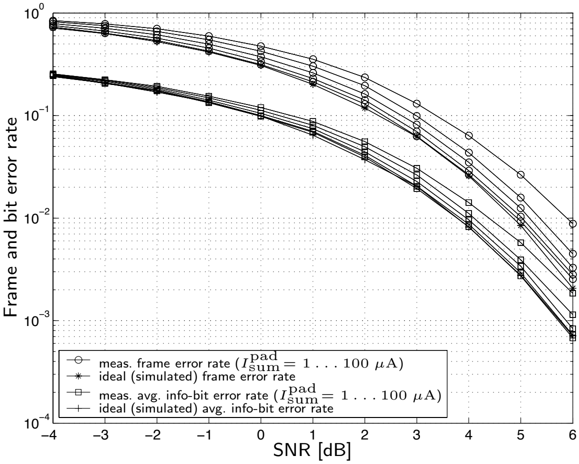


Figure 2.17: Measured error rates for varying sum currents ($I_{\mathrm{sum}}^{\mathrm{pad}} = 1, 3, 10, 30, 100\ \mu\mathrm{A}$ (from best to worst), $T_{\mathrm{dec}} = 10$ ms) of the decoder of Figure 2.1 for the AWGN channel. Ideal error rates for such a decoder.

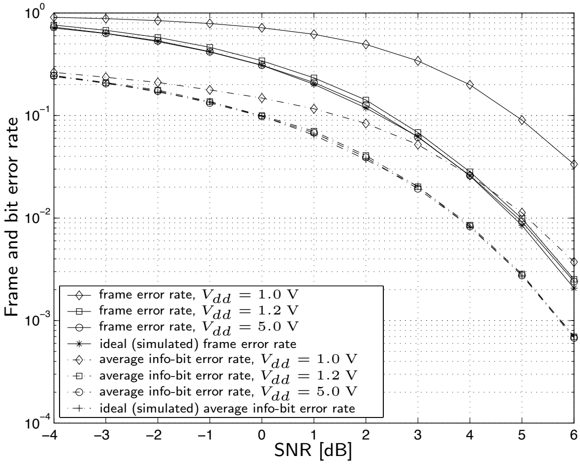


Figure 2.18: Measured error rates for varying supply voltages ($V_{\mathrm{dd}} = 5$ V, 1.2 V, 1.0 V, $I^{\mathrm{pad}}_{\mathrm{sum}} = 1\ \mu$A, $T_{\mathrm{dec}} = 10$ ms) of the decoder of Figure 2.1 for the AWGN channel. Ideal error rate curves for such a decoder.

Conclusions

A summary of the first-generation softgates is given in Table 2.1.

We have presented measurements of bit error rates and frame error rates for an analog decoder implemented with probability gates on the breadboard level. We have observed that the measured overall performance is virtually identical with that of an ideal analog decoder despite the substantial measured non-idealities on the softgate level.

Although the circuits are nearly the same, the softEQU gate circuits seem to be more vulnerable against transistor non-idealities than the softXOR gates. Furthermore, the unsymmetry of the softgate circuits can be observed: while the circuit is generally under-confident (i.e., the currents are closer to a uniform distribution than they should be and the entropy is smaller), the softXOR gate's "output 3" is overconfident. These effects cannot be explained with high-level simulations incorporating simple mismatch models.

While the circuits were designed for a supply voltage of 5 V, we have

Technology:	0.8 μm CMOS process (Philips C175SC), semi-custom mixed-signal array "MD300"
Die size:	2.88×2.4 mm^2 (2 tiles[a])
nMOS size ($W \times L$):	24×6.0 μm $\times$ 5.6 μm
pMOS size ($W \times L$):	24×6.0 μm $\times$ 5.6 μm
Supply voltage (nom./min.):	5 V/1.2 V
Sum current ($I_{\mathrm{sum}}/I_{\mathrm{sum}}^{\mathrm{pad}}$)	1 μA/1 μA
Pin-count:	24
Features:	Input- and output-scaling is supported

---

[a] Only a small part of the area was used; the major part of each tile contained digital logic circuits that were left unused.

Table 2.1: Summary of the first-generation softgates.

observed no substantial loss of performance when operating them at a supply voltage as low as 1.2 V.

2.2.2 Second-Generation Softgates

A second-generation of softgates—again softEQU as well as SoftXOR—was designed and produced. Patrik Strebel designed the chips and conducted all the measurements. The main features and differences to the first-generation softgates are as follows:

- Reset transistors at all the inputs are added, in order to set $I_{\mathrm{in}}(0) = I_{\mathrm{in}}(1) = I_{\mathrm{sum}}^{\mathrm{pad}}/2$.
- Only output scaling is supported, but no input scaling.
- The reference voltages are generated on-chip with a simple diode-connected transistor ($\mathrm{D}_1 \ldots \mathrm{D}_3$ in Figures 2.19 and 2.20).
- The "computing transistors" (e.g., the two differential pairs of the matrix multiplier) use transistors that are six times smaller than in the first-generation of softgates, the sum currents are scaled accordingly; the input transistors are of same size as in the first-generation softgates.

- The three unidirectional softgates inside a fully bidirectional softgate chip all have the same topology.

The second-generation softgate circuit was also implemented in a 0.8 μm CMOS process (Philips C175SC) as a semi-custom IC. The dimensions of the transistors of the input stage were chosen as in the first-generation softgates: $W_{\mathrm{N,P}} = 24 \times 6$ μm $= 144$ μm, $L_{\mathrm{N,P}} = 5.6$ μm for nMOS and pMOS. Special care was taken to minimize the mismatch of corresponding transistors in current mirrors and differential pairs. The three reset transistors were designed to be of size $W_{\mathrm{N,P}} = 32 \times 6$ μm $=$ 192 μm, $L_{\mathrm{N,P}} = 5.6\mu$m. In order to conserve chip area, the remaining "computing transistors", as well as the internal sum current, were dimensioned 6 times smaller: $W_{\mathrm{N,P}} = 4 \times 6$ μm $= 24$ μm, $L_{\mathrm{N,P}} = 5.6\mu$m, $I_{\mathrm{sum}} = \frac{1}{6} I_{\mathrm{sum}}^{\mathrm{pad}} = 0.17$ μA. Again, the nMOS- and pMOS-transistors were sized equally and operated in weak inversion. The nominal supply voltage was fixed to $V_{\mathrm{dd}} = 5$ V.

The schematics of these second-generation softgates are shown in Figures 2.19 and 2.20.

Statistical Measurements

The characteristics of the softgates were first examined: with our lab's measurement device 50 bidirectional softEQU gates as well as 50 bidirectional softXOR gates were measured and simulated.

The results for nominal operating conditions are shown in Figure 2.21 for the softEQU gates and in Figure 2.22 for the softXOR gates. The solid lines represent the ideal characteristics of the softgates, whereas the dashed lines show the 10 and 90 percentiles for the 3×50 measured characteristics of the two types of gates (i.e., 80 percent of the measurements lie between the dashed lines).

Under nominal conditions, the second-generation softEQU gate shows very similar behavior to the first-generation softEQU gate characterized in Figure 2.7. The softXOR gates of the second-generation have slightly worse characteristics than the first-generation did (see Figure 2.7); the output probabilities tend to be overconfident, i.e., the output probabilities tend to be farther away from a uniform distribution than they ought to be.

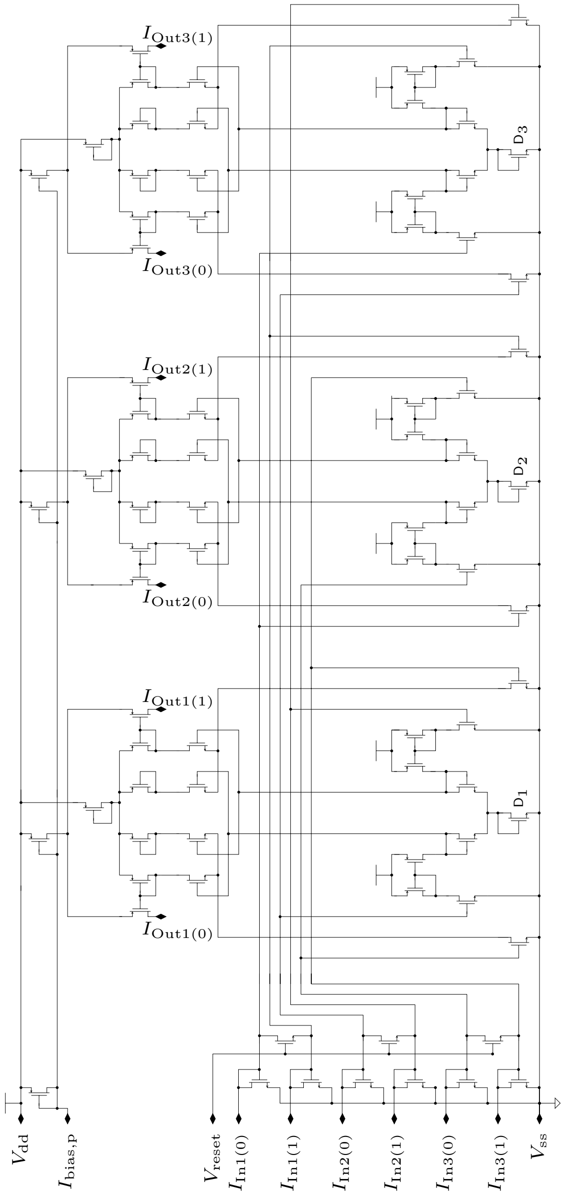


Figure 2.19: Schematic of the first-generation softEQU.

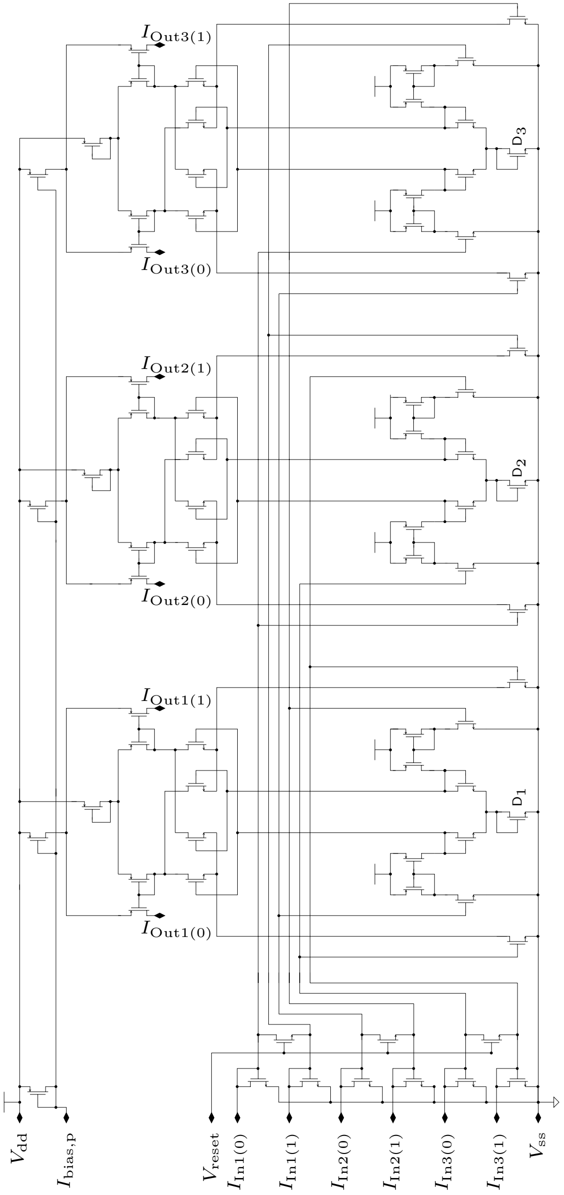


Figure 2.20: Schematic of the second-generation softXOR.

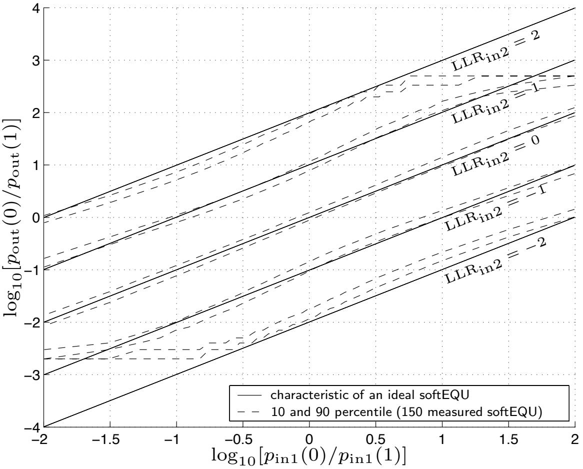


Figure 2.21: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the second-generation softEQU operating in weak inversion. The varying parameter is the log-likelihood ratio of the second input.

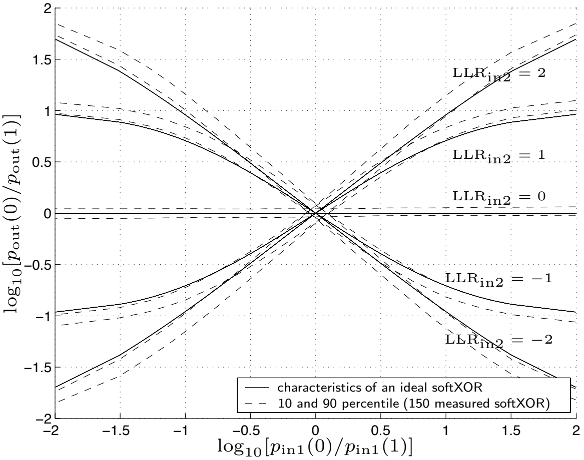


Figure 2.22: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the second-generation softXOR operating in weak inversion. The varying parameter is the log-likelihood ratio of the second input.

The sum current is now increased to $I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$: the softgate's transistors leave the subthreshold region and move away from the exponential characteristics. The behavior of the discrete softgates in strong inversion is illustrated by the statistics presented in Figures 2.23 and 2.24. With the increased sum current, the measured characteristics of the softEQU gate deviates strongly from the ideal characteristics. The softXOR gate however, reveals nearly identical behavior in strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$) as in weak inversion ($I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$).

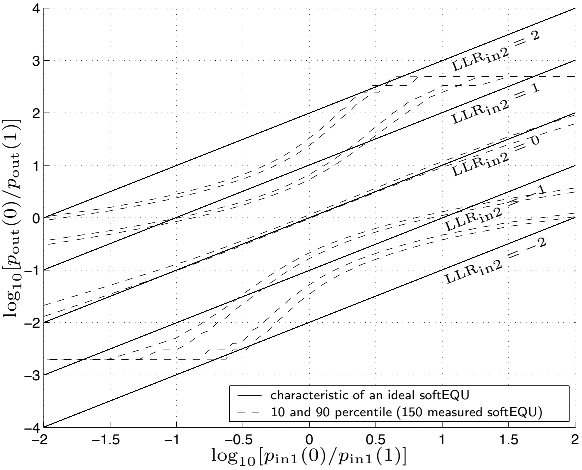


Figure 2.23: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the second-generation softEQU operating in strong inversion. The varying parameter is the log-likelihood ratio of the second input.

Discrete (8,4,4) Extended Hamming Decoder

The second-generation softgates were used to build discrete Hamming decoders on breadboard level described by the factor graph presented in Figure 2.1.

Bit error rate and frame error rate curves were measured for the AWGN channel. This time, however, the assumed channel SNR was fixed to $\text{SNR}_{\text{assumed}} = 4$ dB, independent of the actual channel SNR

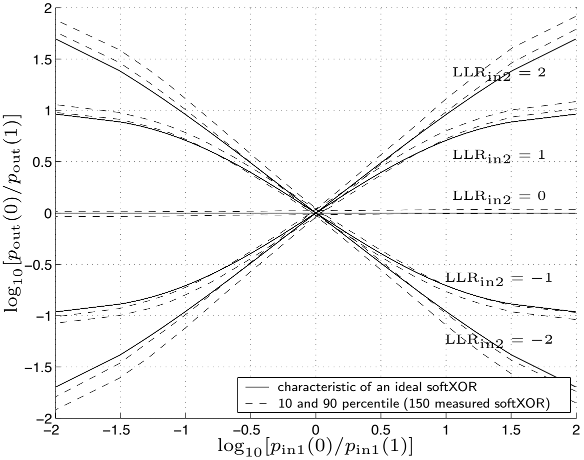


Figure 2.24: Ideal and measured (10 and 90 percentile, 150 measurements) characteristics of the second-generation softXOR operating in strong inversion. The varying parameter is the log-likelihood ratio of the second input.

(see Section 2.1.2). All the measurements were performed with the same measurement device and in the same way as described in Section 2.2.1.

The effect of transistor mismatch was studied by measuring 10 decoders with different chip sets (all stemming from the same wafer) for the AWGN channel up to an SNR of 6 dB. The results of these measurements are shown in Figures 2.25 and 2.26 for weak inversion ($I_{\text{sum}}^{\text{pad}} = 1\ \mu$A) and for strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu$A), respectively. Both the frame error rate (which includes errors both in the information bits and in the parity bits) and the average information-bit error rate, as well as the corresponding "ideal" curves are presented. Compared to the decoders built from the first-generation of softgates, the performance of these decoders deviates more from the ideal curves[5].

Increasing the sum current to $I_{\text{sum}}^{\text{pad}} = 30\ \mu$A (strong inversion), leads to a behavior of the softgates that lies farther away from the

---

[5] The error rate curves of the first-generation and second-generation cannot be compared directly, as for these measurements the assumed channel SNR was kept fixed (SNR$_{\text{assumed}}$ = 4 dB), whereas for the first measurements, the assumed channel SNR was adapted to the real channel SNR (SNR$_{\text{assumed}}$ = SNR$_{\text{channel}}$).

ideal curves (see Figures 2.23 and 2.24) and, therefore, also to an error rate curve that deviates more from the ideal curve. This can also be seen in Figure 2.27 where error rate curves for varying sum currents ($I_{\text{sum}}^{\text{pad}} = 1, 3, 10, 30, 100\ \mu$A) are presented; the performance decreases monotonically for increased sum currents.

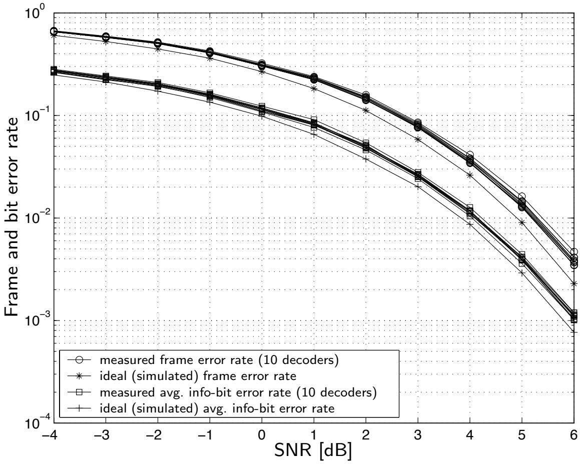


Figure 2.25: Error rate curves of 10 versions (each with a different set of second-generation softgates) of the decoder of Figure 2.1 under nominal conditions for the AWGN channel. Also shown are ideal error rate curves for such a decoder.

From Figure 2.28 it can be seen that the standard Hamming decoder built out of second-generation softgates can operate at a supply voltage as low as $V_{dd} = 2.7$ V for a nominal sum current of $I_{\text{sum}}^{\text{pad}} = 1\ \mu$A. For an increased sum current ($I_{\text{sum}}^{\text{pad}} = 30\ \mu$A) the decoder functions properly (i.e., without any loss of performance) down to $V_{\text{dd}} = 3.5$ V. This is illustrated in Figure 2.29.

Redundant Discrete (8,4,4) Extended Hamming Decoder

The advantage of operating the decoders at higher current levels is the increased decoding speed and also the decreased mismatch. However, as the transistors move away from their ideal exponential characteristics, a loss of error correcting performance comes along with an increased cur-

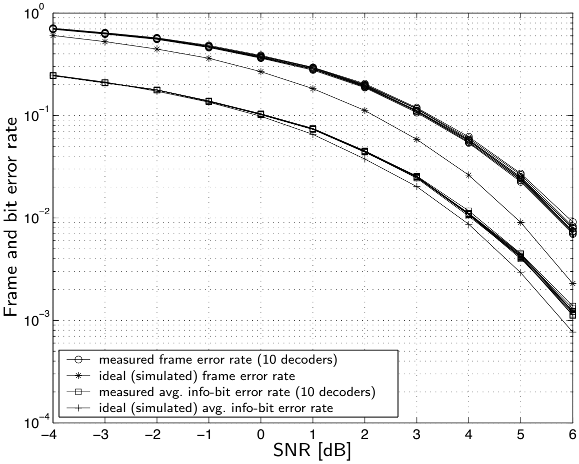


Figure 2.26: Error rate curves of 10 versions (each with a different set of second-generation softgates) of the decoder of Figure 2.1 in strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$) for the AWGN channel. Also shown are ideal error rate curves for such a decoder.

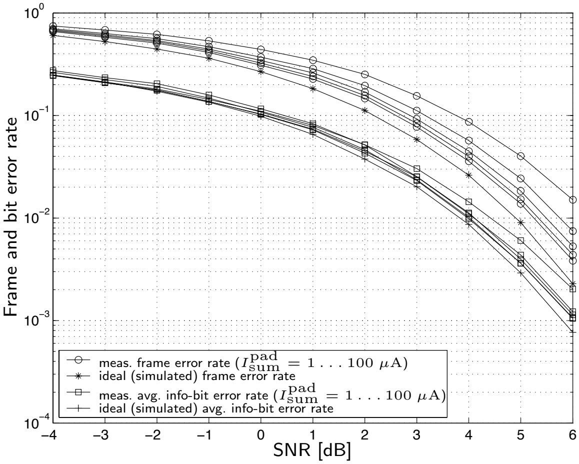


Figure 2.27: Measured and simulated error rate curves of the second-generation decoder of Figure 2.1 measured with varying sum current ($I_{\text{sum}}^{\text{pad}} = 1, 3, 10, 30, 100\ \mu\text{A}$, best to worst performance) for the AWGN channel.

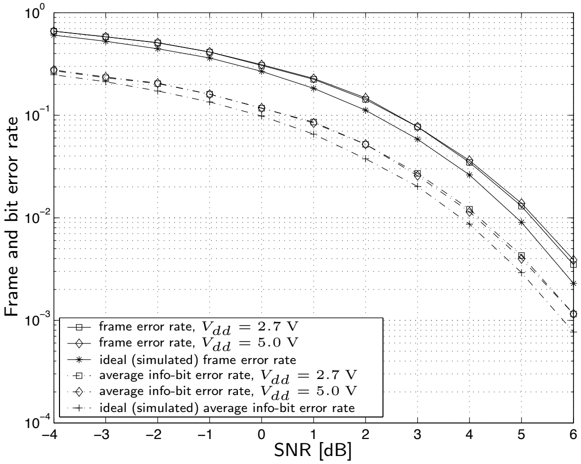


Figure 2.28: Error rate curves of the decoder built from second-generation softgates measured with varying supply voltages ($V_{\text{dd}} = 2.7\ \text{V}, 5.0\ \text{V}$, $I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$) for the AWGN channel.

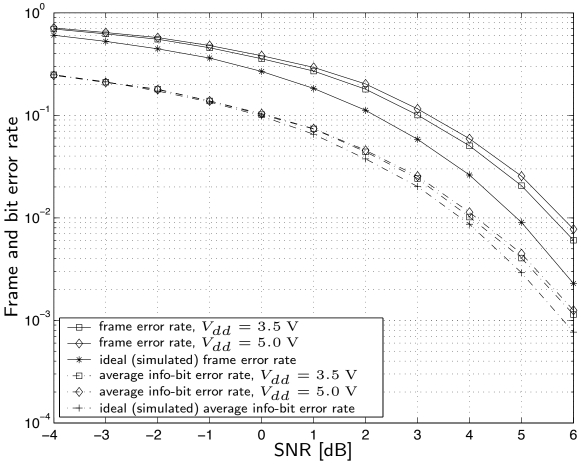


Figure 2.29: Error rate curves for the standard Hamming decoder built out of second-generation softgates measured with varying supply voltage ($V_{\text{dd}} = 3.5\ \text{V}, 5.0\ \text{V}$, $I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$) for the AWGN channel.

rent level. An idea that was described and corroborated in [20] is to use redundant decoders. A redundant decoder can lock up in certain states while decoding and no input likelihoods can then lead the decoder back to a “neutral” state. Therefore, reset switches between the softgates are essential to free the decoder from such a state and decode each received word independently of the past.

In order to obtain a redundant realization, four symmetric parity checks were added to the parity check matrix of the standard realization given in (2.5); the resulting parity check matrix is shown in (2.12)—the upper part of the matrix is the same as in (2.5), the lower part is made up of the additional (redundant) parity checks.

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ & & & & & & & \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{2.12}$$

The factor graph describing the redundant decoder for the (8,4,4) extended Hamming code (corresponding to (2.12)) is given in Figure 2.30. This redundant realization is further described in [50].

The redundant decoder was first measured for 10 different chip sets under nominal conditions ($V_{\mathrm{dd}} = 5$ V, $I_{\mathrm{sum}}^{\mathrm{pad}} = 1$ μA). Especially for higher SNR, the transistor mismatch can be clearly observed in Figure 2.31: the curves obtained from the 10 different chip sets deviate from the ideal (redundant) error rate curves and the curves are also spread out. It can also be seen that the performance of the redundant realization clearly outperforms the standard realization (straight line vs. dashed line in Figure 2.31). This increased performance, however, comes with an increased complexity and with a higher power consumption.

As described in [20] the redundant realization of the (8,4,4) extended Hamming code should be more immune towards non-ideal (exponential) or even quadratic characteristics of the transistors. This behavior was

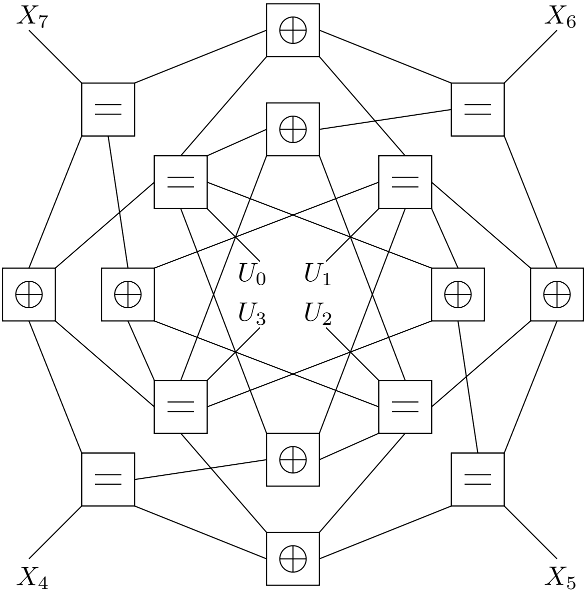


Figure 2.30: A Forney-style factor graph of a redundant decoder for the (8,4,4) extended Hamming code.

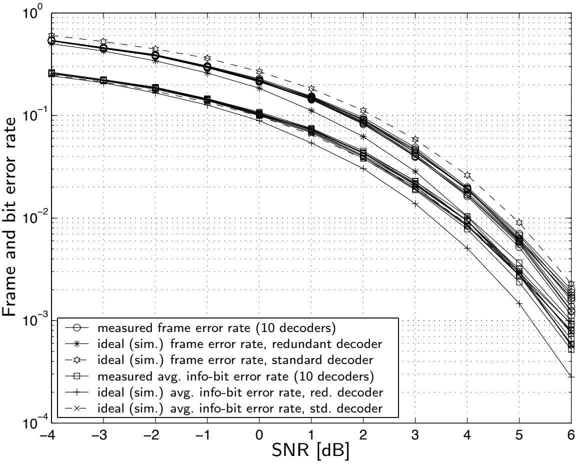


Figure 2.31: Measured error rate curves of 10 versions (each with a different set of second-generation softgates) and simulations of the redundant decoder of Figure 2.30 under nominal conditions ($I_{\text{sum}}^{\text{pad}} = 1$ μA).

verified with the measurements shown in Figure 2.32. The performance of the redundant decoder is improved with an increased sum current ($V_{\text{dd}} = 5$ V, $I_{\text{sum}}^{\text{pad}} = 30$ μA), leaving the transistors to operate in strong inversion. By moving away from weak inversion, the transistor mismatch decreases (see measurements in [12] and Figures 2.53) and the error rate curves lie closer together.

The decoder's optimal performance was estimated by adjusting the sum current from $I_{\text{sum}}^{\text{pad}} = 1$ μA to $I_{\text{sum}}^{\text{pad}} = 100$ μA and measuring the error rate curves for the AWGN channel. The results obtained from 30 chips are presented in Figure 2.33. Up to $I_{\text{sum}}^{\text{pad}} = 30$ μA, the error rate keeps getting smaller (i.e., the performance improves), for $I_{\text{sum}}^{\text{pad}} = 100$ μA the performance is worse than for the decoder's performance with $I_{\text{sum}}^{\text{pad}} = 1$ μA. Therefore, the redundant decoder performs best with a sum current of approximately $I_{\text{sum}}^{\text{pad}} = 30$ μA.

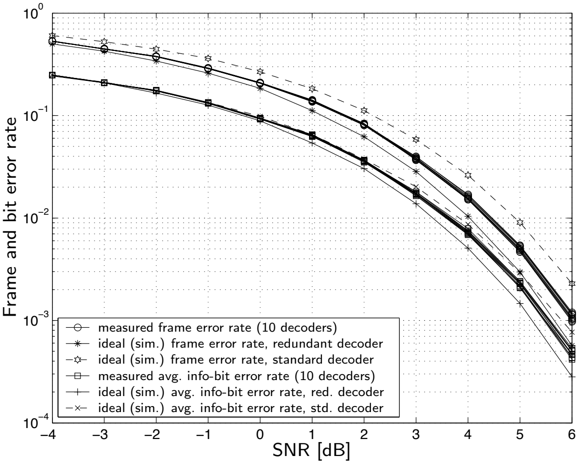


Figure 2.32: Measured and simulated error rate curves of 10 versions (each with a different set of second-generation softgates) of the redundant decoder of Figure 2.30 in strong inversion ($I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu$A). channel.

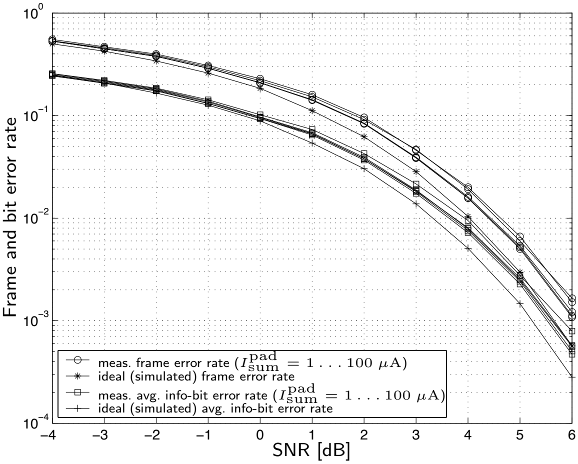


Figure 2.33: Measured and simulated error rate curves of the redundant decoder of Figure 2.30 with varying sum current ($I_{\mathrm{sum}}^{\mathrm{pad}} = 1, 3, 10, 30, 100\ \mu$A) for the AWGN channel. Best performance for $I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu$A.

Conclusions

A summary of the second-generation softgates is given in Table 2.2.

Technology:	0.8 μm CMOS process (Philips C175SC), semi-custom mixed-signal array "MD300"
Die size:	1.44×2.4 mm^2 (1 tile[a])
I/O-stage nMOS size ($W \times L$):	24×6.0 μm $\times$ 5.6 μm
I/O-stage pMOS size ($W \times L$):	24×6.0 μm $\times$ 5.6 μm
Reset nMOS size ($W \times L$):	32×6.0 μm $\times$ 5.6 μm
Internal nMOS size ($W \times L$):	4×6.0 μm $\times$ 5.6 μm
Internal pMOS size ($W \times L$):	4×6.0 μm $\times$ 5.6 μm
Supply voltage (nom./min.):	5 V/2.7 V
Sum current ($I_{\mathrm{sum}}/I_{\mathrm{sum}}^{\mathrm{pad}}$)	$\frac{1}{6}$ μA/1 μA
Pin-count:	16
Features:	Simplified circuit without reference voltages Output scaling Input stage with reset-transistors

---

[a] Only a small part of the area was used; the major part of each tile contained digital logic circuits that were left unused.

Table 2.2: Summary of the second-generation softgates.

First, statistical measurements on softgate level were presented. While the second-generation softEQU gate's characteristics resembles the one from the first-generation, a clear difference can be observed from the first-generation to the second-generation softXOR gate: the second-generation softXOR gate is systematically overconfident (i.e., the entropy of the output probabilities is higher than ideal) while the characteristics of the first-generation softXOR gate deviates around the ideal (expected) curves.

Bit error rates and frame error rates for two realizations—a standard as well as a redundant realization—of an (8,4,4) extended Hamming decoder were presented. The decoders were tested under various operating conditions (e.g., varying supply voltage, varying sum currents). While the standard realization's performance deteriorates for higher sum currents, the redundant realization's performance improves up to $I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$. Increasing the sum current has positive effects on the transistor mismatch behavior.

Despite the mismatch that was observed on softgate level, the small current levels and the poor—surely suboptimal—wiring on breadboard level, the decoders show a rather robust behavior and function under different operating conditions. While the characteristics of the first-generation and second-generation softgates resemble each other strongly on softgate level, the error rate curves of the corresponding decoders show deviant behavior.

By adding redundancy and therefore increasing the hardware complexity and the power consumption, the decoder's performance can be improved.

2.3 The (8,4,4) Extended Hamming Decoder Chip

In this section we report on a fully integrated decoder chip for the (8,4,4) extended Hamming code. The factor graph of Figure 2.1 is implemented. Since this factor graph has cycles, the standard sum-product algorithm operating on this graph (and thus also our analog decoder) is not a maximum-likelihood decoder.

The decoder was implemented in an IBM 0.25 μm BiCMOS process (IBM BiCMOS 6HP) with 6 metal layers [51]; only CMOS transistors were used.

The implemented circuit consists of 5 types of sub-blocks:

- Current scaling I/O-circuits
- Biasing circuits
- SoftXOR gates (parity check nodes)
- SoftEQU gates (equality constraint nodes)
- Test structures: nMOS and pMOS current mirrors accessible on pad level

In order to have the same interface as the breadboard level analog decoder described in Section 2.2, the sum current of the chip was designed to be $I_{\mathrm{sum}}^{\mathrm{pad}} = 1\ \mu\mathrm{A}$. However, in order to keep the transistors operating in the subthreshold region, the softgate sub-circuits need to operate with a smaller sum current of $I_{\mathrm{sum}} = 100$ nA only. Therefore, the current scaling sub-circuits were designed to scale the outer currents by a current mirror with ratio 10:1.

The sum current can be adjusted by an external bias current. On chip, this bias current is scaled by a ratio 10:1 and multiple copies are distributed to the different softgate circuits.

The structure of the softgates is basically the same as the one described in Section 2.2.1; additionally, these softgates contain reset transistors for resetting the input probabilities to $P(0) = P(1) = 0.5$.

For mismatch measurements two current mirrors were included, one with nMOS transistors and one with pMOS transistors, both of the same size as the transistors used in the softgates.

The size of all the nMOS transistors (except the reset transistors) is $L_{\mathrm{n}} = 1\ \mu\mathrm{m}$, $W_{\mathrm{n}} = 10\ \mu\mathrm{m}$; the size of the pMOS transistors is $L_{\mathrm{p}} = 1\ \mu\mathrm{m}$, $W_{\mathrm{p}} = 30\ \mu\mathrm{m}$. The inversion coefficient IC [13] for these transistors at a current level of 100 nA is

$$IC_{\mathrm{n}} = \frac{I_{\mathrm{D}}}{2n\beta {U_{\mathrm{T}}}^2} = 0.04 \tag{2.13}$$

$$IC_{\mathrm{p}} = \frac{I_{\mathrm{D}}}{2n\beta {U_{\mathrm{T}}}^2} = 0.06, \tag{2.14}$$

i.e., well in the subthreshold region. All the known techniques to minimize transistor mismatch were applied [32].

The chip layout—annotated with the structure of the decoder—is shown in Figure 2.34. The correspondence to the factor-graph of Figure 2.1 is apparent.

The chip is pad limited, i.e., the area of the complete chip ($1.984\ \mathrm{mm} \times 1.984\ \mathrm{mm} \approx 3.9\ \mathrm{mm}^2$) is much larger than the core of the chip, which measures only $0.7\ \mathrm{mm}^2$. The 16 sub-circuits located in the middle of the chip are the softgate circuits. The 8 I/O scaling circuits, denoted as "interface circuitry" in Figure 2.34, are placed to the left and right of the softgates.

The chip was designed to operate at a supply voltage of 1.8 V. With

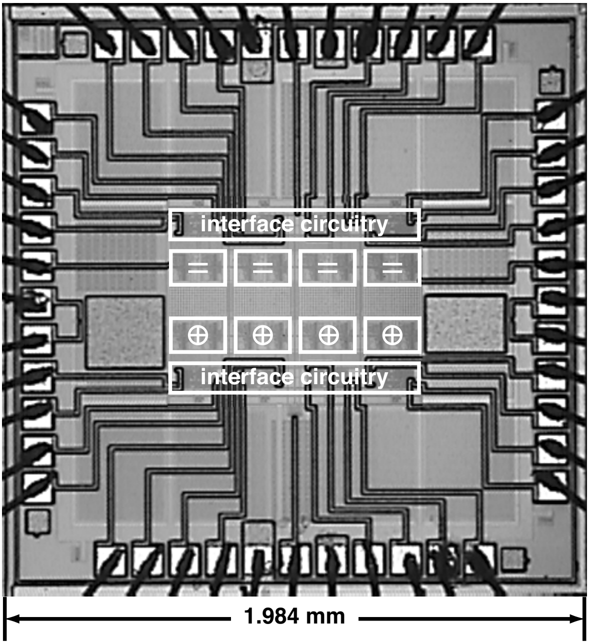


Figure 2.34: Layout of the Hamming decoder chip.

this supply voltage the average power consumption lies below 50 μW, which is smaller than the leakage power of an alkaline battery. By lowering the supply voltage, the power consumption can be further reduced.

The Hamming decoder was designed in a joint effort by Patrick Merkli (interface circuitry) and Matthias Frey (decoder core). The error rate measurements were carried out by Patrik Strebel, the transistor-mismatch measurements by Patrick Merkli and Matthias Frey.

2.3.1 Measurements

For the measurements presented in this section we used the same measuring device as for the measurements shown in Section 2.2. The device contains 16 12-bit D/A- and A/D-converters to set the input currents and to measure the output currents.

Transistor Mismatch

In order to get an idea of the transistor mismatch on the chip, the characteristics of the two current mirrors, as shown in Figure 2.35, were measured. The mismatch was characterized by the standard deviation of the normalized current differences δ_I, defined as

$$\delta_I = \frac{I_{\mathrm{D1}} - I_{\mathrm{D2}}}{I_{\mathrm{D1}}}. \tag{2.15}$$

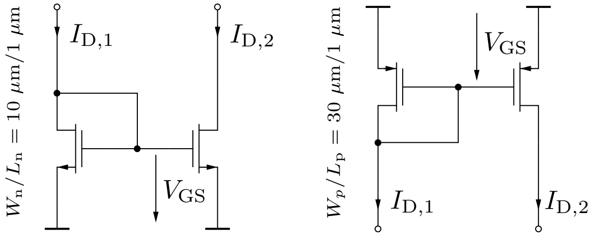
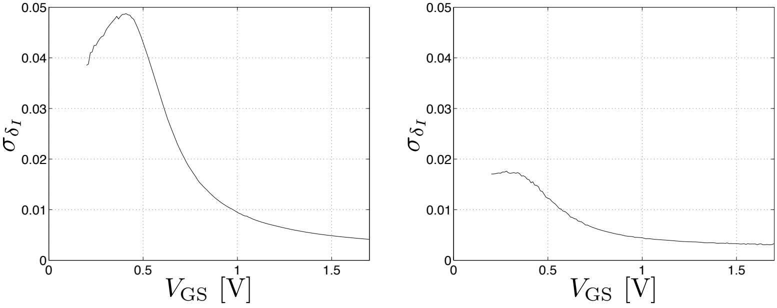


Figure 2.35: nMOS and pMOS current mirrors used for measuring transistor mismatch.

For various gate-source voltages ($\mathrm{V_{GS}}$) (for the nMOS as well as for the pMOS current mirrors) the standard deviations of the current differences were measured. The results are shown in Figure 2.36. The standard deviation is smaller for the pMOS current mirrors due to their larger size. We are operating the transistors in the subthreshold region ($I_{\mathrm{D}} = 100$ nA corresponds to a gate-source voltage of $V_{\mathrm{GS}} \approx 0.45$ V) where, unfortunately, current mismatch is maximal. The measurements are consistent with measurements in [12].

Error Rate Measurements

The following figures show the frame error rate and the average information-bit error rate, both measured for a variety of operating conditions and from simulating an ideal decoder without mismatch. The decoder



(a) nMOS current mirror **(b)** pMOS current mirror

Figure 2.36: Standard deviation for the current differences δ_I of the nMOS and the pMOS current mirrors.

was operated with a fixed assumed SNR of $\mathrm{SNR}_{\mathrm{assumed}} = 4\mathrm{dB}$, independently of the actual SNR.

10 decoder-chip samples were measured; the results are presented in Figure 2.37. The curves vary only little and lie all above the curves of the simulated ideal decoder. Other than expected, the influence of transistor mismatch, which was shown in Figure 2.36, does not seem to greatly influence decoder performance.

The decoder's performance depends on the decoding time; the performance saturates for decoding times greater than 10 ms and deteriorates substantially for decoding times smaller than 1 ms. Figure 2.38 shows error rate curves for different decoding times, $T_{\mathrm{settling}} = 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100$ ms.

By raising the sum current, we hope to increase the speed of the decoder and lessen the effect of transistor mismatch. Figure 2.39 shows the result for a varying sum current $I_{\mathrm{sum}}^{\mathrm{pad}} = 1\ \mu\mathrm{A} \ldots 100\ \mu\mathrm{A}$. The error correcting capability is approximately the same for currents below $I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu\mathrm{A}$ and diminishes for higher currents.

With the increased sum current, the decoding time can be decreased without increasing the implementation loss. This is illustrated with the

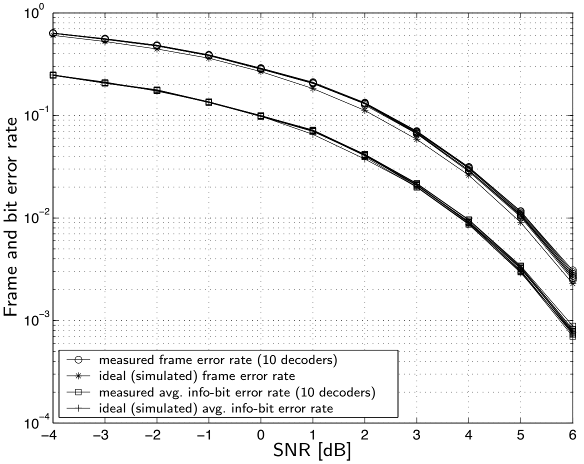


Figure 2.37: Measured and simulated error rate curves of 10 Hamming decoder chips in weak inversion ($I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$) for the AWGN channel.

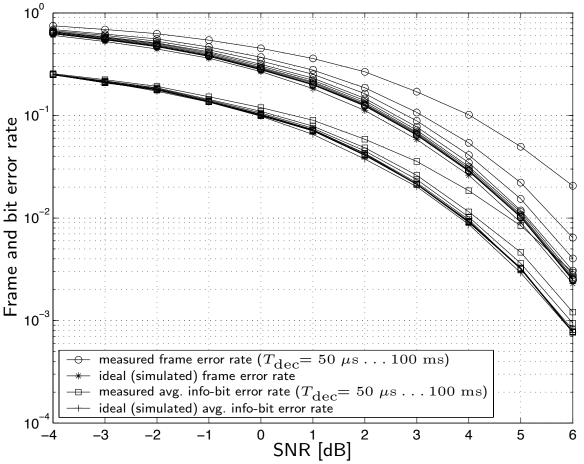


Figure 2.38: Measured BER curves for different decoding times: $T_{\text{dec}} = 50\ \mu\text{s} \ldots 100$ ms (from worst to best performance) and a sum current of 1 μA.

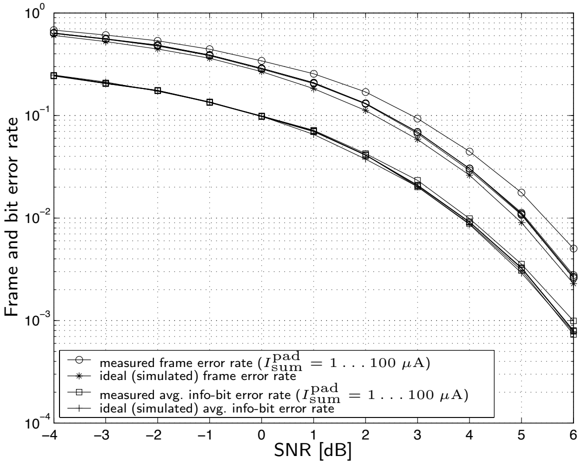


Figure 2.39: Measured error rate curves for various sum currents: $I_{\text{sum}}^{\text{pad}} = 1, 3, 10, 30, 100\ \mu$A for the AWGN channel. Also shown are ideal simulations of the error rates for such a decoder. Best performance measured for $I_{\text{sum}}^{\text{pad}} = 30\ \mu$A.

measurement curves presented in Figure 2.40. For these measurements, the sum current was set to 30 μA and the decoding times were varied from 50 μs up to 100 ms ($T_{\text{dec}} = 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\,\text{ms}$). No change in performance could be observed for decoding times above 1 ms.

The increased sum current also has a positive effect on the scattering of the error rate curves; with the better transistor mismatch behavior, the decoder's performance seems to stay more constant over different chips, resulting in all error rate curves lying on top of each other as shown in Figure 2.41.

The decoder chip was designed for a supply voltage of $V_{\text{dd}} = 1.8\ V$. However, without any loss of performance, the supply voltage can be lowered to $V_{\text{dd}} = 0.7\ V$ as illustrated in Figure 2.42. Note that the power consumption decreases linearly with the supply voltage.

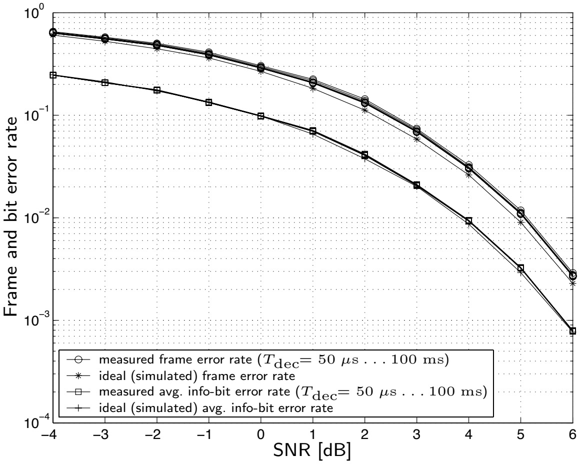


Figure 2.40: Measured and simulated error rate curves for the Hamming decoder operated in strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu$A) for various decoding times: $T_{\text{dec}} = 50\ \mu\text{s} \ldots 100$ ms for the AWGN channel.

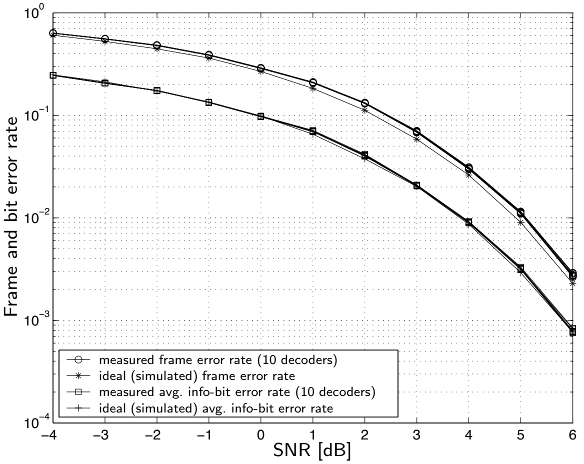


Figure 2.41: Measured error rate curves for 10 Hamming Decoder Chips operated in strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu$A) for the AWGN channel. Also shown are ideal error rate curves for such a decoder.

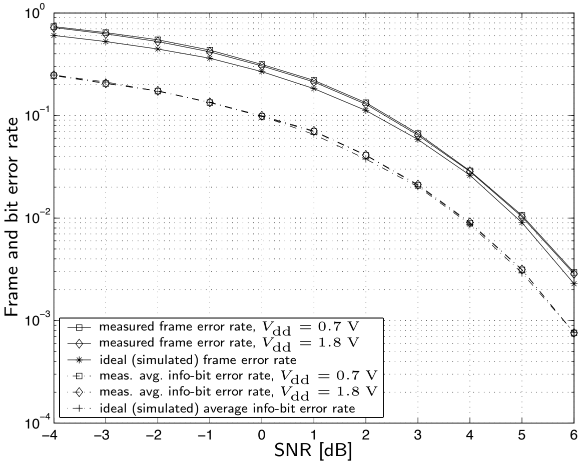


Figure 2.42: Measured error rate curves of the Hamming decoder chip for low ($V_{\mathrm{dd}} = 0.7$ V) and for nominal ($V_{\mathrm{dd}} = 1.8$ V) supply voltage for the AWGN channel. Also shown are ideal error rate curves for such a decoder.

Transient Measurements

In order to investigate the settling behavior of the decoder, transients of the output currents were measured.

Figure 2.43 shows the time response of the 16 output currents representing the output probabilities for the decoder in weak inversion (left: $I_{\mathrm{sum}}^{\mathrm{pad}} = 1\ \mu\mathrm{A}$) and in strong inversion (right: $I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu\mathrm{A}$), respectively. The inputs were switched from reset to the probability vector [0.1, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9]. The i^{th} element of this vector represents the probability that for the i^{th} bit a "1" was transmitted, based on the corresponding channel output symbol y_i, i.e., element "i" corresponds to $\gamma \cdot p(y_i|X_i = 1)$ with $\gamma = 1/\big(p(y_i|X_i = 0) + p(y_i|X_i = 1)\big)$.

It can be observed in Figure 2.43 that the decision on each bit can be made after 25 μs (left: weak inversion) and after less than 10 μs (right: strong inversion), as the error in the 1st bit is then already corrected.

After having observed the difference of the error rate curves for a decoding time of $T_{\mathrm{dec}} = 10$ ms and $T_{\mathrm{dec}} = 1$ ms (see Figures 2.38 and

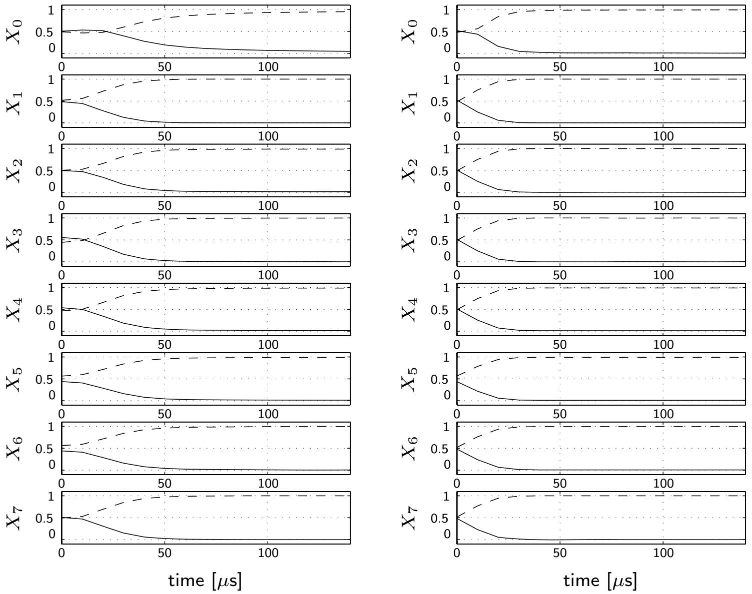


Figure 2.43: The 2×16 currents representing the output probabilities; dashed line: current proportional to $p(y_i|x_i = 1)$, solid line: current proportional to $p(y_i|x_i = 0)$. Codeword with 1 bit-error coming from the reset state; the input probabilities have following values: $[p(y_i|x_i = 1)]_{i=0}^{7} = [0.1, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9]$; left: decoder in weak inversion, right: decoder in strong inversion

2.40), we identified words that were decoded correctly with the longer decoding time, but failed to be decoded correctly with the shorter one. One example is presented in Figure 2.44. Again, the measurement starts when the decoder is released out of the reset-state. The input likelihood vector is as follows: [0.716, 0.000, 0.726, 0.999, 0.064, 0.001, 0.021, 0.999]. Figure 2.44 shows that for the decoder operated in weak inversion (left: $I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$) even after 5 ms a steady state has not been reached, and the decision (in bit X_4) is still not stable. For the decoder operated with a higher sum current (right: $I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$), the bit X_4 is corrected after approximately 0.5 ms.

These two examples illustrate the difficulty of measuring the analog decoder's speed by looking at single decoding operations. In order to get a reliable performance measure, error rate curves need to be measured for several operating conditions with different decoding times. We therefore prefer to apply Definition 2.8 to determine the speed of a decoder, rather than doing transient measurements.

All these transient measurements were conducted with a sum current of either $I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$ or $I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$. The supply voltage was set to the nominal $V_{\text{dd}} = 1.8$ V. We demonstrated with the measurements shown in Figure 2.42 that the decoder operates properly with a supply voltage as low as $V_{\text{dd}} = 0.7$ V. It can be seen in the left plots of Figure 2.45 that the output scaling does not work correctly (see current I_{X_0}) for a supply voltage of $V_{\text{dd}} = 0.7$ V: the output currents I_{X_0} do not always add up to the sum current $I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$.

The plots on the right hand side of Figure 2.45 point out that the output probabilities (computed from the output currents as $P(X_i = 0) = I_{X_i}(0)/(I_{X_i}(0) + I_{X_i}(1))$, $P(X_i = 1) = I_{X_i}(1)/(I_{X_i}(0) + I_{X_i}(1))$) are still correct, despite the fact that output scaling does not work properly.

2.3.2 Conclusions

An analog decoder chip for the (8,4,4) Hamming code and a vast collection of measurements thereof were presented. The decoder chip operates in the subthreshold region and consumes very little power, but the decoding speed is rather slow. Most codewords can be decoded correctly after a few microseconds, but some configurations have a much larger settling time in the millisecond-range. The bit rate of the decoder depends on

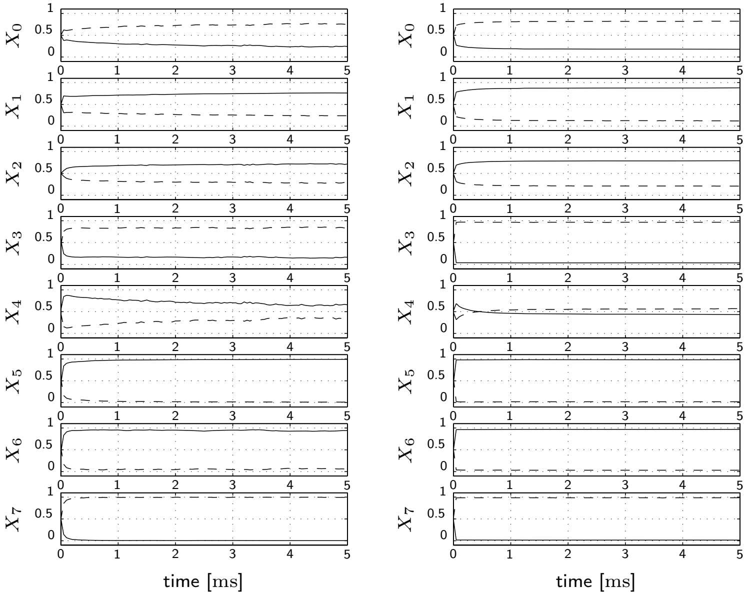


Figure 2.44: The 16 currents representing the output probabilities; dashed line: current representing $P(X_i = 1)$, solid line: current representing $P(X_i = 0)$. The input probabilities have following values: $[P(X_i = 1)]_{i=0}^{7} = [0.716, 0.000, 0.726, 0.999, 0.064, 0.001, 0.021, 0.999]$; left: decoder in weak inversion, right: decoder in strong inversion

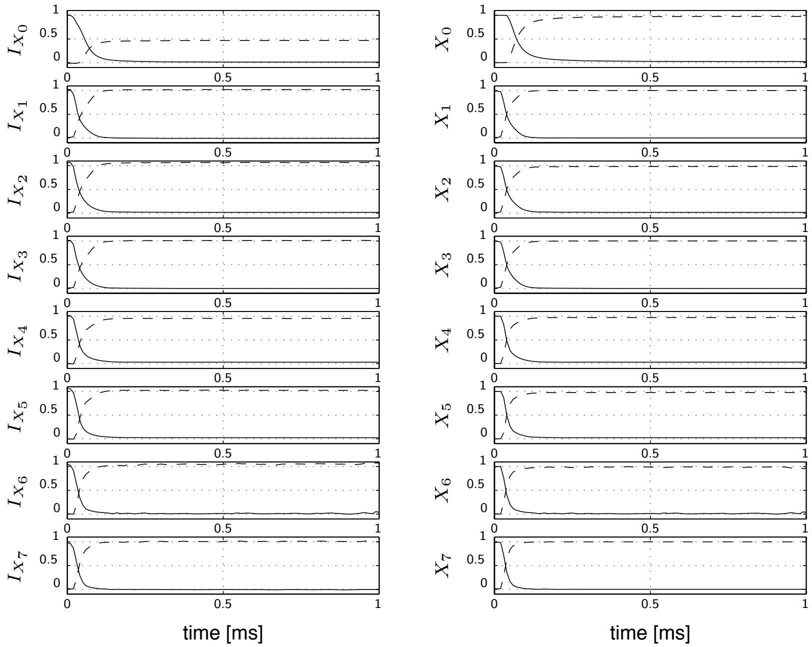


Figure 2.45: The 16 currents (left, in μA) and the represented output probabilities (right) for $V_{\mathrm{dd}} = 0.7$ V; dashed line: current representing $P(X_i = 1)$, solid line: current representing $P(X_i = 0)$. Codeword with 1 bit-error; the input probabilities are switched from $[P(X_i = 1)]_{i=0}^{7} = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$ to $[P(X_i = 1)]_{i=0}^{7} = [0.1, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9]$.

Technology:	0.25 μm BiCMOS process (IBM 6HP) only CMOS transistors were used
Die size:	1.984×1.984 mm^2 (pad-limited)
Active area:	0.5×0.7 mm^2
Pin-count:	44
nMOS size ($W \times L$):	10 μm $\times$ 1 μm
pMOS size ($W \times L$):	30 μm $\times$ 1 μm
Supply voltage (nom./min.):	1.8 V/0.7 V
Sum current ($I_{\mathrm{sum}}/I_{\mathrm{sum}}^{\mathrm{pad}}$)	100 nA/1 μA (WI) 3 μA/30 μA (SI)
T_{settling}	10 ms (WI) 1 ms (SI)
P_{tot} (decoder core and interface[a])	55 μW (WI, measured[b]) 1650 μW (SI, measured[c])
Energy per decoded info bit:	140 nJ (WI, measured[b]) 420 nJ (SI, measured[c])
Implementation loss	min. ... max. (10 samples)
@ FER $= 10^{-2}$, WI:	0.11 dB ... 0.20 dB
@ FER $= 10^{-2}$, SI:	0.15 dB ... 0.20 dB
@ FER $= 10^{-2.5}$, WI:	0.06 dB ... 0.22 dB
@ FER $= 10^{-2.5}$, SI:	0.10 dB ... 0.17 dB
Features:	Parallel input and output stage with 10:1 scaling Reset-transistors at the input

---

[a] Simulations show that the decoder core consumes about 50% of the total power.

[b] Measured in weak inversion with $V_{\mathrm{dd}} = 1.8$ V, $T_{\mathrm{dec}} = 10$ ms, $I_{\mathrm{sum}}^{\mathrm{pad}} = 1$ μA.

[c] Measured in strong inversion with $V_{\mathrm{dd}} = 1.8$ V, $T_{\mathrm{dec}} = 1$ ms, $I_{\mathrm{sum}}^{\mathrm{pad}} = 30$ μA.

Table 2.3: Summary of the analog (8,4,4) extended Hamming decoder ASIC.

different parameters—on the sum current as well as on the desired performance. The decoder's performance depends on the decoding time and the sum current. Mismatch was observed on the transistor level but has only little effect on the performance of the decoder. The specifications of the Hamming decoder are summarized in Table 2.3.

2.4 (16,5,8) Reed-Muller Decoder Chip

The second ASIC of this chapter implements the factor graph of Figure 2.46 (due to Forney [17]).

The factor graph's cycles could be eliminated by considering the super nodes A, B, and C (indicated by the dashed boxes in Figure 2.46) and by clustering the variables between these super nodes. The bit-wise MAP estimation is attained by applying the sum-product algorithm to the resulting cycle-free graph and marginalizing the obtained output distributions.

However, we chose to implement a suboptimal algorithm that uses overlapping double-soft-bit messages as illustrated in Figure 2.46. The resulting algorithm (which is vaguely related to "generalized belief propagation" [67]) is iterative with some very short feedback loops; for details see [48].

The stability of such non-standard algorithms is not obvious; indeed, digital (discrete-time) versions of such algorithms usually require heavy damping to work at all. Fortunately, the corresponding analog decoder does not seem to have any problems with stability.

The implemented circuit consists of two main sub-blocks: a serial-to-parallel input interface with analog storage elements and the analog decoder core. The analog decoder core is based on the circuits introduced in [42] and is further described in [48]. The input interface is characterized in the following subsection.

The Reed-Muller decoder was jointly designed by Patrick Merkli (decoder core) and Matthias Frey (serial-to-parallel interface). All the measurements were conducted by Patrik Strebel.

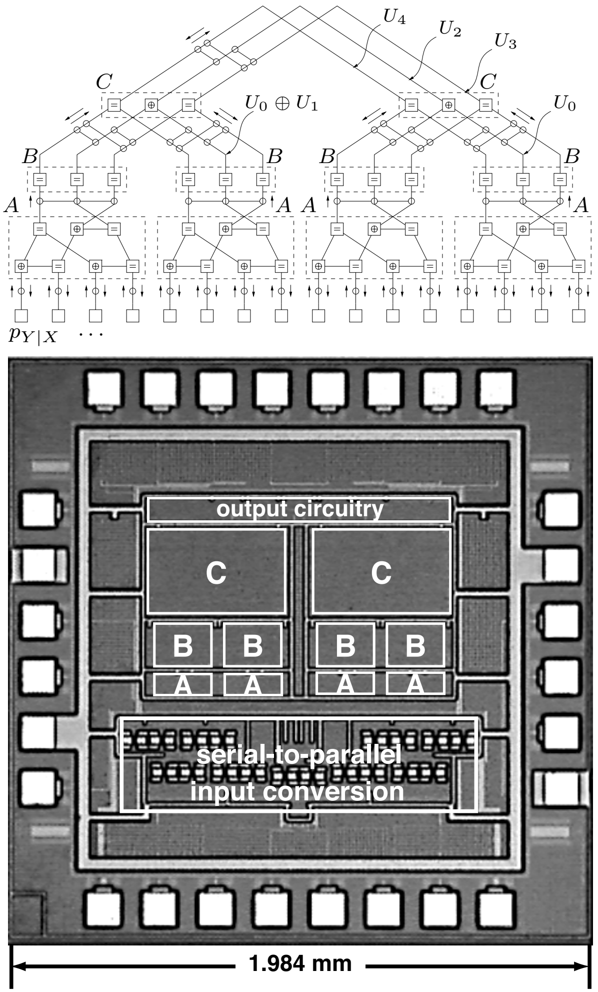


Figure 2.46: Factor graph of the Reed-Muller decoder and the corresponding chip.

2.4.1 The Serial-to-Parallel Input Interface

The input interface differs from previously implemented interfaces of analog decoders. The interface accepts the input messages serially as currents (for this decoder as four currents representing the joint probability densities of two input bits) and stores these currents in parallel by means of the circuit whose schematic is depicted in Figure 2.47. The scaled serial input currents are stored with a simple circuit shown in Figure 2.48; this is a current-mirror configuration where the gate source voltage can be stored on a capacitor. The output resistance of the current mirror is enhanced with a cascode transistor biased with a voltage V_{casc}. 4 circuits of Figure 2.48 can be found in each "current memory & rescaling"-block of Figure 2.47.

In a subsequent stage the output currents I_{out} are scaled to the desired sum current with a pMOS scaling circuit.

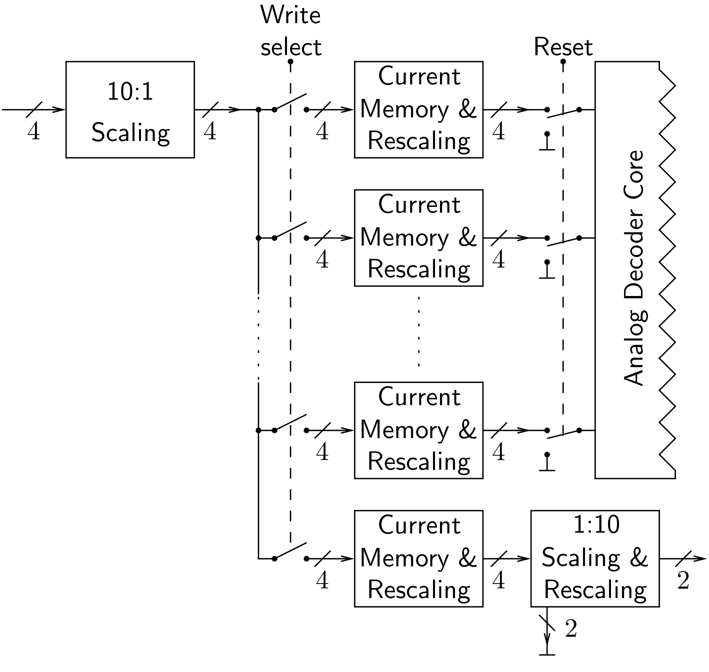


Figure 2.47: Schematic of the complete input interface including one memory stage with 2 outputs for measurements.

The gate-leakage currents are usually rather poorly modeled and simulations thereof cannot be fully trusted. It is therefore difficult to deter-

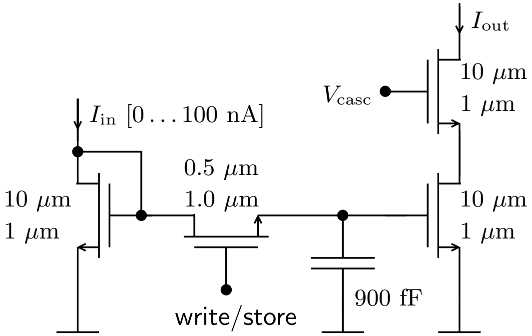


Figure 2.48: One slice of the input circuit. The numbers next to the transistors denote their widths and their lengths.

mine the minimal size of the storage capacitors so that the charge is kept stable long enough for the decoder core to finish decoding the input. To be on the safe side, the storage capacitor was dimensioned rather large with a length and width of $L_{\text{storage}} = W_{\text{storage}} = 30\ \mu\text{m}$ leading to a capacitance of $C_{\text{storage}} = 900$ fF. I/O-transistors with a thicker gate-oxide show less leakage than the standard transistors do. Therefore, such an I/O-transistor was used for the "write/store"-switch.

The layout of four current-storage cells (making up a "current memory & rescaling"-block of Figure 2.47) is depicted in Figure 2.49. It is clear that the size of this circuit is mainly determined by the size of the MIM-capacitors (Metal-Insulator-Metal capacitor), the current-mirror transistors and the switching transistors lie in-between the capacitors, the pMOS current scaler is stacked on top of the storage circuit.

In order to test the storage capability of the implemented current memory, one extra memory cell was implemented, and two (out of four) current outputs were rescaled and led to two output pins of the chip for measurement; the other two remaining currents were dissipated.

The memory cells were measured with 3 different sets of input probabilities:

- $p(0) = 0.01$, $p(1) = 0.99$, $p(2) = 0.50$, $p(3) = 0.50$
  ($\log_{10}[p(1)/p(0)] \approx 2.00$)

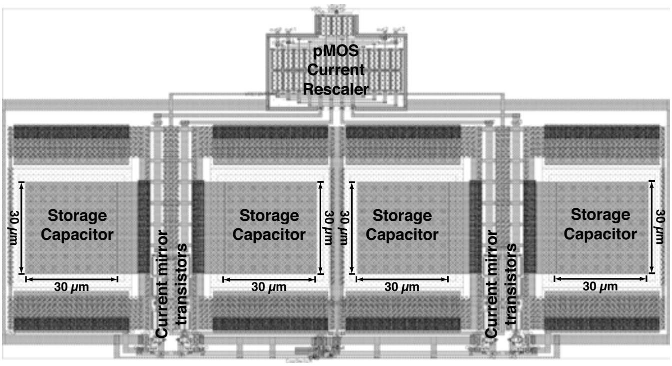
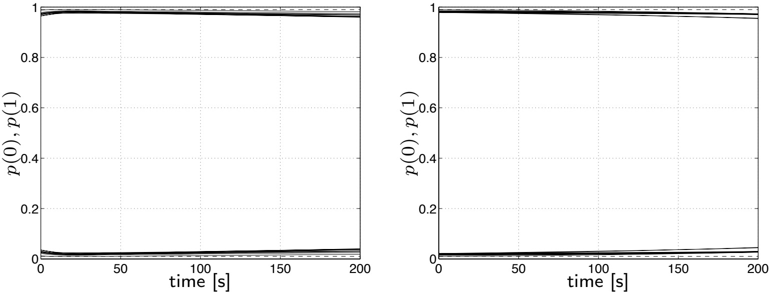


Figure 2.49: The layout of one slice of the input circuit. The size is mainly determined through the 4 storage capacitors.

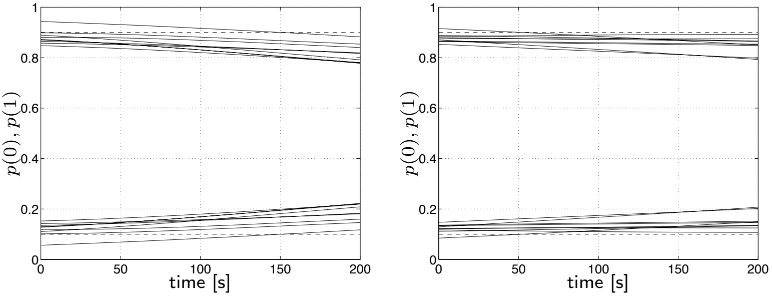
- $p(0) = 0.10$, $p(1) = 0.90$, $p(2) = 0.50$, $p(3) = 0.50$ ($\log_{10}[p(1)/p(0)] \approx 0.95$)
- $p(0) = 0.50$, $p(1) = 0.50$, $p(2) = 0.50$, $p(3) = 0.50$ ($\log_{10}[p(1)/p(0)] = 0$)

The memory cell measurements obtained from 10 decoder chips are presented in Figure 2.50; as one of the decoder chip's test-circuit was defective, the corresponding curves were omitted and only the result of 9 chips are actually shown. Measurements for the decoders operating both in weak inversion (left) as well as in strong inversion (right) can be seen in Figure 2.50.

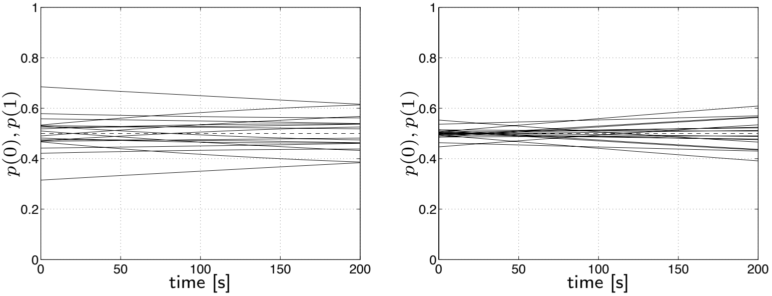
It should be noted that the time-axis of the plots (200 s) is much larger than what is needed for the chip to decode an input word (millisecond-range). The plots demonstrate the storage-capability of the circuit. It can, however, also be seen that the accuracy of such a circuit is limited; the curves in Figure 2.50 are scattered around the ideal dashed lines. This becomes even more apparent for the same measurements presented in logarithmic scale in Figure 2.51. It is unclear what the cause of this inaccuracy is—either the memory cell, the subsequent pMOS rescalers, the 1:10-scaler, or their combination cause this imprecision. Although these curves reveal the imprecision of the input stages, the following error rate measurements prove the functionality of the whole decoder.



(a) $p(1) = 0.99$, weak inversion **(b)** $p(1) = 0.99$, strong inversion

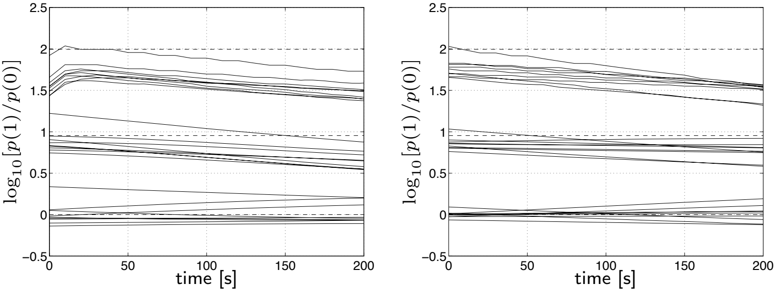


(c) $p(1) = 0.90$, weak inversion **(d)** $p(1) = 0.90$, strong inversion



(e) $p(1) = 0.50$, weak inversion **(f)** $p(1) = 0.50$, strong inversion

Figure 2.50: Measurements demonstrating the storage capability of the current memory for different stored values. Straight lines: measurement results; dashed lines: ideal (stored) values.



(a) Weak inversion **(b)** Strong inversion

Figure 2.51: Storage capability of the current memory cell for LLR = $0, 1, 2$. The (ideal) stored value is given as the dashed line for reference.

2.4.2 Error Rate Measurements

All the following measurements were obtained from the same measuring device that was also used for the measurements shown in the previous chapters. The device contains 16 12-bit D/A- and A/D-converters to set the input currents and to measure the output currents.

For the Reed-Muller decoder the bit error rate (BER) and the frame error rate (FER) of 10 different chips under nominal operating conditions, as well as with an increased sum current, are shown in Figure 2.52 and Figure 2.53, respectively. (We remark that the nominal operating conditions were determined before testing the chip; the best performance is not achieved with these conditions as is demonstrated in Figure 2.53 where the chip is tested with an increased current level.) For the Reed-Muller decoder the plotted bit error rate is the error rate of the last information bit U_4, denoted in Figure 2.46. The frame error rate includes errors in all the information bits, $U_1 \dots U_4$. Also shown in Figures 2.52 and 2.53 is the performance of the algorithms obtained by ideal (discrete-time) simulations. It is obvious from these figures, that the effect of transistor mismatch is small, but visible.

By increasing the sum current, the transistors move away from the weak inversion with its exponential characteristic. The exponential behavior of the transistors is necessary for obtaining the correct circuit functionality. However, by increasing the current-level, transistor mismatch is reduced. These two opposing effects lead to a trade-off between high current levels (decreased mismatch) and low current levels (better exponential behavior).

For the (16,5,8) Reed-Muller decoder chip, the higher current level results in a better performance as is shown in Figure 2.53.

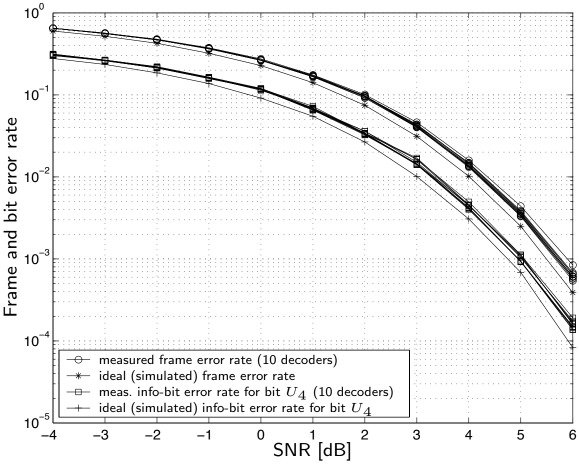


Figure 2.52: Measured error rate curves of 10 Reed-Muller decoder chips operating in weak inversion ($I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$) for the AWGN channel. Also shown are ideal (simulated) curves for such a decoder.

The curves shown in the following figures allow to estimate the speed of the Reed-Muller decoder. Figures 2.54 and 2.55 show the error rates of the decoder operating in weak and in strong inversion, with decoding times (see Definition 2.7) varying from 10 μs to 50 ms, and from 10 μs to 50 ms.

For nominal conditions, the Reed-Muller decoder's performance saturates for $T_{\text{dec}} > 1$ ms and it deteriorates for $T_{\text{dec}} < 100\ \mu$s.

The speed of the analog decoders can be substantially augmented by increasing the sum current, i.e., by operating the transistors in moderate

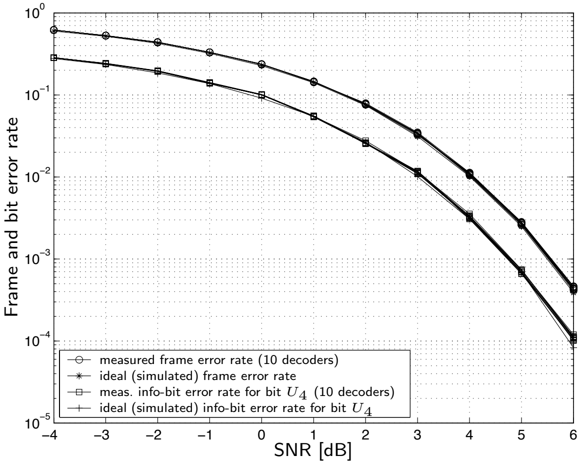


Figure 2.53: Measured error rate curves of 10 Reed-Muller decoder chips operating in strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$) for the AWGN channel. Also shown are ideal (simulated) curves for such a decoder.

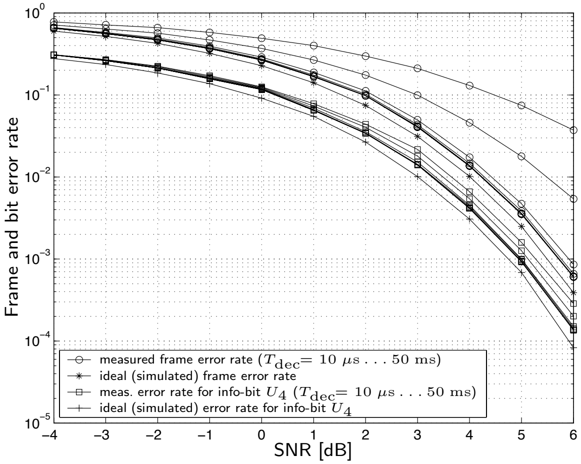


Figure 2.54: Measured and simulated error rate curves of the Reed-Muller decoder chip for various decoding times: $T_{\text{dec}} = 10, 20, 50, 100, 200, 500\ \mu\text{s}, 1, 2, 5, 10, 20, 50\ \text{ms}$, $I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$, best performance for $T_{\text{dec}} > 100\ \mu\text{s}$.

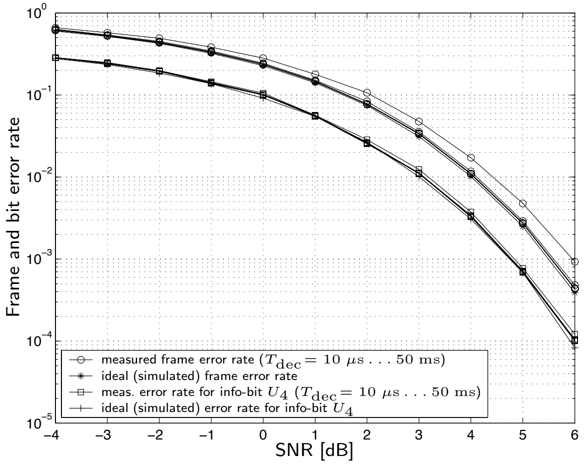


Figure 2.55: Measured and simulated error rate curves of the Reed-Muller decoder chip for various decoding times: $T_{\mathrm{dec}} = 10, 20, 50, 100, 200, 500\ \ \mu\mathrm{s}, 1, 2, 5, 10, 20, 50\ \ \mathrm{ms}$, $I_{\mathrm{sum}}^{\mathrm{pad}} = 30\ \mu\mathrm{A}$. Best performance for $T_{\mathrm{dec}} \geq 20\ \mu\mathrm{s} = T_{\mathrm{settling}}$.

inversion instead of in weak inversion. Figure 2.55 shows the frame and bit error rate curves for the Reed-Muller decoder chip: only the curves for a decoding time of less than $T_{\mathrm{dec}} = 20\ \mu\mathrm{s}$ indicate a slight degradation.

Although the decoder was designed for $V_{\mathrm{dd}} = 1.8$ V, the measured bit error rates do not change if the supply voltage is reduced to 0.9 V. This is shown in Figure 2.56.

2.4.3 Conclusions

A new decoder in CMOS technology was presented. This decoder incorporates an experimental version of an extension of the sum-product algorithm with “overlapping” multi-variable messages and very short feedback loops, as well as a novel serial interface. Extensive measurement results were presented: the decoder chip is fully operational for supply voltages as low as $V_{\mathrm{dd}} = 0.9$ V.

It should be emphasized that the chips show better performance

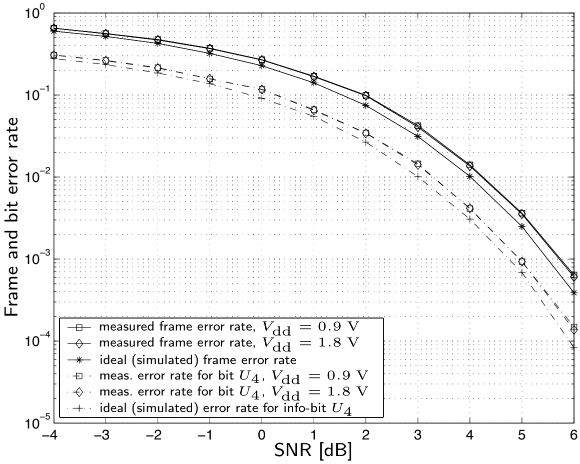


Figure 2.56: Measured and ideal (simulated) error rate curves of the Reed-Muller decoder chip operating in weak inversion ($I_{\text{sum}}^{\text{pad}} = 1\ \mu$A) for the nominal and minimal supply voltages $V_{\text{dd}} = 1.8$ V and $V_{\text{dd}} = 0.9$ V, respectively.

(shorter decoding time, lower error rates) by increasing the unit current and moving away from the nominal conditions.

Some key data of the Reed-Muller decoder chip is summarized in Table 2.4. The power and energy consumption numbers were obtained while the chip was measured under nominal conditions.

2.5 An Analog Coprocessor for Decoding a Turbo Code

Several decoders for error correcting codes have been presented so far, however, only few were suitable for "real-world applications", for decoding modern codes, e.g., turbo codes [3, 60] or LDPC codes. Decoders for such codes can, in principle, be implemented as large non-linear analog circuits, which can be much faster or consume less power than the conventional digital implementations. However, such decoders are not very versatile, i.e., they cannot be used for various block lengths, as is

Technology:	0.18 μm BiCMOS process (IBM 7HP) only CMOS transistors were used
Die size:	1.984×1.984 mm^2
Active area:	1×0.65 mm^2
Pin-count:	28
nMOS size ($W \times L$):	10 μm $\times$ 1 μm
pMOS size ($W \times L$):	30 μm $\times$ 1 μm
Supply voltage (nom./min.):	1.8 V/0.9 V
Sum current ($I_{\text{sum}}/I_{\text{sum}}^{\text{pad}}$)	100 nA/1 μA (WI)
	3 μA/30 μA (SI)
T_{settling}	1 ms (WI)
	20 μs (SI)
P_{tot} (decoder core and interface[a])	55 μW (WI[b], measured)
	1650 μW (SI[c], measured)
Energy per decoded info bit:	7 nJ (WI[b], measured)
	4 nJ (SI[c], measured)
Implementation loss	min. … max. (10 samples)
@ FER $= 10^{-2}$, WI:	0.18 dB … 0.36 dB
@ FER $= 10^{-2}$, SI:	0.03 dB … 0.09 dB
@ FER $= 10^{-2.5}$, WI:	0.19 dB … 0.37 dB
@ FER $= 10^{-2.5}$, SI:	0.02 dB … 0.08 dB
Features:	Serial input stage with 10:1 scaling
	Non-standard sum-product algorithm inside

---

[a] Simulations show that the decoder core consumes about 50% of the total power.

[b] Measured in weak inversion with $V_{\text{dd}} = 1.8$ V, $T_{\text{dec}} = 1$ ms, $I_{\text{sum}}^{\text{pad}} = 1$ μA.

[c] Measured in strong inversion with $V_{\text{dd}} = 1.8$ V, $T_{\text{dec}} = 20$ μs, $I_{\text{sum}}^{\text{pad}} = 30$ μA.

Table 2.4: Summary of the analog (16,5,8) binary Reed-Muller decoder ASIC.

necessary for modern communications standards. Especially a reprogrammable deinterleaver that is necessary represents a great challenge for implementation [23].

A different approach is to out-source the interleaver and to hardwire only a trellis sliding window decoder which functions as an analog coprocessor that interacts with a digital processor [49]. The digital processor stores and relays the values computed by the analog coprocessor and functions as a deinterleaver.

We have implemented an analog integrated circuit that can be used as an analog coprocessor [21], decoding several trellis-sections of a turbo code according to the UMTS specifications [14].

The chip has not yet returned from fabrication, therefore, no measurement results, only a brief description of the circuit, can be presented.

This analog coprocessor design was mainly carried out by Daniel Furrer under the guidance of Matthias Frey and Jonas Biveroni.

2.5.1 Turbo Encoder and Sliding Window Decoder

The turbo encoder according to the UMTS standard [14] is given in Figure 2.57. The resulting code is a rate 1/3 systematic block code. The encoder consists of two recursive systematic convolutional encoders with memory 3, and an interleaver.

In practical applications, turbo codes are decoded by means of iterative decoding. The factor graph of this turbo code is shown in Figure 2.58.

The inputs μ_{in1}, μ_{in2} and μ_{in3} of the decoder are the likelihoods of the received bit streams. The horizontal edges of the factor graph represent the trellis states of the two component decoders. Iterative turbo decoding on this factor graph is in principle an alternating decoding of the two component trellises.

Initially, the forward-backward (BCJR) algorithm [1] runs on the upper component trellis, whereas the intrinsic information μ_{inA} at this first iteration is equal to the likelihood μ_{in1}. The output message μ_{outA} of the upper component decoder, also called the extrinsic information, is

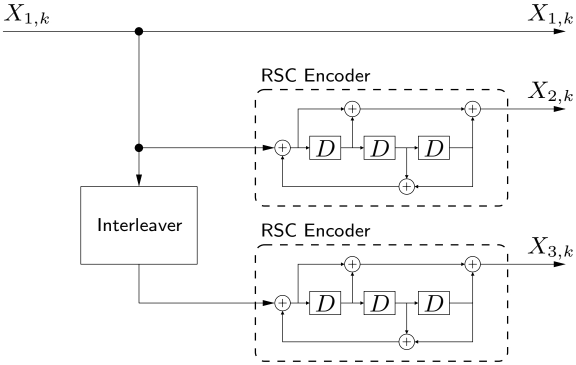


Figure 2.57: Turbo encoder

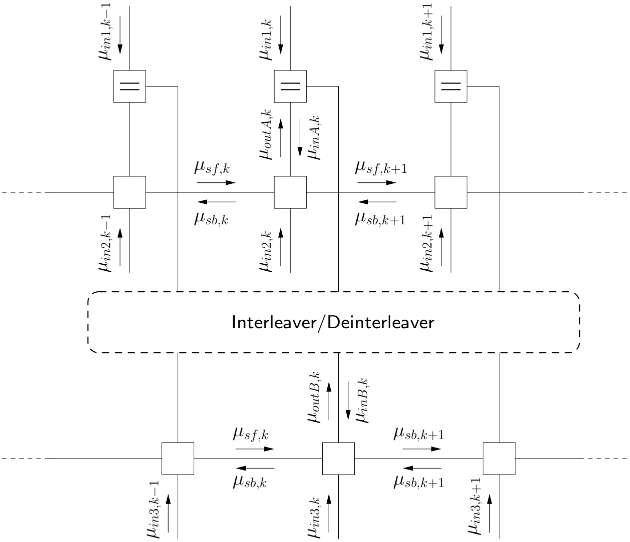


Figure 2.58: Schematic of the analog coprocessor (slding window decoder) used in a turbo decoder.

then propagated together with μ_{in1} through the interleaver and becomes the intrinsic information μ_{inB} of the second component decoder on the lower trellis.

The forward-backward algorithm then decodes the lower component trellis and the extrinsic information μ_{outB} is returned through the deinterleaver to the upper trellis decoder. The alternating decoding of the two component trellises stops when the messages do not change anymore.

The above-mentioned decoding scheme is further described in Figure 2.59. The "analog coprocessor" performs the forward-backward algorithm on a component trellis, the digital processor acts as an interleaver and a deinterleaver and combines the information correctly.

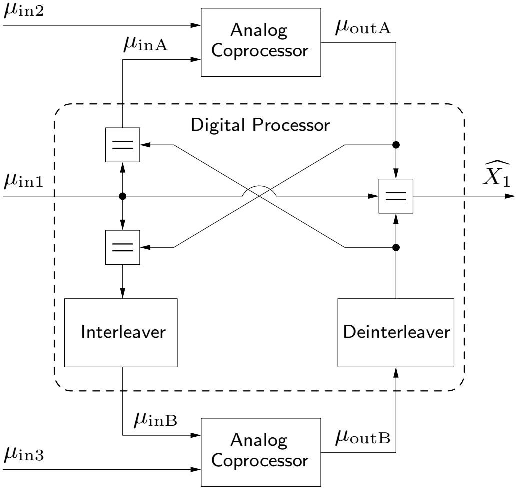


Figure 2.59: Schematic of a turbo decoder with two analog coprocessors and a digital processor.

As the block length of the UMTS turbo code is variable, the analog coprocessor needs to be able to perform the forward-backward algorithm on trellises of various lengths. A feasible approach is to implement a "sliding window decoder" that decodes the trellis section-wise and "moves" along the trellis.

The sliding window technique is a common practice for decreasing the latency and for minimizing the required memory while decoding long convolutional codes. The idea is that for the decoding of a single bit not the whole trellis is necessary, but only the information in the vicinity of this bit. In other words, the forward-backward algorithm performed on a sub-block closely approximates the exact output distributions (obtained from the forward-backward algorithm performed on the full trellis) if the algorithm can settle on a "stabilization length" L of five to six times the constraint length of the code. The sliding window decoder operating on the factor graph is illustrated in Figure 2.60. In every step the analog coprocessor is fed with two messages ($\mu_{\text{in2},k}$ and $\mu_{\text{inA},k}$) and, with a lag of L steps, it returns one output message $\mu_{\text{outA},j}$.

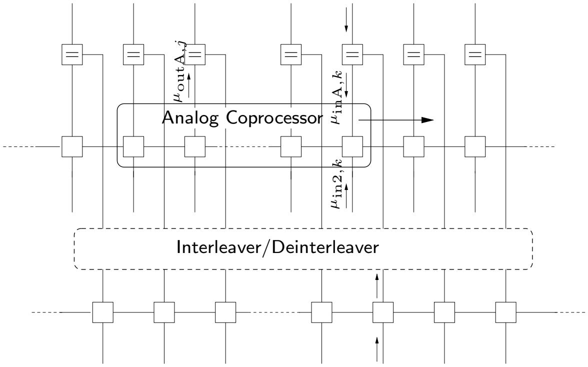


Figure 2.60: Factor graph of a turbo code with the analog coprocessor operating as a sliding window decoder.

2.5.2 Description of the Circuit

The analog coprocessor was implemented in an IBM 0.18 μm BiCMOS technology (IBM 7WL). The decoder core makes use of CMOS transistors in weak inversion and covers an area of 1 mm^2. The I/O-interface uses few bipolar transistors to obtain maximum speed. The whole chip area is 3 mm^2,with a pin-count of 28.

The transistors used in the multipliers and in the scaling-circuits rely on the transistors' exponential characteristics. Therefore, the corresponding nMOS and pMOS transistors need to operate in the subthreshold region (weak inversion). To maximize the speed of the decoder, the other transistors (mainly current mirrors) operate in the moderate inversion. The nominal unit current (current corresponding to a uniform distribution) was chosen to be 1 μA. For the different types of transistors, the sizes given in Table 2.5 were chosen.

	W/L [μm]	IC	$\sigma\left(\frac{\Delta I_\mathrm{D}}{I_\mathrm{D}}\right)$
pMOS mirror:	4.0/0.5	0.8	8.5 %
nMOS mirror:	2.0/1.0	0.1	16.8 %
nMOS scaler:	4.0/0.5	0.5	20.2 %

Table 2.5: Width and length, as well as resulting inversion coefficients (IC) and mismatch for the different types of transistors.

The inversion coefficient (IC) is computed with the equations given in Section 2.3. The drain-current mismatch numbers are obtained with formulas derived from [34], with the EKV-transistor model introduced in [13] and with measured process data given in the design manual. It is clear that these numbers are only valid (as estimations) for the given unit current of 1 μA.

The Decoder Core

40 trellis sections arranged in a ring-structure were integrated on the chip. The state-metrics obtained from the forward recursion (i.e., information on the trellis-states obtained from the previous steps) are not stored when the window "moves" over the trellis. Therefore, for the forward as well as for the backward iteration, a "stabilization length" of L (i.e., $L = 6 \times 3 = 18$ for the UMTS turbo code with constraint length 3) is necessary. To be on the safe side, $L = 20$ was chosen. In order to decode the beginning and the end of the terminated trellis code correctly, the state-metrics μ_sf and μ_sb can be set to $\mu_\mathrm{sf,b}(0) = 1$ and $\mu_\mathrm{sf,b}(i) = 0$, for $i = 1 \ldots 7$. For the other sections of the trellis, the messages are initialized with a uniform distribution: $\mu_\mathrm{sf,b}(i) = 0.125$, for $i = 0 \ldots 7$.

One of the 40 implemented slices is depicted in Figure 2.61.

The module "A" precomputes the branch metric $\mu_{inA2,k}$ from the input messages $\mu_{inA,k}$ and $\mu_{in2,k}$. Module "B" performs the computation of the forward recursion and module "C" performs the computation

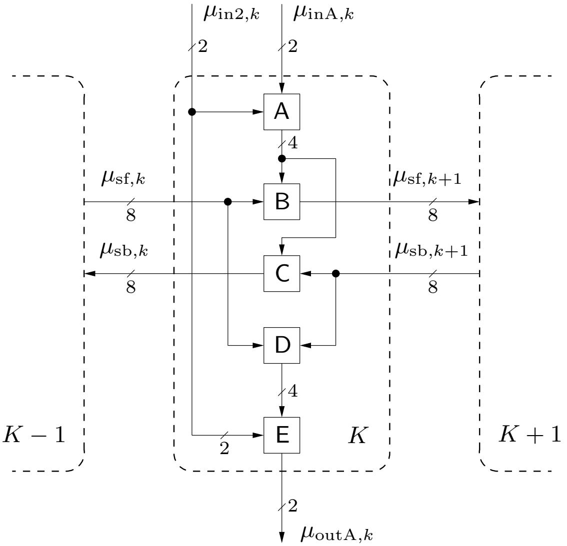


Figure 2.61: Slice of the trellis decoder

of the backward recursion. Finally, module "D", followed by module "E", compute the extrinsic information $\mu_{outA,k}$ from the forward and backward recursions $\mu_{sf,k}$ and $\mu_{sb,k+1}$, respectively. These modules were implemented according to the recipes described in [43].

The Input Interface

The input messages $\mu_{\mathrm{inA},k}$ and $\mu_{\mathrm{in2},k}$ need to be stored during the decoding of a trellis-section. This is realized in a similar way as the serial-to-parallel interface of the Reed-Muller decoder described in Section 2.4.1. This input stage however converts the input current by a factor of 1:1000. This large factor is achieved by an nMOS stage and a pMOS stage. The advantage of down scaling the current across the memory-stage is that the larger current can be used to charge the storage-capacitor, which may lead to an increased loading speed. The large current of 1 mA at the pins can be provided directly by a DAC with current output.

The reset transistor can be used to discharge the capacitor before loading the new values; this functionality can be switched off globally.

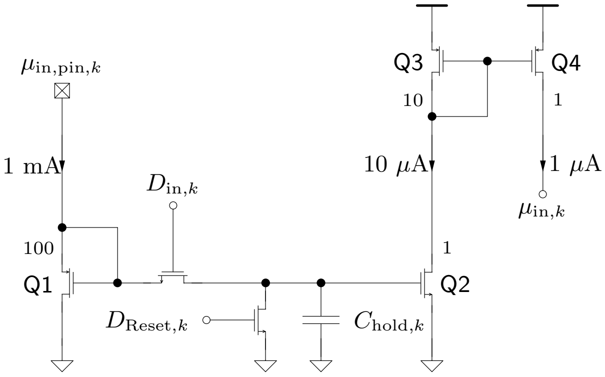


Figure 2.62: Input stage of the trellis decoder with the level of "neutral" currents (currents representing a neutral message) denoted.

The Output Interface

It is difficult to provide a fast I/O interface to measure the maximum speed of the decoder core, if the current levels in the decoder core are around 1 μA. In order to drive an off-chip A/D-converter, output currents of at least 1 mA are necessary. Therefore, a fast current-converter with a factor of 1:1000 is necessary. A simple 1:1000 current mirror is unsuitable as the capacitor of the larger transistor needs to be driven by the small current of the decoder core and this would limit the maximum speed. A possible solution is the BiCMOS current scaler circuit shown in Figure 2.63. The bipolar transistors make up the scaling circuit, whereas the two nMOS transistors Q1 and Q2 provide the basis current for the bipolar devices. The output current level can be adjusted with the bias current $I_{\mathrm{sum}}^{\mathrm{pad}}$; nominally, it is set to $I_{\mathrm{sum}}^{\mathrm{pad}} = 2$ mA. The bipolar current scaler circuit has only a current-converter ratio of 1:100, however, there is a preceding pMOS stage with a converter ratio of 1:10.

2.5.3 Preliminary Conclusions

This most recent analog decoder has not been measured yet and no reliable conclusions can be drawn.

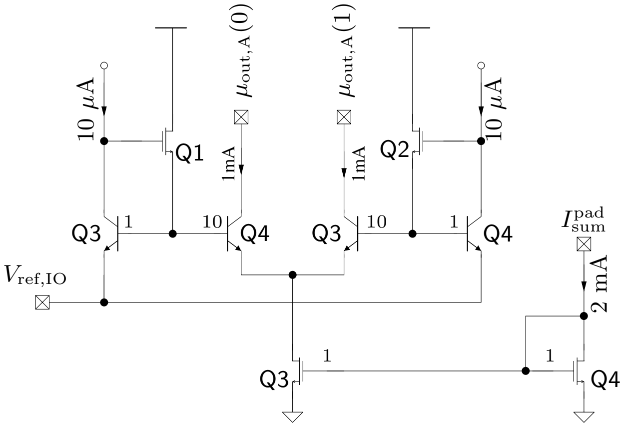


Figure 2.63: Output stage of the analog coprocessor with the level of the "neutral" currents (currents representing a neutral message) denoted.

The design is novel in the way that so far all decoders have relied on structures with feedback, whereas this analog coprocessor contains a forward-backward processor without any feedback[6]. Transistor-level simulations (without considering mismatch) showed that the lack of feedback and the current-scaler's limited output resistance may affect the precision of the computed output probabilities. Furthermore, in order to optimize the chip for high data-rates and low power-consumption, the transistors were chosen much smaller then they were in the previous designs; this may further limit the precision of the circuit.

Only measurements can reveal whether such a mixed-signal turbo decoder is a feasible alternative to a purely digital design. It is also unclear if the power consumption of the analog decoder core and the converters is smaller than that of a digital decoder.

---

[6] Feedback is introduced through iterations over the digital "interleaver-/deinterleaver"-structure.

2.6 Concluding Remarks

After an introduction to the history of analog decoding, two sets of softgates, the building blocks of analog decoders, as well as thorough measurements thereof were presented.

We observed that analog decoders are rather robust: the softgates were wired on breadboard level with long connection wires, surely in a suboptimal way. Nevertheless, the decoders were fully operational and their frame error rate curves lie less than 0.15 dB away from the ideal curves (determined by a discrete time simulation) at an error rate of FER $= 10^{-2}$. Error rate measurements were performed under various operating conditions, e.g., variable supply voltages or variable sum currents (from weak inversion to strong inversion). Thanks to the CMOS transistors operating in weak inversion, the decoders operate correctly down to supply voltages as low as $V_{\mathrm{dd}} = 1.2$ V which is substantially lower than the nominal supply voltage of $V_{\mathrm{dd}} = 5$ V. By increasing the sum current, the transistors leave the subthreshold region. The speed of the decoder increases and the mismatch decreases, but especially the softEQU gate shows a strong deviation from the ideal characteristics, causing a degradation of the decoder's performance.

The second-generation of softgates incorporates a simplified circuit that does not require any external reference voltages. Other than the first-generation of softgates, the second-generation is fully symmetrical towards the three outputs. The reduced transistor area and the smaller (internal) sum current (both reduced by a factor of $\frac{1}{6}$) deteriorate the performance of the decoders built out of these softgates. The smaller transistors and the resulting increased mismatch could not be clearly observed on softgate-level, however, for the discrete Hamming decoder, the error rate curves lie further away from the ideal curves, and the transistor mismatch is apparent. The implementation loss for a frame error rate of FER $= 10^{-2}$ measured up to 0.45 dB. Furthermore, the second-generation softXOR shows systematically different characteristics than the softXOR of the first-generation: the output probabilities are overconfident, i.e., the output entropy is increased. These aberrations from the ideal characteristics appear systematically and can therefore not result from mismatch only. High-level simulation tools (incorporating simple mismatch models) could not predict the error rate curves of these analog decoders properly.

As the second-generation of softgates also included input reset transistors, it was possible to build a redundant Hamming decoder. This decoder showed an improved performance, but the hardware complexity and the power consumption were doubled. The redundant Hamming decoder showed improved behavior with the transistors operating in the moderate or strong inversion; this coincides with the simulations in [20].

Two fully operational CMOS ASIC analog decoders—a decoder for an (8,4,4) extended Hamming code as well as a decoder for a (16,5,8) Reed-Muller code—were presented. Especially the second decoder may be of interest, since the implemented ad-hoc algorithm shows relatively short cycles in the recursive computation of the messages. A serial-to-parallel interface which is new in this context was introduced. The storage capability of the interface was measured and it proved to store the currents much longer than transistor-level simulations predicted. In weak inversion (i.e., with $I_{\text{sum}}^{\text{pad}} = 1\ \mu\text{A}$) the implementation loss is larger for the Reed-Muller decoder (0.36 dB at FER $= 10^{-2}$) than for the Hamming decoder (0.20 dB at FER $= 10^{-2}$). In strong inversion ($I_{\text{sum}}^{\text{pad}} = 30\ \mu\text{A}$), with reduced mismatch, the implementation loss of the Reed-Muller decoder (0.08 dB at FER $= 10^{-2}$) is smaller than for the Hamming decoder (0.20 dB at FER $= 10^{-2}$). The sample size of ten chips is, however, too small to be able to draw some statistically relevant conclusions. All the transistors for the two ASICs were chosen rather large (approximately 100 times the area of a minimal-sized transistor), and the transistor mismatch was kept below 5 %, i.e., $\sigma_{\delta_I} < 0.05$. Both decoders can be operated with a supply voltage below 1 V, although they were designed for a nominal supply voltage of $V_{\text{dd}} = 1.8$ V. The "energy per decoded info bit" of both analog decoder ASIC's (140 nJ for the Hamming decoder and 7 nJ for the Reed-Muller decoder) are substantially higher than for the digital LDPC decoder presented in [4]. This comparison, however, should not be made directly as the block lengths of the codes vary greatly (1024 bit LDPC code vs. (8,4,4) extended Hamming code and (16,5,8) Reed-Muller code, respectively) and the "energy per decoded bit" depends on the length of the code. The assumption that an analog decoder consumes 100 times less energy than a digital implementation may still hold.

Our latest design, an analog coprocessor for decoding turbo codes, was introduced. Concerning the transistor sizing, this new chip was designed more aggressively with smaller transistors for higher speeds. A forward-backward algorithm was implemented and, other than the previous decoders, this newest chip does not contain any feedback in the circuit. We believe that such mixed-signal designs unite the advantages of both digital and analog worlds: the flexibility of digital reprogrammable circuits and the power-consumption or speed advantage of analog circuits. But only measurements will tell the feasibility and the advantages of such a solution.

Chapter 3

Soft Symbol Detection

3.1 Introductory Remarks

Analog circuits have revealed some advantages over their digital counterparts, mainly in terms of area and power-consumption; moving towards a fully analog receiver seems to be rewarding. The proposed circuits can be used as part of the demodulator between the channel and a soft decoder, where the channel is modeled as an additive white Gaussian noise channel. The following circuits blend in nicely with the analog decoders of the previous chapter.

Pulse-amplitude modulation (PAM) as well as quadrature-amplitude modulation (QAM) are common schemes used in digital communication [53]. The proposed analog circuits detect soft symbols for any type of baseband PAM and QAM signals, transmitted over an additive white Gaussian noise (AWGN) channel. The output of the circuits are likelihoods (soft symbols) that are being represented by currents. The outputs of these circuits can be used as an input to soft-input decoders, in the analog or in the digital domain.

An analog circuit computing the likelihoods for 16-QAM was published by Seguin et al. in [55]. However, their circuit gives only an approximate solution. The research on this soft symbol detection circuit was done in collaboration with Patrick Merkli and Hans-Andrea Loeliger and it led to a patent application [18, 19].

The structure of this chapter is as follows: after these introductory remarks, PAM and QAM and the resulting likelihoods are further described. The soft symbol detector circuit for PAM is described in its most generic form, and with an example in Section 3.3. Section 3.4 describes the extension of the circuit to detect the soft symbols of QAM signals. Further applications of the circuit for trellis coded modulation systems, as well as for systems with iterative receivers, are discussed in Section 3.5. A generic current-rescaling circuit, related to the soft symbol detector circuit, is introduced in Section 3.6. The chapter is then concluded with an outlook and some concluding remarks in Section 3.7.

3.2 PAM, QAM and Probabilities

In a digital communication system, the modulator maps a sequence of binary digits onto a corresponding signal waveform s_m. In pulse-amplitude modulation (PAM) the k-bit sequence a_m is mapped onto the signal amplitude A_m. The baseband signal waveforms can therefore be represented as is shown in (3.1).

$$s_m(t) = A_m g(t), \quad m = 0, 1, \ldots, M-1 \tag{3.1}$$

$\mathcal{A}_M = \{A_m |\ 0 \le m \le M-1\}$ denotes the set of M possible signal amplitudes, corresponding to $M = 2^k$ possible k-bit blocks of symbols $\{a_m |\ 0 \le m \le M-1\}$ that need to be transmitted. The waveform $g(t)$ is a real-valued signal pulse. We do not focus on the modulation and demodulation blocks necessary for transmission; for the following discussions we will assume a discrete memoryless channel (DMC) and will therefore not review the properties of the waveform $g(t)$.

The signal space diagram for the most generic M-PAM scheme is shown in Figure 3.1.

As a channel model, the additive white Gaussian noise (AWGN) channel depicted in Figure 3.2 is used. The probabilities for the channel output $Y = y$ conditioned on a possibly sent symbol $\{a_1, a_2, \ldots, a_{M-1}\}$ with corresponding signal-amplitude $\{A_0, A_1, \ldots, A_{M-1}\}$ are given in (3.2).

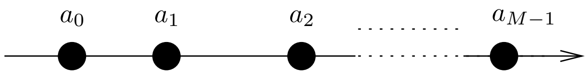


Figure 3.1: Signal space diagram for a digital M-PAM signal.

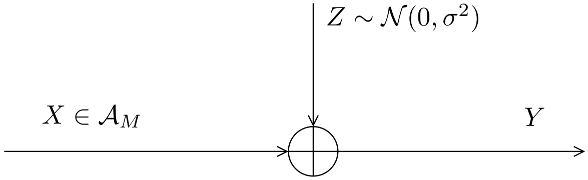


Figure 3.2: The AWGN-channel.

$$P(Y = y|\ X = A_m) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-A_m)^2}{2\sigma^2}} \tag{3.2}$$

A new function $\tilde{P}(A_m)$ is now introduced in (3.3): a fixed input $Y = y$ is assumed, and the likelihood $P(Y = y|\ X = A_m)$ is normalized with the sum of all possible likelihoods.

$$\begin{aligned}
\tilde{P}(A_m) &= \frac{P(Y = y|\ X = A_m)}{\sum_{j=0}^{M-1} P(Y = y|\ X = A_j)} && (3.3)\\
&= \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-A_m)^2}{2\sigma^2}}}{\sum_{j=0}^{M-1} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-A_j)^2}{2\sigma^2}}} && (3.4)\\
&= \frac{e^{\frac{2yA_m - A_m^2}{2\sigma^2}}}{\sum_{j=0}^{M-1} e^{\frac{2yA_j - A_j^2}{2\sigma^2}}} && (3.5)\\
&= \left(\sum_{j=0}^{M-1} e^{\frac{2y(A_j - A_m) + (A_m^2 - A_j^2)}{2\sigma^2}} \right)^{-1} && (3.6)
\end{aligned}$$

Rectangular quadrature amplitude modulation (QAM) can be represented as a linear combination of two orthonormal signal waveforms as shown in (3.7) and (3.8).

$$s_m(t) = A_m g_I(t), \qquad m = 0, 1, \ldots, M-1 \tag{3.7}$$

$$s_n(t) = A_n g_Q(t), \qquad n = 0, 1, \ldots, N-1. \tag{3.8}$$

A signal space diagram for generic rectangular QAM is depicted in Figure 3.3.

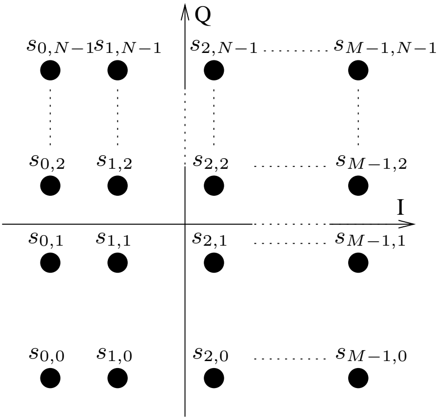


Figure 3.3: Signal space diagram for rectangular QAM signals.

Again the AWGN channel model is applied. As the coordinates of $s_{m,n}$ can be split into two orthogonal parts—the in-phase (I) and the quadrature-phase (Q)—the conditional probability of having sent a sequence $s_{m,n}$ with amplitudes A_m and A_n, given the received signals y_m and y_n, is given as the multiplication of two probabilities as is shown in (3.9).

$$\begin{aligned} &P(Y_\mathrm{I} = y_\mathrm{I}, Y_\mathrm{Q} = y_\mathrm{Q} |\, X = S_{m,n}) \\ &\qquad = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_\mathrm{I} - A_m)^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_\mathrm{Q} - A_n)^2}{2\sigma^2}} \end{aligned} \tag{3.9}$$

Again the input signal is fixed to $Y = y$ and a normalized likelihood function $\tilde{P}(S_{m,n})$ is introduced:

$$\tilde{P}(S_{m,n}) = \frac{P(Y_\mathrm{I} = y_\mathrm{I}, Y_\mathrm{Q} = y_\mathrm{Q} |\, X = S_{m,n})}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} P(Y_\mathrm{I} = y_\mathrm{I}, Y_\mathrm{Q} = y_\mathrm{Q} |\, X = S_{i,j})} \tag{3.10}$$

$$= \frac{e^{\frac{2y_\mathrm{I} A_m - A_m^2}{2\sigma^2}} \cdot e^{\frac{2y_\mathrm{Q} A_n - A_n^2}{2\sigma^2}}}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(e^{\frac{2y_\mathrm{I} A_i - A_i^2}{2\sigma^2}} \cdot e^{\frac{2y_\mathrm{Q} A_j - A_j^2}{2\sigma^2}} \right)} \tag{3.11}$$

These normalized likelihoods, $\tilde{P}(A_m)$ and $\tilde{P}(S_{m,n})$, are the values needed as the input to soft decoders (e.g., analog decoders). The desired “soft symbol detector” circuit should provide output currents that are proportional to these normalized likelihoods.

After this short introduction to pulse- and quadrature amplitude modulation and the related probabilities and likelihoods, some circuits will be introduced.

3.3 PAM Soft Symbol Detector Circuit

For the following circuits, all the transistors are assumed to function as described in Section 2.1.3; the following equation for MOS transistors in weak inversion operating in the active region holds:

$$I_\mathrm{D} = I_\mathrm{D0} e^{\frac{V_\mathrm{GS}}{nU_\mathrm{T}}} \tag{3.12}$$

First, the well-known “differential pair” circuit, shown in Figure 3.4, will be examined.

For a differential pair containing two identical transistors at the same temperature, the following equations hold:

$$\frac{I_0}{I_\mathrm{sum}} = \frac{e^{\frac{\alpha y}{nU_T}}}{e^{\frac{\alpha y}{nU_T}} + e^{\frac{-\alpha y}{nU_T}}} \tag{3.13}$$

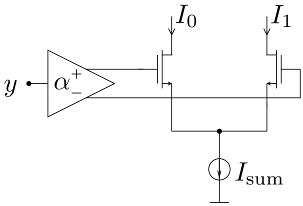


Figure 3.4: The differential pair.

$$\frac{I_0}{I_{\text{sum}}} = \left(1 + e^{\frac{-2\alpha y}{nU_T}}\right)^{-1} \tag{3.14}$$

$$\frac{I_1}{I_{\text{sum}}} = \left(1 + e^{\frac{2\alpha y}{nU_T}}\right)^{-1} \tag{3.15}$$

It is obvious that these two normalized currents can be proportional to the normalized likelihoods of (3.6). The differential pair can therefore be used to compute normalized likelihoods for 2-PAM signals. The gain-factor α needs to be adjusted to the channel's SNR and the transistor parameters. This now well-known fact was first presented in [38] and later nicely derived in the appendix of [48].

The circuit introduced in Figure 3.5 is an extension, or generalization, of the differential pair.

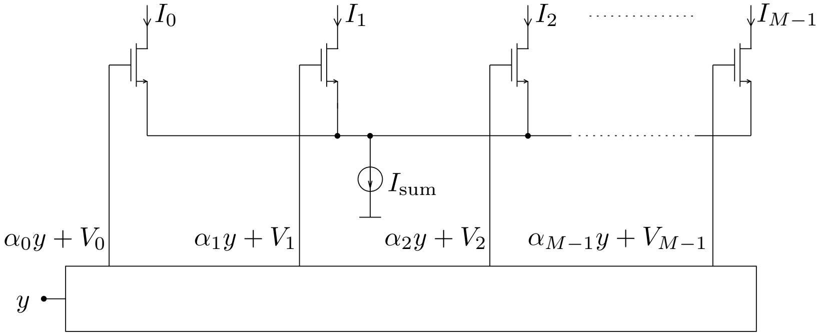


Figure 3.5: Generic circuit used for determining the soft symbols of a PAM signal.

The input to this circuit, y (e.g., an input voltage), is amplified with various gain-factors α_i and shifted by offsets V_i. The resulting voltages, $\alpha_i y + V_i$, are then applied to the gates of the transistors as illustrated in Figure 3.5.

The equations of the normalized currents $\frac{I_m}{I_{\text{sum}}}$ of the circuit shown in Figure 3.5 are given in (3.16).

$$\frac{I_m}{I_{\text{sum}}} = \frac{I_{\text{D0}} \cdot e^{\frac{\alpha_m V_{\text{in}} + V_m}{nU_{\text{T}}}}}{\sum_{j=0}^{M-1} I_{\text{D0}} \cdot e^{\frac{\alpha_j V_{\text{in}} + V_j}{nU_{\text{T}}}}} \tag{3.16}$$

$$= \left(\sum_{j=0}^{M-1} e^{\frac{(\alpha_j - \alpha_m)V_{\text{in}} + (V_j - V_m)}{nU_{\text{T}}}} \right)^{-1} \tag{3.17}$$

Again, these equations strongly resemble the equations for the likelihoods given in (3.6). By choosing the circuit parameters as shown in (3.21) and (3.22), the normalized output currents of the proposed circuit are proportional to the scaled likelihoods of (3.6). By setting the gain-factors α_j and the offset-voltages V_j appropriately, this circuit can be used to determine the soft symbols for a received PAM signal.

$$\forall \{m |\, m \in \mathbb{N}_0,\ 0 \le m \le M-1\} :$$

$$\frac{I_m}{I_{\text{sum}}} = \tilde{P}(A_m) \tag{3.18}$$

$$\sum_{j=0}^{M-1} e^{\frac{(\alpha_j - \alpha_m)V_{\text{in}} + (V_j - V_m)}{nU_{\text{T}}}} = \sum_{j=0}^{M-1} e^{\frac{2(A_j - A_m)y + (A_m^2 - A_j^2)}{2\sigma^2}} \tag{3.19}$$

$$\forall \{j, m |\, j, m \in \mathbb{N}_0,\ 0 \le j, m \le M-1\} :$$

$$\frac{(\alpha_j - \alpha_m)V_{\text{in}} + (V_j - V_m)}{nU_{\text{T}}} = \frac{2(A_j - A_m)y + (A_m^2 - A_j^2)}{2\sigma^2} \tag{3.20}$$

$$\Rightarrow \alpha_j - \alpha_m = \frac{A_j - A_m}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} \tag{3.21}$$

$$\Rightarrow V_j - V_m = \frac{A_m^2 - A_j^2}{2\sigma^2} \cdot nU_{\text{T}} \tag{3.22}$$

It must be emphasized that the absolute value of the voltages V_i, as well as the absolute value of the gain-factors α_i, are not fixed, only the gain-differences are determined by (3.21) and the voltage-differences by (3.22).

The gate voltages, $\alpha_i y + V_i$, can easily be obtained by using operational amplifiers; a possible circuit realization is shown in Figure 3.6.

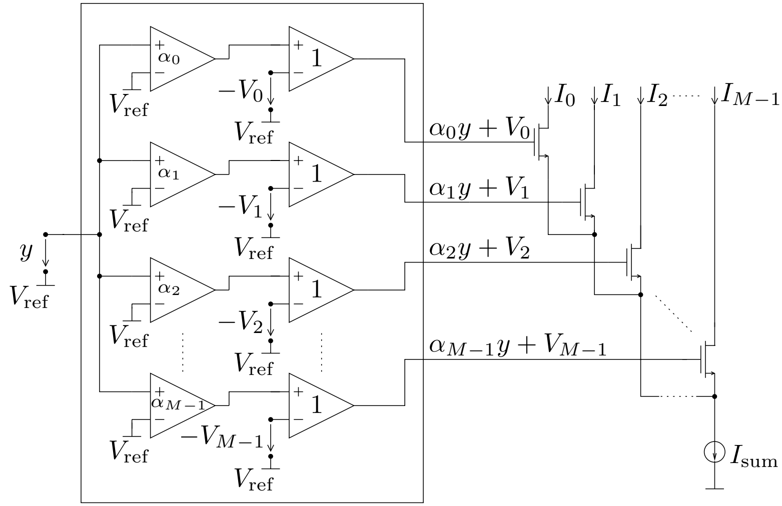


Figure 3.6: Example of a "soft symbol detection" circuit using operational amplifiers to create $V_\text{gate} = \alpha_i V_\text{in} + V_i$.

The same functionality can also be achieved with the circuit shown in Figure 3.7. This circuit may simplify the input stages, especially when the offset-voltages V_i are equal for several gates. The output currents of this alternative realization are exactly the same as for the configuration shown in Figure 3.6.

We have now introduced a circuit that, given a baseband PAM signal transmitted over an AWGN channel, computes output currents that are proportional to the likelihoods. Several parameters need to be adjusted—parameters that depend on the symbol alphabet, on the channel-SNR and, unfortunately, on transistor parameters and the temperature.

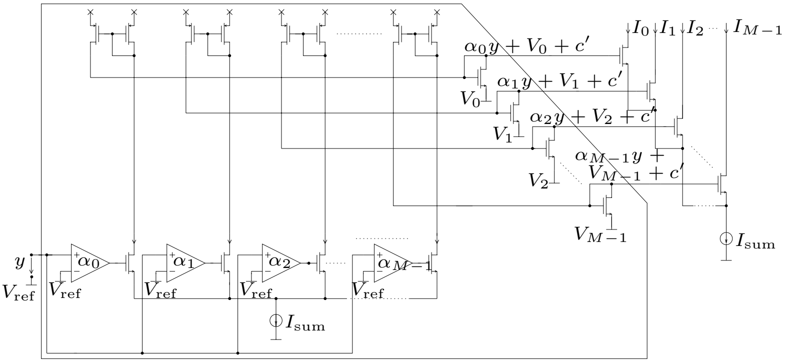


Figure 3.7: Alternative realization of a circuit detecting the soft symbols of a PAM signal.

3.3.1 Example: Soft Symbol Detection Circuit for 4-PAM

In order to clarify the application of a soft symbol detection circuit, an example for a 4-PAM constellation and an AWGN-channel with a variance of σ^2 is given. The signal space diagram is shown in Figure 3.8. This is a special case of the signal space diagram of Figure 3.1, as the signal amplitudes are symmetric to a reference, i.e., $A_m \in \{\pm a, \pm b\}$. The corresponding scaled likelihoods are given in (3.23)–(3.26).

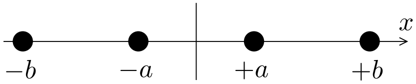


Figure 3.8: Signal space diagram for a symmetric 4-PAM signal.

$$\tilde{P}(X = +a) = \left(1 + e^{\frac{-4ya}{2\sigma^2}} + e^{\frac{2y(b-a)-(b^2-a^2)}{2\sigma^2}} + e^{\frac{2y(-b-a)-(b^2-a^2)}{2\sigma^2}}\right)^{-1} \quad (3.23)$$

$$\tilde{P}(X = -a) = \left(e^{\frac{4ya}{2\sigma^2}} + 1 + e^{\frac{2y(b+a)-(b^2-a^2)}{2\sigma^2}} + e^{\frac{2y(-b+a)-(b^2-a^2)}{2\sigma^2}}\right)^{-1} \quad (3.24)$$

$$\tilde{P}(X = +b) = \left(e^{\frac{2y(a-b)-(a^2-b^2)}{2\sigma^2}} + e^{\frac{2y(-a-b)-(a^2-b^2)}{2\sigma^2}} + 1 + e^{\frac{-4yb}{2\sigma^2}}\right)^{-1} \quad (3.25)$$

$$\tilde{P}(X=-b) = \left(e^{\frac{2y(a+b)-(a^2-b^2)}{2\sigma^2}} + e^{\frac{2y(-a+b)-(a^2-b^2)}{2\sigma^2}} + e^{\frac{4yb}{2\sigma^2}} + 1\right)^{-1} \tag{3.26}$$

The circuit for determining the soft symbols of a 4-PAM signal is shown in Figure 3.9. In (3.27)–(3.30), the circuit's normalized output currents, $\frac{I_0}{I_{\text{sum}}}, \ldots, \frac{I_3}{I_{\text{sum}}}$, are given.

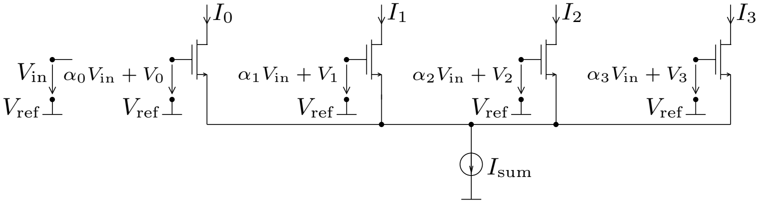


Figure 3.9: Circuit for detecting the soft symbols of a 4-PAM signal.

$$\begin{aligned}\frac{I_0}{I_{\text{sum}}} = \Bigg(&1 + e^{\frac{(\alpha_1-\alpha_0)V_{in}+(V_1-V_0)}{nU_{\text{T}}}} \\ &+e^{\frac{(\alpha_2-\alpha_0)V_{in}+(V_2-V_0)}{nU_{\text{T}}}} + e^{\frac{(\alpha_3-\alpha_0)V_{in}+(V_3-V_0)}{nU_{\text{T}}}}\Bigg)^{-1}\end{aligned} \tag{3.27}$$

$$\begin{aligned}\frac{I_1}{I_{\text{sum}}} = \Bigg(&e^{\frac{(\alpha_0-\alpha_1)V_{in}+(V_0-V_1)}{nU_{\text{T}}}} + 1 \\ &+e^{\frac{(\alpha_2-\alpha_1)V_{in}+(V_2-V_1)}{nU_{\text{T}}}} + e^{\frac{(\alpha_3-\alpha_1)V_{in}+(V_3-V_1)}{nU_{\text{T}}}}\Bigg)^{-1}\end{aligned} \tag{3.28}$$

$$\begin{aligned}\frac{I_2}{I_{\text{sum}}} = \Bigg(&e^{\frac{(\alpha_0-\alpha_1)V_{in}+(V_0-V_1)}{nU_{\text{T}}}} + e^{\frac{(\alpha_1-\alpha_2)V_{in}+(V_1-V_2)}{nU_{\text{T}}}} \\ &+1 + e^{\frac{(\alpha_3-\alpha_1)V_{in}+(V_3-V_1)}{nU_{\text{T}}}}\Bigg)^{-1}\end{aligned} \tag{3.29}$$

$$\begin{aligned}\frac{I_3}{I_{\text{sum}}} = \Bigg(&e^{\frac{(\alpha_0-\alpha_1)V_{in}+(V_0-V_1)}{nU_{\text{T}}}} + e^{\frac{(\alpha_1-\alpha_2)V_{in}+(V_1-V_2)}{nU_{\text{T}}}} \\ &+e^{\frac{(\alpha_2-\alpha_3)V_{in}+(V_2-V_3)}{nU_{\text{T}}}} + 1\Bigg)^{-1}\end{aligned} \tag{3.30}$$

The equations describing the normalized likelihoods can now be compared to the normalized currents in order to determine the correct values for $\alpha_0 \dots \alpha_3$, as well as for $V_0 \dots V_3$. The resulting set of linear equations is, however, under-determined. Three independent linear equations can be found for $\alpha_0 \dots \alpha_3$ as well as for $V_0 \dots V_3$. These are given in (3.31)–(3.33) and (3.34)–(3.36):

$$\alpha_1 - \alpha_0 = \frac{-2a}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} \tag{3.31}$$

$$\alpha_2 - \alpha_0 = \frac{b-a}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} \tag{3.32}$$

$$\alpha_3 - \alpha_0 = \frac{-b-a}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} \tag{3.33}$$

$$V_0 - V_1 = 0 \tag{3.34}$$

$$V_3 - V_2 = 0 \tag{3.35}$$

$$V_1 - V_2 = \frac{b^2 - a^2}{2\sigma^2} \cdot nU_{\text{T}} \tag{3.36}$$

The term $\frac{y}{V_{\text{in}}}$ can be interpreted as a mapping-factor from the input-space to the voltage-space, $\frac{a}{\sigma^2}$ and $\frac{b}{\sigma^2}$ are the normalized signal amplitudes and nU_{T} is a material constant of the transistors. The absolute values of the gain-factors $\alpha_0 \dots \alpha_3$, as well as the absolute values of the voltages $V_0 \dots V_3$, are (theoretically) irrelevant, only the gain- or voltage-differences matter. However, in practical applications other constraints apply, e.g., it is essential that the transistors stay in saturation at all time.

For this example we (arbitrarily) choose a further constraint, $\alpha_0 = -\alpha_1$, which leads to the following gain-factors:

$$\alpha_0 = \frac{a}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} = \alpha \tag{3.37}$$

$$\alpha_1 = \frac{-a}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} = -\alpha \tag{3.38}$$

$$\alpha_2 = \frac{b}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} = \beta \tag{3.39}$$

$$\alpha_3 = \frac{-b}{\sigma^2} \cdot \frac{y}{V_{\text{in}}} \cdot nU_{\text{T}} = -\beta \tag{3.40}$$

The 4-PAM soft symbol detection circuit, with the appropriate gain-factors denoted, is shown in Figure 3.10.

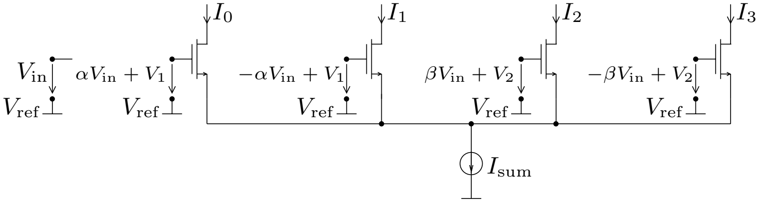


Figure 3.10: Circuit used for detecting the soft symbols of a symmetric 4-PAM signal.

In Section 3.3 an alternative circuit realization was introduced which can now be utilized. The possible advantage of such a circuit is demonstrated on the basis of this example.

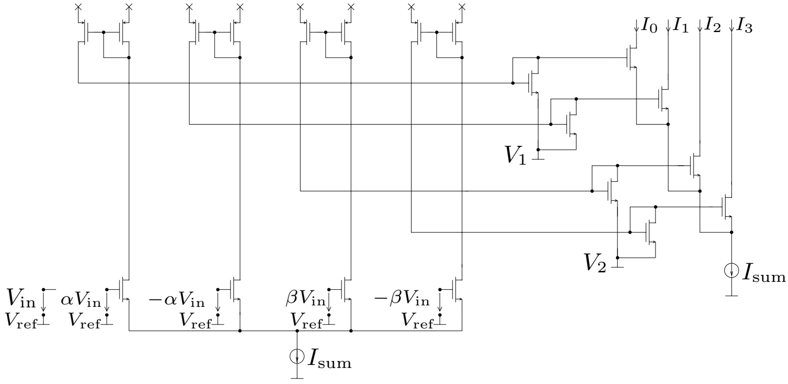


Figure 3.11: An alternative circuit used for detecting the soft symbols of a symmetric 4-PAM signal.

The input gate voltage of the circuit shown in Figure 3.11 can be easily obtained by using two simple differential amplifiers as input stage; one such stage is depicted in Figure 3.12[1].

---

[1] This circuit may not be of practical use for integration; it should be seen as a

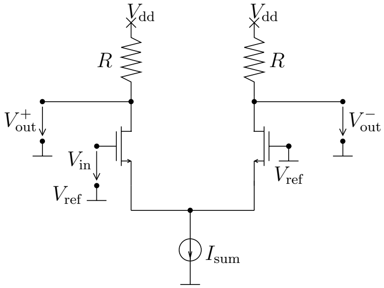


Figure 3.12: A simple circuit that produces the inputs to the PAM soft symbol detector circuit.

The amplifier's linearized differential output voltages V_{out}^{+} and V_{out}^{-} are given in (3.41) and (3.42). It can be seen that the gain of the amplifiers is adjusted by changing the transconductance g_{m} or the resistor R. By changing V_{dd}, R or I_{sum}, the common-mode voltage can be adjusted. For both input stages, the common mode voltages need to be equal.

$$V_{\mathrm{out}}^{+} = V_{\mathrm{dd}} - R\frac{I_{\mathrm{sum}}}{2} + R\frac{g_{\mathrm{m}}}{2}V_{\mathrm{in}} \tag{3.41}$$

$$V_{\mathrm{out}}^{-} = V_{\mathrm{dd}} - R\frac{I_{\mathrm{sum}}}{2} - R\frac{g_{\mathrm{m}}}{2}V_{\mathrm{in}} \tag{3.42}$$

3.4 QAM Soft Symbol Detector Circuit

In Section 3.2 we have stated that, as the in-phase and the quadrature-phase of the I/Q-signal are independent, computing the soft symbols for a rectangular QAM I/Q signal can be done separately for the I-part and the Q-part of the I/Q signal. These two values can then be multiplied to obtain the soft symbols for each possibly transmitted symbol. A QAM soft symbol detection circuit can therefore be separated into 3 separate circuits: the likelihoods related to the in-phase part and to the quadrature-phase part of the signal can be computed separately with

---

simplified example circuit. We further remark that it is a linearized circuit and has nothing in common with the translinear circuits introduced earlier in this chapter, although it may resemble the differential pair of Figure 3.4.

the circuit of the previous chapter; with a multiplier matrix circuit [45] the outputs of these two circuits can then be used to compute the joint likelihoods.

Furthermore, if not all symbols of an M-QAM signal are transmitted, i.e., if a cross-shaped constellation is being used, the same structure as for the underlying rectangular M-QAM constellation can be used. The currents related to the omitted symbols can be dissipated and the remaining outputs need to be rescaled with a current-scaling circuit [45].

The two following examples should further clarify the structure of the circuits that can be used to compute the soft symbols of QAM signals.

3.4.1 Example: Soft Symbol Detection Circuit for 16-QAM

The soft symbols of a 16-QAM signal with the constellation given in Figure 3.13 can be obtained by the circuit shown in Figure 3.14. As described before, it consists of 3 parts: two circuits compute the soft symbols for a 4-PAM signal, extracted from the I-phase or the Q-phase of the QAM-signal, respectively. The matrix multiplier then computes all the symbol metrics out of the metrics from the two phases.

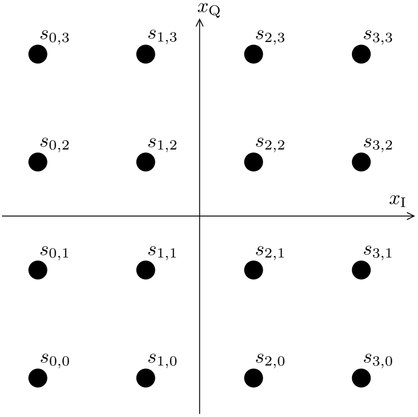


Figure 3.13: Rectangular 16-QAM signal constellation.

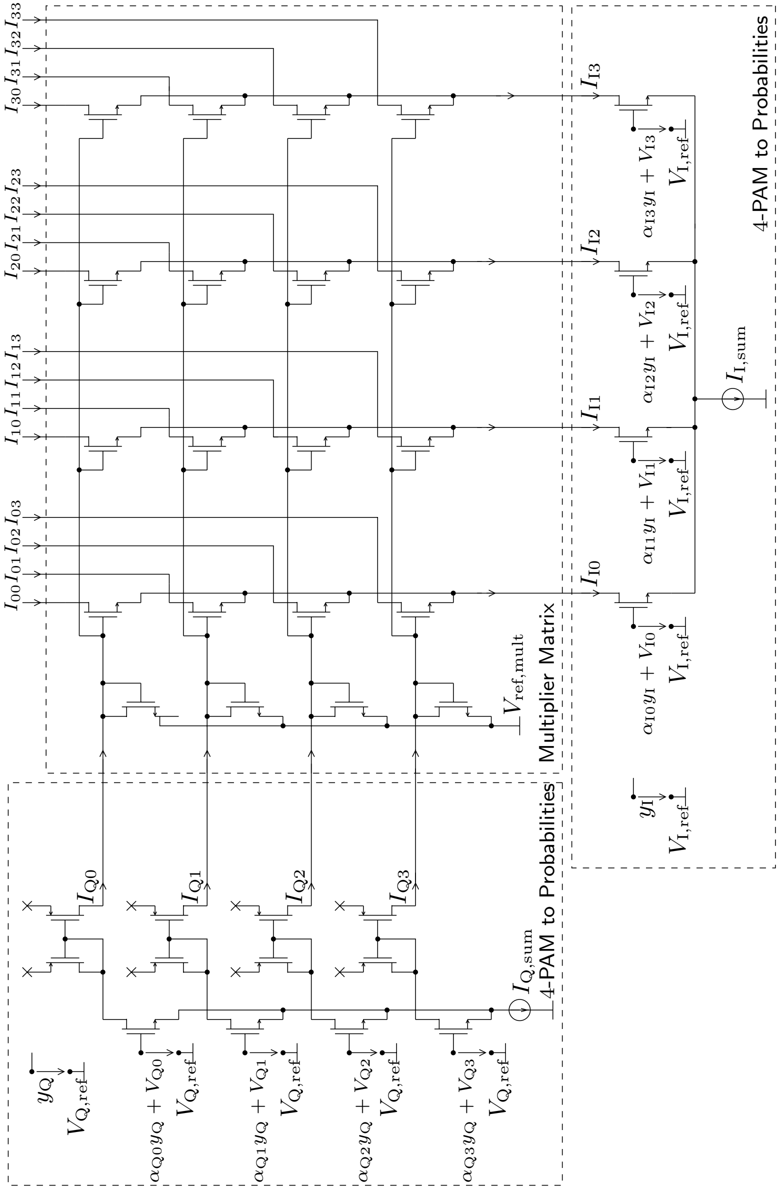


Figure 3.14: Soft symbol detection circuit for rectangular 16-QAM signals.

3.4.2 Example: Soft Symbol Detection Circuit for 12-QAM

If not all possible 16 symbols are being transmitted, as for example in a 12-QAM setup whose signal space diagram is pictured in Figure 3.15, the output currents related to the probabilities of these neglected symbols can be dissipated. If desired, the remaining currents representing the probabilities of the 12 remaining symbols can be rescaled with a simple current-scaling circuit discussed in [25] and [45] and which is depicted in Figure 3.16.

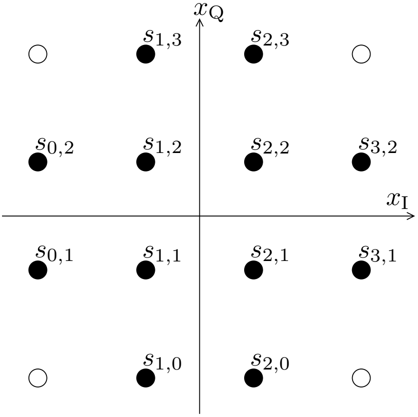


Figure 3.15: Rectangular 12-QAM signal constellation.

3.5 Applications for Trellis-Coded Modulation and Iterative Receivers

3.5.1 From Symbol Metrics to Label Metrics

In trellis coded modulation [56] and bit-interleaved coded modulation [7, 36], the likelihoods $P(Y = y|X = s_i)$ of several symbols s_i are combined to form subset likelihoods or bit likelihoods (also called "label metrics"), which are then used by the decoder. For example for the 4-

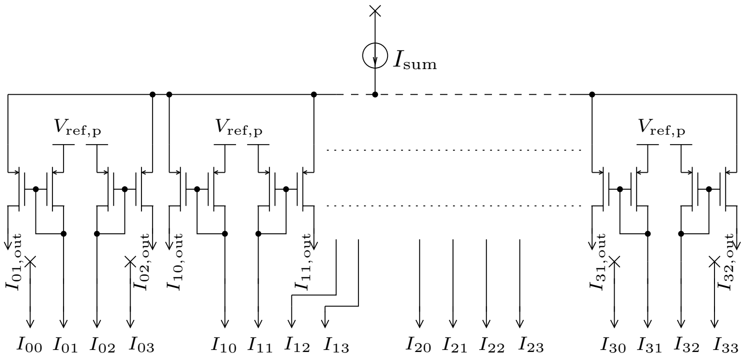


Figure 3.16: Current rescaling circuit used for a soft symbol detection circuit for rectangular 12-QAM signals.

PAM constellation given in Figure 3.8, each symbol can be labeled by two binary numbers as it is shown in Figure 3.17.

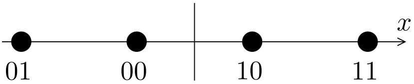


Figure 3.17: Signal space diagram for symmetric 4-PAM signals labeled with two binary numbers.

The metrics of the least significant bit, μ_{lsb}, and the metric of the most significant bit, μ_{msb}, are then:

$$\mu_{\mathrm{lsb}} = [\mu_{\mathrm{lsb}}(0), \mu_{\mathrm{lsb}}(1)] \tag{3.43}$$

$$\mu_{\mathrm{lsb}}(0) = P(Y = y|X = 00) + P(Y = y|X = 10) \tag{3.44}$$

$$\mu_{\mathrm{lsb}}(1) = P(Y = y|X = 01) + P(Y = y|X = 11) \tag{3.45}$$

$$\mu_{\mathrm{msb}} = [\mu_{\mathrm{msb}}(0), \mu_{\mathrm{msb}}(1)] \tag{3.46}$$

$$\mu_{\mathrm{msb}}(0) = P(Y = y|X = 00) + P(Y = y|X = 01) \tag{3.47}$$

$$\mu_{\mathrm{msb}}(1) = P(Y = y|X = 10) + P(Y = y|X = 11) \tag{3.48}$$

The circuit computing the currents that are proportional to the metrics μ_{lsb} and μ_{msb}, I_{lsb} and I_{msb}, respectively, is given in Figure 3.18.

The additions of the likelihoods in (3.44), (3.45) and (3.47), (3.48) are realized by adding the currents that correspond to these likelihoods. It is clear that this scheme is not limited to PAM constellations or to two-bit binary numbers, but can be readily extended to more complicated scenarios.

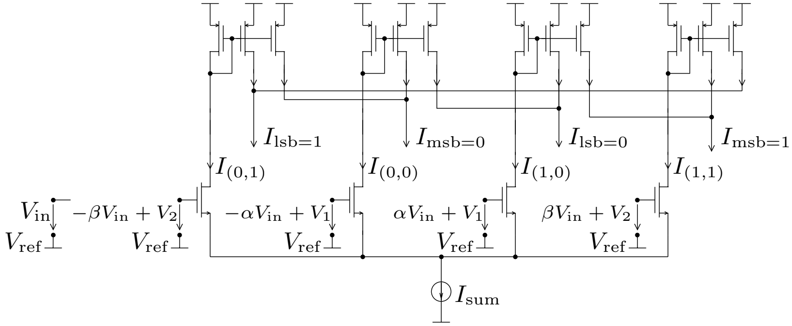


Figure 3.18: Circuit used for detecting the soft-symbols of a 4-PAM signal and then computing the label metrics μ_{lsb} and μ_{msb}.

3.5.2 Iterative SISO Modules

Iterative receivers often utilize a feedback link from the soft decoder to the label metric computation unit. The label metric computation then becomes a soft-input soft-output (SISO) unit as it is schematically shown in Figure 3.19. The corresponding factor graph as well as the corresponding equations are described in [39].

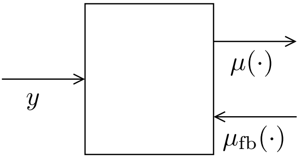


Figure 3.19: A general soft-input soft-output (SISO) label metric computation unit.

Again, we assume to have a 4-PAM constellation with binary labeled symbols (see Figure 3.17), which is now used in an interleaved coded modulation scheme (see [36]).

Let $\mu_{\text{lsb,fb}}$ and $\mu_{\text{msb,fb}}$ be the metrics that are fed back from a decoder. The outgoing bit metrics (according to [39]) are then:

$$\begin{aligned} \mu_{\text{lsb}} &= [\mu_{\text{lsb}}(0), \mu_{\text{lsb}}(1)] && (3.49) \\ \mu_{\text{msb}} &= [\mu_{\text{msb}}(0), \mu_{\text{msb}}(1)] && (3.50) \end{aligned}$$

$$\begin{aligned} \mu_{\text{lsb}}(0) &= P(Y = y|X = 00)\mu_{\text{msb,fb}}(0) \\ &\quad + P(Y = y|X = 10)\mu_{\text{msb,fb}}(1) && (3.51) \\ \mu_{\text{lsb}}(1) &= P(Y = y|X = 01)\mu_{\text{msb,fb}}(0) \\ &\quad + P(Y = y|X = 11)\mu_{\text{msb,fb}}(1) && (3.52) \end{aligned}$$

$$\begin{aligned} \mu_{\text{msb}}(0) &= P(Y = y|X = 00)\mu_{\text{lsb,fb}}(0) \\ &\quad + P(Y = y|X = 10)\mu_{\text{lsb,fb}}(1) && (3.53) \\ \mu_{\text{msb}}(1) &= P(Y = y|X = 01)\mu_{\text{lsb,fb}}(0) \\ &= + P(Y = y|X = 11)\mu_{\text{lsb,fb}}(1) && (3.54) \end{aligned}$$

The complete circuit that computes these bit metrics is shown in Figure 3.20. Next to the soft symbol detection circuit that computes the likelihoods from given input values y, the circuit comprises a matrix multiplier which was described in [43].

Circuits for specific communication systems were presented. However, more generally, many metric computation schemes—for various constellations, labeling configurations, and feedback schemes—can be represented by factor graphs. And all these computation schemes amount to sums of products of non-negative numbers that, if all required likelihoods and probabilities are represented by currents, can be realized with the class of circuits presented in [43]. The new symbol detection circuit fits in with these well-known circuits as it provides the symbol likelihoods as currents and enables metric computations.

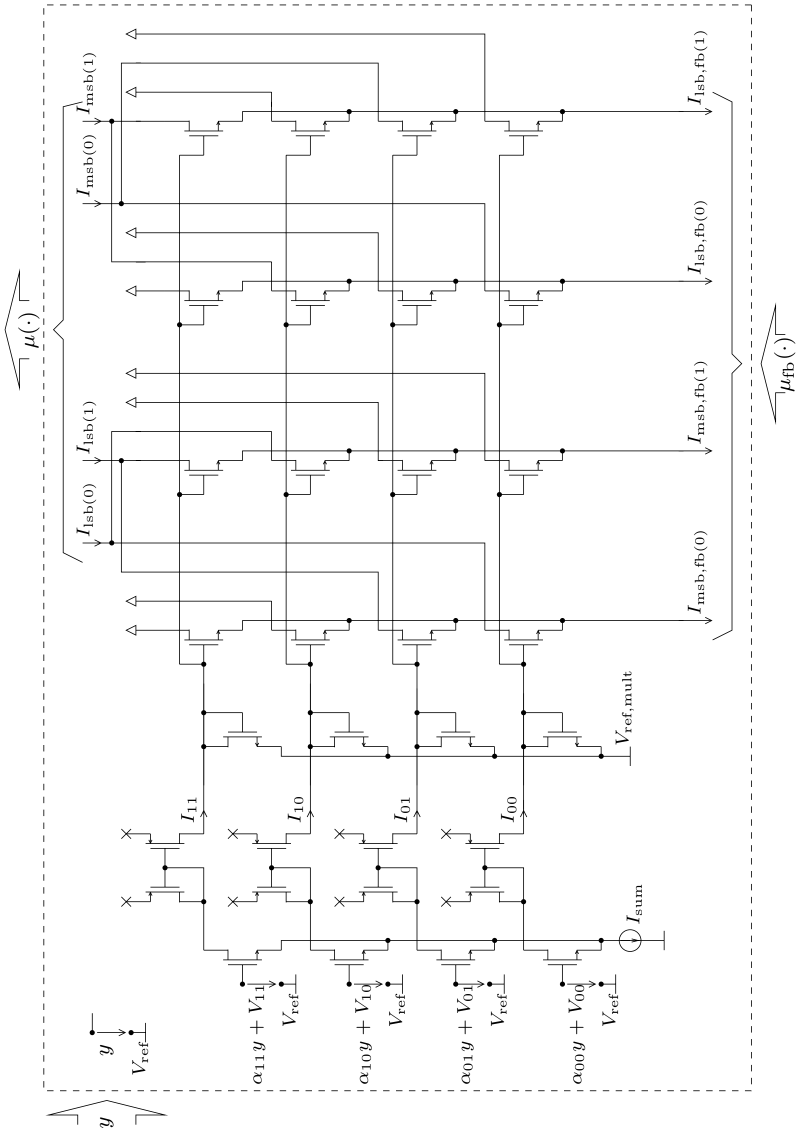


Figure 3.20: An example of a SISO label metric computation circuit.

3.6 Generalized Scaling and Damping

A generalized current scaling can be achieved with the circuit presented in Figure 3.21. The right part of the circuit is again the soft symbol detection circuit described in Section 3.3, whereas the diodes and the operational amplifiers can be seen as the input stage producing an input voltage to the soft-symbol detection circuit. Alternatively, the circuit can be interpreted as a generalized current-scaling circuit with an offset- and gain-stage in between the input diode and the output transistor.

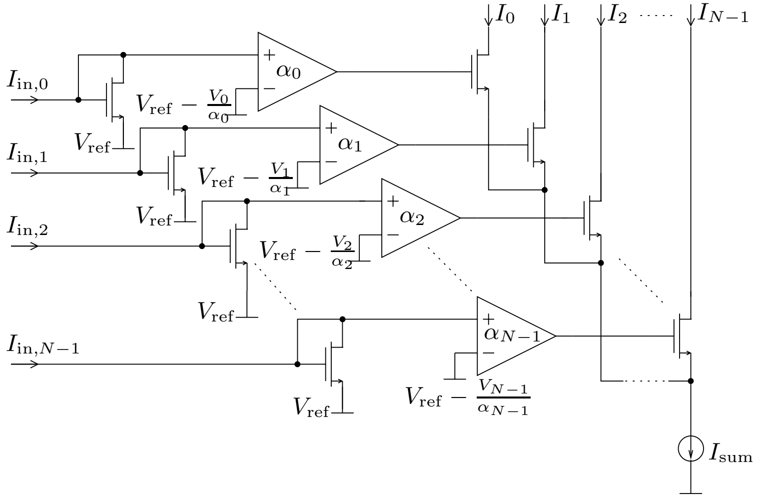


Figure 3.21: A generalized current-rescaling circuit that can be interpreted as a circuit for metric-rescaling and -damping.

The normalized output currents of this circuit, $\frac{I_i}{I_{\text{sum}}}$, amount to

$$\frac{I_i}{I_{\text{sum}}} = \gamma \cdot \beta_i \cdot (I_{\text{in},i})^{\alpha_i} . \tag{3.55}$$

$$\beta_i = e^{\frac{V_i}{nU_{\text{T}}}} \tag{3.56}$$

$$\gamma = \left(\sum_{j=0}^{N-1} \beta_j \, (I_{\text{in},j})^{\alpha^j} \right)^{-1} \tag{3.57}$$

If the input currents $(I_{\text{in},0}, I_{\text{in},1}, \ldots, I_{\text{in},N-1})$ are proportional to the metric $\big(\mu(0), \mu(1), \ldots, \mu(N-1)\big)$, the output currents of the before mentioned circuit are proportional to the following output metric:

$$\begin{aligned}&\big(\mu(0), \mu(1), \ldots, \mu(N-1)\big) \\ &\qquad \mapsto \gamma \cdot \big(\beta_0 \cdot \mu(0)^{\alpha_0}, \beta_1 \cdot \mu(1)^{\alpha_1}, \ldots, \beta_{N-1} \cdot \mu(N-1)^{\alpha_{N-1}}\big)\end{aligned} \tag{3.58}$$

with

$$\gamma = \left(\sum_{j=0}^{N-1} \beta_j \mu(j)^{\alpha_j} \right)^{-1} \tag{3.59}$$

The vector $\beta = [\beta_0, \beta_1, \ldots, \beta_{N-1}]$ weights the probabilities in a nonlinear way, whereas $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_{N-1}]$ resembles a generalized damping. Damping, a term that is used in generalized belief-propagation [67] and also in non-standard sum-product algorithms [48], rescales metrics with an exponential function: $\mu \mapsto \mu^{\alpha}$.

Although we do not have an application in mind, we do find it interesting and think that this circuit might be useful in the future.

3.7 Conclusions and Outlook

A novel circuit that blends in nicely with the family of analog decoders was presented. This analog soft symbol detection circuit can be seen as a further step towards a fully-analog receiver. The soft symbol detection circuit can be used to compute the soft symbols of baseband PAM and QAM signals, transmitted over an AWGN channel. The circuit can easily be extended to related modulation schemes, such as, trellis-coded modulation or bit-interleaved coded modulation. Example circuits demonstrated how this could be accomplished. Not just symbol metrics, but also bit metrics (or label metrics), can be extracted using these circuits.

Other than the circuits used for analog decoders, the soft symbol detection circuit is temperature-dependent. Therefore, for practical applications, a feedback-circuit needs to be incorporated that adjusts the gain-factor to compensate for temperature changes and for drifts in transistor parameters. Furthermore, it would be of interest to detect the channel SNR (or the optimal estimated SNR) and adjust the gain-factor of the circuit automatically. Both tasks have not been approached or solved yet.

Chapter 4

A/D-Conversion with Imprecise Elements

4.1 Introductory Remarks

The world we live in is analog; speech is analog; radio signals are analog (although the transmitted data may be digital and many people may think the world has become “digital” in the “digital age”). The decoder chips we have presented in the previous chapter were all analog, however, most subsequent stages, usually microprocessors or digital signal processors work with digital data only. Analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) are necessary for the analog decoders to interface with the surrounding “digital world.”

As memory is inherently digital, it may also be interesting to store the analog input data of (analog) serial-to-parallel interfaces in (cheap) digital memory, prepended by a single ADC and appended by a bank of DACs. As this could eliminate the costly storage capacitors necessary in fully analog interfaces (e.g., the interface introduced in Section 2.4.1), we are further motivated to find “cheap” integrated ADCs and DACs.

Nyquist-rate ADCs are traditionally designed so that all comparators have predictable thresholds close to the thresholds of an ideal uniform quantizer. Such specifications are hard to meet. In particular, limiting the effects of transistor mismatch requires large transistors—"large" by the standards of digital circuits—and large transistors require large currents to achieve high speed.

However, in many applications (e.g., in communications receivers) there is no need for such stringent specifications. All that is really needed is a sufficient density of comparator thresholds; the position of the individual thresholds need not be predictable at design time and may vary substantially from chip to chip. Clearly, an ADC built in this way needs some digital correction. The effective resolution that can be obtained in this way is smaller than the resolution of an ideal uniform quantizer with the same number of thresholds. However, this loss may be more than compensated by the reduced precision requirements in the analog part of the ADC.

ADCs designed (more or less) according to this principle have been reported in [11, 15, 16, 44, 52]. In all these designs, the ADC contains a bank of low-precision comparators. In [11], a training signal is employed to generate a look-up table that maps the original output codes to a new linearized set. In [15] and [16], only a fraction of the comparators is actually used: during calibration, those comparators with a threshold close to some ideal threshold are identified and the others are deactivated. In [52], all comparators are used and a digital circuit converts the non-monotonic comparator outputs into monotonic digital codes; in this way, an effective resolution of 6 bits (at very high speed) was obtained from 255 low-precision comparators. The mathematical background of optimal ADC post-correction, as well as an example using experimental ADC data are given in [44]. A new architecture of digitally corrected ADCs called "beta converters"—based on the redundant representation of a real number with respect to a base $\beta \in (1, 2)$, instead of a binary representation with base $\beta = 2$—are introduced in [9]. The effects of imperfect quantizers in ADCs are mathematically analyzed in [10]. In addition, the robustness of $\Delta\Sigma$-ADCs to component mismatch is studied which leads to a new topology called "filtered $\Delta\Sigma$ modulator"[1]. The state of the art of error compensation in ADCs is summarized in [2].

In this chapter we point out that the effective resolution of suitably corrected ADCs is virtually independent of the comparator mismatch.

---

[1] We would like to thank Prof. Dr. H. Bölcskei and Dr. G. Ungerboeck for pointing out the publications by Daubechies et al.

In consequence, it seems attractive to deliberately use "cheap and dirty" (i.e., small) comparators that are fast without using much power. We illustrate this approach with measurements from an integrated flash ADC containing 256 low-precision comparators that achieves an effective resolution of almost 7 bits.

This chapter is structured as follows: after a brief introduction to analog-to-digital conversion in Section 4.2, we describe the principle of the linearized ADC and give some simulation results from a simple model in Section 4.3. In Section 4.4 we present a chip that was actually manufactured, and the corresponding measurements are reported in Section 4.5. We conclude this chapter with conclusions and an outlook in Section 4.6.

4.2 Introduction to A/D-Conversion

An analog-to-digital converter transforms a continuous-amplitude, continuous-time (analog) input signal into a discrete-amplitude, discrete-time (digital) output signal. The process leading to a discrete-amplitude signal is called *quantization*, whereas the process leading to a discrete-time signal is named *sampling*.

A general scheme of an ADC is shown in Figure 4.1. The low-pass filter assures that no frequencies above the Nyquist frequency $f_\mathrm{s}/2$ are present in the signal to be sampled by the sampling circuit. The following block quantizes the sampled signal, and the decoder assigns a digital number to each quantized signal value.

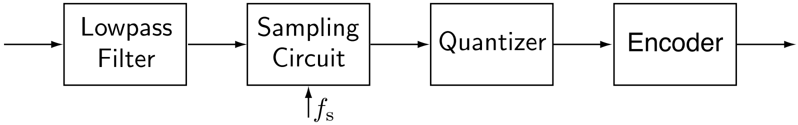


Figure 4.1: Elements of an analog-to-digital converter.

Despite the fact that sampling—especially for high frequencies— is a very difficult and important task, we neglect it here and only examine amplitude quantization.

For the measurements and simulations presented in this chapter, the n-bit ADC contains 2^n comparators and it produces 2^n+1 quantized output values[2]. The "analog input/digital output" characteristic of an ideal 2-bit ADC with 4 comparators and according to the following definitions is shown in Figure 4.2.

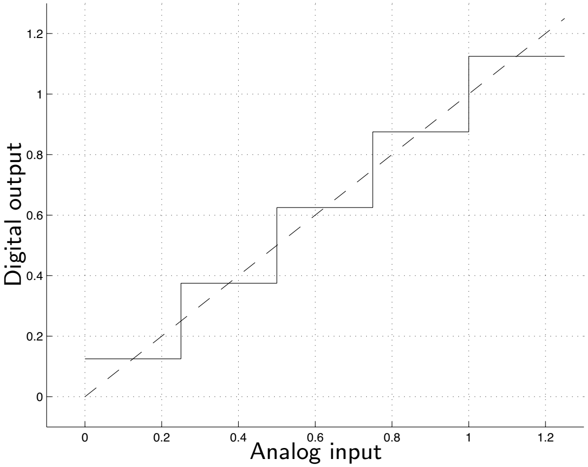


Figure 4.2: Ideal transfer characteristic of an ADC with 4 comparators. The input range of this ADC is $[0 \dots \frac{2^n+1}{2^n}]$, with $n = 2$.

Definition 4.1 (Threshold voltages of an n-bit ADC). *The 2^n ideal threshold voltages of an n-bit ADC are defined as:*

$$V_{\text{threshold},i} = V_{\text{ref, low}} + (V_{\text{ref, high}} - V_{\text{ref, low}}) \cdot \frac{i}{2^n}, \; i = 1 \dots 2^n \quad (4.1)$$

Definition 4.2 (Input range of an n-bit ADC). *In order to keep the quantization error constant over all input-segments, we define the input range of the ADC to depend on the number of comparators. The input range for an ADC with 2^n comparators is $[0 \dots \frac{2^n+1}{2^n}]$.*

Definition 4.3 (RMS quantization error Q). *The rms quantization*

---

[2] In most publications, an n-bit ADC contains only $2^n - 1$ comparators producing 2^n quantized output values. The disadvantage of this common definition is that the quantization error computed between two adjacent threshold voltages is not constant for all thresholds.

error Q is defined as follows:

$$Q^2 \triangleq \text{average over whole input range of} \quad \big((\text{analog input}) - \text{DAC}_{\text{ideal}}(\text{digital output})\big)^2. \tag{4.2}$$

Remark 4.4 (RMS quantization error of an ideal ADC). *The rms quantization error of an ideal n-bit ADC with bin width LSB is*

$$Q_{\text{ideal}} = \frac{2^{-n} \cdot (\text{“input range”})}{\sqrt{12}} = \frac{\text{LSB}}{\sqrt{12}}. \tag{4.3}$$

We can now transform the previous equation and solve it for n, the number of bits.

Definition 4.5 (Effective resolution of an ADC). *The effective resolution (in bits) of an ADC, where Q is the rms quantization error, is defined as follows:*

$$\text{Res}_{\text{eff}} = \frac{-\log(\frac{\sqrt{12}Q}{\text{“input range”}})}{\log(2)} \quad [\text{bits}] \tag{4.4}$$

It can be seen from (4.4) that in order to maximize the effective resolution, the quantization error needs to be minimized.

Definition 4.6 (SNR, SNDR). *The signal-to-noise ratio (SNR) is the ratio of the signal power to the total noise power at the output, usually measured for a sinusoidal input. The signal-to-noise-and-distortion ratio (SNDR) is the ratio of the signal power to the total noise and distortion power at the output when the input is a sinusoid. For oversampled converters, SNR is defined as the ratio of the signal power to only the in-band uncorrelated noise power, whereas SNDR also accounts for the in-band harmonic distortions and spurious tones.*

Remark 4.7 (ENOB vs. effective resolution). *The effective number of bits (ENOB) is usually defined by the following equation:*

$$\text{ENOB} = \frac{\text{SNDR}_p - 1.76}{6.02}, \tag{4.5}$$

where SNDR_p *is the peak SNDR of the converter expressed in decibels [54]. As we do not consider the time-behavior of the Nyquist-rate converters, i.e., no errors introduced by sampling is considered, we do not like to use the term ENOB and prefer the more theoretical definition for “effective resolution” (see Definition 4.5) as a measure.*

The subtrahend “−1.76” appears in (4.5) because the input signal is a sinusoid (instead of a ramp-signal with a uniform distribution over the full input range assumed implicitly in Definition 4.5 for the “effective resolution”). This subtrahend is only exact for a converter with a quantization error that is constant over all input ranges (i.e., the quantization error created in the mid-range of the converter is about the same as the error introduced from the outer range of the converter).

4.3 The Principle

For the sake of concreteness, we will focus on flash ADCs as shown in Figure 4.3 consisting of a bank of “cheap and dirty” (i.e., small) comparators followed by a digital correction.

Let n be an integer and let $m = 2^n - 1$ be the number of comparators. The digital correction in our ADC consists of two parts. First, the (non-monotonic) comparator outputs are converted into monotonic digital n-bit codes[3] (as in [52]). Then follows an $(m + 1) \times L$-bit static look-up table that implements a finer correction: for each of the $m + 1$ quantization bins, the look-up table stores the average of the (measured) upper threshold and the (measured) lower threshold of the bin [37], cf. Figure 4.4; this real number is represented by an L-bit number with $L > n$. (As we shall see, $L = n + 2$ or $L = n + 3$ will suffice in most cases.)

Obviously, this approach requires measurements of each individual chip. Many ways have been proposed to carry out on-chip calibration of ADCs, e.g., [2, 30]. Although the design of the (analog and digital) circuitry to enable such calibration is not trivial, we will not consider this topic any further.

The performance that can be achieved by such an approach may be obtained by the following simple model. Assume that each comparator threshold is generated randomly by adding a zero-mean Gaussian random variable with variance σ_ε^2 to the nominal (ideal) threshold position. (Comparators whose threshold happens to fall outside the nominal range of the input voltage are simply discarded.)

---

[3] With the ADC definitions of the previous chapter, i.e., by defining an n-bit ADC to have 2^n comparators, one more bit is necessary for the monotonic digital code.

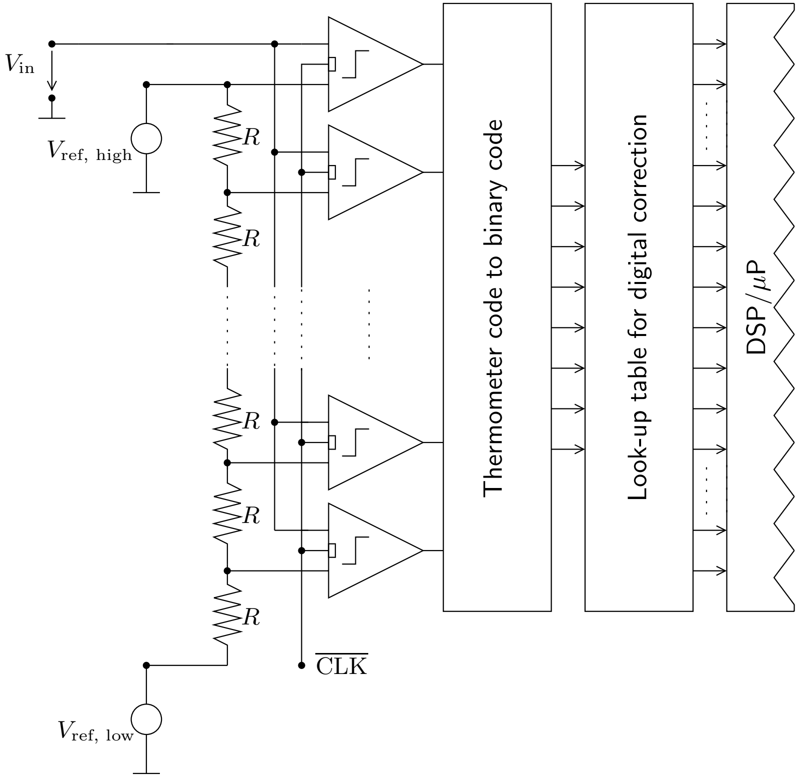
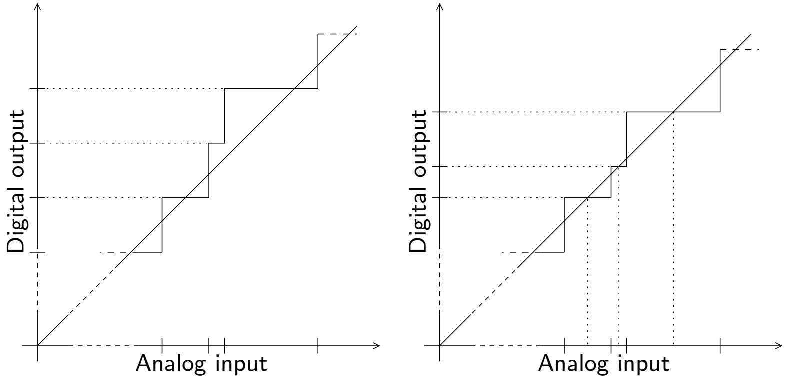


Figure 4.3: The complete ADC setup with a comparator bank, conversion from a (non-monotonic) thermometer code to a (monotonic) binary code, and the look-up table.



(a) Conventional digital output **(b)** Proposed digital output

Figure 4.4: Input-output characteristics of an ADC with comparator mismatch: (a) conventional digital output (digital bins with constant height); (b) proposed digital output.

The average performance of such "random" ADCs is shown in Figure 4.5; the (average) effective resolution (see Definition 4.5) as a function of σ_ε^2 with the number of comparators (2^6, 2^8, 2^{10}, and 2^{12}) as a parameter is shown.

As is obvious from Figure 4.5, the (average) effective resolution (as defined in Definition 4.5) of a properly corrected "random" ADC is virtually independent of the mismatch variance σ_ε^2, unless σ_ε^2 is very small, i.e., for very large comparators. This means that instead of using the nominal number of high-precision and large comparators for an ADC, twice as many minimal-sized comparators can be integrated to attain a similar effective resolution. Furthermore, it can be shown analytically that for corrected ADCs a limit on the loss of effective resolution exists: the expected value of the effective resolution of a digitally corrected ADC with 2^n comparators that are strongly afflicted by mismatch amounts to

$$\mathrm{E}\left[\mathrm{Res}_{\mathrm{eff},\sigma_\varepsilon \gg 0}\right] \approx \frac{\log(2^n)}{\log(2)} - \frac{\log(\sqrt{6})}{\log(2)} \approx n - 1.3 \quad [\text{bits}]. \tag{4.6}$$

The derivation of (4.6) is presented in Appendix B. In the simulation setup leading to Figure 4.5, the comparators with threshold voltages outside the nominal range are simply discarded. In this figure, the limit on the effective resolution does not tend to the value proposed in (4.6), but it lies slightly below.

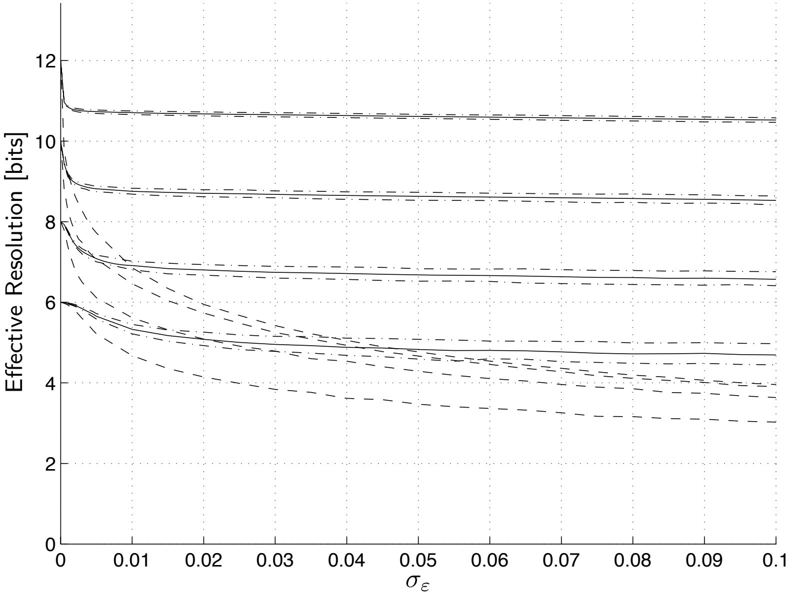


Figure 4.5: Average effective resolution of “random” ADCs. Dashed lines: conventional digital output (see Figure 4.4(a)). Solid lines: proposed digital output (see Figure 4.4(b)). Dash-dotted lines: 10 and 90 percentiles for proposed digital output.

4.4 Design Example

In order to demonstrate the concept, a flash ADC with 256 comparators was implemented in a 0.25-μm BiCMOS process (IBM 6HP) [51] using only CMOS transistors. The chip was designed for a supply voltage of $V_{\mathrm{dd}} = 1.8$ V. The chip also contains a 256-to-8 bit multiplexer, but the digital correction is not part of the chip and was realized in software.

The resistor ladder (which generates the reference voltages for the comparators) uses p+ polysilicon resistors; the width of each resistor is $W_R = 6\ \mu$m and the length of each resistor is $L_R = 3\ \mu$m.

The nominal resistance of such a resistor is $R = 143\ \Omega$ and the resistance of the whole ladder is $R_{\text{ladder}} = 36.6\ \text{k}\Omega$. The two voltages $V_{\text{ref, high}}$ and $V_{\text{ref, low}}$ determine the input range of the ADC; in all our simulations and measurements, we chose $V_{\text{ref, high}} = 1.5$ V and $V_{\text{ref, low}} = 0.5$ V. With a voltage difference of 1 V, the current through the ladder is only 27.3 μA.

The comparator circuit (taken from [6]) is shown in Figure 4.6. Other than in [6], we use very small transistors (as indicated in Figure 4.6). For example, the two input transistors of the differential pair have width $W_{\text{comp}} = 1\ \mu$m and length $L_{\text{comp}} = 0.5\ \mu$m, which is only slightly larger than the minimal transistor size of the process given by $W_{\text{min}} = 0.3\ \mu$m and $L_{\text{min}} = 0.24\ \mu$m. Not only are the transistors of very small size, but the design is even non-symmetric and only one output inverter is attached to the cross-coupled transistor pair.

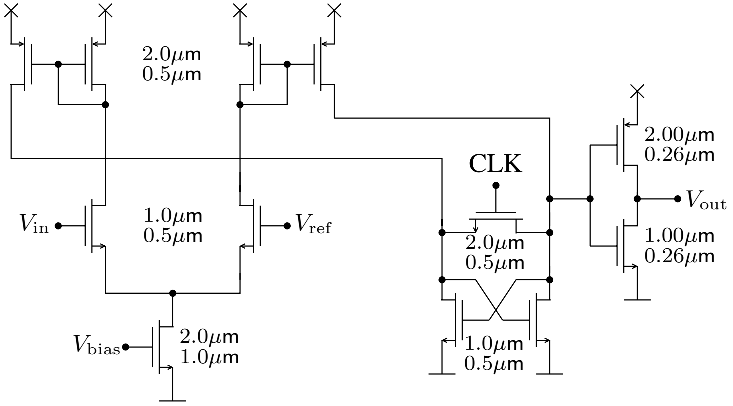


Figure 4.6: Schematic of the comparator. The numbers next to the transistors represent the width and the length of the corresponding transistor.

4.5 Measurements

10 chips in DIL-24 packages were available for measurements. The threshold of every comparator was measured with a custom-built measurement device using 12-bit ADCs. The measurement equipment was built by Patrik Strebel who also carried out all the measurements.

4.5.1 The Comparators

A statistics for all comparators was compiled. All measured comparator thresholds $V_{\text{meas},i}$ (with the offset $V_{\text{ref, low}} = 0.5$ V subtracted, and amplified by a factor of $\frac{257}{256}$) were compared with the thresholds $V_{\text{threshold},i}$ of the comparators of an ideal 8-bit ADC with an input range of $[0 \ldots \frac{257}{256}$ V] and threshold voltages as given in Definition 4.1:

$$V_{\text{threshold},i} = V_{\text{ref, low}} + (V_{\text{ref, high}} - V_{\text{ref, low}}) \cdot \frac{i}{2^8}. \tag{4.7}$$

Table 4.1 presents a list of means m_ε and standard deviations σ_ε of the comparators' threshold errors $\varepsilon_i = V_{\text{threshold},i} - V_{\text{meas},i}$ for each chip separately.

Chip:	m_ε $[mV]$	m_ε [LSB$_8$]	σ_ε $[mV]$	σ_ε [LSB$_8$]
1	1.29	0.33	14.74	3.77
2	-0.51	-0.13	16.26	4.16
3	-4.65	-1.19	16.36	4.19
4	-0.34	-0.09	15.59	3.99
5	-0.23	-0.06	14.77	3.78
6	-0.70	-0.18	14.91	3.82
7	-1.17	-0.30	15.97	4.09
8	-0.81	-0.21	15.42	3.95
9	-1.29	-0.33	15.64	4.00
10	0.64	0.16	14.84	3.80

Table 4.1: Mean m_ε and standard deviation σ_ε of the threshold errors for all comparators of one chip.

The mean and the standard deviation of the threshold errors—expressed in volts as well as in LSBs ($1\ \mathrm{LSB}_8 = 2^{-8} \cdot (V_{\mathrm{ref,\ high}} - V_{\mathrm{ref,\ low}}) = \frac{1}{256}\ \mathrm{V} \approx 3.91\ \mathrm{mV}$)—of the comparators over all chips amount to

$$m_\varepsilon = -0.78\ \mathrm{mV} = -0.20\ \mathrm{LSB}_8 \tag{4.8}$$

$$\sigma_\varepsilon = 15.51\ \mathrm{mV} = 3.97\ \mathrm{LSB}_8. \tag{4.9}$$

Note that, according to a popular rule of thumb, $\sigma_\varepsilon \leq \frac{1}{6}$ LSB is required in order to achieve a reasonable yield with an uncorrected ADC [57]. From (4.9), the accuracy of our converters would thus be adequate for a 4-bit converter. From Figure 4.5 (with $\sigma_\varepsilon = 0.0155$), however, we can expect our chips to yield an effective resolution of about 7 bits, which will be confirmed below.

The statistics of the offset voltages of all comparators for each chip separately, as well as for all chips together, are shown in Figure 4.7. The Gaussian curves with the mean and standard deviation taken from (4.8) and (4.9) are plotted for reference.

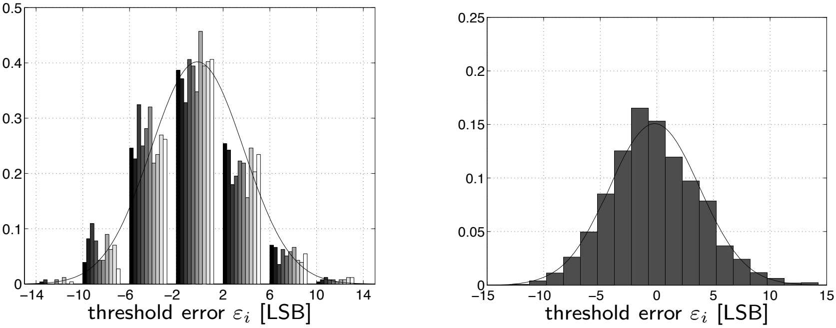


Figure 4.7: Distribution of threshold errors. Left: all 10 chips individually. Right: all 10 chips together.

4.5.2 The Complete ADC

The ADC chips do not contain any logic circuitry to convert the thermometer code into a binary value, therefore, post processing (bubbleerror correction [33], offset- and gain-correction [59]) was done in software. For each chip the digital correction was carried out as follows.

- The comparators were rearranged in order to have a monotonic output.
- Gain correction and offset correction was applied, resulting in an ADC with the first comparator switching at an input voltage of $V_{\text{in}} = 2^{-8}$ V, the last one at $V_{\text{in}} = 1$ V (see Definition 4.1). The input range of the ADC was defined according to Definition 4.2 as $[0 \text{ V} \ldots \frac{2^8+1}{2^8} \text{ V}]$.
- The mean value of each bin was identified and stored in a correction table as described in Section 4.3.

For comparison, the (conventional) uncorrected digital output was also recorded. For chip 4, the chip with the worst characteristics, both the calibrated (as in Figure 4.4(b)) and conventional (as in Figure 4.4(a)) input-output characteristics and quantization error spectrums are plotted in Figures 4.8 and 4.9, respectively.

As in Section 4.3 we will measure the performance of the ADCs in terms of their effective resolution, cf. Definition (4.5). (It should be noted that the popular "differential nonlinearity" (DNL) and "integral nonlinearity" (INL), which describe the deviation of the thresholds from their ideal positions, do not appear to make sense for our ADCs.)

The effective resolution of all 10 chips is listed in Table 4.2. Also shown in Table 4.2 is the rms quantization error Q_{calib} of the ADC with corrected digital output, as well as the rms quantization error Q_{raw} of the ADC with conventional uncorrected digital output. Note that the effective resolution varies from chip to chip, but is close to 7 bits for all 10 chips.

The effective resolution in Table 4.2 is based on a digital correction table with full measurement precision. In Table 4.3, the effective resolution (of chips 1 and 2) is shown for a (corrected) digital output using $L = 8, 9, \ldots, 12$ bits. Note that $L = 10$ suffices to essentially get the maximum possible effective resolution.

By assuming that there is no distortion, $\text{SNR}_{\text{calib}}$, the SNR computed from the quantization error spectrum shown in Figure 4.9, coincides with

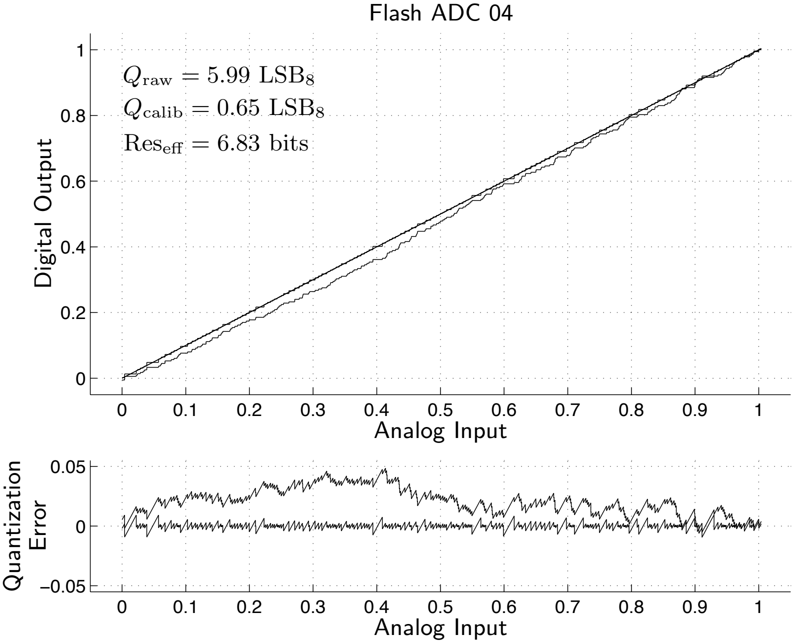


Figure 4.8: Top: input-output characteristics of the ADC chip 4, both conventional and corrected. Bottom: quantization error of the ADC with conventional and corrected output.

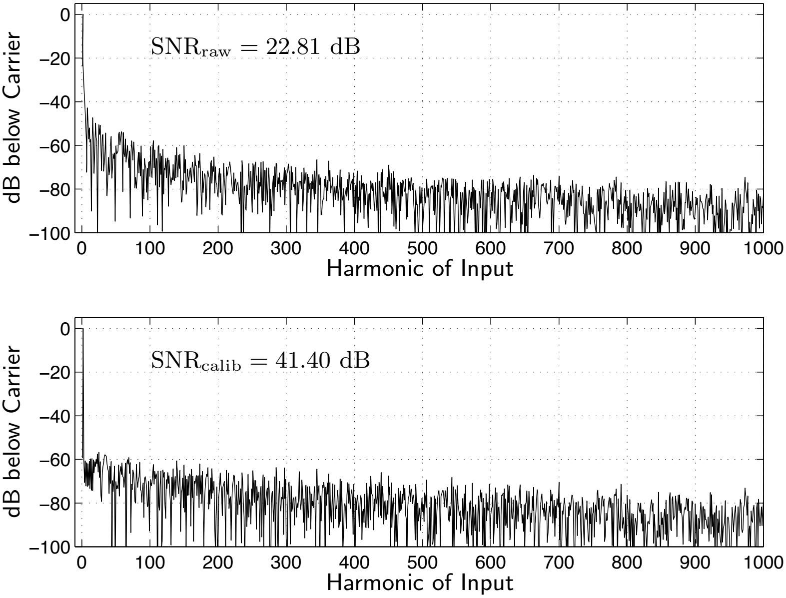


Figure 4.9: Quantization error spectrum for the uncalibrated (top) and calibrated flash ADC chip 4 assuming ideal sampling.

Chip:	Q_{raw} [LSB$_8$]	Q_{calib} [LSB$_8$]	effective Resolution [bits]
1	1.97	0.64	6.84
2	5.49	0.67	6.79
3	4.63	0.72	6.68
4	5.99	0.65	6.83
5	3.85	0.64	6.85
6	4.51	0.57	7.01
7	4.29	0.60	6.95
8	4.83	0.63	6.86
9	3.08	0.58	6.99
10	5.86	0.60	6.94

Table 4.2: The rms quantization error and effective resolution.

Chip:	$Q_{8\text{b}}$	$Q_{9\text{b}}$	$Q_{10\text{b}}$	$Q_{11\text{b}}$	$Q_{12\text{b}}$	Q_{calib}
1 [LSB$_8$]:	0.70	0.66	0.65	0.64	0.64	0.64
4 [LSB$_8$]:	0.71	0.67	0.65	0.65	0.65	0.65

Table 4.3: The rms quantization error for the digital output with $L = 8 \ldots 12$ bits of two example chips.

the SNDR (or it can be seen as an upper bound for the SNDR) and can be used to determine the ENOB:

$$\text{ENOB} = \frac{\text{SNDR}_p - 1.76}{6.02} = 6.58 \text{ bits} \tag{4.10}$$

For the effective resolution, a ramp-input was considered, whereas for the ENOB a sinusoid was used as input signal. The error introduced by quantizing signals in the middle of the ADC's input range is less than the error introduced by quantizing signals in the outer skirts of the ADC's input range, therefore, the effective resolution (given in Table 4.2) and ENOB do not coincide exactly.

4.6 Conclusions and Outlook

There is an increasing awareness that A/D-converters can be built using low-precision comparators with “random” thresholds. We have pointed out that—nearly independent of the comparators’ mismatch—the effective output resolution of the calibrated ADC is only approximately 1 bit smaller than the output resolution of an ADC with high-precision comparators.

The digital correction of such converters should produce the mean value of each quantization bin (which is a real number), or some adequate approximation thereof. We believe that this digital correction can easily be implemented in the ADC’s subsequent stage, which usually is a microprocessor or a digital signal processor.

As an example, we presented a chip with 256 low-precision converters that achieves a static precision of almost 7 bits.

Such ADCs show a very high linearity after the digital correction, and they can therefore also be used as part of a multi-bit sigma-delta ADC, where linearity is essential.

Chapter 5

D/A-Conversion with Imprecise Elements

5.1 Introductory Remarks

Digital-to-analog converters (DACs) are a key component of any modern communication system. Not only as converters in itself, interfacing the digital with the analog world, but also as part of analog-to-digital converters (e.g., oversampling or successive-approximation ADCs) do they play an important role. The performance of the before mentioned ADCs even depends mainly on the linearity and precision of the DAC. For current-steering DACs the trade-off between high accuracy and speed is given through the size of the current sources: highly accurate current sources require large transistors, and large transistors are slow (assuming a constant power consumption) [57]. Without introducing any post correction or self-calibration (e.g., component trimming or dynamic element matching), the maximum achievable resolution (for an ASIC in a present VLSI technology) is limited to $\mathrm{Res}_{\text{eff.}} = 12$ bits [61]. In this chapter, a digital correction technique to enhance the effective output resolution of DACs is proposed.

The chapter is structured as follows. A brief introduction to analog-to-digital conversion and some definitions of performance measures are given in Section 5.2. For concreteness sake, the principle of quantization error minimization is explained on the basis of current-steering DAC simulations in Section 5.3.The application of corrected DACs in multi bit $\Delta\Sigma$-ADCs is discussed with simulation results in Section 5.5 before concluding the chapter with an outlook in Section 5.6.

5.2 Introduction to D/A-Conversion

A digital-to-analog converter produces an analog output A that is proportional to a digital input D:

$$A = \alpha D, \tag{5.1}$$

where α is a gain-factor with a dimension, i.e., for current-steering DACs with currents as outputs, the dimension of α is [A] (Ampère), as D is a dimensionless quantity[1].

The ideal input-output characteristics of a 3-bit DAC is shown in Figure 5.1.

Same as for the analog-to-digital converter in Chapter 4 we only consider the static behavior of the DAC. For the analysis of the DAC, as a performance measure, we choose the *effective resolution* (in bits), which is related to the DAC's quantization error.

Definition 5.1 (RMS quantization error of a DAC). *The rms quantization error of a DAC, Q_{DAC} is defined as follows:*

$$\begin{aligned} Q_{\mathrm{DAC}}^2 \triangleq\ & \text{average over whole input range of} \\ & \big(\mathrm{DAC}_{\mathrm{ideal}}(\text{digital input}) - (\text{analog output})\big)^2 \end{aligned} \tag{5.2}$$

This definition may seem odd at first, as we sweep over the ideally converted digital input. Here, as a "digital input", we assume a real number (i.e., a number of infinite resolution) in the range of $-\frac{1}{2^{n+1}}$ to $1 - \frac{1}{2^{n+1}}$[2].

---

[1] Although nonlinear DACs do exist, we only address linear DACs who's ideal behavior can be described by (5.1).

[2] With this definition of the input range, the size of each (digital) input interval "mapped" onto an analog output is the same for every output. If the input range were defined as $[0 \ldots 1]$, the two outermost outputs would not be flanked by identically-sized input intervals. For higher resolutions this difference is, however, marginal.

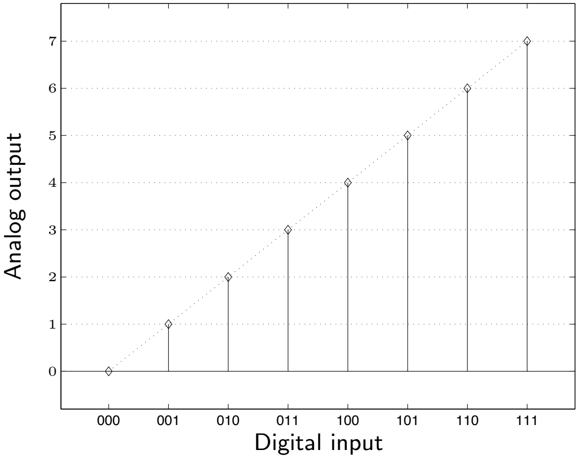


Figure 5.1: Ideal digital-input/analog-output characteristics of a 3-bit DAC.

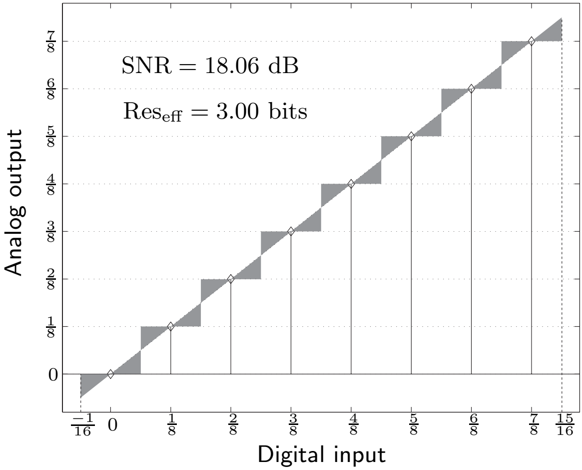


Figure 5.2: Input/output-characteristics of an ideal 3-bit DAC with the quantization error depicted.

Remark 5.2 (RMS quantization error of an ideal DAC). *Same as for the ideal n-bit ADC, the rms quantization error of an ideal n-bit DAC amounts to:*

$$Q_{\text{DAC, ideal}} = \frac{2^{-n} \cdot (\text{“output range”})}{\sqrt{12}} = \frac{\text{LSB}}{\sqrt{12}}. \tag{5.3}$$

This definition is in correspondence with [63].

Definition 5.3 (Effective resolution of a DAC). *The effective resolution (in bits) of a DAC, where Q_{DAC} is the quantization error from Definition 5.1, is defined as follows:*

$$\text{Res}_{\text{eff}} = -\frac{\log(\frac{\sqrt{12}Q_{\text{DAC}}}{\text{“output range”}})}{\log(2)} \quad [\text{bits}] \tag{5.4}$$

Again, in order to maximize the DAC's effective resolution, the quantization error needs to be minimized. For an ADC, the voltage thresholds at the input are determined by hardware and only the digital output can be adjusted. For the DAC, however, the possible outputs are fixed by the given hardware. The digital input leading to these analog outputs, on the other hand, can be adjusted to minimize the resulting quantization error.

The effective resolution computed for a non-ideal “3-bit” DAC (containing 3 non-ideal binary-scaled current sources) is illustrated in Figure 5.3. The non-ideality causes the outputs to be shifted away from the ideal outputs resulting in a decreased effective resolution; in this case the effective resolution is $\text{Res}_{\text{eff, nonideal}} = 2.48$ bits.

In order to minimize the quantization error of such a non-ideal DAC, digital correction (e.g., a look-up table) can be prepended to the DAC in order to adjust the digital input to the analog output (which is determined by the hardware). The corresponding input/output characteristics is illustrated in Figure 5.4. For this example, the DAC's digital correction has increased the effective resolution from $\text{Res}_{\text{eff, nonideal}} = 2.48$ bits to $\text{Res}_{\text{eff, corrected}} = 2.65$ bits, but is still well below the maximum achievable resolution (for 3 ideal binary-scaled current sources) of $\text{Res}_{\text{eff, ideal}} = 3.00$ bits. The gain in effective resolution may not be impressive, especially because the mismatch was not that big. However, other examples may show a larger improvement.

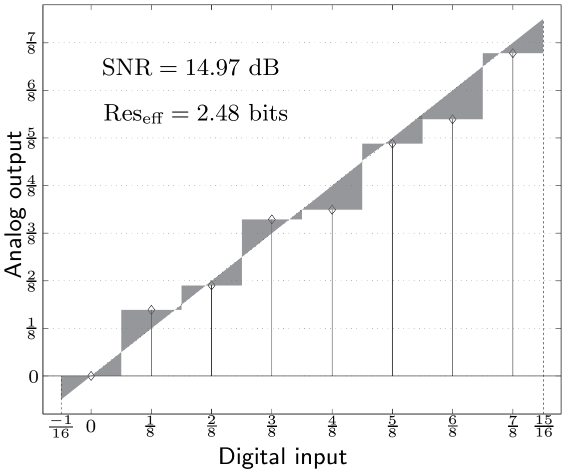


Figure 5.3: I/O characteristics and illustration of the quantization error of a non-ideal "3-bit" DAC.

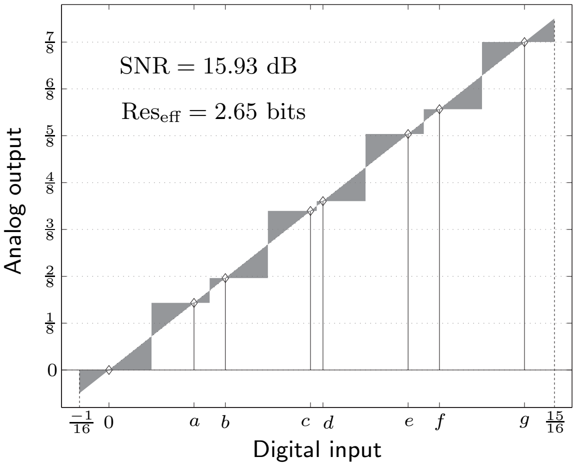


Figure 5.4: I/O characteristics and illustration of the quantization error of an non-ideal, corrected "3-bit" DAC.

It is clear that by adding redundant (non-ideal) current sources, higher effective resolutions can be achieved. As non-ideal current sources can be rather cheap in terms of chip-area and power-consumption[3], it seems to be an attractive approach to use redundant non-ideal (cheap) current sources instead of precise and large (expensive) ones.

In many analog chip realizations, the current-level of current sources is increased by multiplying unit current sources and connecting them in parallel. Therefore, a current source with an m-times larger current than a unit current source takes up m-times the area of the unit current source. We now define a DAC's "relative bit cost" as a further performance-measure.

Definition 5.4 (Relative bit cost of a DAC). *The area of a DAC's current sources, A_{DAC}, expressed in multiples of "unit current source area" A_{unit}, divided by the DAC's effective resolution, is defined as the DAC's "relative bit cost", RBC:*

$$\mathrm{RBC} = \frac{A_{\mathrm{DAC}}}{\mathrm{Res}_{\mathrm{eff}}} \left[\frac{A_{\mathrm{unit}}}{\mathrm{bits}}\right] \tag{5.5}$$

In the following section, we will apply a quantization error minimization technique to current-steering DACs and present some simulation results.

5.3 Introduction to Current-Steering Digital-to-Analog Converters

The topology of a general multi-bit DAC is shown in Figure 5.5. The analog output A_{out} is then, in a subsequent stage, offset- and gain-corrected.

In current-steering DACs, the analog output is attained through the superposition (or addition) of currents. The current sources can all be unit current sources (full redundancy), they can be binary-scaled current sources (no redundancy), or any other segmentation is possible [5, 58].

---

[3] In terms of current output, the "price" of current sources scale linearly with the current, i.e., a current source with a doubled current level can be seen as two unit current sources in parallel. Precision of current sources can be achieved by increasing the chip-area.

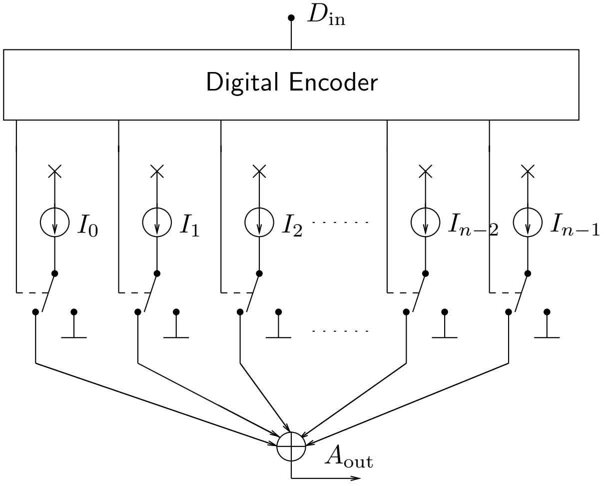


Figure 5.5: The topology of a general multi-bit DAC.

We will show that current source redundancy helps the DAC perform under mismatch.

The digital encoder can be seen as a look-up table that maps the digital input D_{in} to binary values that determine the positions of each current source switch and with that, it maps digital inputs to analog outputs: the digital encoder approximates the digital input D_{in} with the analog output A_{out}. The technique to minimize the rms quantization error (see Definition 5.3) is as follows:

a) Determine the set of all possible analog outputs.

b) For each digital input D_{in}, find the closest analog output (from the set of all possible outputs).

c) Store the switch-positions leading to this analog output in a look-up table.

Although in the literature noise and mismatch are sometimes confused, we emphasize that our technique can cope with mismatch (that can be measured prior to operation and is, for one specific realization, deterministic), but not with noise which is completely random. Often, converter mismatch, which may lead to undesired harmonic distortion, is transformed (e.g., by means of dynamic element matching) into noise.

5.4 Simulations of Digital-to-Analog Converters with Imprecise Current Sources

For the following simulations, current source mismatch is modeled as a Gaussian deviation (with a variance of σ^2, relative to the unit current source) from the nominal value [63] as shown in (5.6).

$$I_{\mathrm{out},n} = I_{\mathrm{nom.},n}(1+\varepsilon), \quad \varepsilon \sim \mathcal{N}(0,\sigma^2) \tag{5.6}$$

Furthermore, to compute the rms quantization error, the input word D_{in} is considered to be a real number, or a digital value with infinite precision. The assumption is made that all the 2^n possible outputs of the DAC (containing n current sources) can be measured to apply the previously discussed error minimization technique.

For DACs, static and dynamic characterizations exist. The static properties are given by the settled outputs and can be seen as a converter's best-case performance [63]. In this thesis only static properties of DACs are considered and further, we will focus on current source mismatch.

5.4.1 Variable Input Interval

For a DAC with non-ideal current sources, the center of the output region tends to be covered best by possible analog outputs, whereas the outer region is often only scarcely populated. (One might refer to the central limit theorem to argue that the density tends to a Gaussian distribution.) Not all the possible outputs need to be used for the DAC. The input interval can be chosen freely (as offset- and gain-correction is applied to the analog outputs) to optimize the performance (where the performance measure is the effective resolution).

As an example, all the possible output values obtained from the combination of 6 non-ideal unit current sources are presented in Figure 5.6. The effective resolution for this realization (including a digital correction based on the recipe given in Section 5.3) is $\mathrm{Res}_{\mathrm{eff}} = 4.83$ bits. By neglecting the 4 outermost possible outputs and rescaling the remaining outputs appropriately (as shown in Figure 5.7) the maximum effective resolution is augmented to $\mathrm{Res}_{\mathrm{eff}} = 4.91$ bits.

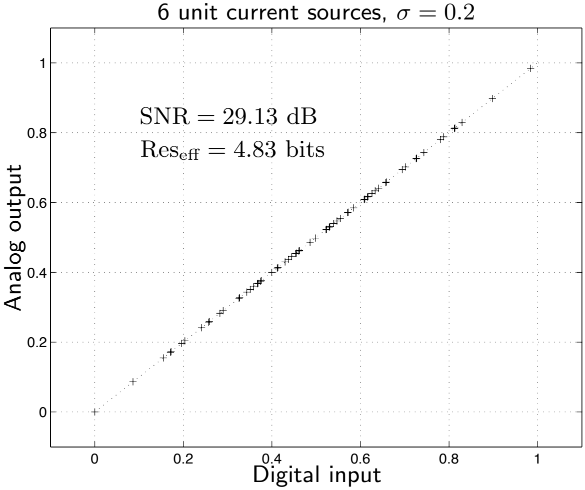


Figure 5.6: The possible output values stemming from 6 unit current sources with mismatch ($\sigma = 0.2$). The center region is covered denser than the outer region.

For the following simulation results, the ideal input interval (i.e., the input interval leading to the highest effective resolution) was determined numerically for each DAC realization by software.

5.4.2 Binary-Scaled Current Sources with Mismatch

With n ideal binary-scaled current sources an effective resolution of $\mathrm{Res}_{\mathrm{eff.}} = n$ bits can be achieved. If the current sources are non-ideal, i.e., the current sources are afflicted by mismatch, the effective resolution decreases, as can be seen in Figures 5.8 and 5.9 for 12 and 14 binary-scaled

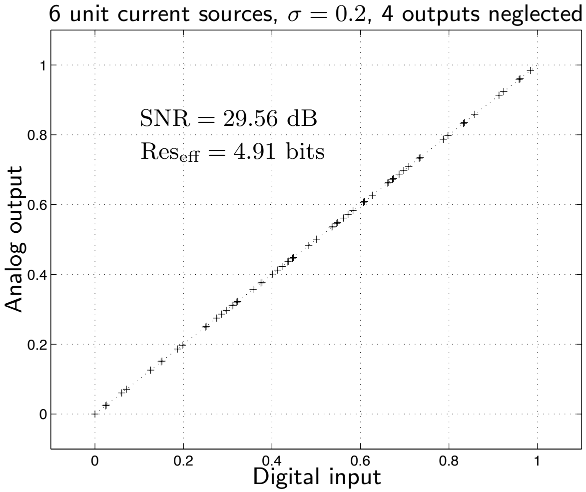


Figure 5.7: The possible output values stemming from 6 unit current sources with mismatch ($\sigma = 0.2$). 4 possible outputs are neglected and the output range is scaled appropriately.

current sources, respectively. As the DACs include digital correction, the effective resolution only degrades gradually with increased mismatch.

The "relative bit cost" (see Definition 5.4) of an ideal n-bit DAC with binary-scaled current sources is given in (5.7). We note that the DAC's area increases exponentially with each additional bit.

$$\mathrm{RBC_{bin.\text{-}scaledCS}} = \frac{2^n - 1}{n} \left[\frac{A_{\mathrm{unit}}}{\mathrm{bits}} \right] \tag{5.7}$$

current sources	$\sigma = 0$	$\sigma = 0.25$	$\sigma = 0.5$
12	341.3	372.5	408.8
14	1170.2	1331.0	1420.8

Table 5.1: Relative bit cost for a DAC with 12 and 14 binary-scaled current sources.

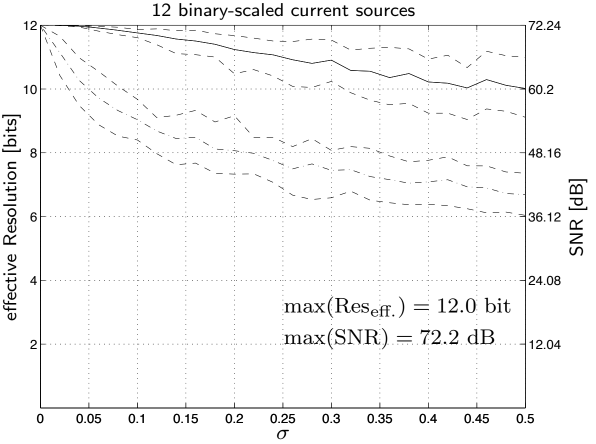


Figure 5.8: Maximum effective resolution (in dB) vs. current source mismatch for a digitally corrected DAC (solid line) and for a DAC without digital correction (dash-dotted line) containing 12 binary-scaled current sources. For both realizations, the 10-90-percentile lines are given (dashed lines).

5.4.3 Unit Current Sources with Mismatch

Instead of using binary-scaled current sources it is also possible to use unit current sources. (To achieve precise current levels and a high resolution, the LSB-part of a DAC is usually realized with unit current sources switched in parallel, instead of with one big current source.) The effective resolution of ideal unit current sources is very small, however, as can be seen in the following figures, mismatch *improves* the effective resolution of the corresponding digitally corrected DAC. The maximum effective resolution of a DAC with 12 and 14 mismatch-afflicted unit current sources (as well as the corresponding 10- and 90-percentile curves) is shown in Figures 5.10 and 5.11, respectively.

For m ideal unit current sources, the DAC's effective resolution amounts to $\mathrm{Res_{eff}} = \log_2(m+1)$; 15 ideal unit current sources are necessary for a 4-bit DAC. The "relative bit cost" of the DAC with m ideal current sources is given in (5.8).

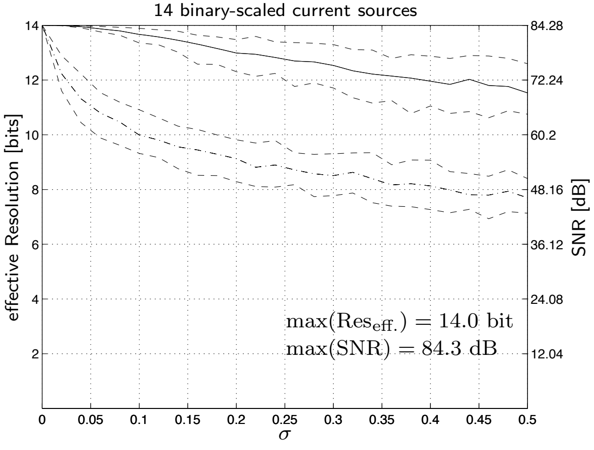


Figure 5.9: Maximum effective resolution (in dB) vs. current source mismatch for a digitally corrected DAC (solid line) and for a DAC without digital correction (dash-dotted line) containing 14 binary-scaled current sources. For both realizations, the 10-90-percentile lines are given (dashed lines).

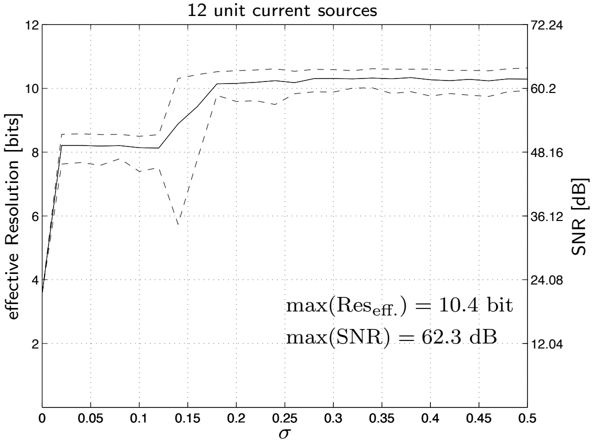


Figure 5.10: Maximum effective resolution (in bits) vs. current source mismatch for 12 unit current sources.

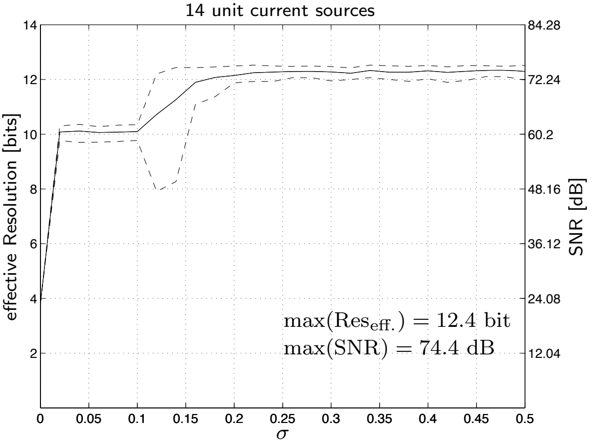


Figure 5.11: Maximum effective resolution (in bits) vs. current source mismatch for 14 unit current sources.

$$\mathrm{RBC}_{\mathrm{unitCS}} = \frac{m}{\log_2(m+1)} \left[\frac{A_{\mathrm{unit}}}{\mathrm{bits}}\right] \tag{5.8}$$

This amounts to the same number for ideal current sources as for the DAC with binary-scaled current sources. However, as can be seen in Figures 5.10 and 5.11, the effective resolution increases with augmented mismatch and the relative bit cost is decreased, as is shown in Table 5.2.

current sources	$\sigma = 0$	$\sigma = 0.25$	$\sigma = 0.5$
12	3.243	1.175	1.166
14	3.583	1.144	1.137

Table 5.2: Relative bit cost for a DAC with 12 and 14 unit current sources.

Comparing the relative bit cost of DACs with unit current sources with the relative bit cost of binary-scaled current sources may not seem fair in terms of absolute area-usage as the binary-scaled current sources take up much more space: an n-bit DAC with binary-scaled current sources takes up the same area as $2^n - 1$ unit current sources. However, the number of switches that need to be operated for a DAC with n unit current sources are the same as for the DAC with n binary-scaled current sources.

We have presented a way to obtain a DAC with high resolution on a small chip-area and with a very low relative bit cost. However, mismatch on the current sources is essential to achieve a high effective resolution.

5.4.4 Geometrically Scaled Current Sources with Mismatch

In the previous subsection, simulations showed that mismatch was necessary to obtain a high effective resolution for a DAC with unit current sources; with identical current sources only small effective resolutions can be achieved. It is, however, also possible to design current sources that have nominally slightly different current-levels.

In a first approach, simulations with geometrically-scaled current sources are presented. The m current sources have a nominal current of

$$c_{i,\mathrm{nom}} = \gamma^i, \quad i = 0 \ldots m-1. \tag{5.9}$$

Binary-scaled current sources can be seen as geometrically-scaled current sources with a factor of $\gamma = 2$, whereas unit current sources are scaled with a factor of $\gamma = 1$. For the simulation results presented in Figures 5.12 and 5.13, 12 and 14 current sources, respectively, scaled with $\gamma = 1.1$, were used.

We point out that for a DAC with geometrically-scaled current sources (and with $\gamma = 1.1$) no mismatch is necessary to achieve a high resolution on a small chip-area. The relative bit cost given in Table 5.3 is nearly constant over the whole range of current source mismatch.

The problem with such geometrically-scaled current sources (with non-integral factors and of minimal size) is that they are impossible to implement. We will therefore examine a feasible option to create nominally different current sources with minimal sizes; DACs with arithmetically-scaled current sources are examined.

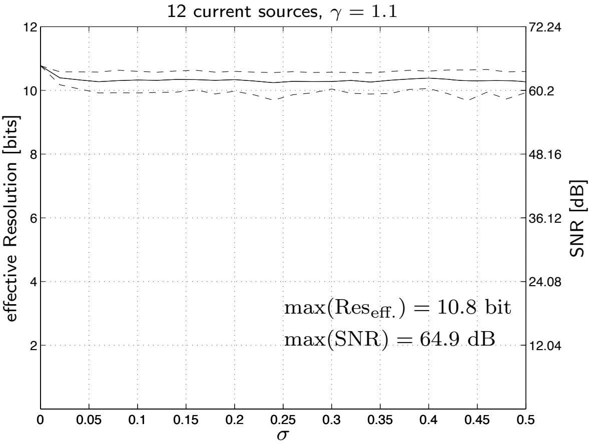


Figure 5.12: Maximum effective resolution (in bits) vs. current source mismatch for 12 geometrically-scaled current sources ($\gamma = 1.1$).

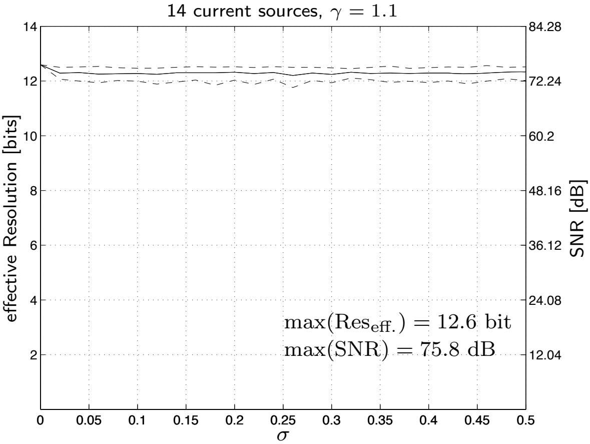


Figure 5.13: Maximum effective resolution (in bits) vs. current source mismatch for 14 geometrically-scaled current sources ($\gamma = 1.1$).

current sources	$\sigma = 0$	$\sigma = 0.25$	$\sigma = 0.5$
12	1.985	2.084	2.083
14	2.222	2.282	2.268

Table 5.3: Relative bit cost for a DAC with 12 and 14 geometrically-scaled current sources ($\gamma = 1.1$).

5.4.5 Arithmetically Scaled Current Sources with Mismatch

From the previous simulation results we have learned that it is best to create a number of current sources that have similar—but not identical—nominal current levels. A feasible approach is to create a DAC with arithmetically-scaled current sources; their nominal currents are as follows:

$$c_{i,\mathrm{nom}} = 1 + \Delta \cdot i, \quad i = 0 \ldots m - 1 \tag{5.10}$$

The simulation results of DACs with 12 and 14 arithmetically-scaled current sources (with $\Delta = 0.1$) are presented in Figures 5.14 and 5.15, respectively. For a mismatch above $\sigma = 0.03$ and a corrected DAC with m current sources, the effective resolution is nearly constant and lies slightly above $m - 2$ bits.

The relative bit cost for such a configuration is listed in Table 5.4 for various levels of mismatch. Above the mismatch-threshold, $\sigma = 0.03$, a mismatch number that is easily surpassed with minimal-sized current sources in a modern VLSI process, the relative bit cost is nearly constant.

current sources	$\sigma = 0$	$\sigma = 0.25$	$\sigma = 0.5$
12	2.525	1.797	1.812
14	2.994	1.888	1.883

Table 5.4: Relative bit cost for a DAC with 12 and 14 arithmetically-scaled current sources ($\Delta = 0.1$).

In Subsection 5.4.1 we remarked that not the whole range of possible outputs needs to be considered for the DAC's optimal performance, i.e.,

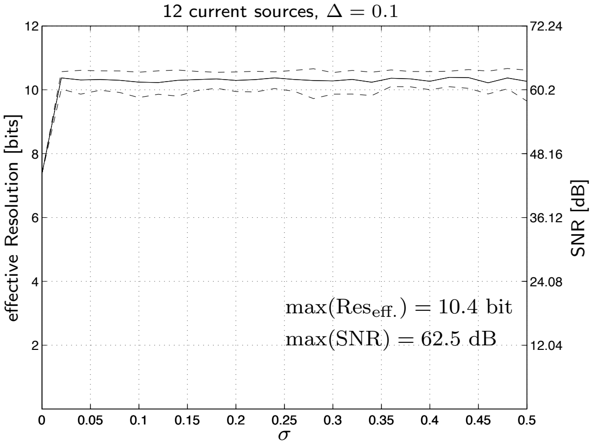


Figure 5.14: Maximum effective resolution (in dB) vs. current source mismatch for 12 arithmetically-scaled ($\Delta = 0.1$) current sources.

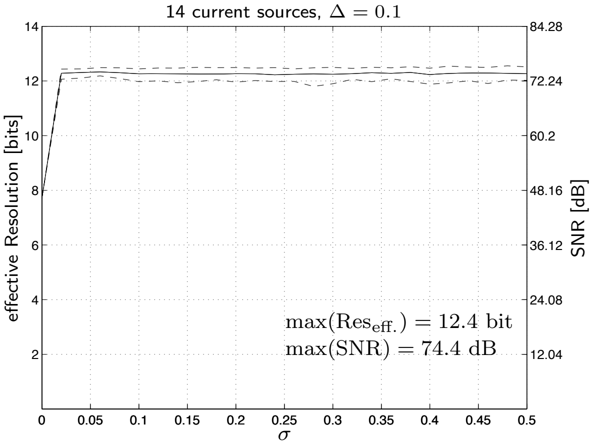


Figure 5.15: Maximum effective resolution (in dB) vs. current source mismatch for 14 arithmetically-scaled ($\Delta = 0.1$) current sources.

to obtain the maximum effective resolution the outermost possible outputs should be neglected. This also holds for a DAC with arithmetically-scaled current sources. Figure 5.16 shows plots of the maximum achievable effective resolution vs. the percentage of possible outputs used. For both cases, with a current source mismatch of $\sigma = 0.25$ and $\sigma = 0.5$, only about 85% of all possible outputs are used to achieve the highest possible effective resolution.

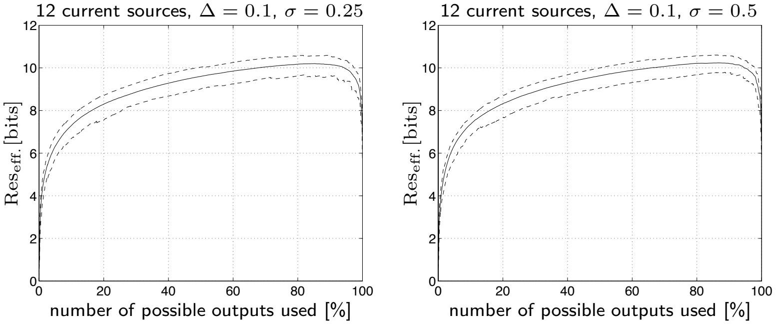


Figure 5.16: Percentage of possible outputs used vs. maximum effective resolution for one realization of a DAC with 12 arithmetically-scaled non-ideal current sources ($\sigma = 0.25$, $\sigma = 0.5$).

For two DAC realizations with 12 arithmetically-scaled current sources with a mismatch of $\sigma = 0.25$ and $\sigma = 0.5$, the distribution of the possible outputs is shown in Figure 5.17. Although approximately 85% of all outputs are used, the output range covers only 38% (for $\sigma = 0.25$) or 46% (for $\sigma = 0.5$) of the full output range. In Figure 5.17, the output ranges leading to the highest effective resolution are marked with dashed lines. In a subsequent stage, the DAC's output range needs to be shifted and amplified to again cover the full output interval.

As the DAC with digital correction was designed to show ideal linearity (and no switching noise was considered), the output spectrum is not expected to reveal any harmonic distortion. Figures 5.18(a)–5.18(d) show the normalized power spectral density (PSD) plots for a DAC with 12 arithmetically-scaled non-ideal current sources with a mismatch of $\sigma = 0.25$ (top row) and $\sigma = 0.5$ (bottom row). The PSDs for DACs using all possible outputs (left) are compared to the PSD plots for DACs with an optimized output range; just by using the ideal output range

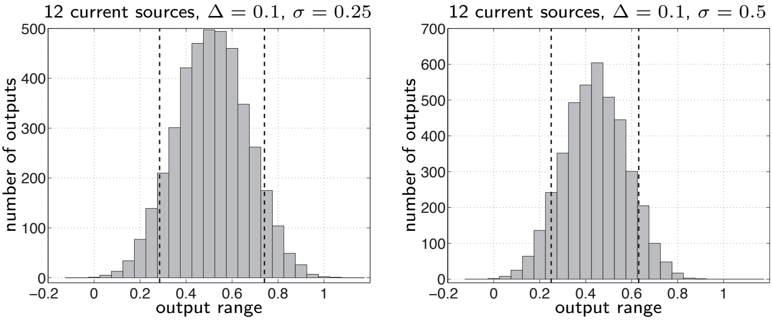


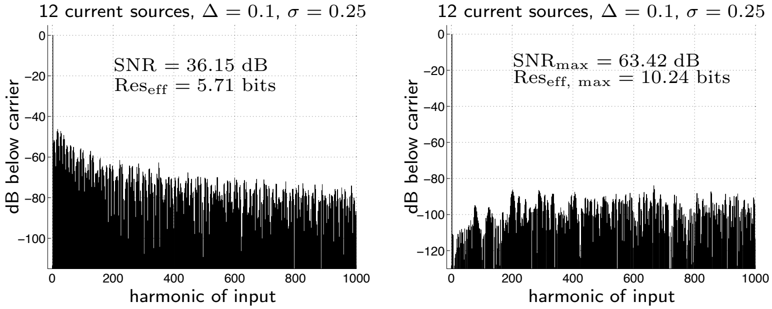
Figure 5.17: Outputs of a DAC with 12 arithmetically-scaled non-ideal current sources with $\Delta = 0.1$ and $\sigma = 0.25$ or $\sigma = 0.5$. Ideal output range maximizing the DAC's effective resolution.

and neglecting some of the possible outputs, the effective resolution can be improved by more than 4 bits.

5.5 ΔΣ-Converters with Corrected ADCs and DACs

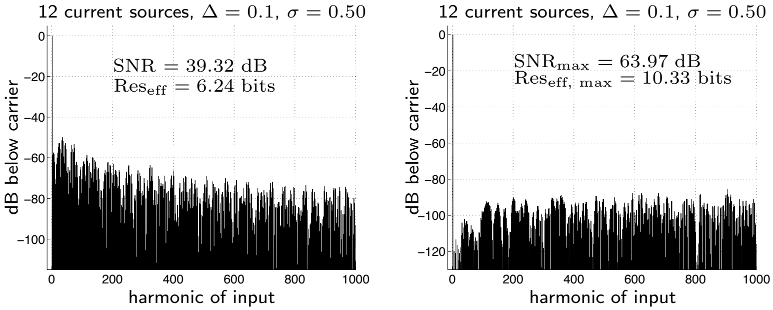
This section does not provide a thorough treatise on multi-bit converters, but rather a "proof of concept" for using digitally corrected low precision converters in ΔΣ-ADCs to reduce the oversampling ratio necessary for a desired resolution. After a brief introduction to ΔΣ-converters, simulation results for multi-bit ΔΣ-ADCs with ideal and non-ideal (but corrected) elements are provided.

The notion of using high sampling rates and low-resolution converters (even single-bit converters) to attain a high-resolution signal representation at the desired (lower) sampling rate, has been around for a long time. In the early years, the results were disappointing: the oversampling ratios necessary to reach a sufficient resolution (for speech encoding) were too high. Only later people realized that digital filtering (which was very expensive at that time) was necessary for a noise-shaping that could greatly improve the resolution-gain vs. the oversampling ratio [31].



(a) PSD of a DAC with all possible outputs used

(b) PSD of a DAC with scaled and optimized output range



(c) PSD of a DAC with all possible outputs used

(d) PSD of a DAC with scaled and optimized output range

Figure 5.18: Power spectral density plots for a DAC with 12 arithmetically-scaled non-ideal current sources and a mismatch of $\sigma = 0.25$ (5.18(a) and (b)) and $\sigma = 0.5$ (5.18(c) and (d)). For the plots in Figures 5.18(a) and (c), all the possible outputs were used, whereas for the plots in Figures 5.18(b) and (d), the output range was chosen in order to maximize the effective resolution.

Other than Nyquist-rate converters, ΔΣ-converters[4] rely on an oversampled input signal, as can be seen in Figure 5.19. The converter consists of a noise-shaping structure and an N-bit ADC in the forward path and an N-bit DAC in the feedback path. The name "ΔΣ" refers to the subtraction (Δ) of the feedback signal and the summation (Σ) in the noise-shaping part of the converter.

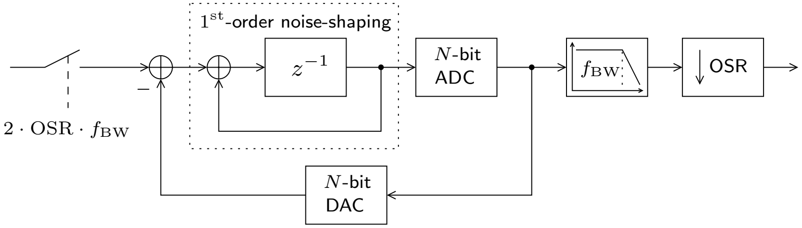


Figure 5.19: Schematic of a discrete time ΔΣ-ADC with first-order noise-shaping.

The signal-to-noise-and-distortion ratio (SNDR) of the ΔΣ-ADC depends on the oversampling ratio (OSR), the order of noise-shaping and the resolution of the internal converters. This dependency can be seen in (5.11) and (5.13) for 0-order and 1^{st}-order noise-shaping, respectively.

(5.12) and (5.14) show that 0.5 bits per octave, and 1.5 bits per octave, are gained for a converter without noise-shaping and with 1^{st}-order noise-shaping, respectively.

$$\text{SNDR}_{\text{p},0} = 6.02N + 1.76 + 10\log_{10}(\text{OSR}) \quad [\text{dB}] \tag{5.11}$$

$$\text{ENOB}_0 = N + 0.5\log_2(\text{OSR}) \quad [\text{bits}] \tag{5.12}$$

$$\text{SNDR}_{\text{p},1} = 6.02N + 1.76 - 5.17 + 30\log_{10}(\text{OSR}) \quad [\text{dB}] \tag{5.13}$$

$$\text{ENOB}_1 = N + 1.5\log_2(\text{OSR}) - 0.86 \quad [\text{bits}] \tag{5.14}$$

---

[4] In the literature, "ΔΣ converters" are sometimes referred to as "ΣΔ converters." The name can be used interchangeably. In this thesis, the traditional name "ΔΣ converter" is used.

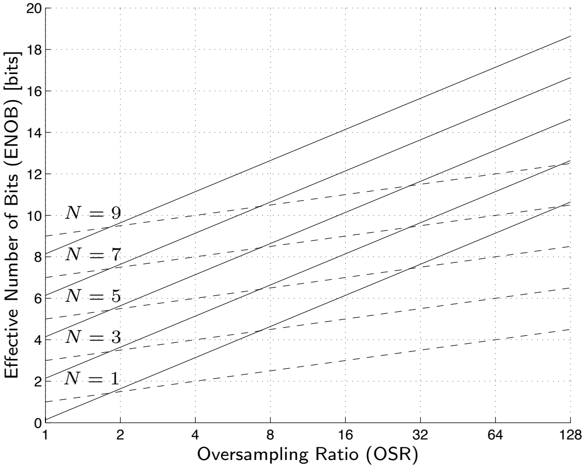


Figure 5.20: Effective number of bits vs. the oversampling ratio (OSR) of a ΔΣ-ADC with internal quantizers with various resolutions. Solid lines: ΔΣ-ADC with first-order noise-shaping, dashed lines: ΔΣ-ADC without noise-shaping.

The most crucial part in a ΔΣ-ADC is the DAC in its feedback-loop: the linearity of the DAC directly affects the resolution of the ΔΣ-ADC, whereas the error introduced by the internal ADC gets filtered with the noise-shaping filter.

The highest degree of linearity can be achieved by using a 1-bit converter, however, with the drawback of an increased operating speed which is necessary to attain a high resolution. A high oversampling ratio is not feasible if the desired bandwidth is already high. The oversampling ratio can be reduced by using highly linear multi-bit ADCs and DACs; ΔΣ-converters can then be used for higher bandwidths. The digitally corrected DACs introduced in this chapter feature highest linearity (as they were corrected for this purpose). They can, therefore, be used in a multi-bit ΔΣ-ADC.

The following setup was considered for all the following simulations: the discrete time ΔΣ-ADC presented in Figure 5.19 was built in software. The input to the converter is an oversampled full-swing sinusoid, the output is measured before the low-pass filter and the down converter.

The following parameters were chosen for the simulations:

- $f_{\mathrm{BW}} = 25$ kHz
- $\mathrm{OSR} = 16$
- $f_{\mathrm{sampling}} = 2 f_{\mathrm{BW}}\,\mathrm{OSR} = 800$ kHz
- $f_{\mathrm{in}} = 2.19$ kHz (367 samples/period)
- Kaiser-window, $\alpha = 14$

Various simulations were made with different internal ADCs and DACs. In a first step, to gain confidence in the simulation setup and for comparison reasons, ΔΣ-ADCs with ideal N-bit ADCs and DACs were considered. The expected SNDR for ΔΣ-ADCs with 1st-order noise-shaping can be computed with (5.13). The SNDRs and ENOBs for internal 6-bit and 7-bit converters are:

$$\begin{aligned} \mathrm{SNDR}_{\mathrm{p},1,N=6} &= 68.83\ \mathrm{dB}, & \mathrm{ENOB}_{1,N=6} &= 11.14\ \mathrm{bits} \\ \mathrm{SNDR}_{\mathrm{p},1,N=7} &= 74.85\ \mathrm{dB}, & \mathrm{ENOB}_{1,N=7} &= 12.14\ \mathrm{bits}. \end{aligned}$$

The power spectral density plots for these two configurations are presented in Figures 5.21 and 5.22, for $N = 6$ and $N = 7$, respectively.

These simulations provide a good estimate of the SNDR: the computed values obtained from spectral analysis differ less than 1 dB from the expected values.

Note that in the previous simulations the ADCs and DACs were not just ideal converters, but they even matched perfectly, i.e., the DAC reproduced the digital values provided by the ADC perfectly and did not introduce any further quantization error. In a practical system this perfect matching between ADC and DAC is very costly, not to say impossible to achieve. The performance of the ΔΣ-converter is greatly affected if the DAC in the feedback path is not ideal and stays uncorrected. Simulation results of a ΔΣ-converter containing an ideal 6-bit ADC and an uncorrected DAC with comparators with small mismatch reveal the loss in performance: for a realization of a ΔΣ-converter with a comparator mismatch of only $\sigma_\varepsilon = 1\%$, the effective number of bits of the ΔΣ-ADC is decreased from $\mathrm{ENOB}_{\sigma_\varepsilon=0} = 11.14$ bits to $\mathrm{ENOB}_{\sigma_\varepsilon=1\%} = 6.72$ bits.

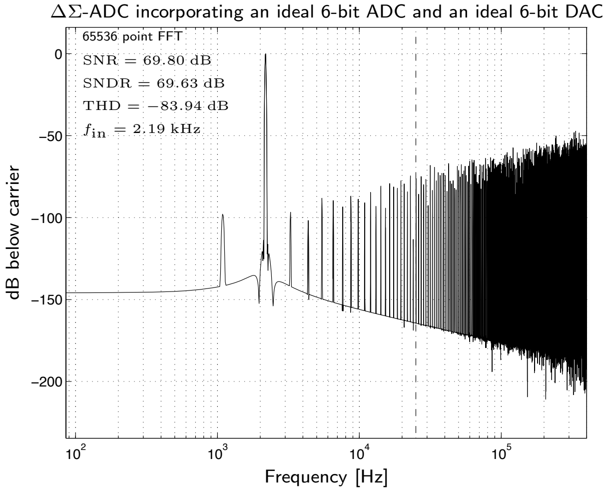


Figure 5.21: Power spectral density of a ΔΣ-ADC with ideal 6-bit ADC and DAC.

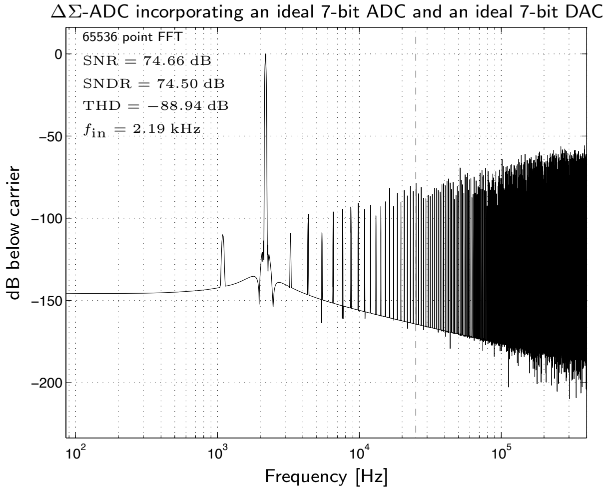


Figure 5.22: Power spectral density of a ΔΣ-ADC with ideal 7-bit ADC and DAC.

Therefore, with only minimal mismatch in the DAC, the resolution-gain due to oversampling is negligible (compared to the resolution of the ideal ADC, $N = 6$ bits, in the forward path). The corresponding power spectral density plot is given in Figure 5.23.

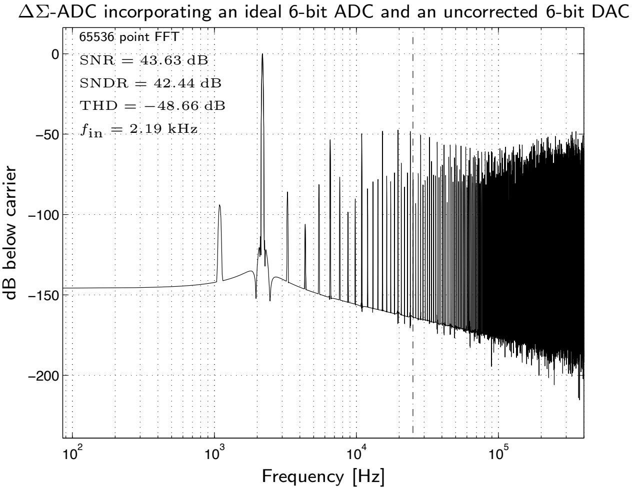


Figure 5.23: Power spectral density of a ΔΣ-ADC with an ideal 6-bit ADC and a non-ideal and uncorrected DAC ($\sigma_\varepsilon = 1\%$).

5.5.1 ΔΣ-Converter with a non-ideal ADC

We now incorporate a non-ideal, corrected ADC (discussed in Chapter 4) into the ΔΣ-ADC of Figure 5.19. This digitally corrected ADC has 256 comparators and reaches an effective resolution of $\text{Res}_{\text{eff}} = 6.84$ bits (the data was obtained from “chip 1” described in Section 4.5); the DAC is considered to perfectly match the non-ideal ADC, i.e., no further quantization error is introduced by the DAC.

The simulation result obtained from this setup is shown in Figure 5.24. The effective number of bits obtained from this simulation is ENOB = 10.94 bits, which is only slightly less than what was achieved with an ideal 6-bit ADC and DAC.

The performance of the attained ΔΣ-ADC is worse if the DAC does

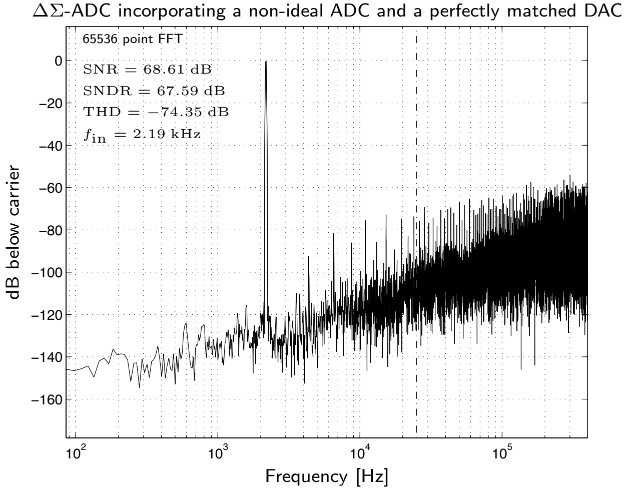


Figure 5.24: Power spectral density of a $\Delta\Sigma$-ADC with a non-ideal ADC and a perfectly matched DAC.

not perfectly match the non-ideal ADC (as it did in the previous simulation). Power spectral density plots resulting from the simulation of a $\Delta\Sigma$-ADC with a non-ideal ADC and N-bit DACs in the feedback path, are presented in Figures 5.25 and 5.26 for $N = 8$ bit and $N = 10$ bit, respectively. Although the non-ideal DAC only has 256 different output values and an effective resolution of $\text{Res}_{\text{eff}} = 6.84$ bits, the resolution of the ideal DAC needs to be higher: with an 8-bit DAC or a 10-bit DAC, the effective number of bits of the $\Delta\Sigma$-ADC amounts to ENOB = 9.61 dB (SNDR = 59.60 bits) or ENOB = 10.89 bits (SNDR = 67.32 dB), compared to ENOB = 10.94 bits (SNDR = 67.59 dB) of the perfectly matched DAC. In order for the DAC not to be the limiting factor, the DAC's resolution needs to be at least 10 bits. (This corresponds to the results for corrected ADCs with a limited number of output bits presented in Table 4.3.)

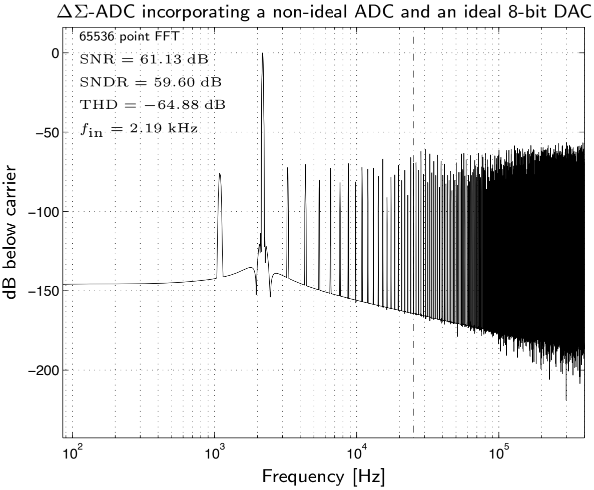


Figure 5.25: Power spectral density of a ΔΣ-ADC with a non-ideal ADC and an ideal 8-bit DAC.

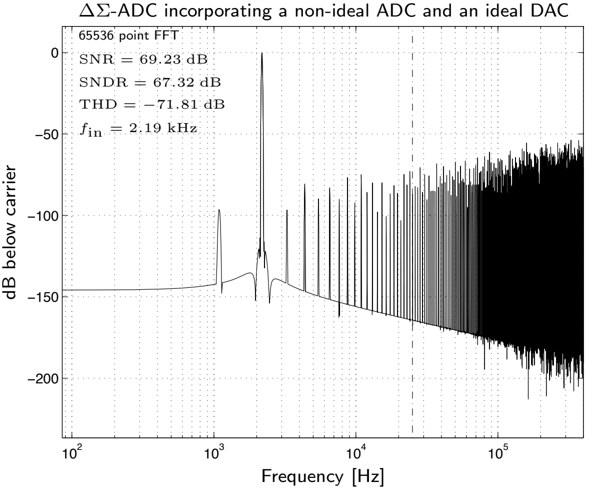


Figure 5.26: Power spectral density of a ΔΣ-ADC with a non-ideal ADC and an ideal 10-bit DAC.

5.5.2 $\Delta\Sigma$-Converters with a non-ideal ADC and a non-ideal DAC

For the following simulations, the $\Delta\Sigma$-ADC will be composed of a non-ideal ADC (256 comparators, $\text{Res}_{\text{eff}} = 6.84$ bits) and a non-ideal DAC.

For the power spectral density plot presented in Figure 5.27, the DAC is built with 9 (nearly) minimal-sized current sources (9 arithmetically-scaled current sources, $\Delta = 0.1$, $\sigma = 0.25$) and it has an effective resolution of $\text{Res}_{\text{eff}} = 7.89$ bits. At an oversampling ratio of OSR = 16, the $\Delta\Sigma$-ADC reaches an effective number of bits of ENOB = 9.39 bits (SNDR = 58.27 dB). This lies just slightly below the effective number of bits achieved by the $\Delta\Sigma$-ADC with an ideal 8-bit DAC in the feedback path (see Figure 5.25).

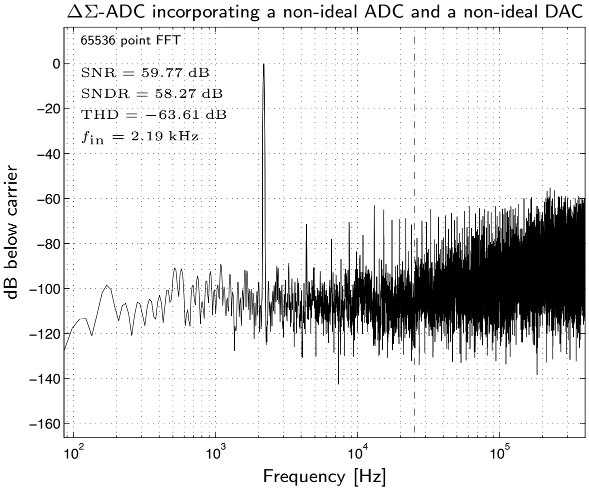


Figure 5.27: Power spectral density of a $\Delta\Sigma$-ADC with a non-ideal ADC and a non-ideal DAC with $\text{Res}_{\text{eff}} = 7.89$ bits.

12 (nearly) minimal-sized current sources leading to a DAC with an effective resolution of $\text{Res}_{\text{eff}} = 10.67$ bits are used in the feedback path of the $\Delta\Sigma$-ADC for the following simulation. The corresponding power spectral density plot is given in Figure 5.28, the effective number of bits, ENOB = 10.66 bits (SNDR = 65.94 dB), of this setup lies close to the performance achieved by the $\Delta\Sigma$-ADC with the same non-ideal ADC and

an ideal 10-bit DAC or a perfectly-matched DAC that does not introduce any further quantization error (see Figures 5.24 and 5.26).

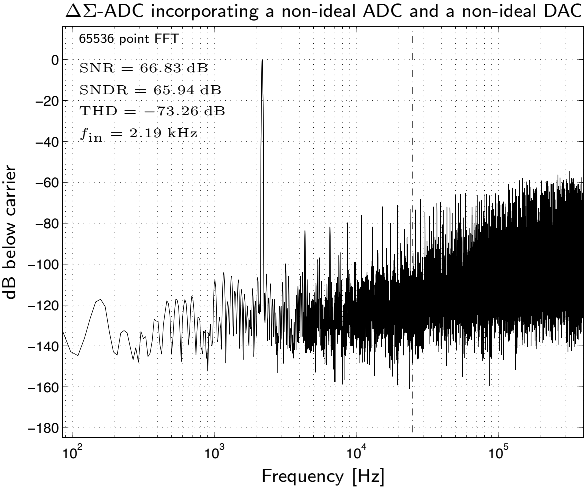


Figure 5.28: Power spectral density of a $\Delta\Sigma$-ADC with a non-ideal ADC and a non-ideal DAC with $\mathrm{Res}_{\mathrm{eff}} = 10.67$ bits.

We have provided a "proof of concept" for multi-bit $\Delta\Sigma$-ADCs incorporating digitally-corrected low-precision converters. With a digitally-corrected low-precision ADC and a corrected DAC consisting of minimal-sized current sources, the precise (and "expensive") elements of a multi-bit $\Delta\Sigma$-ADC can be circumvented. Some simulation results for $\Delta\Sigma$-ADCs with ideal and non-ideal multi-bit converters were presented.

5.6 Conclusions, Outlook

A simple technique to achieve a high effective resolution for a digitally-corrected DAC with small current sources was presented. The best performance is achieved if the current sources show some mismatch, or the current sources do not exhibit nominally identical output currents. All the outputs of the DAC need to be measured first and the digital encoder allocates these possible outputs to the digital input values with a look-up table. Not all possible outputs should be considered; in order to achieve best performance it is optimal to neglect the marginal outputs and rescale the other outputs. The achieved resolution of a DAC with n (nearly) minimal-sized current sources amounts to more than $n-2$ bits.

Although we have not yet demonstrated a fully functional self-calibrating DAC based on minimal-sized current sources, the ideas presented in this chapter point out a possibility of how to break out of the trade-off triangle, “speed, accuracy, power-consumption.” As the linearity is guaranteed due to the DAC’s calibration, such DAC’s may also be used in multi-bit sigma-delta ADCs or other types of ADCs containing DACs, as was demonstrated with various simulation results.

Chapter 6

Conclusion and Outlook

6.1 Conclusions

This thesis contains extensive measurements of analog decoders that are mainly based on the circuits introduced in [41]. Error rate curves for various decoders under different operating conditions were presented and discussed.

First, error rate curves for discrete Hamming decoders built out of two generations of softgates were discussed, and the mismatch on softgate level was characterized. Although mismatch was observed on gate-level, the discrete implementations proved to be very robust. It could further be shown that a redundant representation of a decoder (built out of softgates with reset-switches) can improve its error-correcting capability.

Furthermore, two analog decoder networks were presented: one for decoding an (8,4,4) extended Hamming code, the second one for decoding a (16,5,8) Reed Muller code. Especially the second implementation might be of interest: it contained a novel serial-to-parallel input interface and was an implementation of a non-standard iterative algorithm. Substantial measurements prove both decoders' full functionality under various operating conditions. The decoders are well-suited for low-voltage operation: both circuits operate in sub-1 V environments without a loss in performance. Although the transistors used in both networks were fairly large (compared with digital, minimal-sized transistors), the transistor mismatch could be measured and it diminished the BER performance of the decoders.

Our latest design, an analog coprocessor chip, was introduced. No reliable measurements could so far be presented.

In a second part, the insight that the effective resolution of a measured and digitally corrected (flash) ADC is basically independent of mismatch was confirmed by high-level simulations and with static measurements of a flash ADC chip. For a post corrected flash ADC with 256 low-precision comparators, an effective resolution of close to 7 bit was achieved.

High-level simulations of digitally-corrected DACs with imprecise current sources revealed that a high effective resolution is feasible, despite—or even thanks to—current source mismatch. With a proper calibration (based on measurements), a high-resolution DAC can be constructed from minimal-sized current sources. The advantages are obvious: the DAC demands minimal chip-area and the power-consumption (even for higher speeds) can be minimized. Instead of 2^n unit current sources (of a DAC with binary-scaled current-sources), only n (nearly) minimal-sized current sources are required. This approach was confirmed with simulations in time-domain and frequency-domain.

6.2 Outlook

We conclude this thesis with ideas for further research directions and describe open issues. These are opinions stemming from the present point of view and will probably change with new insights.

- **Decoder for large codes.** Up to now, only relatively small decoder networks have been successfully implemented. For a good BER performance, long codes are essential. We are confident that such a decoder is feasible, however, it can only be proven by a functional analog decoder chip for a large code.

- **High-speed decoder or equalizer.** In the last few years, mainly CMOS implementations of analog decoders were investigated. If operated in weak inversion, such decoders proved to be very economical concerning power consumption. For high-speed applications, where huge computational power and small latency is required, implementations with (hetero-)bipolar transistors are of interest. As a next step, the design of a channel equalizer or a simple analog decoder and the maximization of its speed could be considered.

- **Fully functional analog coprocessor with fast ADCs and DACs for a real-world application.** The analog coprocessor chip for decoding UMTS turbo codes has not yet been measured. However, if functional or not, it needs to be expanded: a fully functional mixed-signal turbo decoder requires an analog core and fast ADCs and DACs on the same chip, as well as an external digital processor.

- **Redundant representation of trellis codes.** The performance of the implemented sliding window trellis decoder will probably be limited by the transistor mismatch. Small transistors are required for high speeds, but small transistors show large mismatch. An interesting approach might be the use of a redundant trellis representation; again precision could be obtained by redundant instead of precise components.

- **Fully functional ADC and DAC with on-chip calibration.** Some research on digitally-corrected ADCs and DACs was presented. However, it is still unclear how to implement the digital calibration in practice. A next step could be to build a fully functional ADC and DAC with minimal-sized elements.

- **Fully functional oversampling ADC incorporating a multi-bit DAC.** The main difficulty of multi-bit oversampling ADCs is the linearity of the DAC in the feedback path. As digitally corrected DACs are (theoretically) perfectly linear, higher resolutions within the oversampling ADC and lower oversampling ratios are feasible.

Appendix A

Computing with Analog Transistors

The following collection of transistor circuits was derived during discussions on the analog coprocessor chip design with Daniel Furrer. These circuits are not proven to be useful, and we do not even claim their originality, but we think that they fit in nicely with the analog decoder circuits. We see most of the circuits as derivations of the translinear "Gilbert multiplier" [24]. A 2 × 2-matrix multiplier is presented in Figure A.1; both sets of transistors, M_1–M_4 and M_1, M_2, M_5, M_6, make up a complete "Gilbert cell."

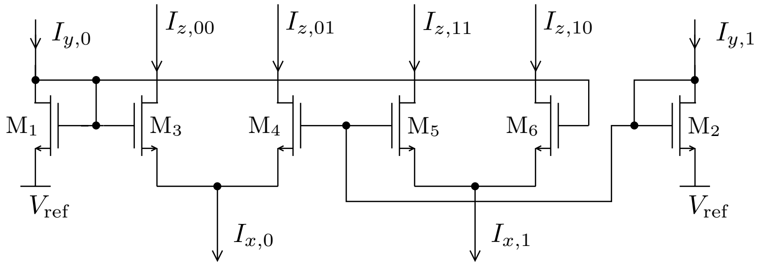


Figure A.1: The Gilbert 2 × 2-matrix multiplier.

Using the simplified transistor model for CMOS transistors in weak inversion given in (2.8), the following equalities hold:

$$\frac{I_{y,0}}{I_{y,0}+I_{y,1}} = \frac{I_0 e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{ref}}}{nU_{\mathrm{T}}}}}{I_0 e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{ref}}}{nU_{\mathrm{T}}}} + I_0 e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{ref}}}{nU_{\mathrm{T}}}}} \tag{A.1}$$

$$\frac{I_{y,0}}{I_y} = \frac{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}}, \quad I_y = I_{y,0} + I_{y,1} \tag{A.2}$$

$$\frac{I_{z,00}}{I_{z,00}+I_{z,01}} = \frac{I_0 e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{x},0}}{nU_{\mathrm{T}}}}}{I_0 e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{x},0}}{nU_{\mathrm{T}}}} + I_0 e^{\frac{V_{\mathrm{y},1}-V_{\mathrm{x},0}}{nU_{\mathrm{T}}}}} \tag{A.3}$$

$$\frac{I_{z,00}}{I_{x,0}} = \frac{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.4}$$

The right hand side of (A.2) and (A.4) is identical; we can therefore write:

$$\frac{I_{z,00}}{I_{x,0}} = \frac{I_{y,0}}{I_y} \tag{A.5}$$

The same procedure can be applied to attain the following equations for all output currents:

$$I_{z,00} = \frac{I_{y,0}I_{x,0}}{I_y} \qquad I_{z,01} = \frac{I_{y,0}I_{x,1}}{I_y} \tag{A.6}$$

$$I_{z,10} = \frac{I_{y,1}I_{x,0}}{I_y} \qquad I_{z,11} = \frac{I_{y,1}I_{x,1}}{I_y} \tag{A.7}$$

A.1 Multiplication and Division Circuits for Three Input Variables

The next circuit can be understood as an extension of the previously discussed Gilbert multiplier. We must, however, admit that the circuit presented in Figure A.2 is not as beautiful as the original Gilbert multiplier, and there are even some constraints on its functionality, i.e., $I_{w,0} - I_{y,0} \geq 0$ and $I_{w,1} - I_{y,1} \geq 0$ has to hold.

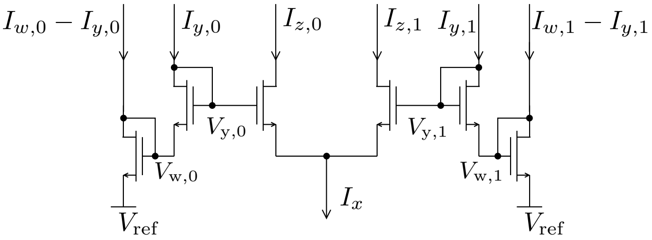


Figure A.2: Circuit performing a triple multiplication.

The following equations for the normalized output currents of the triple multiplication circuit hold:

$$\frac{I_{z,0}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.8}$$

$$\frac{I_{z,1}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.9}$$

$$V_{\mathrm{y},0} = nU_{\mathrm{T}} \cdot \log\left(\frac{I_{\mathrm{y},0} I_{\mathrm{w},0}}{I_0^2}\right) + V_{\mathrm{ref}} \tag{A.10}$$

$$V_{\mathrm{y},1} = nU_{\mathrm{T}} \cdot \log\left(\frac{I_{\mathrm{y},1} I_{\mathrm{w},1}}{I_0^2}\right) + V_{\mathrm{ref}} \tag{A.11}$$

By combining and reformulating these two sets of equations, an expression for the drain currents of the differential pair can be obtained:

$$I_{z,0} = \frac{I_x I_{\mathrm{y},0} I_{\mathrm{w},0}}{I_{\mathrm{y},0} I_{\mathrm{w},0} + I_{\mathrm{y},1} I_{\mathrm{w},1}} = \gamma I_x I_{\mathrm{y},0} I_{\mathrm{w},0} \tag{A.12}$$

$$I_{z,1} = \frac{I_x I_{\mathrm{y},1} I_{\mathrm{w},1}}{I_{\mathrm{y},0} I_{\mathrm{w},0} + I_{\mathrm{y},1} I_{\mathrm{w},1}} = \gamma I_x I_{\mathrm{y},1} I_{\mathrm{w},1} \tag{A.13}$$

We now have a circuit whose output currents are proportional to the product of three currents. The proportionality constant, however, is not constant for all inputs. This is of no concern, as the currents could easily be rescaled (and normalized) with a current-scaling circuit.

One application of such a triple-multiplication circuit could be a unidirectional softEQU gate for 4 variables (similar to the bidirectional softEQU gate for three variables described in Section 2.2).

The softEQU gate computes the outputs according to (A.14), where the scale factor γ is implicitly defined by the condition $p_{\mathrm{Z}}(0)+p_{\mathrm{Z}}(1)=1$.

$$\begin{bmatrix} p_{\mathrm{Z}}(0) \\ p_{\mathrm{Z}}(1) \end{bmatrix} = \gamma \begin{bmatrix} p_{\mathrm{W}}(0)\, p_{\mathrm{X}}(0)\, p_{\mathrm{Y}}(0) \\ p_{\mathrm{W}}(1)\, p_{\mathrm{X}}(1)\, p_{\mathrm{Y}}(1) \end{bmatrix} \tag{A.14}$$

The schematic of a unidirectional softEQU gate is given in Figure A.3, the output currents are as follows:

$$I_{z,0} = \frac{I_{x,0} I_{\mathrm{y},0} I_{\mathrm{w},0}}{I_{\mathrm{y},0} I_{\mathrm{w},0} + I_{\mathrm{y},1} I_{\mathrm{w},1}} = \gamma I_{x,0} I_{\mathrm{y},0} I_{\mathrm{w},0} \tag{A.15}$$

$$I_{z,1} = \frac{I_{x,1} I_{\mathrm{y},1} I_{\mathrm{w},1}}{I_{\mathrm{y},0} I_{\mathrm{w},0} + I_{\mathrm{y},1} I_{\mathrm{w},1}} = \gamma I_{x,1} I_{\mathrm{y},1} I_{\mathrm{w},1} \tag{A.16}$$

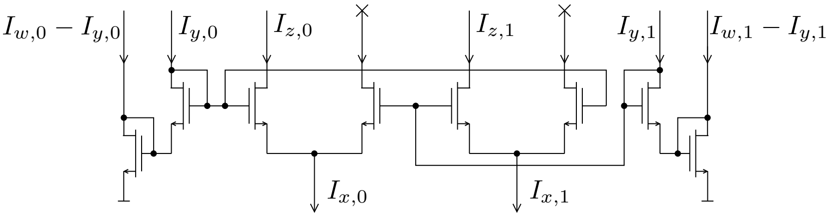


Figure A.3: Unidirectional softEQU gate circuit for 4 variables.

A similar circuit is proposed in Figure A.4, the derivation of the corresponding equations is the same as for the triple-multiplication circuit of Figure A.2.

Again, the following equations for the the circuit's normalized output currents hold:

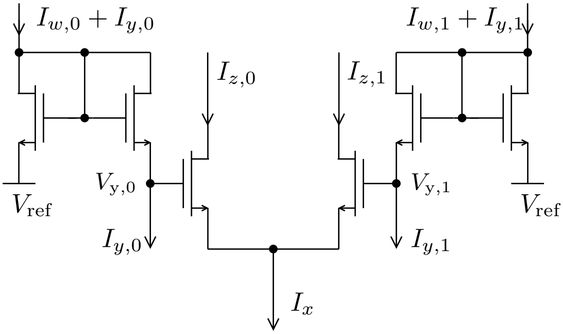


Figure A.4: Division and multiplication circuit for 3 variables.

$$\frac{I_{z,0}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.17}$$

$$\frac{I_{z,1}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.18}$$

$$V_{\mathrm{y},0} = V_{\mathrm{ref}} - nU_{\mathrm{T}} \cdot \log\left(\frac{I_{\mathrm{y},0}}{I_{\mathrm{x},0}}\right) \tag{A.19}$$

$$V_{\mathrm{y},1} = V_{\mathrm{ref}} - nU_{\mathrm{T}} \cdot \log\left(\frac{I_{\mathrm{y},1}}{I_{\mathrm{x},1}}\right) \tag{A.20}$$

Combining these two sets of equations leads to the following output currents:

$$I_{z,0} = \frac{I_x \frac{I_{\mathrm{w},0}}{I_{\mathrm{y},0}}}{\frac{I_{\mathrm{w},0}}{I_{\mathrm{y},0}} + \frac{I_{\mathrm{w},1}}{I_{\mathrm{y},1}}} = \gamma \frac{I_{\mathrm{w},0}}{I_{\mathrm{y},0}} I_x \tag{A.21}$$

$$I_{z,1} = \frac{I_x \frac{I_{\mathrm{w},1}}{I_{\mathrm{y},1}}}{\frac{I_{\mathrm{w},0}}{I_{\mathrm{y},0}} + \frac{I_{\mathrm{w},1}}{I_{\mathrm{y},1}}} = \gamma \frac{I_{\mathrm{w},1}}{I_{\mathrm{y},1}} I_x \tag{A.22}$$

We have now obtained a circuit that can "divide" and "multiply" currents.

A.2 Squaring and Square Root Extracting Circuit

The following set of circuits allows us to square and to extract square roots.

The output currents of the squaring circuit, as shown in Figure A.5, are as follows:

$$\frac{I_{z,0}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.23}$$

$$\frac{I_{z,1}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}}{e^{\frac{V_{\mathrm{y},0}}{nU_{\mathrm{T}}}} + e^{\frac{V_{\mathrm{y},1}}{nU_{\mathrm{T}}}}} \tag{A.24}$$

$$V_{\mathrm{y},0} = nU_{\mathrm{T}} \cdot \log\left(\frac{I_{\mathrm{y},0}^2}{I_0^2}\right) + V_{\mathrm{ref}} \tag{A.25}$$

$$V_{\mathrm{y},1} = nU_{\mathrm{T}} \cdot \log\left(\frac{I_{\mathrm{y},1}^2}{I_0^2}\right) + V_{\mathrm{ref}} \tag{A.26}$$

Combining these two sets of equations leads to the following output currents:

$$I_{z,0} = \frac{I_x I_{\mathrm{y},0}^2}{I_{\mathrm{y},0}^2 + I_{\mathrm{y},1}^2} = \gamma I_{\mathrm{y},0}^2 \tag{A.27}$$

$$I_{z,1} = \frac{I_x I_{\mathrm{y},1}^2}{I_{\mathrm{y},0}^2 + I_{\mathrm{y},1}^2} = \gamma I_{\mathrm{y},1}^2 \tag{A.28}$$

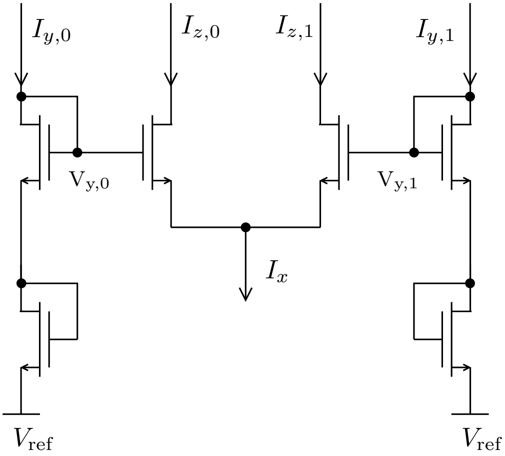


Figure A.5: Squaring circuit.

The output currents are now proportional to the squared input currents. It is clear, however, that powers of higher order can be achieved by stacking further diode-connected transistors underneath the input transistors.

The circuit presented in Figure A.6 is closely related to the squaring circuit.

This time, the diode-connected transistor is included in the differential pair. For the normalized output currents, the following equations hold:

$$\frac{I_{z,0}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{z},0}}{nU_\mathrm{T}}}}{e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{z},0}}{nU_\mathrm{T}}} + e^{\frac{V_{\mathrm{y},1}-V_{\mathrm{z},0}}{nU_\mathrm{T}}}} \tag{A.29}$$

$$\frac{I_{z,1}}{I_x} = \frac{e^{\frac{V_{\mathrm{y},1}-V_{\mathrm{z},1}}{nU_\mathrm{T}}}}{e^{\frac{V_{\mathrm{y},0}-V_{\mathrm{z},0}}{nU_\mathrm{T}}} + e^{\frac{V_{\mathrm{y},1}-V_{\mathrm{z},1}}{nU_\mathrm{T}}}} \tag{A.30}$$

$$V_{\mathrm{y},0} = nU_\mathrm{T} \cdot \log\left(\frac{I_{\mathrm{y},0}}{I_0}\right) + V_\mathrm{ref} \tag{A.31}$$

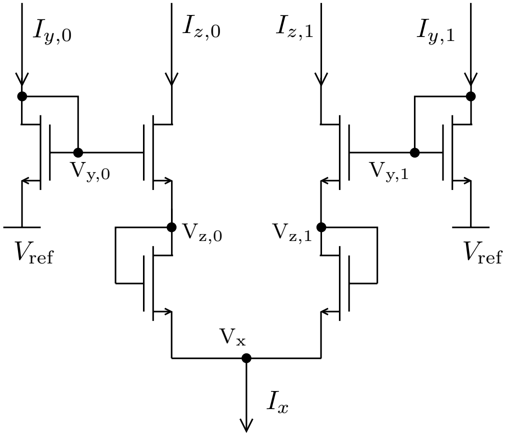


Figure A.6: Square root extracting circuit.

$$V_{z,0} = nU_T \cdot \log\left(\frac{I_{z,0}}{I_0}\right) + V_x \tag{A.32}$$

$$V_{y,1} = nU_T \cdot \log\left(\frac{I_{y,1}}{I_0}\right) + V_{ref} \tag{A.33}$$

$$V_{z,1} = nU_T \cdot \log\left(\frac{I_{z,1}}{I_0}\right) + V_x \tag{A.34}$$

We can now solve (A.29) and (A.30) for the output currents, $I_{z,0}$ and $I_{z,1}$:

$$I_{z,0} = \frac{\sqrt{I_{y,0}}\, I_x}{\sqrt{I_{y,0}} + \sqrt{I_{y,1}}} = \gamma\sqrt{I_{y,0}} \tag{A.35}$$

$$I_{z,0} = \frac{\sqrt{I_{y,1}}\, I_x}{\sqrt{I_{y,0}} + \sqrt{I_{y,1}}} = \gamma\sqrt{I_{y,1}} \tag{A.36}$$

The circuit of Figure A.6 has output currents that are proportional to the square-root of the input currents.

Appendix B

Limits on the Effective Resolution of Flash ADCs

A digitally corrected flash ADC with imprecise elements (as described in Chapter 4) can be modeled as having 2^n comparators with threshold voltages generated randomly by adding a zero-mean Gaussian random variable with variance σ_ε^2 to the ideal thresholds. The comparators are then rearranged to produce a monotonous digital output.

For large mismatch, i.e., for $\sigma_\varepsilon \gg 0$, the rearranged thresholds can be modeled as having been created by a Poisson process. λ is then the average number of thresholds on a certain interval $V_{\text{input-range}} = V_{\text{i-r}}$. The voltage difference between two thresholds, $V_{\Delta i} = V_{\text{th},i+1} - V_{\text{th},i}$, is then exponentially distributed with the probability density function given in (B.1).

$$f(V_{\Delta i};\lambda) = \begin{cases} \lambda e^{-\lambda V_{\Delta i}} & , \ V_{\Delta i} \geq 0 \\ 0 & , \ V_{\Delta i} < 0 \end{cases} \tag{B.1}$$

The rms quantization error of an ADC with 2^n imprecise comparators dependent on $V_{\Delta i}$ (the distance between two consecutive thresholds) and with $V_{\text{i-r}} = \sum_i V_{\Delta i}$, is derived in the following equations.

$$Q^2 = \mathrm{E}\left[\left(A_{\text{in}} - D_{\text{out}}(A_{\text{in}})\right)^2 \right] \tag{B.2}$$

$$= \frac{1}{V_{\text{i-r}}} \int_0^{V_{\text{i-r}}} \left(A_{\text{in}} - D_{\text{out}}(A_{\text{in}})\right)^2 \, \mathrm{d}A_{\text{in}} \tag{B.3}$$

$$= \frac{1}{V_{\text{i-r}}} \sum_i \int_0^{V_{\Delta i}} \left(A_{\text{in}} - \frac{V_{\Delta i}}{2} \right)^2 \, \mathrm{d}A_{\text{in}} \tag{B.4}$$

$$= \frac{1}{V_{\text{i-r}}} \sum_{i=0}^{2^n-1} \frac{V_{\Delta i}^3}{12} \tag{B.5}$$

$$Q^2 = \frac{1}{12} \frac{\sum_{i=0}^{2^n-1} V_{\Delta i}^3}{\sum_{i=0}^{2^n-1} V_{\Delta i}} \tag{B.6}$$

Now that the probability density function of $V_{\Delta i}$ is given (see (B.1)) and an expression for the expectation of the quantization error dependent on the random variable $V_{\Delta i}$ exists (see (B.6)), the expectation of the rms quantization error (for an ADC with converters strongly afflicted by mismatch) can be computed as follows:

$$\mathrm{E}\left[Q^2\right] = \mathrm{E}\left[\frac{\sum_i V_{\Delta i}^3}{12 \sum_i V_{\Delta i}} \right] \tag{B.7}$$

$$= \mathrm{E}\left[\frac{\sum_i V_{\Delta i}^3}{12 V_{\text{i-r}}} \right] \tag{B.8}$$

$$= \frac{1}{12 V_{\text{i-r}}} \mathrm{E}\left[\sum_{i=0}^{2^n-1} V_{\Delta i}^3 \right] \tag{B.9}$$

$$= \frac{1}{12 V_{\text{i-r}}} \sum_{i=0}^{2^n-1} \mathrm{E}\left[V_{\Delta i}^3\right] \tag{B.10}$$

$$= \frac{2^n}{12 V_{\text{i-r}}} \mathrm{E}\left[V_{\Delta i}^3\right] \tag{B.11}$$

$$\mathrm{E}\left[Q^2\right] \big|_{\sigma_\varepsilon \gg 0} = \frac{2^n}{12 V_{\text{i-r}}} \int_{-\infty}^{\infty} V_{\Delta i}^3 f(V_{\Delta i}; \lambda) \, \mathrm{d}V_{\Delta i} \tag{B.12}$$

$$= \frac{2^n}{12V_{\text{i-r}}} \int_0^\infty V_{\Delta i}^3 \lambda e^{-\lambda V_{\Delta i}} \, \mathrm{d}V_{\Delta i} \tag{B.13}$$

$$= \frac{2^n}{12V_{\text{i-r}}} 3! \lambda^{-3} \tag{B.14}$$

With $\lambda = 2^n$ and $V_{\text{i-r}} = 1$ the expectation of the mean-squared quantization error is

$$\mathrm{E}\left[Q^2\right]\big|_{\sigma_\varepsilon \gg 0} = \frac{2^{-2n}}{2}. \tag{B.15}$$

For comparison, the rms quantization error of an ideal ADC amounts to

$$Q_{\text{ideal}} = \frac{2^{-n}}{\sqrt{12}}. \tag{B.16}$$

The expectation of the effective resolution of a non-ideal and corrected ADC can now be computed by applying Definition 4.5:

$$\text{Res}_{\text{eff}} = \frac{-\log\left(\frac{\sqrt{12}Q}{\text{“input range”}}\right)}{\log(2)} \;\; \text{[bits]} \tag{B.17}$$

$$\text{Res}_{\text{eff}}\Big|_{\sigma_\varepsilon \gg 0} = n - \frac{\log(\sqrt{6})}{\log(2)} \;\; \text{[bits]} \tag{B.18}$$

$$\approx n - 1.3 \;\; \text{[bits]} \tag{B.19}$$

The effective resolution of a corrected ADC with 2^n comparators with large mismatch lies in average only 1.3 bits below the maximum achievable effective resolution.

Bibliography

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[2] E. Balestrieri, P. Daponte, and S. Rapuano, “A state of the art on ADC error compensation methods,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 4, pp. 1388–1394, August 2005.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes (1),” in *Proceedings of the IEEE International Conference on Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[4] A. Blanksby and C. Howland, “A 690-mW 1-Gbps 1024-b, rate-1/2 low-density parity-check code decoder,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, March 2002.

[5] J. W. Bruce, “Nyquist-rate digital to analog converter architectures,” *IEEE Potentials*, vol. 20, no. 3, pp. 24–28, August 2001.

[6] K. Bult and A. Buchwald, “An embedded 240-mW 10-b 50-MS/s CMOS ADC in 1-mm^2,” *IEEE Journal of Solid-State Circuits*, vol. 32, no. 12, pp. 1887–1895, December 1997.

[7] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Transactions on Communications*, vol. 44, pp. 927–946, May 1998.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley Series in Telecommunications, 1991.

[9] I. Daubechies, R. DeVore, C. S. Güntürk, and V. A. Vaishampayan, "Beta expansions: A new approach to digitally corrected A/D conversion," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. II, Phoenix, Arizona, May 2002, pp. 784–787.

[10] ——, "A/D conversion with imperfect quantizers," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 874–885, March 2006.

[11] A. C. Dent and C. F. N. Cowan, "Linearization of analog-to-digital converters," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 6, pp. 729–737, June 1990.

[12] P. G. Drennan and C. C. McAndrew, "Understanding MOSFET mismatch for analog design," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 450–456, March 2003.

[13] C. C. Enz, F. Krummenacher, and E. A. Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," *Analog Integrated Circuits and Signal Processing Journal*, vol. 8, pp. 83–114, July 1995.

[14] *Univeral Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD) (3GPP TS 25.212 version 3.4.0 Release 1999)*, ETSI TS 125.212 v3.4.0 (2000-09) ed., ETSI, 1999.

[15] M. P. Flynn and I. Bogue, "Using redundancy to break the link between accuracy and speed in an ADC," in *Instrumentation and Measurement Technology Conference*, vol. I, Vail, Colorado, May 2003, pp. 850–853.

[16] M. P. Flynn, C. Donovan, and L. Sattler, "Digital calibration incorporating redundancy of flash ADCs," *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 50, no. 5, pp. 205–213, May 2003.

[17] G. D. Forney Jr., "Codes on graphs: Normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, February 2001.

[18] M. Frey, H.-A. Loeliger, and P. Merkli, "Multi-level soft detector/quantizer," U.S. Patent Application 11/081,182, patent pending, March 2005.

[19] ——, “Multi-level soft detector/quantizer,” International Patent Application PCT/CH2006/000133, patent pending, March 2005.

[20] G. Fromherz and E. Schinca, “Convergence of sum product algorithm in CMOS technology,” Diploma Thesis, Swiss Federal Institute of Technology, Signal- and Information Processing Laboratory, Zurich, Switzerland, March 1999.

[21] D. Furrer, “An analog coprocessor for decoding a turbo code,” Masters’s Thesis, Swiss Federal Institute of Technology, Signal- and Information Processing Laboratory, Zurich, Switzerland, March 2005.

[22] V. Gaudet, “Architecture and implementation of analog iterative decoders,” PhD Thesis, University of Toronto, Edward S. Rogers Sr. Graduate Department of Electrical and Computer Engineering, Toronto, Ontario, November 2003.

[23] V. Gaudet and G. Gulak, “A 13.3-Mb/s 0.35-μm CMOS analog turbo decoder IC with a configurable interleaver,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 11, pp. 2010–2015, November 2003.

[24] B. Gilbert, “A precise four-quadrant multiplier with subnanosecond response,” *IEEE Journal of Solid-State Circuits*, vol. SC-3, no. 4, pp. 365–373, December 1968.

[25] ——, “A monolithic 16-channel analog array normalizer,” *IEEE Journal of Solid-State Circuits*, vol. SC-19, no. 6, pp. 956–963, December 1984.

[26] P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd ed. John Wiley and Sons, Inc., 1993.

[27] J. Hagenauer, “Decoding of binary codes with analog networks,” in *Proceeding of the IEEE Information Theory Workshop*, San Diego, California, February 1998, pp. 13–14.

[28] J. Hagenauer and M. Winklhofer, “The analog decoder,” in *Proceedings of the IEEE International Symposium on Information Theory*, Cambridge, MA, August 1998, p. 145.

[29] J. Hagenauer, E. Offer, C. Méasson, and M. Mörz, “Decoding and equalization with analog non-linear networks,” in *European Transaction on Telecommunications (ETT)*, October 1999, pp. 107–128.

[30] P. Händel, M. Skoglund, and M. Pettersson, “A calibration scheme for imperfect quantizers,” *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 5, pp. 1063–1068, October 2000.

[31] H. Inose and Y. Yasuda, “A unity bit coding method by negative feedback,” in *Oversampling Delta-Sigma Data Converters. Theory, Design and Simulation*, J. C. Candy and G. C. Temes, Eds. IEEE Press, 1992.

[32] D. A. Johns and K. Martin, *Analog Integrated Circuit Design.* John Wiley and Sons, 1997.

[33] R. Kanan, F. Kaess, and M. Declercq, “A 640-mW high accuracy 8-bit 1 GHz flash ADC encoder,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. II, Orlando, Florida, May 1999, pp. 420–423.

[34] P. R. Kinget, “Device mismatch and tradeoffs in the design of analog circuits,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1212–1224, June 2005.

[35] F. A. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, February 2001.

[36] X. Li and J. A. Ritcey, “Bit-interleaved coded modulation with iterative decoding using soft feedback,” *IEE Electronic Letters*, vol. 34, no. 10, pp. 942–943, May 1998.

[37] S. P. Lloyd, “Least squares optimization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, March 1982.

[38] H.-A. Loeliger, “Decoding and equalization: iterative algorithms and analog decoding,” presented at 1999 IMA Workshop on ”Codes, Systems, and Graphical Models”, Minneapolis, August 1999.

[39] ——, “Some remarks on factor graphs,” in *Proceedings of the International Symposium on Turbo Codes and Related Topics*, Brest, France, September 2003.

[40] ——, “An introduction to factor graphs,” *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, January 2004.

[41] H.-A. Loeliger, F. Lustenberger, M. Helfenstein, and F. Tarköy, "Probability propagation and decoding in analog VLSI," in *Proceedings of the IEEE International Symposium on Information Theory*, Cambridge, Massachusetts, August 1998, p. 146.

[42] ——, "Decoding in analog VLSI," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 99–101, April 1999.

[43] ——, "Probability propagation and decoding in analog VLSI," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 837–843, February 2001.

[44] H. Lundin, M. Skoglund, and P. Händel, "Optimal index-bit allocation for dynamic post-correction of analog-to-digital converters," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 660–671, February 2005.

[45] F. Lustenberger, "On the design of analog iterative VLSI decoders," PhD Thesis no. 13879, Swiss Federal Institute of Technology, Signal and Information Processing Laboratory, Zurich, Switzerland, November 2000.

[46] F. Lustenberger, M. Helfenstein, H.-A. Loeliger, F. Tarköy, and G. S. Moschytz, "All analog decoder for a binary (18,9,5) tail-biting trellis code," in *Proceedings of the European Solid-State Circuits Conference*, Duisburg, Germany, September 1999, pp. 362–365.

[47] N. R. Malik, *Electronic Circuits: Analysis, Simulation, and Design.* Prentice Hall International, Inc., 1995.

[48] P. Merkli, "Message-passing algorithms and analog electronic circuits," PhD Thesis no. 15942, Swiss Federal Institute of Technology, Signal and Information Processing Laboratory, Zurich, Switzerland, April 2005.

[49] M. Moerz, "Analog sliding window decoder core for mixed signal turbo decoder," in *Proceedings of the 5th International ITG Conference on Source and Channel Coding (SCC)*, Erlangen, Germany, January 2004.

[50] S. M. Moser, "Investigation of algebraic codes of small block length using factor graphs," Masters's Thesis, Swiss Federal Institute of Technology, Signal- and Information Processing Laboratory, Zurich, Switzerland, March 1999.

[51] S. A. S. Onge, D. L. Harame, J. S. Dunn, S. Subbanna, D. C. Ahlgren, G. Freeman, B. Jagannathan, J. Jeng, K. Schonenberg, K. Stein, R. Groves, D. Coolbaugh, N. Feilchenfeld, P. Geiss, M. Gordon, P. Gray, D. Hershberger, S. Kilpatrick, R. Johnson, A. Joseph, L. Lanzerotti, J. Malinkowski, B. Orner, and M. Zierak, "A 0.24 μm SiGe BiCMOS mixed-signal RF production technology featuring a 47 GHz f_t HBT and 0.18 mm l_{eff} CMOS," in *Bipolar/BiCMOS Circuits and Technology Meeting Proceedings*, September 1999, pp. 117–120.

[52] C. Paulus, H.-M. Blüthgen, M. Löw, E. Sicheneder, N. Brüls, A. Courtois, M. Tiebout, and R. Thewes, "A 4 GS/s 6b flash ADC in 0.13 μm CMOS," in *Symposium on VLSI Circuits*, Honolulu, Hawaii, June 2004, pp. 420–423.

[53] J. G. Proakis, *Digital Communications.* Mc. Graw Hill International Edition, 2001.

[54] B. Razavi, *Principles of Data Conversion System Design.* IEEE Press, 1995.

[55] F. Seguin, C. Lahuec, J. Lebert, M. Arzel, and M. Jezequel, "Analogue 16-QAM demodulator," *IEE Electronic Letters*, vol. 40, no. 18, pp. 1138–1139, September 2004.

[56] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets. Part I: Introduction," *IEEE Communications Magazine*, vol. 25, no. 2, pp. 5–11, February 1987.

[57] K. Uyttenhove and M. S. J. Steyaert, "Speed-power-accuracy tradeoff in high-speed CMOS ADCs," *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 49, no. 4, pp. 280–287, April 2002.

[58] P. van Beek, K. Doris, J. Hegt, and A. van Roermund, "Optimum segmentation for high speed current steering digital-to-analog converters," in *Proceedings of the ProRISC Workshop 2004*, Eindhoven, Netherlands, November 2004, pp. 465–473.

[59] R. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters.* Kluwer Academic Publishers, 2003.

[60] D. Vogrig, A. Gerosa, A. Neviani, A. G. i Amat, G. Montorsi, and S. Benedetto, “A 0.35-μm CMOS analog turbo decoder for the 40-bit, rate 1/3, UMTS channel code,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 3, pp. 753–762, March 2005.

[61] J. Welz and I. Galton, “Necessary and sufficient conditions for mismatch shaping in a general class of multibit DACs,” *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 49, no. 12, pp. 748–759, December 2002.

[62] N. Wiberg, H.-A. Loeliger, and R. Kötter, “Codes and iterative decoding on general graphs,” *European Transaction on Telecommunications (ETT)*, vol. 6, pp. 513–525, September, October 1995.

[63] J. J. Wikner and N. Tan, “Modeling of CMOS digital-to-analog converters for telecommunication,” *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 46, no. 5, pp. 489–499, May 1999.

[64] C. Winstead, J. Dai, S. Yu, C. Myers, R. R. Harrison, and C. Schlegel, “CMOS analog MAP decoder for (8,4) Hamming code,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 122–131, January 2004.

[65] C. Winstead, J. Die, S. Yu, R. Harrison, C. J. Myers, and C. Schlegel, “Analog MAP decoder for (8,4) Hamming code in subthreshold CMOS,” in *Proceedings of the IEEE International Symposium on Information Theory*, Washington DC, USA, June 2001, p. 330.

[66] C. Winstead, “Analog iterative error control decoders,” PhD Thesis, University of Alberta, Department of Electrical and Computer Engineering, Edmonton, Alberta, 2005.

[67] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Generalized belief propagation,” *Advances in Neural Information Processing Systems*, vol. 13, pp. 689–695, December 2000.

About the Author

Matthias U. Frey was born in Lucerne, Switzerland on December 31, 1975. He enjoyed his childhood in Hong Kong and in Rothenburg, Switzerland, where he attended primary school for three years each. After having spent a year at the high school "Kantonsschule Reussbühl", he attended Brookfield High School in Ottawa, Canada. Back in Switzerland, he obtained the "Matura Typus C" in 1995.

In the same year, he joined the Swiss Federal Institute of Technology (ETH Zurich) to study electrical engineering. After three years of studying he took one year off for internships in Switzerland and Thailand, and for a research project on linear block codes with Prof. R. E. Blahut at the University of Illinois in Urbana-Champaign. In 2001, after having finished his diploma thesis on analog chip design at the Tokyo Institute of Technology, he graduated from ETH Zurich with a diploma degree in electrical engineering.

In fall 2001 he joined the Signal and Information Processing Laboratory (ISI) at ETH Zurich as a research and teaching assistant. Besides his work on his Ph.D., he organized an international workshop on analog decoding and was responsible for the proceedings of two international conferences in Switzerland. His main research interests are analog decoders, as well as analog-to-digital and digital-to-analog converters. In his spare time, he taught judo at the academic sports club of Zurich and for the central Swiss regional team. He also enjoyed taking achromatic analog photographic pictures.

Series in Signal and Information Processing

edited by Hans-Andrea Loeliger

Vol. 1: Hanspeter Schmid, **Single-Amplifier Biquadratic MOSFET-C Filters.** ISBN 3-89649-616-6

Vol. 2: Felix Lustenberger, **On the Design of Analog VLSI Iterative Decoders.** ISBN 3-89649-622-0

Vol. 3: Peter Theodor Wellig, **Zerlegung von Langzeit-Elektromyogrammen zur Prävention von arbeitsbedingten Muskelschäden.** ISBN 3-89649-623-9

Vol. 4: Thomas P. von Hoff, **On the Convergence of Blind Source Separation and Deconvolution.** ISBN 3-89649-624-7

Vol. 5: Markus Erne, **Signal Adaptive Audio Coding using Wavelets and Rate Optimization.** ISBN 3-89649-625-5

Vol. 6: Marcel Joho, **A Systematic Approach to Adaptive Algorithms for Multichannel System Identification, Inverse Modeling, and Blind Identification.** ISBN 3-89649-632-8

Vol. 7: Heinz Mathis, **Nonlinear Functions for Blind Separation and Equalization.** ISBN 3-89649-728-6

Vol. 8: Daniel Lippuner, **Model-Based Step-Size Control for Adaptive Filters.** ISBN 3-89649-755-3

Vol. 9: Ralf Kretzschmar, **A Survey of Neural Network Classifiers for Local Wind Prediction.** ISBN 3-89649-798-7

Vol. 10: Dieter M. Arnold, **Computing Information Rates of Finite State Models with Application to Magnetic Recording.** ISBN 3-89649-852-5

Vol. 11: Pascal O. Vontobel, **Algebraic Coding for Iterative Decoding.** ISBN 3-89649-865-7

Vol. 12: Qun Gao, **Fingerprint Verification using Cellular Neural Networks.** ISBN 3-89649-894-0

Vol. 13: Patrick P. Merkli, **Message-Passing Algorithms and Analog Electronic Circuits.** ISBN 3-89649-987-4

Vol. 14: Markus Hofbauer, **Optimal Linear Separation and Deconvolution of Acoustical Convolutive Mixtures.** ISBN 3-89649-996-3

Vol. 15: Sascha Korl, **A Factor Graph Approach to Signal Modelling, System Identification and Filtering.** ISBN 3-86628-032-7

Hartung-Gorre Verlag Konstanz – http://www.hartung-gorre.de