

Diss. ETH No. 17716

# Low-Power Circuit Architectures and Clocking Strategies for Digital Hearing Aids

A dissertation submitted to

ETH ZURICH

for the degree of  
Doctor of Sciences

presented by

FELIX BÜRGIN

Dipl. El.-Ing. ETH

born 24 October 1976

citizen of Liestal BL and Buus BL

accepted on the recommendation of

Prof. Dr. Wolfgang Fichtner, examiner

Dr. Paul Zbinden, co-examiner

2008



# Acknowledgements

First of all, I want to express my gratitude to Prof. Dr. Wolfgang Fichtner for giving me the opportunity to pursue my Ph.D. at the Integrated Systems Laboratory (IIS) and for providing the excellent research infrastructure. I also want to thank Dr. Paul Zbinden for having read and co-examined this thesis.

Furthermore, I would like to thank Dr. Norbert Felber, head of the “IC and System Design and Test” research group at IIS, and Dr. Hubert Kaeslin, head of the “Microelectronics Design Center” (DZ) at the Department of Information Technology and Electrical Engineering, for motivating me to pursue the Ph.D. thesis at IIS and for their ongoing support during my period as a Ph.D. student. Norbert, thank you for proofreading my thesis and providing many valuable suggestions to improve the quality of this work! Many thanks belong to Matthias Brändli and Christoph Balmer of the DZ for establishing and maintaining the outstanding EDA software infrastructure. Matthias, I will never forget your willingness to help and patience in answering my numerous technical questions on chip design. Thank you! And Matthias, I really appreciated the many fruitful and ongoing discussions and comments about worm-eaten apples! I will miss them!

I am very grateful to my friends from IIS. Especially, I would like to thank Dr. Flavio Carbognani. Flavio and I successfully worked on the research project about low-power chip design. My special thank also belongs to Dr. Marc Wegmüller for his helpfulness and friendship. We three guys from ETZ J89 had a really funny, friendly, and productive research atmosphere during our Ph.D. period. Many thanks also belong to our former office mates, Dr. Chiara Martelli and Robert Reutemann. Jürg Treichler, thank you for answering my many small

questions during the last five years and bearing company during the last few months of my time here at IIS. I also would like to thank Craig Keogh for helping me improve my English in this thesis.

Many thanks belong to the former and current members of the digital research group, namely Dr. Andy Burg, Dr. Frank K. Gürkaynak, Dr. David Perels, Dr. Simon Haene, Peter Lüthi, Stephan Eberli, Christoph Studer, Markus Wenk, and Luca Henzen. Working with you at IIS was a great pleasure. Thank you for the friendship!

Among the students I supervised during their semester projects, I would like to mention Martin Hediger, Hektor Meier, Robert Meyer-Piening, and Rafael Santschi. Their work is the basis of one chapter in this thesis.

I would like to express my gratefulness to Bruno Fischer and Verena Roffler from the secretariat team. My special thanks belong to Christine Haller for her many encouraging words. Many thanks also go to our institute technicians Hansjörg Gisler and Hanspeter Mathys. I am also thankful to the computer administration group of IIS, namely Christoph Wicki, Christoph Richardet, Adam Feigin, and Anja Böhm, for maintaining the outstanding IT infrastructure at our laboratory. Their immediate and unbureaucratic help in computer problems is superb.

Zu guter Letzt möchte ich mich hiermit ganz herzlich bei meinen Eltern, Karl und Therese Bürgin-Singeisen, bedanken; sie haben es mir ermöglicht und mich dazu motiviert, Elektrotechnik an der ETH zu studieren. Weiter gebührt ihnen herzlichen Dank für die wertvolle Unterstützung während den vergangenen Jahren meiner Dissertation.

Zürich, Juli 2008

Felix Bürgin

# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>xi</b>
<b>Zusammenfassung</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Outline of this Thesis . . . . .	2
1.3 Contributions . . . . .	2
<b>2 Overview</b>	<b>5</b>
2.1 Hearing Aids . . . . .	5
2.1.1 Digital Hearing Aids . . . . .	5
2.1.2 System Overview . . . . .	6
2.1.3 Power Supply . . . . .	7
2.1.4 Signal Processing Algorithms in Digital Hearing Aid Applications . . . . .	10
2.2 Power Consumption in Digital CMOS Circuits . . . . .	21
2.2.1 History of CMOS Circuits . . . . .	21
2.2.2 Sources of Power Consumption . . . . .	23
2.2.3 Countermeasures . . . . .	28
<b>3 Architectures for Low-Power Circuits</b>	<b>31</b>
3.1 Architectural Trade-Offs . . . . .	31
3.1.1 The Isomorphic Architecture . . . . .	31
3.1.2 Iterative Decomposition . . . . .	33

3.2	Spectral Sharpening Chip Design . . . . .	34
3.2.1	Implemented Architectures . . . . .	34
3.2.2	Synthesis and Power-Estimation Flow . . . . .	37
3.3	Results . . . . .	38
3.3.1	Synthesis and Simulation Results . . . . .	38
3.3.2	Post-Layout Power Simulations and Power Measurements on Silicon . . . . .	45
3.4	Discussion of Low-Power Architectures . . . . .	45
<b>4</b>	<b>Impact of Number Format</b>	<b>49</b>
4.1	Introduction to Number Formats . . . . .	49
4.1.1	Signed Number Representations . . . . .	49
4.1.2	Low-Power Arithmetic . . . . .	59
4.2	Impact of Number Format in FIR Filters . . . . .	60
4.2.1	MAC Unit . . . . .	60
4.2.2	Simulation Results . . . . .	61
4.2.3	Discussion . . . . .	62
4.3	Impact of Number Format in Lattice Filters . . . . .	64
4.3.1	Analysis Stage . . . . .	64
4.3.2	Simulation Results . . . . .	65
4.3.3	Discussion . . . . .	67
4.4	Final Remarks to Number Formats . . . . .	69
<b>5</b>	<b>Clocking Strategies for Low-Power</b>	<b>71</b>
5.1	Synchronous Clocking Disciplines . . . . .	71
5.1.1	Single-Edge-Triggered One-Phase Clocking . . . . .	72
5.1.2	Dual-Edge-Triggered One-Phase Clocking . . . . .	75
5.1.3	Unsymmetric Level-Sensitive Two-Phase Clocking . . . . .	79
5.1.4	Symmetric Level-Sensitive Two-Phase Clocking . . . . .	83
5.1.5	Mixed Unsymmetric-Symmetric Level-Sensitive Two-Phase Clocking . . . . .	87
5.2	Fabricated Designs . . . . .	89
5.2.1	FIR Filter I . . . . .	89
5.2.2	FIR Filter II . . . . .	106
5.2.3	Model Hearing Aid System . . . . .	117
<b>6</b>	<b>Summary and Conclusions</b>	<b>133</b>

<b>A Efficiency Comparison between Time- and Frequency-Domain Filtering</b>	<b>137</b>
<b>B Energy Effective Capacitance Calculation</b>	<b>141</b>
<b>List of Abbreviations</b>	<b>143</b>
<b>Bibliography</b>	<b>147</b>
<b>Curriculum Vitae</b>	<b>155</b>



# Abstract

Battery-powered portable applications demand for extremely low power consumption in order to ensure reasonable battery life and autonomy. Moreover, they ask for small silicon area because of cost and often also space limitations. Hearing aids, in particular, require extremely tight constraints on power dissipation *and* chip area to render these instruments as small as possible.

Driven by Moore's Law, transistor dimensions in modern semiconductors are progressively shrinking. New VLSI technologies decrease the dynamic power consumption of digital integrated circuits mainly thanks to the reduced supply voltage. On the other hand, smaller device sizes allow the integration of more functionality on the same silicon area, hence again increasing the total power consumption.

This thesis aims at evaluating different low-power VLSI circuit design techniques that are especially suited for low-frequency audio applications, such as hearing aids. Thereby, several circuit architectures are compared for different levels of resource sharing, number formats, and clocking strategies in established VLSI technologies (minimum transistor lengths of  $0.25\ \mu\text{m}$  and  $0.18\ \mu\text{m}$ ). Additionally, cell redesign for reduced gate capacitance and hence lower dynamic power consumption is investigated. All these techniques have been verified by simulations and measurements of several state-of-the-art hearing aid signal processing algorithms that have been implemented and integrated on silicon.



# Zusammenfassung

Batteriebetriebene portable Anwendungen verlangen einen extrem tiefen Energieverbrauch, um eine vernünftige Batterielebensdauer und Autonomie zu gewährleisten. Zudem erfordern sie eine kleine Siliziumfläche aufgrund von Kosten- und oft auch Grössenbeschränkungen. Hörgeräte im speziellen sind eine Anwendung, welche strenge Anforderungen an den Energieverbrauch *sowie* an die Siliziumfläche stellt, um diese Instrumente so klein als möglich zu halten.

Getrieben durch Moores Gesetz werden Transistordimensionen in modernen Halbleitern schrittweise kleiner. Neue VLSI-Technologien reduzieren den dynamische Energieverbrauch von digitalen integrierten Schaltkreisen hauptsächlich dank der herabgesetzten Versorgungsspannung. Andererseits erlauben kleinere Bauelemente die Integration von mehr Funktionalität auf der selben Chipfläche, was den totalen Energieverbrauch folglich wiederum erhöht.

Diese Arbeit beabsichtigt die Beurteilung verschiedener Entwurfstechniken für hochintegrierte Schaltkreise betreffend tiefem Energieverbrauch. Diese Techniken sind speziell für Audio-Anwendungen mit tiefer Taktrate wie Hörgeräte geeignet. Mehrere Schaltungsarchitekturen für verschiedene Grade von Ressourcenteilung, Zahlenformate und Taktgebungsverfahren werden in etablierten Halbleitertechnologien (minimale Transistorlänge von  $0.25\ \mu\text{m}$  und  $0.18\ \mu\text{m}$ ) miteinander verglichen. Zudem wird das Redesign von Standardzellen, um Transistor-Gatekapazitäten und folglich den dynamischen Energieverbrauch zu reduzieren, untersucht. All diese Techniken sind mithilfe von Simulationen und Messungen an mehreren implementierten und in Silizium integrierten aktuellen Hörgeräatalgorithmen verifiziert worden.



# Chapter 1

## Introduction

### 1.1 Motivation

Today's digital hearing aids are portable devices that impose extraordinarily tight constraints on chip area and power consumption. Their signal processing power is remarkable when considering the small source of energy that supplies both the actual audio processing and the power amplifier for the audio signal. Some advanced hearing aid models completely fit into the auditory canal of the hearing impaired patient, making them completely invisible.

Although power consumption has increased with the advent of digital signal processing into modern hearing aids, they offer many advantages over their analog predecessors [1]. Nowadays, these devices contain many different functions to increase the intelligibility of speech in different situations (noise, "cocktail party", auditorium etc.). Most of these functions are implemented by some kind of adaptive filtering. Back in the analog era, such elaborate speech processing was hardly possible, if at all.

The development goes on, using even more energy in the future: A wireless one-way communication between a digital or analog audio source such as a mobile phone, cinema sound equipment, a teacher's microphone and a hearing aid can be realized. Thereby, the sound signal is transmitted by a wireless link from the source to the hearing

aid. A (duplex) communication between a pair of hearing aids has recently been presented in order to enable the exchange of parameters, such as synchronization of volume between the left and right device. Eventually, interchange of audio data between the two devices to fully explore the potential of binaural hearing will be the last step of this development [2].

The digital part of a modern hearing aid typically eats up around 50% of the total power budget<sup>1</sup>. Given the assumption that the consumption of the analog part is fixed by physical constraints, namely the power amplification, and the amount of functionality (digital signal processing) in the hearing instrument is, as stated above, continually increasing, low-power digital circuit techniques are mandatory to maintain an acceptable battery lifetime of around one week.

## 1.2 Outline of this Thesis

Chapter 2 provides background information about hearing aids in general and presents examples of implemented signal processing algorithms used in modern devices. Additionally, the fundamentals on power consumption in digital VLSI ASICs are introduced. Chapter 3 investigates the search for an optimal hardware architecture of a hearing aid algorithm. In Chapter 4, the influence of number formats on power consumption is discussed. Chapter 5 proposes different clocking strategies aiming at reducing the power consumption of the clock distribution and sequential elements. Eventually, in Chapter 6, the results are summarized and conclusions are drawn.

## 1.3 Contributions

The goal of this work was to explore methodologies for the reduction of power consumption in low-power digital audio processing devices, such as hearing aids. The contributions of this thesis to low-power ASICs for hearing aid applications are presented in the main Chapters 3 to 5.

---

<sup>1</sup>These 50% do not include the power consumption of wireless interfaces. Their dissipation would mainly contribute to digital power consumption.

**Chapter 3:** Based on iterative decomposition, the optimum degree of resource sharing of a hardware implementation of a typical hearing aid algorithm has been found. By means of measurements and simulations, it could be concluded that the most power efficient hardware configuration lies between a fully parallel (isomorphic) and a completely time-shared (processor-like) architecture. It has been shown that the optimum is represented by an architecture that does not break up the fundamental building blocks of the algorithm. The related publication by the author is [3].

**Chapter 4:** The effect of different number formats on the power consumption of building blocks for signal processing algorithms has been studied. Two's complement, sign-magnitude and a combination thereof (hybrid) have been investigated. No general rule for an optimal number format could be established. Nevertheless, it has been shown that the sign-magnitude and hybrid formats are more efficient in FIR filters based on multiply-accumulate (MAC) units, whereas two's complement is better suited in a lattice filter stage that has been implemented as a fully parallel (isomorphic) architecture, where each operation is performed on its own arithmetic unit. Results of this research have been published in [3].

**Chapter 5:** Diverse clocking schemes have been compared with regard to power efficiency. It has been demonstrated that replacing the common single-edge-triggered one-phase clocking by latch-based clocking strategies is more efficient. The savings of these alternatives are due to the simpler clock tree. Additionally, relocating latches can decrease the negative impact of glitch propagation. The findings of this chapter have been published in [4], [5] and [6].



# Chapter 2

## Overview

### 2.1 Hearing Aids

#### 2.1.1 Digital Hearing Aids

Since the nineties, analog hearing aids have been superseded by their more powerful digital counterpart. Nowadays, both types of hearing instruments exhibit comparable power dissipation. Nevertheless, this comparison is not fair, since digital hearing aids contain much more functionality than their analog predecessors. It can even be stated that today, a simple digital device that contains the same functionality as an analog instrument would dissipate less power [7].

In the eighties, when the very first digital hearing instruments have been brought on the market, they were larger in size and very power hungry. These drawbacks prevailed their advantage of increased flexibility compared to the well established analog devices [8]. Not until the nineties with the advent of more advanced CMOS processes featuring smaller transistor sizes, digital hearing aids could compete and even outstrip analog instruments in terms of power consumption.

Today's digital devices comprise a multitude of functions to increase the comfort and quality of life of the hearing impaired patient. A few examples are named here: noise reduction, feedback cancellation, directional microphone, and speech-enhancement [8], all of which would be hardly implementable in analog technique. In reference [1],

which dates back to the nineties, it has been stated that the limits of analog signal processing for hearing aids have been reached at that time. Moreover, it has been anticipated that a further increase in functionality and hence circuit complexity is possible only with digital signal processing.

Today, hearing aid manufacturers distinguish mainly between four types of devices (in order of their package size) [10]:

- completely-in-the-canal (CIC)
- in-the-canal (ITC)
- in-the-ear (ITE)
- behind-the-ear (BTE)

Figure 2.1 shows examples of these types. The instrument on the left is a CIC device, which is hardly visible when worn by the patient. The next two represent different sizes of ITC instruments. The hearing aid in the middle is an ITE type, whereas the three instruments to the right are BTE examples of different sizes.

Depending on the degree of hearing loss, the size of the patient's ear canal and his cosmetic wishes, the appropriate type of hearing aid is chosen by the audiologist.

### 2.1.2 System Overview

Figure 2.2 shows a simplified block diagram of the main signal processing path in a modern hearing aid. Auxiliary blocks, such as clock generation, power supply stabilization and wireless interfaces (remote control, telecoil, bluetooth etc.) have been omitted in the figure since the topic of this work is the power consumption reduction of the digital



Figure 2.1: Different categories of a hearing aid family (© by Bernafon AG [9]).

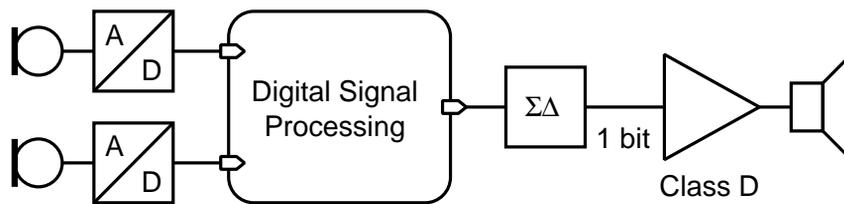


Figure 2.2: System overview of a digital hearing aid.

signal processing path<sup>1</sup>. The example in the figure features the directional microphone technology and hence includes two microphones. Details about this feature are given in Sec. 2.1.4.1.

Typically,  $\Sigma\Delta$ -modulators are used as analog-to-digital converters. Their downsampled output is then forwarded to the main signal processing block, which works with sampling rates around 16 kHz to 22 kHz and input/output word widths of around 16 bit. The output stage consists of a  $\Sigma\Delta$ -modulator, a class-D power-amplifier<sup>2</sup> and the loudspeaker. Note that the impedance of the speaker can be used as the reconstruction low-pass filter [11], [12].

As a rough rule of thumb, it can be stated that a digital hearing aid is powered by a supply voltage of around 1 V, at which it drains about 1 mA. A recent model that features the differential microphone technology (see Sec. 2.1.4.1) is known to be supplied at 0.75 V, at which it drains around 900  $\mu$ A. The digital signal processing circuitry takes around one half of the power budget, whereas the other half is spent in the analog part (voltage regulators, ADC, DAC, microphone and loudspeaker).

### 2.1.3 Power Supply

The most frequently employed type of battery in a hearing aid is a zinc-air element, which provides one of the highest energy densities of all commercially available battery systems [15]. It is cheap in production. Zinc, its anode material, is relatively harmless compared to other anode or cathode metals, such as lithium or mercury, which are used

<sup>1</sup>This does not mean that results presented in this work cannot be applied to the auxiliary blocks just named.

<sup>2</sup>Class-D amplifiers produce binary (on/off) output waveforms.

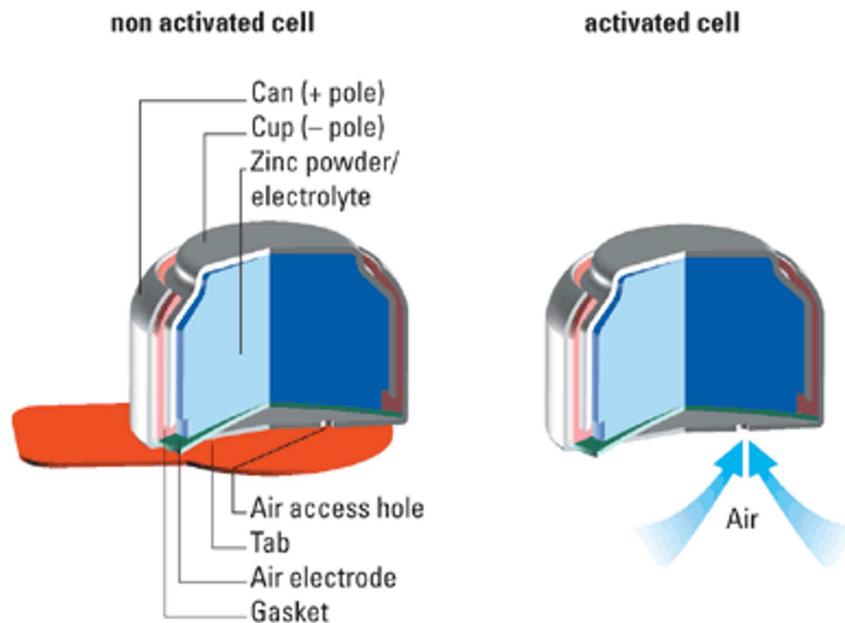


Figure 2.3: Construction of a zinc-air battery (© by Renata SA [13]).

in other battery types. Current-generation is based on the oxidation of zinc, thereby oxygen serves as the cathode reactant.

Inactivated, i.e., sealed to prevent air and hence oxygen to enter the battery, zinc-air elements have long shelf lives. Once activated, they feature an unloaded voltage in excess of 1.4 V. Under the load of a hearing aid, this voltage is slightly reduced, but remains fairly constant until the end of lifetime, where it drops fast [16]. Nevertheless, these cells have some drawbacks; they cannot be recharged electrically and, once activated, they show a high self-discharge rate. Usually, voltage regulators are employed to derive different supply voltages for the parts in the hearing aid system.

Figure 2.3 shows the construction of a zinc-air battery. The sample on the left is sealed, whereas the one on the right has been activated by removing the tab. In this state, the battery is ready to be inserted into the hearing aid. Zinc-air elements are offered with a capacity ranging from 35 mAh to 650 mAh, depending on their physical dimensions [16], [17]. A quick back-of-the-envelope calculation reveals a battery lifetime of around one to two weeks (175 mAh capacity, 16 h activity per day) in a hearing aid. In Fig. 2.4 lifetimes of the models offered by a battery manufacturer are listed.



## Hearing Aid Battery Life

Expected use at 16 hours per day.

Average Hearing Aid Drain (milliamps)	10 (92 mAh) <sup>1</sup> Hours Days*	13 (285 mAh) <sup>1</sup> Hours Days*	312 (160 mAh) <sup>1</sup> Hours Days*	675 (640 mAh) <sup>1</sup> Hours Days*			
0.2	460	29	1425	89	800	50	
0.4	230	14	713	45	400	25	
0.6	153	10	475	30	267	17	
0.8	115	7	356	22	200	13	
1	92	5.8	285	18	160	10	
1.2	77	4.8	238	15	133	8	
1.4	66	4.1	204	13	114	7	
1.6	58	3.6	178	11	100	6.3	
1.8	51	3.2	158	10	89	5.6	
2	46	2.9	143	9	80	5.0	
2.2	42	2.6	130	8	73	4.5	
2.4			119	7	67	4.2	
2.6			110	6.9	62	3.8	
2.8			102	6.4	57	3.6	
3			95	5.9	53	3.3	
4			71	4.5	40	2.5	
5						128	8
6						107	6.7
7						91	5.7
8						80	5.0

<sup>1</sup> Typical 2004 capacity averages for Energizer hearing aid batteries on industry standard tests. Source: Energizer® Research

Figure 2.4: Lifetime of different battery models under different loads (© by Energizer [14]).

## 2.1.4 Signal Processing Algorithms in Digital Hearing Aid Applications

As mentioned in Sec. 2.1.1, modern hearing aids can include a variety of different signal processing algorithms. In this thesis, three of them have been selected as examples in order to study different power consumption reduction techniques. These algorithms are presented in the following subsections.

### 2.1.4.1 Differential Microphone

The differential microphone technology, or “differential microphone” for short, has been presented by G. W. Elko in [18] and [19]. The basic idea of this algorithm is illustrated in Fig. 2.5a.

This technique implements the principle of a *directional* microphone or beamforming by exploiting the phase delay of two omni-directional microphones. Sound, which is represented by the signal  $s(t)$  and its wave vector  $\mathbf{k}$  in Fig. 2.5a, is attenuated or amplified depending on the angle of incidence ( $\theta$ ). This feature can be useful in a hearing aid, since hearing impaired people often suffer from the disability of separating background noise from the wanted signal, usually the conversational partner’s speech which arrives from the front. Hence, the adaptive differential microphone algorithm always assumes that the wanted signal (speech) comes from the front of the hearing impaired person and the unwanted noise originates in the back half-plane.

The signals  $S_f^*$  and  $S_b^*$  from a front and a back omnidirectional microphone are combined to realize two directional responses, represented by the so-called cardioids  $C_f$  and  $C_b$ . Then, the output value  $S_x$  is calculated by the following linear combination with the degree of freedom  $\beta$ :

$$S_x = C_f - \beta C_b. \quad (2.1)$$

Figure 2.5b shows a polar diagram with the plots of the cardioids  $C_f$  and  $\beta C_b$  ( $\beta = 0.5$ ) and output  $S_x$  (solid). The noise comes from  $\nu = 120^\circ$  and the desired speech signal from the front ( $\theta = 0^\circ$ ).

The distance  $d$  between the two microphones is given by

$$d = c \cdot T_d \quad (2.2)$$

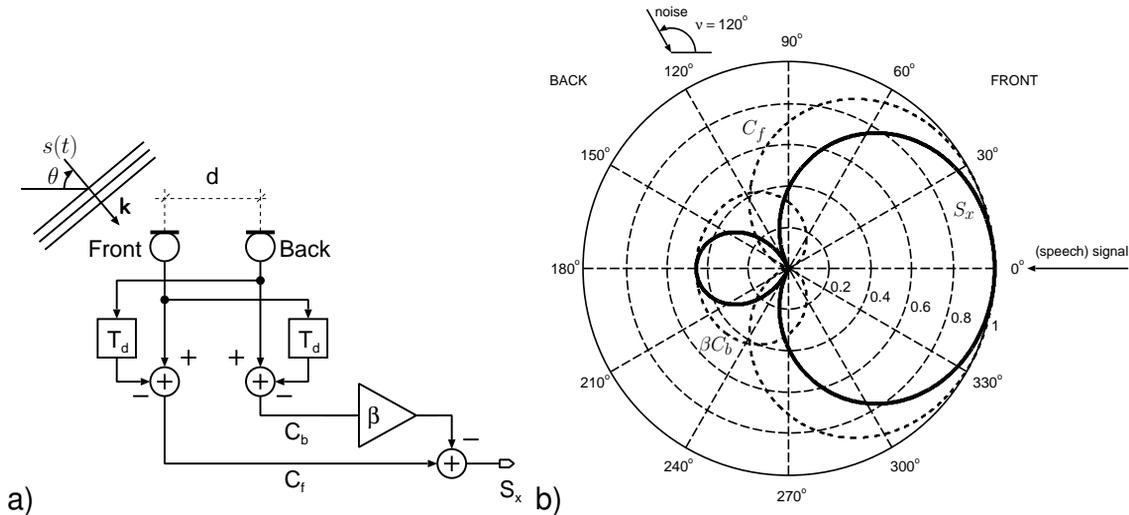


Figure 2.5: a) Differential microphone and b) resulting polar directional sensitivity diagram.

where  $c$  is the speed of sound, and  $T_d$  denotes the sampling period of the system. Figure 2.6 shows a possible implementation of a time-discrete version of this algorithm (green solid frame). It includes a modified creation of the two cardioids  $C_f$  and  $C_b$  (blue dash-dotted frame). A typical hearing aid sampling frequency  $f = 1/T_d$  of 22.05 kHz and  $c = 340$  m/s result in a microphone distance of  $d = 15.4$  mm, which is too large for a small in-the-ear device. Therefore, in this implementation, the sample delays ( $T_d$ ) in Fig. 2.5a cannot be realized by a simple register.

As a solution to this problem, the modified input stage in Fig. 2.6 (blue dash-dotted frame) has been proposed [20]. The delay blocks have been replaced by a symmetric FIR filter of odd order  $N$  with flat frequency response and a delay of  $N/2 \cdot T_d = (N - 1)/2 \cdot T_d + 0.5 \cdot T_d$ . By subtracting the original signals, which are only delayed by  $(N - 1)/2 \cdot T_d$  samples, a difference of 0.5 sample delays results. The distance between the two microphones can now be halved, resulting in  $d = 7.7$  mm. The order  $N$  of the two identical low-pass FIR filters has to be chosen such that an approximately flat frequency response up to almost the Nyquist frequency can be realized. Here  $N$  amounts to 17.

Additional details of a time-discrete implementation are given in Fig. 2.6. Apart from the above mentioned input stage to calculate

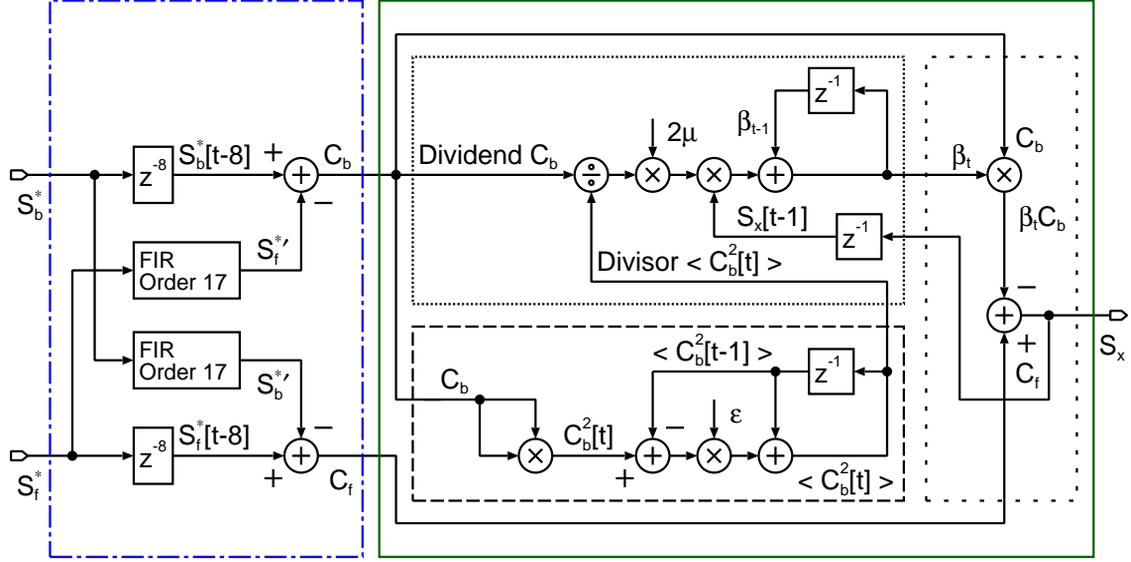


Figure 2.6: Elko's differential microphone algorithm with modified input stage.

the cardioids, the updating of  $\beta_t$  (dashed and fine dotted frames) and the calculation of the output signal  $S_x$  (coarsely dotted frame) are illustrated. The pertaining equations are given in the subsequent paragraphs.

The calculation of the two cardioids  $C_f$  and  $C_b$  can be expressed as follows:

$$C_f[t] = S_f^*[t-8] - S_b'^*[t] \quad (2.3)$$

$$C_b[t] = S_b^*[t-8] - S_f'^*[t]. \quad (2.4)$$

Thereby,  $S_b^*[t-8]$  and  $S_f^*[t-8]$  denote the delayed input signals. The two filtered input signals are represented by  $S_b'^*[t]$  and  $S_f'^*[t]$ , see the blue dash-dotted frame in Fig. 2.6.

The output value  $S_x$  is calculated in the coarsely dotted frame in Fig. 2.6 according to Eq. 2.1. The update equation for of the coefficient  $\beta_t$  is found according to the normalized least mean squares (LMS) algorithm [21]:

$$\beta_t = \beta_{t-1} + 2\mu S_x[t-1] \frac{C_b[t]}{\langle C_b^2[t] \rangle}, \quad 0 < \mu \ll 1. \quad (2.5)$$

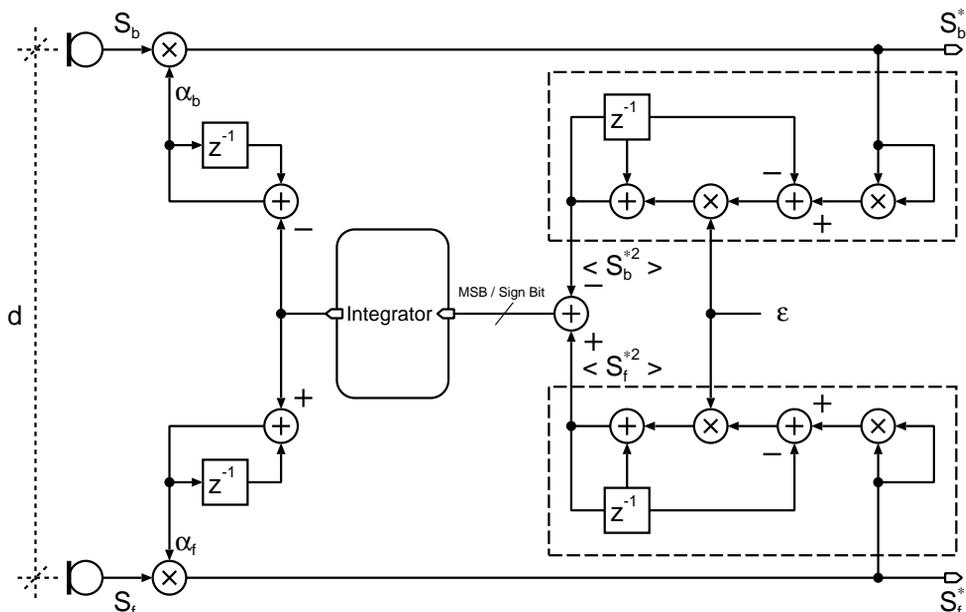


Figure 2.7: Power equalization preprocessor for Elko's differential microphone.

The implementation of this equation is depicted in the fine dotted frame in Fig. 2.6. The step size  $\mu$  has to be chosen small enough to trade precision against speed. The normalization factor, which corresponds to the divisor in Eq. 2.5 is given by:

$$\langle C_b^2[t] \rangle = \varepsilon(C_b^2[t] - \langle C_b^2[t-1] \rangle) + \langle C_b^2[t-1] \rangle, \quad 0 < \varepsilon \ll 1. \quad (2.6)$$

The dashed frame in Fig. 2.6 corresponds to this equation. The division required for the normalization can be performed by a barrel shifter. Thereby, the division is approximated by bit shifting. The shift amount depends on the MSB position of the divisor. The small constant  $\varepsilon$  has been chosen to be  $2^{-8}$ . This enables an efficient hardware implementation of the multiplication in Eq. 2.6 using bit shifting.

Since a small mismatch in the power level of the two microphone signals can reduce the benefit of Elko's algorithm, it has been enhanced to compensate for possible level differences, which can be caused by fabrication tolerances, temperature dependent changes in sensitivity or earwax accumulation. Figure 2.7 shows the corresponding preprocessor. The sign bit of the power difference  $\langle S_f^{*2}[t] \rangle - \langle S_b^{*2}[t] \rangle$  of the output

signals is integrated, and, based thereon, the two coefficients  $\alpha_f$  and  $\alpha_b$  are incremented or decremented, respectively.

#### 2.1.4.2 Channel-Based Noise Reduction

After having removed a considerable fraction of environmental noise by Elko's algorithm, additional noise can be removed by further processing the signal. The patented noise reduction algorithm ([22], [23]) shown in Fig. 2.8 splits its input signal into two signal paths: the main signal processing path (lower) and the analysis path (upper).

The main processing path consists of a symmetric FIR filter of order 47 with variable coefficients. Working in the time-domain keeps the latency small, i.e., in the order of a few microseconds. This would not be possible by an FFT-based solution.

The identical blocks F0–F6 realize a filter bank with 8 sub-bands. Each block is a half-band filter to create down-sampled high-pass and low-pass outputs. The filter bank outputs are then fed into the blocks B0–B7 to calculate the subband signal power in the logarithmic domain. Subsequently, blocks C0–C7 determine the logarithmic power variation range in each sub-band. If this variation does not exceed 15 dB, the related frequency band is attenuated by the difference to 15 dB. Thereby, it is assumed that frequency bands polluted by noise exhibit a more or less constant signal energy. The resulting, so-called gain reduction factors are then used to update the 48 coefficients of the FIR filter in the main signal processing path.

While implementing this algorithm, several optimizations can be applied in order to reduce the computational effort and hence the power consumption. These optimizations are based on the properties of the filters. Careful design of the half-band filters allows the sharing of most calculations between the high-pass and low-pass filters, which are of order 30. All eight high-pass and low-pass filters are identical. Additionally, only 17 out of the total 31 filter coefficients are different from zero. Moreover, they are symmetrical filters, hence, by adding together pairs of data samples (pre-addition) before multiplying by the coefficient, the total number of multiplications can be halved. The pertaining equation pair for each of the 8 half-band filter can be

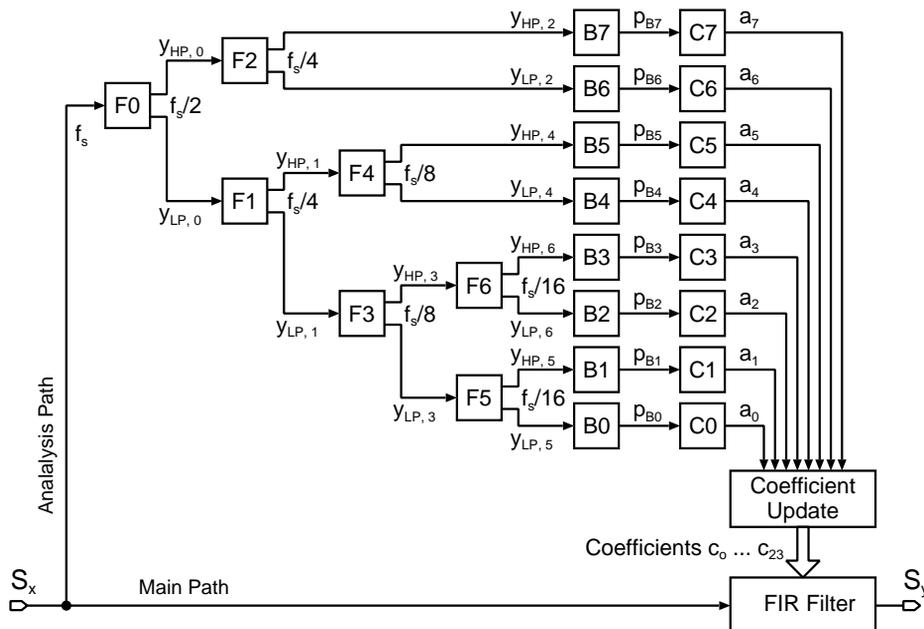


Figure 2.8: Channel-based noise reduction.

written as:

$$y_{LP,i}[t] = A + B \quad (2.7)$$

$$y_{HP,i}[t] = A - B \quad (2.8)$$

where,

$$A = b_0 x[t - K]$$

$$B = \sum_{k=1,3,\dots}^K b_k \underbrace{(x[t - K - k] + x[t - K + k])}_{\text{pre-addition}}$$

$$b_k = 0 \text{ if } k = 2, 4, \dots$$

$$i = 0, \dots, 6; K = 30/2 = 15 \text{ (half of the filter order).}$$

From Eqs. 2.7 and 2.8 it can be seen that the low-pass and high-pass output differ in only one single operation and hence the computation of  $A$  and  $B$  can be shared between the two filters. Those non-vanishing filter coefficients  $b_k$  are known at design time and stored in a lookup table (LUT). The set of coefficients  $b_k$  is identical for all half-band filters. Each filter output has to be down-sampled by a factor of two, discarding every second output sample. By making use of the

“Noble Identities” [24], the calculation of these unused samples has been avoided.

The filter bank outputs are then fed into the units B0–B7 to calculate the sub-band signal power in the logarithmic domain:

$$p_{Bj} = \log \sum_{n=1}^N y_{\{\text{LP, HP}\}, l}^2[n], \quad j = 0, \dots, 7; l = 5, 6, 4, 2. \quad (2.9)$$

Depending on the sampling frequency  $f_s$  of the related filter output,  $N$  amounts to 32 for B0 to B3, 64 for B4 to B5 and 128 for B6 to B7.

In an iterative process, blocks C0–C7 update an upper and a lower bound,  $s_{uj}$  and  $s_{lj}$ , respectively, of each sub-band  $j$ :

$$s_{uj, t+1} := \max(s_{uj, t} - \delta, p_{Bj}) \quad (2.10)$$

$$s_{lj, t+1} := \min(s_{lj, t} + \delta, p_{Bj}) \quad (2.11)$$

$$j = 0, \dots, 7; \delta = 0.25 \text{ dB.}$$

Additionally, the power variation ranges  $r_j$  are iteratively updated:

$$r_{j, t+1} := r_{j, t} + \gamma \cdot (s_{uj, t} - s_{lj, t} - r_{j, t})^3, \quad 0 < \gamma \ll 1. \quad (2.12)$$

Then the logarithmic gain reduction factors can be calculated:

$$\Delta G_j = \max(A - \beta \cdot r_{j, t}, 0), \quad A = 15 \text{ dB}; \beta \approx 1. \quad (2.13)$$

These factors are transformed back into the linear domain yielding  $a_j$ :

$$a_j = 10^{\Delta G_j / 20}. \quad (2.14)$$

These linear factors  $a_j$  are used to periodically update the complete set of coefficients of the symmetric FIR filter in the main processing path, whose output can be written as:

$$S_y[t] = \sum_{m=0}^{M-1} c_m \underbrace{(x[t - M - m] + x[t - (M - 1) + m])}_{\text{pre-addition}}. \quad (2.15)$$

The symmetry of the impulse response has again been employed by making use of a pre-adder. For this reason, the upper bound of

summation  $M$  in Eq. 2.15 is equal to 24, half of what would be required if no pre-adder was used<sup>3</sup>. The coefficients  $c_m$  are given according to:

$$c_m = h_m \sum_{j=0}^{J-1} a_j \cdot c_{m,j}, \quad m = 0, \dots, M-1, \quad (2.16)$$

where  $h_m$  are coefficients of a Hamming window, which are pre-calculated at design time and stored in a LUT. The upper bound of summation  $J$  is given by the number of sub-bands, in our case  $J = 8$ . The factors  $c_{m,j}$  can be interpreted as coefficients of eight band-pass filters that are combined in parallel and whose outputs are attenuated by the gain reduction factors  $a_j$ . This can be seen by inserting Eq. 2.16 into Eq. 2.15 and by exchanging the order of summation:

$$S_y[t] = \sum_{j=0}^{J-1} a_j \sum_{m=0}^{M-1} h_m \cdot c_{m,j}(x[t-M-m]+x[t-(M-1)+m]). \quad (2.17)$$

Again, the band-pass coefficients  $c_{m,j}$  are pre-calculated at design time and stored in a LUT.

### 2.1.4.3 A Model Hearing Aid System

The two presented hearing aid algorithms, differential microphone and channel-based noise reduction, are well suited to be combined sequentially, giving rise to the system depicted in Fig. 2.9. After exploiting the spatial diversity between signal and noise, the SNR is additionally improved by suppressing noisy frequency bands by the channel-based noise reduction block.

This system serves as a model to study different low-power techniques throughout this work. In Chapter 5, different implementations of these algorithms are presented.

### 2.1.4.4 Speech Enhancing through Spectral Sharpening

Hearing impairment is often accompanied with a reduced frequency selectivity which leads to a decreased speech intelligibility in noisy

---

<sup>3</sup>The FIR filter is of order 47, which requires 48 coefficients. Because these are symmetric, only half, i.e., 24 equal pairs are actually required to be processed.

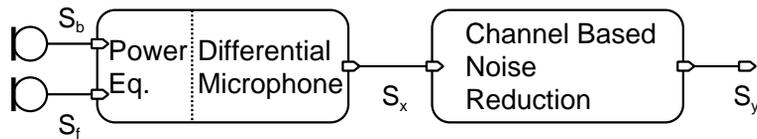


Figure 2.9: Block diagram of a digital hearing aid with a differential microphone and channel-based noise reduction.

environments. One possibility to alleviate this deficiency is the spectral sharpening for speech enhancement as proposed in [25]: based on adaptive filtering, the frequency contributions which are important for intelligibility in the speech (formants) are identified and accentuated. The idea of the algorithm is based on a concept presented in [26]. A block diagram is given in Fig. 2.10.

Again, the input signal is divided into two paths. The lower main signal path contains an analysis and a synthesis filter as central processing blocks. In the upper path, the decorrelator calculates the so-called “reflection coefficients”  $k_1, \dots, k_8$  that are used in the two filters in the lower path. The decorrelator signal output itself is not used.

In the lower branch, at first, the signal is processed by means of the analysis filter  $1 - A(z/\beta)$  and then by the synthesis filter  $[1 - A(z/\gamma)]^{-1}$ . Since  $A(z)$  can be written as a polynomial, it immediately follows that the analysis is a FIR filter, whereas the synthesis is an IIR or all-pole filter. The actual speech sharpening takes place in the synthesis filter. The analysis filter, which makes use of the same coefficient

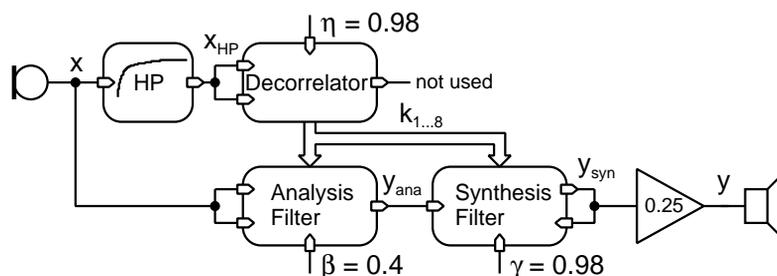


Figure 2.10: Block diagram of speech enhancing through spectral sharpening.

set, serves to compensate the spectral roll-off of the synthesis filter<sup>4</sup>. It adds zeroes, which have the same phase angle as the poles of the synthesis filter, but smaller radii in the pole-zero diagram, into the transfer function of the system. The degree of spectral sharpening and compensation of spectral tilt is given by the two parameters  $\beta$  and  $\gamma$ . Choosing  $0 < \beta < \gamma < 1$  results in a spectral sharpening effect. If  $\beta$  is set equal to  $\gamma$ , the overall signal processing transfer function would be the identity.

The analysis- and synthesis-filter, as well as the decorrelator, are implemented as lattice-filters (Chapter 6.6 in [27] and Chapter 5 in [28]). As suggested in [25], a filter order of eight has been chosen for these three main building blocks. Lattice filters of order  $N$  are a cascade of  $N$  stages, such stages are shown in Figs. 2.11 and 2.12. Implementing an 8th-order filter as an isomorphic architecture<sup>5</sup> would require to line up eight such stages with three multipliers, two adders, and one register, each.

As depicted in Fig. 2.11 by the  $i$ -th stage in the middle, an analysis stage is given by the following set of formulae [27]:

$$u_i[t] = u_{i-1}[t] - \beta k_i l_{i-1}[t - 1], \quad i = 1, 2, \dots, N, \quad (2.18)$$

$$l_i[t] = \beta l_{i-1}[t - 1] - k_i u_{i-1}[t], \quad i = 1, 2, \dots, N. \quad (2.19)$$

When joining together  $N$  such stages to compose a filter of order  $N$ , the first (left-most) has a common input for the lower and upper

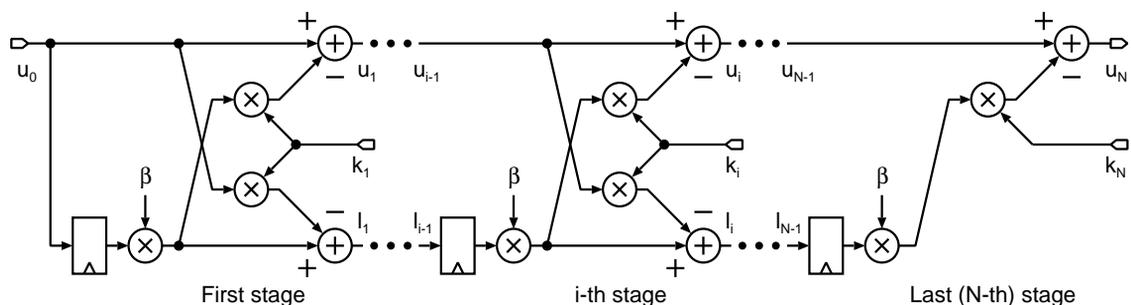


Figure 2.11: Structure of the analysis filter.

<sup>4</sup>For the same reason, a first-order high-pass filter  $(1 - z^{-1})$  precedes the decorrelator [25], [26].

<sup>5</sup>In an isomorphic architecture, each operation is executed by its own dedicated hardware unit, i.e., adder, multiplier. See Sec. 3.1.1 for a more thorough discussion.

branch, i.e.,  $l_0[t] = u_0[t] = x[t]$ . The last (right-most) stage can be simplified by removing the path from the upper left to the lower right and by discarding the addition in the lower path, i.e.,  $y_{ana}[t] = u_N[t]$ . These simplifications can be seen in Fig. 2.11 to the left and right.

Figure 2.12 shows a synthesis stage, described by:

$$f_{i-1}[t] = f_i[t] + \gamma k_i b_{i-1}[t-1], \quad i = 1, 2, \dots, N, \quad (2.20)$$

$$b_i[t] = \gamma b_{i-1}[t-1] - k_i f_{i-1}[t], \quad i = 1, 2, \dots, N. \quad (2.21)$$

Similar simplifications as in the analysis filter are possible here:  $f_N[t] = y_{ana}[t]$  and the output  $b_N[t]$  is not used. Hence, the units in the dotted frame in Fig. 2.12 can be omitted in the very first stage. Additionally, the filter output is fed back:  $y_{syn}[t] = f_0[t] = b_0[t]$ , which results in an IIR filter when lining up such stages. The parameters  $\beta$  and  $\gamma$  are constant throughout all eight stages of each type.

Figure 2.13 shows the structure of one section of the decorrelator (adaptive gradient decorrelator, Chapter 5.4.1 in [28]). The upper part in this figure forms the decorrelator filter, whose output is not required. It shares the structure with the analysis filter, the only difference being the omission of the constant multiplication with  $\beta$ . The simplifications similar to the analysis filter cannot be applied here in the last stage. In the lower part, the filter coefficients are calculated (one per stage), i.e., eight decorrelator stages are needed for eighth-order analysis- and synthesis filters. The necessary division has been implemented by means of a barrel shifter, whose shift value is controlled by the position of the most significant non-zero bit of the divisor. The introduced error is barely audible.

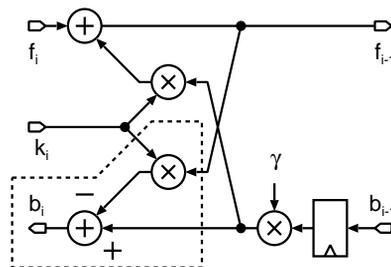


Figure 2.12: Structure of the synthesis filter stage.

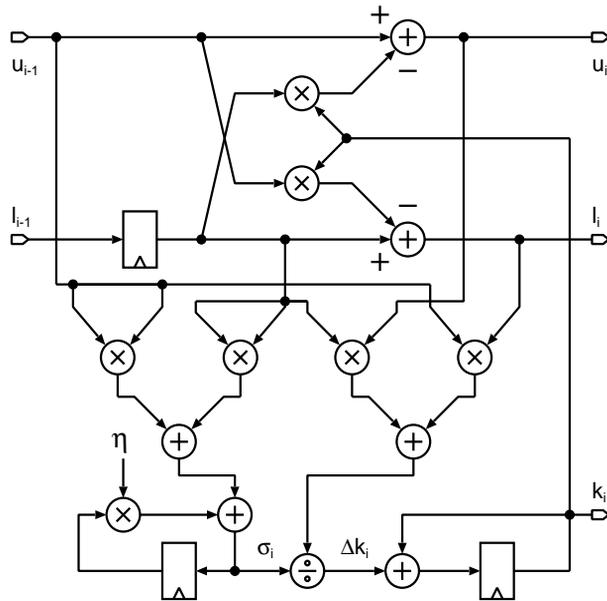


Figure 2.13: Structure of the decorrelator stage.

## 2.2 Power Consumption in Digital CMOS Circuits

### 2.2.1 History of CMOS Circuits

The idea of CMOS circuits has first been presented by F. M. Wanlass in 1963 [29]. He realized an inverter and other fundamental logic gates, such as a NOR-gate, out of discrete silicon MOS transistors. He also proposed and patented the concept of monolithic CMOS devices, where both transistor types are placed on the same piece of silicon. Nevertheless, he was not able to produce such a single-chip device [30], because at that time, it was not yet possible to fabricate enhancement-mode n-channel devices. Hence, he used discrete depletion-mode n-type MOSFETs and back-biased them accordingly to proof the feasibility of his concept.

His first CMOS inverter consumed a few nanowatts of standby power, which was around six orders of magnitude less than equivalent bipolar and pMOS gates at that time [30]. He measured a propagation delay of less than 100 ns, this was around half the speed of the traditional circuit technologies. Most important however was Wanlass's

final remark that it should be possible to construct field-effect transistor circuits with a very high packing density. History has shown that his statement was absolutely true, and, according to Moore's Law, package density increased, and is still increasing. It is interesting to note is the fact that at that time, Gordon Moore – articulating his famous law two years later in 1965 [31] – was Wanlass's manager at Fairchild Semiconductor [32].

Nevertheless, CMOS was considered a slow, but low-power circuit technology. Most applications asked for high speed, and power consumption was not yet an issue. Any type of solid-state electronics was low-power compared to vacuum tubes, whose era had already ended then. In spite of that, there were some niche applications that demanded real low-power circuits. One prominent example were – and still are – quartz wristwatches [33]. A first prototype with a chip in bipolar technology has been presented by CEH, a Swiss research laboratory, in 1967 [34], [33]. The first quartz wrist watch was brought to market by Seiko in 1969 [35]. Shortly after, the bipolar chips have been replaced by their CMOS counterpart, since in the meantime, it became possible to integrate monolithic CMOS devices [36], [37]. This development has been driven by the watch industry [38], [39].

However, until the eighties of the last century, CMOS lived in the shadow of bipolar and nMOS technologies. The latter was preferred to pMOS because it is faster thanks to the higher carrier mobility. The situation started to change when packing density of integrated circuits grew so high that heat removal became a severe problem [40], [41]. Additionally, portable consumer electronic products were commercialized that asked for low-power to increase battery lifetime. These circumstances paved the way for the shift from traditional circuit technologies to CMOS. Nevertheless, this change does not mean that the issue of power consumption has been solved once and for all, be it because of heat removal or battery life.

As predicted by Moore's Law, transistor sizes are steadily shrinking. While proceeding from one CMOS technology generation to the next one, (dynamic) power consumption of a given design is certainly decreasing. The ongoing trend to integrate more and more functionality on the same silicon area, enabled by smaller transistors, somehow offsets the initial low-power advantage of CMOS. The packing density has become so high that again, heat removal is a primary issue in

high-speed applications such as microprocessors. In other words, in terms of power consumption, CMOS has reached the point where its predecessor arrived about ten years ago, before being superseded by CMOS itself [41]. On the other hand, portable applications, where heat removal is usually not a primary problem, include more and more functionality. Additionally, they increasingly penetrate our daily life. In order to retain reasonable battery lifetimes, their power consumption cannot be neglected. One could be tempted to say that Moore's Law is the enemy of any low-power chip designer.

Certainly, hearing aid chips belong to the second category of low-power applications. Heat removal is of no importance; instead, they demand low-power techniques in medium-complexity chips because the capacity of their energy supply (battery) is highly limited by physical dimensions.

### 2.2.2 Sources of Power Consumption

Figure 2.14 shows a CMOS inverter made up of a transistor pair (MOSFETs). This simplest digital CMOS element serves as a model for more complex combinational units, such as NAND, NOR, AOI and other gates, and sequential units (flip-flops and latches). It allows the study of the processes that contribute to the overall power dissipation in any static CMOS cell. To simplify the drawing, body or substrate connections have been omitted in this figure.

Usually, the gates of such a transistor pair are driven by another pair which is part of a preceding CMOS unit. This driving pair

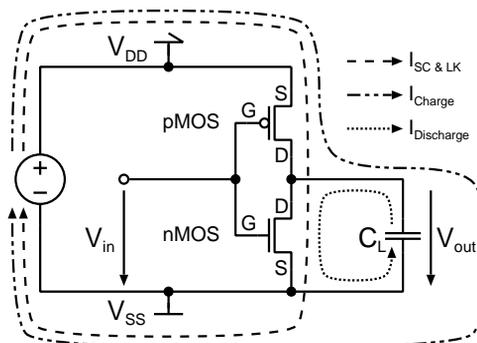


Figure 2.14: Power consumption of a CMOS inverter.

creates the input voltage  $V_{in}$  in Fig. 2.14. The load consists of all gate capacitances of CMOS pairs in the fanout, plus the capacitance of the connecting wire. The total load capacitance is lumped together in  $C_L$ , the voltage over this load, i.e., the output voltage, is denoted as  $V_{out}$ .

Power consumption of a static CMOS element mainly consists of two parts:

- Dynamic power consumption
  - Switching power
  - Short-circuit or cross-over power
- Static power
  - Subthreshold leakage power

Note that in reality, the static power consists of further components, a more detailed discussion of which can be found in [42]. The currents giving rise to the above listed power dissipations are drawn in Fig. 2.14, these are namely the short-circuit and leakage currents ( $I_{SC\&LK}$ ), and the currents associated with the charging and discharging of the load capacitance ( $I_{Charge}$  and  $I_{Discharge}$ ).

### 2.2.2.1 Dynamic Power Consumption

As suggested by the name, dynamic power consumption emerges from currents flowing because the input of a CMOS element toggles, i.e., undergoes a transition from logic ‘1’ to ‘0’ or vice versa. In case of an inverter, this leads to a transition of the output in the opposite direction. In case of more complex cells (combinational and sequential), it is also possible that no transition or a transition in the same direction occurs.

Anyway, a transition at the input of a CMOS cell leads to dynamic power dissipation, whose extent depends on the logic function, the direction of the change at the input, the state of possible other inputs, and, in case of a sequential cell, the previous state<sup>6</sup>. Additionally, it is depending on electrical parameters as shown in the following.

In the simple case of an inverter, as in Fig. 2.14, the energy drawn from the power supply can be calculated as follows. Consider a

---

<sup>6</sup>For instance in a D-type flip-flop (D-FF) comprising a master and a slave latch (see Fig. 5.6), internal dynamic power consumption caused by the active clock edge depends on whether the input and output of the slave latch are different or not. The input of the slave corresponds to the state of the D-FF.

transition from ‘1’ to ‘0’ at the input and consequently the output going from ‘0’ to ‘1’. Since static CMOS circuits feature rail-to-rail swing signals, the load capacitance  $C_L$  is charged to  $V_{DD}$ , while the input transitions to  $V_{in} = V_{SS} = 0V$ . The current which is charging  $C_L$  is given as:

$$I_{Charge}(t) = C_L \frac{d}{dt} V_{out}(t). \quad (2.22)$$

This current flows through the conducting pMOS transistor, which can be regarded as a general nonlinear resistor, while the nMOS transistor is considered as an ideal turned off switch. Integrating the instantaneous power yields the drawn switching energy:

$$E_{Switch} = \int_0^T V_{DD} \cdot I_{Charge}(t) dt. \quad (2.23)$$

Substitution of Eq. 2.22 into Eq. 2.23 and adaption of the limits ( $V_{out}(t)|_{t=0} = 0$ ,  $V_{out}(t)|_{t=T} = V_{DD}$ ) gives:

$$E_{Switch} = V_{DD} C_L \int_0^{V_{DD}} dV_{out} = C_L V_{DD}^2. \quad (2.24)$$

This energy drawn from the power supply is independent of the waveforms, their transition times and the resistor characteristics of the conducting pMOS transistor. It is well known that the energy stored in a charged capacitor is given by the general formula  $E = \frac{1}{2}CV^2$ , i.e., in our case:

$$E_{C_L} = \frac{1}{2} C_L V_{DD}^2. \quad (2.25)$$

From the principle of conservation of energy directly follows:

$$E_{pMOS} = E_{Switch} - E_{C_L} = \frac{1}{2} C_L V_{DD}^2. \quad (2.26)$$

This means that during charging the load capacitance  $C_L$  half of the total drawn energy is transferred to  $C_L$  while the other half is dissipated and converted into heat in the pMOS transistor. It immediately follows that for all output transitions from ‘1’ to ‘0’ (because the input toggles in the other direction), the capacitance  $C_L$  is discharged via the nMOS transistor and the converted energy equals:

$$E_{nMOS} = \frac{1}{2} E_{Switch} = \frac{1}{2} C_L V_{DD}^2. \quad (2.27)$$

Similarly, it is assumed here that the pMOS is switched off ideally.

These calculations are valid for a single period, this means for a ‘0’–‘1’–‘0’ transition cycle at a single inverter output. Typically, one is interested in the switching power consumption of a complete chip running at a certain clock frequency. Therefore, it is necessary to expand the above formula (Eq. 2.24):

$$P_{Switch} = V_{DD}^2 f_{ci} \sum_{k=1}^K C_{L,k} \frac{\alpha_k}{2}. \quad (2.28)$$

The sum over all  $K$  nodes yields the total power consumption in the chip. The factor  $f_{ci}$  denotes the inverse of the computation interval  $T_{ci}$  ( $f_{ci} = 1/T_{ci}$ ) of the design. In case of the common single-edge-triggered one-phase clocking,  $T_{ci}$  is equal to the clock period and hence  $f_{ci} = f_{clk}$ . More details on clocking strategies will be discussed in Chapter 5. The factor  $\alpha_k$  takes into account that not all nodes inside the chip show the same activity. This factor, called “switching activity” or “toggle rate factor”, corresponds to the average number of toggles a node undergoes during one computation interval. In the already mentioned single-edge-triggered one-phase clocking, the clock signal itself exhibits an  $\alpha_{clk}$  which is equal to two [43]. Finally,  $C_{L,k}$  denotes the load capacitance associated with node  $k$ .

Unfortunately, the assumption that one transistor of the MOSFET pair is conducting and the other one is completely switched off while the input is subject to a transition is too optimistic. In fact, as presented in [42] and [44], a short-circuit current  $I_{SC}$ , also referred to as cross-over current, is flowing from  $V_{DD}$  to  $V_{SS}$  through the transistors of both types, motivating its name. This happens, whenever  $V_{TH,nMOS} < V_{in} < V_{DD} - |V_{TH,pMOS}|$ , where  $V_{TH,nMOS}$  and  $V_{TH,pMOS}$  denote the threshold voltages of the nMOS and pMOS transistors, respectively. By making use of the  $\alpha$ -power MOSFET model [45], the short-circuit energy for a rising plus a falling transition at the input can be expressed as follows:

$$E_{SC} \Big|_{t_T \gg \tau_N} = V_{DD} t_T I_{D0}(V_{DD}) \frac{1}{\alpha + 1} \frac{1}{2^{\alpha-1}} \frac{(1 - 2V_{TH}/V_{DD})^{\alpha+1}}{(1 - V_{TH}/V_{DD})^\alpha}. \quad (2.29)$$

To derive this formula, the driven capacitance  $C_L$  has been set to 0 pF, hence it can be considered as an approximation for the case, where the

transition time  $t_T$  of  $V_{in}$  is much larger than the one at the inverter output  $\tau_N$ . Under these circumstances, the nMOS transistor will be in saturation. Additionally, the inverter is assumed to be symmetric, which means  $V_{TH} = V_{TH,pMOS} = V_{TH,nMOS}$ . The drain current at  $V_{GS} = V_{DS} = V_{DD}$  is denoted by  $I_{D0}(V_{DD})$ . This so-called saturated drain current is linearly dependent on the transistor dimension ratio  $W/L$ . The transistor parameter<sup>7</sup>  $\alpha$  is called the velocity saturation index. Setting  $\alpha$  equal to 2, i.e., unsaturated carrier velocity because of a long channel, results in the traditional Shockley model for the MOSFET in the saturation region, whereas the ultimate short channel would result in  $\alpha = 1$  [45]. Using the method described in [45], values of  $\alpha_n = 1.2$  and  $\alpha_p = 1.44$  for nMOS and pMOS transistors, respectively, in a 0.25  $\mu\text{m}$  technology have been found. Although Eq. 2.29 is an approximation, some conclusions can be drawn: as reported in [5],  $E_{SC}$  increases linearly with the transition time  $t_T$  at the inverter input. Additionally, since  $I_{D0}$  super-linearly depends on  $V_{GS}$  and hence also on  $V_{DD}$ , the dependency of  $E_{SC}$  on the supply voltage is more than quadratic. In [46] and [44] it has been shown that the short-circuit power of an inverter is minimal when the transition times at the input and output are equal.

### 2.2.2.2 Static Power Consumption

Static power consumption is caused by leakage current that is flowing from  $V_{DD}$  to  $V_{SS}$  through the transistors. Examples of this type are subthreshold leakage and pn-junction reverse bias (junction leakage) currents [42]. Also gate oxide tunneling currents constitute to static power.

### 2.2.2.3 Taxonomy in Design Tools

Today's design tools support the estimation of the power consumption of a design under development. Usually, these estimations are based on gate-level simulations performed on netlists either after synthesis or after final placement and routing (also known as post-layout). Apparently, post-layout estimations yield more accurate results [47].

---

<sup>7</sup>This transistor parameter  $\alpha$  is not to be confused with the node switching activity  $\alpha_k$ .

The estimation tool collects the toggle activities  $\alpha_k$  of all nodes in the design. Subsequently, the netlist is back-annotated with these toggle activities and, in case of a post-layout simulation, parasitic and timing information is considered as well. Then, the estimation tool<sup>8</sup> creates the power reports on the basis of power figures included in the cell library.

Typically, the taxonomy of dynamic power consumption in such tools does not strictly adhere to the one presented above. Instead, they differentiate the dynamic power between switching and internal power (Synopsys [48]) or net and instance power (Cadence [49]), respectively. Switching or net power denotes the dissipation consumed by charging and discharging the load capacitance of the cell, which consists of the sum of the capacitance of the wire plus the driven transistor-gate capacitances. Internal or instance power is used for the total dissipated power inside a cell. It consists of the short-circuit plus the switching power associated with charging and discharging of internal node capacitances. This simplification is justified because in the analog simulations for the power characterization of a cell, a strict separation between short-circuit and cell-internal switching power would be very difficult.

### 2.2.3 Countermeasures

Equations 2.24 and 2.28 provide a basis for different methods to reduce (dynamic) energy consumption in digital CMOS designs. Dissipated dynamic energy can be saved by reducing:

- Supply voltage  $V_{DD}$
- Clock frequency  $f_{ci}$
- Transistor gate and wire capacitance  $C_{L,k}$
- Switching activity  $\alpha_k$
- Circuit complexity  $K$

As a positive side effect, the reduction of the supply voltage lessens static power. If gate capacitances are lowered by shrinking the transistor width, static power is minimized as well.

Unfortunately, the above listed factors cannot be lowered independently. For instance, as will be seen in Chapter 3, energy can be

---

<sup>8</sup>e.g., Power Compiler of Synopsys Design Compiler [48] or Low Power Synthesis (LPS) flow for Cadence BuildGates [49].

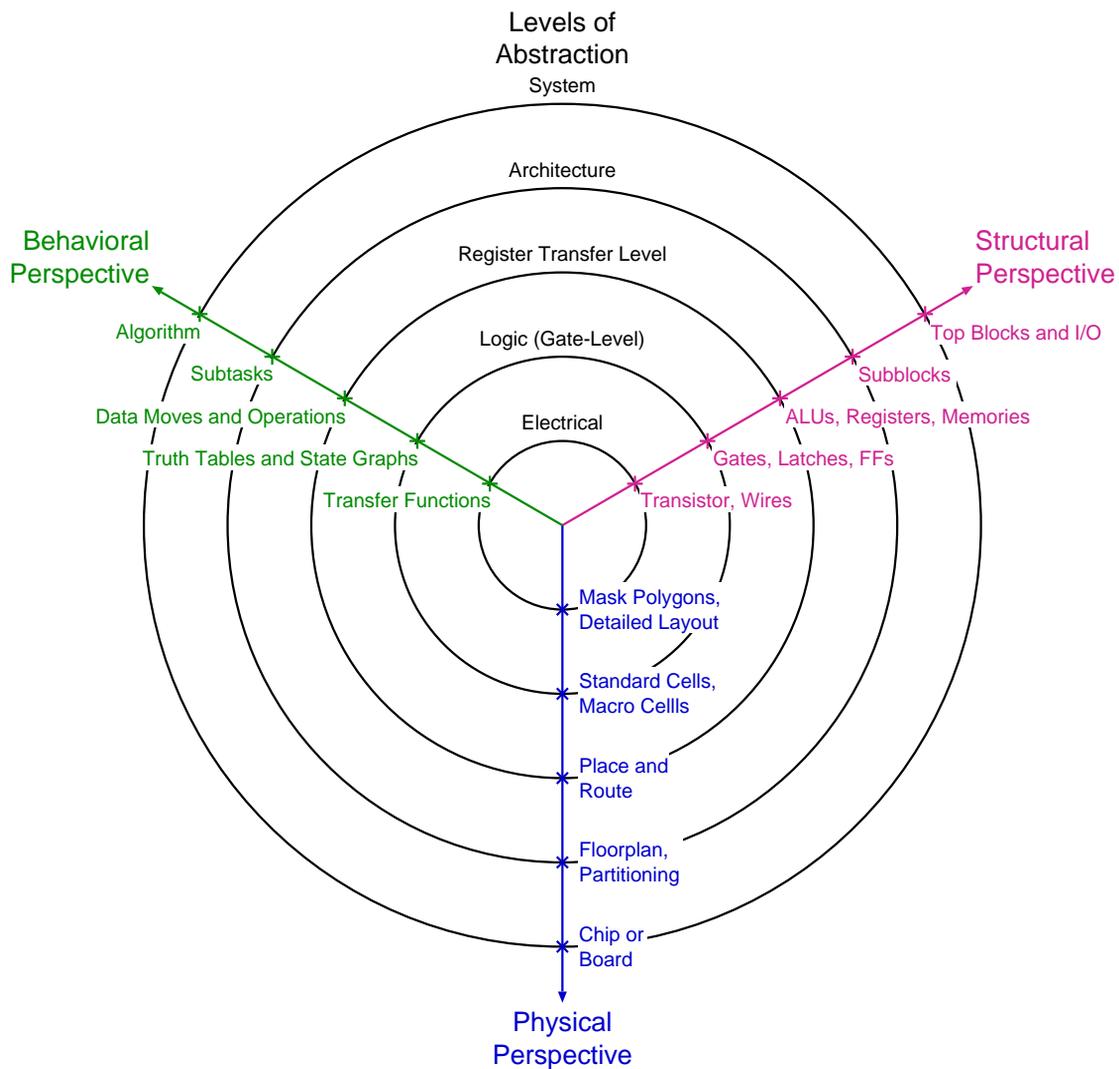


Figure 2.15: Gajski's Y-chart as shown in [43].

saved by an architecture transformation that yields an activity and complexity reduction, but at the cost of an increased clock frequency.

The factors listed above are restricted to those which a circuit designer can influence, that is, without modification of the underlying CMOS process technology. They apply to different abstraction levels in digital systems according to Gajski's Y-chart as shown in Fig. 2.15 [50], [43]. For instance, switching activity reduction can take place at different levels. Depending on the chosen algorithm for a given computational task, a different overall toggle rate will result. Clock gating, another technique to lower the toggle activity, applies

to the gate-level. Thereby, the clock network activity is minimized. Down-scaling of transistor gate capacitances certainly belongs to the electrical level. And as a last example, reducing the transistor gate capacitance by shrinking the transistor width certainly belongs to the electrical level, which represents the lowest level of abstraction.

In the remainder of this thesis, several techniques based on these factors are discussed and compared in order to reduce energy consumption in digital circuits.

# Chapter 3

## Architectures for Low-Power Circuits

### 3.1 Architectural Trade-Offs

In this section an optimum hardware architecture in terms of toggle activity and hence power consumption will be derived for the example of the spectral sharpening algorithm shown in Sec. 2.1.4.

#### 3.1.1 The Isomorphic Architecture

Implementing a digital signal processing algorithm usually starts with the related formulas, possibly re-arranging and simplifying them by some mathematical equivalence transformations [43]. For instance, in Eqs. 2.7, 2.8 and 2.15, the symmetry of the filter coefficients has been exploited. The next step consists in the derivation of a data dependency graph (DDG). This representation of the algorithm results from the direct mapping of each occurrence of an operation in the formulae to a dedicated operator (adder, multiplier, delay element) in the graph. Then, the register-transfer level (RTL) hardware architecture is devised. Each operator in the data-dependency graph is mapped to a dedicated hardware unit (adder unit, multiplier unit or register). The resulting hardware organization is called *isomorphic architecture*, since DDG

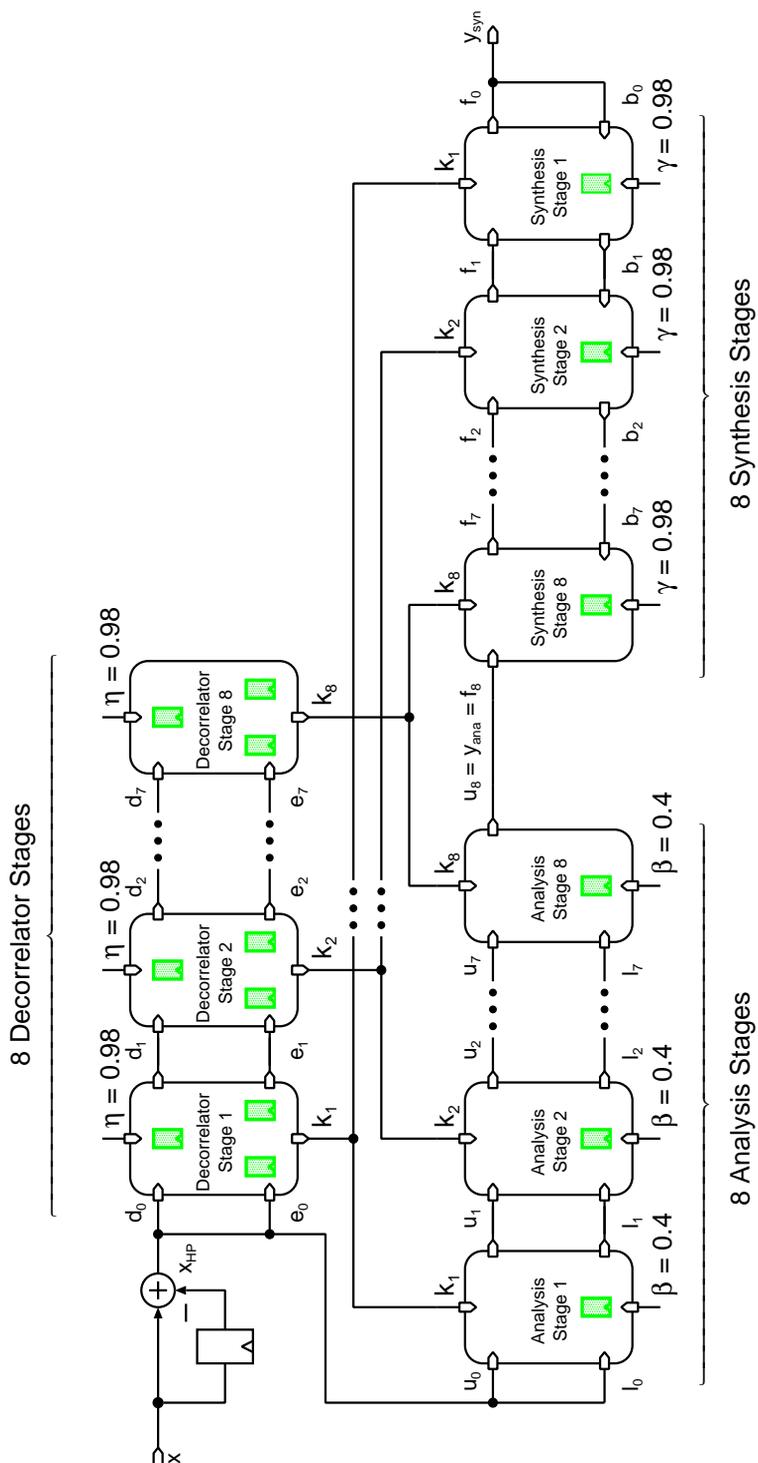


Figure 3.1: Isomorphic architecture of the speech-enhancement algorithm.

and block diagram of the architecture are isomorphic [43]: There is a direct correspondence between each instance of any operation in the equations and each hardware element. These architectures are also known under the name “fully-parallel architectures” [40]. To summarize, they show the following characteristics [43]:

1. each operation is executed in its own hardware unit,
2. no control is needed, since the data dependency graph and the block diagram are isomorphic, and
3. clock for all registers and sample rate are identical.

Figures 2.11, 2.12, and 2.13 show such isomorphic architectures for the single lattice-filter stages based on Eqs. 2.18–2.21. The isomorphic architecture of the complete speech-enhancement algorithm is depicted in Fig. 3.1. Because of practical reasons, the detailed schematics of the single filter stages given in Figs. 2.11, 2.12, and 2.13 have not been drawn. Though, the correct number of registers are sketched in light green color in the boxes representing the stages. Since a filter order eight is realized by the algorithm, eight stages have to be instantiated for both analysis and synthesis filters, and decorrelator in the isomorphic architecture.

Most of the additions and multiplications are 32 bit wide. The constant multiplications by  $\beta$  in the analysis and by  $\gamma$  in the synthesis stages are 28 bit wide, whereas the multiplication with  $\eta$  in the decorrelator is 44 bit wide. The adder for the update of the reflection coefficients  $k_i$  is 16 bit wide. The widest adder (44 bit) is used to update the sum of the signal powers  $\sigma$  in the decorrelator. These bit numbers illustrate the huge hardware effort required for the isomorphic architecture.

### 3.1.2 Iterative Decomposition

As a consequence of chip-area constraints, signal processing algorithms are usually implemented in iteratively decomposed architectures, in which multiple operations are scheduled to run on a few processing units [51], [52]. In other words, iterative decomposition of a function  $f$  means to break it up into subtasks which are then executed in a step-by-step fashion on fewer (or even only one) computational units. This approach is also referred to as resource sharing or time-multiplexing [43], [40]. A single-core microprocessor is the extreme

case of such an architecture with only one computational unit, the arithmetic logic unit (ALU)<sup>1</sup>. Another typical example is a finite-impulse-response (FIR) filter implemented by means of one MAC unit, which is controlled by a simple controller. The opposite extreme would be the isomorphic architecture, in which  $N$  adders and registers, and  $N + 1$  multipliers would be instantiated. Obviously, the larger the level of decomposition, the higher the clock frequency needs to be to maintain data throughput.

In order to compare different degrees of iterative decomposition, the “ID-index”  $x$  of a design  $y$  is defined as:

$$x := \frac{f_{clk, Design\ y}}{f_{clk, Isomorphic\ Design}}. \quad (3.1)$$

Each design is named as  $IDx$ , where  $x$  denotes its ID-index [3]. This index is a measure for the level of iterative decomposition, and hence the degree of resource sharing.

## 3.2 Spectral Sharpening Chip Design

### 3.2.1 Implemented Architectures

In search of the optimum degree of iterative decomposition concerning power consumption, numerous functionally identical architectures of the speech enhancing algorithm presented in Sec. 2.1.4.4 have been devised by iterative decomposition:

ID1: By definition, this implementation corresponds to the isomorphic architecture presented in Sec. 3.1.1. It serves as a starting point from which the other designs are derived. Certainly, this architecture occupies the largest area.

ID2: Operating at doubled frequency, this design does with only four stages in both filters and the decorrelator in Fig. 3.2. The samples are looped twice through the stages. This architecture has been integrated on silicon.

---

<sup>1</sup>What differentiates microprocessors from the architectures considered in this work is the fact that the ALU is controlled by a software program and not by a hardwired control block (FSM).

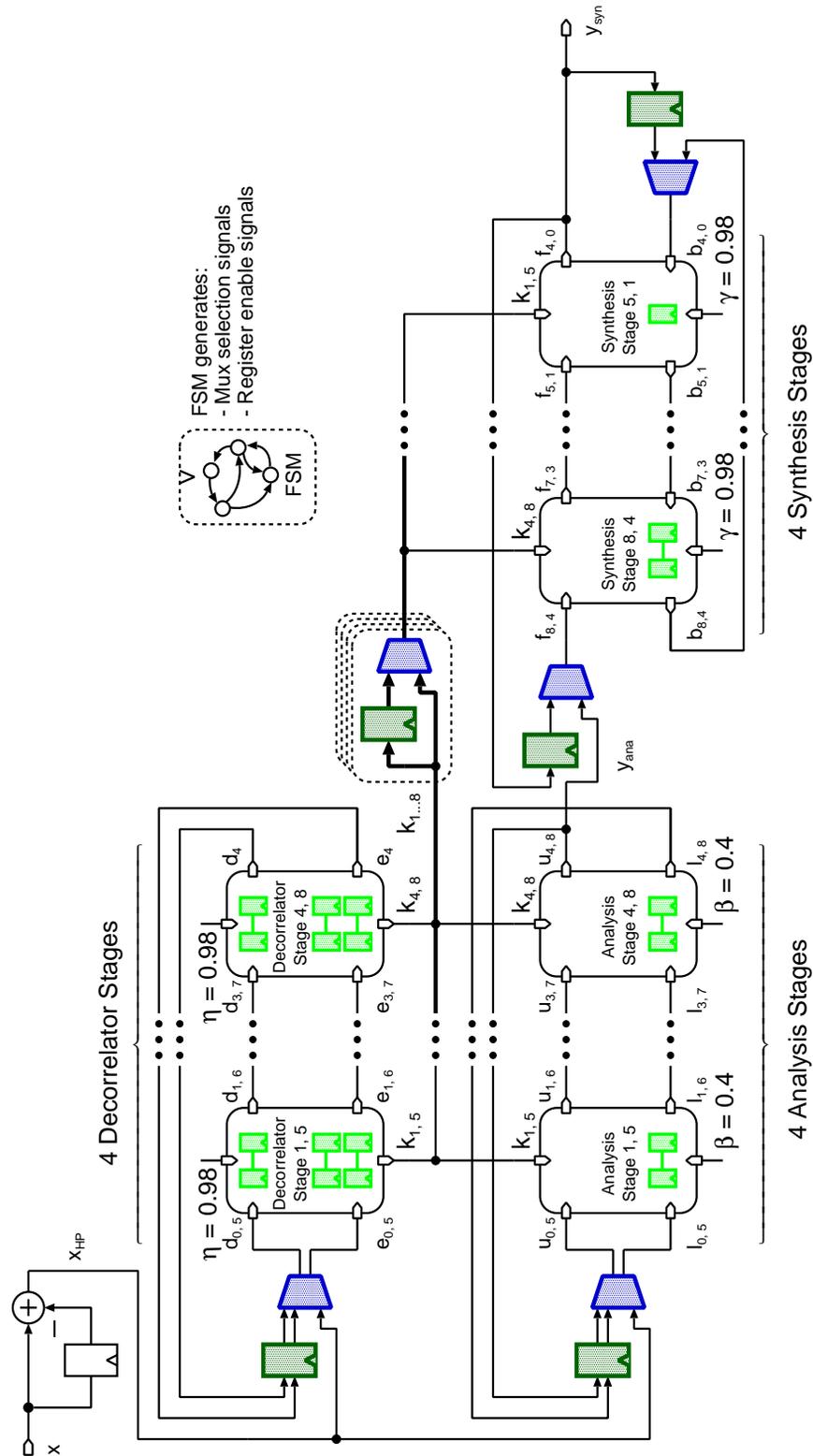


Figure 3.2: Architecture ID2 of the speech enhancement algorithm.

Table 3.1: Overview of the different architectures. Architectures ID1 to ID41 have been implemented; *ID111* has not been realized.

	Isom.	ID2	ID4	ID8	ID41	<i>ID111</i>
$f_{clk}$ [kHz]	22	44	88	176	902	<i>2442</i>
Control required	no	yes	yes	yes	yes	<i>yes</i>
Number of:						
stages (each type)	8	4	2	1	-	-
add. 16 bit	1 · 8	1 · 4	1 · 2	1 · 1	0	<i>0</i>
add. 32 bit	8 · 8	8 · 4	8 · 2	8 · 1	2	<i>0</i>
add. 44 bit	1 · 8	1 · 4	1 · 2	1 · 1	0	<i>1</i>
mult. 32 bit	10 · 8	10 · 4	10 · 2	10 · 1	2	<i>0</i>
mult. 28 bit	2 · 8	2 · 4	2 · 2	2 · 1	0	<i>0</i>
const. mult. 44 bit	1 · 8	1 · 4	1 · 2	1 · 1	1	<i>0</i>
mult. 44 bit	0	0	0	0	0	<i>1</i>
barrel shifters	1 · 8	1 · 4	1 · 2	1 · 1	1	<i>1</i>

ID4: The working frequency is doubled again and the hardware effort is halved, except for a small overhead.

ID8: It represents the highest degree of natural iterative decomposition. This means that each block – analysis, synthesis, and decorrelator – consists of one stage, through which data are fed eight times. The overhead is still small.

ID41: This architecture goes further by even time-multiplexing the remaining stages of ID8. Only three multipliers, two adders and one barrel shifter are required to perform all necessary operations. As a consequence, it takes 41 clock cycles to compute one output sample of the spectral sharpening filter. It is the smallest of all implementations and is the second one which has been integrated on silicon.

ID111: This architecture makes use of one adder, one multiplier, and one barrel shifter. They are scheduled like the arithmetic units in a DSP. However, this architecture has not been implemented.

It could be observed that the resulting hardware overhead in ID2, ID4 and ID8 in terms of chip area is very small. It mainly consists of additional registers, multiplexers and a small finite state machine, basically a counter. The sequence ID1, ID2, ID4 and ID8 follows a natural development which is given by the lattice filter structure and the filter order (eight in the considered case). Thereby, each step of iterative decomposition halves the number of instantiated filter stages while doubling the required clock frequency.

As mentioned above, Fig. 3.2 represents architecture ID2 with the required registers per filter stage indicated in light green. It can be seen that in nearly each stage, the register has been doubled, which builds a short shift register. Architectures ID4 and ID8 can be constructed accordingly, and they have not been illustrated. Design ID41 represents a highly decomposed architecture. Even a higher order of resource sharing would be possible by merging the remaining arithmetic units into a single (general-purpose) multiplier, an adder and a barrel shifter giving rise to design ID111, which is shown in the last column in Tab. 3.1 (in italics). This approach has not been followed any longer because of its high clock frequency and more complicated control part. In ID41, one multiplier (44 bit) performs all multiplications by the constants ( $\beta$ ,  $\gamma$  and  $\eta$ ), while the other two (32 bit) are scheduled to perform all remaining multiplications. The additions are 32 bit wide. A summary of the main design characteristics of the implemented architectures is given in Tab. 3.1. Note that no reasonable design could be devised between ID8 and ID41. Additional details of word widths can be found in [53]. In all arithmetic units, the common two's complement number format has been employed.

At first glance, it seems obvious that the isomorphic architecture (ID1, see Sec. 3.1) represents the most power-efficient hardware organization because it runs with the lowest clock frequency and no control at all is needed. As will be shown later, this is not true.

### 3.2.2 Synthesis and Power-Estimation Flow

In order to compare and study the effect of iterative decomposition on the expected power consumption, the above presented designs have been implemented using VHDL. ModelSim by Mentor Graphics Corp. has been employed as functional verification tool. The designs have

been synthesized with Synopsys Design Compiler targeting a  $0.25\ \mu\text{m}$  ( $V_{DD} = 2.5\ \text{V}$ ) standard CMOS cell library by Virtual Silicon and a fabrication process by UMC. This resulted in Verilog netlists and standard delay format (SDF) files. The latter contains the cell delays and additionally an estimation of interconnection delays. Really accurate interconnect delays are not available yet because placement and routing has not been performed in this early step of chip development. However, this netlist, together with the SDF file, have been used for a post-synthesis gate-level simulation in ModelSim, using real-world stimuli.

This design and power-estimation flow is graphically presented in Fig. 3.3. The gate-level simulation generates a “switching activity interchange format” (SAIF) back-annotation file. This file mainly contains the activity factors  $\alpha_k$  for the calculation of the switching power according to Eq. 2.28. Node capacitances  $C_{L,k}$  are estimated by Synopsys Design Compiler. The SAIF forward-annotation file is cell-library dependent; it contains information about cells with state-dependent and path-dependent power models [48]. If desired, a re-compilation of the design that takes account of the collected toggle activities can be started to further optimize the design.

## 3.3 Results

### 3.3.1 Synthesis and Simulation Results

The resulting power and area figures based on the aforementioned post-synthesis simulations are presented in Fig. 3.4. The other results shown in this figure will be discussed in Sec. 3.3.2. For designs ID1 to ID8, the scaling of area with each step of decomposition can be seen. As the number of memory cells does not differ significantly after every decomposition step, and the additional control area overhead is small, the total area almost halves with each degree of decomposition. It is impossible, however, to extrapolate the area and power figures from ID4 to ID8 in order to get an estimation of ID41, because the modular structure of the algorithm and hence filter stages has to be broken up completely.

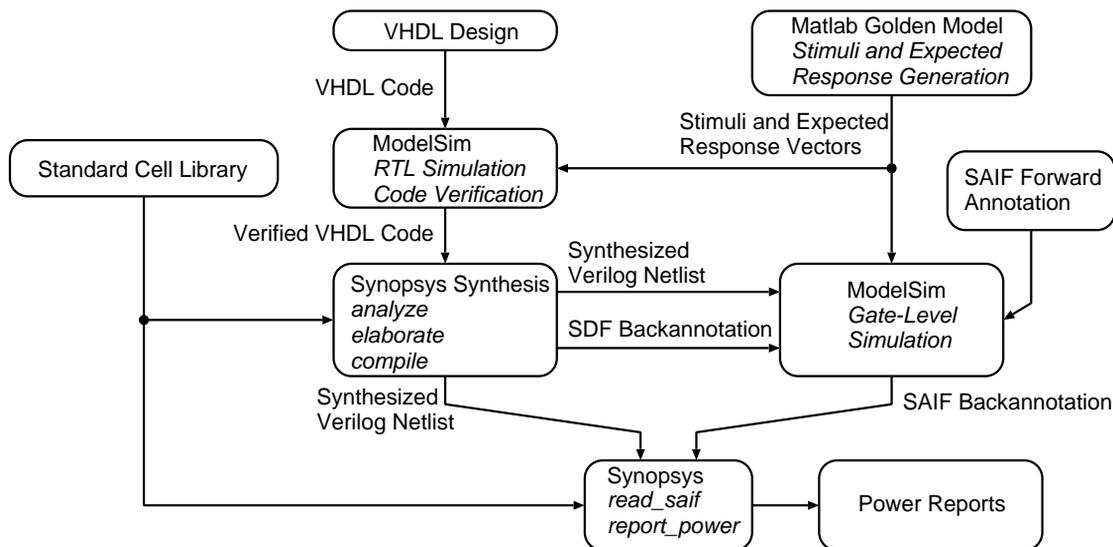


Figure 3.3: Power-estimation design flow using Synopsys Power Compiler (simplified).

The power dissipation shown in the same plot behaves totally unexpected: ID1, which is the only design that does not require any control nor multiplexing overhead, consumes more power than all other considered designs. From ID1 to ID8 the power consumption drops instead of the expected rise; only ID41 shows the expected power dissipation trend.

In order to understand this unexpected behavior of power consumption, the (total) toggle activities in four subsequent decorrelator stages of ID2 have been analyzed. The activities visualized in Fig. 3.5 are the averaged values of each input port of the corresponding arithmetic unit. In case of the constant-multipliers ( $\eta$ ), only the non-constant input has been considered for the calculation of the average toggle activity. The annotated activities have been read out of a switching activity interchange format (SAIF) file produced by a gate-level simulation according to Fig 3.3. The following observations hold accordingly for ID1:

- Arithmetic units placed behind at least one other unit, without any register in between, suffer from glitches. The longer the chain, the more glitches occur.

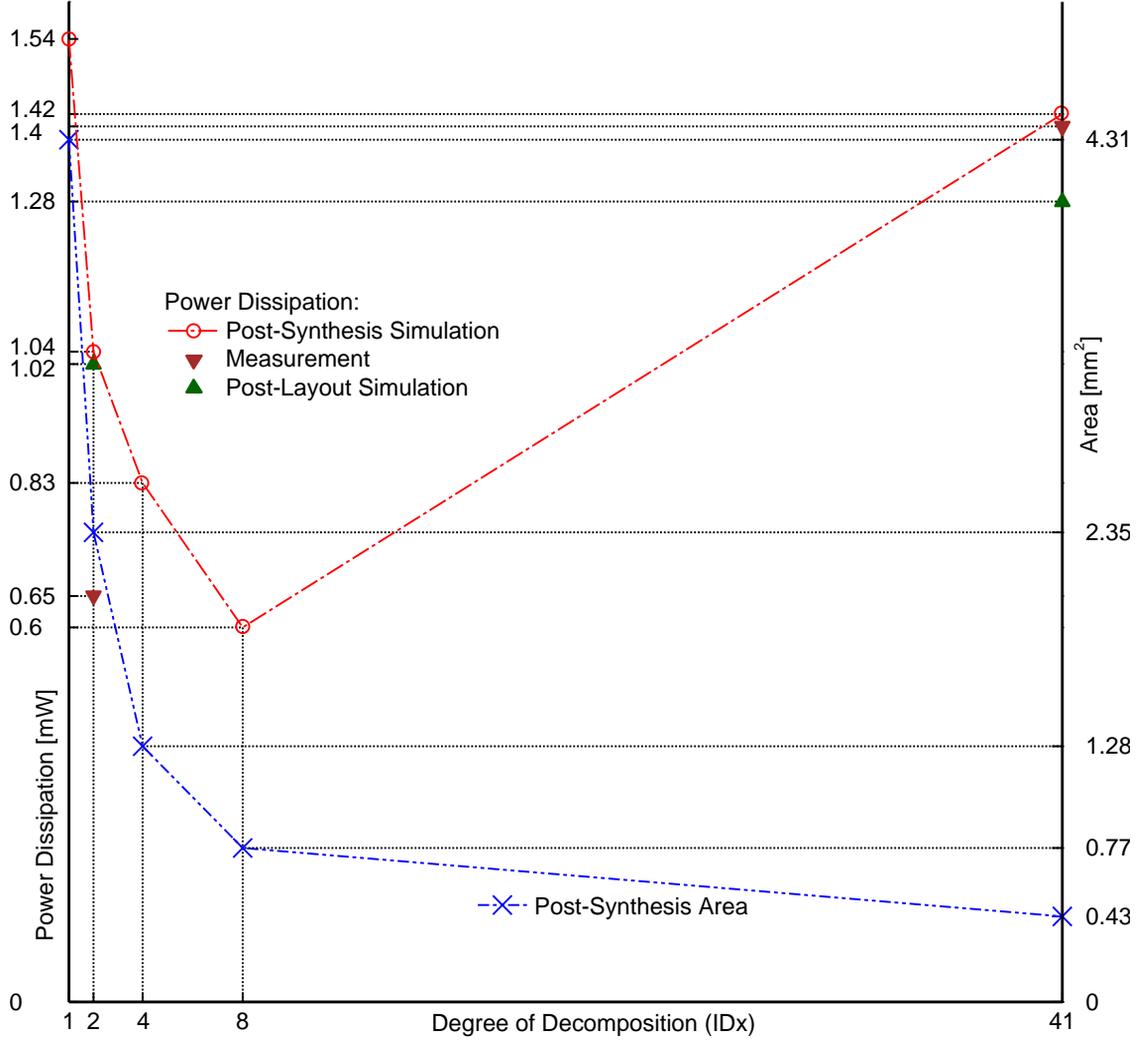


Figure 3.4: Power (left axis) and area (right axis) consumption of designs ID1, ID2, ID4, ID8 and ID41.

- Downstream stages suffer more from glitches than, for example, the very first stage.
- A unit placed directly after a register exhibits a low switching activity which is independent from the stage number. This can be seen, for example, in all light-blue marked multipliers. In fact, these multipliers show a toggle activity of  $\alpha_k \approx 0.5$  at their inputs.

In general, the switching activity can be expressed as a sum of two constituents [54]:

$$\alpha_k = \alpha_{k, f} + \alpha_{k, s} \quad (3.2)$$

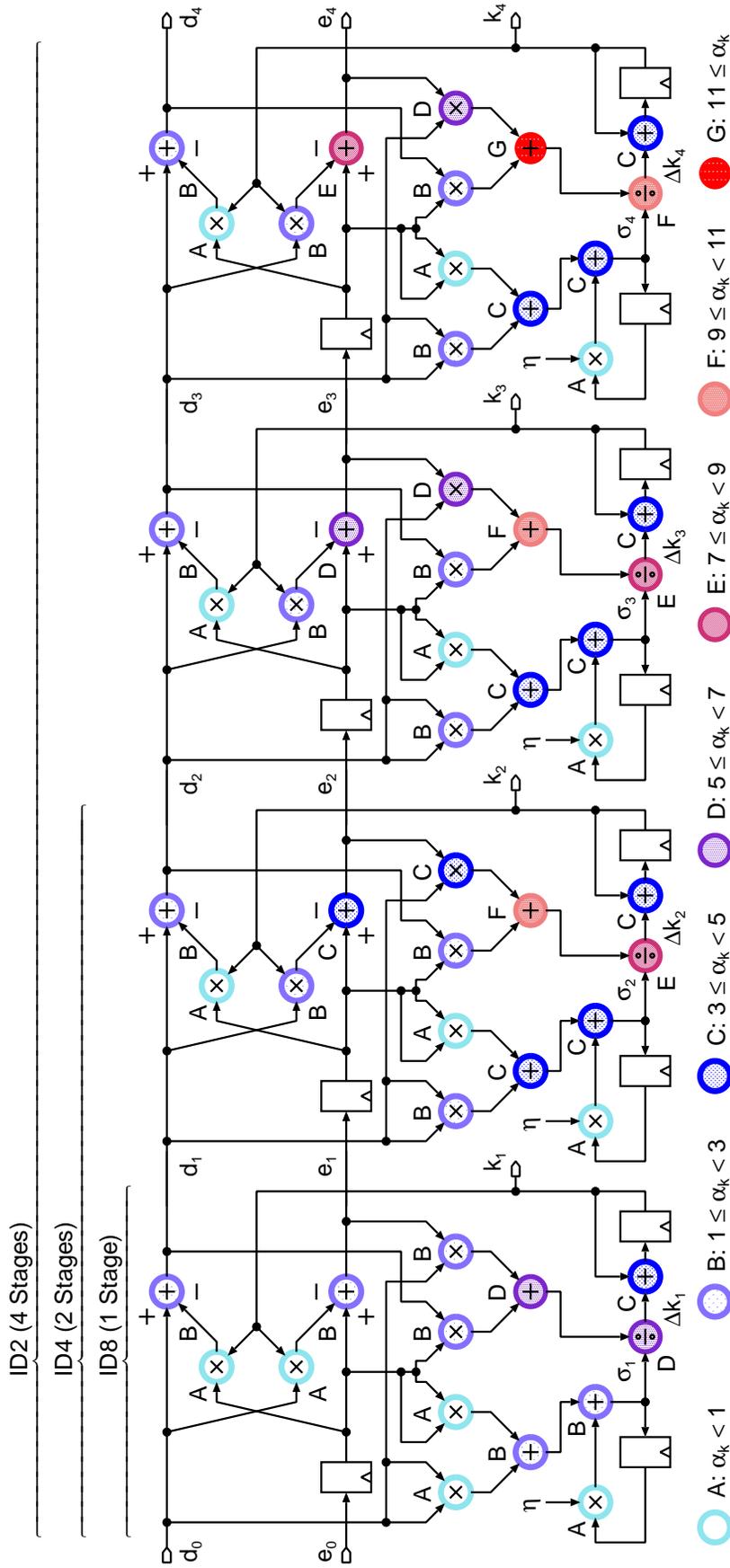


Figure 3.5: Activity of the decorrelator block in ID4. The approximate activity of ID4 can be read from this figure as well by observing only the first two stages. For ID8, only stage one is representative.

where  $\alpha_{k,f}$  denotes the functional toggle activity and  $\alpha_{k,s}$  is the spurious activity caused by glitches. In Fig. 3.5, the total activity  $\alpha_k$  has been annotated.

A simple back-of-the-envelope calculation for the output of a D-FF shows an average toggle activity  $\alpha_k = \alpha_{k,f} = 1/2$ , which is equivalent to a functional transition every second clock period. This toggle activity factor follows from the fact that the output of a D-FF can exhibit a maximum  $\alpha_k = 1$  (1 transition per clock cycle) and a minimal  $\alpha_k = 0$  (constant value) at the minimum. If this flip-flop is fed with random data, it immediately follows that  $\alpha_k = 1/2$ .

An additional gate-level simulation of the four decorrelator stages in ID2 has been performed. Thereby, cell propagation delays of each D-FF and combinational cell have been set to zero. Proceeding this way allows the simulation of almost ideal cells. No glitches are therefore created, because all combinational paths are forced to show the same delay of zero. This is justified by the fact that spurious activity is a result of different propagation delays in re-convergent paths [54]. The resulting pure functional activities  $\alpha_k = \alpha_{k,f}$ , ( $\alpha_{k,s} = 0$ ) is around 0.5 for all flip-flop outputs, regardless of the stage number. This confirms the above back-of-the-envelope calculation. In the optimum, the ratio of spurious activity to functional activity amounts to  $\alpha_{k,s}/\alpha_{k,f} = 0$  when no spurious activity is present. This is the case in the light-blue marked multipliers in Fig. 3.5, because their inputs emanate from D-type flip-flops. The maximum activity in ID2 adds up to  $\alpha_k = 12$ . This is the case for the red adder marked with a ‘G’ in Fig. 3.5, whose both inputs are at the tail end of long combinational paths. The resulting activity ratio amounts to  $\alpha_{k,s}/\alpha_{k,f} = (12 - 0.5)/0.5 = 23$ .

Since the toggle activity  $\alpha_k$  is normalized to the clock period [43], different levels of iterative decomposition can be compared with the help of Fig. 3.5. This means that the approximate activity of the decorrelator in ID4 corresponds to the activity of the first two stages in this figure, whereas the one in ID8 corresponds to only the first stage. These approximations hold if it can be assumed that the toggle activity is dominated by spurious activity  $\alpha_{k,s}$  and not by data to be processed, i.e., not by  $\alpha_{k,f}$ . It follows that, the more stages are put in sequence, the higher the average toggle activity becomes and hence, the more power is dissipated.

Table 3.2: Average spurious toggle activities per decorrelator stage in ID2 and deduction for the decorrelators in ID4 and ID8.

Decorr. Stage	$\langle \alpha_{k,s} \rangle$	Approximations		
		ID2 $\langle \alpha_{k,s} \rangle$	ID4 $\langle \alpha_{k,s} \rangle$	ID8 $\langle \alpha_{k,s} \rangle$
0	1.71	ID2 2.85	ID4 2.23	ID8 1.71
1	2.74			
2	3.20			
3	3.76			

Table 3.3: Normalized toggle activities and power consumptions.

Architecture	$\frac{\langle \alpha_{k,s,Decorrelator.IDx} \rangle}{\langle \alpha_{k,s,Decorrelator.ID2} \rangle}$	$P_{IDx}/P_{ID2}$
ID2	$2.85/2.85 = 1.00$	$1.04/1.04 = 1.00$
ID4	$2.23/2.85 = 0.78$	$0.83/1.04 = 0.80$
ID8	$1.71/2.85 = 0.60$	$0.60/1.04 = 0.58$

In the first column of Tab. 3.2, the average spurious activity  $\langle \alpha_{k,s} \rangle$  over all inputs of all computational units has been annotated for each decorrelator stage in ID2. Taking the average of these four stages yields the average spurious activity of ID2's decorrelator (2.85). Considering only the first two stages (stages 0 and 1) allows the deduction of the average in the one in ID4 (2.23), whereas the first stage (stage 0) corresponds to the one in ID8 (1.71). As shown in Tab. 3.3, in a next step, activities of decorrelators in ID4 and ID8 have been normalized to the one of ID2. Correspondingly, the simulated power consumptions of the designs according to Fig. 3.4 (magenta circles) have been normalized to the one of ID2. Note that these designs contain decorrelator, analysis, and synthesis. Comparing the ratios in the first and second column in Tab. 3.3 shows the almost linear

relationship between spurious activity in the decorrelator stages and power consumption in the complete designs.

In the sequence of designs ID1 to ID8, the clock frequency doubles for each additional degree of decomposition according to Eq. 3.1. Concurrently, the number of computational units halves according to Tab. 3.1. It follows that the ratio of the consumption of any two designs approximately equals the ratio of the corresponding switching activity:

$$\frac{P_{IDx}}{P_{IDy}} \approx \frac{\langle \alpha_{k,s, IDx} \rangle}{\langle \alpha_{k,s, IDy} \rangle}. \quad (3.3)$$

As shown above in Tab. 3.2 and especially in Tab. 3.3, this approximation is justified by the linear relationship between the toggle activity in the decorrelator and the power dissipation in designs ID2 to ID8. Thereby, it has been assumed that the dissipation of the control overhead is negligible. Moreover, since the lattice structure is common to all building blocks of the speech enhancement algorithm, it can be assumed that the overall behaviour of the algorithm is similar to the one of the decorrelator depicted in Fig. 3.5.

When decomposing the algorithm, registers are introduced to store the intermediate results of the stages. In Fig. 3.2 they have been placed in front of the multiplexers; these registers and multiplexers are colored dark green and blue, respectively. As a positive side effect, these additional registers filter glitches emanating from the arithmetic units inside the stages. Glitches produce unproductive switching activity leading to unnecessary power consumption. In synchronous designs they are not captured into registers, because they have died out before the next triggering clock edge.

Design ID41 does not profit from its high degree of decomposition. It suffers from a dominating overhead as a result of more complex control and many large multiplexers in front of the few arithmetic units and registers. Additionally, the high degree of resource sharing increases the switching activity in the few arithmetic units by destroying all signal correlations of subsequent inputs (Par. 7.2.4 in [40]).

In order to prevent glitches from propagating along arithmetic chains in designs with low decomposition, one might be tempted to add pipeline registers to block glitches. Yet, this is no option as this algorithm uses feedback for the recursive (IIR) filter (see Fig. 2.12).

### 3.3.2 Post-Layout Power Simulations and Power Measurements on Silicon

It was impossible to integrate designs ID1 and ID41, which represent the extreme poles of all considered designs, on the same die because of chip area limitations. As a compromise, instead of ID1, ID2 has been integrated. Therefore, placement and routing has been performed only for designs ID2 and ID41. The designs contain I<sup>2</sup>S interfaces at audio inputs and outputs. ID2 and ID41 have been integrated on the same silicon die shown in Fig. 3.6. Both designs have separate power nets and distinct power supply pads to allow individual power measurements. The separate power rings can unfortunately not be seen because of tiling.

Including power supply rings, design ID2 occupies 2.72 mm<sup>2</sup> on silicon, whereas ID41 requires 0.78 mm<sup>2</sup>. In the upper left corner of Fig. 3.6, separately powered multiplexers and demultiplexers are located because both designs share the same IO pads. Chip dimensions including seal ring are 2.435 mm × 2.435 mm.

Measurement results at  $V_{DD} = 2.5$  V can be found in Fig. 3.4 (brown triangles). ID2 dissipates 0.65 mW and ID41 1.4 mW. Note that these figures are not directly comparable with the simulation results in the same figure which are based on post-synthesis netlists (no clock tree, no back-annotation of parasitics etc.). For these two integrated designs, the flow presented in Fig. 3.3 has been enhanced: after place and route, parasitics have been back-annotated before the post-layout gate-level simulations. They show estimated power dissipations of 1.02 mW and 1.28 mW, respectively (see Fig. 3.4, green triangles).

## 3.4 Discussion of Low-Power Architectures

Based on post-synthesis gate-level simulations, it has been demonstrated that the design with the highest possible degree of decomposition without destroying the regularity of the algorithm, ID8, is the most power-efficient solution. Lower levels of iterative decomposition suffer from extensive glitch creation and propagation in their longer combinational paths. Higher levels suffer from energy-hungry control

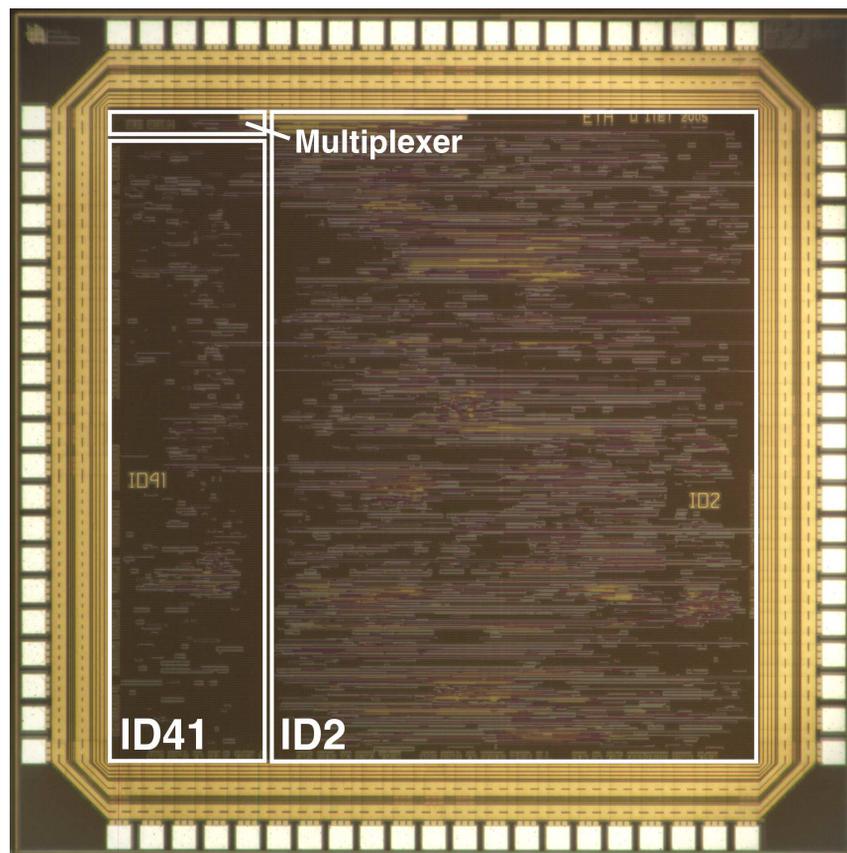


Figure 3.6: Chip photograph.

blocks and large multiplexers in front of arithmetic units that destroy signal correlations.

As can be seen in Fig. 3.4, the agreement of measurement and post-layout simulation in design ID41 is much better than in ID2. A possible explanation is an overestimation of glitches in gate-level simulations which do not give information about the quality of the glitches: Each glitch is assumed to exhibit a full swing from ground to  $V_{DD}$  and vice versa. In reality, only runt pulses or even no glitches might occur at all [43]. Experience has shown that glitches are not accurately enough simulated to get reliable power estimations neither with post-synthesis nor with post-layout gate-level simulations. Generally, their number and extent are overestimated, and hence power consumption is overestimated as well. Future gate-level simulators should incorporate more precise techniques to estimate glitching activity and related power dissipation. The work by Ruiz de Clavijo et al. [55] points into this direction.

Because of the high degree of decomposition in ID41, arithmetic units are mostly followed by registers to eliminate glitches. Power estimation is therefore accurate. In ID2, the chains of arithmetic units propagate glitches, leading to overestimation of glitch power. However, as predicted by the simulations, architecture ID2 is still more power efficient than ID41.

The findings of this work go beyond the publication [56]. There, the switching activity resulting from the two adders in the isomorphic architecture of a second order FIR filter<sup>2</sup> with the fully iteratively decomposed (time-multiplexed) architecture has been compared. It has been stated that in this small filter, time-multiplexing can increase the overall switching activity due to destruction of signal correlations. This certainly also holds for architecture ID41, as confirmed in Sec. 3.3.1. No statement about the impact of glitch propagation in longer combinational paths in less decomposed architectures has been made, since the author of [56] studied only a short FIR filter. Additionally, the chain of adders has been registered (pipeline registers). The thereby introduced overhead regarding power consumption has not been considered. As already stated in Sec. 3.3.1, inserting (pipeline) registers between lattice stages was no option in this work.

---

<sup>2</sup>An isomorphic architecture of an  $N$ -th order FIR filter requires  $N$  adders,  $N + 1$  multipliers, and  $N$  registers.



# Chapter 4

## Impact of Number Format

### 4.1 Introduction to Number Formats

#### 4.1.1 Signed Number Representations

##### 4.1.1.1 Two's Complement Format

Usually, arithmetic operations in digital signal processing use the common two's complement (TC) numbering format [56], [57]. This fact is even reflected in VHDL, where this format is used to represent signed numbers [58]. A signed integer number  $x_B$  in the range  $-2^{B-1} \leq x_B \leq 2^{B-1} - 1$  is represented in this format by  $B$  bit [27]:

$$x_B = -b_{B-1}2^{B-1} + \sum_{i=0}^{B-2} b_i 2^i \quad (4.1)$$

where the  $b_i$ 's represent the *binary digits* (bits) that are either '0' or '1'. The digit  $b_{B-1}$  has a special meaning: it is the sign bit. If  $b_{B-1} = 1$ , we have a negative value, hence  $-2^{B-1} \leq x_B < 0$  and if  $b_{B-1} = 0$ , we have a positive number and hence  $0 \leq x_B \leq 2^{B-1} - 1$ . In both cases

Table 4.1: The set of signed integer numbers representable with 4 bit in two's complement (TC) format.

$-2^3$	$2^2$	$2^1$	$2^0$	decimal
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	1	1	1	-1
1	1	1	0	-2
1	1	0	1	-3
1	1	0	0	-4
1	0	1	1	-5
1	0	1	0	-6
1	0	0	1	-7
1	0	0	0	-8

the number  $x_B$  can be represented as:

$$x_B = \underbrace{b_{B-1}}_{\substack{\text{sign} \\ \text{bit}}} b_{B-2} b_{B-3} \dots b_1 b_0 \quad (4.2)$$

The sign bit  $b_{B-1}$  is equal to the most significant bit (MSB) and the bit  $b_0$  is equal to the least significant bit (LSB). In Tab. 4.1, the signed integer numbers representable with  $B = 4$  bit in two's complement format are shown. Their numerical range is asymmetric: The most negative number has no (positive) additive inverse.

In order to describe fractional or fixed-point values, the representation in Eq. 4.1 is multiplied with  $2^{-C}$ ,  $0 \leq C \leq B - 1$ . This scaling is equivalent to place a binary point ( $\cdot$ ) at the  $C^{\text{th}}$  position starting from right in Eq. 4.2. For instance, placing the point at the right of digit  $b_2$

in a 4 bit wide number gives rise to the following representation:

$$x_B = \underbrace{b_3}_{\substack{\text{sign} \\ \text{bit}}} \underbrace{b_2}_{\substack{\text{integer} \\ \text{bit(s)}}} \cdot \underbrace{b_1 b_0}_{\substack{\text{fractional} \\ \text{bit(s)}}}, \quad B = 4 \text{ and } C = 2 \quad (4.3)$$

The bits left of the binary point but right of the sign bit are called integer bits, the ones right of the point are referred to as fractional bits. The decimal range of such numbers amounts to  $-2^{B-C-1} \leq x_B \leq 2^{B-C-1} - 2^{-C}$  with a smallest difference of  $2^{-C}$  between two consecutive numbers. In the example of Eq. 4.3, this is equivalent to a range of  $-2 \leq x_B \leq 1\frac{3}{4}$  and a minimum difference of  $1/4$ . If  $C = 0$ , i.e., if the binary point is placed immediately to the right of the LSB (bit  $b_0$ ), the pure integer representation results as given in Eqs. 4.1 and 4.2.

#### 4.1.1.2 Sign-Magnitude Format

An alternative to the two's complement number format is the sign-magnitude format. A signed integer number  $x_B$ ,  $-2^{B-1} < x_B < 2^{B-1}$  in this format can be represented as [27]:

$$x_B = (-1)^{b_{B-1}} \cdot \sum_{i=0}^{B-2} b_i 2^i \quad (4.4)$$

Again, the  $b_i$ 's are the binary digits ('0' or '1'), the most significant bit  $b_{B-1}$  is the sign bit and  $B$  is the bit width of the number. Similar to Eq. 4.2, the number  $x_B$  can be written as

$$x_B = \underbrace{b_{B-1}}_{\substack{\text{sign} \\ \text{bit}}} \underbrace{b_{B-2} b_{B-3} \dots b_1 b_0}_{\text{magnitude}} \quad (4.5)$$

Note that the bits  $b_{B-2} b_{B-3} \dots b_1 b_0$  represent the magnitude of  $x_B$  in binary representation. Table 4.2 shows the signed integer numbers representable with  $B = 4$  bit in sign-magnitude format. One should take notice of the presence of two representations of '0' and the symmetric range. As in the case of TC numbers, SM numbers according to Eq. 4.4 can be scaled to derive any sign-magnitude fixed-point format.

Table 4.2: The set of signed integer numbers representable with 4 bit in sign-magnitude (SM) format.

$\pm$	$2^2$	$2^1$	$2^0$	decimal
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	0	0	0	(-)0
1	0	0	1	-1
1	0	1	0	-2
1	0	1	1	-3
1	1	0	0	-4
1	1	0	1	-5
1	1	1	0	-6
1	1	1	1	-7

### 4.1.1.3 Comparison

**Range:** Comparing Eqs. 4.1 with 4.4, and Tabs. 4.1 with 4.2 reveals that for positive numbers, i.e.,  $b_{B-1} = 0$ , the binary representations in Eqs. 4.2 and 4.5 are identical. Therefore, the largest positive number representable with a given number of bits is the same in both number formats. Because of the presence of a “positive” and a “negative” zero in the sign-magnitude format, these numbers exhibit a symmetric range, e.g.,  $-(2^{B-1} - 1) \leq x_B \leq 2^{B-1} - 1$  compared to  $-2^{B-1} \leq x_B \leq 2^{B-1} - 1$  for TC numbers. Although it allows for a symmetric range, this second zero in sign-magnitude is one of the major drawbacks of this number format, since it requires special treatment in the algorithms. In two’s complement, the position of this number is taken by the most negative value  $-2^{B-1}$ , which has no corresponding positive number in TC and yields a asymmetric range. This number

cannot be represented in SM. Moreover, this asymmetry needs special care when converting numbers from SM into TC representation and can cause overflow during negation. In two's complement, negation is performed by flipping all bits and adding '1'. In sign-magnitude numbers, negation is easily fulfilled by inverting the sign bit.

**Arithmetics:** In two's complement format, addition and subtraction hardware are almost the same, because the latter can be transformed into adding the negated subtrahend, i.e.,  $c = a - b = a + (-b)$ . The negation is performed by building the bitwise complement  $\bar{b}$ , also referred to as the one's complement, of the subtrahend  $b$  and adding 1:  $-b = \bar{b} + 1$ . Creating the one's complement  $\bar{b}$  is equal to flipping all bits of  $b$ .

As shown in Fig. 4.1, addition and subtraction operations can be even merged into the same hardware unit. Setting the "Operation" input, results in the subtraction  $c = a + \bar{b} + 1 = a - b$ , since then, the input  $b$  is complemented prior to addition (pre-complementation) by means of the green XOR gates. The addition of '1' is performed by setting the carry-in (CI) of the adder [59]. As shown later, this proceeding of negation also serves as basis of the format conversion between SM and TC, and vice versa. Note, a pure adder unit would do without the XOR gates and either a zero would be fed into the carry in (CI) or, alternatively, the rightmost full adder (FA) (blue in Fig. 4.1) could be replaced by a half adder (HA).

No special action is to be taken for negative number processing. Figure 4.2a shows the situation of adding 5 plus  $-3$  represented with

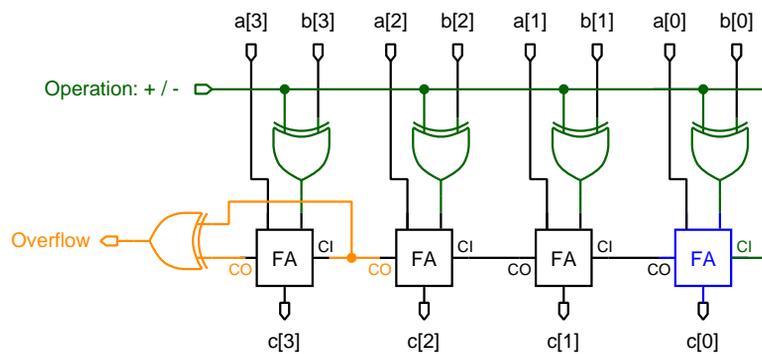


Figure 4.1: Addition and subtraction in two's complement.



Figure 4.2: Overflow detection in two's complement format: a) no overflow detected, and b) overflow detected.

4 bit in two's complement format. The addition yields the correct result 2 (green). The carry bit (orange) into the leftmost position (blue) can be ignored. In Fig. 4.2b, adding  $-5$  plus  $-7$  would yield  $-12$ , which cannot be represented with 4 bit. In this figure, the incorrect 4 bit wide result 4 is marked red. This condition, also referred to as *arithmetic overflow*, can be detected by comparing the two leftmost carry bits (orange). If they are equal, the result is correct, as in Fig. 4.2a; if they are different, as in Fig. 4.2b, the result is incorrect because an overflow has occurred. In other words, arithmetic overflow occurs during addition when the signs of the two operands are equal and the one of the result is different. The orange XOR gate in Fig. 4.1 creates the overflow status output.

Addition and subtraction in sign-magnitude format require more effort. Usually, a pre- and a post-complementation before and after the adder are needed. Additionally, a control unit creates the control signals of the two complementation units and the sign bit of the output [59].

Multiplication and division are more complicated when negative numbers are involved. However, the latter operation is usually avoided in ASICs by using approximations that employ a barrel shifter. Booth encoding is a method to perform signed multiplication [57], [59].

**Word Width Adaption:** As shown in Fig. 4.3a, expanding a number to a wider width is straightforward in two's complement: the integer part is expanded on the left by copying the sign bit to the added most significant bits. This process is called sign extension,

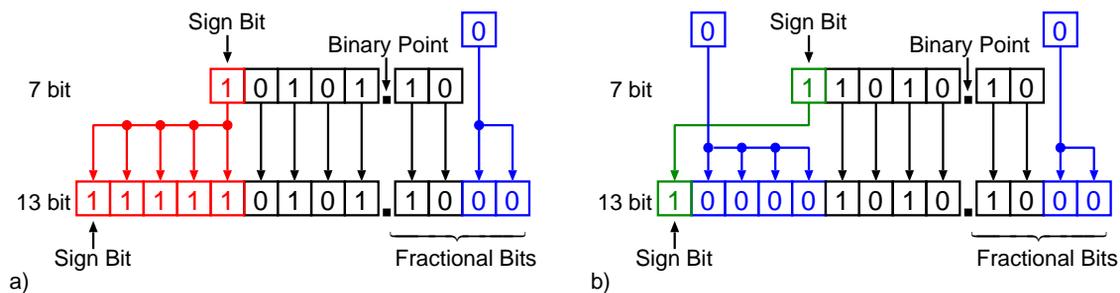


Figure 4.3: Adaptation of bit widths: a) Two's complement, and b) Sign-magnitude.

it has been marked red in Fig. 4.3a. The fractional part is padded with zeroes into the new least significant bits (blue in the figure). In sign-magnitude representation, see Fig. 4.3b, the sign bit is copied to the new most significant position in the wider number (green), while all appended bits are filled with zeroes (again blue). In both number formats, the binary point keeps its position.

**Truncation** Truncating a binary number means to reduce the number of binary digits (bits), usually right of the binary point, by discarding some less significant bits. In TC, this operation is equivalent to rounding towards minus infinity, whereas in SM, this operations corresponds to rounding towards zero. In Fig. 4.4a and b, the truncation of 2 bit (red) of a positive number in TC and SM format is shown. Apparently, the numeric effect on both truncated numbers is the same. The situation gets different when negative numbers are truncated: Fig. 4.4c illustrates the result of rounding towards minus infinity when a negative number in TC format is truncated. Eventually, in Fig. 4.4d, the truncation of the same negative number in SM is depicted, which is equivalent to rounding versus zero.

In both number formats, an error will be introduced by truncation. In case of a FIR filter that has been implemented by means of a MAC unit, this error tends to cancel out during accumulation if the truncation is performed after the multiplication in SM format. In TC, the error is propagated and accumulated, which leads to higher overall errors [60].

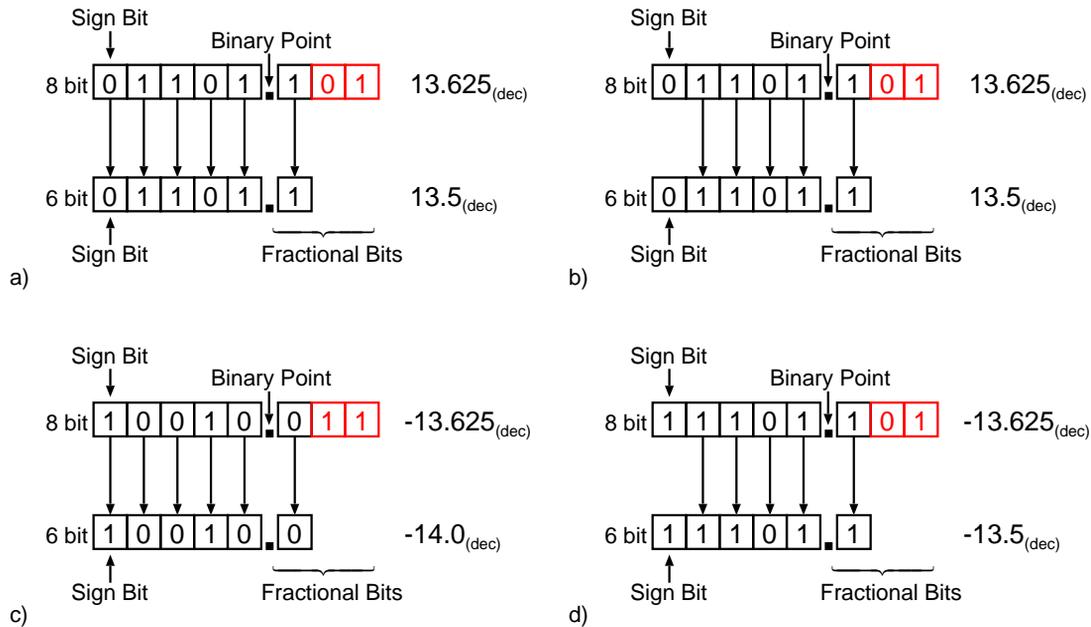


Figure 4.4: Truncation: a) Two's complement positive number, b) Sign-magnitude positive number, c) Two's complement negative number, and d) Sign-magnitude negative number.

**Format Conversion:** As mentioned earlier, negation can be employed to perform a format conversion of numbers between the two number formats:

- Positive number: No action is required, since the binary representations of positive numbers in SM and TC format are identical, compare Tab. 4.1 to Tab. 4.2.
- Negative number: The two representations differ apart from the sign bit (MSB), the conversion is performed as follows:
  - TC  $\Rightarrow$  SM: Extract the magnitude of the TC number. Since this number is negative, this can be performed by inversion. Set the sign bit.
  - SM  $\Rightarrow$  TC: Interpret the magnitude of the SM number as a TC number and invert it. Set the sign bit.

Since both conversion directions involve exactly the same operations, they can be executed by the same hardware unit, see Fig. 4.5. In case of negative numbers, the XOR gates together with the row of half adders (HA) perform the inversion, which is equal to complementing and incrementing. An overflow can only occur, when the most negative

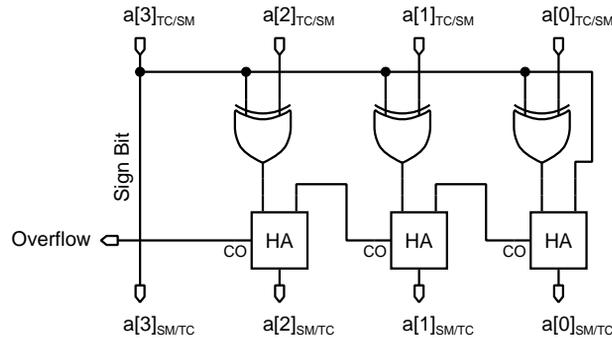


Figure 4.5: Format conversion between two's complement and sign-magnitude, or vice versa.

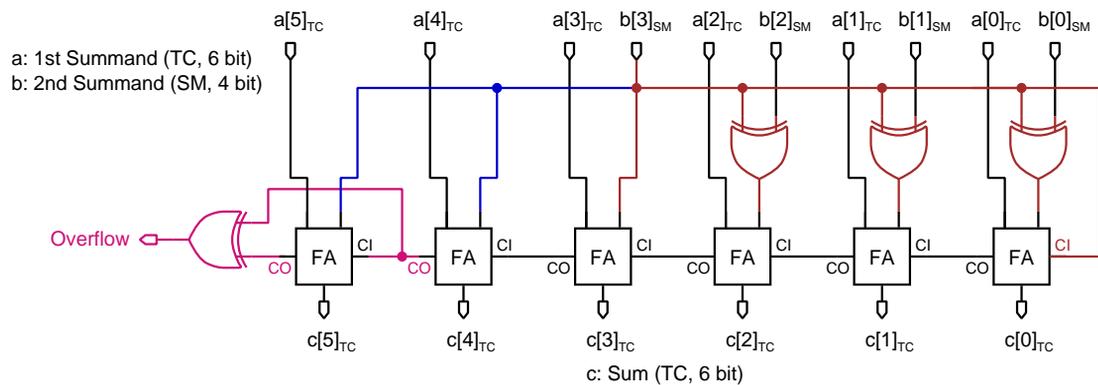


Figure 4.6: Number format conversion combined with subsequent sign extension and TC addition.

number in TC is to be converted into SM. In the 4bit hardware example in Fig. 4.5, this is the case with  $-8$ , which is the number that cannot be represented in sign-magnitude format using only 4 bit.

As shown later, this conversion – possibly coming after a sign-magnitude multiplication – is sometimes followed by a TC addition of arbitrary wider width. This necessitates a sign extension at one adder input. A rather naive approach to build such a system would be to make use of a distinct conversion and subsequent addition. In the conversion unit in Fig. 4.5, the row of HAs simply increment by 1, but what is really needed, is an increment plus a full addition. As shown in Fig. 4.6, replacing the half adders by full adders (FA) and feeding the sign bit ( $b[3]_{SM}$ ) of the SM number into the rightmost

carry-in (CI) of the adder fulfills both tasks. Expressed in other words, the increment by 1 is done virtually for free in the adder by making use of its carry-in input. The same idea has already been followed in Fig. 4.1. In the example in Fig. 4.6, a 6bit value ( $a_{TC}$ ) in TC format is added to a 4bit value ( $b_{SM}$ ) in SM giving a 6 bit wide sum ( $c_{TC}$ ) in TC. The required sign extension has been marked blue, whereas the format conversion has been marked brown and the overflow detection is pink.

**Toggle Activity:** Sign-magnitude operations seem to reduce overall switching activity. This can be seen for a small number changing from positive to negative, or vice versa. The example in Fig. 4.7 shows the total of flipping bits in TC and SM format during the transition from  $+3$  to  $-3$  represented with different word widths, in our case 4 bit and 16 bit. In the figure, the flipping bits have been marked red and the sign bits, which toggle regardless of the number format and word width, have been marked blue. When the number is represented with only 4 bit, the number of flipping bits amounts to 3 and 1 for TC and SM, respectively. In SM, the only toggling bit is the sign bit (blue). The situation completely changes when the word width amounts to 16 bit: In two's complement, the number of flipping bits dramatically increases to 15 bit, whereas in sign-magnitude, this number remains constant (1 single bit). This large difference in toggling bits is the result of sign extension in two's complement format. This contrasts to the sign-magnitude format where only constant zeroes are appended to extend a minimum-width representation.

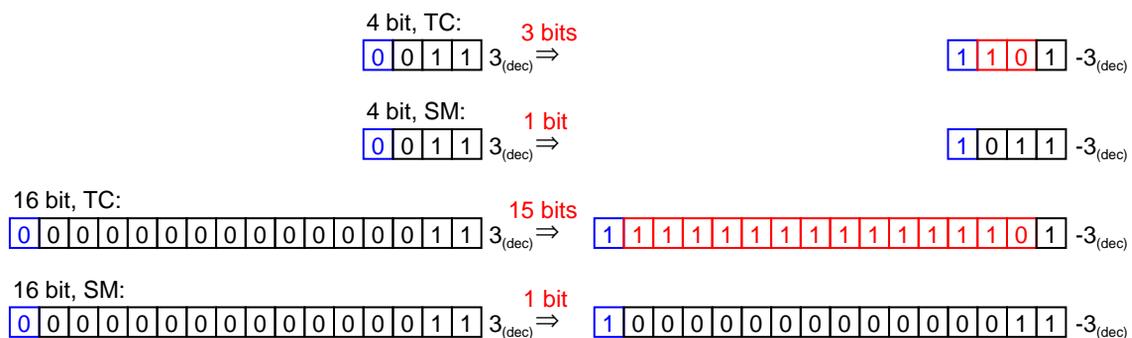


Figure 4.7: Flipping bits in different number formats and word widths.

### 4.1.2 Low-Power Arithmetic

The previous paragraph in the last section (Sec. 4.1.1.3) reveals a problem of TC numbers regarding power consumption. Consider a 16 bit  $\times$  16 bit TC multiplier yielding a 32 bit wide product. Small numbers as in the example in Fig. 4.7 are applied to this multiplier. If one – or even both – factors change their sign, many bits toggle at the input of the multiplier. This creates significant switching activity and hence power consumption inside the combinational logic of the multiplier.

Based on this observation, several sources indicate that from a power-saving point of view, it is advantageous to perform multiplications in the SM format [60], [40]. On the other hand, additions and subtractions should be performed using the TC format, because a simple pre-complementation, as shown in Fig. 4.1, is sufficient to process positive and/or negative numbers. Multiplication (and division) allow the determination of the sign of the result solely based on the signs of the two factors regardless of their magnitudes. This “decoupling” of sign and magnitude calculation additionally motivates the use of the sign-magnitude format for multiplications. Moreover, magnitudes are always unsigned and unsigned multiplications are more easily implemented in hardware than a signed one. This contrasts with addition and subtraction, where both – signs *and* magnitudes of the inputs – determine the sign of the output. The related TC hardware is intrinsically able to process signed numbers.

Moreover, since multiplication is much more complicated than addition and subtraction – basically, a multiplication involves numerous additions – it seems even more beneficial to completely switch the number format to SM representation and avoid a possibly required format conversion [60].

Probably the most widely used operation in digital signal processing is a combination of multiplication and addition, the so-called multiply-accumulate (MAC), which is the heart of virtually all filter operations. Examples of such MAC units can be found in Fig. 4.8, where they are used in a FIR filter. For minimal power consumption in such a MAC unit, there are three options to be investigated:

- two’s complement format
- sign-magnitude format
- a combination of the two

The third one, also known as *hybrid (HY) format*, tries to combine the advantages and mitigate the disadvantages of both formats. It performs multiplication in sign-magnitude format and addition in two's complement format. A format conversion is required between the operations, and sometimes prior to multiplication if arguments are only available in the two's complement format.

## 4.2 Impact of Number Format in FIR Filters

### 4.2.1 MAC Unit

In order to evaluate the potential of power savings depending on the number format, an iteratively decomposed 99th-order FIR filter containing a single MAC unit has been implemented and simulated for each format. The sampling frequency of 22.05 kHz requires a clock frequency of 2.205 MHz.

Although these number formats have already been compared in the thesis by J. Wassner [60], they have been re-evaluated in a single-MAC unit FIR filter. In contrast to Wassner's work, the results are based on more accurate post-layout simulations, whereas his data are based on post-synthesis simulations. Additionally, the consumption inside the MAC unit is broken down in more detail.

From a simple power saving point of view, it would make sense to perform this filtering in the frequency domain by means of a fast Fourier transform (FFT) and a multiplication with the frequency response followed by the inverse FFT (iFFT). This would reduce the amount of operations and hence, most likely, the dissipated power. Unfortunately, this approach is inappropriate for filtering in hearing aids because the human ear is irritated by the resulting latency of the processed signal in presence of the unprocessed sound that bypasses the hearing instrument. In Appendix A, the number of real-valued multiplications in a time domain implementation is compared with an equivalent frequency domain implementation of a FIR filter.

The MAC units for the three number formats are depicted in Fig. 4.8. It shows that in the sign-magnitude implementation (b) the processing of the sign bits is completely decoupled from the calculation

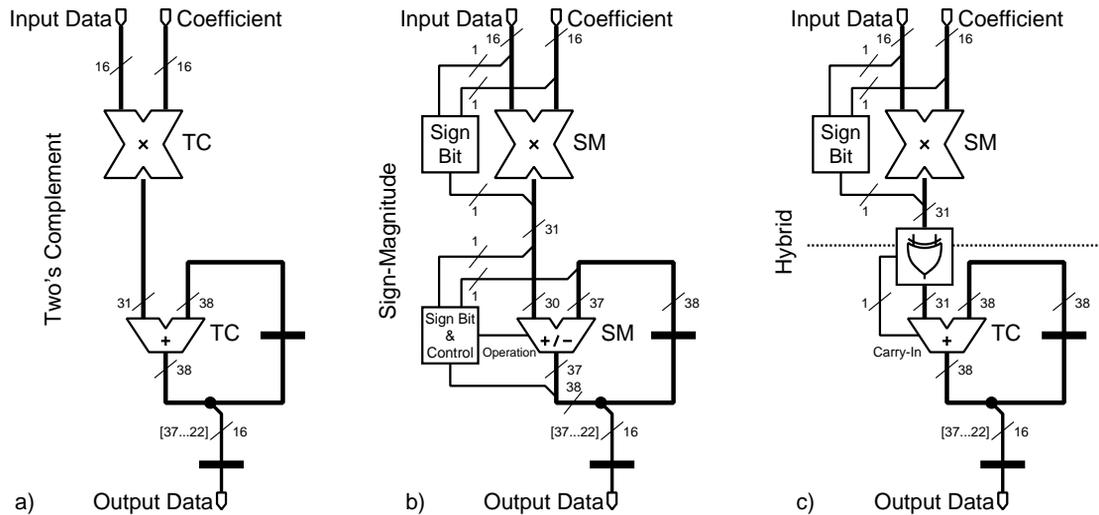


Figure 4.8: The implemented MAC units: a) Two's complement (TC), b) Sign-magnitude (SM), and c) Hybrid (HY).

of the magnitude, and the multiplication and addition are unsigned operations. In the hybrid implementation (c) the same observation only holds for the multiplication. Immediately after the multiplication, the value is converted into two's complement format and passed to the adder. The principle of this format conversion with subsequent word width adaption is illustrated in Fig. 4.6.

## 4.2.2 Simulation Results

The filter has been synthesized using Synopsys Design Compiler, targeting a standard  $0.25\ \mu\text{m}$  CMOS process. The resulting netlists have been placed and routed employing Silicon Ensemble by Cadence. The multiplier architecture was not restricted: in all three cases, a Wallace-tree multiplier was chosen by the synthesis tool. A ripple carry adder has been hard-coded in VHDL.

Table 4.3 summarizes the simulated power consumptions of the three MAC units processing two types of signals, which are typical in a hearing aid application. The first one is a male speaking voice, while in the second one, a babel of voices can be heard. Filter coefficients and data to be processed are applied in the required format, in two's complement for the architecture in Fig. 4.8a, and sign-magnitude

format for the others in Fig. 4.8b and c. Accordingly, the filtered data is output in the format of the accumulator. The hierarchical block denoted as “Converter” in the table contains only a bank of XOR gates used to conditionally complement all bits of the magnitude. The “Accumulator” contains the accumulation register, the adder, and the clock-gated<sup>1</sup> output register.

### 4.2.3 Discussion

As assumed, the hybrid implementation profits from the advantages of both data formats: its power consumption is the lowest for the two considered signals. The straightforwardly implemented two’s complement MAC unit exhibits the highest power dissipation, whereas the sign-magnitude implementation lies in between.

Passing from two’s complement to sign-magnitude mainly reduces power consumption because of the simpler multiplication, which compensates the more complicated and hence more dissipative addition in the accumulator (plus 15% in case of the speaking voice signal). A total power reduction of 22% (speaking voice) and 39% (babel of voices) can be observed. In the TC format, for both signals, the consumption of the multiplier is higher than the one of the accumulation. This is completely different in the second format, where a lot of power (42% and 61%, respectively) is saved in the multiplication. In fact, this operation now takes less power than the accumulator.

Compared to SM, the HY format adds a negligible overhead given by the conversion (XOR gates). It can be seen that the dissipation in the accumulation is reduced (around 27% for both signals), whereas the consumption of the multiplication stays approximately constant since in both cases it is of sign-magnitude type. In total, the hybrid format saves around 10% over SM for both signals.

Comparing the two signals shows that all sub-blocks, and hence the MAC unit, consume less power when processing the babel of voices. This can be explained by the signal’s smaller amplitudes than in the pure speaking voice signal.

Although these results favor the hybrid (or sign-magnitude) representation in a single-MAC implementation, it must be stated that any

---

<sup>1</sup>See Chapter 5 for more details of this low-power technique.

Table 4.3: Post layout power simulation of the three MAC units.

Hierarchy	Switch <sup>a</sup> Power [mW]	Internal <sup>b</sup> Power [mW]	Leakage Power [μW]	Total Power [mW]
Two's Complement: Male Speaking Voice				
Multiplier	0.284	0.225	5.370	0.515
Accumulator	0.059	0.201	5.293	0.265
Total MAC	0.343	0.426	10.664	0.779
Sign-Magnitude: Male Speaking Voice				
Multiplier	0.171	0.121	4.772	0.297
Accumulator	0.083	0.215	5.996	0.305
Total MAC	0.254	0.337	10.768	0.602
Hybrid: Male Speaking Voice				
Multiplier	0.158	0.117	4.839	0.280
Converter (XOR)	0.017	0.022	0.132	0.039
Accumulator	0.051	0.166	5.208	0.221
Total MAC	0.226	0.305	10.179	0.541
Two's Complement: Babel of Voices				
Multiplier	0.259	0.210	5.467	0.475
Accumulator	0.056	0.199	5.287	0.260
Total MAC	0.315	0.409	10.754	0.735
Sign-Magnitude: Babel of Voices				
Multiplier	0.106	0.075	4.969	0.186
Accumulator	0.064	0.190	5.934	0.260
Total MAC	0.170	0.265	10.903	0.446
Hybrid: Babel of Voices				
Multiplier	0.096	0.071	5.035	0.171
Converter (XOR)	0.013	0.018	0.101	0.030
Accumulator	0.041	0.146	5.202	0.192
Total MAC	0.149	0.234	10.338	0.393

<sup>a</sup> Dissipated power by (dis-) charging the load capacitance of a cell.<sup>b</sup> Dissipated power inside a cell. Refer to Sec. 2.2.2 for details.

possibly required conversions at the input and output of this processing unit are not considered in this study. Since the most common data format in computing is two's complement, such a conversion at the input is probably required for the hybrid MAC unit, and additionally at the output for the sign-magnitude unit. As seen in the next section (Sec. 4.3), additional converters can have a detrimental impact on the overall power consumption of a processing unit.

### 4.3 Impact of Number Format in Lattice Filters

Inspired by the results presented in the preceding section (Sec. 4.2) and in Wassner's thesis [60], the influence of the number format in the context of a lattice filter has been studied. For this reason, the analysis filter stage presented in Fig. 2.11 has also been implemented in the three well-known data formats.

#### 4.3.1 Analysis Stage

Details of the hybrid implementation are given in Fig. 4.9. Incrementers (adding '1' in case of a negative number) are sometimes needed to complete the format conversion. Their locations are marked with a  $\boxplus$  symbol. The inputs of the two subtractors do without a decrement, instead it is sufficient to feed the sign bit to the carry input of the subtraction. These two inputs are marked with a  $\boxminus$  symbol in Fig. 4.9. Both stage outputs are in TC format because they emanate from subtractions, which perform their operation by an addition of the negated subtrahend. Consequently, the two data inputs  $u_{i-1}$  and  $l_{i-1}$  are in two's complement format, which enables lining up stages for higher order filters. Likewise, the coefficient inputs  $k_i$  are in two's complement format, because the coefficients emerge from a registered adder output, see Fig. 2.13 for details. The bitwise conditional complementation in case of negative numbers is indicated by a single XOR symbol in a box. For the conversion from two's complement to sign-magnitude, special care has been taken to prevent numeric overflows due to the incongruent ranges of the two number formats.

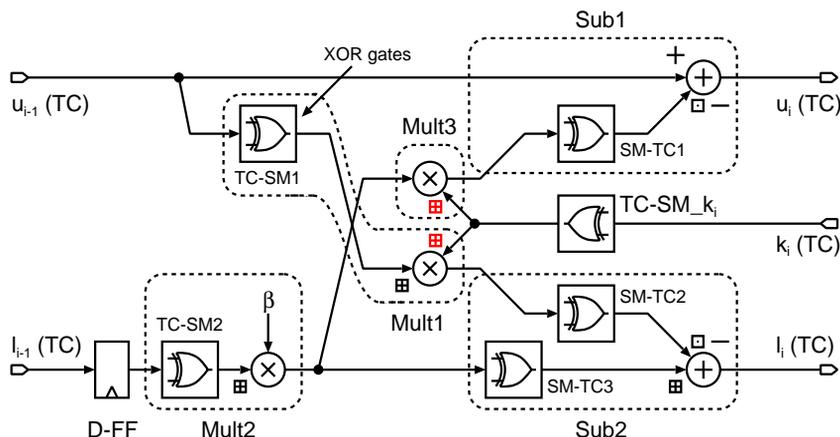


Figure 4.9: Lattice stage of analysis filter as a hybrid.

The two's complement lattice stage contains inputs and outputs in this standard format, which allows a simple line up of stages as shown in Fig. 2.11. Obviously, all inputs and outputs in the sign-magnitude implementation are in SM format.

### 4.3.2 Simulation Results

As for the MAC simulation in Sec. 4.2, VHDL has been used as hardware description language and Synopsys Design Compiler for synthesis, targeting a standard  $0.25\ \mu\text{m}$  CMOS process. However, no place and route has been performed: the results in Tab. 4.4 are solely based on post-synthesis gate-level simulations. Since a single analysis stage is relatively small and contains only a few flip-flops in the register preceding the multiplication with  $\beta$ , it can be assumed that the power consumption dissipated in the clock tree, which is typically added during place and route, is negligible. The simulations are based on the processing of a female voice with added noise.

The arithmetic units in Tab. 4.4 correspond to the dashed boxes in Fig. 4.9. Each arithmetic unit together with the converter(s) has been combined into one VHDL entity, whose power consumption is reported in the table. Since the coefficient  $k_i$  is used in two multipliers, namely *Mult1* and *Mult3*, the bitwise complementation has been shared between these two units. Though, two incrementers ( $\boxplus$ , red) have been coded in VHDL, and they have been placed at the input

Table 4.4: Post Synthesis power simulation and cell area of the three lattice stages.

Hierarchy	Switch <sup>a</sup> Power [ $\mu$ W]	Internal <sup>b</sup> Power [ $\mu$ W]	Leakage Power [ $\mu$ W]	Total Power [ $\mu$ W]	% Power	Area [ $\mu\text{m}^2$ ]
Two's Complement						
Mult1	5.2	6.5	4.4	16.1	35.0	30 452
Mult2	1.0	0.8	1.2	3.0	6.5	8 569
Mult3	7.0	8.0	4.5	19.5	42.3	30 421
Sub1	0.9	1.2	0.7	2.8	6.1	3 524
Sub2	0.9	1.4	0.7	3.0	6.5	3 540
D-FF	0.2	0.5	0.9	1.6	3.5	2 788
Total Stage	15.3	18.4	12.4	46.1	100.0	79 295
Hybrid						
Mult1	5.3	5.4	4.6	15.3	32.0	32 432
(TC-SM1)	(0.0)	(0.0)	(0.1)	(0.2)	-	(942)
Mult2	1.9	1.2	1.4	4.5	9.4	10 209
(TC-SM2)	(0.1)	(0.1)	(0.1)	(0.2)	-	(942)
Mult3	6.7	7.2	4.2	18.1	37.9	28 845
Sub1	1.2	2.2	0.8	4.2	8.8	5 908
(SM-TC1)	(0.5)	(0.8)	(0.3)	(1.5)	-	(2 606)
Sub2	1.1	1.9	1.0	4.0	8.4	8 474
(SM-TC2)	(0.1)	(0.1)	(0.1)	(0.3)	-	(1 283)
(SM-TC3)	(0.3)	(0.5)	(0.3)	(1.0)	-	(2 598)
D-FF	0.1	0.5	0.9	1.5	3.1	2 788
TC-SM <sub>k</sub> <sub>i</sub>	0.1	0.0	0.1	0.2	0.4	942
Total Stage	16.4	18.3	13.0	47.8	100.0	89 615
Sign-Magnitude						
Mult1	5.2	5.4	4.3	14.9	27.7	28 963
Mult2	1.4	0.7	1.2	3.3	6.1	7 999
Mult3	6.3	6.8	4.1	17.2	32.1	27 570
Sub1	2.9	4.8	1.4	9.1	16.9	11 928
Sub2	2.2	4.0	1.5	7.7	14.3	11 532
D-FF	0.2	0.4	0.9	1.6	3.0	2 788
Total Stage	18.2	22.1	13.5	53.8	100.0	90 795

<sup>a</sup> Dissipated power by (dis-) charging the load capacitance of a cell.<sup>b</sup> Dissipated power inside a cell. Refer to Sec. 2.2.2 for details.

of the actual multiplier inside the pertaining entity. For information purposes, the small dissipation of the conversion units are given in parentheses in Tab. 4.4.

### 4.3.3 Discussion

Contrary to expectations motivated by the findings in Sec. 4.2, the lattice filter does not profit from sign-magnitude or hybrid number formats. The hybrid dissipates 3.7% more than the 46.1  $\mu\text{W}$  of two's complement format. Sign-magnitude requires 16.7% more than TC.

Power efficiency of the two multipliers *Mult1* and *Mult3* in the sign-magnitude based implementations is slightly better than in the two's complement design, which corresponds to expectations. On the other hand, *Mult2*, implemented as a sign-magnitude multiplier, requires more power, although one of its factors is constant ( $\beta$ ) and the hardware compiler optimized according to this fact.

In order to understand the rather unexpected behavior of the multiplier *Mult2*, the corresponding block in Fig. 4.9 has been modified. This situation is illustrated in Fig. 4.10. The actual multiplier and its preceding conversion have been coded in distinct VHDL entities, they correspond to the dashed boxes in the figure. More important, the register (D-FF's) has been relocated in between these two combinational blocks. The converter contains the conditional bit inversion and increment ( $\boxplus$ ). Glitches emanating from the incremter are blocked out by means of this configuration. The power consumption reported in Tab. 4.5 contains solely the purely functional dissipation of the constant-multiplier *Mult2*, since both factors, data  $l_{l-1}$  and constant  $\beta$ , can now be assumed glitch-free and all bits of the data input are synchronized with the clock.

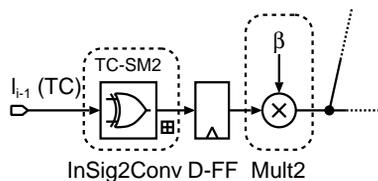


Figure 4.10: Modified *Mult2* block in the hybrid stage.

Table 4.5: Post Synthesis power simulation and cell area of the modified hybrid lattice stage.

Hierarchy	Switch <sup>a</sup> Power [ $\mu$ W]	Internal <sup>b</sup> Power [ $\mu$ W]	Leakage Power [ $\mu$ W]	Total Power [ $\mu$ W]	% Power	Area [ $\mu$ m <sup>2</sup> ]
Hybrid						
Mult1	5.3	5.4	4.6	15.3	33.1	32 432
(TC-SM1)	(0.0)	(0.0)	(0.1)	(0.2)	-	(942)
Mult2	1.3	0.7	1.2	3.2	6.9	7 999
InSig2Conv	0.1	0.2	0.2	0.4	0.9	2 210
(TC-SM2)	(0.0)	(0.0)	(0.1)	(0.2)	-	(942)
Mult3	6.4	6.9	4.2	17.6	37.9	28 845
Sub1	1.2	2.1	0.8	4.1	8.9	5 908
(SM-TC1)	(0.4)	(0.8)	(0.3)	(1.5)	-	(2 606)
Sub2	1.1	1.9	1.0	3.9	8.4	8 474
(SM-TC2)	(0.1)	(0.1)	(0.1)	(0.3)	-	(1 283)
(SM-TC3)	(0.3)	(0.5)	(0.3)	(1.0)	-	(2 598)
D-FF	0.2	0.4	0.9	1.6	3.4	2 788
TC-SM_k <sub>i</sub>	0.1	0.0	0.1	0.2	0.5	942
Total Stage	15.7	17.6	13.0	46.3	100.0	89 615

<sup>a</sup> Dissipated power by (dis-) charging the load capacitance of a cell.

<sup>b</sup> Dissipated power inside a cell. Refer to Sec. 2.2.2 for details.

Despite the more complex circuit, which cannot profit from the optimized converter-multiplication combination, its power consumption of 3.6  $\mu$ W is lower than the 4.5  $\mu$ W of the original block. This is due to the reduced glitch activity. Furthermore, this effect propagates downstream, leading to reduced dissipation in *Mult3* (17.6  $\mu$ W vs. 18.1  $\mu$ W) and to some extent in the two subtractors *Sub1* (4.1  $\mu$ W vs. 4.2  $\mu$ W) and *Sub2* (3.9  $\mu$ W vs. 4.0  $\mu$ W). Nevertheless, overall dissipation of 46.3  $\mu$ W of this modified stage is still higher than the 46.1  $\mu$ W of the two's complement implementation.

In both the hybrid and modified hybrid stages, power savings in the multipliers cannot compensate for the additional dissipation of the subtractors because of their input conversions.

Regarding area efficiency, both alternatives – sign-magnitude and hybrid implementation of the lattice stage – give rise to an area overhead compared to two’s complement. The reason can be mainly found in the increased area of the subtractors *Sub1* and *Sub2*.

## 4.4 Final Remarks to Number Formats

As demonstrated in Sec. 4.2, circuit structures of low complexity, such as single-MAC filters, can profit from hybrid number representations. For convenience, such a MAC unit has been redrawn in Fig. 4.11a. Typical for this structure is the possibility to merge the increment ( $\boxplus$ ) with the adder. Moreover, the accumulation register shields any downstream logic, i.e., adder or any block at the output, from glitches.

However, the analysis in Sec. 4.3 discloses a fundamental problem of hybrid signal processing circuits of nontrivial complexity, such as lattice filter stages implemented as isomorphic architectures. Switching back and forth between sign-magnitude and two’s complement representation is responsible for a power dissipation overhead dominated by glitch generation.

Special care needs to be taken when a format conversion is required before a multiplication, because this rather complex operation is sensitive to glitches. This situation can be found in lattice filters stages and for any of the multiplier inputs in the MAC unit in Fig. 4.11a that requires a format conversion from TC to SM. A direct, and hence inefficient, converter-multiplication implementation is shown in Fig. 4.11b. However, possibly an already existent functional register (D-FFs, red) can take over the task of a “glitch shield”. This register is relocated between the glitch creating converter including increment and the now shielded multiplier by means of retiming (green) [43].

The circuit structure surrounding the arithmetic units is largely responsible for the choice of the optimal number format with minimal power dissipation. For a hybrid solution, careful design of conversion units is required to merge increments and decrements into already existent adders or subtractors, respectively. As shown above, retiming of functional registers has to be considered in order to stop glitch

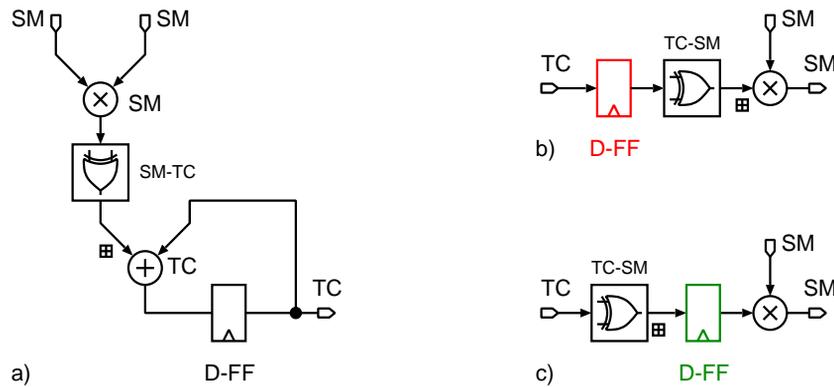


Figure 4.11: a) Hybrid MAC unit, b) inefficient multiplication with preceding converter, c) retimed and hence optimized multiplication with converter.

propagation in critical places. Breaking glitch paths between number conversion and subsequent arithmetic unit is often a good choice<sup>2</sup>.

As a concluding remark, the number format is likely not the dominating factor for overall power consumption of a digital signal processing system. As an example, the power dissipation of the MAC unit discussed in Sec. 4.2 is 21% (TC) and 17.7% (SM) of the overall consumption of the complete FIR-filter containing a large shift-register at the data input. In the following Chapter 5, several clocking strategies are evaluated in order to reduce the power consumption of the clock distribution network (clock tree). In order to fulfill timing constraints, the clock tree usually contains many power-hungry buffers because of the large shift register.

---

<sup>2</sup>Register retiming is not only limited to the context of number format conversion. It should be considered as a general low-power technique to shield any combinational logic from glitches created in an upstream circuit.

# Chapter 5

## Clocking Strategies for Low-Power

### 5.1 Synchronous Clocking Disciplines

While exploring possible power savings by the choice of optimal number formats, another, more dominant, factor has been encountered in the FIR-filter of order 99 in Sec. 4.2. There, the straightforward approach of using a shift-register for storing the input samples requires a lot of activity in this memory structure. Moreover, tight hold-time restrictions for the single-edge-triggered one-phase clocked flip-flops ask for many buffers in the clock tree.

Single-edge-triggered one-phase (SETOPC) clocking is a member of the family of synchronous clocking disciplines, the classification scheme of which is depicted in Fig. 5.1. The disciplines are divided into edge-triggered, flip-flop based, and level-sensitive, latch based, branches. Both branches are further subdivided. All disciplines share the common property that all data transactions are periodically triggered by one or two clock signals.

Furthermore, Fig. 5.1 shows that the members of the synchronous clocking family can be derived (dotted arrows) from SETOPC, considered to be the “mother” of all other considered disciplines. Implemen-

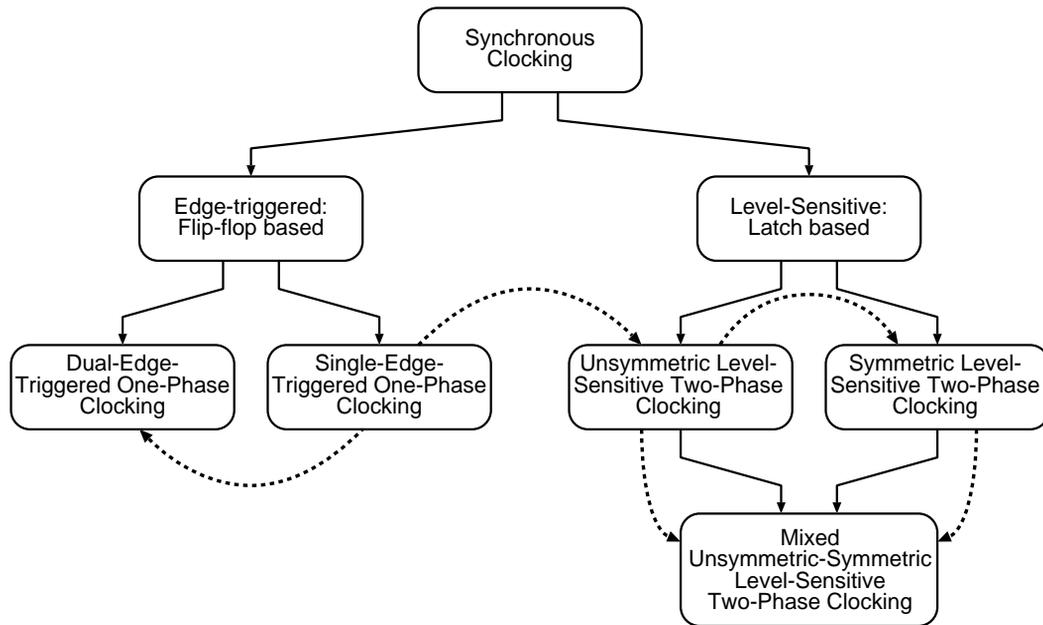


Figure 5.1: Taxonomy of synchronous clocking disciplines (simplified).

tation details and timing conditions of the clocking regimes in Fig. 5.1 will be discussed in the following subsections.

## 5.1.1 Single-Edge-Triggered One-Phase Clocking

### 5.1.1.1 Operation Theory and Timing Conditions

Nowadays, the vast majority of all digital signal processing chips make use of the single-edge-triggered one-phase clocking discipline. Finite state machines for the control path and data-dependency graphs for the data path of any digital signal processing application are easily transformed into a circuit with this clocking regime during register-transfer-level design.

All data transactions are scheduled to happen at *discrete* and periodic points in time, defined by the active edge of the single clock signal, usually the rising one. This leads to quite simple and easily understandable timing conditions that guarantee correct functioning: Set-up and hold conditions of all flip-flops must be satisfied. A detailed derivation and formulation of the related inequalities can be found in [43].

For the sake of simplicity, these inequalities are repeated in the following. The *set-up condition* of a flip-flop transforms into:

$$T_{clk} \geq \max(t_{pd,ff} + t_{pd,C} + t_{su,ff}) + \max|t_{sk}| = t_{lp} + \max|t_{sk}| \quad (5.1)$$

with the *hold condition*:

$$\max|t_{sk}| \leq \min(t_{cd,ff} + t_{cd,C} - t_{ho,ff}) = t_{sp}, \quad (5.2)$$

where  $T_{clk} = 1/f_{clk}$  denotes the clock period, which is equal to the computation period  $T_{cp}$ , i.e.,  $T_{clk} = T_{cp}$ , in the case of single-edge-triggered one-phase clocking. The propagation delay of a flip-flop and of the combinational logic is given by  $t_{pd,ff}$  and  $t_{pd,C}$ , respectively. The set-up and hold times of a flip-flop are identified by  $t_{su,ff}$  and  $t_{ho,ff}$ . Quantities  $t_{lp}$  and  $t_{sp}$  denote the longest (*lp*) and shortest (*sp*) of all possible combinational paths between any pair of flip-flops. The contamination delays are given by  $t_{cd,ff}$  and  $t_{cd,C}$ . The maximum difference of the arrival time of the active clock edges at any pair of flip-flops is given by:

$$\max|t_{sk}| = \max(t_{di}) - \min(t_{di}), \quad (5.3)$$

where  $t_{di}$  denotes the distribution or insertion delay of the clock signal. The quantity  $t_{sk}$  is commonly referred to as the skew of the clock net. This global view gives an upper bound of the maximum allowable skew, hence only the absolute value is considered. A more detailed approach analyses the skew between pairs of flip-flops which are connected, either directly (e.g. in a shift register) or via combinational logic. Thereby, a positive skew results whenever the receiving D-FF is triggered after the sending flip-flop, and a negative if the sending is triggered after the receiving one. In a shift-register a negative (or zero) skew is desired, since it prevents data-fallthrough.

Equation 5.1 directly translates into a measure on how fast a design can be clocked depending on its longest path and maximum skew. In hearing aid applications with their low clock frequencies, this constraint can be met without special care at nominal supply voltage. However, since hearing aids are usually supplied far below the nominal supply voltage of the CMOS technology, which results in increased propagation delays, special care is required in order to

prevent timing violations. The situation is completely different with the hold condition in Eq. 5.2. As can be seen there, this constraint is independent of the clock frequency as it solely relates the shortest combinational path to the clock skew, which depends only on the quality of the clock tree. Note that violating a hold condition, because of poor clock tree design, is worse than disobeying a set-up condition. While the latter can be “fixed” by reducing the clock frequency (at the cost of data throughput), the former cannot be corrected to make the silicon work. For this reason, special care has to be taken when synthesizing the clock network. Especially, skew between connected pairs of flip-flops has to be adjusted. Usually, this is done by balancing the related insertion delays by means of a tree of buffers. Additionally, hold time fixing can be applied, thereby delay buffers along data paths that violate the hold condition (Eq. 5.2) are inserted in order to delay such critical paths [43]. These buffers increase the contamination delay ( $t_{cd}C$ ) in Eq. 5.2.

As will be shown later, clock tree elements and delay buffers render single-edge-triggered one-phase clocking a rather unattractive solution for low-power and low-speed applications, such as hearing aids or audio processing in general.

A typical configuration of a SETOPC circuit is depicted in Fig. 5.2a. Purely combinational parts are denoted with **C** and **D**. In this simple example, *clock gating* has been applied; the pertaining clock gate is depicted in Fig. 5.2b [43]. This technique is an absolute must for low-power applications, since it dramatically reduces clocking activity and hence power consumption in temporarily inactive regions of the circuit. Tools can automatically introduce this technique during synthesis, without changing the VHDL code. Later on, during clock tree synthesis,

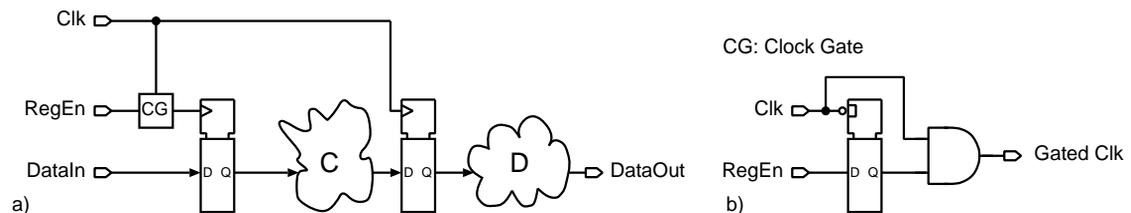


Figure 5.2: Single-edge-triggered one-phase clocking: a) circuit example and b) clock gate implementation.

which is typically carried out concurrently to place and route, the clock gate elements are recognized as part of the clock tree. Their additional propagation or insertion delays are compensated. Special care has to be taken to retain scan-chain testability: clock gates have to be switched off or bypassed completely during scan-chain testing, whereby the originally gated clock becomes completely equivalent to its ungated origin.

Because of its simplicity, excellent support by design tools, and broadest use in practice, single-edge-triggered one-phase clocking serves as a reference and starting point for all later presented clocking alternatives.

### 5.1.1.2 Practical Implications

Single-edge-triggered one-phase clocking can be considered as state-of-the-art, hence, automatic inclusion of clock gating and test structures (scan chains) is well supported by today's synthesis tools. Cell libraries (e.g., Virtual Silicon VIP<sup>TM</sup> Standard Cells for UMC 0.18  $\mu\text{m}$  CMOS technology) contain scannable flip-flops and dedicated clock gating cells. During all steps of circuit design, automated timing analysis can be executed. Creation of scan vectors for automatic test equipment (ATE) is well supported by software tools.

## 5.1.2 Dual-Edge-Triggered One-Phase Clocking

### 5.1.2.1 Operation Theory and Timing Conditions

As stated in Sec. 5.1.1, single-edge-triggered one-phase clocking (SE-TOPC) suffers from a large overhead in clock-tree elements because of its tight timing constraints imposed by the hold condition. Moreover, as a consequence of  $T_{clk} = T_{cp}$ , the clock tree, or at least its ungated subtrees, are transitioning twice (rising and falling edges) during each period, although only the active edge triggers a data transaction. Each edge of the clock-tree, whether productive or not, is equivalent to charging or discharging nodes inside the tree, which consumes power. As a remedy, both clock edges can be made active to trigger a data transaction. This leads to the *dual-edge-triggered* one-phase clocking (DETOPC). The energy wasted in the clock tree should be halved, at least in theory.

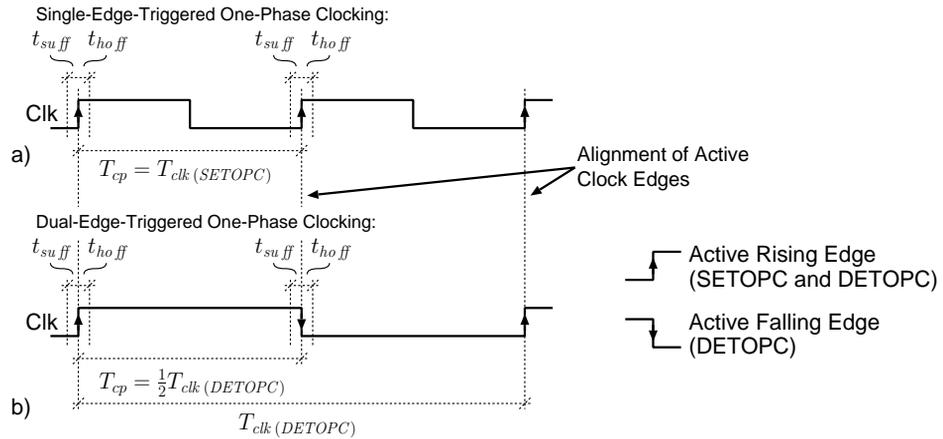


Figure 5.3: Edge-triggered clocking: a) Single-edge-triggered and b) dual-edge-triggered one-phase clocking.

The set-up and hold conditions are the same as in case of SETOPC although they are related to both the rising and falling edges of the clock signal (refer to Fig. 5.3b). Since two computation periods are executed during one clock period, the clock frequency of a dual-edge-triggered one-phase clocking (DETOPC) design has to be halved compared to the one of a SETOPC in order to maintain a constant computation period and data throughput:

$$T_{cp} = \frac{1}{2}T_{clk(DETOPC)} = T_{clk(SETOPC)} \geq t_{lp}. \quad (5.4)$$

Figure 5.3 compares the clock and computation periods of these two edge-triggered clocking disciplines. Note the alignment of the active edges as a consequence of Eq. 5.4.

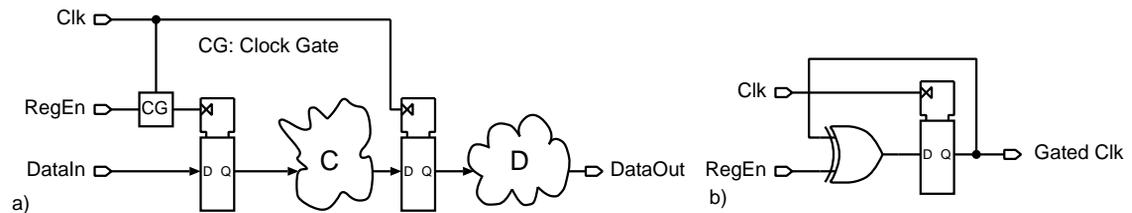


Figure 5.4: Dual-edge-triggered one-phase clocking: a) circuit example and b) clock gate.

The corresponding hardware situation is represented in Fig. 5.4a where single-edge-triggered (SET) flip-flops have been replaced by their dual-edge-triggered (DET) counterpart. Combinational blocks **C** and **D** remain unchanged. Figure 5.4b shows an implementation of a clock gate [43].

### 5.1.2.2 Practical Implications

Dual-edge-triggered one-phase clocking requires significant manual interaction with electronic design automation (EDA) tools. Although this clocking strategy is closest related to single-edge-triggered clocking, its tool support is worst. Synthesis tools, such as Design Compiler by Synopsys and Ambit BuildGates by Cadence, do not recognize the hardware description language construct that describes a dual-edge-triggered bistable element. In addition, current standard cell libraries do not offer such flip-flops. Even worse, the syntax of these libraries does not support this cell type and hence does not offer a possibility to constrain timing paths regarding both edges of the clock signal.

Nevertheless, a workaround for devising a dual-edge-triggered design with current EDA tools is possible by exploiting the close relation to single-edge-triggered clocking. For this work, a dual-edge-triggered flip-flop has been implemented including layout drawing and timing characterization. This flip-flop cell internally consists of a multiplexer and two latches, see Fig. 5.5 [43] for details. The cell description including timing data has been provided to Ambit BuildGates. The functionality in this cell model equals a single-edge-triggered flip-flop. The VHDL code provided to the synthesis tool has been the same as

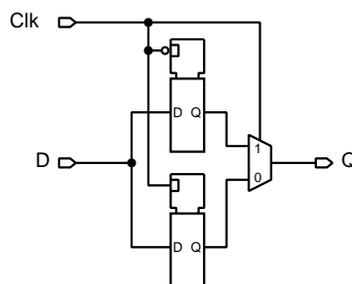


Figure 5.5: Equivalent circuit diagram of a dual-edge-triggered flip-flop.

in case of a SETOPC design. Especially, VHDL statements describing SET D-FFs have been left unaltered.

Although the final DET design is intended to run with half of the clock frequency of the corresponding SETOPC design, see Fig. 5.3 and Eq. 5.4, the clock period for this synthesis has to be equal to the one of a SET synthesis. Timing engines that are part of synthesis tools accept only rising edges as reference points for set-up and hold conditions, and the workaround must constrain an intended DET design by making use of rising edges only. This “over-constraining” of the clock frequency by a factor two during this SET-based synthesis restricts the length of the longest path ( $t_{lp}$ ) to the length of a computation period ( $T_{cp}$ ), corresponding to half of the clock period in a DET design, compared with the two waveforms in Fig. 5.3. Therefore, the longest path is constrained to half of the DET clock period:  $T_{cp} = \frac{1}{2}T_{clk(SETOPC)}$ . This workaround does not consider any possible differences between rising and falling clock edges: Set-up ( $t_{su,ff}$ ) and hold ( $t_{ho,ff}$ ) times, and propagation delays ( $t_{pd,ff}$ ) of a D-FF may be different depending on the direction of the clock edge. Thus, timing data of the DET flip-flop provided to the synthesis tool have to contain the worst case between rising and falling edge.

A variation of this workaround could be to perform an “over-constrained” synthesis of a netlist with SET D-FFs. In the next step, these flip-flops are replaced with their DET counterparts or with a macro that contains a multiplexer and two latch cells as in Fig. 5.5. This very last variant is appropriate if no dual-edge-triggered flip-flop can be developed because the design kit does not support full-custom layout or, simply, the development effort has to be reduced.

Testability can be added by placing a multiplexer in front of the flip-flop or by integrating the same functionality into a scannable flip-flop cell, which is to prefer when designing and layouting a dual-edge-triggered flip-flop from scratch. Creating test vectors by automatic test pattern generation (ATPG) imposes similar problems to synthesis, since again, the required software tools (e.g., Synopsys TetraMAX) do not support DET flip-flops. Nevertheless, test patterns can be created by again “cheating” the tool with a final netlist containing scannable single-edge-triggered flip-flops instead of dual-edge-triggered.

### 5.1.3 Unsymmetric Level-Sensitive Two-Phase Clocking

#### 5.1.3.1 Operation Theory and Timing Conditions

An approach to circumvent the tight clock-tree constraints of SETOPC and DETOPC is the latch-based, unsymmetric level-sensitive two-phase clocking (ULSTPC) scheme. As shown in Fig. 5.6, the clock period  $T_{clk}$ , which is equal to the computation period  $T_{cp}$ , is subdivided into four intervals ( $T_1$ – $T_4$ ). Instead of one single clock signal, a master-slave clock pair is used. Each flip-flop is replaced by a master-slave latch with no combinational logic in between. Functionally, these latches are identical to a single-edge-triggered flip-flop, which itself consists of a master-slave. The second clock is internally derived from the original clock, see the left part of Fig. 5.6<sup>1</sup>. With ULSTPC, the two clocks are accessible from outside and can therefore be generated completely non-overlapping. This ensures that at any time only one latch is transparent (either during  $T_1$  or  $T_3$ ) and the duration of the non-overlapping or dead time phases  $T_2$  and  $T_4$  are externally adjustable, and with them, the skew margins.

The clock **Clk** in SETOPC corresponds to the slave clock **SlaveClk** in ULSTPC, because these signals specify when data appears at the output of the sequential element. Accordingly, the internally inverted clock  $\overline{\text{Clk}}$  corresponds to the master clock **MasterClk** since both define when data is captured into the flip-flop or latch.

An in-depth analysis of the circuit configuration in Fig. 5.7a yields the following set-up conditions for ULSTPC [43]:

$$\max |t_{sk}| \leq T_{clk} - T_2 - \max(t_{pd\,lc\,S} + t_{pd\,C} + t_{su\,l\,M}), \quad (5.5)$$

$$\max |t_{sk}| \leq T_2 + T_3 - \max(t_{pd\,ld\,M} + t_{su\,l\,S}). \quad (5.6)$$

The pair of conditions is required because constraints of both the master and the slave latch have to be met. Additionally, note the presence of two propagation delay paths in a latch: the first starts at the clock input ( $t_{pd\,lc}$ ) while the second one begins at the data

---

<sup>1</sup>In SETOPC, the small negative skew resulting from the clock inversion in this implementation is desired. Thus, the slave changes its state before the master, which prevents the flip-flop from a short transparent interval after the falling edge of the clock.

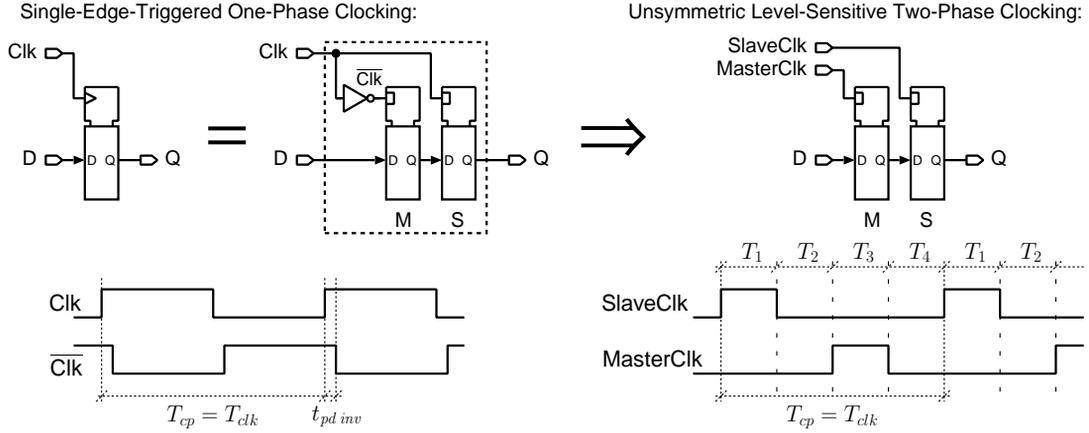


Figure 5.6: Transformation of a single-edge-triggered one-phase clocked flip-flop into a level-sensitive two-phase clocked latch pair with related clock waveforms.

input ( $t_{pdld}$ ). The extra subscript of these timing quantities ( $M$  or  $S$ ) denotes whether they are related to the master or slave latch, respectively. The set-up time of the latches is given by  $t_{sul}$ , again with the extra subscript for the master or slave latch. The first condition (Eq. 5.5) corresponds to Eq. 5.1, the second one (Eq. 5.6) does not appear in SETOPC, but has to be met by design of the D-FF. The pair of hold conditions can be found as [43]:

$$\max |t_{sk}| \leq T_2 + \min (t_{cdlcS} + t_{cdC} - t_{holM}), \quad (5.7)$$

$$\max |t_{sk}| \leq T_4 + \min (t_{cdlcM} - t_{holS}). \quad (5.8)$$

The timing quantity  $t_{cdlc}$  denotes the contamination delay of a latch regarding its clock input, whereas  $t_{hol}$  denotes the hold time of latch. The first condition (Eq. 5.7) and Eq. 5.2 correspond to each other and the second condition (Eq. 5.8) does not have an equivalent in SETOPC. Additionally, the following equation must be met:

$$T_{clk} = T_1 + T_2 + T_3 + T_4. \quad (5.9)$$

This equation reflects the fact that all phases must be strictly non-overlapping.

From the point of view of the set-up and hold conditions, single-edge-triggered one-phase clocking can be considered as a special case

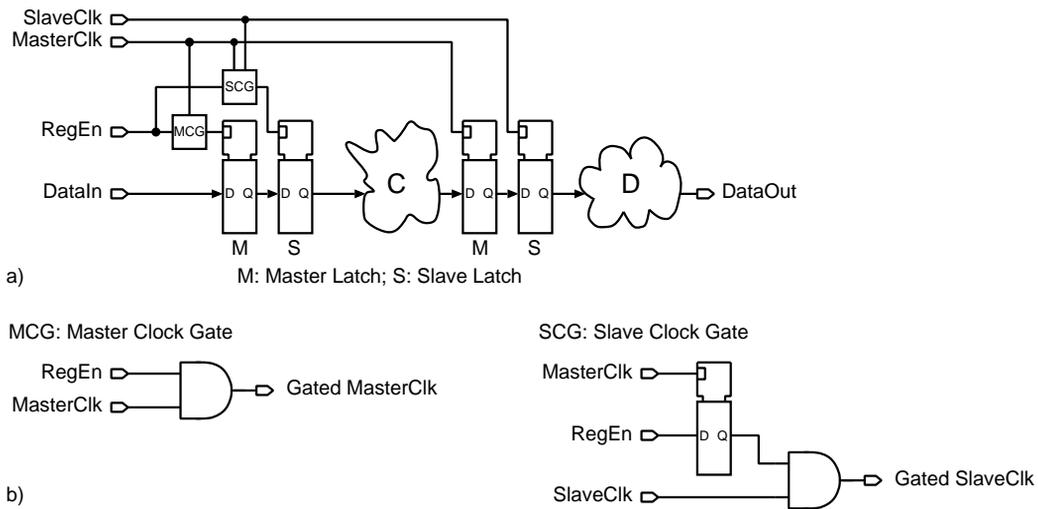


Figure 5.7: Unsymmetric level-sensitive two-phase clocking: a) circuit example, and b) master and slave clock gate implementation.

of unsymmetric level-sensitive two-phase clocking in which the non-overlapping phases  $T_2$  and  $T_4$  are close to zero. As can be seen from Eq. 5.7, the presence of the non-productive phase  $T_2$  renders ULSTPC more skew tolerant than SETOPC. Moreover, the maximum tolerable skew can be adjusted from outside by increasing the length of  $T_2$ . On the other hand, from Eq. 5.5 it follows that this advantage has to be bought for a reduced maximum working frequency. The word *unsymmetric* in the name of this clocking regime indicates that the combinational logic is completely unsymmetrically distributed between the latches, and hence, phase  $T_2$  is totally unproductive.

### 5.1.3.2 Practical Implications

Since tools do support level-sensitive latch design and cell libraries do contain completely characterized latches, developing an unsymmetric level-sensitive two-phase circuit is straightforward. Presumably, already existing VHDL code describing a SETOPC design is required to be transformed into a power-saving ULSTPC counterpart. Nevertheless, if such flip-flop based code does not yet exist, a direct approach to ULSTPC doing without the detour via SETOPC can be followed. However, the basic idea consists in superseding flip-flops by a latch pair and replacing all clock signals by a master and a slave clock. These steps

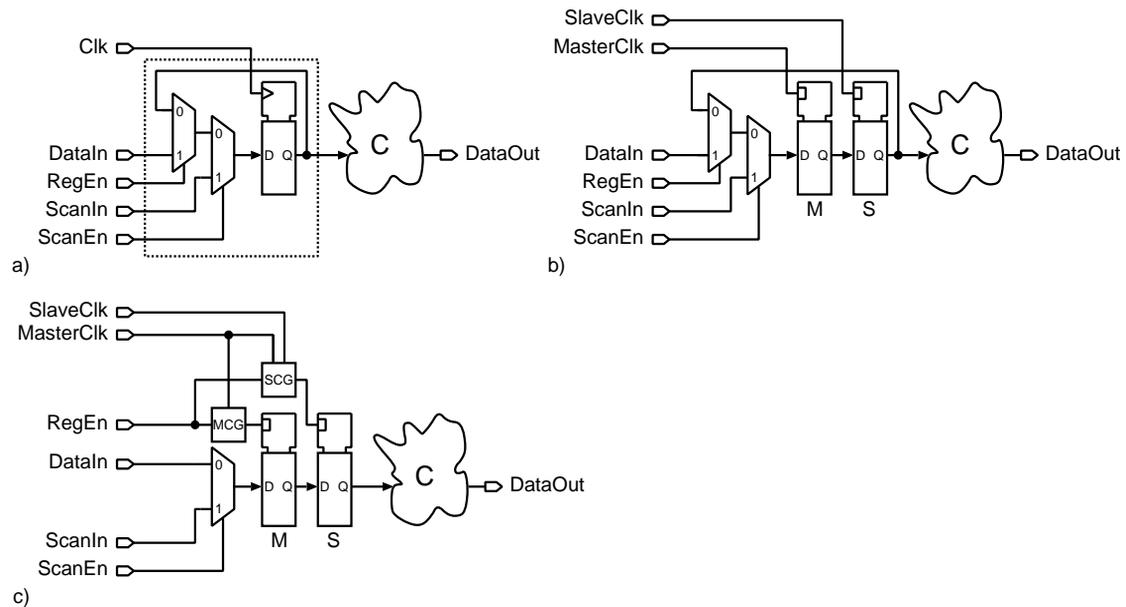


Figure 5.8: Developing a ULSTPC circuit: a) scannable SETOPC reference circuit, b) equivalent ULSTPC circuit, and c) ULSTPC circuit with added clock gating.

are most easily carried out on the register-transfer level immediately before hardware synthesis. Instead of hardware description language statements (e.g., VHDL process statements [61]) representing flip-flops, a pair of statements yielding latches is used. If a scan chain facility is required, a multiplexer should be added in front of the master latch. Additionally, since automatic inclusion of clock gating is currently not supported for latch-based designs, this important low-power technology has to be manually added by VHDL coding. This development process is delineated in Fig. 5.8, which shows the circuit of a scannable register with enable signal and subsequent combinational logic **C**. The SETOPC circuit in Fig. 5.8a serves as a reference that is functionally replicated as a ULSTPC design in Fig. 5.8b. The multiplexed scan flip-flop in the reference provides a scan and an enable input. Usually, such flip-flops (dashed frame) are included in cell libraries. Figure 5.8b shows the equivalent ULSTPC circuit, where the flip-flop has been replaced by a master and a slave latch. In Fig. 5.8c clock gating has been added. The clock gates are implemented as shown in Fig. 5.7b.

## 5.1.4 Symmetric Level-Sensitive Two-Phase Clocking

### 5.1.4.1 Operation Theory and Timing Conditions

Additional power savings can be expected by employing *symmetric* level-sensitive two-phase clocking (SLSTPC). As the name suggests, the combinational logic is distributed between each pair of latches, see Fig. 5.9, and that phase  $T_2$  is also productive. Similar observations as in the case of *unsymmetric* level-sensitive two-phase clocking lead to the following set of timing constraints [43].

Set-up conditions:

$$\begin{aligned} \max |t_{sk}| &\leq T_{clk} - T_2 - \max(t_{pd\ lc\ S} + t_{pd\ C2} + t_{sul\ M}) \\ \max |t_{sk}| &\leq T_{clk} - T_4 - \max(t_{pd\ lc\ M} + t_{pd\ C1} + t_{sul\ S}) \\ 0 &\leq T_{clk} - \max(t_{pd\ ld\ M} + t_{pd\ C1} + t_{pd\ ld\ S} + t_{pd\ C2}) \end{aligned} \quad (5.10)$$

Hold conditions:

$$\begin{aligned} \max |t_{sk}| &\leq T_2 + \min(t_{cd\ lc\ S} + t_{cd\ C2} - t_{hol\ M}) \\ \max |t_{sk}| &\leq T_4 + \min(t_{cd\ lc\ M} + t_{cd\ C1} - t_{hol\ S}) \end{aligned} \quad (5.11)$$

It can be seen that these constraints impose similarly loose hold conditions as the related equations of *unsymmetric* level-sensitive two-phase clocking (Sec. 5.1.3, Eq. 5.5–5.8). *Symmetric* level-sensitive two-phase clocking can be considered as an enhancement of *unsymmetric* level-sensitive clocking. The clock gates used in Fig. 5.9 can be implemented exactly the same way as in ULSTPC, see Fig. 5.7b.

*Symmetric* level-sensitive two-phase clocking is an attractive alternative for low-power circuits due to the following observations: Breaking the combinational blocks (**C** and **D**) into halves (**C1**, **C2** and **D1**, **D2**), and relocating latches in between, blocks glitches emanating from the first halves (**C1**, **D1**). They are inhibited to penetrate into the second half of each logic block and to create unnecessary power consuming activity. This feature can be fully exploited only if sub-blocks **C1** and **D1** complete evaluation of their inputs before the slave latches get transparent and would pass the glitches (during phase  $T_1$ ).

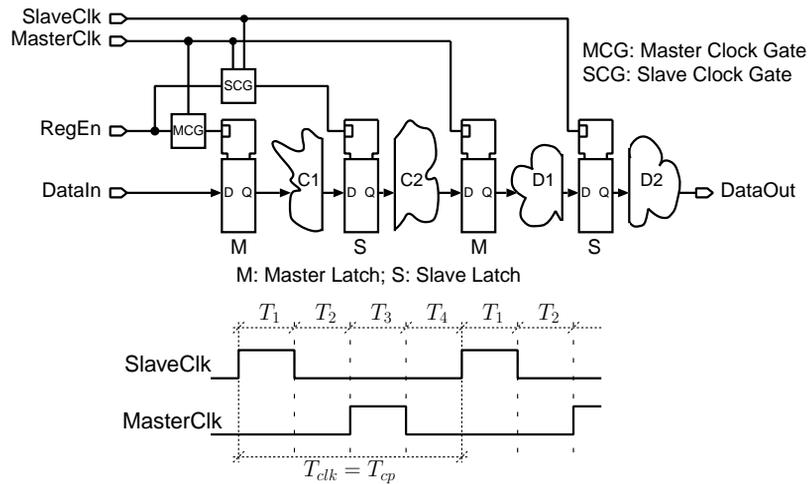


Figure 5.9: Symmetric level-sensitive two-phase clocking with related clock waveforms.

Likewise, sub-blocks **C2** and **D2** have to complete their evaluation before the master latches get transparent (during phase  $T_3$ ). In other words, the so-called *time borrowing* [43] has to be prevented, i.e., no unused processing time should be traded between **C1** and **C2**, or **D1** and **D2**. Provided that the timing constraints in Eqs. 5.10 and 5.11 are met, time borrowing does not let the circuit fail.

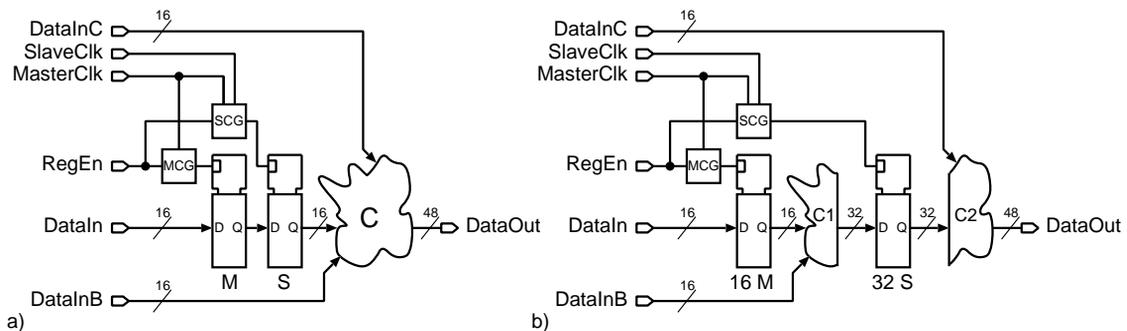


Figure 5.10: Developing an SLSTPC circuit: a) non-scannable ULSTPC clock-gated reference circuit, and b) equivalent SLSTPC circuit.

#### 5.1.4.2 Practical Implications

Since *symmetric* level-sensitive two-phase clocking can be considered as an evolution of *unsymmetric* level-sensitive two-phase clocking, the former is best derived from the latter. Figure 5.10 shows how the combinational logic **C** (a) is split into halves **C1** and **C2** (b). However, this situation has already been shown in Fig. 5.9. In the following, this circuit serves as a model to study several issues of *symmetric* level-sensitive two-phase clocking. Sub-blocks **C1** and **C2** could be considered as multipliers whose output widths equal the sum of their input widths.

As can be seen in the example in Fig. 5.11b, the number of master and slave latches does not match as they amount to 16 and 32, respectively. This complicates the building of a scan chain. A possible solution to this problem would be to add “dummy” master latches in order to equalize the number of both latch types. These additional latches would be active only during scan chain testing. In Sec. 5.2.2.4 this solution is described in more details. There, only functional slave latches are present and the corresponding number of dummy master latches has been added in order to create scannable pairs comprising a master and a slave latch.

Another issue are flip-flops with enable inputs. Figure 5.11a shows the ULSTPC reference circuit without clock gating that is to be transformed into its SLSTPC equivalent. In Fig. 5.11b, two attempts to create the feedback that is required in order to implement the functionality of an enable signal are shown in this SLSTPC circuit. The first try (①) takes the feedback from the output of **C2**, which is 48 bit wide. This creates a conflict when closing the feedback loop, since the multiplexer and the master latch bank are only 16 bit wide. The second attempt (②), which takes the feedback after the slave latch bank, suffers from a similar mismatch: its width amounts to 32 bit. Moreover, even if the bit widths can be matched by some combinational reorganization, both feedback variants of the circuit in Fig. 5.11b are likely to fail. In both cases, combinational logic, either **C1** (②), or **C1** and **C2** (①), is present in the feedback loop. This logic presumably changes the value being fed back, which causes oscillations that destroy correct functionality.

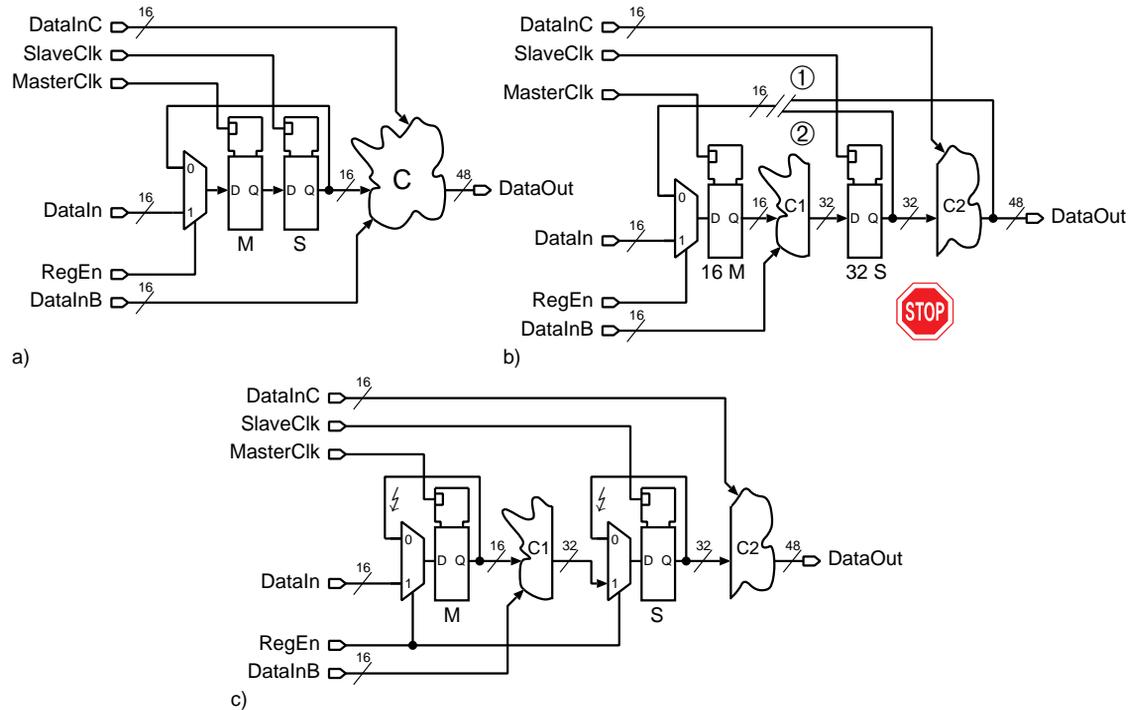


Figure 5.11: Issues in ungated *symmetric* level-sensitive two-phase clocking: a) *unsymmetric* level-sensitive two phase clocking reference circuit, b) not realizable *symmetric* level sensitive two-phase clocking circuit, c) *symmetric* level sensitive two-phase clocking circuit realizable with precaution.

Seemingly, the circuit in Fig. 5.11c solves both problems. The two local feedbacks that immediately start behind both latch banks exhibit a consistent width, and, apart from the multiplexers, no disturbing combinational logic is present in the loops. Nevertheless, this circuit suffers from two new issues. First, assume that either the master or slave latches are transparent (pass mode), because one of the clocks is active, and that the register enable (**RegEn**) is disabled. This leads to a closed feedback loop that is equal to a so called zero-latency loop [43], a purely combinational circular path without latency. Their possible locations have been marked with a lightning ( $\zeta$ ) in the figure. Although this fact does not let the circuit functionally fail – because the feedback value is constant – it requires manual interaction with all timing tools during synthesis, and place and route. These tools only recognize the feedback loop but are not aware of the constant feedback

value. Second, the register-enable signal (**RegEn**) has to be glitch free, since a glitch ('0'-'1'-'0') on this signal may cause a malfunction when the feedback value does not match the other multiplexer input during the pass mode of a disabled latch (**RegEn** disabled except for the glitch).

Figure 5.11 demonstrates that a simple and robust solution without clock gating is difficult to be found. In Fig. 5.10b, all these issues have been solved by making use of clock gating. It can be stated that in the SLSTPC scheme, clock gating serves not only as a low-power technique, but, in addition, it simplifies the complete design phase of such level-sensitive designs. Nevertheless, the alternative depicted in Fig. 5.11c can be employed under certain circumstances.

### 5.1.5 Mixed Unsymmetric-Symmetric Level-Sensitive Two-Phase Clocking

Unfortunately, it is not always possible to turn an ULSTPC design into its SLSTPC equivalent. Macrocells and IP-Blocks such as arithmetic units in the Synopsys DesignWare library are not amenable to the therefore required architectural reorganization because it is hardly possible to subdivide these blocks. Other subcircuits split up easily, but a transformation to SLSTPC asks for so many so-called shimming latches at the inputs or outputs to compensate for the modified latency that the proposition becomes unattractive. This situation is comparable to retiming or pipelining of a combinational block that asks for shimming registers at its inputs or outputs in order to balance their latencies [43]. Figure 5.12a shows an example of a ULSTPC design containing a combinational block (**E**) with two unrelated outputs featuring different bit widths (16 bit and 32 bit). The block easily splits up into sub-blocks **E1** and **E2**, though by relocating the slave latches (blue) of the input register between the two sub-blocks in order to realize an SLSTPC design, shimming slave latches are required at **DataOut B**. These latches have been colored red in Fig. 5.12b. In this example, **DataOut A** is a function of **E1** and **E2**, whereas **DataOut B** depends on **E1** only. This means that only the data path yielding in **DataOut A** passes through the actual relocated latches (green). In order to maintain the same latency conditions at both outputs, 32 shimming

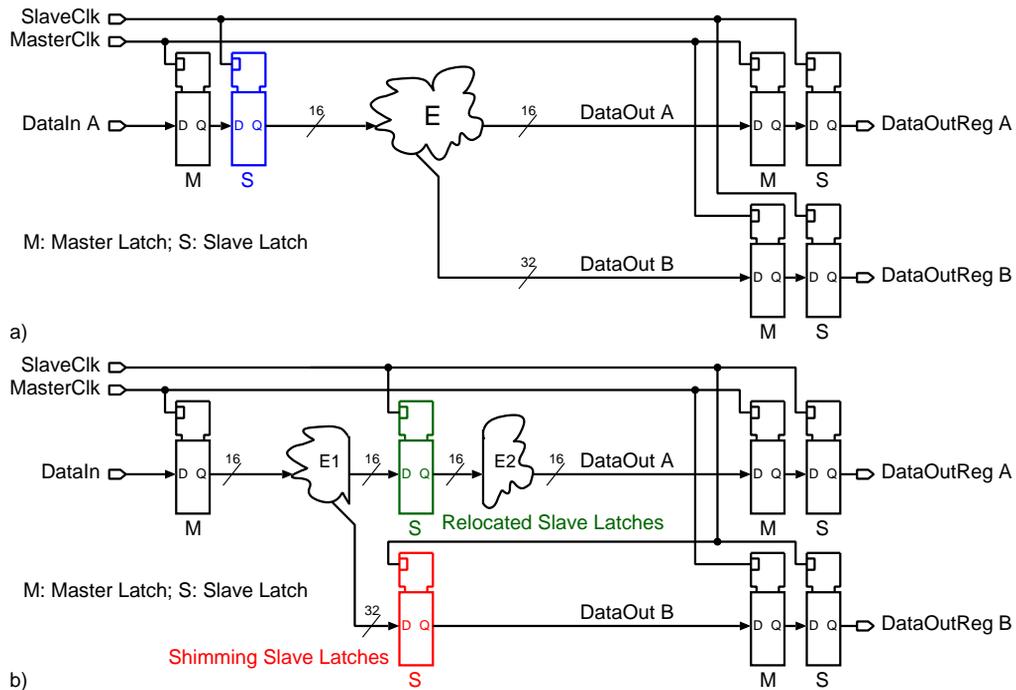


Figure 5.12: Transformation of an unsymmetric level-sensitive two-phase clocking design (a) into a symmetric level-sensitive two-phase clocking design that asks for shimming latches (b).

slave latches have been added at **DataOut B**. They are responsible for an area and power overhead.

The good news is that *unsymmetric* and *symmetric* level-sensitive two-phase clocking can be combined yielding the *mixed* unsymmetric-symmetric level-sensitive two-phase clocking (MLSTPC) symbolically illustrated in Fig. 5.13, where logic block **F** could not be subdivided. The separation between *symmetric* and *unsymmetric* parts is indicated with a dashed line in the figure. In this circuit example, clock gating has been applied to the *symmetric* part only. If the registers in the *unsymmetric* part were featuring an enable signal, clock gating could be applied there as well. It should be clear that depending on the actual type of the sub-block at hand, either timing conditions for *unsymmetric* (Eqs. 5.5–5.8) or *symmetric* (Eqs. 5.10–5.11) level-sensitive clocking apply.

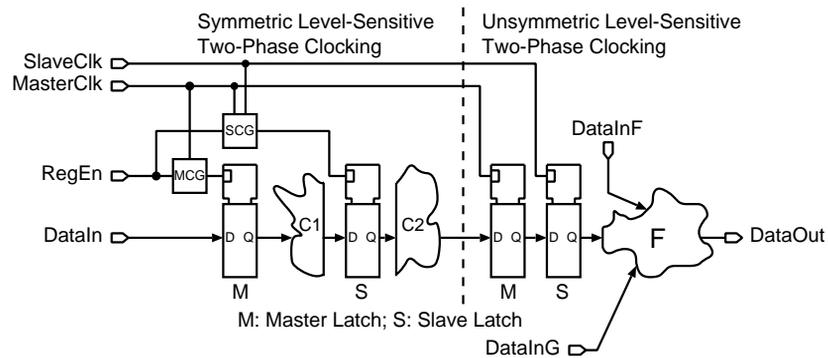


Figure 5.13: Mixed unsymmetric-symmetric level-sensitive two-phase clocking.

## 5.2 Fabricated Designs

### 5.2.1 FIR Filter I

#### 5.2.1.1 Chip Design

In order to study the effect of level-sensitive two-phase clocking on power consumption, three variants of a general-purpose FIR filter of order 100 have been integrated in a 0.25  $\mu\text{m}$  CMOS process. Variant **A** is a two's complement, single-edge-triggered one-phase clocked reference without clock gating. Version **B** features clock gating and the hybrid number format, but still makes use of single-edge-triggered clocking. Variant **C** is a clock gated, *unsymmetric* level-sensitive two-phase design. Data and coefficient register files are latch-based in all three versions. The block diagram of variant **C** is shown in Fig. 5.14. Rectangles represent sequential blocks, whereas boxes with rounded corners stand for combinational logic. Variants **A** and **B** have the same basic architecture as variant **C**. More detailed design characteristics can be found in Tab. 5.1.

The lack of clock gating makes design **A** a very power-inefficient reference. As a matter of fact, the un-gated clock net capacitance of **A**, which is charged and discharged during each period, is very large. Therefore, the clock tree generator is forced to instantiate many energy consuming buffers for clock distribution. This fact, together with the increased flip-flop-internal (dynamic) power consumption,

Table 5.1: Characteristics of the FIR Filter I variants. (simulated entries are in italics).

<b>Design Characteristic</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
Number Format	TC	HY	HY	<i>HY</i>
Clocking Strategy	SETOPC	SETOPC	LSTPC	<i>LSTPC</i>
Clock Gating	No	Yes	Yes	<i>Yes</i>
Clock Tree Cells	9 Inverters	1 Buffer	None	<i>None</i>
Core Area (mm <sup>2</sup> )	0.71	0.55	0.56	<i>0.34</i>
Standard Cell Density	82%	82%	82%	<i>n.a.</i>
Energy ( $\mu\text{W}/\text{MHz}$ ) at 2.5 V/1.25 V	1030/-	220/51	280/41	<i>186.5/36</i>
Clock Transition Time (ns) at 2.5 V	<i>1.4</i>	<i>0.9</i>	<i>8.5 (Master)</i>	<i>7.1 (Master)</i>

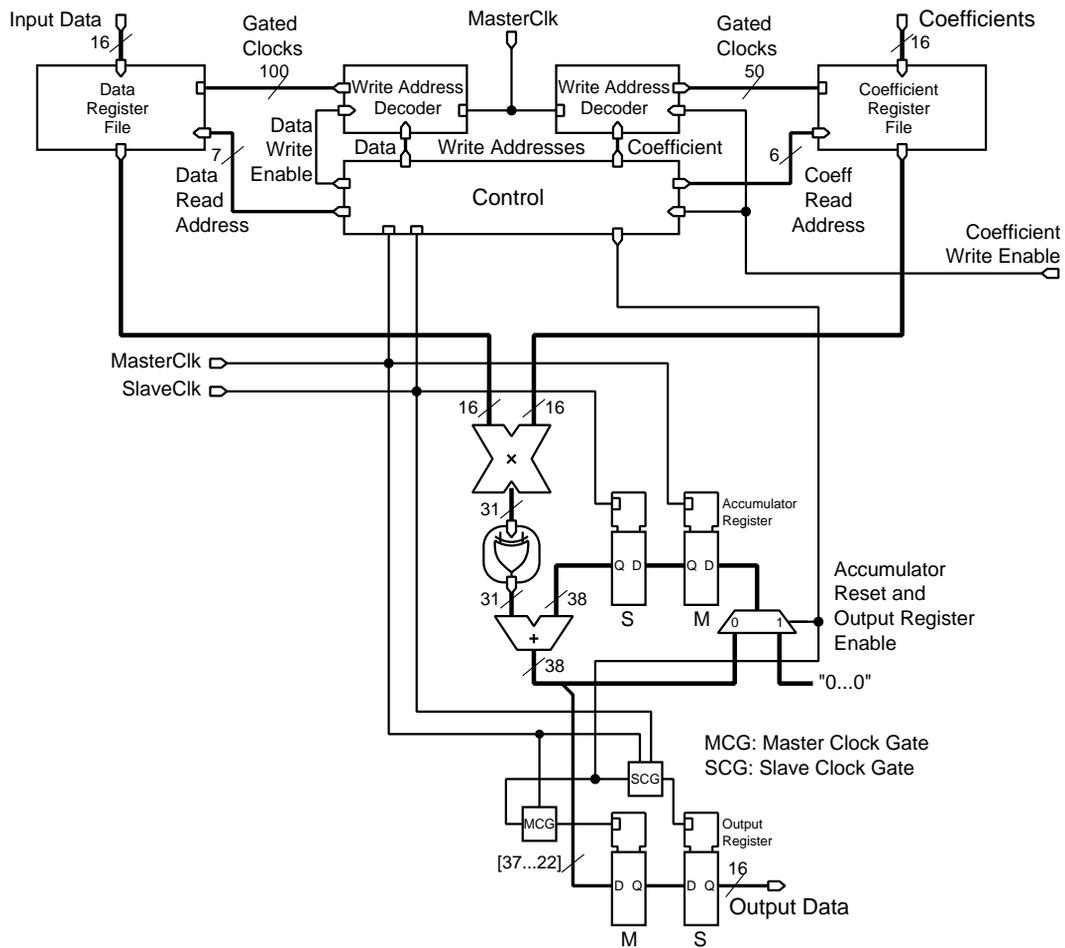


Figure 5.14: Block diagram of the fabricated FIR filter I featuring hybrid number format and *unsymmetric* level-sensitive two-phase clocking (variant C).

causes an overall dissipation that is 4 to 5 times higher than for the other designs.

The clock gated, single-edge-triggered version **B** is a robust state-of-the-art design. It has a single clock buffer to drive the numerous clock gates. Clock gating in this version has been implemented in two ways: For the latch-based register files it has been manually coded in VHDL because, as stated already in Sec. 5.1.3.2, EDA tools do not support clock gating for blocks based on latches. On the other hand, the hardware compiler was able to automatically add clock gating for the output register and the control block.

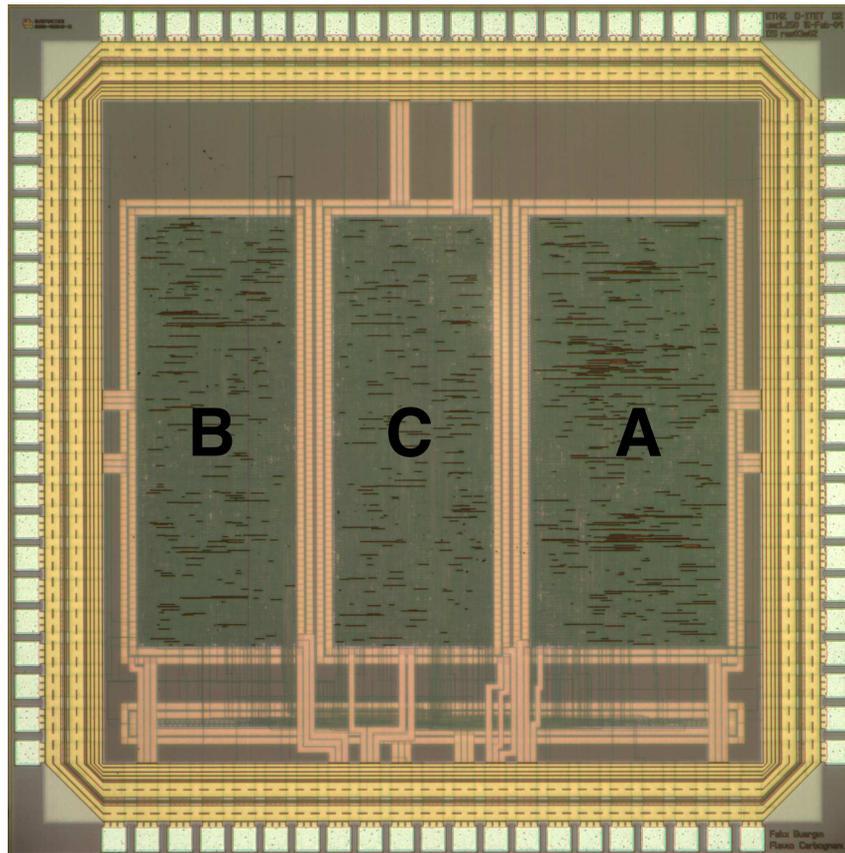


Figure 5.15: Die photo that contains the three variants of FIR Filter I.

As can be seen in Fig. 5.14, the two write address decoders in variant **C** are sequential blocks because they not only decode the addresses but also create the gated master clocks that trigger writing into the register files. These master clock gates consist of **AND** gates (see Fig. 5.7b). Clock gating in such write address decoders is discussed in more details in Sec. 5.2.2.2. These decoders in variant **C** contain clock gating as shown in Fig. 5.26. All clock gates have been added manually. Intentionally, no inverters and buffers are present in the clock network of variant **C**. Therefore, the benefits of this level-sensitive two-phase clocked design are exploitable only at low supply voltages, as will be shown later.

All three filter cores have been coded in VHDL and synthesized using a standard cell library and Synopsys Design Compiler. Relaxed timing constraints for a clock period of 500 ns have been chosen as minimum requirement for the signal processing of data sampled at

20 kHz. The ratio of clock and sampling frequency is a consequence of decomposing the FIR filter<sup>2</sup>. Clock tree generation, cell placement, and routing have been performed using Silicon Ensemble by Cadence. The same timing constraints as during synthesis have been applied in Silicon Ensemble.

As can be seen in the photo in Fig. 5.15, the three versions – in the order **B**, **C**, **A** on silicon – share the same die. Nevertheless, their power consumption can be separately measured due to distinct power rings and pads. The wide block of small height at the lower edge of the die contains a multiplexer that selects the output of **A**, **B**, or **C**.

### 5.2.1.2 Power Measurements

The measured energy consumptions (in  $\mu\text{W}/\text{MHz}$ ) of the three designs are plotted in Fig. 5.16 as functions of the supply voltage for a typical speech signal. It can be seen that the power dissipation of the reference design **A** is worse than the one of designs **B** and **C** for all supply voltages. The following discussion focuses on the behavior of the latter two designs. The two curves of variants **B** and **C** show a quite unanticipated behavior, which can be summarized as follows:

1. The two curves exhibit an unexpected cross-point at around 2.1 V: for larger voltages, **B** dissipates less, for lower voltages, **C** is more efficient (20% energy saving at 1.25 V).
2. The energy consumption of **B** and **C** is not proportional for all supply voltages. The two curves do not show the same characteristic.

### 5.2.1.3 Discussion of Measurement Results

The following discussion gives an explanation for the unexpected cross-point in Fig. 5.16 by comparing results of analog simulations performed with Spectre by Cadence with the measurements.

The difference between the clock network of design **B** and design **C** is illustrated in Fig. 5.17. Version **B** contains a single central clock buffer driving a tree-like net (dark blue) with flip-flops or, where required, clock gates as leaf cells. Some flip-flops have a delay cell

---

<sup>2</sup>More precisely, the multiplier is time shared between the filter taps, and the addition is iteratively decomposed yielding one accumulator adder [43].

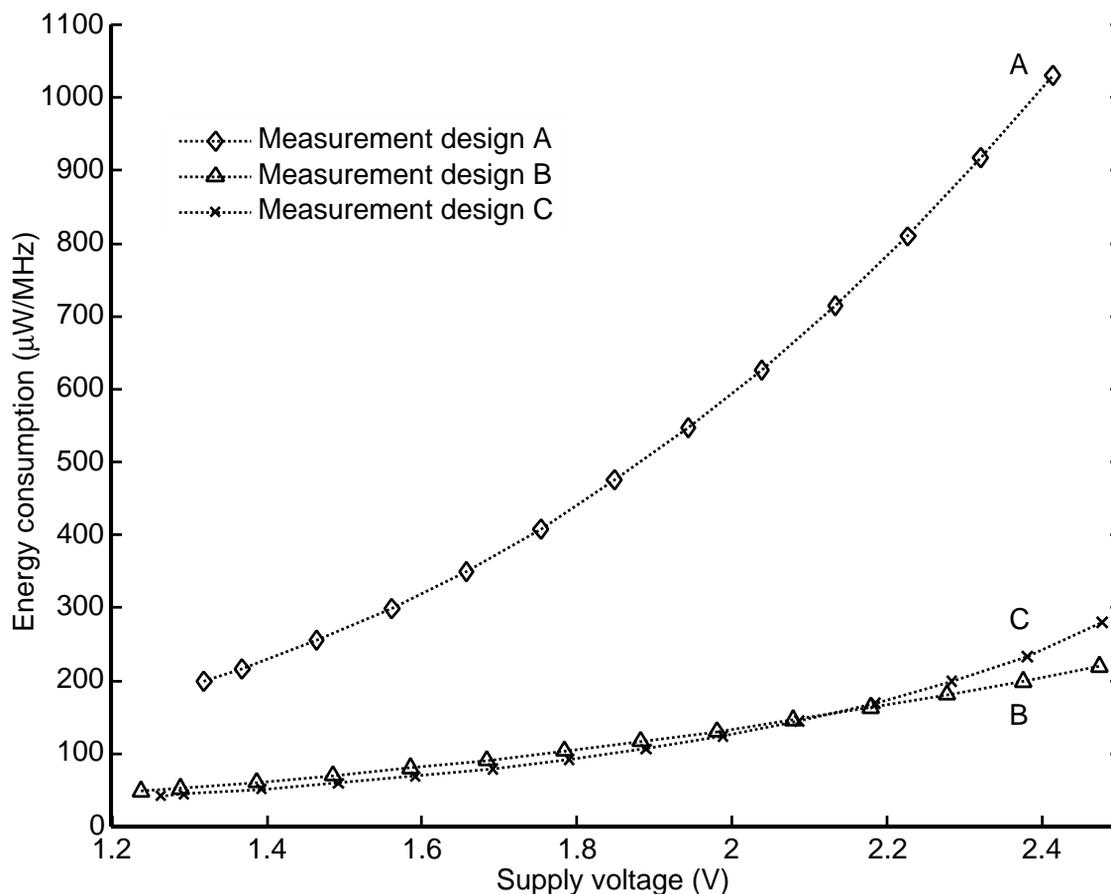


Figure 5.16: Measured energy dissipation of designs **A**, **B**, and **C** as function of supply voltage.

(buffer) along its data input to meet hold time requirements. In design **C**, neither clock buffers nor delay cells are required. Simulating the circuits in Fig. 5.17a and b yields the pair of curves in Fig. 5.18. The power dissipations of these circuits will be called “sequential energy” later on. In order to simplify the comparison between the behavior of the simulated sequential energies in Fig. 5.18 and the measured power consumptions of designs **B** and **C** in Fig. 5.16, these measurements have been re-plotted in Fig. 5.19, using the same scale as in Fig. 5.18.

The pair of curves in Fig. 5.18 and Fig. 5.19 shows similar characteristics. In particular, both pairs cross at around 2.1 V. The difference between the curve of design **B** in Fig. 5.19 and its sequential energy in Fig. 5.18 is equivalent to the power dissipation of the combinational logic. The same observation also holds for design **C**. For both

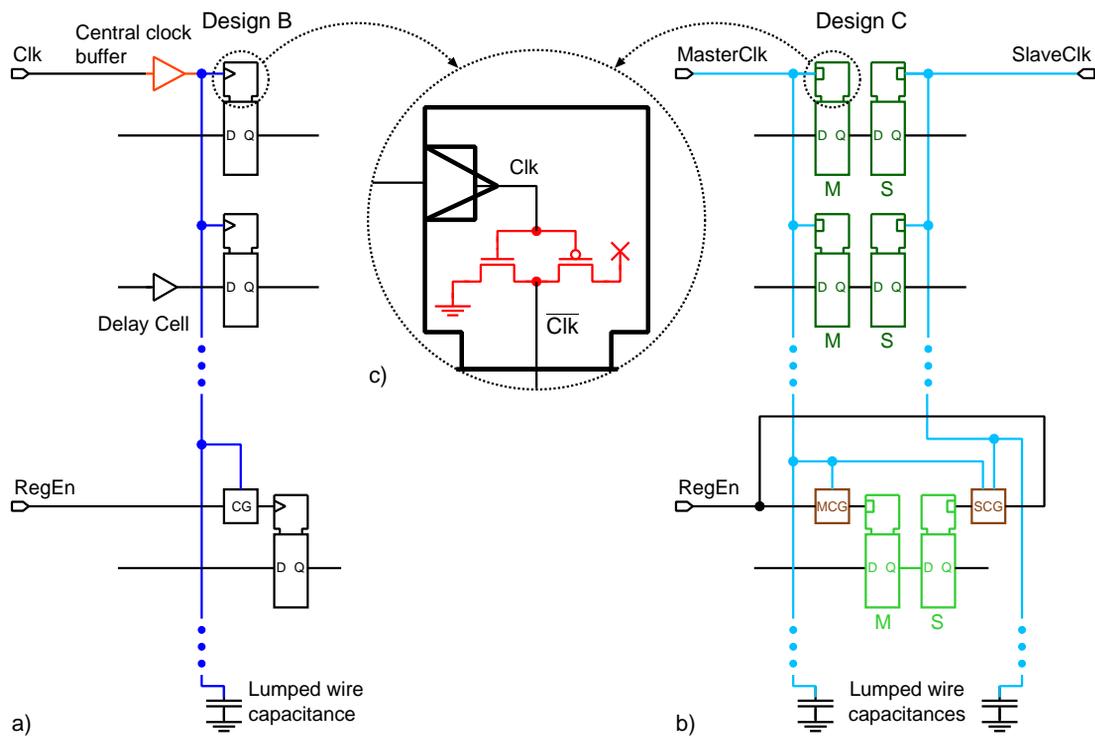


Figure 5.17: Clock network differences of designs **B** (a) and **C** (b). Clock input inverter of the used flip-flops and latches (c).

designs, the combinational logic is based on the same VHDL code and therefore can be expected to show the same power consumption. Moreover, the dissipation of the combinational logic does not depend on whether single-edge-triggered one-phase clocking – as in design **B** – or *unsymmetric* level-sensitive two-phase clocking – as in design **C** – is employed. Hence, the behavior of the “sequential energy” is actually the reason for the cross-points of the curves in the plots in Figs. 5.18 and 5.19.

#### 5.2.1.4 Analog Simulations of a Flip-Flop, Latch Pair, and Inverter

In order to find the main source of power dissipation, circuit simulations for a single D-FF of **B** and a latch pair of **C** have been performed with Spectre by Cadence. They revealed that up to 80% of the energy of the sequential cells is dissipated in the inverter at the clock input. This inverter is shown in magnified detail in Fig. 5.17c (red). In Fig. 5.22a,

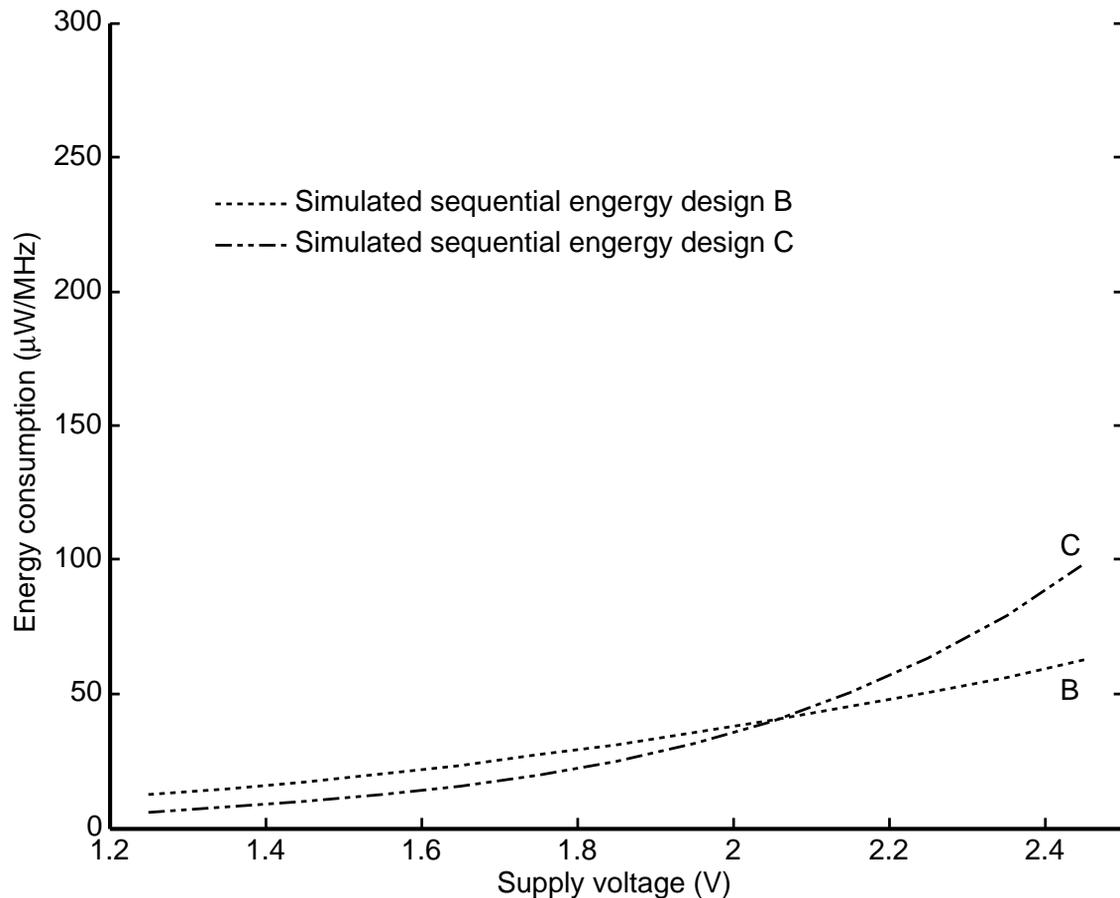


Figure 5.18: Simulated sequential energy dissipation of designs **B** and **C** as function of supply voltage.

which shows the schematic of the original latch from the cell library, this inverter has been colored red again. The energy consumption of these inverters as a function of the supply voltage is plotted in Fig. 5.20 (solid and dashed curves).

The difference between the power consumption of design **B** and **C** is mainly caused by the clock gates (brown in Fig. 5.17b) and the latches that are not gated (dark green) in design **C**. In this design, the clock input inverters inside the gated latches are driven by a clock gate. These inverters do not suffer from slow clock transitions and cross-over currents, since clock gates (brown) restore the clock waveform and drive only a limited number of sequential cells (light green). Nevertheless, the lack of a central buffer in design **C** makes the clock transition times at ungated latches (dark green) and at clock

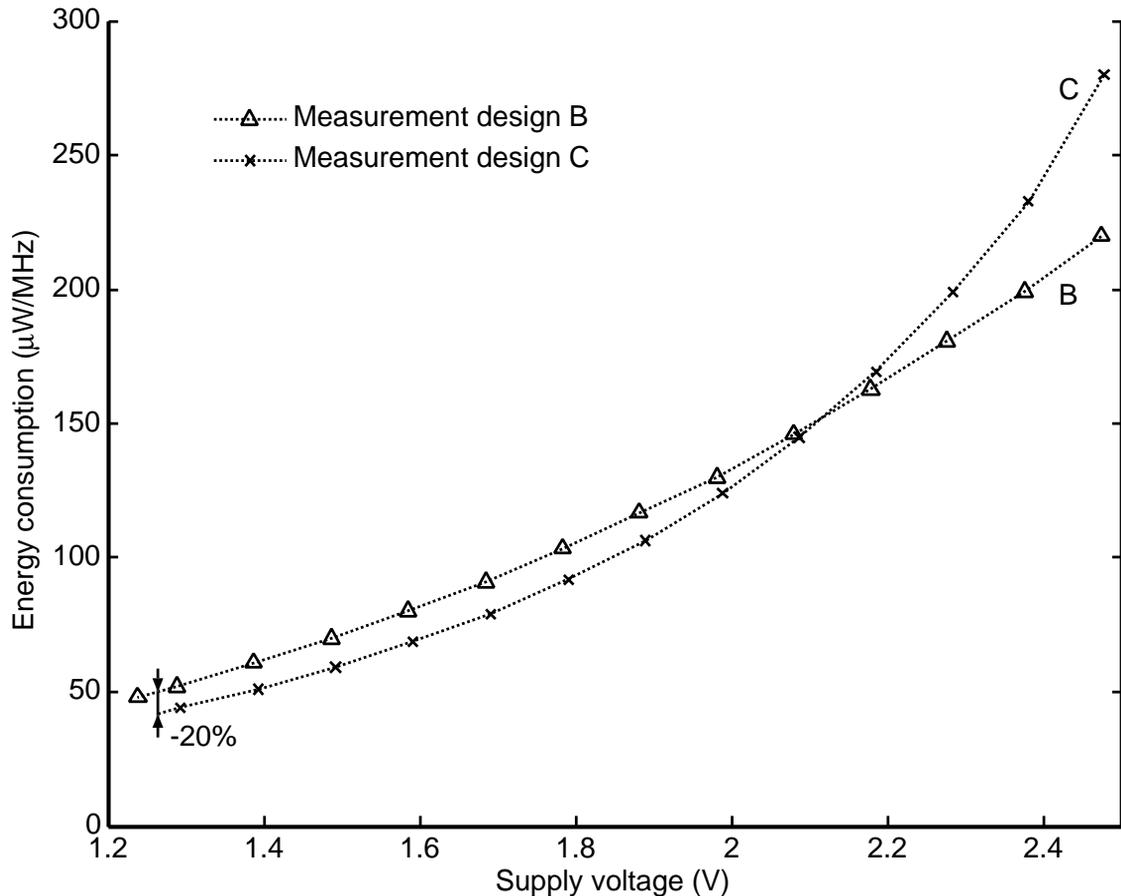


Figure 5.19: Measured energy dissipation of designs **B** and **C** as function of supply voltage.

gates (brown) extremely high (light blue nets in Fig. 5.17b): around 8.5 ns for the master clock at 2.5 V. This is more than ten times larger than in design **B**, where the clock waveform is restored by a central clock buffer (orange in Fig. 5.17a). While the non-overlapping phases in design **C** provide ample margins against malfunctioning, the lack of clock tree buffers causes much energy to be dissipated by cross-over currents in the clock input inverter inside ungated latches and clock gates itself.

### 5.2.1.5 Approximation with the $\alpha$ -Power Transistor Model

To confirm the assumption about large cross-over currents caused by slow clock transitions (small slew rates), calculations using the

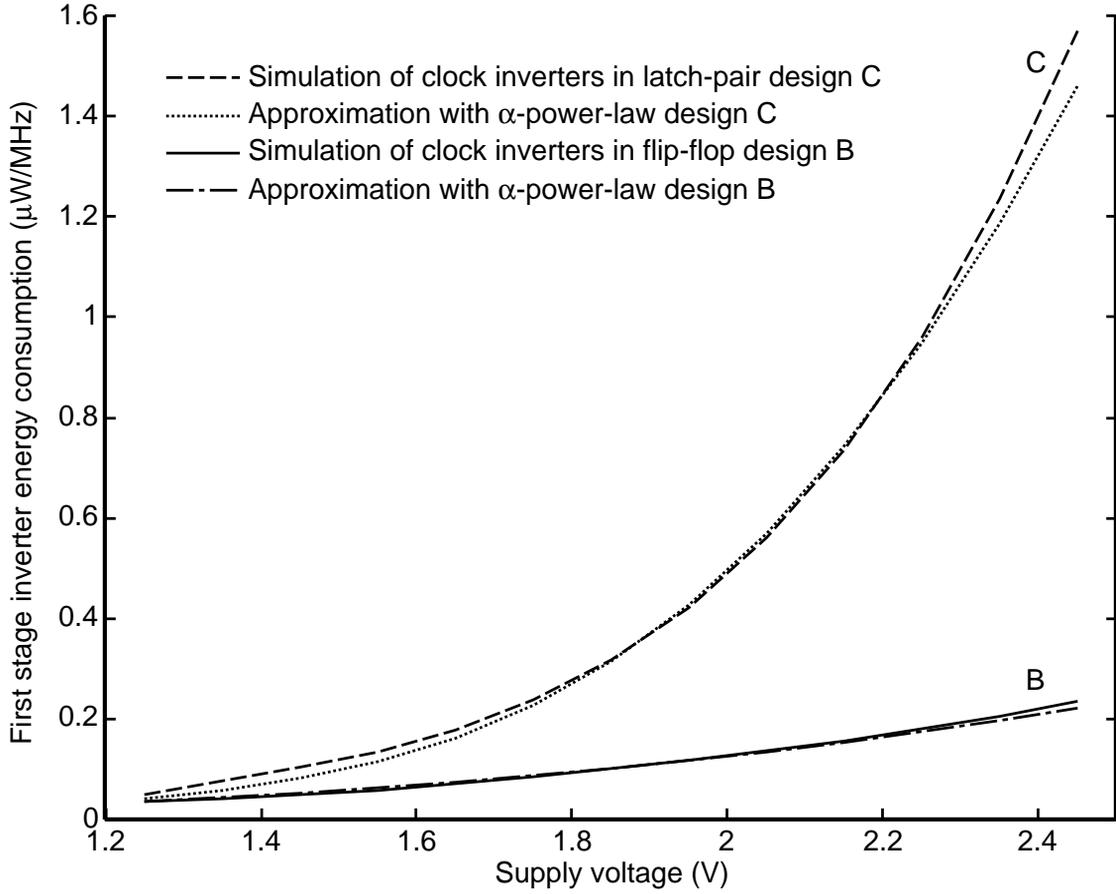


Figure 5.20: Simulation and  $\alpha$ -power-law approximation of the energy consumption inside the first clock inverter of a flip-flop (in design **B**) and the both first clock inverters in the corresponding latch pair (in design **C**).

$\alpha$ -power transistor model [45] have been carried out. This model has been calibrated to the nMOS and pMOS current-voltage characteristics of the transistors employed in the standard cells. The cross-over or short-circuit energy dissipated in a CMOS inverter that discharges a small output capacitance can be expressed as (compare with Eq. 2.29):

$$E_{SC} \Big|_{t_T \gg \tau_N} = V_{DD} t_T I_{D0p} \frac{1}{\alpha_p + 1} \frac{1}{2^{\alpha_p}} \frac{(1 - v_{tn} - v_{tp})^{\alpha_p + 1}}{(1 - v_{tp})^{\alpha_p}}. \quad (5.12)$$

The threshold voltages have been normalized to the supply voltage:  $V_{TH, nMOS}/V_{DD} = v_{tn}$ ,  $V_{TH, pMOS}/V_{DD} = v_{tp}$ . The parameter  $\alpha_p$

models the velocity saturation of a pMOS transistor. The other constituents of the formula correspond to the one in Eq. 2.29.

The dotted and dash-dotted curves in Fig. 5.20 show good agreement of this simple transistor model with the more precise analog simulations. Two conclusions can be drawn from Eq. 5.12:

- $E_{SC}$  increases linearly with the input transition time  $t_T$ .
- $E_{SC}$  increases more than quadratically with the supply voltage since the dependency of  $I_{D0p}$  is progressively dependent on  $V_{DD}$ .

Slow clock transitions as in design **C** in conjunction with high supply voltages greatly increase the contribution of short-circuit currents to the overall energy dissipation.

#### 5.2.1.6 Maximum Operating Frequency at Low Supply Voltages

Figure 5.21 shows the minimum measured power that is necessary to operate designs **B** (solid) and **C** (dashed) at a given frequency. The points along the curves specify the supply voltages that are needed to reach the corresponding frequency value on the x-axis. These data points have been read out from a Shmoo plot of the corresponding device [62]. For instance, in order to operate design **C** at 10.5 MHz, a minimum supply voltage of 1.35 V is required to fulfill the timing conditions, where it dissipates 1.01 mW. Both designs functionally fail at supply voltages below 1.25 V. At this voltage, design **B** can be operated at clock frequencies of up to 16.7 MHz, whereas **C** already fails at clock frequencies in excess of 6.7 MHz.

As can be seen in Fig. 5.21, there exists a frequency above which the two-phase clocked design dissipates more than the one-phase equivalent. According to the figure, design **C** is more energy efficient only up to 10.65 MHz; for higher frequencies, it is necessary to increase the supply voltage. For design **C**, 1.4 V is required, compared to 1.25 V for design **B**, and the design is dissipating more energy.

#### 5.2.1.7 Original Latch Circuit from the Cell Library

The simulation results and their approximations in Fig. 5.20 have shown that the cross-over power consumption of the clock inverter inside each latch in design **C** is responsible for a large fraction of the

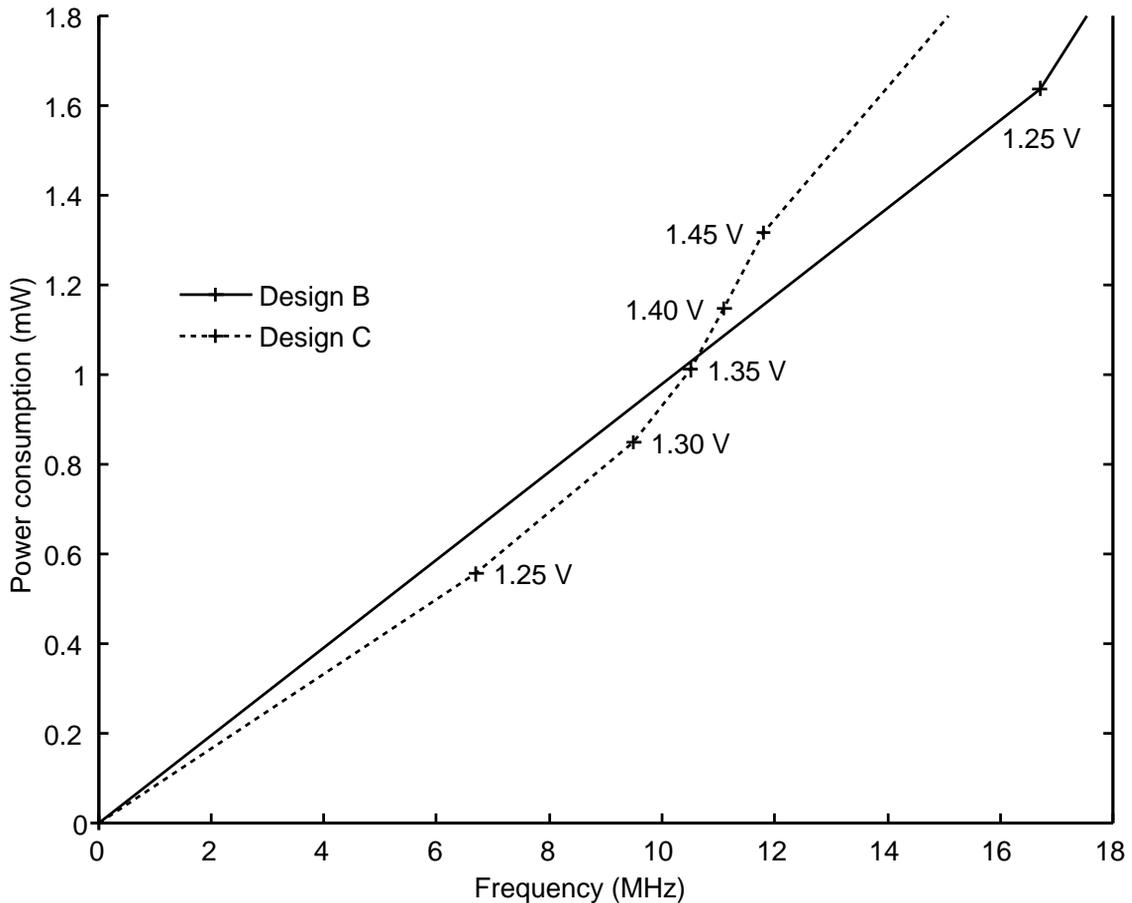


Figure 5.21: Minimum measured power consumption to operate designs **B** (solid) and **C** (dashed) at a given frequency, with the corresponding required minimum supply voltage.

overall consumption. Figure 5.22a shows the schematic of such a latch. The basic structure of this latch is a switched memory loop [43]. If the clock is not set ( $\text{Clk} = 0$ ), the feedback loop is closed. It consists of the brown inverter and the enabled clocked inverter<sup>3</sup> (dark green). The other clocked inverter (light green) is in high-impedance mode, which means that the data input ( $\text{D}$ ) is decoupled from the loop. The latched value is captured in the loop and present at the outputs ( $\text{Q}$  and  $\overline{\text{Q}}$ ). If the clock is set, the feedback loop is open because the dark green inverter is in a high-impedance state. The light green clocked

<sup>3</sup>Clocked inverters – also referred to as controlled inverters [43] – are inverters whose output can be disabled or enabled by a dedicated signal, such as a clock.



a clock inversion. Figure 5.22b shows the new clock-slew insensitive latch (CSILatch) circuit. Its fundamental structure is still the one of a switched memory loop. Yet, in order to keep wide margins of reliability, the following alternatives have been rejected:

1. All dynamic and semi-static solutions [63].
2. All ratioed schemes, in which the functionality itself depends on transistor sizing.

The CSILatch in Fig. 5.22b is basically a 6-transistor latch (light green, dark green, and brown) [63]. An output inverter (2 transistors, blue) has been added. This prevents pulses from back-propagating from the output pin  $Q$  to the internal node  $\textcircled{Z}$ . Such pulses might undesirably toggle the latch. An asynchronous reset facility (2 transistors, violet) has been added in order to keep compatibility with the original latch from the cell library. In the two circuits in Fig. 5.22, functionally corresponding circuit elements share the same color. The CSILatch counts 10 transistors against the 21 of the original latch, occupying roughly half of the area. No inverted output is included because in unsymmetric two-phase clocking, at least half of the latches, namely the master-clocked ones, are directly connected via the  $Q$  output to the slave-clocked ones. In order to emulate a  $\overline{Q}$  output, an external low-drive-strength inverter is added by the hardware synthesizer solely if required.

### 5.2.1.9 Comparison of the Two Latch Circuits

By means of Tabs. 5.2 and 5.3 the energy dissipation of the original latch circuit from the cell library (Fig. 5.22a) can be compared with the one of the proposed CSILatch (Fig. 5.22b). The figures in the tables indicate the dissipated energy if one of the latch inputs undergoes a rising ( $\nearrow$ ) or falling ( $\searrow$ ) transition, while the other inputs keep their value (logic ‘1’ or ‘0’). Only the six cases for which the reset is not asserted ( $\overline{\text{Reset}} = 1$ ) have been considered because only those are relevant during normal operation of the latch circuits.

For both latches, these six cases can be grouped into four classes according to the transition type and the corresponding dissipated energy. The first class, equivalent to the first case ( $\textcircled{1}$ ), is given by a rising or falling transition at the data input  $D$  while the latch is not transparent ( $\overline{\text{Clk}} = 0$ ). The output keeps its value ( $Q_n = Q_{n-1}$ ). The

Table 5.2: Energy consumption per input transition of the original latch (Fig. 5.22a) for different cases. The transition time of the toggling input pin amounts to 1.8 ns. The output pins  $Q$  and  $\bar{Q}$  are loaded with a small capacitance of 0.01 fF. The supply voltage amounts to 2.5 V.

Transition	Condition	Energy per	
		Rise ( $\nearrow$ )	Fall ( $\searrow$ )
		Transition	
		[pJ]	
① $D \nearrow / \searrow, (Q_n = Q_{n-1})$	$\text{Clk} = 0, \overline{\text{Reset}} = 1$	0.015	0.012
② $\text{Clk} \nearrow / \searrow, (Q_n = Q_{n-1})$	$D = 0, \overline{\text{Reset}} = 1$	0.307	0.378
③ $\text{Clk} \nearrow / \searrow, (Q_n = Q_{n-1})$	$D = 1, \overline{\text{Reset}} = 1$	0.306	0.380
④ $D \nearrow / \searrow \Rightarrow Q \nearrow / \searrow$	$\text{Clk} = 1, \overline{\text{Reset}} = 1$	0.401	0.489
⑤ $\text{Clk} \nearrow \Rightarrow Q \searrow$	$D = 0, \overline{\text{Reset}} = 1$	-	0.545
⑥ $\text{Clk} \nearrow \Rightarrow Q \nearrow$	$D = 1, \overline{\text{Reset}} = 1$	0.506	-

dissipated energy is very small for both latch types. Especially, the CSILatch dissipates nearly no energy in this case.

The second class (cases ② and ③) is given by a clock transition, while the data input is at logic ‘1’ or ‘0’ ( $D = 1$  or  $D = 0$ ) and the output does not toggle. For the original latch, the two inverters (red and brown in Fig. 5.22a) are toggling, while the rest of the circuitry keeps its value. This yields an energy dissipation of medium size. For the CSILatch, only the two transistors whose gates are connected to the clock input are changing their state. Again, the rest of the circuit keeps its state. This results in a still negligible energy consumption of a few picojoules.

The third and fourth class (cases ④, ⑤, and ⑥) represent the cases where the output toggles. In case ④, the transition of the output is caused by a transition at the data input while the latch is transparent ( $\text{Clk} = 1$ ). In cases ⑤ and ⑥, the output toggles because the latch gets transparent (rising edge of the clock) and just before this transition, the input  $D$  and the output  $Q$  have opposite values. Therefore, in case ⑤, only the falling edge of the output can be considered and in case

Table 5.3: Energy consumption per input transition of the CSILatch (Fig. 5.22b) for different cases. The transition time of the toggling input pin amounts to 1.8 ns. The output pin  $Q$  is loaded with a small capacitance of 0.01 fF. The supply voltage amounts to 2.5 V.

Transition	Condition	Energy per Transition [pJ]	
		Rise ( $\nearrow$ )	Fall ( $\searrow$ )
① $D \nearrow / \searrow, (Q_n = Q_{n-1})$	$\text{Clk} = 0, \overline{\text{Reset}} = 1$	0.000	0.000
② $\text{Clk} \nearrow / \searrow, (Q_n = Q_{n-1})$	$D = 0, \overline{\text{Reset}} = 1$	0.001	0.001
③ $\text{Clk} \nearrow / \searrow, (Q_n = Q_{n-1})$	$D = 1, \overline{\text{Reset}} = 1$	0.004	0.004
④ $D \nearrow / \searrow \Rightarrow Q \nearrow / \searrow$	$\text{Clk} = 1, \overline{\text{Reset}} = 1$	0.338	0.229
⑤ $\text{Clk} \nearrow \Rightarrow Q \searrow$	$D = 0, \overline{\text{Reset}} = 1$	-	0.145
⑥ $\text{Clk} \nearrow \Rightarrow Q \nearrow$	$D = 1, \overline{\text{Reset}} = 1$	0.355	-

⑥ only the rising one<sup>4</sup>. Obviously, these cases dissipate most energy because almost all transistors inside the latches are involved. Again, the CSILatch is more efficient than the original latch.

For all cases, the proposed CSILatch is more power efficient than the original latch. Especially for the cases where the output keeps its value (①, ②, and ③), the CSILatch dissipates almost no energy because no inverter has to be charged or discharged. The cell area of the original latch amounts to  $118.8 \mu\text{m}^2$ , whereas the one of the CSILatch is only  $63.36 \mu\text{m}^2$ .

#### 5.2.1.10 Simulation Results of FIR Filter I

By substituting the new slew-insensitive latch-circuit (CSILatch) for the original one in design **C**, a more energy efficient design **D** has been derived. Additional characteristics of this simulated design can be found in the last column of Tab. 5.1. Spectre simulations confirm the functionality and the energy savings of the new latch design. An

<sup>4</sup>The other clock edges would result in cases ② and ③.

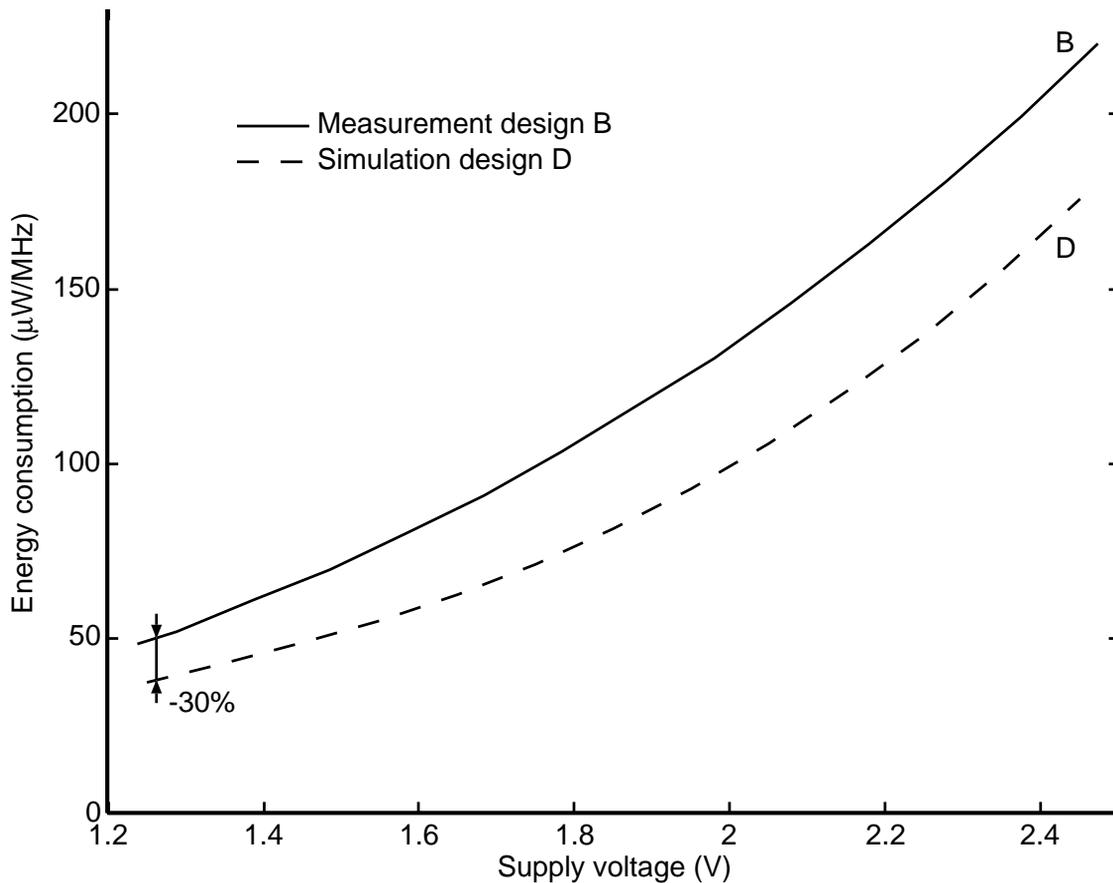


Figure 5.23: Simulated energy consumption of design **D** (dashed) compared to the energy consumption of design **B** (solid).

overall energy saving of around 30% at 1.25 V of design **D** compared to **B** can be observed, see Fig. 5.23. Moreover, the new slew-insensitive latch circuit avoids any cross-point if we compare Fig. 5.19 to Fig. 5.23.

#### 5.2.1.11 Conclusions of FIR Filter I

Relaxed hold-time constraints make clock buffers and delay cells superfluous in level-sensitive two-phase clocked designs. However, in such clock-buffer-less designs, the lower clock-slew rate tends to negate the low-power feature of level-sensitive clocking. The reason is an increased power consumption in ungated latches and clock gates. These cells suffer from an increased cross-over power consumption because of the low slew rates. At higher supply voltages this effect is worse.

At very low supply voltages however, this effect does not cancel out the advantages of level-sensitive clocking. Therefore, in the reported circuit that uses traditional latches, 20% energy could be saved at 1.25 V. Above a certain supply voltage (2.1 V), this circuit consumes more power than the single-edge-triggered design containing a single central clock buffer.

A simple voltage scaling of a design for nominal supply voltage leads to non-optimum solutions regarding energy consumption. As a matter of fact, when limiting the voltage to above 2.5 V in the analysis reported in Figs. 5.16 and 5.19, the better choice would have been to go for the single-edge-triggered design **B**. At 1.25 V supply voltage, however, both designs are still meeting the requested data throughput, and the clock-buffer-less and level-sensitive design **C** dissipates 20% less energy.

In buffer-less level-sensitive two-phase clocked designs, latches tend to become the critical energy sinks. Clock-slew insensitive latches in level-sensitive two-phase clocked designs that contain no clock buffers preserve the energy advantage over single-edge-triggered one-phase clocked designs up to high frequencies. In the considered filter application a 30% lower energy consumption than a clock-gated single-edge-triggered one-phase-clocked design could be observed by simulation.

## 5.2.2 FIR Filter II

A second test object that serves as a vehicle for experimenting with various clocking approaches is the input stage of the differential microphone algorithm presented in Sec. 2.1.4.1. Several ideas presented in the preceding section (Sec. 5.2.1) are resumed and improved. The new, clock-slew insensitive CSILatch has been integrated into one of the fabricated implementation alternatives.

### 5.2.2.1 Architecture and Chip Design of FIR Filter II

This input stage or front end (dash-dotted frame in Fig. 2.6), which calculates the two cardioids  $C_f$  and  $C_b$ , has been integrated. Apart from other blocks, this front end contains two identical FIR filters of order 17. In order to meet area constraints, they have been implemented in a time-shared and iteratively decomposed architecture.

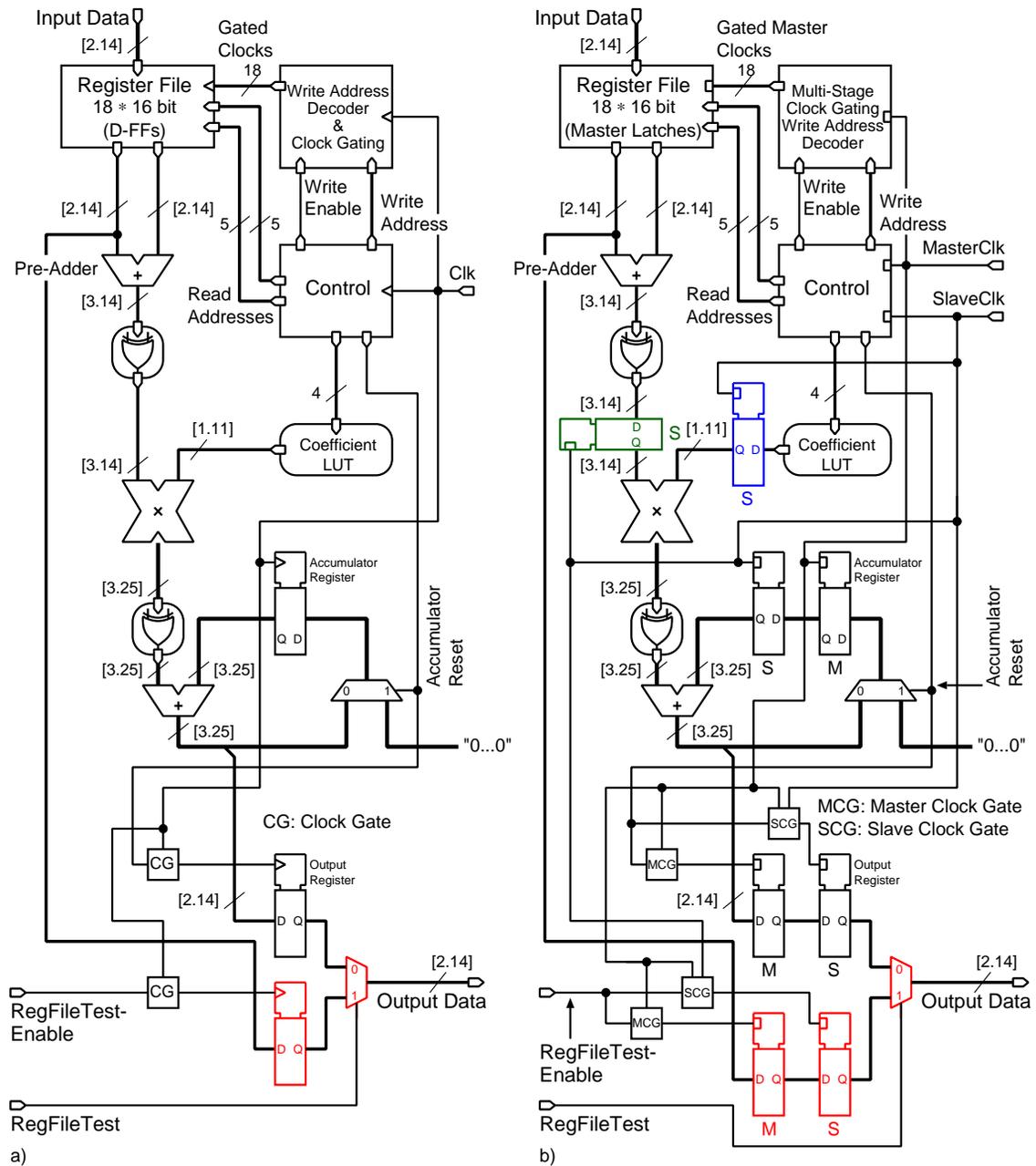


Figure 5.24: Block diagrams of the integrated hybrid FIR filters. a) Single-edge-triggered one-phase clocking in variant A, b) Mixed unsymmetric-symmetric level-sensitive two-phase clocking in variants B and C.

The hardware implementation on silicon in a 0.25  $\mu\text{m}$  CMOS process contains three functionally equivalent versions of the front end referred to as variants **A**, **B**, and **C**. All these variants make use of the hybrid number format. They differ only in the clocking strategy, clock gating style and the employed latches. Table 5.4 gives an overview of these three variants. A block diagram of the reference single-edge-triggered one-phase filter, denoted as variant **A**, is shown in Fig. 5.24a. This reference contains traditional clock gating introduced by Synopsys Design Compiler. Apart from the flip-flops in the register file, all D-FF are part of a scan chain. In order to test the register file, one of its output can be read out via a register and a multiplexer (both red) to the filter output. In the figure, sequential blocks are rectangular, whereas combinational blocks are boxes with rounded corners.

The optimized variants **B** and **C** employ *mixed* unsymmetric-symmetric level-sensitive two-phase clocking, and, they contain the more advanced multi-stage clock gating technique for the memory decoder as described in Sec. 5.2.2.2. Their identical block diagram is depicted in Fig. 5.24b. The purpose of the green and red colored latch banks will be explained in Sec. 5.2.2.3. How these latches have been added to the scan chain will be described in Sec. 5.2.2.4. The register file, which contains latches, follows the same testing strategy (red) as the one in the reference design in Fig. 5.24a. Apart from the latches in this register file, all other latches are member of a scan chain.

Synthesis has been performed by Synopsys Design Compiler whereas Silicon Ensemble by Cadence has been used for placement and routing of the final chip layout. The three version can be independently measured thanks to distinct power supplies.

### 5.2.2.2 Multi-Stage Clock Gating

Before actually explaining “multi-stage clock gating”, some comments about terminology are appropriate here. In the literature (e.g., [51]), concepts such as “hierarchical gating” are used. The documentation of the Power Compiler<sup>TM</sup>, a part of Synopsys Design Compiler<sup>®</sup>, additionally explains a technology called “multi-stage clock gating” [48]. Sometimes it is not really clear whether the two expressions – “multi-stage clock gating” and “hierarchical gating” – refer to the same gating

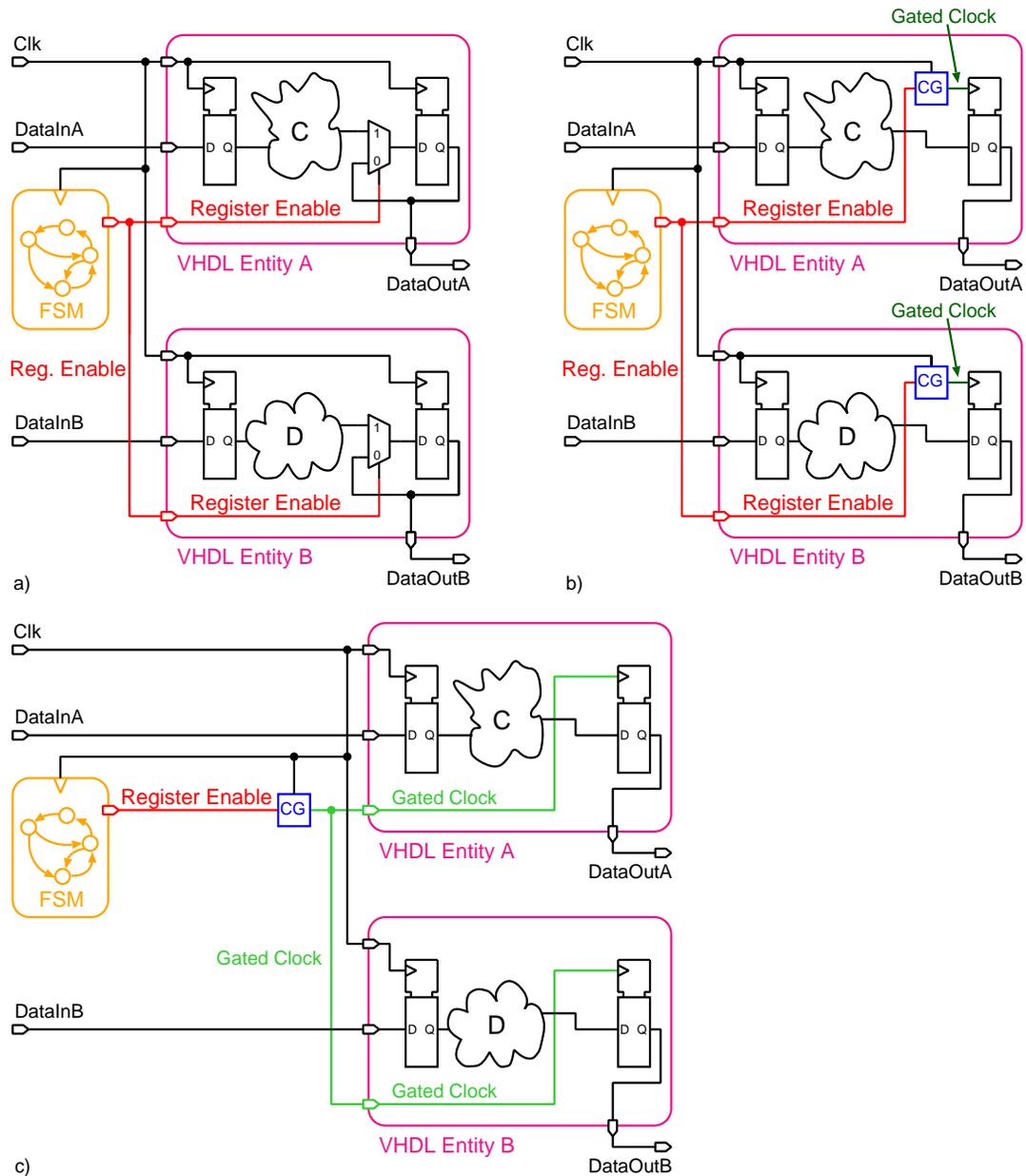


Figure 5.25: Hierarchical clock gating: a) original ungated circuit, b) circuit with traditional clock gating, c) circuit with hierarchical clock gating.

Table 5.4: Characteristics of the three FIR Filter II variants.

Design Characteristic	Variant <b>A</b> (Reference)	Variant <b>B</b>	Variant <b>C</b>
Clocking Strategy	SETOPC	MLSTPC	
D-FF/Latches	D-FF (Std. Lib.)	Latches (Std. Lib.)	CSILatch (Fig. 5.22b)
Architecture	Fig. 5.24a	Fig. 5.24b	
Number Format	Hybrid		
Clock Gating	Yes		
Clock Gating Style in Address Decoder	Traditional (Synopsys)	Multi-Stage (Sec. 5.2.2.2 and Fig. 5.27)	
Testability	Register File: Test Access All other Registers: Scan Chain		

technique. The following discussion strictly adheres to Synopsys terminology.

In Fig. 5.25c hierarchical clock gating is explained. The original, ungated circuit is shown in Fig. 5.25a. It can be seen that the register enable signal (red), which is created in an finite state machine (orange), is used in two sub-blocks (VHDL Entity **A** and **B**, both pink). In case of traditional clock gating as in Fig. 5.25b, one clock gate (**CG**, blue) per sub-block is instantiated. These gates derive gated clocks (dark green) that are locally distributed within these sub-blocks. If hierarchical clock gating is employed, as shown in Fig. 5.25c, one single clock gate for the common enable signal is instantiated at an upper level of hierarchy. Instead of the enable signal, the related gated clock (light green) is distributed to the lower-level sub-blocks (VHDL Entity **A** and **B**). In other words, common enable signals that are shared across registers in different sub-blocks are extracted and then a shared clock gating cell is introduced [48]. Thereby, the number of clock gates is reduced. However, this technique has not been applied in the FIR Filter in Fig. 5.24 for the simple reason that there are no such shared enable signals.

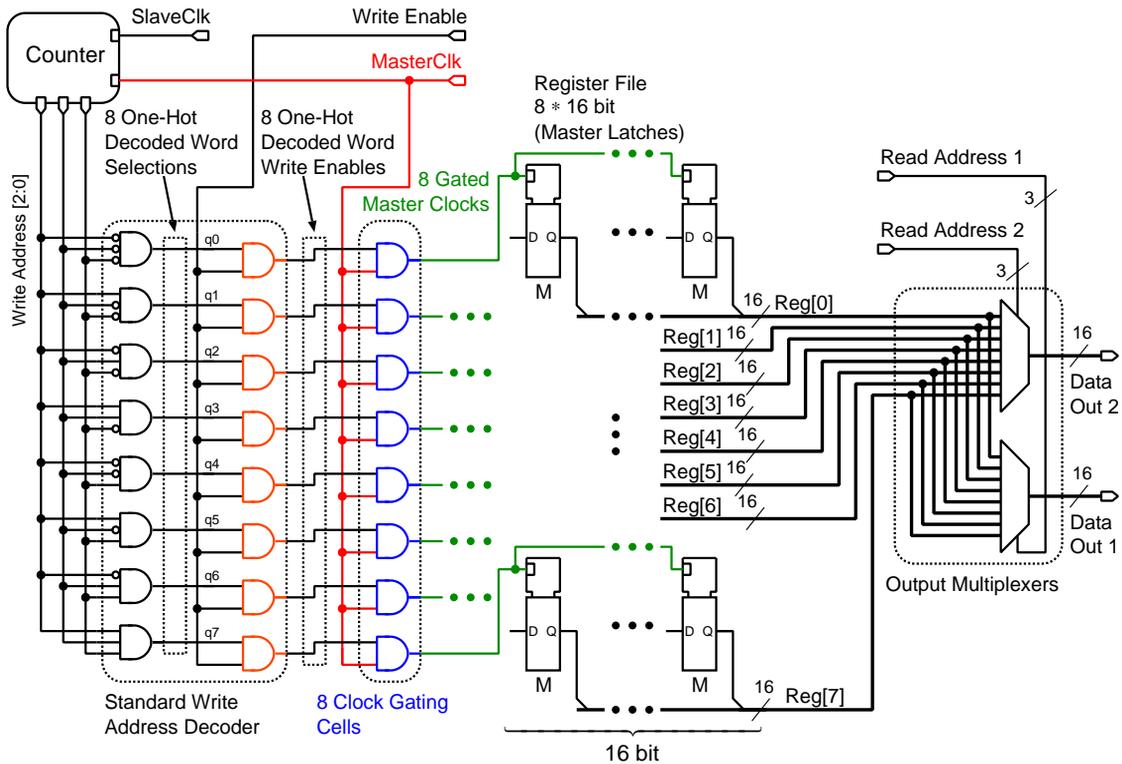


Figure 5.26: Traditional clock gating in a register file employing level-sensitive two-phase clocking.

Note that although the example in Fig. 5.25b makes use of single-edge-triggered one-phase clocking, this technique can be combined with any other clocking style that allows clock gating. However, in SETOPC designs it can be automatically introduced by means of modern design tools [48], whereas in all other clocking styles manual VHDL recoding is probably required.

Contrary to hierarchical clock gating, multi-stage clock gating has been employed in the register file in Fig. 5.24b. Writing into this memory in a circular fashion, which is typical for FIR filter applications, involves a counter and an address decoder. Figure 5.26 depicts such a latch-based and hence level-sensitive register file that still makes use of traditional clock gating. In order to simplify the schematic drawing, its size has been reduced to 8 data words. This contrasts with the register file size in Fig. 5.24b that contains 18 words. However, the principle of address decoding and clock gating remains unaltered. Based on the write address, the decoder in Fig. 5.26 creates the 8

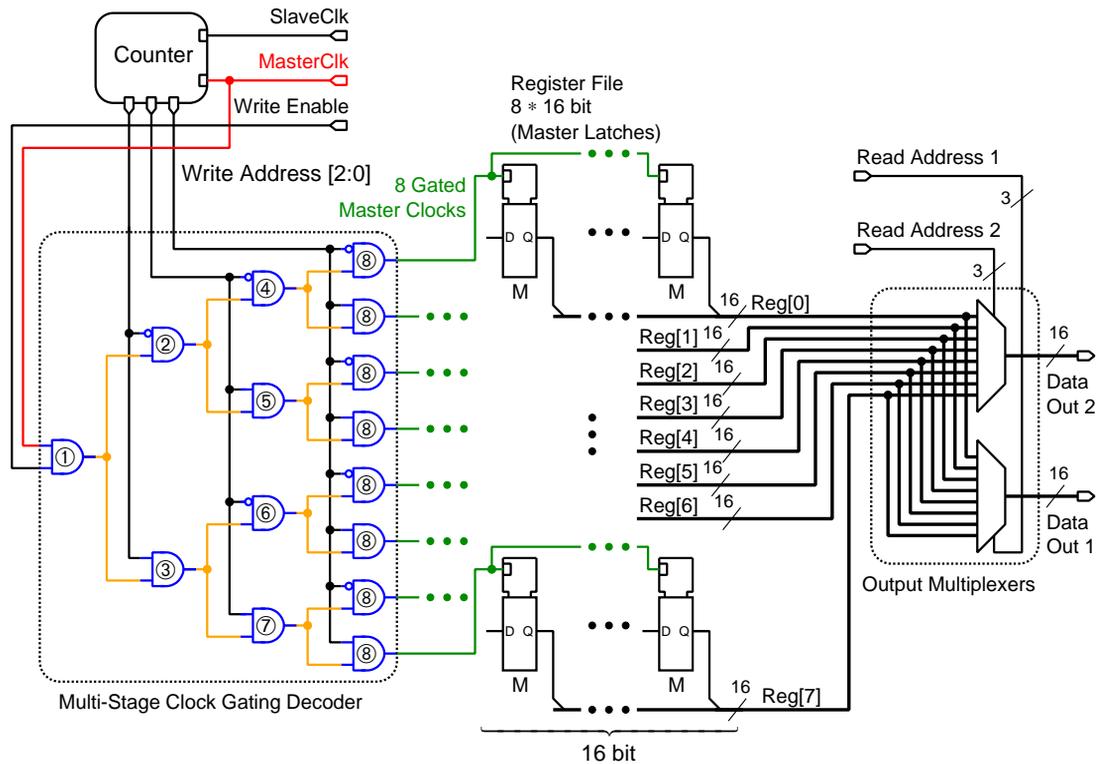


Figure 5.27: Multi-stage clock gating in a register file employing level-sensitive two-phase clocking.

one-hot decoded word selection signals that each address exactly one register file location. By means of **AND** gates (orange), the one-hot decoded word write enable signals are created. Eventually, the blue clock gates derive the 8 gated clocks (green) that drive the same number of latch banks. In this level-sensitive example, these clock gates are **AND** gates according to Fig. 5.7b in Sec. 5.1.3.

Especially in larger decoders, which require more clock gates, all these gates (blue in Fig. 5.26) represent a large capacitive load on the (master) clock net (red). The idea of multi-stage clock gating is to identify common sub-conditions of the enable signals and thereby minimize toggle activity of heavily loaded nets. As can be seen in Fig. 5.27, the high-activity clock net (red) is gated (①) by the write enable signal at the root of a tree of clock gates (blue). This creates a first gated clock, which drives two clock gates (②, ③) that in turn drive another two clock gates each (④ – ⑦), and so on (⑧). Each gated clock is derived from another gated clock (all orange) by clock gates

that are cascaded hierarchically to minimize useless toggling of heavily loaded nodes.

The relevant quantity is the *activity*  $\times$  *capacitance* product, which is summed up over all circuit nodes driven by the clock. This sum of products will be referred to as “energy-effective capacitance”. An estimate for uninterrupted memory writing (write enable permanently active) for the traditional address decoder as in Fig. 5.26 is

$$C_1 = 4n \cdot C_0 \quad (5.13)$$

where  $n$  is the number of data words and  $C_0$  is the capacitance of an **AND** gate input. A detailed derivation of  $C_1$  can be found in Appendix B. Applying the multi-stage clock gating technique, the overall energy-effective capacitance  $C_2$  becomes:

$$C_2 = \left( 4 \cdot \frac{n^2 - 1}{3n} + 4 \log_2 n + 2 \right) \cdot C_0, \quad (5.14)$$

and the reduction in dynamic energy due to multi-stage clock gating can be expressed as follows:

$$\frac{\Delta E}{E_1} = \frac{C_1 - C_2}{C_1} \xrightarrow{n \rightarrow \infty} \frac{2}{3}. \quad (5.15)$$

This equation suggests that, as an upper bound, up to 67% of dynamic energy can be saved when the register file grows very large. Moreover, the multi-stage clock gating is  $n$  times more efficient than the traditional flat clock gating when no memory access occurs (write enable remains passive). This happens because the master clock is gated at the root of the address decoder tree by combining the master clock with the write enable in Fig. 5.27 (blue **AND** gate ①). This avoids any unproductive switching activity. Thus, large registers with relatively few write accesses, such as those found in high-order FIR filters, benefit the most from hierarchical clock gating.

### 5.2.2.3 Glitch-Stop Latch Barriers

Comparing the optimized filter architectures of variants **B** and **C** in Fig. 5.24b with the reference (variant **A**) in Fig. 5.24a reveals some additional latch banks (blue and green) at both multiplier inputs. The blue slave latch bank intentionally placed between the combinational lookup table (LUT) and the multiplier coefficient input is functionally not necessary. The purpose of this so-called glitch-stop latch barrier is to shield glitches emanating from the output multiplexer of the coefficient LUT from penetrating into the multiplier. These latches are clocked by the slave clock, because the bistables (green) at the sample input of the multiplier are slave-clocked latches as well.

These green latches at the other input are the slaves of the master latches in the register file, see Fig. 5.24b and Fig. 5.27. In this register file, each of the 18 memory locations<sup>5</sup> consists of a master latch bank, whose width is given by the sample word with (16 bit). Designing this register file in the *unsymmetric* level-sensitive two-phase clocking style would mean to place the corresponding slave latches immediately behind the masters, as shown in Fig. 5.7. However, as shown in Fig. 5.9 and Fig. 5.13, *symmetric* and *mixed* unsymmetric-symmetric level-sensitive two-phase clocking offer the potential to relocate combinational logic between master and slave latches. In Fig. 5.24b this combinational logic corresponds to the two output multiplexers of the register file (Fig. 5.27), the pre-adder, and the number format conversion. In general, the total number of bits at the outputs of multiplexers and adders is smaller than at their inputs. The number of slave latches (green) can therefore be reduced when they are relocated after such combinational logic, as shown in Fig. 5.24b. In this filter, the number of slave latches could be reduced from  $18 \times 16 = 288$ , which is equal to the number of master latches in the register file, to only 17 latches! Additionally, this relocation lets this latch bank act as a glitch-stop barrier. It shields the sample input of the multiplier from glitches emanating from the combinational logic (multiplexers, adder, and number conversion) that is now located between the register file (master latches) and this slave latch bank itself.

---

<sup>5</sup>Note that in the example in Fig. 5.27 the register file contains only 8 memory locations.

Both slave-triggered latch barriers in front of the multiplier are particularly effective: post-layout simulations indicate more than 50% power consumption reduction in this multiplier.

#### 5.2.2.4 Testability of Register Files and Latch Barriers

The glitch-stop latch barriers presented in Sec. 5.2.2.3 cannot be added to a scan chain without precaution. As shown in Fig. 5.8, this test structure requires pairs of latches, i.e., the number of master latches has to match the one of slave latches. This prerequisite is not fulfilled in both latch barriers in Fig. 5.24. Moreover, the same observation holds for the register files of the optimized designs, which contain only master latches. As shown above, their number intentionally does not match the number of slave latches in the brown barrier in Fig. 5.24b. In case of the blue barrier, no master latches are present at all.

Testability of the register file in all three filter variants has been realized by forwarding one register-file output to the filter output via an additional register and multiplexer (both red in Fig. 5.24).

Since the additional slave latches in the two glitch-stop barriers do not have a direct master complement, additional measures need to be taken in order to achieve scan chain testability of these bistables. As shown in Fig. 5.28, “dummy” master latches (upper row, brown) have been placed in front of the actual (slave) latch barrier, which constitutes the lower row. This row corresponds to the blue and green slave latches in Fig. 5.24b. There, dummy latches are not shown for the sake of simplicity.

In Fig. 5.28, the master clock of the dummy latches is gated by means of the test mode signal in order to enable these latches only during scan chain testing. Throughout this test, the test mode signal has to be asserted. When the scan enable signal is active, a multiplexer in front of the slave latches selects the directly preceding master output as slave latch input. In the example in Fig. 5.28, the scan chain (orange) of this barrier starts at the leftmost master latch. The output of this chain is the LSB slave latch output. While in normal operation, e.g., test mode and scan enable are disabled, this scannable latch barrier behaves as the slave latch barriers in Fig. 5.24b. In this case, the data input of this barrier corresponds to one of the latch barrier inputs in

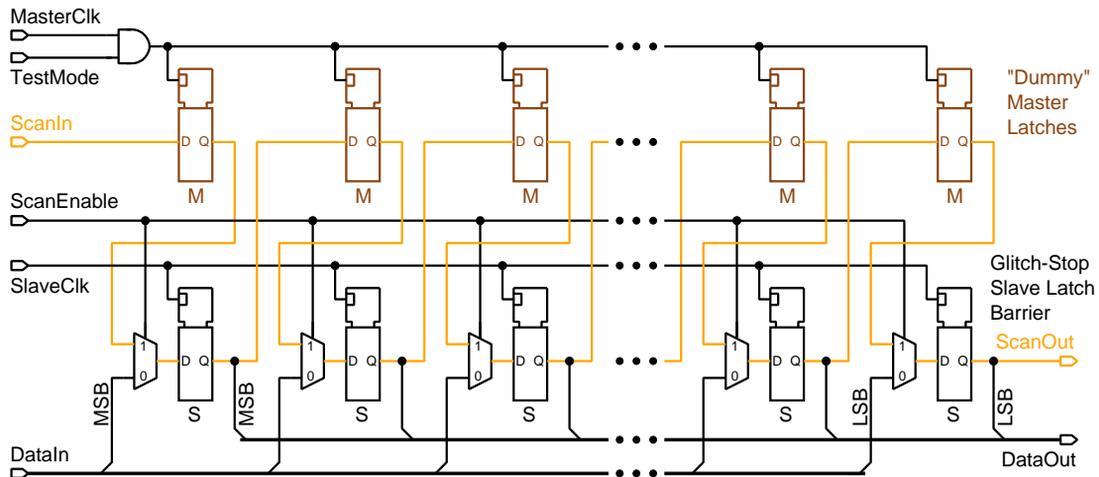


Figure 5.28: Glitch-stop (slave) latch barrier containing master latches in order to build a scan chain.

Fig. 5.24b, whereas the data output corresponds to the outputs which are directly connected to the multiplier inputs of the filter.

### 5.2.2.5 Measurements of FIR Filter II

The measurement results presented in in Tab. 5.5 confirm the considerable efficiency of the proposed glitch-aware, mixed unsymmetric-symmetric level-sensitive two-phase clocking. These enhancements have been reached without any additional (pipeline) registers, nor by inflating the latency of the design. Table 5.5 shows that more than 30% energy reduction is possible, even without redesigned library cells. The new low-power CSILatch leads to the final 42% energy savings.

### 5.2.2.6 Conclusions of FIR Filter II

A lot of energy is wasted by glitch activities in the datapath of digital filters and in their address decoders for circular memory access. This situation is common to many digital signal processing applications.

Even with clock gating, a typical front-end block of a digital adaptive directional microphone for hearing aids is expected to dissipate about  $34\ \mu\text{W}$  at its regular clock frequency of  $374\ \text{kHz}$ . An alternative that does with an overall power dissipation of  $20\ \mu\text{W}$  has been designed, fabricated and measured. The superior energy efficiency has been

Table 5.5: Measured energy (E) savings of optimized design variants **B** and **C** compared with the SET one-phase reference design variant **A** ( $V_{DD} = 1.4\text{ V}$ ).

Design	Meas. E ( $\mu\text{W}/\text{MHz}$ )	Meas. $\Delta\text{E}/\text{E}$ (%)
Variant <b>A</b>	90	0
Variant <b>B</b>	60	-33
Variant <b>C</b>	52	-42

obtained by combining mixed unsymmetric-symmetric level-sensitive two-phase clocking with multi-stage clock gating and by using a new low-power clock-slew insensitive latch. Additionally, glitch-stop latch barriers placed in front of combinational logic effectively mitigate excessive glitch transmission.

### 5.2.3 Model Hearing Aid System

In order to study the effects of different low-power clocking schemes and cell redesign on the power consumption of a larger and more representative object, the hearing aid system introduced in Sec. 2.1.4.3 has been integrated in two variants.

Section 5.2.3.1 explains the architecture development and chip design of the two variants. Both are based on the same architecture, i.e., they make use of the same level of resource sharing. Section 5.2.3.2 explains the redesign and optimization of cells which have been used in the second variant. The first model hearing aid system variant, which is based on single-edge-triggered one-phase clocking, is presented in 5.2.3.3, it makes use of standard cells only and serves as a reference. The second variant is a mixed unsymmetric-symmetric level-sensitive two-phase clocking design, it contains optimized latches. This level-sensitive variant will be explained in Sec. 5.2.3.4.

### 5.2.3.1 Architecture and Chip Design

Both design variants have been integrated on the same multi-project waver run. The total available silicon area was a square with an edge length of 5 mm. This area was quartered resulting in four dies that have been separately bonded and packaged in four packages. Note that not all of those chips are discussed in this work. Additionally, the area on such a die has been further subdivided to incorporate more than one hearing aid system variant. Individual power rings enable separate power measurements of these subcircuits, although they share the same pins for inputs and outputs. The two variants share their die area with circuitry not discussed in this work.

Since an isomorphic architecture of the hearing aid system would have led to a suboptimal solution, as observed in Sec. 3.4, such an architecture has not been chosen. As was also seen in Sec. 3.4, a fully iteratively decomposed, and hence area-optimized hardware layout, would have resulted in a non-optimum solution in terms of power consumption. Instead, a solution has been chosen which represents a good compromise between low power consumption and costly silicon area.

Taking a closer look at Figs. 5.30 (gray dashed block) and 5.29<sup>6</sup>, reveals that the Elko algorithm and the preceding power equalization share similar functional blocks. Figure 5.31 shows the implemented hardware architecture that combines the signal power equalization and Elko's differential microphone. In this figure, all arithmetic units have been colored as the corresponding units in Figs. 5.30 and 2.7.

The normalization factor  $\langle C_b^2[t] \rangle$  (Eq. 2.6 and red dashed block in Fig. 5.30) is calculated exactly the same way as the average signal power  $S_f^{*2}[t]$  and  $S_b^{*2}[t]$  marked with the red dashed boxes in Fig. 5.29, hence, one multiplier and two adders are foreseen to perform these operations. The multiplication by  $\varepsilon$  has been replaced by a simple shift operation. In Fig. 5.31, these arithmetic units have been colored red and blue. The updating of the coefficients  $\beta_t$ ,  $\alpha_b$ ,  $\alpha_f$  and their multiplication with  $C_b$ ,  $S_b$  and  $S_f$  is executed by means of the brown adder and violet multiplier. Finally, the subtraction at the Elko output, resulting in the signal  $S_x$ , has been merged with the subtraction of the signal power  $\langle S_f^{*2}[t] \rangle$  and  $\langle S_b^{*2}[t] \rangle$ , the corresponding hardware unit

---

<sup>6</sup>They are copies of Figs. 2.6 and 2.7 on pages 12 and 13, respectively.

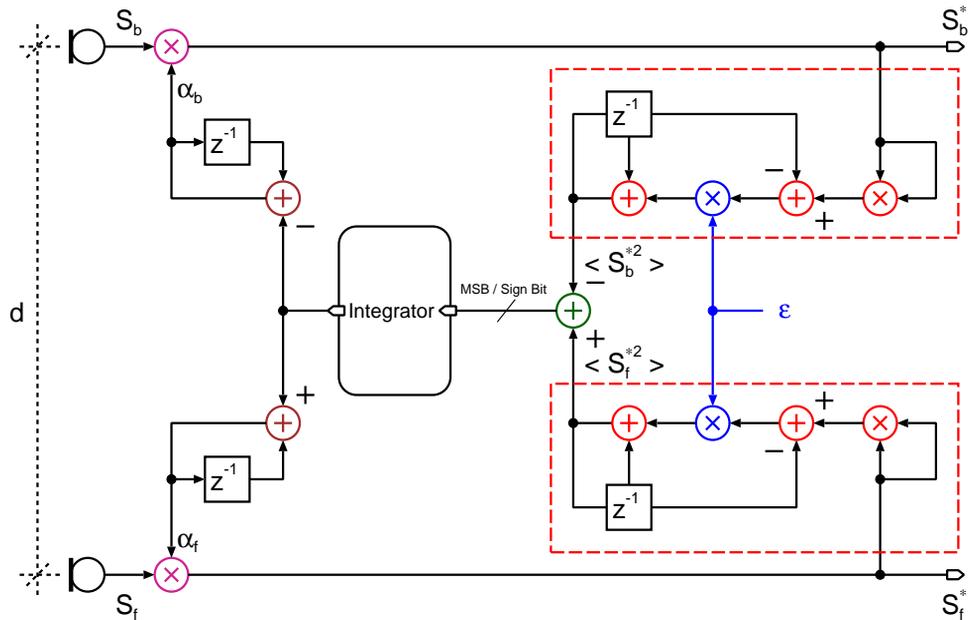


Figure 5.29: Power equalization preprocessor for Elko's differential microphone.

is colored dark green. The two preceding light green colored blocks perform either a sign or a zero extension depending on whether the subtractor inputs are signed or not, respectively<sup>7</sup>.

To complete the operations of the power equalization, only the integrator is needed. Additionally, the Elko algorithm requires a division (orange) and a multiplication (purple) to calculate the second summand of Eq. 2.5 ( $2\mu S_x[t-1] \times C_b[t] / \langle C_b^2[t] \rangle$ ). Thereby, the multiplication by  $2\mu$  corresponds to a simple multiplication by a negative power of two, which results in a rewiring of the bits (light blue).

The input stage, which calculates the two cardioids  $C_b$  and  $C_f$  based on  $S_f^{*2}[t]$  and  $\langle S_b^{*2}[t] \rangle$ , has been implemented by two iteratively decomposed FIR filters (two MAC units) plus two delay lines implemented with register files. These functional blocks are not shared with any other processing step. They have the same architecture as the ones in Fig. 5.24.

<sup>7</sup> $S_f^{*2}[t]$  and  $S_b^{*2}[t]$  are always positive, hence unsigned.

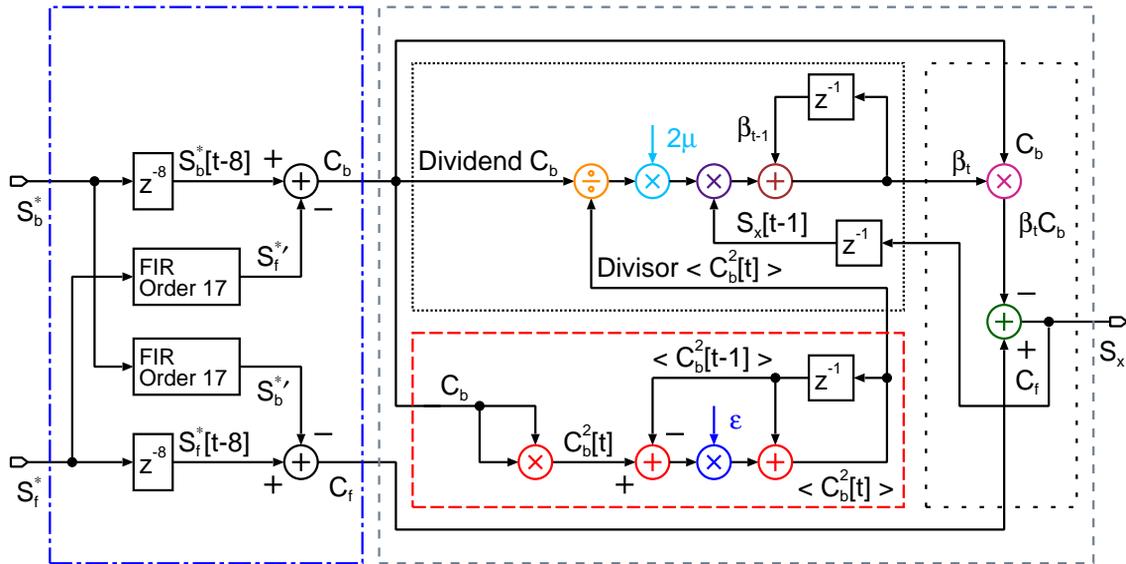


Figure 5.30: Elko's differential microphone algorithm with modified input stage.

Figure 5.32<sup>8</sup> shows a top-level block diagram of the second main building block, the channel-based noise reduction algorithm. Its architecture development is guided by similar observations as for the directional microphone. The filter bank operations in Eqs. 2.7 and 2.8 and the filtering in the main data path according to Eq. 2.15 are performed on a single general-purpose MAC unit containing a pre-adder. This block is illustrated in Fig. 5.33. The samples at the inputs  $x[t - K \pm k]$  correspond to the samples at the inputs of the half-band filters F0–F6 in Fig. 5.32. The additional input  $x[t - K]$ , which bypasses the pre-adder and is multiplied by  $\mathbf{b}_0$ , is required because the number of filter coefficients is odd. These constant half-band filter coefficients are applied at the  $\mathbf{b}_k$  input. The filter-bank outputs are present at the MAC output  $y_{\{\text{LP,HP}\},l}$ . Outputs of single half-band filters inside the filter-bank are present at the **AccuOut** output.

The samples of the FIR filter in the main processing path are applied at the  $x[t - M - m]$  and  $x[t - (M - 1) + m]$  input, whereas the corresponding adaptive coefficients are applied at the  $\mathbf{c}_m$  input. The filter output, equivalent to the noise reduction algorithm and system output, corresponds to the MAC unit output  $\mathbf{S}_y$ .

<sup>8</sup>This figure is a copy of Fig. 2.8 on page 15.

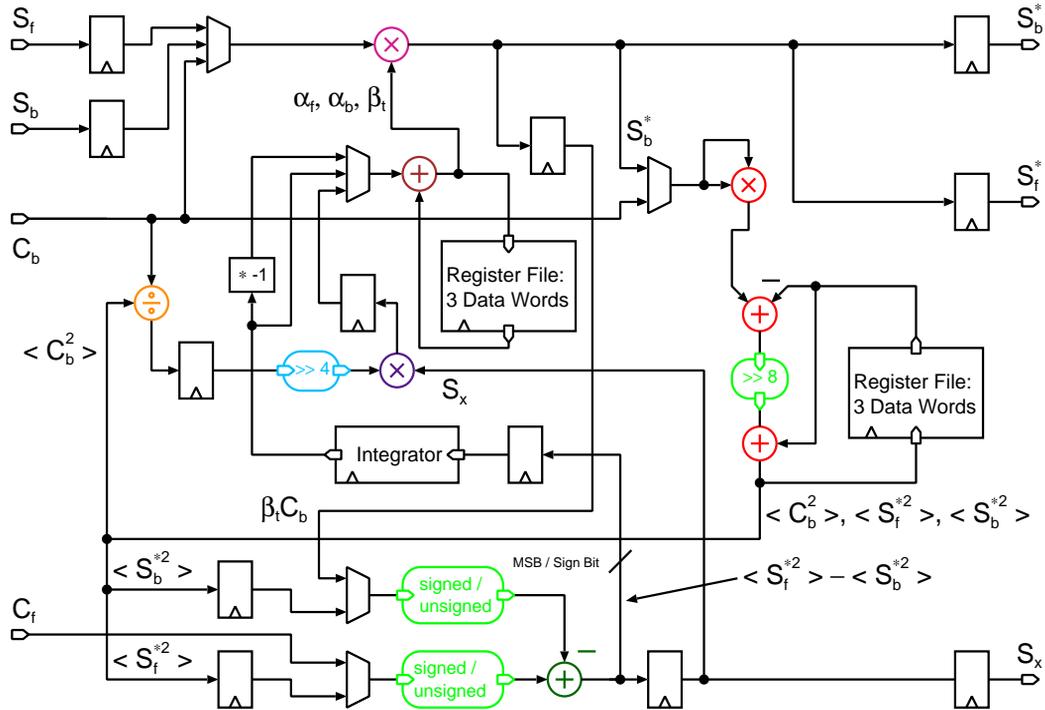


Figure 5.31: Implemented hardware architecture for the power equalization and Elko's algorithm.

The remaining inputs of the MAC unit in Fig. 5.33, namely  $h_m$ ,  $a_j$ ,  $c_{m,j}$  and **AccuOut**, are required for updating the FIR filter coefficients  $c_m$  according to Eq. 2.16. Thereby, in a first step, the sum  $\sum_{j=0}^{J-1} a_j \cdot c_{m,j}$  is calculated by the MAC unit. In a second step, this sum is multiplied by the Hamming coefficient applied at input  $h_m$ . The updated coefficient  $c_{m,j}$  is output at the **AccuOut** output of the MAC unit.

A careful scheduling of both filter and the update operations has been devised in order to not completely destroy signal correlations, which would inflate the switching activity and power consumption.

The filter-bank outputs  $y_{\{LP,HP\},l}$  ( $l = 5, 6, 4, 2$ ) are forwarded to the so-called squaring MAC unit that calculates the sub-band signal powers according to Eq. 2.9. In Fig. 5.32, these operations are performed in blocks B1–B8. The architecture of the implemented squaring MAC unit is depicted in Fig. 5.34. Note that this block has no coefficient input because the sample input is squared, motivating its name, and accumulated. This squaring MAC unit is time-shared between different filter-bank outputs, therefore a register file (blue)

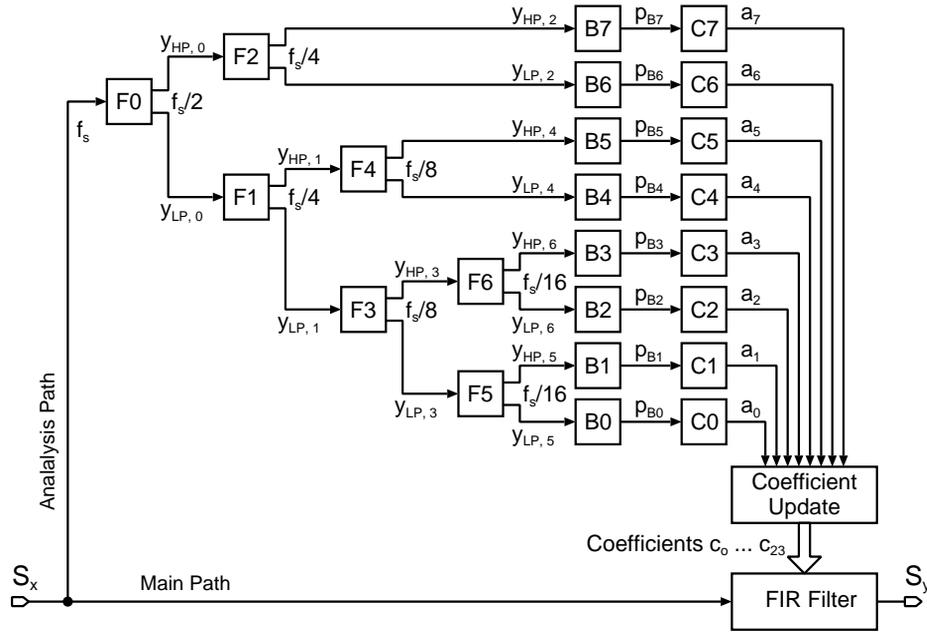


Figure 5.32: Channel-based noise reduction.

instead of a single accumulation register is required to accumulate the signal powers. After conversion to the logarithmic domain, the signal powers  $p_{Bj}$  ( $j = 0, \dots, 7$ ) are stored in the red register file. The **DiffReg** input and **Square** output are used for the update of the logarithmic power variation range in each sub-band. These ranges are eventually used to adapt the coefficients  $c_m$  of the main FIR filter. The implementation of this update procedure will be explained in the following.

According to Eqs. 2.10 and 2.11, an upper and a lower bound of the power variation range in each sub-band is updated. In Fig. 5.35 this takes place in the blue dashed frame. Based on these ranges, the red dashed block calculates the cube in Eq. 2.12. Thereby, squaring is performed in the multiplier mentioned above and shown in Fig. 5.34. The actual updating of the variation range  $r_j$  takes place in the light-green dashed box. Subsequently, the logarithmic gain reduction factors  $\Delta G_j$  are calculated. They are transformed into the linear domain resulting in  $a_j$ . These operations correspond to Eqs. 2.13 and 2.14, they have been highlighted by the brown dashed frame.

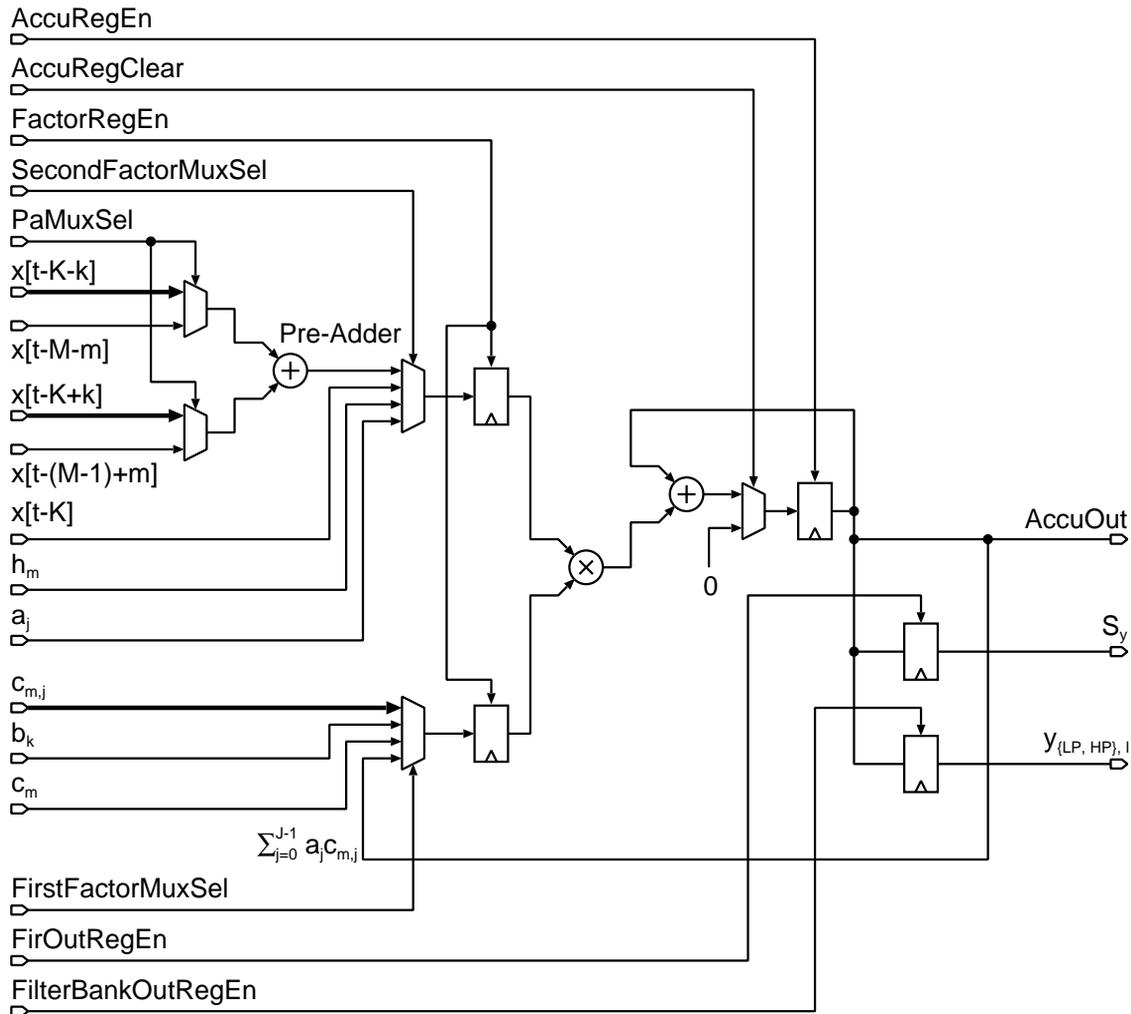


Figure 5.33: Implemented MAC unit in the channel-based noise reduction algorithm.

Figure 5.36 shows the register files and lookup tables (LUTs) that store data samples and updated as well as constant coefficients. Additionally, the figure contains the modulo counters that generate the read and – if required – the write addresses of all these LUTs and register files. Figure 5.37 depicts the top-level arrangement of the blocks shown in Fig. 5.33 to Fig. 5.36. The control signals, i.e., multiplexer selection, counter enable, register write-enable and clear signals have been aggregated in the blue signal busses for the sake of clearness. These signals are created in a finite state machine. In order to prevent glitches on these control signals, pipeline registers

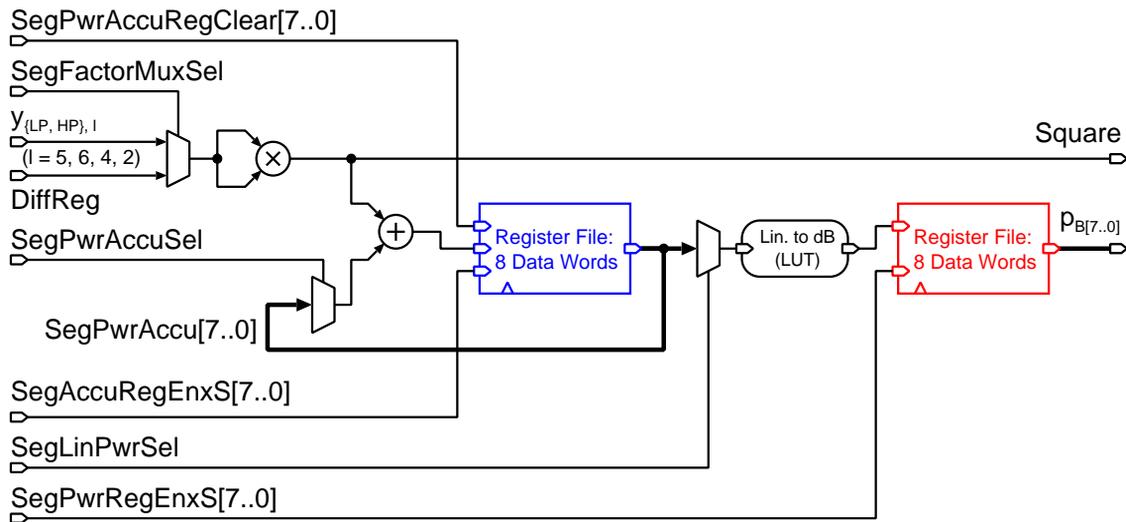


Figure 5.34: The implemented squaring MAC unit for the calculation of the logarithmic power variation ranges  $p_{B_j}$  ( $j = 0, \dots, 7$ ).

have been introduced at the state-machine outputs. Especially the selection signals of the multiplexers in front of the pre-adder and the multiplier inside the MAC unit are very sensitive to glitches because these arithmetic in conjunction with the multiplexers itself represent a large combinational block.

All architectural decisions presented in this section necessitate a clock frequency of 1 MHz in order to achieve a sampling rate of 16 kHz. Clock-gating has been extensively introduced in all designs.

The designs have been integrated in a standard 0.18  $\mu\text{m}$  CMOS process. As simulation and verification tool, the NC-VHDL/Verilog simulator by Cadence has been used. Synthesis has been performed by means of Ambit BuildGates by Cadence, targeting a digital standard cell library which was characterized for 0.75 V. In case of designs with own, redesigned cells, these cells have been added to the set of cells that were available to the hardware synthesizer.

As shown in Fig. 5.38, a low-power synthesis flow has been applied, and, the actual synthesis has been split up into two steps. A gate-level simulation has been performed after the first step. The toggle information collected has been used in the second synthesis step to further optimize the power consumption.

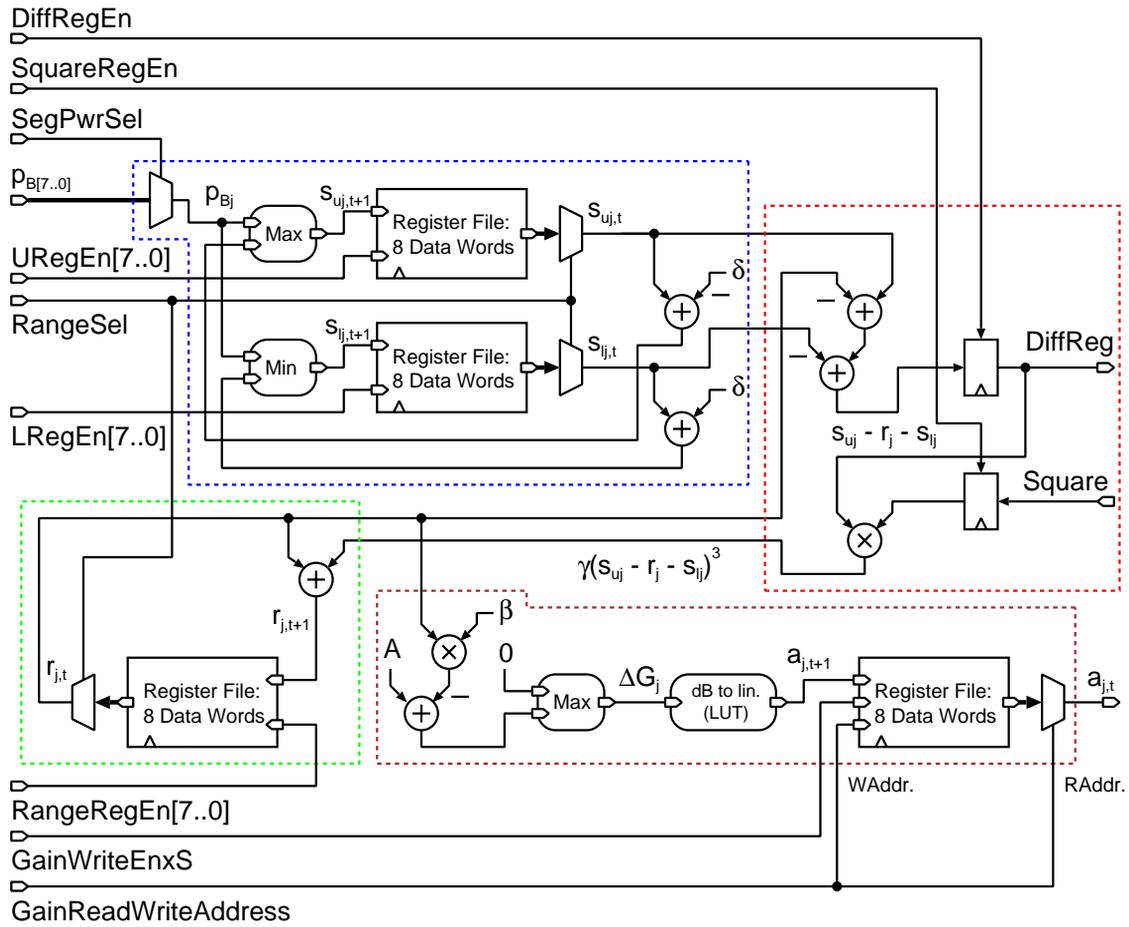


Figure 5.35: Updating of the power variation ranges  $r_j$  and gain reduction factors  $a_j$ , ( $j = 0, \dots, 7$ ).

### 5.2.3.2 Redesign of Cells and their Utilization

The cells whose layout has been redesigned in this work have the same circuit structure as the original in the standard cell library. In order to save power, however, transistor widths have been made narrower. Their lengths have been unchanged because they have already been at the minimum of  $0.18\mu\text{m}$ . In the original standard cell library, transistor widths are designed for optimal propagation delays and transition times, and only in second priority for low-power. For low-speed audio applications, it can be expected that timing violations in the longest combinational path are very unlikely to occur. Usually, the transistors are wider than the allowed minimum feature size given

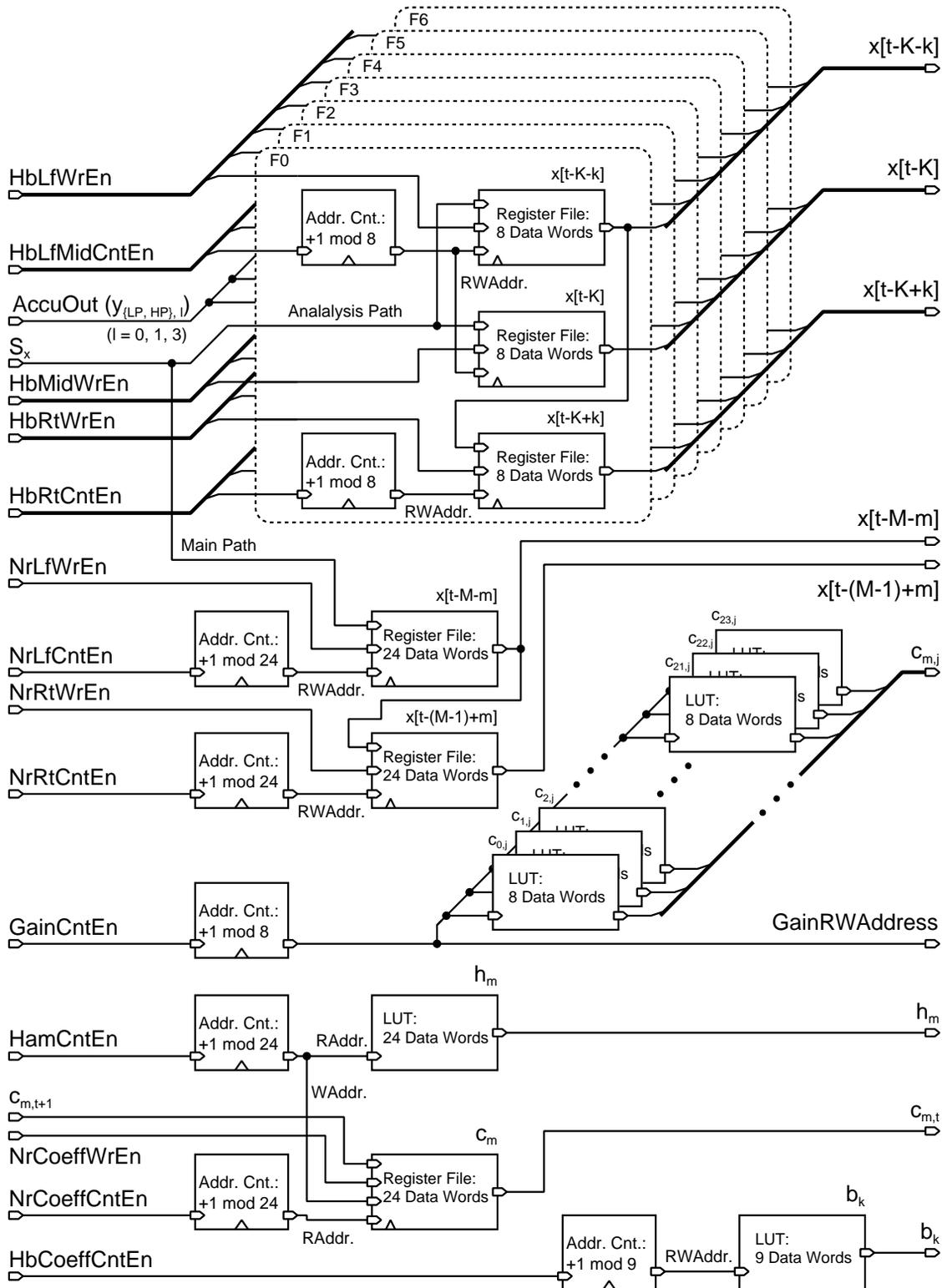


Figure 5.36: Register files, lookup tables (LUTs), and their address counters for the noise reduction algorithm.

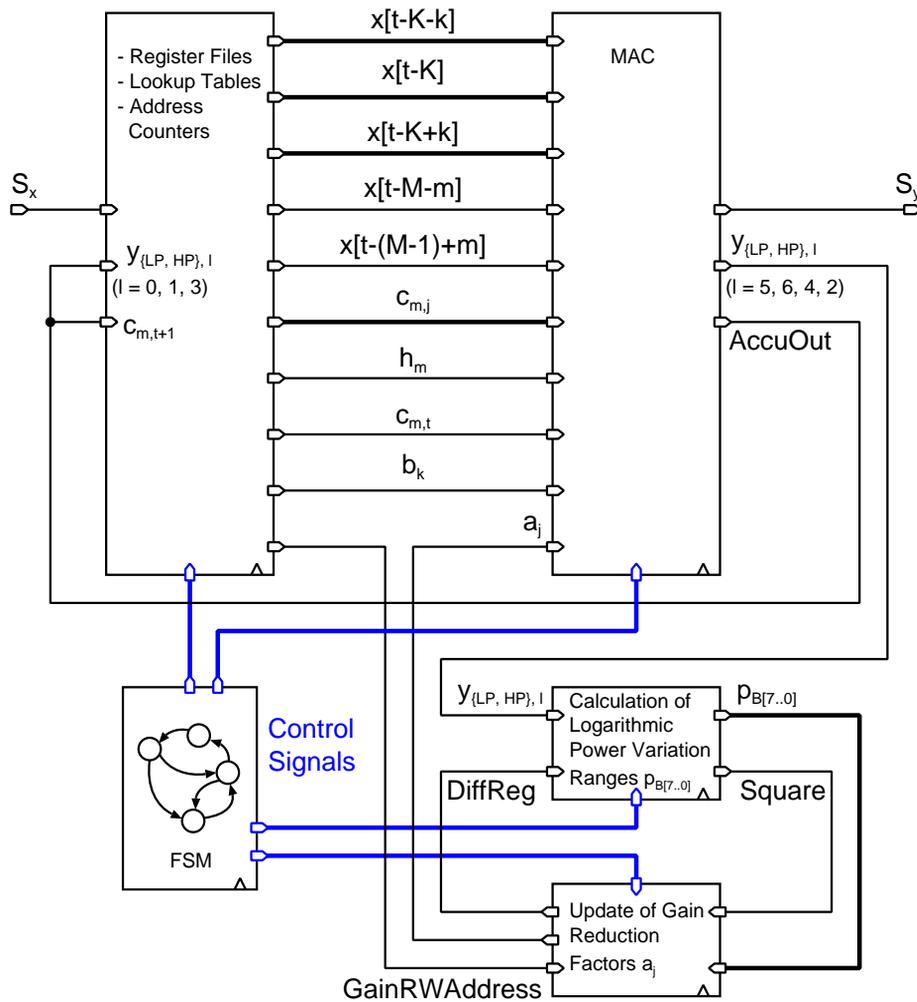


Figure 5.37: Top-level block arrangement of the noise reduction algorithm.

by the design rules. This provides scope for power optimization by narrowing transistors at the cost of speed. The minimum possible transistor width in this CMOS technology is  $0.24 \mu\text{m}$ . Nevertheless, in order to simplify the layout drawing of the redesigned cells, their transistor widths have not been down-scaled to this minimum. Instead, a constant transistor-width of  $0.44 \mu\text{m}$  has been chosen, which is equal for nMOS and pMOS devices. This results in the rectangular-shaped diffusion area of the transistor in Fig. 5.39b, compared to the more complicated dumbbell-shaped diffusion area of the minimum-width

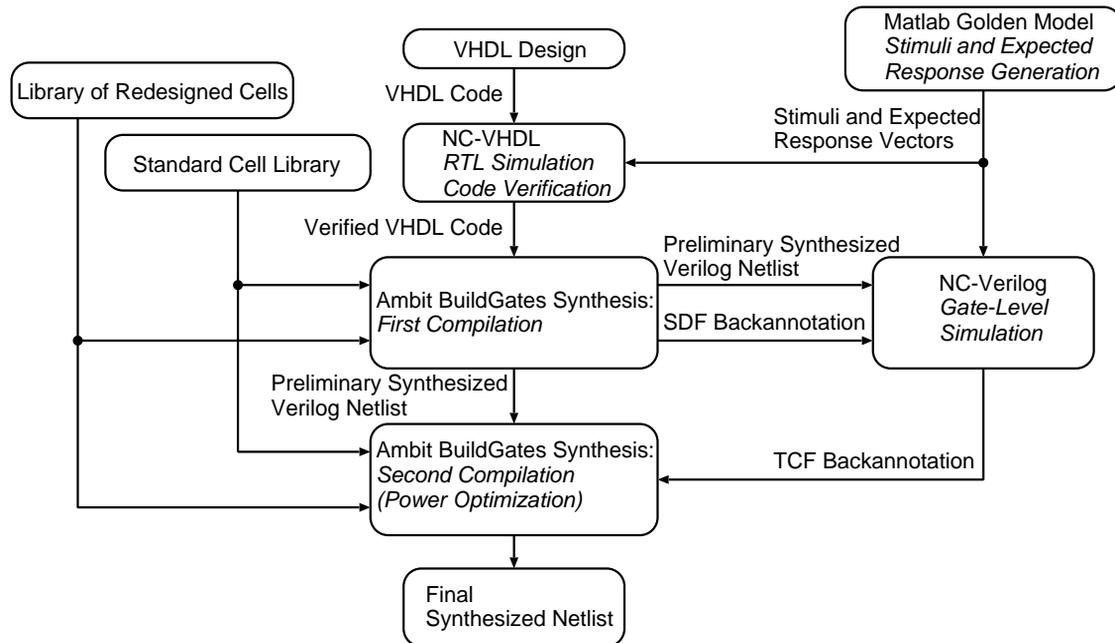


Figure 5.38: Low-power optimization flow in Ambit BuildGates consisting of two synthesis steps.

transistor in Fig. 5.39a, which is given by layout rules<sup>9</sup>. Paradoxically, these requirements result in a transistor layout of minimum-width devices that are longer (1.3  $\mu\text{m}$  in Fig. 5.39a) than the one of “larger” i.e., wider transistors used in this work (1.16  $\mu\text{m}$  in Fig. 5.39b).

All redesigned cells have been re-characterized in terms of timing and power quantities. Analog simulations at 0.75 V using Spectre by Cadence have been performed for this re-characterization.

### 5.2.3.3 Single-Edge-Triggered One-Phase Clocking Variant

As mentioned above, this single-edge-triggered one-phase clocking design serves as a state-of-the-art low-power reference. It does not contain any redesigned cells but uses clock gating and is fully scan-testable. Its VHDL code served as a basis from which all other designs have been derived. This reference implementation takes 1.60  $\text{mm}^2$  of silicon area.

<sup>9</sup>Contact size: 0.24  $\mu\text{m} \times 0.24 \mu\text{m}$ , minimum diffusion overlap (enclosure) around contact: 0.1  $\mu\text{m}$ , minimum distance between poly-silicon and contact: 0.15  $\mu\text{m}$ , and minimum distance between poly-silicon and diffusion: 0.12  $\mu\text{m}$ .

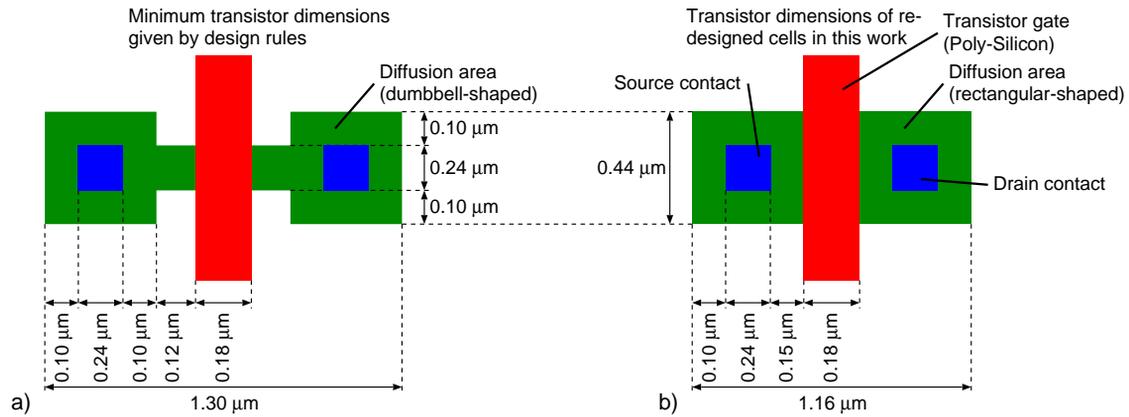


Figure 5.39: Transistor layouts: a) Minimum feature sized according to design rules, b) Transistor sizes in the redesigned cells of this work.

#### 5.2.3.4 Level-Sensitive Two-Phase Clocking Variant

Starting from the VHDL code of the single-edge-triggered reference, a *unsymmetric* level-sensitive two-phase clocking design has been devised as an intermediate step. Next, wherever possible, architecture transformations have been applied to turn it into its *symmetric* counterpart, actually giving rise to a *mixed* unsymmetric-symmetric level-sensitive clocking design, which has been integrated on silicon.

Figure 5.40 shows a die photo. The level-sensitive two-phase design is highlighted. Its area occupation amounts to  $2.26 \text{ mm}^2$ . The rest of the chip contains circuitry not discussed in this work.

The key figures of all designs are collected in Tab. 5.6. In addition to the measurements, results from post-layout gate-level simulations with the NC-Verilog simulator by Cadence<sup>10</sup> are separated for the adaptive directional microphone stage and the noise reduction algorithm stage.

The conversion from SETOPC to ULSTPC has a dramatic impact on dynamic power consumption, both for the directional microphone and the noise reduction stage. The reduction is caused by the relaxed skew and slew-rate sensitivity requirements of level-sensitive clocking, which allows for a much leaner clock distribution network. In spite

<sup>10</sup>The concept behind this simulation is exactly the same as in Fig. 3.3. Instead of a SAIF file, NC-Verilog creates a so-called toggle count format (TCF) file containing the switching activity information.

Table 5.6: Measured (upper part) and simulated (lower part) power in  $\mu\text{W}$  at  $V_{DD} = 0.75\text{ V}$  and  $f_{clk} = 1\text{ MHz}$ . Cell area in  $\text{mm}^2$ . Simulation results have been broken down into the differential microphone and the channel based noise reduction algorithm (diff. mic./noise reduction). (N.I.: Not integrated on silicon.)

	Standard Flip-Flops or Latches		Redesigned Latch	
	Variant I	not integrated	not integrated	Variant II
	SETOPC	ULSTPC	MLSTPC	MLSTPC
	Reference			
Cell Area	1.32	1.22	1.18	1.53
Dynamic	112.5	-	-	38.7
Leakage	28.5	-	-	11.3
Total (measured)	141.0	N.I.	N.I.	50.0
Dynamic	27.3 / 93.0	11.5 / 39.7	9.3 / 40.4	11.9 / 35.0
Leakage	4.8 / 15.9	4.2 / 17.3	3.0 / 17.3	2.6 / 12.7
Total (simulated)	32.1 / 108.9	15.7 / 57.0	12.3 / 57.7	14.5 / 47.7

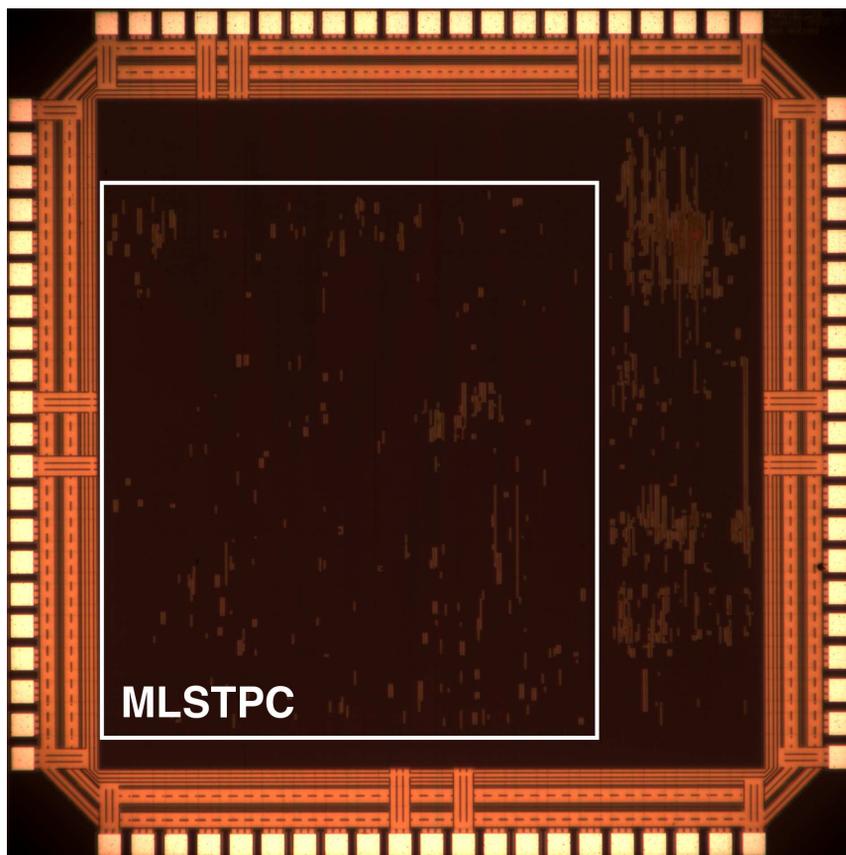


Figure 5.40: Photo of the die containing the mixed unsymmetric-symmetric level-sensitive two-phase clocking (MLSTPC) variant of the hearing aid system.

of the two clock trees, the total number of clock buffers has dropped from 944 to 16. The impact on leakage power is minor.

The transformation from *unsymmetric* level-sensitive to *mixed* unsymmetric-symmetric level-sensitive clocking moderately reduces the dynamic power for Elko’s algorithm stage, but is worth the effort: a power reduction from  $11.5\ \mu\text{W}$  to  $9.3\ \mu\text{W}$ , which corresponds to 19%, could be stated. The situation is quite different for the noise-reduction algorithm, which turns out to be unsuitable for *symmetric* level-sensitive clocking, thus effectively forcing the circuit into a *mixed* unsymmetric-symmetric level-sensitive clocking design (3rd column in Tab. 5.6). The other differences are so small that no further investigations of the underlying reasons have been conducted.

The 4th column in Tab. 5.6 refers to the integrated MLSTPC design with all standard cell library latches replaced by latches using

transistors sized as in Fig. 5.39b. The benefit of this modification is a leakage power reduction of 25% because of the narrower MOSFETs. The dynamic power in the noise reduction block is decreased by 13%. The same applies to the registers in the directional microphone implementation, yet its overall power consumption is actually increased. A possible explanation are parasitic effects caused by place and route which nullify those savings as the latch cells are outnumbered by unmodified combinational cells.

One valid objection to the above comparison is the fact that the reference single-edge-triggered design contains scan chains, whereas the level-sensitive clocked designs do not. Testing of SLSTPC circuits needs to be addressed in future work.

These investigations have shown that the conversion from classical single-edge-triggered one-phase to level-sensitive two-phase clocking significantly improves energy efficiency, provided the supply voltage is low enough that crossover currents do not compensate for the savings obtained from thinning out the clock tree. *Symmetric* or *mixed* unsymmetric-symmetric level-sensitive two-phase clocking further saves energy, but only if the combinational logic in a block lends itself to being split into two parts without necessitating too many shimming latches. Refer to Sec. 5.1.5 for details.

# Chapter 6

## Summary and Conclusions

Optimal adaption of hearing aids requires enhanced functionality. This leads to complex algorithms that consume a major part of the total energy for the digital signal processing. The extremely small dimensions of modern hearing aids severely limit their battery size. Acceptable battery lifetime asks for ultra low-power circuit techniques. This work tried to attack unnecessary energy dissipation on several levels of circuit implementation.

The strategies against the waste of energy are based on the formula, which describes that dynamic power consumption of digital CMOS circuits (Eq. 2.28) originates from charging capacitances when signals transition:

$$P_{dyn} = V_{DD}^2 f_{ci} \sum_{k=1}^K C_{L,k} \frac{\alpha_k}{2}.$$

The investigated techniques aim at reducing power consumption by decreasing overall toggle activity  $\alpha_k$  and node capacitance  $C_{L,k}$ . Lowering the supply voltage  $V_{DD}$  has been widely adopted: the presented circuits integrated in a 0.18  $\mu\text{m}$  technology use standard cells that have been re-characterized for 0.75 V, which is considerably less than the nominal supply voltage of 1.8 V, and comparable to the sum of nMOS and pMOS thresholds.

## Circuit Architecture Low-Power Techniques

In Chapter 3, different levels of iterative decomposition, i.e., resource sharing, have been explored in order to minimize the total switching activity ( $\alpha$ ) in the hardware implementation of a typical hearing aid algorithm. It has been shown that the optimum lies between a totally parallel (isomorphic) and completely decomposed (DSP-like) architecture. The ideal architecture uses the level of resource sharing that does not disintegrate the fundamental building blocks of the algorithm, in this case the lattice filter stages. Lower degrees of decomposition are subject to excessive creation and propagation of glitches that cause undesirable toggle activity and hence power consumption, despite the fact that these less decomposed architectures work with lower clock frequencies  $f_{ci}$ . On the other hand, higher levels of resource sharing require numerous multiplexers to switch between the different data streams (at the inputs of the few arithmetic units), thereby, augmenting switching activity as well. These results have been confirmed by measurements on silicon and simulations. The latter analyzed the toggle activities of the processing units (adders and multipliers).

## Number Format Techniques

Chapter 4 addresses the optimal number format for signed numbers. Whether a given algorithm is implemented in two's complement, sign-magnitude or hybrid number format significantly influences overall toggle activity. Based on the observation that multipliers are more energy-efficient in sign-magnitude format, different signal processing units have been realized in these three number formats. Relatively simple processing units, such as a MAC, have shown to profit from a replacement of the common two's complement format by sign-magnitude or even hybrid format. However, careful design of conversion units between the number formats is required for a low-power hybrid implementation. In particular, the necessary incrementers should be merged into subsequent adders, as it is possible in a MAC unit. More complicated units like the considered lattice stage do not allow for such merging. The permanent switching between number formats, which in

this case requires dedicated incrementers, creates unnecessary toggle activity. This in turn inflates power consumption. Likewise, a pure sign-magnitude implementation of the lattice stage has shown to be less efficient than a simple two's complement realization.

## Clocking Strategy Power Efficiency

In Chapter 5, different clocking strategies have been reviewed regarding their energy efficiency. Tight timing constraints of single-edge-triggered one-phase clocking necessitate a properly balanced clock tree with numerous power hungry buffers and inverters. The considered level-sensitive two-phase clocking disciplines allow for a minimal buffering of the clock tree. In case of *unsymmetric* level-sensitive two-phase clocking, where flip-flops are replaced by directly connected latch pairs, scan-chain testability can easily be maintained. The enhancement to *symmetric* level-sensitive two-phase clocking offers the possibility to relocate either master or slave latches between combinational blocks, whereby glitches are isolated from penetrating deeply into combinational logic. Yet, this advantage is a detriment to straightforward scan-chain testability. Nevertheless, by the inclusion of dummy latches, testability can be restored.

## Outlook

The solutions proposed in this thesis mainly reduce dynamic power consumption in digital VLSI. The related techniques have proved their feasibility and usefulness by simulations and measurements of hardware implementations of hearing aid algorithms. For this purpose, CMOS technologies with minimum feature sizes of 0.25  $\mu\text{m}$  and 0.18  $\mu\text{m}$  have been used. Clock frequencies of all considered designs have been in the range of a few MHz, which is still fairly low for these established CMOS technologies.

Future work should expand these low power techniques to higher-speed designs that possibly require more advanced and hence faster technologies (90 nm–65 nm). Although, dynamic power dissipation is expected to decrease, the amount of leakage consumption will become

larger. Therefore, future studies will have to investigate the influence of leakage dissipation on the overall power consumption. In the extreme case, excessive leakage dissipation may even ask for the smallest, i.e., fully iteratively decomposed architecture in order to minimize the total, dynamic *and* static, power consumption. This would contrast with the findings in Chapter 3.

# Appendix A

## Efficiency Comparison between Time- and Frequency-Domain Filtering

The output  $y[l]$  of a FIR filter of length  $K^1$  can be described as follows:

$$y[l] = \sum_{k=0}^{K-1} b_k x[l-k] \quad (\text{A.1})$$

Performing this filter operation in the time domain, which is equivalent to directly evaluate the sum in Eq. A.1, requires  $K$  real-valued multiplications per sample. The latency of this operation amounts to one sample when an architecture with a single MAC unit is chosen<sup>2</sup>, because for each processed data sample  $x[l]$ , the pertaining sample  $y[l]$  is the output. In an isomorphic architecture, even a latency of zero sample results [43].

If the filter length  $K$  grows large the filtering operation in Eq. A.1 can be performed more efficiently – in terms of arithmetic operations

---

<sup>1</sup>Such a filter has  $K$  coefficients  $b_k$ . The filter order  $N$  amounts to  $N = K - 1$ .

<sup>2</sup>In terms of clock cycles, the latency amounts to  $K$  in a single MAC architecture.

and hence power consumption – in the frequency domain by means of an FFT and a subsequent iFFT [21], [27]. In order to keep the latency of filtered output samples within reasonable range, a block-wise processing of data is required. However, this proceeding yields the result of the *circular* convolution and not of the desired *linear* convolution as described in Eq. A.1. Therefore, special care is needed in order to restore the linear convolution. The *overlap-add* and *overlap-save* methods fulfill this task [64], [27], [21]. In the former, filtered blocks overlap and overlapping parts are added together to build the filtered output signal  $y[l]$ . In the latter, the blocks of the input signal  $x[l]$  are selected overlapping and overlapping parts are saved for the convolution of the next block. Then, filtered blocks are abutted to assemble  $y[l]$ .

However, for both methods, filter length  $K$ , FFT length  $C$ , and block shift  $L$  are parameters that define the properties of a filter implementation according to one of the two methods. Usually, the filter order and the maximum tolerable latency of the output samples, which is given by the block size, are constraints given by the application. To allow for an efficient FFT and iFFT calculation,  $C$  is best chosen as an integer power of 2. In this case, an FFT or iFFT requires  $C \log_2 C$  real-valued multiplications (under the assumption of a real-valued signal). Additionally, in order that the linear convolution can be reconstructed by the overlap-add or overlap-save method the following condition has to be met:

$$L \leq C - K + 1 \quad (\text{A.2})$$

For the overlap-save method, the total number of real-valued multiplications per output sample amounts to:

$$M = \frac{2C \log_2 C + C}{L} \quad (\text{A.3})$$

The factor “2” is required because one FFT and one iFFT have to be calculated. The additional  $C$  multiplications are given by the element-wise multiplication of the transformed filter coefficients by the transformed samples in a block. Since only  $L$  samples per processed block of length  $C$  are correct and used – the remaining  $K - 1$  samples are discarded – the sum in Eq. A.3 has to be divided by  $L$ . Note that

the overlap-add method would require the same number of real-valued multiplications, but – as mentioned above – additional additions are required to add overlapping parts of the filtered blocks.

Table A.1 reports the number of real-valued multiplications  $M$  per data sample of a FIR filter implemented with the overlap-save method according to Eq. A.3. The filter order amounts to  $N = K - 1 = 99$  in accordance with the filter in Sec. 4.2. The block shift  $L$  has been set to  $L = C - K + 1$  in compliance with the inequality in Eq. A.2. The table shows that the optimum of 24 real-valued multiplications can be realized by an FFT length  $C$  of 512 or 1024. These 24 multiplications are significantly less than the  $K = 100$  multiplications of a direct evaluation of A.1.

Unfortunately, the block shift  $L$  gives rise to a latency of the filtered output samples. This latency amounts to:

$$\tau = L \cdot 1/f_s \quad (\text{A.4})$$

According to the filter in Sec. 4.2, the sampling frequency  $f_s$  is 22.05 kHz. The last column in Tab. A.1 shows the resulting latency  $\tau$ .

A real-time system like a hearing aid demands for small latencies of processed data. Latencies in excess of around 10 ms are undesired [23]. Only the first two combinations of  $C$  and  $L$  in Tab. A.1 fulfill this requirement at a first glance. The number of multiplications of the first one ( $C = 128$ ,  $L = 29$ ) is certainly worse than the optimum, but still better than a time domain implementation. However, this simple study does not consider the additional memory and control overhead of a frequency domain implementation. The second combination ( $C = 256$ ,  $L = 157$ ) is near to the optimum regarding number of multiplications. Though, the delay of 7 ms is close to 10 ms. Moreover, real delay values will be larger than the one reported in Tab. A.1 because the actual processing times for the FFT, multiplication, and iFFT have to be added to the reported values.

Table A.1: Number of real-valued multiplications  $M$  and latency  $\tau$  of a FIR filter implemented with the overlap-save method. The filter order  $N$  amounts to 99 and the sampling frequency  $f_s$  to 22.05 kHz. The FFT length is  $C$  and the block shift is  $L$ .

C	L	M	$\tau$ (in ms)
128	29	67	1
256	157	28	7
512	413	24	19
1 024	925	24	42
2 048	1 949	25	88
4 096	3 997	26	181
8 192	8 093	28	367
16 384	16 285	30	739
32 768	32 669	32	1 482
65 536	65 437	34	2 968
131 072	130 973	36	5 940
262 144	262 045	38	11 884
524 288	524 189	40	23 773
1 048 576	1 048 477	42	47 550

## Appendix B

# Energy Effective Capacitance Calculation

Equations 5.13 and 5.14 express the energy effective capacitances respectively of a standard memory decoder and a memory decoder featuring multi-stage clock gating (Fig. 5.27) as they can be calculated from node switching activities ( $\alpha$ ).

Given a memory of  $n$  (a power of two) data words, and defining  $k$  as  $\log_2 n$ , the switching activities of the counter outputs  $c_0 \dots c_{k-1}$  are:

$$\alpha_{c_0} = 1 \quad \alpha_{c_1} = \frac{1}{2} \quad \dots \quad \alpha_{c_{k-1}} = \frac{1}{2^{k-1}} \quad (\text{B.1})$$

The signals  $q_0 \dots q_{n-1}$  toggle only twice during a whole writing cycle, therefore:

$$\alpha_{q_j} = \frac{2}{n} = \frac{1}{2^{k-1}} \quad j = 0, \dots, n-1 \quad (\text{B.2})$$

The input gate capacitance is  $C_0$  and the switching activity of the master phase  $\alpha_m$  is two by definition. The energy effective capacitance of the traditional decoder ( $C_1$ ) can be expressed through proper combination of Eq. B.1 and Eq. B.2:

$$C_1 = \sum_{j=0}^{k-1} \alpha_{c_j} \cdot nC_0 + \sum_{j=0}^{n-1} \alpha_{q_j} \cdot C_0 + \alpha_m \cdot nC_0 = 4nC_0 \quad (\text{B.3})$$

After rearranging the decoder in a hierarchical way, the switching activity of the nodes  $w_{ij}$  can be expressed as:

$$\alpha_{w_{00}} = 2, \dots, \alpha_{w_{ij}} = \frac{1}{2^{i-1}} \quad (\text{B.4})$$

Equations B.1 and B.4 let express the energy effective capacitance of a multi-stage decoder ( $C_2$ ):

$$\begin{aligned} C_2 &= (2\alpha_{c_{k-1}} + \dots + \frac{n}{2} \cdot \alpha_{c_1} + n\alpha_{c_0}) \cdot C_0 \\ &\quad + \sum_{i=0}^{k-1} 2 \sum_{j=0}^{2^i-1} \alpha_{w_{ij}} \cdot C_0 + \alpha_m C_0 \\ &= nC_0 \cdot \sum_{i=0}^{k-1} \left(\frac{1}{4}\right)^i + 4kC_0 + 2C_0 \\ &= \left(4 \cdot \frac{n^2 - 1}{3n} + 4 \log_2 n + 2\right) \cdot C_0 \end{aligned} \quad (\text{B.5})$$

# List of Abbreviations

<b>ADC</b>	analog-to-digital converter
<b>ALU</b>	arithmetic logic unit
<b>AOI</b>	AND-OR-INVERT, binary operation
<b>ASIC</b>	application specific integrated circuit
<b>ATE</b>	automatic test equipment
<b>ATPG</b>	automatic test pattern generation
<b>BTE</b>	behind-the-ear
<b>CI</b>	carry-in
<b>CIC</b>	completely-in-the-canal
<b>CMOS</b>	complementary metal-oxide-semiconductor
<b>D-FF</b>	D-type flip-flop
<b>DAC</b>	digital-to-analog converter
<b>DDG</b>	data dependency graph
<b>DET</b>	dual-edge-triggered
<b>DETOPC</b>	dual-edge-triggered one-phase clocking
<b>DSP</b>	digital signal processing or digital signal processor

<b>EDA</b>	electronic design automation
<b>FA</b>	full adder
<b>FFT</b>	fast fourier transform
<b>FIR</b>	finite-impulse-response
<b>FSM</b>	finite state machine
<b>HA</b>	half adder
<b>HY</b>	hybrid
<b>iFFT</b>	inverse fast fourier transform
<b>IIR</b>	infinite impulse response
<b>ITC</b>	in-the-canal
<b>ITE</b>	in-the-ear
<b>LMS</b>	least mean squares
<b>LSB</b>	least significant bit
<b>LUT</b>	lookup table
<b>MAC</b>	multiply-accumulate
<b>MLSTPC</b>	mixed unsymmetric-symmetric level-sensitive two-phase clocking
<b>MOSFET</b>	metal-oxide-semiconductor field-effect transistor
<b>MPW</b>	multi-project waver
<b>MSB</b>	most significant bit
<b>NAND</b>	NOT AND, binary operation
<b>nMOS</b>	n-type MOSFET
<b>NOR</b>	NOT OR, binary operation

<b>pMOS</b>	p-type MOSFET
<b>RTL</b>	register-transfer level
<b>SAIF</b>	switching activity interchange format
<b>SDF</b>	standard delay format
<b>SET</b>	single-edge-triggered
<b>SETOPC</b>	single-edge-triggered one-phase clocking
$\Sigma\Delta$	sigma-delta modulation is a technique used in analog-to-digital and digital-to-analog converters
<b>SLSTPC</b>	symmetric level-sensitive two-phase clocking
<b>SM</b>	sign-magnitude
<b>SNR</b>	signal-to-noise ratio
<b>TC</b>	two's complement
<b>TCF</b>	toggle count format
<b>ULSTPC</b>	unsymmetric level-sensitive two-phase clocking
<b>VHDL</b>	VHSIC Hardware Description Language, VHSIC in turn stands for Very High Speed Integrated Circuits, VHDL is a widely used hardware description language.
<b>VLSI</b>	very large scale integration
<b>XOR</b>	EXCLUSIVE OR, binary operation



# Bibliography

- [1] A. M. Engebretson, “Benefits of Digital Hearing Aids,” *IEEE Eng. Med. Biol. Mag.*, vol. 13, no. 2, pp. 238–248, Apr./May 1994.
- [2] B. Edwards, “The Future of Hearing Aid Technology,” *Trends in Amplification*, vol. 11, no. 1, pp. 31–46, 2006. [Online]. Available: <http://tia.sagepub.com/cgi/content/refs/11/1/31>
- [3] F. Buergin, F. Carbognani, M. Hediger, H. Meier, R. Meyer-Piening, R. Santschi, H. Kaeslin, N. Felber, and W. Fichtner, “Low-power Architectural Trade-Offs in a VLSI Implementation of an Adaptive Hearing Aid Algorithm,” in *Design Automation Conference, 2006 43rd ACM/IEEE*, 24–28 Jul. 2006, pp. 558–561.
- [4] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner, “42% Power Savings through Glitch-Reducing Clocking Strategy in a Hearing Aid Application,” in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, Island of Kos, Greece, 21–24 May 2006, pp. 2941–2944.
- [5] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner, “Two-Phase Clocking and a New Latch Design for Low-Power Portable Applications,” in *PATMOS '05: Proceedings of the 15th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation*, ser. Lecture Notes in Computer Science. Leuven, Belgium: Springer-Verlag, Sep. 2005, pp. 446–455. [Online]. Available: <http://www.springerlink.com/content/dmjru91r3a7v/>

- [6] F. Buergin, F. Carbognani, H. Kaeslin, N. Felber, and W. Fichtner, "Latch-Based Clocking for Portable Low-Power Audio Applications," in *Proc. International Conference on Solid-State and Integrated Circuits Technology (ICSICT'06)*, Shanghai, China, Oct. 2006, pp. 1640–1642.
- [7] "Information for Medical Applications," Solution Guide, Texas Instruments Incorporated, Post Office Box 655303 Dallas, Texas 75265, 2005. [Online]. Available: <http://www.ti.com/litv/pdf/slyb108a>
- [8] T. A. Ricketts. Digital Hearing Aids: Current "State-of-the-Art". American Speech-Language-Hearing Association (ASHA). 10801 Rockville Pike, Rockville, Maryland 20852. [Online]. Available: [http://www.asha.org/public/hearing/treatment/digital\\_aid.htm](http://www.asha.org/public/hearing/treatment/digital_aid.htm)
- [9] Symbio XT - Produkte. [Online]. Available: [http://bernafon.ch/eprise/main/ch\\_de/Consumers/Products/SymbioXT](http://bernafon.ch/eprise/main/ch_de/Consumers/Products/SymbioXT)
- [10] Four categories. hear-it AISBL. Avenue Tervuren 35, B-1040 Brussels, Belgium. [Online]. Available: <http://www.hear-it.org/page.dsp?page=339>
- [11] Low Power Real-Time Programmable DSP Development Platform for Digital Hearing (spra657.pdf). Texas Instruments. [Online]. Available: <http://www.ti.com>
- [12] AIC111 IC Design Specification (aic111.pdf). Texas Instruments. [Online]. Available: <http://focus.ti.com/docs/prod/folders/print/aic111.html>
- [13] A Quick Guide to Zinc Air Batteries. [Online]. Available: <http://www.renata.com/content/hearingaid/quickguide.php>
- [14] Hearing Aid Battery Life. [Online]. Available: <http://www.energizer.com/products/haprofessionals/patientCare.aspx>
- [15] (2005) Technical Bulletin. [Online]. Available: [http://duracell.com/oem/primary/Zinc/zinc\\_air\\_tech.asp](http://duracell.com/oem/primary/Zinc/zinc_air_tech.asp)
- [16] Hearing Aid Batteries. [Online]. Available: <http://www.renata.com/content/hearingaid/>

- [17] (2005) Product Data. [Online]. Available: <http://duracell.com/Procell/productdata/>
- [18] G. W. Elko and A.-T. N. Pong, "A simple adaptive first-order differential microphone," in *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, New Paltz, NY, 15–18 Oct. 1995, pp. 169–172.
- [19] H. Teutsch and G. W. Elko, "First- and Second-Order Adaptive Differential Microphone Arrays," in *7th International Workshop on Acoustic Echo and Noise Control*, Darmstadt, Germany, Sep. 2001, pp. 35–38.
- [20] F. Carbognani, F. Bürgin, L. Henzen, H. Koch, H. Magdassian, C. Pedretti, H. Kaeslin, N. Felber, and W. Fichtner, "A 0.67-mm<sup>2</sup> 45- $\mu$ W DSP VLSI Implementation of an Adaptive Directional Microphone for Hearing Aids," in *Proc. of European Conference on Circuit Theory and Design (ECCTD)*, vol. 3, Cork, Ireland, Aug. 2005, pp. 141–144.
- [21] G. S. Moschytz and M. Hofbauer, *Adaptive Filter, Eine Einführung in die Theorie mit Aufgaben und MATLAB-Simulationen auf CD-ROM*. Berlin, Heidelberg, New York: Springer-Verlag, 2000.
- [22] A. Schaub, "Hörhilfe," European Patent EP 1 067 821 A2, 2001.
- [23] A. Schaub, "Hearing Aid," U.S. Patent US 6 580 798 B1, 2003.
- [24] P. P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. IEEE*, vol. 78, no. 1, pp. 56–93, Jan. 1990.
- [25] A. Schaub and P. Straub, "Spectral Sharpening for Speech Enhancement/Noise Reduction," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, Toronto, Ont., 14–17 Apr. 1991, pp. 993–996.
- [26] J.-H. Chen and A. Gersho, "Real-time vector APC speech coding at 4800 bps with adaptive postfiltering," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87*, vol. 12, Apr. 1987, pp. 2185–2188.

- [27] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall International, Inc., 1989.
- [28] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*. Englewood Cliffs, NJ: Prentice-Hall International, Inc., 1985.
- [29] F. Wanlass and C. Sah, "Nanowatt Logic Using Field-Effect Metal-Oxide Semiconductor Triodes," in *Solid-State Circuits Conference. Digest of Technical Papers. 1963 IEEE International*, Feb. 1963, pp. 32–33.
- [30] M. J. Riezenman, "Wanlass's CMOS circuit," *IEEE Spectrum*, vol. 28, no. 5, May 1991.
- [31] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, vol. 38, no. 8, 19 Apr. 1965. [Online]. Available: <http://download.intel.com/research/silicon/moorespaper.pdf>
- [32] C. Lécuyer and D. C. Brock, "Biographies," *IEEE Ann. Hist. Comput.*, vol. 28, no. 3, pp. 89–95, Jul./Sep. 2006.
- [33] C. Piguet, "The First Quartz Electronic Watch," in *PATMOS '02: Proceedings of the 12th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation*, ser. Lecture Notes in Computer Science. Seville, Spain: Springer-Verlag, Sep. 2002, pp. 1–15.
- [34] Smithsonian Institute. Beta 21. [Online]. Available: <http://invention.smithsonian.org/centerpieces/quartz/coolwatches/beta21.html>
- [35] Smithsonian Institute. Seiko Quartz 35 SQ. [Online]. Available: <http://invention.smithsonian.org/centerpieces/quartz/coolwatches/seiko.html>
- [36] K. K. Yagura, G. M. Catlin, and J. D. Hutchenson, "Monolithic MOS complementary pairs," in *IEEE Electron Devices Meeting, 1966 International*, vol. 12, 1966, pp. 64–64.

- [37] A. K. Rapp, L. P. Wennick, H. Borkan, and K. R. Keller, "Complementary-MOS Integrated Binary Counter," in *Solid-State Circuits Conference. Digest of Technical Papers. 1967 IEEE International*, Feb. 1967, pp. 52–53.
- [38] F. Leuenberger and E. A. Vittoz, "Complementary-MOS Low-Power Low-Voltage Integrated Binary Counter," in *Proc. IEEE*, vol. 57, no. 9, Sep. 1969, pp. 1528–1532.
- [39] E. A. Vittoz, "Low-Power Design: Ways to Approach the Limits," in *Solid-State Circuits Conference, 1994. Digest of Technical Papers. 41st ISSCC. IEEE International*, San Francisco, CA, 16–18 Feb. 1994, pp. 14–18.
- [40] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Norwell, MA: Kluwer Academic Publishers, 1995.
- [41] K. Azar, "The history of power dissipation," *Electronics Cooling Magazine*, vol. 6, no. 1, Jan. 2000. [Online]. Available: <http://www.electronics-cooling.com/articles/>
- [42] K.-S. Yeo and K. Roy, *Low-Voltage, Low-Power VLSI Subsystems*. New York, NY: McGraw-Hill, 2005.
- [43] H. Kaeslin, *Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication*, 1st ed. Cambridge University Press, 2008.
- [44] H. J. M. Veendrick, *Deep-Submicron CMOS ICs From Basics to ASICs*, 2nd ed. Ten Hagen en Stam, Deventer, The Netherlands: Kluwer Academic Publishers, Dec. 2000.
- [45] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Application to CMOS Inverter Delay and Other Formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.
- [46] H. J. M. Veendrick, "Short-Circuit Dissipation of Static CMOS Circuitry and Its Impact on the Design of Buffer Circuits," *IEEE J. Solid-State Circuits*, vol. SC-19, no. 4, pp. 468–473, Aug. 1984.

- [47] F. Carbognani, “Low-Power Techniques for Low-Frequency VLSI Applications,” Ph.D. dissertation, Swiss Federal Institute of Technology Zurich, 2007.
- [48] *Power Compiler™ User Guide*, Version Y-2006.06, Synopsys, Inc., Mountain View, CA, USA, 2006.
- [49] *Low Power for BuildGates® Synthesis and Cadence® PKS*, Product Version 5.15, Cadence Design Systems, Inc., San Jose, CA, USA, Jan. 2005.
- [50] D. D. Gajski and R. H. Kuhn, “New VLSI Tools,” *IEEE Computer*, vol. 16, no. 12, pp. 11–14, Dec. 1983.
- [51] P. Mosch, G. van Oerle, S. Menzl, N. Rougnon-Glasson, K. van Nieuwenhove, and M. Wezelenburg, “A 660- $\mu$ W 50-Mops 1-V DSP for a Hearing Aid Chip Set,” *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1705–1712, Nov. 2000.
- [52] A. T. Erdogan, E. Zwyszig, and T. Arslan, “Architectural trade-offs in the design of low power FIR filtering cores,” *IEE Proceedings - Circuits, Devices and Systems*, vol. 151, no. 1, pp. 10–17, 5 Feb. 2004.
- [53] S. Grassi, “Optimized implementation of speech processing algorithms,” Ph.D. dissertation, University of Neuchâtel, Neuchâtel, Switzerland, Feb. 1998. [Online]. Available: <http://doc.rero.ch/search.py?recid=5188&ln=en>
- [54] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner, “A Self-Timed 16-bit Multiplier for Low-Power Low-Frequency Applications,” in *Proc. Mid-West Symposium on Circuits and Systems (MWSCAS'06)*, vol. 2, San Juan, Puerto Rico, 6–9 Aug. 2006, pp. 433–437.
- [55] P. Ruiz-de Clavijo, J. Juan-Chico, M. J. Bellido-Díaz, A. M. Calderón, D. G. Martos, O. E., and J. Viejo, “Logic-Level Fast Current Simulation for Digital CMOS Circuits,” in *PATMOS '05: Proceedings of the 15th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and*

- Simulation*, ser. Lecture Notes in Computer Science. Leuven, Belgium: Springer-Verlag, Sep. 2005, pp. 425–435.
- [56] A. P. Chandrakasan, “Ultra Low Power Digital Signal Processing,” in *VLSI Design, 1996. Proceedings., Ninth International Conference on*, Bangalore, India, 3–6 Jan. 1996, pp. 352–357.
- [57] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1998.
- [58] P. J. Ashenden, *The Designer’s Guide to VHDL*, 2nd ed. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 2002.
- [59] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*. New York, NY: Oxford University Press, Inc., 2000.
- [60] J. Wassner, “Data Statistics and Low-Power Digital VLSI,” Ph.D. dissertation, ETH, Zurich, Switzerland, 2001. [Online]. Available: <http://e-collection.ethbib.ethz.ch/show?type=diss&nr=14395>
- [61] *HDL Compiler<sup>TM</sup> (Presto VHDL) Reference Manual*, Synopsys, Inc., Mountain View, CA, USA, 2006.
- [62] K. Baker and J. van Beers, “Shmoo Plotting : The Black Art of IC Testing,” in *Proceedings International Test Conference 1996*, 1996, pp. 932–933.
- [63] L. P. Ching and O. G. Ling, “Low-power and low-voltage D-latch,” *Electronics Letters*, vol. 34, no. 7, pp. 641–642, 2 Apr. 1998.
- [64] H. D. Helms, “Fast Fourier Transform Method of Computing Difference Equations and Simulating Filters,” *IEEE Trans. Audio Electroacoust.*, vol. AU-15, no. 2, pp. 85–90, Jun. 1967.



# Curriculum Vitae

Felix Bürgin was born in Basel (Switzerland) on 24 October 1976. He studied electrical engineering at ETH Zurich. In 2003 he received his Dipl.El.-Ing. degree writing a thesis about a  $\Sigma\Delta$  analog-to-digital converter for high quality audio applications at the Integrated Systems Laboratory (IIS) of ETH Zurich.

Immediately afterwards, he joined the Integrated Systems Laboratory at ETH Zurich as a research and teaching assistant. He pursued the Ph.D. degree at the same laboratory. His primary research interest is in low power ASIC design, focused on hearing aids and mobile applications.