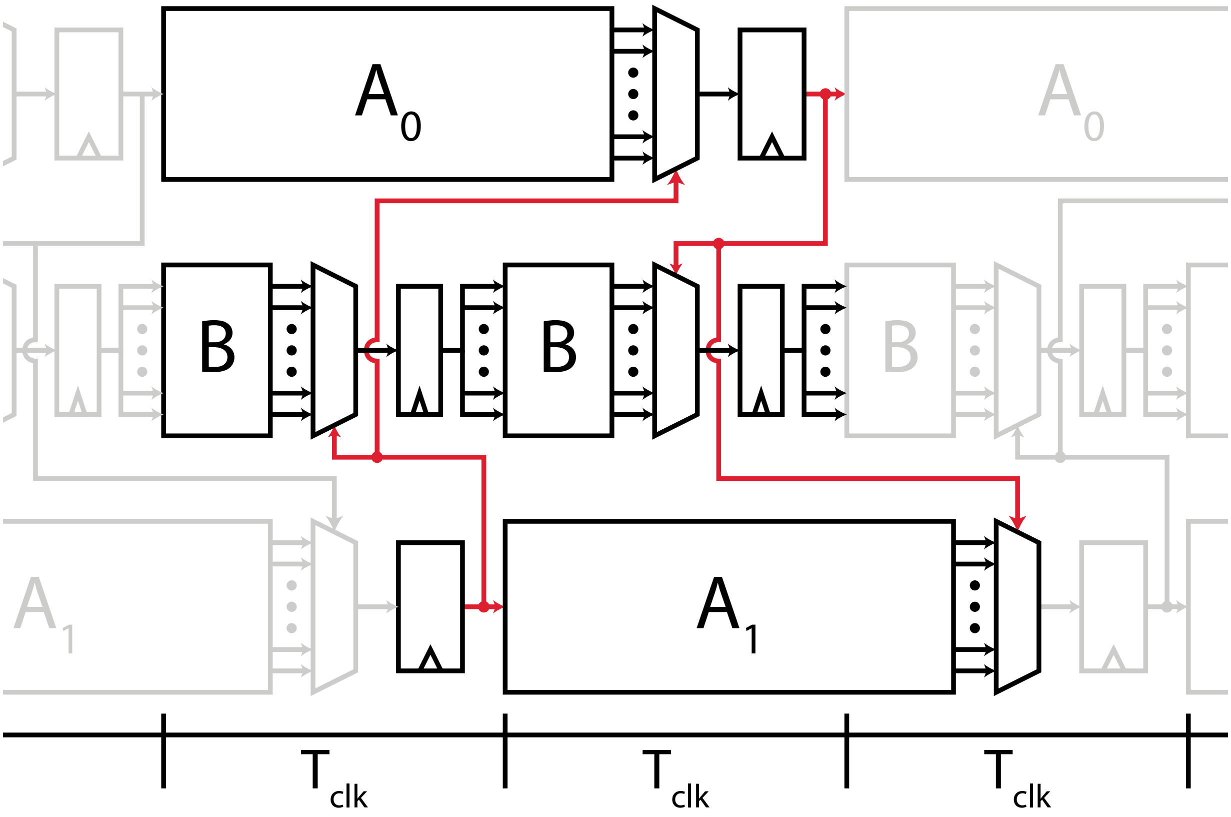
CHALMERS
UNIVERSITY OF TECHNOLOGY



A Novel Speculative Pseudo-Parallel Delta-Sigma Modulator

Using FPGAs to produce a high-speed digital signal for use in MIMO transmission systems

Master of Science Thesis, MSc in Embedded Electronic System Design

JESPER JOHANSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014

A Novel Speculative Pseudo-Parallel Delta-Sigma Modulator
Using FPGAs to produce a high-speed digital signal for use in MIMO transmission systems

Jesper Johansson

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Tel: +46-(0)31-772 1000
Fax: +46-(0)31-772 3663

[Graphical representation of the different blocks in the speculative modulator array and their interaction in time.]

Department of Computer Science and Engineering
Gothenburg, 2014

Contents

Abstract

As digital communication using MIMO antenna technology has emerged as a significant technological breakthrough in modern communications, the need for reconfigurable experimental MIMO systems has increased.

This project is a feasibility study of using $\Delta\Sigma$ modulators for generating high-speed digital signals on an FPGA chip synchronized to one local oscillator. The targeted performance specifications are a 1 GHz carrier frequency, 100 MHz bandwidth, and 45 dB in-band SNDR. The approach used to achieve the performance specifications is a novel speculative pseudo-parallel $\Delta\Sigma$ modulator based on the the method for unrolling a $\Delta\Sigma$ modulator proposed by Hatami et al.

The speculative modulator structure, although it fails to achieve the tentative system specifications, exhibits a 46 % increase in operating frequency versus the regular Hatami structure. The speculative modulator structure has the potential for further investigation and design space exploration. The main reason for failing to reach the performance goal is suboptimal mapping in the FPGA chip, which leads to relatively long routing delays with 69 % of the delay situated in routing.

Acknowledgements

Throughout this project there are many individuals who have supported me and the work I have done but some of them deserve extra credit.

First and foremost I would like to thank my supervisor, Lars Svensson, whose office door has always been open for me and who has supported me and the work diligently.

Secondly, a thank you to Christoffer Fougstedt who has kept me company throughout most of the work, and although working on his own thesis has discussed problems and solutions with me when I have needed.

Thank you also to Thomas Eriksson and Christian Fager for discussions about MIMO system specifications, filter characteristics for the transmitters, and $\Delta\Sigma$ optimization.

Thank you Ioannis Sourdis for lending me the FPGA card used in the project. Thank you Mary Sheeran for interest and discussion about the FPGA mapping issues. Thank you Per Larsson-Edefors for lending me the lab room where I spent phase one last summer.

Finally, thank you Ulrika Lindgren for your support throughout my entire thesis work.

List of Figures

Abbreviations

BP	Band-pass
$\Delta\Sigma$	Delta-sigma
CIFF	Cascade-of-integrators, feed-forward
CSD	Canonical signed digit
HP	High-pass
LO	Local oscillator
LP	Low-pass
MIMO	Multiple-in-multiple-out
MLM	Multiplier-less multiplication
OSR	Oversampling ratio
PA	Power amplifier
P$\Delta\Sigma$M	Parallel delta-sigma modulator
PE	Processing element
PISO	Parallel in, serial out
RF	Radio frequency
RoF	Radio-over-fiber
SCM	Single constant multiplication
SNDR	Signal-to-noise and distortion ratio

SNR	Signal-to-noise ratio
SPΔΣM	Speculative pseudo-parallel delta-sigma modulator
SQNR	Signal-to-quantization-noise ratio

Chapter 1

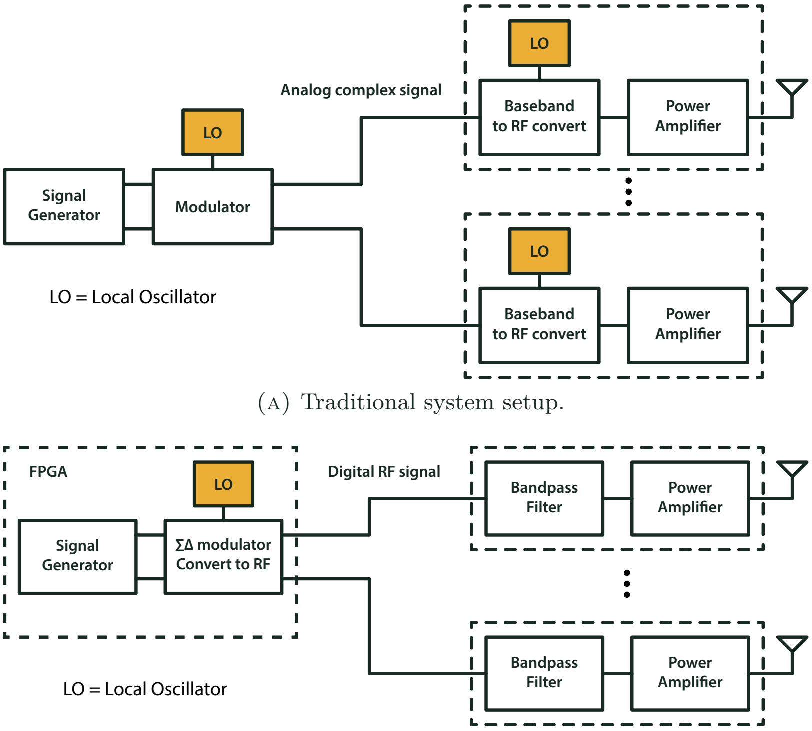
Introduction

Digital communication using Multiple-Input-Multiple-Output (MIMO) antenna technology has emerged as a significant technological breakthrough in modern communications [1]. A MIMO system uses multiple antennas at both the transmitting and the receiving end, and combines the received signals in such a way that the bit-error rate or data rate is improved. A key feature of MIMO is the ability to turn multi-path propagation, which is typically a complication in wireless communication, into a benefit [1]. The possibility for an improvement in wireless performance at no extra cost in frequency spectrum has led to MIMO getting increased attention in both research and industry.

Research into MIMO wireless systems requires the construction and assembly of actual test benches in order to verify the theoretical findings. In such systems, several transmitted signals must be generated, with highly accurate phase alignment. The straightforward approach would be to have the transmitter architecture include a separate local oscillator (LO) for each transmission antenna, located with the power amplifier [2] (see Fig. 1.1a), which would however make the necessary phase alignment difficult to achieve. The separate local oscillators will suffer from random phase noise; this will limit and eventually squander the MIMO gain [3].

In order to retain the MIMO gain, it is necessary to control the mutual phase noise between the generated signals. An intriguing possibility is to generate waveforms digitally via a delta-sigma ($\Delta\Sigma$) procedure, transfer the resulting bitstream to the PA/antenna via optical fiber, and then simply filter the bitstream to retrieve the desired signal for subsequent power amplification. Thus all transmitted signals will be derived from the same LO (see Fig. 1.1b).

This work aims to investigate how such a modulator can be implemented in a high-end FPGA development board by utilizing a high-speed parallel to serial (PISO) transceiver internal to



(A) Traditional system setup.

(B) Proposed system setup.

Figure 1.1: Block diagrams showing a straightforward system setup (A) in which the upconversion to RF takes place at the location of the PA and antenna using a local oscillator, together with the proposed system setup (B) where the upconversion is performed at waveform generation and distributed to the PA/antennas using optical fiber.

the FPGA chip.

1.1 Aims and goals

This project aims to investigate the viability of the proposed system setup in two steps. The first step comprises designing and building a single 1 GHz waveform generator. In the second step, multiple waveform generators will be included on the same FPGA board, using the same local oscillator.

Goal no. 1: Deciding upon an efficient and accurate implementation of the $\Delta\Sigma$ modulator in a general FPGA environment, assuming the presence of a serial optical-output capable of the required transmission speeds.

Goal no. 2: Design and implement a single $\Delta\Sigma$ modulator capable of providing sufficient SNDR (45 dB) for the bandwidth and frequency defined by the needs of the MIMO algorithm and with a static phase difference of at most 45°.

Goal no. 3: Combine several $\Delta\Sigma$ modulators in a single FPGA using the same local oscillator while keeping the phase-alignment.

Goal no. 4: Evaluate the system performance and resource economy.

1.2 Scope and limitations

This project will cooperate with another project at MC^2, which focuses on the low-pass filter, PA, and antenna design; therefore these parts are excluded from the scope of the thesis project.

Determining and generating the signals related to the MIMO algorithm as well as signal contents will also be considered out-of-scope for this project.

1.3 Thesis outline

This report starts with an account of the method that was initially established at the start of the thesis project, then continues with some background information of the major parts of the proposed system setup. After these chapters the theory chapter will describe the theory behind the design, which is presented in the design chapter. These two chapters are the bulk of this report. Lastly, there is a discussion that focuses on evaluating the project as well as the design, together with an account of suggested future work and a conclusion.

Chapter 2

Method

This project was planned in two phases, the first one focusing on finding, implementing, and evaluating a method for generating the high-frequency digital bitstream required for the proposed system setup. The second phase was planned as a design re-spin on the system together with evaluation, and report writing.

2.1 Prestudy

In order to design and implement a $\Delta\Sigma$ modulator suitable for the proposed system setup, knowledge of the design methodology and different flavors of $\Delta\Sigma$ modulators are necessary. To fill the gaps in knowledge a prestudy was planned as the first project activity. In parallel to this the intention was to examine the available FPGA development board and bring it operational.

2.2 Simulation

After completing the prestudy, the plan was to create and evaluate simulation models of plausible $\Delta\Sigma$ modulators with MATLAB. The purpose of these simulations was to arrive at a $\Delta\Sigma$ modulator design that performs sufficiently well in order to achieve the tentative system specifications. Subject for evaluation are primarily attainable SNDR, operating frequency, bandwidth, oversampling rate (OSR), and stability. A utility for facilitating these simulations is the $\Delta\Sigma$ toolbox created by Richard Schreier [4].

2.3 Initial implementation

After choosing a $\Delta\Sigma$ modulator design, the intention was to implement it in the FPGA chip using Xilinx ISE 14.3. The ISE could also used for extensive evaluation of the critical paths of the design in order to facilitate the planned design re-spin in phase two. The intention was to bring the FPGA PISO module operational in this stage as well. This stage was planned to be the last stage in the first phase of the project.

2.4 Design re-spin

Planning of the second phase was intended to be carried out in the beginning of the second phase because the two phases was separated by six months. The general plan was for the second phase to encompass a design re-spin of the project, with a focus to improve the performance of the designed modulator.

2.5 Evaluation and verification

During phase two, we also planned for an evaluation of the design using Xilinx ISE v14.3 for evaluating used resources and PlanAhead v14.3 for evaluating mapping and critical paths. In order to evaluate and verify the functional parameters of the design, the intention was to simulate synthesized versions of the design and then analyze the result of these using Matlab.

2.6 Project closure

When four weeks remained of the project, the plan was to have finished all technical aspects of the design and evaluation in initiate writing of the report.

Chapter 3

Background

The background knowledge necessary for the concept of this project consist mainly of knowledge about $\Delta\Sigma$ modulators, radio-over-fiber, and MIMO antenna systems. This chapter aims to provide a brief insight into these areas in order to facilitate the reasoning performed later in this report.

3.1 Delta-Sigma modulators ($\Delta\Sigma$Ms)

The idea of using feedback to improve the accuracy of data converters is not a new one, it has been around for about 60 years [5]; the noise-shaping concept was probably first proposed in 1962 by Inose et al. [6]. The system they proposed, however, achieved only 40 dB SNR and a signal bandwidth of 5 kHz and was implemented using a Schmitt trigger as a quantizer and a continuous-time integrator as a loop filter. Due to the fact that digital circuitry was very costly at the time, the trade-offs between analog accuracy and higher speed (and more digital hardware) was not very attractive at the time.

As Moore's Law continued to reduce the cost and disadvantages of digital circuitry, the amount of research in the field expanded. Higher-order loop filters was proposed in 1974 [5, 7] and from there on research in the field exploded. Richard Schreier emerged as one of the leading researchers on $\Delta\Sigma$-devices and has amongst many articles written two books on the subject [5,8], as well as produced the Matlab toolbox that will be partly used in this project [4]. In an article published in 1993 [9], Schreier presents an empirical study of high-order $\Delta\Sigma$ modulators as well as the problem with analytically determining their stability.

Bandpass $\Delta\Sigma$ modulators emerged in the late 1980s [5, 10, 11] and their use for narrowband RF signals has been established and documented by amongst others Keyzer et al. [12] and Jayaraman et al. [13]. Implementation of $\Delta\Sigma$ modulators in FPGAs is also well documented and there exists documented approaches to efficient implementation considerations and schemes [14–16].

3.2 Multiple-Input-Multiple-Output (MIMO)

The use of MIMO in wireless systems has gained increased popularity due to its performance-enhancing capabilities. Traditionally, communication in wireless channels is predominantly impaired by multi-path fading [1, 2] in which the arrival at the receiver of a transmitted signal differs in space and/or frequency and/or time. This fading can severely affect the reliability and quality of wireless communications [2].

MIMO technology offers a number of benefits that help meet the challenges presented by both impairments in the wireless channel as well as resource constraints [2]. Apart from the time and frequency dimensions that are exploited in conventional single-antenna systems [1], the leverages of MIMO are realized by in addition to these exploit the spatial dimension as well [2].

3.3 Radio-over-Fiber in MIMO systems

The increasing amount of wireless devices has led to a requirement for increased capacity in wireless systems [17, 18]. Radio-over-Fiber (RoF) centralizes most of the transceiver functionality in wireless systems by transmitting the wireless signals in a modulated state over optical fiber. This reduces the fielded access points to antennas with amplifiers and frequency converters [18]. As this centralization reduces the complexity of access points, the access points can be produced at a lower price. However, as the optical connections are more complex than a regular electrical connection, this adds to the complexity and cost of the central unit.

Some research has already been made in which the benefits of RoF are taken advantage of in MIMO systems. Liu et al. proposed in 2008 a method for transmitting MIMO radio signals over a single fiber using a phase quadrature double sideband frequency translation technique [19]. Seeds et al. published in 2009 a summary of the latest technologies for sending radio over fiber [20], including a technique similar to the one proposed by Liu et al. Chowdhury et al. demonstrates a novel broad spectrum band-shifting technique for simultaneously transmitting broadband multi-service, multi-carrier MIMO signals for in-building RoF based wireless systems in a proceedings article from 2010 [21].

Chapter 4

Theory

The theoretical background of this project consists of $\Delta\Sigma$ design theory and methods for reducing complexity in digital signal processing. Firstly, I present a theoretical introduction to $\Delta\Sigma$ modulators focusing on the impact of oversampling rate, after which I describe a selection of methods for parallelizing $\Delta\Sigma$ modulators with a large focus on the Hatami method. Single constant multiplication and LP-to-HP transformation are described as means for reducing processing complexity.

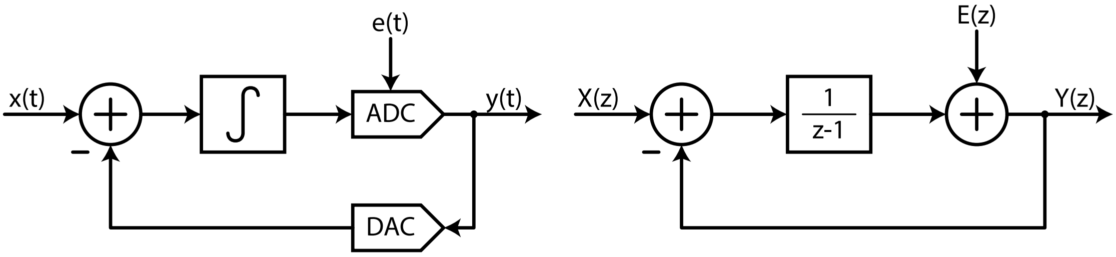
4.1 $\Delta\Sigma$ modulators

$\Delta\Sigma$ modulators are available in a large variety of different structures and implementations, but the underlying principle is the same for all of them. The general structure of an ADC $\Delta\Sigma$ converter is shown in Fig. 4.1a. The input to the integrator is formed by subtracting the quantized output signal $y(t)$ from the input signal $x(t)$. The quantization noise added in the quantization process is the additive term $e(t)$. This error is accumulated in the integrator and then quantized by a two-level quantizer.

If the ADC in Fig. 4.1a is replaced by its linear model, the linear system (with sampled data) in Fig. 4.1b is obtained. Simple analysis of this system gives

$$y[n] = x[n-1] + e[n] - e[n-1] \tag{4.1}$$

It is clear from this that the output signal contains a delayed but otherwise intact replica of the input signal ($x[n-1]$), and a differentiated version of the quantization error e. This



(A) Time domain.

(B) z-domain.

FIGURE 4.1: The general structure of a first order ΔΣ modulator represented in both the time- (A) and z-domain (B).

differentiation of e results in it being suppressed for frequencies that are small compared to the sampling frequency of the system f_s. If the loop filter has high gain in the signal band the in-band quantization noise is strongly attenuated. It is this process that is called *noise shaping* [5].

As Eq. (4.1) shows, the output noise created by the quantization error is $q[n] = e[n] - e[n-1]$. z-transformation of this yields

$$Q(z) = E(z) - z^{-1}E(z) = (1 - z^{-1})E(z) \tag{4.2}$$

and after substituting z for $e^{j2\pi f T_s}$ the *power spectral density* (PSD) [5] of the quantization noise is

$$S_q(f) = (2\sin(\pi f T_s))^2 S_e(f) \tag{4.3}$$

where $T_s = 1/f_s$ is the sampling period and $S_e(f)$ is the one-sided PSD of the quantization error of the quantizer. For rapidly varying input signals it is possible to approximate the quantization error with white noise of mean-square value $e^2_{\text{rms}} = \Delta^2/12$ where Δ is the step size of the quantizer [5]. This means that

$$S_e(f) = \frac{\Delta^2}{6f_s} \tag{4.4}$$

In Eq. (4.2) the filtering function $(1 - z^{-1})$ is denoted the *noise transfer function* (NTF) and has high-pass (HP) characteristics. The consequences of this is that it suppresses e at low frequencies, but it also enhances e at frequencies around $f_s/2$.

This characteristic can be taken advantage of by oversampling the system. The *oversampling rate* (OSR) describes how many times higher the sampling frequency in the oversampled modulator is compared to a Nyquist-rate modulator and is calculated as

$$\mathrm{OSR} = \frac{f_s}{2f_B} \tag{4.5}$$

where f_B is the maximum desired signal frequency.

Integration of $S_q(f)$ from 0 to f_B results in the in-band noise power, which by assuming an $\mathrm{OSR} \gg 1$ can be approximated [5] with

$$q_{\mathrm{rms}}^2 = \frac{\pi^2 e_{\mathrm{rms}}^2}{3(\mathrm{OSR})^3} \tag{4.6}$$

This indicates that it is possible to decrease the in-band noise by increasing the OSR. The decrease in in-band noise by increasing OSR is relatively small however, as a doubling of the OSR only reduces the in-band noise by 9 dB, which corresponds to an increase in effective number of bits by about 1.5 as $\mathrm{SNR} = 6.02 \cdot \mathrm{ENOB} + 1.76 \Leftrightarrow \mathrm{ENOB} = (\mathrm{SNR} - 1.76)/6.02$.

4.1.1 Second-order loop

A way of increasing the SNR and thus the ENOB of the modulator is to implement a higher-order loop filter. By adding one more integrator and feedback path to the diagram of Fig. 4.1 a diagram which gives

$$Y(z) = X(z) + (1 - z^{-1})^2 E(z) \tag{4.7}$$

is obtained (see Fig. 4.2). The NTF is now $(1 - z^{-1})^2$ instead of $(1 - z^{-1})$ and the shaping function applied to e becomes $(2\sin(\pi f T_s))^4$ instead, which shapes the noise more aggressively than the first order loop does. By the same approximation as before it follows that the in-band noise power is

$$q_{\mathrm{rms}}^2 = \frac{\pi^4 e_{\mathrm{rms}}^2}{5(\mathrm{OSR})^5} \tag{4.8}$$

A doubling in OSR now results in an increased ENOB by 2.5 instead of 1.5.

4.1.2 Higher-level loop

In the same way as before, the loop order can be increased even further by continuing to add integrators and feedback branches to the loop. For an L^{th}-order loop with an $\mathrm{NTF} = (1 - z^{-1})^L$

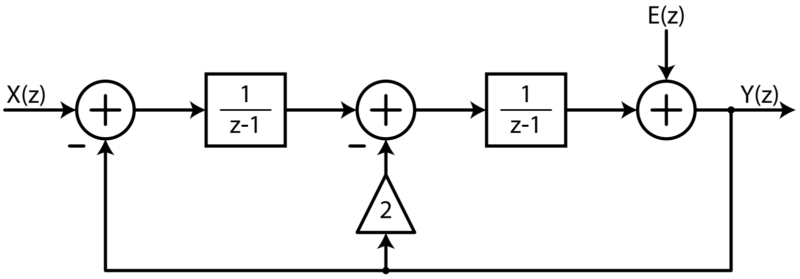


FIGURE 4.2: The structure of a second order ΔΣ modulator represented in the z-domain.

the power of the in-band noise is approximately [5]

$$q_{\mathrm{rms}}^2 = \frac{\pi^L e_{\mathrm{rms}}^2}{(2L+1)(\mathrm{OSR})^{2L+1}} \tag{4.9}$$

and subsequently the number of bits added to the resolution by doubling the OSR is $L + 0.5$. However, for high-order loops there are stability considerations that reduce the achievable resolution to lower values than given with the previous equations [5].

4.2 Parallel ΔΣ modulators

Parallelism can be used in ΔΣ modulators to improve conversion performance, i.e. by increasing conversion bandwidth or reducing the amount of oversampling needed. Several methods for parallelizing a ΔΣ modulator have been proposed [22–26] and this section briefly describes some of the methods and focuses on the one proposed by Hatami et al.

The simplest parallel ΔΣ modulator (PΔΣM) is obtained by placing analog-to-digital ΔΣ modulators in parallel and applying the same input value to all of them [22]. The output is then recombined on the digital side for an increase in resolution. This yields a 0.5 bits [22] increased resolution per doubling of the number of ΔΣ modulators in parallel.

4.2.1 Frequency-band-decomposition PΔΣM

Much in the same manner that filter banks are utilized for increasing bandwidth of filters, the concept can be used to derive the frequency-band-decomposition PΔΣM. Each channel converts a smaller band of frequencies and when the output signals are combined the system is all-pass [26, 27].

Each channel consist of a ΔΣ modulator that converts a part $1/M$ (where M is the number of channels) of the total required bandwidth. This is followed by a digital filter that passes the frequencies that are nulled by the NTF of the preceding ΔΣ modulator. The quantization noise of the different channels is uncorrelated and thus the SNR of the system is [26]

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{P_s}{\sum P_{n_i}}\right) \tag{4.10}$$

where P_s is the signal power and P_{n_i} is the quantization noise in the i-th channel. For a given modulator order L used the resolution is increased by L bits [22].

The foremost challenge in designing a frequency-band-decomposition PΔΣM is that every channel is unique and thus requires different ΔΣ modulators and digital filters on each channel. However, this approach is also less sensitive to channel mismatch than the modulation-based PΔΣM [22].

4.2.2 Modulation-based PΔΣM

A modulation-based PΔΣM achieves wide-band conversion by converting a complex set of frequencies on each channel [22, 25]. It is named *modulation-based* because both the input signals and output signals are modulated by an external signal [25]. For each channel, the modulation signal is a row in a modulation matrix, where each column is a time instance. The general criteria for this modulation matrix is that it is a unitary matrix [22].

Modulating the input signal at each channel results in the signal and quantization noise effectively becoming decoupled. By using this property, a filter (placed on each channel after the ΔΣ modulator) can be designed to pass the signal but filter the quantization noise [22, 25]. Before obtaining the final output, the signal on each channel is demodulated and recombined into one signal.

A few examples of matrices that are viable for use in modulation-based PΔΣM include the simple identity matrix, the DFT matrix, and the Hadamard matrix. With the identity matrix follows that only one channel will process an input value at any given clock cycle, meaning that the PΔΣM is time-interleaved. Out of these alternatives, the Hadamard modulated PΔΣM provides the best performance but requires large digital filters [22].

4.2.3 Hatami ΔΣ modulator structure

The ΔΣ modulator structure proposed by Hatami et al. [24] works by unrolling the ΔΣ loop and calculating N consecutive output values for one input value. As described in Section 4.1, ΔΣ modulators traditionally relies on a high OSR in order to create a high-SNR in-band signal whereas the Hatami structure also allows trading loop-unrolling for SNR. A factor 2 in OSR provides the same gain in SNR as unrolling the ΔΣ loop by a factor 2 [24].

The general structure of the Hatami ΔΣ modulator can be seen in Fig. 4.3. The blocks that are named PE with a number subscript are processing elements which compute the output values $\hat{y}[n]$, $\hat{y}[n+1]$, $\hat{y}[n+2]$, and $\hat{y}[n+3]$, which are subsequently quantized into $y[n]$, $y[n+1]$, $y[n+2]$, and $y[n+3]$. The processing element denoted *update* re-evaluates the internal accumulator states of the ΔΣ modulator and distributes these to the other processing elements in order to enable processing of the next input value $x[n+1]$.

The effective sampling frequency (the frequency of the output bits if viewed as a bitstream) of the output is

$$f_s' = N \cdot f_s \tag{4.11}$$

where f_s is the sampling frequency of the input signal and N is the unrolling factor.

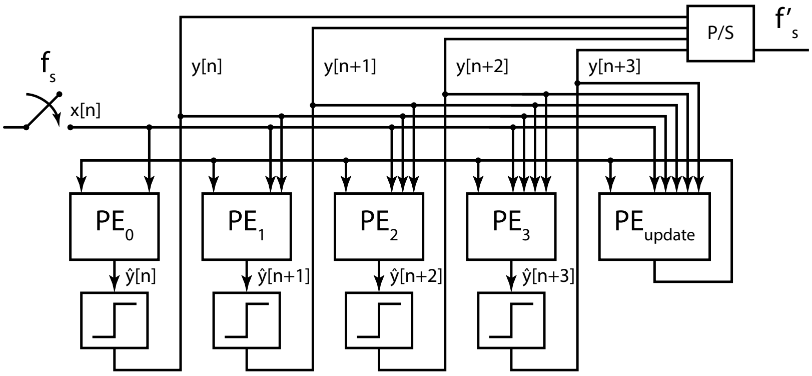


FIGURE 4.3: The general structure of a Hatami ΔΣ modulator structure. This figure shows a modulator unrolled four times, producing the output values $y[n]$, $y[n+1]$, $y[n+2]$, and $y[n+3]$ for the input value $x[n]$. For this amount of unrolling, $f_s' = 4f_s$.

Description of a third-order, four-unrolled ΔΣ modulator

As an example of the method for unrolling ΔΣ loops, the third-order modulator structure that is later on chosen for this project is unrolled four times. This description is in large an adaptation of Hatami et al. 2014 [24].

Fig. 4.4 shows a general third-order CIFF (cascade of integrators, feed-forward) LP ΔΣ modulator before unrolling. As the structure is unrolled $N = 4$ times, the input value $x[n]$ remains consistent for 4 clock cycles, i.e $x[n] = x[n+1] = x[n+2] + x[n+3]$ and n is a multiple of four.

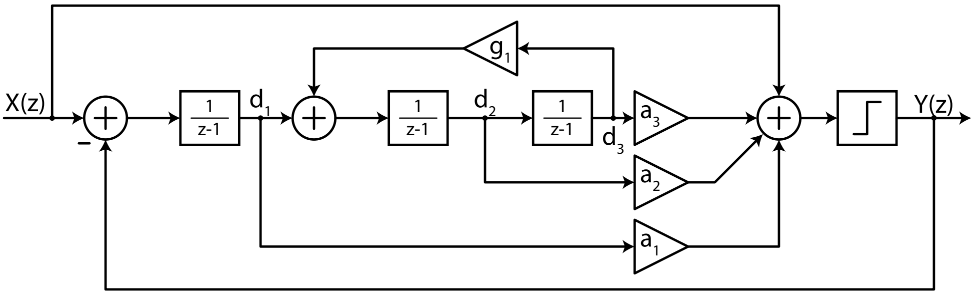


FIGURE 4.4: A parametrized linear z-domain model of a third-order LP CIFF ΔΣ modulator.

From Fig. 4.4 it is easy to derive the relations

$$\begin{aligned} d_1[n+1] &= d_1[n] + x[n] - y[n] \\ d_2[n+1] &= d_1[n] + d_2[n] + g_1 d_3[n] \\ d_3[n+1] &= d_2[n] + d_3[n] && (4.12) \\ \hat{y}[n] &= a_1 d_1[n] + a_2 d_2[n] + a_3 d_3[n] && (4.13) \\ y[n] &= Q(\hat{y}[n]) && (4.14) \end{aligned}$$

where

$$Q(\hat{y}[n]) = \begin{cases} 1 & \text{if } \hat{y}[n] \geq 0 \\ -1 & \text{if } \hat{y}[n] < 0 \end{cases} \tag{4.15}$$

As can be seen the output value $y[n]$ can be formed from the internal states $d_1[n], d_2[n], d_3[n]$ modified by the parameters a_1, a_2, a_3 and then quantizing it by extracting the sign. It is Eq. (4.13) that corresponds to the functionality of block PE_0 in Fig. 4.3. Eq. (4.14) is then the output value after the quantizer. The updates to the internal states $d_1[n+1], \ldots d_3[n+1]$ in Eq. (4.12) are needed in order to form the output and updated states in PE$_1$.

In order to calculate the next output value $y[n+1]$, the following processing is necessary

$$d_1[n+2] = d_1[n+1] + x[n+1] - y[n+1]$$

$$
\begin{aligned}
&= d_1[n] + 2x[n] - y[n] - y[n+1] \\
d_2[n+2] &= d_2[n+1] + g_1 d_3[n+1] + d_1[n+1] \\
&= 2d_1[n] + (g_1+1)d_2[n] + 2g_1 d_3[n] + x[n] - y[n] \\
d_3[n+2] &= d_3[n+1] + d_2[n+1] \\
&= d_1[n] + 2d_2[n] + (g_1+1)d_3[n] && (4.16) \\
\hat{y}[n+1] &= a_3 d_3[n+1] + a_2 d_2[n+1] + a_1 d_1[n+1] + x[n+1] \\
&= (a_2+a_1)d_1[n] + (a_3+a_2)d_2[n] + (a_3+g_1a_2)d_3[n] + (a_1+1)x[n] - a_1 y[n] \\
&= \underbrace{r_1 d_1[n] + q_1 d_2[n] + p_1 d_3[n] + s_{11}x[n]}_{\text{Parallel}} \underbrace{+ s_{12}y[n]}_{\text{Sequential}} && (4.17) \\
y[n+1] &= Q(\hat{y}[n+1]) && (4.18)
\end{aligned}
$$

In Eq. (4.17) it is clear that the coefficients that operate on the internal states can be combined into new coefficients in order to reduce complexity, resulting in

$$
\begin{aligned}
r_1 &= a_2 + a_1 \\
q_1 &= a_3 + a_2 \\
p_1 &= a_3 + g_1 a_2 \\
s_{11} &= a_1 + 1 \\
s_{12} &= -a_1
\end{aligned}
$$

The resulting equation in Eq. (4.17) becomes PE_1, and Eq. (4.18) is the output value after quantization. In the same manner as in Eq. (4.12), Eq. (4.16) is the updated states needed for the operation of PE$_2$.

In the same was as the resulting equations of PE$_0$ is used in PE$_1$, the resulting equations of PE$_1$ is used in the equations of PE$_2$.

$$
\begin{aligned}
d_1[n+3] &= d_1[n] + 3x[n] - y[n] - y[n+1] - y[n+2] \\
d_2[n+3] &= (g_1+3)d_1[n] + (3g_1+1)d_2[n] + (g_1^2+3g_1)d_3[n] + 3x[n] - 2y[n] - y[n+1] \\
d_3[n+3] &= 3d_1[n] + (3+g_1)d_2[n] + (3g_1+1)d_3[n] + x[n] - y[n] && (4.19) \\
\hat{y}[n+2] &= \underbrace{r_2 d_1[n] + q_2 d_2[n] + p_2 d_3[n] + s_{21}x[n]}_{\text{Parallel}} \underbrace{+ s_{22}y[n] + s_{23}y[n+1]}_{\text{Sequential}} && (4.20) \\
y[n+2] &= Q(\hat{y}[n+2]) && (4.21)
\end{aligned}
$$

where

$$\begin{aligned}
r_2 &= a_3 + 2g_1 + a_1 \\
q_2 &= 2a_3 + (g_1 + 1)a_2 \\
p_2 &= (g_1 + 1)a_3 + 2g_1 a_2 \\
s_{21} &= a_2 + 2a_1 + 1 \\
s_{22} &= -a_2 - a_1 \\
s_{23} &= -a_1
\end{aligned}$$

Analogous to the cases for PE$_0$ and PE$_1$, the result of PE$_2$ is given by Eq. (4.20) and the output of the quantizer by Eq. (4.21). For the last processing element the equations are

$$\begin{aligned}
\hat{y}[n+3] &= \underbrace{r_3 d_1[n] + q_3 d_2[n] + p_3 d_3[n] + s_{31} x[n]}_{\text{Parallel}} \\
&\quad \underbrace{+ s_{32} y[n] + s_{33} y[n+1] + s_{34} y[n+2]}_{\text{Sequential}} && (4.22) \\
y[n+3] &= Q(\hat{y}[n+3]) && (4.23)
\end{aligned}$$

where

$$\begin{aligned}
r_3 &= 3a_3 + a_2(g_1 + 3) + a_1 \\
q_3 &= a_3(3 + g_1) + a_2(3g_1 + 1) \\
p_3 &= a_3(3g_1 + 1) + a_2(g_1^2 + 3g_1) \\
s_{31} &= a_3 + 3a_2 + 3a_1 + 1 \\
s_{32} &= -a_3 - 2a_2 - a_1 \\
s_{33} &= -a_2 - a_1 \\
s_{34} &= -a_1
\end{aligned}$$

It is also clear now that all the significant computations for $\hat{y}[n+3]$ are dependent on values that are obtained at time n, meaning that the computations for $\hat{y}[n]$, $\hat{y}[n+1]$, $\hat{y}[n+2]$, and $\hat{y}[n+3]$ can be started at the same time. However, the computations in their entirety cannot be completed until the two-level values $s_{32}y[n]$, $s_{33}y[n+1]$, and $s_{34}y[n+2]$ are available. The parts that can be started at the same time are marked *parallel*, and the parts that need to wait for the previous PEs are marked *sequential*. Since the sequential values are two-level values, the results can be pre-calculated and multiplexed once the results of the previous PEs arrive.

This pre-calculation is not needed in PE$_0$ since it has no dependence on earlier values of y. For PE$_1$ there are two different possibilities, for PE$_2$ there are four possibilities, and for PE$_3$ there are eight possibilities. Fig. 4.5 illustrates the pseudo-parallel execution of the first three of the processing elements described above.

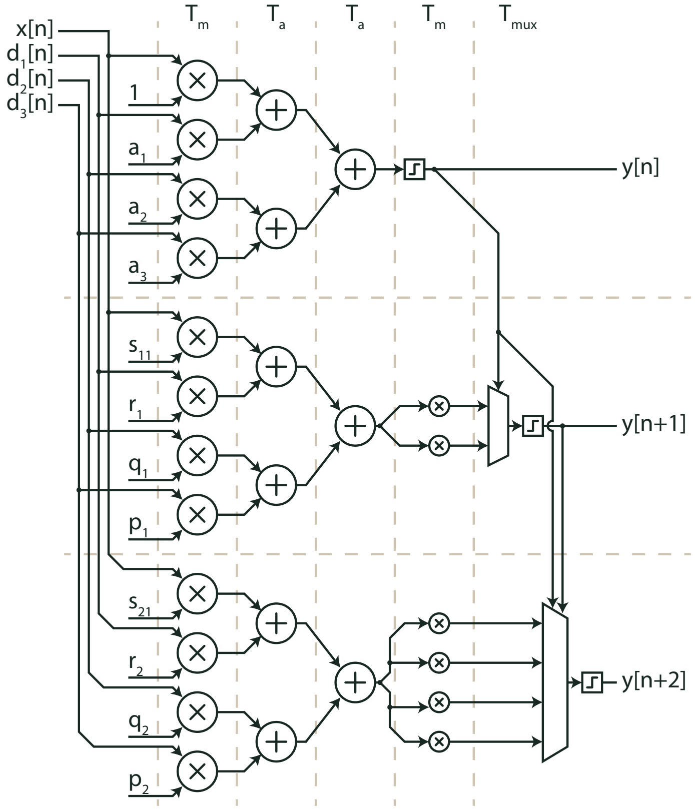


Figure 4.5: Functional illustration of the first three processing elements in the Hatami ΔΣ modulator structure for a third-order, four-unrolled LP ΔΣM in the CIFF structure. The figure shows that the most computation-intensive processing can be initiated simultaneously for the processing elements as the input values are available at the same time.

In Fig. 4.5 the total delay also becomes clear with the critical path spanning $T = 2T_m + 2T_a + T_{mux}$. If another processing element is included in the structure the delay remains the same, although the T_{mux} delay becomes longer. There are methods to exchange the T_m delays for T_a delays instead. One of these methods is described in Section 4.3.

The final step in order to prepare for the next clock cycle of f_s is to update the internal states with $d_1[n+4], \ldots d_3[n+4]$. These equations are derived from the basic expression in Eq. (4.12) for time $n+4$ and utilizing expressions from Eqs. (4.17), (4.20), (4.22).

$$\begin{aligned}
d_1[n+4] &= \underbrace{d_1[n] + 4x[n]}_{\text{Parallel}} \underbrace{-y[n] - y[n+1] - y[n+2] - y[n+3]}_{\text{Sequential}} \\
d_2[n+4] &= \underbrace{(4g_1+4)d_1 + (g_1^2+6g_1+1)d_2[n] + (4g_1^2+4g_1)d_3[n] + (6+g_1)x[n]}_{\text{Parallel}} \\
&\quad \underbrace{+(-3-g_1)y[n] - 2y[n+1] - y[n+2]}_{\text{Sequential}} \\
d_3[n+4] &= \underbrace{(g_1+6)d_1[n] + (4g_1+4)d_2[n] + (g_1^2+6g_1+1)d_3[n] + 4x[n]}_{\text{Parallel}} \\
&\quad \underbrace{-3y[n] - y[n+1]}_{\text{Sequential}} \qquad (4.24)
\end{aligned}$$

These equations are represented in the block denoted $\text{PE}_{\text{update}}$ in Fig. 4.3. The parts marked with *parallel* can be computed simultaneously as they only depend on the input values available at the start of f_s, while the parts marked with *sequential* relies on the output of the four PEs and are pre-calculated in the same manner as described in Fig. 4.5, and multiplexed when the output bits arrive.

A k-th order modulator unrolled N times

The derivation described in Section 4.2.3 can be generalized to a k-th order, N-unrolled PΔΣM [24]. $x(i)$ is the i-th element of the input sequence x. $\hat{y}$ and y remains the output of the ΔΣM before and after quantization. The array $[d]_{k\times1}$ contains the internal states of the modulator, where k is the order of the modulator. The ΔΣ modulator coefficients are described by the matrices A, B, and C [5, 8]. The equations describing a k-th order modulator that is fed with the i-th value of an input sequence is

$$[d(i+1)]_{k\times1} = [A]_{k\times k} \times [d(i)]_{k\times1} + [B]_{k\times2} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}_{2\times1} \qquad (4.25)$$

$$\hat{y}(i) = [C]_{1\times k}[d(i)]_{n\times1} + x(i) \qquad (4.26)$$

$$y(i) = Q(\hat{y}(i)) \qquad (4.27)$$

4.3 Single Constant Multiplication

A multiplication with a fixed-point constant can be implemented without using a multiplier, using only a sequence of adders (or subtracters) and arithmetical shifts. This is called multiplier-less multiplication (MLM). An example that demonstrates this property is $y = 8.75x$, which can be calculated as

$$\begin{aligned} y &= 8x + x - 0.25x \\ &= (2^3 + 1 - 2^{-2}) \cdot x \\ &= (x << 3) + x - (x >> 2) \end{aligned} \tag{4.28}$$

In this specific example the multiplier has been exchanged for two adder operations and two shift operations. If the multiplication is performed with the same constant each time, the shift operations is implemented solely through wiring.

The most straightforward method for deriving the way of multiplying with a constant C by using adders and shifts can be read straight from the binary representation of C. In case of $C_{10} = 10.875$, $C_2 = 1010.111$

$$\begin{aligned} y &= C \cdot x \\ y &= (2^3 + 2^1 + 2^{-1} + 2^{-2} + 2^{-3}) \cdot x \\ y &= (x << 3) + (x << 1) + (x >> 1) + (x >> 2) + (x >> 3) \end{aligned} \tag{4.29}$$

However, as the adders just as easily can be used as subtracters, a better way of finding a add/subtract/shift sequence is to recode the constant as a canonical signed digit (CSD). The constant in the above example can be written as $C_{\text{csd}} = +0{+}{+}.00\text{-}$, which corresponds to

$$\begin{aligned} y &= C_{\text{csd}} \cdot x \\ y &= (2^3 + 2^1 + 2^0 - 2^{-3}) \cdot x \\ y &= (x << 3) + (x << 1) + x - (x >> 3) \end{aligned} \tag{4.30}$$

which requires one adder less than the binary method. Finding an optimal solution for the MLM problem is an NP complete problem [28], however, for short lengths of the constant the CSD method is a near-optimal solution. There are other heuristic methods for determining a better near-optimal solution available but these are more complex [28, 29].

4.4 LP to HP transformation

In digital signal processing, the frequency spectra of a signal can be mirrored around π rad/sample by inverting the sign of every alternate value. This is useful because it enables processing of a signal at baseband frequency instead of carrier frequency. An example of this transformation can be seen in Fig. 4.6 where the output signal of a 5th order LP $\Delta\Sigma$ modulator with an OSR of 32 has been transformed into a HP signal.

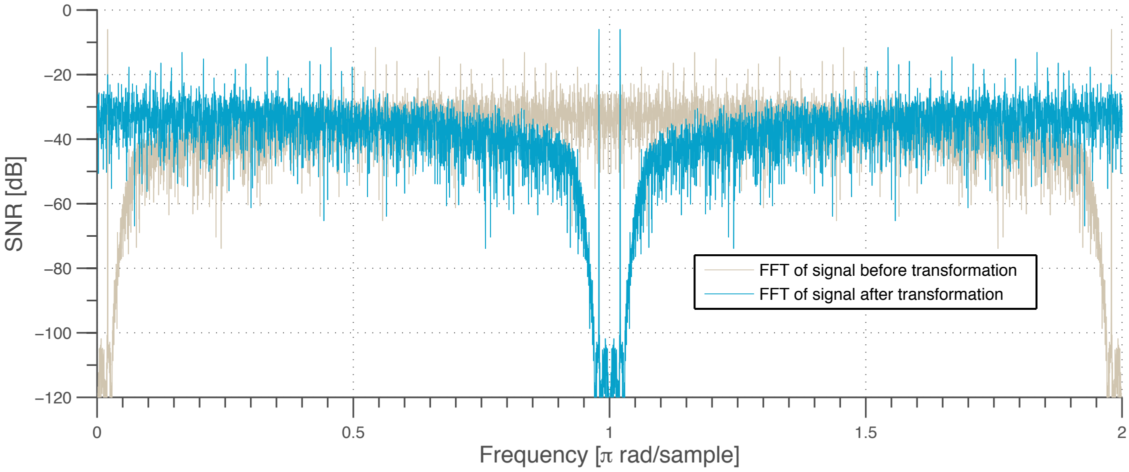


FIGURE 4.6: The frequency spectrum of the input signal has been mirrored around π by inverting the sign of every alternate value.

However, since this transformation mirrors the spectra around π rad/sample the first replica of the spectra lies close to the desired signal, increasing the demands on the band-pass filter of the transmission modules.

Chapter 5

Design

Designing a modulator suitable for the system specifications has been a process based on simulating, implementing, verifying, and iterating. This chapter describes the design process, as well as encountered problems and solutions. It ends by evaluating the performance and resource economy of the achieved implementation.

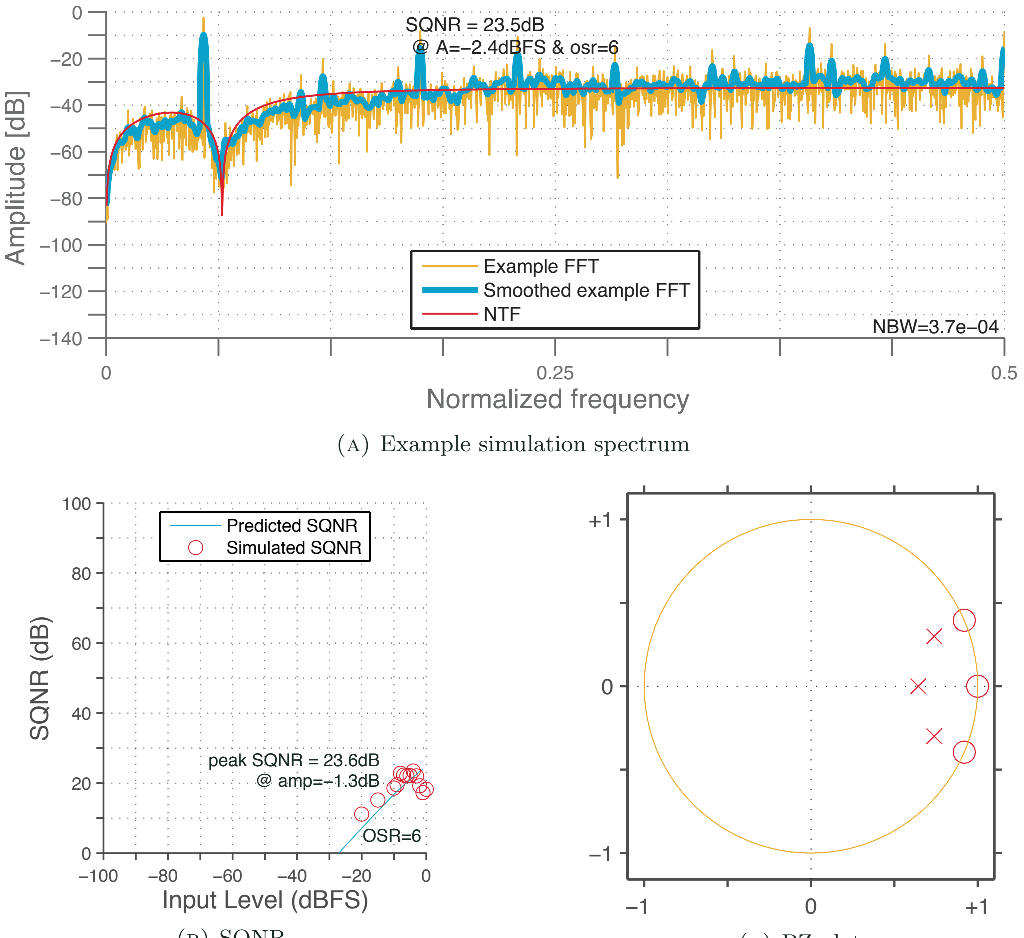
5.1 Architectural considerations

Delta-sigma modulators traditionally uses a high oversampling ratio in order to improve SNDR [5, 8]. Thus a high clock frequency is needed for the best performance. The FPGA board used in this project is based on a Virtex 6 LX550-T which has a maximum fabric speed of 600 MHz [30]. The highest OSR attainable in the system assuming a single-sided bandwidth $f_B = 50$ MHz is then

$$OSR = \frac{f_s}{2f_B} = \frac{600\text{ MHz}}{2 \cdot 50\text{ MHz}} = 6. \tag{5.1}$$

The SNDR that is possible to achieve with an OSR of six is less than the required in-band SNDR of 45 dB [5, 8, 9]. With a third-order LP $\Delta\Sigma$ modulator with optimized zeros the achievable in-band SNDR is only 23.6 dB as can be seen in Fig. 5.1.

Since the fabric of the FPGA itself is limited to 600 MHz the OSR cannot be increased further and it is clear that the traditional approach to a $\Delta\Sigma$ modulator is insufficient.



(A) Example simulation spectrum

(B) SQNR

(C) PZ-plot

FIGURE 5.1: Graphs showing the attainable performance from a third-order LP ΔΣ modulator with optimized zeros working at an OSR of six.

5.1.1 Parallel ΔΣ modulators

It is desirable to be able to use a carrier frequency that exceeds the FPGA clock frequency. A traditional ΔΣ modulator generates only a single output value in each clock cycle, and thus, parallelization is needed in order to reach higher output bitstream frequencies. A number of methods for parallelizing a ΔΣ modulator have been proposed [22–27], but most of them are unsuitable for the current application as they yield Nyquist-rate, single-bit-output modulators. The exception is the architecture proposed by Hatami et al. [24] in which a ΔΣ modulator is unrolled and produces multiple consecutive output-bits in each cycle. The methodology for unrolling a ΔΣ modulator is described in Section 4.2.3. This characteristic fits well with the high-speed PISO transceiver found in most high-end FPGA chips, enabling the carrier frequency to be higher than the fabric speed of the FPGA. The PISO present in the Virtex 6 LX550-T is capable of outputting a bitstream at 6.6 Gbps [30].

5.2 Low-pass, band-pass, or high-pass $\Delta\Sigma$ modulators

Apart from the straightforward approach of having a LP $\Delta\Sigma$ modulator it is also possible to design a HP or BP modulator. These work similarly to the LP $\Delta\Sigma$ modulator but instead of shaping the noise towards higher frequencies, they shape the noise either towards lower frequencies (high-pass) of towards the sides of the spectrum (band-pass). The design of a HP or BP $\Delta\Sigma$M is, however, more complex than that of a LP $\Delta\Sigma$M [10, 13, 15, 31]. As the system is already limited in operating frequency by the FPGA fabric, the BP $\Delta\Sigma$M is unsuitable. The HP $\Delta\Sigma$M has the usable property of producing a spectrum with its in-band frequency range centered around the Nyquist frequency, which is where we want the signal content to be. However, it also requires the input signal sequence to be generated at higher frequencies.

An alternative to the HP $\Delta\Sigma$M is to design a LP $\Delta\Sigma$M and perform a LP-to-HP transformation on the output sequence. This method is used for the design and it is described in Section 4.4. It is implemented by simply inverting every other bit in the output signal sequence.

5.3 Design choices

As always, $\Delta\Sigma$ design is a trade-off between demands on SNDR, OSR, and loop order [5, 8]. Based on the tentative system specifications, we chose a third-order low-pass $\Delta\Sigma$ modulator with a single-sided bandwidth of 50 MHz in a CIFF structure which we unrolled four times and operate at an OSR of four. The third order is necessary in order to achieve an SNDR higher than 45 dB, but as can be seen in Fig. 5.2, increasing the order further has provides only a very small performance improvement but add complexity. As seen in Fig. 5.3, the modulator has an in-band SNDR of approximately 52.2 dB. The simulations have been performed using Schreiers $\Delta\Sigma$ toolbox [4].

The operating frequency that the modulator needs to operate at can be calculated as

$$f_s = \mathrm{OSR} \cdot 2f_B \cdot C_{OH} = 4 \cdot 100 \text{ MHz} \cdot 1.25 = 500 \text{ MHz} \tag{5.2}$$

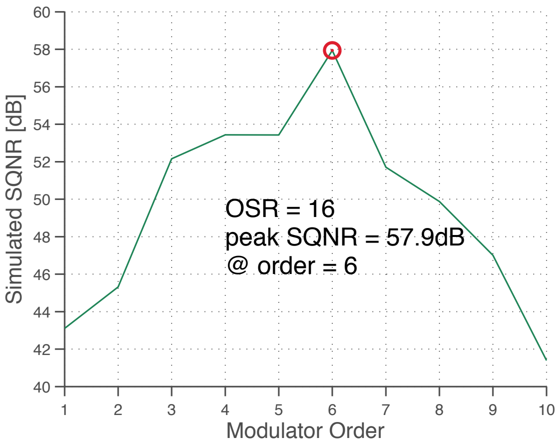


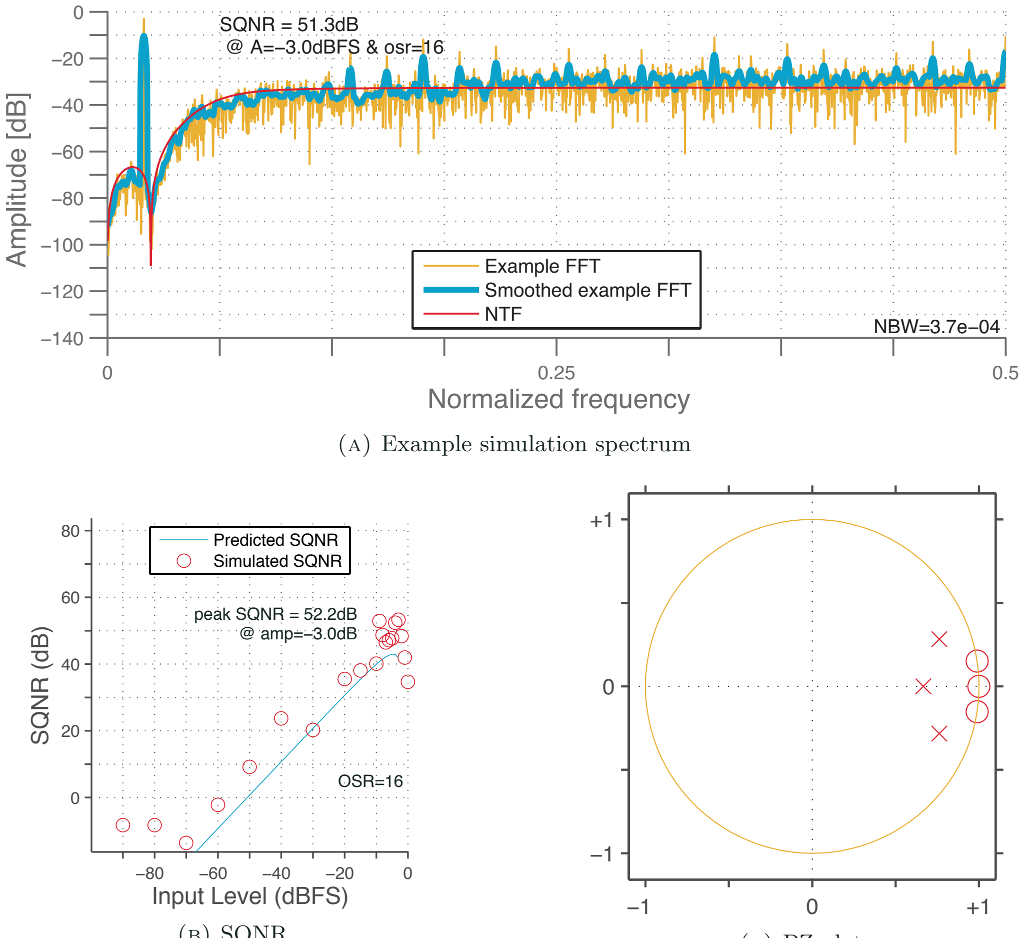
Figure 5.2: Simulation of the achievable SNQR of a CIFF $\Delta\Sigma$ modulator working with an oversampling rate of 16 for different orders of the $\Delta\Sigma$ loop. The loop filter has optimized zeros [5].

and the output bitstream sampling frequency is

$$f_s' = f_s \cdot N = 500 \text{ MHz} \cdot 4 = 2 \text{ GHz} \tag{5.3}$$

where f_B is the required single-sided bandwidth, C_{OH} is the frequency spectrum transition region needed for the system as a fraction of the f_B, and N is the amount of unrolling applied to the $\Delta\Sigma$ loop. For this system, the C_{OH} is set to 25% to allow for subsequent band-pass filtering at the transmission antennas. A representation of the chosen modulator before unrolling can be seen in Fig. 5.4a. In order to move the frequency content into HF range, every other bit in the output bitstream is inverted in order to transform the LP characteristic into a HP representation. This procedure is further described in Section 4.4. Fig. 5.4b shows a simulation of the chosen modulator before unrolling but at the effective OSR of 16, confirming that the criteria for SNDR and bandwidth are fulfilled.

As can be seen in Fig. 5.5, which shows the unrolled Hatami modulator structure, the unrolling has an unfortunate side-effect: it lengthens the critical path of the $\Delta\Sigma$ modulator. As performance is key to this architecture, reducing the critical path is essential when implementing the modulator. The P$\Delta\Sigma$M consists of four processing elements (PE) and one block for updating the internal states of the P$\Delta\Sigma$M. These all incorporate multiplications with constants which are cumbersome and unwanted in the critical path. By modifying the coefficients into values that can be represented by short binary words and at the same time sacrifice some accuracy in the placement of the poles [5,8,9] for the P$\Delta\Sigma$M, these multiplications can each be reduced



(A) Example simulation spectrum

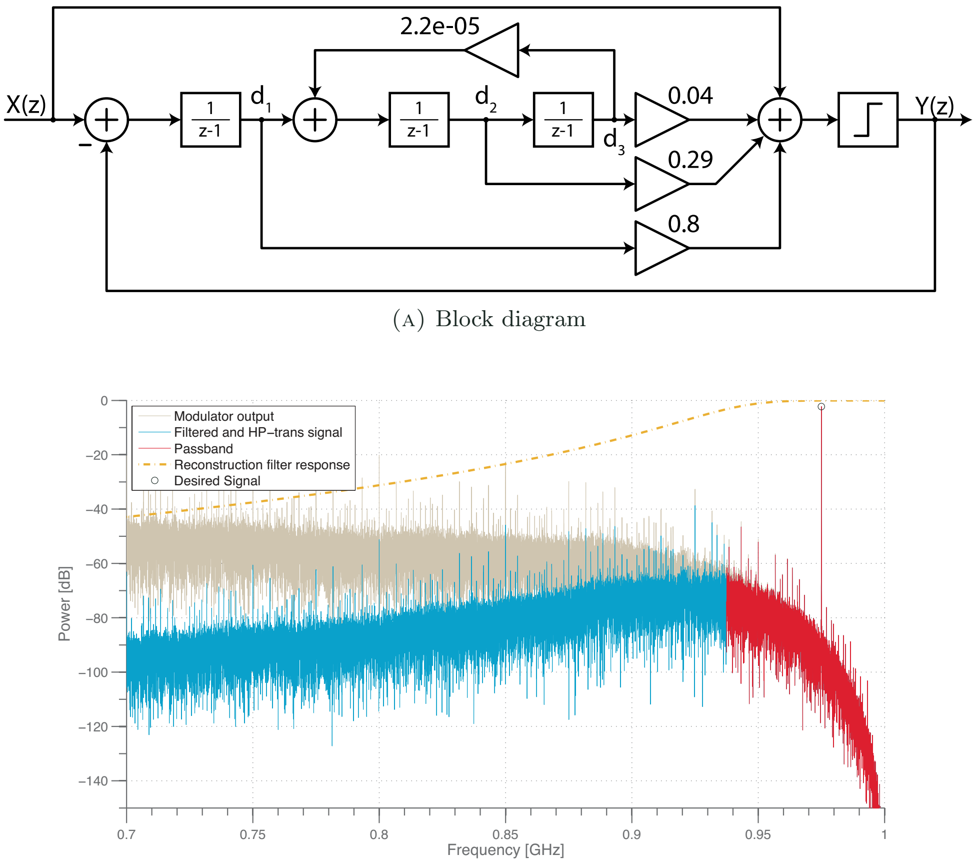
(B) SQNR

(C) PZ-plot

FIGURE 5.3: Graphs showing the attainable performance from a third-order LP $\Delta\Sigma$ modulator with optimized zeros working at an OSR of 16.

to a single addition and an arithmetic shift of the operands in the Hatami modulator. This method is called single constant multiplication and is further described in Section 4.3. This significantly shortens the critical path of the P$\Delta\Sigma$M.

Fig. 5.5 also illustrates that each PE depends on the result of all previous PEs. Since the result of a PE is a single bit, there are only two possible outcomes for each block. As the PEs all get access to the same information (excluding the result bits of the previous PEs) at the beginning of each clock cycle, the possible outcomes of the present PE calculations can be pre-calculated in parallel to the rest of the calculations, in turn reducing the response to the PE results to a choice in a multiplexer, in the manner of a carry-select adder. This is illustrated in Fig. 4.5 in



(A) Block diagram

(B) Simulation spectrum

FIGURE 5.4: Block diagram (A) and simulation spectrum (B) of the chosen third-order LP ΔΣ modulator at an effective OSR of 16, after LP-to-HP transforming the output.

Section 4.2.3.

5.4 SPΔΣM: Speculative Architecture

Preliminary verifications of the implemented PΔΣM showed that even though the Hatami modulator architecture shows great potential for use with the proposed system setup, the critical path limits the operating frequency so that it does not meet the performance requirements. This is due to inefficient mapping of the structure performed by the place-and-route toolset used. The heuristic methods used results in an operating frequency of only 152 MHz, with approximately 75 % of the delay placed in net. Fortunately, the modulator only occupies 0.1 %

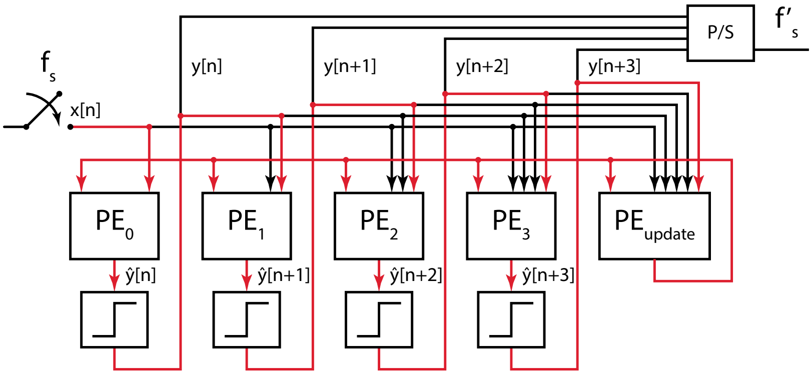


Figure 5.5: Block diagram of a general Hatami modulator structure for a modulator unrolled 4 times with the critical path hilighted.

(specifically 476 of 343680 available LUTs) of the resources in a Virtex 6 LX550-T FPGA chip. Hence, a new structure is proposed that makes use of the excess resources in the FPGA chip in order to further reduce the critical path.

The primary goal of the speculative PΔΣM (SPΔΣM) is to remove the time waiting for the result of previous PEs *completely*, by assuming that this previous result is already known. This removes the multiplexer from the critical path at the expense of having to duplicate the modulator for each possible output bitstream outcome. A second goal of the SPΔΣM is to separate computing output bits from updating the internal states of the modulator, thereby allowing the next input value ($x[n+1]$) to be processed before the current one ($x[n]$) is finished.

5.4.1 Component descriptions

The speculative version of the aforementioned third order LP PΔΣM consists of two different block types arranged in arrays that operate in parallel. The first block type (Block A, Fig. 5.7a) is tasked with calculating the four-point output bitstream using the internal states of the modulator $d_1[n], d_2[n], d_3[n]$ (see Fig. 5.4a), an input value $x[n]$, and a speculative result y_s (see Figs. 5.5 and 5.8). Block A consists of N PEs and they now have the structure described in Fig. 5.6. The speculative result y_s is a constant and ranges from 0 to $2^N - 1$. It selects a value from a list of coefficients that are determined at design-time. For instance, PE$_3$ in an A block that speculates on the result 1010_2 will have $y_s = 1010_2$. The multiplexer present in the regular Hatami structure is then replaced by the extra adder shown in Fig. 5.6. From Eq. (4.22) we know that the dependence on $y[n], \ldots y[n+2]$ is $y_v = s_{32}y[n]+s_{33}y[n+1]+s_{34}y[n+2] = -s_{32}+s_{33}-s_{34}$.

The second block type (Block B, Fig. 5.7b) is responsible for updating the internal states of the modulator between each calculation using the previous internal states $d_1[n-1], d_2[n-1], d_3[n-1]$, the previous input value $x[n-1]$, and the speculative result y_s. Block B removes the multiplexers needed in the regular Hatami structure in the same manner that block A removes the multiplexer and uses a precomputed value based on y_s.

The modulator structure has three arrays in total. Two arrays, each consisting of 16 (2^N) type A blocks, and one array of 16 type B blocks. By dividing the functionality into these two blocks the critical path is reduced to that of Block B, by having the arrays of Block A operate during two clock cycles as in Fig. 5.8.

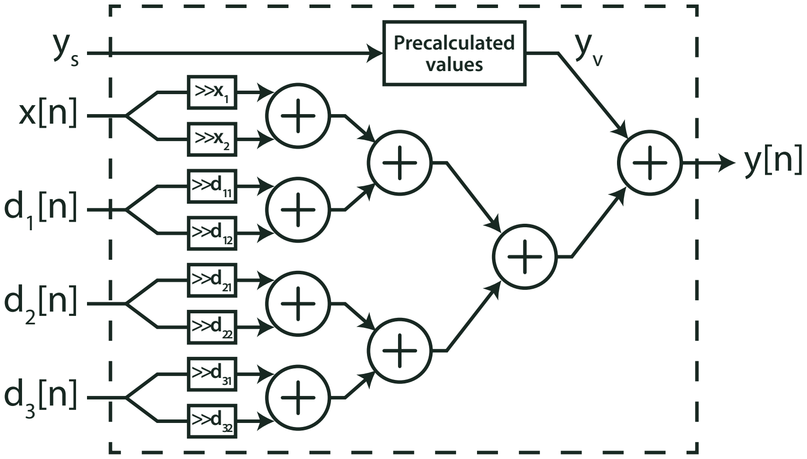


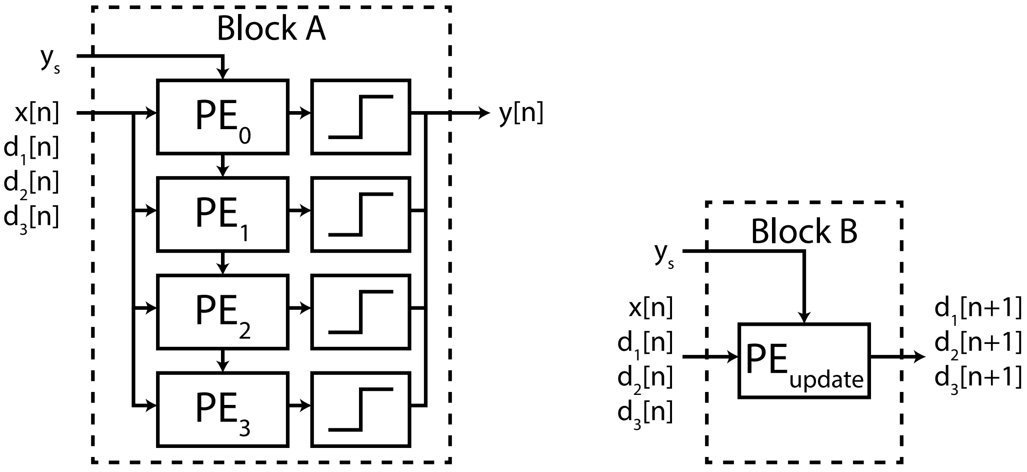
Figure 5.6: General structure of the PE used in the SPΔΣM architecture. The signal y_s is a constant that choose from a list of constants defined at design-time. The arithmetic shift blocks also shift a constant amount and are implemented with wiring.

5.4.2 Speculative operation

The arrays in the speculative pseudo-parallel ΔΣ modulator structure work in a leap-frogs fashion when processing the input signal sequence fed into the SPΔΣM.

First cycle of f_s: Array A_0 initiates processing of $x[n]$ at the same time as array B initiates the process of calculating the updated internal states $d_1[n+4], d_2[n+4]$, and $d_3[n+4]$.

Second cycle of f_s: Array B has finished its processing and feeds the 16 different versions of the updated internal states to the 16 blocks in array A_1 that uses these to initiate processing of the next value in the input signal sequence $x[n+4]$. Array B then initiates a new update of the internal states in order to produce $d_1[n+8], d_2[n+8]$, and $d_3[n+8]$.



(A) Block A

(B) Block B

FIGURE 5.7: Graphical representation of the division of functionality between the two different block types that comprise the SPΔΣM.

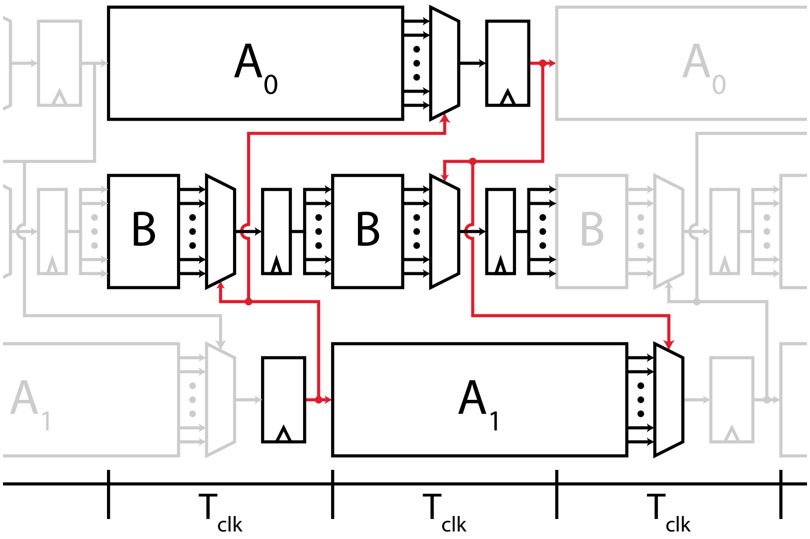


FIGURE 5.8: Graphical representation of the different blocks in the speculative modulator array and their interaction in time. The A_0 and A_1 blocks calculates that output bitstream and the B block updates the internal states. There is only one physical instantiation of each block. The critical path is emphasized.

Third cycle of f_s: Array A_0 has now finished processing of $x[n]$ and the correct result is used to indicate to both array B and A_1 which of the 16 copies they contain will produce the next valid result. After this it initiates the processing of $x[n+8]$, using the updated internal states from array B. Array B reiterates its behavior from the second cycle of f_s but with array A_0 instead.

This behavior is repeated until the input signal sequence is empty.

5.5 Performance results

The designs are synthesized using Xilinx ISE v14.3 for a Xilinx Virtex 6 LX550-T with a speed grade of -2. The delays are evaluated using Xilinx PlanAhead v14.3. The synthesis effort is set at continue on impossible in order to reach an implementation that is as fast as can be achieved by the tool and implementation technology. The clock net constraints are specified as 500 MHz. The designs are place-and-routed twice, once with regards to area and once with regards to speed and the fastest alternative is chosen.

The implementation results are found in Table. 5.1. Part-wise implementation of the SPΔΣM indicates a large potential for the architecture with a clock speed of 278 MHz, which is 46 % faster than the non-speculative modulator topology at 152 MHz. However, for the full design the mapping tool fails to properly place and route the design, resulting in a reduction of processing speed to 144 MHz. A breakdown of the delay paths show that 69 % of the delay is located in the net with relatively long net delays, several in the range of nanoseconds. Modifying the mapping, placement, and routing parameters does not change the relation between logic and routing delay considerably. The utilized FPGA resources remain low at only 1.9 % of the available LUTs.

At its current performance level, the implemented SPΔΣM design achieves the targeted SNDR with 46.2 dB, but only for a single-sided bandwidth $f_B = 29$ MHz and at a carrier frequency of 580 MHz.

Table 5.1
Implementation results

Modulator	LUTs	T_s [ns]	f_s [MHz]	Design Goal
Hatami				
Full design	476	6.6	152	Area
SPΔΣM				
Block A	-	6.0	167	Area
Block B	-	3.6	278	Area
Full design	6746	6.9	145	Speed

Chapter 6

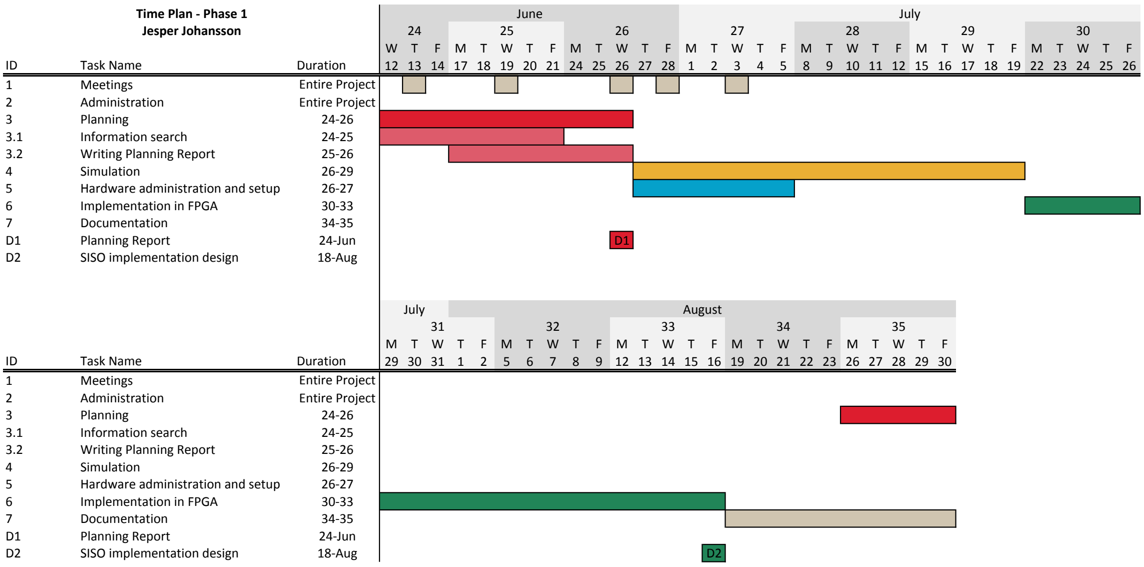
Discussion

This project initially started as a feasibility study of using $\Delta\Sigma$ modulators for generating high-frequency digital signals for use in MIMO systems using a single local oscillator. However, the first phase showed that the speed of the FPGA fabric is not yet fast enough for implementing a $\Delta\Sigma$ modulator with sufficient performance. Therefore, the focus of the second phase shifted from implementing multiple modulators using a single local oscillator towards investigating methods for increasing the implementation performance of the Hatami modulator evaluated in the first phase.

This resulted in the SP$\Delta\Sigma$M architecture which shows that the block-wise critical path in the Hatami modulator can be reduced further by speculating on the previous result. This architecture has, for small unrolling factors N, a larger performance potential than the regular Hatami modulator structure by trading area and complexity. When the unrolling factor N becomes larger the amount of interconnection grows as the number of speculative modulators is 2^N. This will likely negate the performance increase versus the regular Hatami modulator.

6.1 Project evaluation

The first phase proceeded according to the time plan in Fig. 6.1 with a few exceptions. The *Information search* activity was not limited to the defined period and has instead been active for most of phase one. This should instead have been planned from the beginning and accounted for in the time plan. The consequence of not including it from the beginning is that there was less time available for each of the following stages in phase one. The *Simulation* activity was also prolonged one week and subsequently the *Implementation* activity was postponed for one week. This was due to problems with the way Simulink in Matlab handled signed words that removed some of the necessary control needed for a bit-true simulation.



Time Plan - Phase 1
Jesper Johansson

ID	Task Name	Duration
1	Meetings	Entire Project
2	Administration	Entire Project
3	Planning	24-26
3.1	Information search	24-25
3.2	Writing Planning Report	25-26
4	Simulation	26-29
5	Hardware administration and setup	26-27
6	Implementation in FPGA	30-33
7	Documentation	34-35
D1	Planning Report	24-Jun
D2	SISO implementation design	18-Aug

FIGURE 6.1: Time plan for the first phase of the project spanning weeks 24–35 of 2013.

The second phase was planned at the beginning of it and was considerably more ad-hoc in its nature. The majority of the time was spent investigating different methods for increasing the speed of the designed Hatami ΔΣ modulator. When we came up with the idea for SPΔΣM, we decided that the remaining time should be spent working with this architecture. Due to difficulties with the implementation, the *evaluation and verification* stage got less attention than initially planned. The last month of the project was spent preparing for and visiting the Swedish System-On-Chip Conference (SSOCC14), as well as writing this report. The time allotted for the conference paper and presentation was sufficient, as well as the time spent writing this report. The general conclusion of the second phase is that it should have been more strictly planned at its beginning.

6.1.1 Fulfillment of project goals

This section comments on the goals that were initially set up for this project in Section 1.1 and whether they have been fulfilled or not.

Goal no. 1

Deciding upon an efficient and accurate implementation of the $\Delta\Sigma$ modulator in a general FPGA environment, assuming the presence a serial optical-output capable of the required transmission speeds.

This goal is fulfilled with the choice of the Hatami $\Delta\Sigma$ modulator structure described in Section. 4.2.3. The Hatami structure allows for efficient utilization of the FPGA chip PISOs and also allows a low oversampling rate, which is necessary due to the generally low ratio between necessary output sampling frequency and the FPGA clock frequencies.

Goal no. 2:

Design and implement a single $\Delta\Sigma$ modulator capable of providing sufficient SNDR (45 dB*) for the bandwidth and frequency defined by the needs of the MIMO algorithm and with a static phase difference of at most* $45°$.

This goal is not fulfilled in its entirety. A $\Delta\Sigma$ modulator achieving the specifications have been designed but not implemented due to the reasons described earlier. Neither has the static phase difference been evaluated.

Goal no. 3:

Combine several $\Delta\Sigma$ modulators in a single FPGA using the same local oscillator while keeping the phase-alignment.

This goal has not been attempted as goal number two was not met.

Goal no. 4:

Evaluate the system performance and resource economy.

The system performance and resource economy have been evaluated for the regular Hatami structure as well as for the corresponding SP$\Delta\Sigma$M structure.

6.2 Future work

Due to the limited time budget of this project, the design space surrounding SPΔΣM has not been fully investigated. It would be interesting to further evaluate the advantages and disadvantages of the architecture itself for different unrolling factors and specifically investigate the performance gains for an unrolling factor of two, where I speculate that the relative performance gains versus the regular Hatami structure are the largest.

Since this project was started there has been a number of FPGA chips launched into the market which sports a higher fabric speed than the FPGA chip used in this project. The Achronix Speedster 22i HD series [32] is one example and is targeted at high-speed DSP applications. Investigating how the SPΔΣM architecture implements on these newer and faster chip would be interesting and possibly bring the project closer to its performance goal of 500 MHz.

The mapping methods used by the tools are heuristic, and for the SPΔΣM architecture, with its large need for wide-spanning interconnections, have not been able to produce optimal mapping results. Taking manual control of this process could possibly increase the performance of the architecture on the Virtex 6 FPGA chip that we have used. An interesting possibility to explore is to use functional programming to aid the mapping process.

Another direction in which to move this work would be to synthesize the architecture towards an ASIC instead of an FPGA. In an ASIC environment the large net delay should be lower as it is possible to optimize the drive traces for these nets in order to increase their performance.

The regular Hatami structure in itself is also interesting to continue working with. In order to facilitate the design space exploration of SPΔΣM, a tool that automatically unrolls a general ΔΣ modulator would be beneficial, as the amount of work involved in unrolling each loop is quite substantial and very well suited for a computer.

6.3 Conclusions

In this project, we have shown that the block-wise critical path in the Hatami modulator can be reduced further by making the modulator speculate on the previous result. The Virtex 6 implementation that we have used cannot exploit this improvement, but there are other FPGAs on the market that have a higher fabric speed [32].

With Moore's law, FPGAs are becoming more and more viable for use in high-speed DSP applications, and using them for experimental MIMO system setups becomes more and more plausible for each year. The SPΔΣM architecture is one method for trading a resource that current FPGAs have in abundance for one they do not have—area for speed.

Bibliography

[1] D. Gesbert, M. Shafi, D. shan Shiu, P. Smith, and A. Naguib, “From theory to practice: an overview of mimo space-time coded wireless systems,” *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 3, pp. 281–302, 2003.

[2] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communcations.* Cambridge University Press, 2007.

[3] G. Durisi, A. Tarable, and T. Koch, “On the multiplexing gain of mimo microwave backhaul links affected by phase noise,” in *Communications (ICC), 2013 IEEE International Conference on*, June 2013, pp. 3209–3214.

[4] R. Schreier. (2011, Dec) Delta sigma toolbox. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/19-delta-sigma-toolbox

[5] R. Schreier and G. Temes, *Understanding Delta-Sigma Data Converters.* Wiley-IEEE Press, 2005.

[6] H. Inose, Y. Yasuda, and J. Murakami, “A telemetering system by code modulation - $\Delta\Sigma$ modulation,” *Space Electronics and Telemetry, IRE Transactions on*, vol. SET-8, no. 3, pp. 204–209, 1962.

[7] G. R. Ritchie, J. Candy, and W. Ninke, “Interpolative digital-to-analog converters,” *Communications, IEEE Transactions on*, vol. 22, no. 11, pp. 1797–1806, 1974.

[8] S. Norsworthy, R. Schreier, and G. Temes, *Delta-Sigma data converters: Theory, Design, and Simulation.* Wiley-IEEE Press, 1997.

[9] R. Schreier, “An empirical study of high-order single-bit delta-sigma modulators,” *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 40, no. 8, pp. 461–466, 1993.

[10] R. Schreier and M. Snelgrove, "Bandpass sigma-delta modulation," *Electronics Letters*, vol. 25, no. 23, pp. 1560–1561, 1989.

[11] P. H. Gailus, W. J. Turney, and F. R. Yester, "Method and arrangement for a sigma delta converter for bandpass signals," US Patent number 4,857,928, Aug, 1989.

[12] J. Keyzer, J. Hinrichs, A. Metzger, M. Iwamoto, I. Galton, and P. Asbeck, "Digital generation of RF signals for wireless communications with band-pass delta-sigma modulation," in *Microwave Symposium Digest, 2001 IEEE MTT-S International*, vol. 3, 2001, pp. 2127–2130 vol.3.

[13] A. Jayaraman, P. Chen, G. Hanington, L. Larson, and P. Asbeck, "Linear high-efficiency microwave power amplifiers using bandpass delta-sigma modulators," *Microwave and Guided Wave Letters, IEEE*, vol. 8, no. 3, pp. 121–123, 1998.

[14] C. Dick and F. Harris, "FPGA signal processing using sigma-delta modulation," *Signal Processing Magazine, IEEE*, vol. 17, no. 1, pp. 20–35, 2000.

[15] C. H. Leong and G. Roberts, "An effective implementation of high-order bandpass sigma-delta modulators for high speed d/a applications," in *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, vol. 1, 1997, pp. 49–52 vol.1.

[16] S. Hongzhi, W. Jie, M. Yichao, K. Yang, and Z. Jie, "A fpga-based 7-order 1-bit sigma-delta modulator for high-precision signal generation," in *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, 2009, pp. 4–160–4–164.

[17] A. Ng'oma, "Radio-over-fibre technology for broadband wireless communication systems," Eindhoven, 2005.

[18] A. Seeds, "Radio over fibre systems," in *IEEE Lasers and Electro-Optics Society, 2008. LEOS 2008. 21st Annual Meeting of the*, 2008, pp. 848–849.

[19] C.-P. Liu and A. Seeds, "Transmission of mimo radio signals over fibre using a novel phase quadrature double sideband frequency translation technique," in *Microwave photonics, 2008. jointly held with the 2008 asia-pacific microwave photonics conference. mwp/apmp 2008. international topical meeting on*, 2008, pp. 23–26.

[20] A. Seeds, C.-P. Liu, and T. Ismail, "Technologies for radio over fibre systems," in *Summer Topical Meeting, 2009. LEOSST '09. IEEE/LEOS*, 2009, pp. 53–54.

[21] A. Chowdhury, H.-C. Chien, J. Wei, and G.-K. Chang, "Multi-service multi-carrier broadband mimo distributed antenna systems for in-building optical wireless access," in *Optical*

Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC), 2010, pp. 1–3.

[22] A. Eshraghi and T. Fiez, “A comparative analysis of parallel delta-sigma adc architectures,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 3, pp. 450–458, 2004.

[23] D. Scholnik, “A parallel digital architecture for delta-sigma modulation,” in *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*, vol. 1, 2002, pp. I–352–5 vol.1.

[24] S. Hatami, M. Helaoui, F. Ghannouchi, and M. Pedram, “Single-bit pseudoparallel processing low-oversampling delta-sigma modulator suitable for sdr wireless transmitters,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 922–931, April 2014.

[25] E. King, A. Eshraghi, I. Galton, and T. Fiez, “A nyquist-rate delta-sigma a/d converter,” *Solid-State Circuits, IEEE Journal of*, vol. 33, no. 1, pp. 45–52, 1998.

[26] J. Cormier, R.F., T. Sculley, and R. Bamberger, “Combining subband decomposition and sigma delta modulation for wideband a/d conversion,” in *Circuits and Systems, 1994. ISCAS ’94., 1994 IEEE International Symposium on*, vol. 5, May 1994, pp. 357–360 vol.5.

[27] J. Cormier, RonaldF., R. Bamberger, and T. Sculley, “A fourth order bandpass delta-sigma modulator with digitally programmable passband frequency,” *Analog Integrated Circuits and Signal Processing*, vol. 12, no. 3, pp. 217–229, 1997. [Online]. Available: http://dx.doi.org/10.1023/A%3A1008245424801

[28] P. Cappello and K. Steiglitz, “Some complexity issues in digital signal processing,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 5, pp. 1037–1041, Oct 1984.

[29] Y. Takahashi, T. Sekine, and M. Yokoyama, “A comparison of multiplierless multiple constant multiplication using common subexpression elimination method,” in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*, Aug 2008, pp. 298–301.

[30] Xilinx Inc., “Virtex-6 family overview,” January 2012. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf

[31] R. Schreier, “Bandpass delta-sigma data converters,” in *Signals, Systems and Computers, 1995. 1995 Conference Record of the Twenty-Ninth Asilomar Conference on*, vol. 1, 1995, pp. 94–97 vol.1.

[32] Achronix Semiconductor, “Speedster 22i hd fpga platform,” 2014. [Online]. Available: http://www.achronix.com/wp-content/uploads/docs/Speedster22i_Product_Brief_PB024.pdf