LOW-POWER
OVERSAMPLED SIGNAL PROCESSING
for DIGITAL RADIO RECEIVERS

by

HONG-KUI YANG, B.Eng., M.Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Department of Electronics

Carleton University
Ottawa, Ontario
March 22, 1998

0-612-32351-X

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis,

"LOW-POWER OVERSAMPLED SIGNAL PROCESSING for DIGITAL RADIO RECEIVERS"

submitted by
HONG-KUI YANG, B.ENG., M.ENG.

in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Chair, Department of Electronics

Thesis Supervisor

External Examiner

Carleton University
May 5, 1998

Abstract

Three techniques for reducing power consumption in oversampled receivers are developed: a method for improving the SNR attainable in power-efficient double-sampled $\Delta\Sigma$ modulators; a combination of polyphase and multistage techniques to minimize power in decimation filtering; and a method of re-timing decimation to avoid the need for a re-sampling filter in the timing recovery circuits that follow the oversampled ADC.

An analysis of mismatch effects in double-sampled SC $\Delta\Sigma$ modulators shows that the feedback path from the quantizer to the first integrator is dominant, and we show how to use a bilinear integrator circuit to obtain first-order noise shaping of this error. While conventional double-sampled circuits are limited to 12-bit resolutions with typical components, the new circuit can go to 16-bit or better in the presence of the same mismatches.

Cascaded accumulators (typical bit-width of 16~24 bits) in a CIC decimator often dominate power consumption and limit clock rates. Simply using multi-stage CIC decimators is not a solution. A combination of multi-stage CIC decimators with polyphase techniques mitigates these problems. We show how to design multi-stage polyphase CIC decimators by considering aliasing rejection for interference and quantization noise, and by budgeting the word-length in each stage. We provide a design scheme to simplify polyphase components again down to a handful of gates. An FPGA implementation of a 100-MHz digital downconverter using the new design shows a 5x power saving. The technique makes low-power and GHz-rate decimators practical by reducing the peak rate.

In an oversampled receiver, we show that moving timing recovery function into an existing decimator offers the fine resolution required at a much lower cost than interpolation method. This allows us to adjust timing by a typical 1/64 of a symbol period. Simply shifting this clock phase is not a solution, however, because it produces large "glitches" at the output. We show that the glitch settles out after *N* (typically 3 or 4) samples, so that it can be eliminated by using a dual-differentiator decimator. We analyze timing jitter and SNR bound due to interferer mixing with jitter and show a good fit with simulation. One experiment shows that the SNR bound result is within 1.5 dB of consistency with the estimate. We verify the stability and validity of the circuit by implementing an FPGA chip for BPSK.

Acknowledgments

I owe special gratitude to my thesis supervisor Dr. Martin Snelgrove of Carleton University. Thank him for providing me the opportunity to continue my Ph.D. program at Carleton and giving me freedom to do what I am interested in. Thank him for his genuine guidance, remarkable insight into problems and valuable advice.

Part of the materials in Chapter 3 comes from my Ph.D. work conducted in the Technical University of Nova Scotia (TUNS, merged into Dalhousie University in 1997), Halifax, NS. I am very grateful to Dr. E.I. El-Masry of TUNS for all he did for me.

I acknowledge the following organizations for providing me financial support: OCRI/NSERC Chair and Micronet program through Dr. Snelgrove, the Department of Electronics of Carleton University for teaching assistantships and scholarships, NSERC and Micronet program through Dr. El-Masry, TUNS for Rosetti Graduate Scholarships, and the Department of Electrical Engineering of TUNS for teaching assistantships.

I would like to thank many people in Dr. Snelgrove's high-speed IC lab. In particular, Nick Longo, research engineer, helped me build and test the FPGA chips. Phil and Theo, research engineers, prepared the test setup. Ash Swaminathan helped me build a double-sampled ΔΣ modulator. Alana, research administrator, assisted in non-academic aspects.

Also, I would like to thank Drs. P. Edmonson, T. Aboulnasr, R. Hafez, M. Copeland, C. Plett, R. Mason and others to help further improve the quality of this thesis.

Finally and most importantly, thank my parents for their love, care and encouragement. Thank my wife for her loving support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog to Digital Converter
ACC	Accumulator
ASIC	Application Specific Integrated Circuit
BER	Bit Error Rate
BiCMOS	Bipolar and Complementary Metal Oxide Semiconductor
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
CIC	Cascaded Integrator and Comb
CMOS	Complementary Metal Oxide Semiconductor
CSD	Canonical Signed-Digit
CUT	Circuit Under Test
DC	Direct Current
DAC	Digital to Analog Converter
DDC	Digital Down Converter
DDS	Digital Direct Synthesis
DSP	Digital Signal Processing
ΔΣM	Delta-Sigma Modulator
EOSR	Effective Oversampling Ratio.
FIR	Finite Impulse Response
FM	Frequency Modulation
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GHz	Giga Hertz
GPIB	General Purpose Interface Bus
GSM	Global System for Mobile Communications
GMSK	Gaussian Minimum Shift Keying
HBT	Heterojunction Bipolar Transistor
IC	Integrated Circuit

IF	Intermediate Frequency
I	In-phase
IOSR	Intermediate Oversampling Ratio
KHz	Kilo Hertz
LF	Lowpass Filter or Loop Filter
LNA	Low Noise Amplifier
ISDN	Integrated Service Digital Network
LO	Local Oscillator
MHz	Mega Hertz
NCO	Numerically Controlled Oscillator
OSR	Oversampling Ratio
PC	Personal Computer
PCS	Personal Communication Services
PLL	Phase Locked Loop
Q	Quadrature
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
ROM	Read Only Memory
RRC	Root Raised Cosine
SAW	Surface Acoustic Wave
SC	Switched Capacitor
SFDR	Spurious Free Dynamic Range
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
VCO	Voltage Controlled Oscillator
VLSI	Very Large Scale Integration

$\Delta y(n)$	transient error signal
δ_i	mismatch error between path gains k_{i1} and k_{i2}, $i = 1, 2, 3$
$e(n)$	quantization noise or the normalized timing error in timing recovery
$e_s(n)$	timing error signal from a timing error detector
$e_\tau(n)$	timing error normalized to the symbol rate
$e_{\tau max}$, $e_{\tau min}$	maximum and minimum timing error, respectively
e_{rms}	rms quantization error
$E(.)$	mean value
$E(f)$	spectral density of quantization noise
E_1, E_2	discrete-time quantization noise on phases 1 and 2 in the z-domain
E_{di}	quantization noise in the decimation stage, $i = 1, 2, 3, 4$
E_{m1}, E_{m1}	error introduced by mismatch in the first and second integrators in double-sampled $\Delta\Sigma$ modulators
$EOSR$	effective OSR
f	frequency variable
f_c	clock rate ($f_s = f_c$ for single-sampling and $f_s = 2f_c$ for double-sampling)
f_{s0}	clock for the implemented timing-phase-adjustable CIC decimator
f_1	clock for differentiator 0 in a dual-differentiator CIC decimator
f_{1a}	clock for differentiator 1 in a dual-differentiator CIC decimator
f_2	clock at the output of halfband filter in the timing recovery loop
f_3	clock at the output of data filter in the timing recovery loop
f_4	clock for the loop filter in the timing recovery loop
f_0	cutoff frequency
f_{ch}	control signal used in gating the comparator in timing recovery
f_{IF} f_{IF1}, f_{IF2}	IF carrier frequencies
f_P	frequency of the saw-tooth timing error waveform, $f_P = 1/T_P$
f_s	sampling rate or sampling frequency, $f_s = 1/T$
f_{sy}	symbol rate, $f_{sy} = 1/T_{sy}$
$F_i(z)$	the i^{th} polyphase component, $i = 0, 1, 2,...$

ϕ_1, ϕ_2	two-phase non-overlapping clocks
ϕ_{1b}, ϕ_{2b}	inverted versions of ϕ_1, ϕ_2 respectively
ϕ_{1db}, ϕ_{2db}	delayed versions of ϕ_1, ϕ_2 respectively
ϕ_{1d}, ϕ_{2d}	delayed versions of ϕ_{1db}, ϕ_{2db} respectively
g_1	comparator gain in a ΔΣ modulator
g_2	DAC gain in a ΔΣ modulator
h	timing phase adjustment step size
$h_i(n)$	coefficients in the impulse response of a FIR filter
$H(z)$	transfer function in the z-domain
$H_1(z)$	the first transfer function in a cascaded structure
$H_2(z)$	the second transfer function in a cascaded structure
$H_X(z)$	the signal transfer function in a ΔΣ modulator
$H_E(z)$	the noise transfer function in a ΔΣ modulator
i	integer variable
$in+$, $in-$	differential inputs
$IOSR$	intermediate oversampling ratio
k	integer variable
k_i	path gain in a linear double sampling ΔΣ modulator model, $i = 1, 2, 3, 4$
k_{i1}, k_{i2}	a pair of path gains in a double sampling SC circuit, $i = 1, 2, 3, 4$
k_s	bit reduction due to the design scheme for polyphase components
m_τ	mean timing error
M	ΔΣ modulator order
M_i	MOS transistors, $i = 1, 2, ...$
M_P	number of subfilters in implementing the re-timing interpolator
M_L	number of taps in a polyphase component
$MSB(.)$	most significant bit
$MIN(x,y)$	minimum value of x and y
n	integer variable
N_{conven}	rms noise power in the desired band of the conventional double-sampled

	$\Delta\Sigma$ modulator due to capacitor mismatch
N_{novel}	rms noise power in the desired band of the novel double-sampled $\Delta\Sigma$ modulator due to capacitor mismatch
n_1, n_2	word-lengths in the input and output of the CUT
N, N_1	N-stage or N_1-stage CIC decimator (the order of a CIC decimator)
N_h, N_d	numbers of taps in the halfband and data filters respectively
N_s	total spur power in the desired band introduced by interferer mixing
$N(f)$	spectral density of shaped modulation noise
outp, *outn*	differential outputs
out+, *out-*	differential outputs
OSR	oversampling ratio
$P_0(M)$	rms noise power in the desired band in an M^{th} order $\Delta\Sigma$ modulator
P_i	quantization noise power at different decimation stage, $i = 1, 2, 3, 4$
P_s	power consumption in a single-sampled SC circuit
r_i	coefficients in the Fourier transform, i is an integer.
R, R_1, R_2	downconversion or upconversion factor where $R = R_1R_2$
$\overline{R}$	average downsampling factor of R due to timing adjustment
R_I	up-sampling ratio
SNR	signal to noise ratio
σ_τ	rms value of timing jitter
T	sampling period, $T = 1/f_s$
T_P	period of the saw-tooth timing error waveform
TH	threshold for the comparator in timing recovery
τ and τ_e	the actual and estimated delays in timing recovery loop
$u_1(n)$, $u_2(n)$	discrete-time signals on phases 1 and 2 at $f_s/2$
U_1, U_2	discrete-time signals on phases 1 and 2 in the z-domain
$\tilde{U}$	$\tilde{U} = U_1z^{-1/2} - U_2$
V_{dd}	positive power supply
$v_1(n)$, $v_2(n)$	discrete-time signals on phases 1 and 2 at $f_c = f_s/2$

v_{y1}, v_{y2}	analog signals converted from outputs y_1 and y_2.
V_1, V_2	discrete-time signals on phases 1 and 2 in the z-domain
V_{rn}, V_{rp}	negative and positive reference voltages in double sampling $\Delta\Sigma$ modulators
V_r	$V_r = \frac{V_{rp} - V_{rn}}{2}$, the effective reference voltage
ΔV_r	$\Delta V_r = \frac{V_{rp} + V_{rn}}{2}$, the common-mode reference voltage
$\hat{V}$	$\hat{V} = V_1 z^{-1/2} - V_2$
$w(n)$	an intermediate variable used in deriving the timing error detector circuit
$w_i(n)$	the output of i^{th} polyphase component, $i = 0, 1, 2, ...$
ω, ω_i	radian frequency, $i = 0, 1, 2$
$x(n)$	discrete-time input signal at f_s
$x_1(n)$, $x_2(n)$	inputs on phases 1 and 2 at $f_s/2$ in double-sampled $\Delta\Sigma$ modulators
X	input signal at f_s in the z-domain
X_1, X_2	inputs on phases 1 and 2 in the z-domain in double-sampled $\Delta\Sigma$ modulators
$\hat{x}$	variable of $\hat{X}$ in the time-domain
$\hat{X}$	$\hat{X} = X_1 z^{-1/2} - X_2$
$y(n)$	discrete-time output signal
$y_1(n)$, $y_2(n)$	output signals in double-sampled $\Delta\Sigma$ modulators
$y_d(n)$	hard-decision value (1 or 0) based on $y(n)$, in the timing recovery circuit
$y_I(n)$, $y_Q(n)$	outputs of the I and Q channels in timing recovery
$Y(z)$	output in the z-domain
Y_1, Y_2	outputs on phases 1 and 2 in the z-domain
$\hat{y}$	variable of $\hat{Y}$ in the time-domain
$\hat{Y}$	$\hat{Y} = Y_1 z^{-1/2} - Y_2$
z, z_1	variables in the z-transform, $z = e^{j\omega T}$ and $z_1 = z^{R1}$

Chapter 1
Introduction

Personal communication services (PCS) are growing rapidly due to advances in digital wireless communication theory, Very Large Scale Integration (VLSI) technology, and Digital Signal Processing (DSP) techniques [Skla88], [Padg95], [Rapp96]. To accommodate this tremendous growth, engineers specializing in radio systems, DSP and VLSI are teaming up to define new architectures and methods for digital radio receivers.

Modern radio receivers digitize input signals early [Meyr95], [Mito95], often with an oversampled Analog-to-Digital Converter (ADC) [Thur95]. A typical architecture of a digital radio receiver is shown in Figure 1.1. The receiver consists of an analog front-end, a delta-sigma (ΔΣ) modulator ADC, Digital Downconverters (DDCs), symbol timing recovery and other DSP functions. The front-end output may be an Intermediate Frequency (IF) signal or a baseband signal. In an IF-digitization receiver, a bandpass ΔΣ modulator digitizes the IF signal and I/Q signals are separated by two DDCs which consist of decimators and Numerically Controlled Oscillators (NCOs). In a baseband-digitization variation, two lowpass ΔΣ modulators digitize the I and Q signals and DDCs are simply decimators.

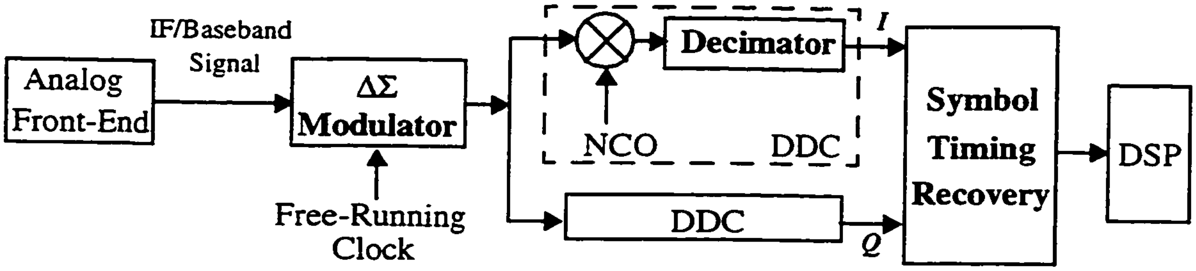


Figure 1.1 Digital architecture for a modern digital radio receiver

A. Motivations

Traditionally, power for cellular telephony has been dominated by the transmitter, but for modern portable applications it is often of key importance to reduce power consumption in receiver: receivers defines standby time, which is of increasing importance as decreasing

with typical components and SNR degradation is negligible compared to the improvement that can be obtained by double-sampling.

3- We verify the proposed architecture by a low power implementation. The HSPICE simulation shows that the SNR is 81 dB with an effective OSR of 100. The estimated power is 1 mW at 50 MHz clock rate compared to a low power second-order $\Delta\Sigma$ modulator consuming 2.5 mW at 4 MHz clock rate [Rabi97].

4- CIC decimators are used in $\Delta\Sigma$ ADCs. The high-speed part of a CIC decimator typically consist of three or four 16~24-bit accumulators which often dominate power consumption and limit clock rates. Simply using a multi-stage CIC decimator is not a solution. We show that these problems can be mitigated by combining multi-stage CIC decimators with polyphase techniques. We derive architectures for the multi-stage polyphase decimators and show that this technique makes decimation possible in micropower and GHz $\Delta\Sigma$ ADCs.

5- We advance the art of design for multi-stage polyphase CIC decimators and develop design methods to simplify circuits. We show that the order of the first full-rate polyphase decimator can be equal to the order of the $\Delta\Sigma$ modulator. This saves $\log_2(R_1)$ bits, where R_1 is the first downsampling ratio. The SNR loss is insignificant in comparison to the actual SNR obtained by a high speed $\Delta\Sigma$ modulator. A design scheme to simplify the realization of polyphase components without using adders for a $\Delta\Sigma$ modulated signal is proposed. It further reduces the word-lengths of the polyphase components and the subsequent CIC decimator by 2~3 bits. We also show how to design a multi-stage polyphase decimator by budgeting the word-length in each stage.

6- We demonstrate the above technique for high speed and low power by implementing a Field Programmable Gate Array (FPGA) DDC chip which consists of a multi-stage polyphase CIC decimator. This chip is able to downconvert an IF signal modulated by a Quadrature Phase Shift Keyed (QPSK) scheme at 100 MHz. We show that it achieves a 5x power saving compared with a conventional multi-stage version. The measured output SNR is 56 dB (80.1 dB in the desired band) and the measured eye diagram shows a negligible Eb/No loss (< 0.1 db), which are in agreement with theoretical results.

7- A technique to move the timing recovery function into the CIC decimation in a $\Delta\Sigma$ modulation based receiver is proposed. The timing resolution is 1/(2*OSR) of a symbol period, where the OSR is typically 64 or higher. Simply changing the re-sampling clock in a CIC decimator, however, creates glitches at the output which settle out after N (typically three or four) samples. The glitches affect the timing error detection. We propose a solution to eliminate the "glitches" by a dual-differentiator CIC decimator. In the scheme, the signal is re-sampled by two different clocks into two differentiators where timing changes are separated by four samples. The output is taken alternately from those two differentiators. The proposed method offers the fine resolution required in modulation schemes such as high-order QAM at a much lower cost than interpolation method.

8- We derive for the mean and variance of the timing jitter introduced by this method. We show that an interferer at the out-of-band channel creates in-band spur noise due to its mixing with timing jitter, which limits the SNR. We present curves that can be used for system design in the specific practical case of an alternate-channel interferer. We show a good fit between theory and simulation. The performance is determined by α (the transmitter and receiver clock rate difference) and OSR: (a) for a small $\alpha < \frac{1}{7OSR}$, the rms timing jitter is dominated by OSR and its slope is -3 dB per octave of OSR. The slope of SNR bound for interferer mixing is -3 dB per octave of α or 3 dB per octave of OSR. (b) for a large $\alpha > \frac{4}{7OSR}$, the rms timing jitter and SNR bound are dominated by the frequency difference α. Their slopes are 3 dB and 6 dB per octave of α respectively.

9- One experiment demonstrates that the method works in the presence of alternate interferer and in conjunction with carrier recovery. The measured SNR bound is within 1.5 dB of consistency with the estimate. The low complexity and stability of the technique is demonstrated by implementing an FPGA chip. The chip is able to do re-timing for a Binary Phase Shift Keying (BPSK) IF signal. It is functional and stable. The measured eye diagram shows that an Eb/No degradation bound of about 0.75 dB is introduced by the timing recovery loop.

C. Thesis Organization

This thesis is organized as follows:

Chapter 1 presents the motivations, contributions and organizations of this thesis.

Chapter 2 provides necessary background on digital radio receiver architectures, in-

imators to wideband and narrowband DDCs are discussed. Detailed circuit designs for an 100 MHz FPGA DDC are also described. In **Appendix C,** detailed circuit designs for the symbol timing recovery FPGA chip are given. A preliminary study on a re-timing scheme based on adjusting delays in the loop is presented in **Appendix D**.

Chapter 2
Background and Overview

Digital techniques play a key role in building high performance, low cost and low power receiver subsystems for cellular and PCS communications [Brod92], [Meyr95], [Bain95], [Wepm95]. Many modern radio receiver architectures utilizing digital techniques have been developed [Coy92], [Ches94], [Abid95], [Mito95], [Thur95] and they can be categorized as baseband and IF digitization receivers. The key components in these architectures are ADCs, decimators or DDCs, re-timing circuits and other DSP functions.

One of the major concerns in handset devices is power consumption. The key to success is developing low power circuits such as $\Delta\Sigma$ ADCs, high-rate decimators and all-digital baseband processing such as symbol timing recovery. Many methods have been developed to minimize power at different levels such as at architecture, circuit and IC technology levels.

This section provides background knowledge and overviews of digital radio architectures, (double-sampled) $\Delta\Sigma$ modulation ADCs, low power decimators (and DDCs), and digital re-timing techniques. The section is organized as follows: an overview of modern digital radio receiver architectures is given in Section 2.1. $\Delta\Sigma$ modulators are reviewed and various double-sampling techniques to mitigate the mismatch effects on $\Delta\Sigma$ modulators are presented in Section 2.2. In Section 2.3, the well-known CIC decimators are reviewed. Issues of power consumption and speed limitations with conventional architectures and their multi-stage variations are discussed. In Section 2.4, we review three all-digital symbol timing recovery techniques. Comparisons among these techniques are given in terms of complexity and DSP computation. Finally, a summary is given in Section 2.5.

2.1 Digital Radio Receiver Architectures

Modern radio receivers have advanced to take advantage of digital techniques. Their architectures are greatly influenced by current VLSI technologies, especially by high speed ADCs, Application Specific Integrated Circuits (ASICs) and DSPs. The key to the choice of digital receiver architectures is determined by high performance and low power ADCs, DDCs, re-timing circuits, etc. In the following, two kinds of digital receiver architectures are reviewed. They are categorized in terms of where to digitize the received signals, that is, baseband and IF digitization.

2.1.1 Receivers with Baseband Digitization

In baseband-digitization receiver architectures, the Radio Frequency (RF) signals are quadrature-downconverted in several stages to I/Q baseband where digitization occurs. They can be further categorized as superheterodyne and direct-conversion receivers.

A. Superheterodyne Receivers

The superheterodyne receiver was introduced by Armstrong in 1918 [Abid95] and has been considered as the radio receiver of choice due to its high selectivity and sensitivity. The idea is to translate the RF signal by several mixing stages down to I/Q baseband signals where the signals are digitized.

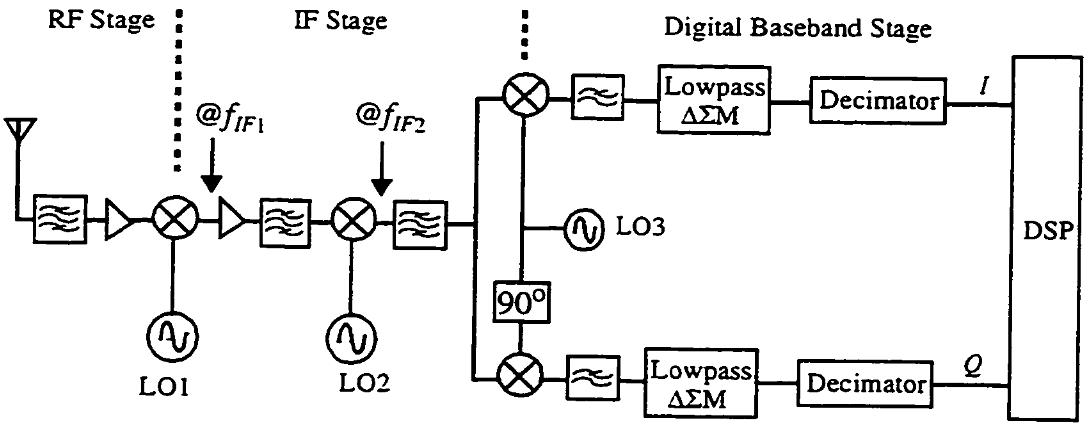


Figure 2.1 A typical superheterodyne receiver with baseband digitization

A typical double-conversion superheterodyne digital radio receiver is depicted in Figure 2.1 [Coy92], [Meyr95]. It is called a baseband-digitization radio receiver. The received RF signal is first filtered and low-noise amplified by a Low Noise Amplifier (LNA). It is then mixed with a variable local oscillator LO1 to an IF signal centered at f_{IF1}. This IF frequency is fixed for the entire reception band to ensures consistent performance across

- DC offset - It results from: (1) I and Q mismatch, (2) LO leakage to the antenna, reflected back to the mixer, (3) a large near-channel interferer and second-order harmonic signals leaking into the LO port of the mixer and self-downconverting to DC.
- $1/f$ noise - This noise in the mixers and amplifiers degrades the SNR.
- high dynamic range for analog front-end (e.g., mixers) - high dynamic range due to strong interferers and weak desired signals.
- high Spurious Free Dynamic Range (SFDR) for ADCs - channel selectivity is preferably achieved by digital filtering and requires a high SFDR for ADCs to handle strong interferers. SFDR is defined as the ratio of the tone signal power to the peak power of the largest spurious signal in the ADC output spectrum.

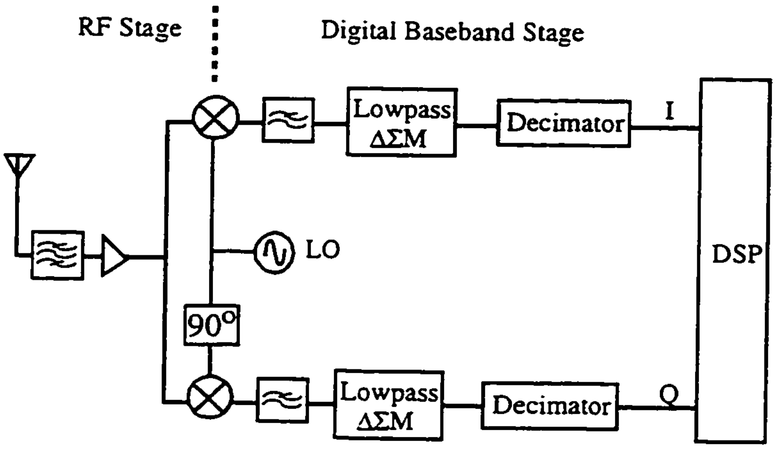


Figure 2.2 A typical direct-conversion receiver

2.1.2 Receivers with IF Digitization

There are some problems with baseband-digitization receiver architectures such as I/Q mismatch, $1/f$ noise, DC offset, etc. To solve these problems, architectures with IF digitization have been proposed, where the incoming analog signal is digitized at an early stage. The key components in the architecture are high speed ADCs and DDCs. The ADC may be a bandpass ΔΣ [Thur95] or a wideband Nyquist ADC [Wepm95]. Digital techniques are used extensively to achieve high performance and high integration. The receivers can be further categorized as narrowband and wideband receivers [Mito95], [Thur95].

A. Narrowband IF-Digitization Receivers

Narrowband IF-digitization receivers shown in Figure 2.3 are designed to demodulate only one frequency channel at one time [Coy92], [Meyr95], [Thur95] and are suitable in a handset design. The bandwidth in a handset device is 200 KHz for GSM in Europe or 30 KHz for IS-54 in North America. Low power consumption is often required (typically below 10 mW).

LO1 is variable and is tuned to the desired band. The first intermediate frequency f_{IF1} is fixed. A fixed LO2 is used to further downconvert the signal centered at f_{IF1} to f_{IF2}. Note that the IF2 filter is often a SAW (surface acoustic wave) filter which does anti-aliasing and help channel selection. The combination of analog filters and digital filters provides the selectivity.

The IF signal at f_{IF2} is digitized by a bandpass ΔΣ ADC. A DDC is used to digitally translate the signal at f_{IF2} into baseband *I* and *Q* signals. The DDC consists of a Numerically Controlled Oscillator, digital mixers and decimators as shown in the figure. Decimators downsample the high data rate to Nyquist rate and suppress the unwanted signals (quantization noise and adjacent interferers). The lower-rate digital *I* and *Q* signals are then processed by DSP for timing recovery and other functions.

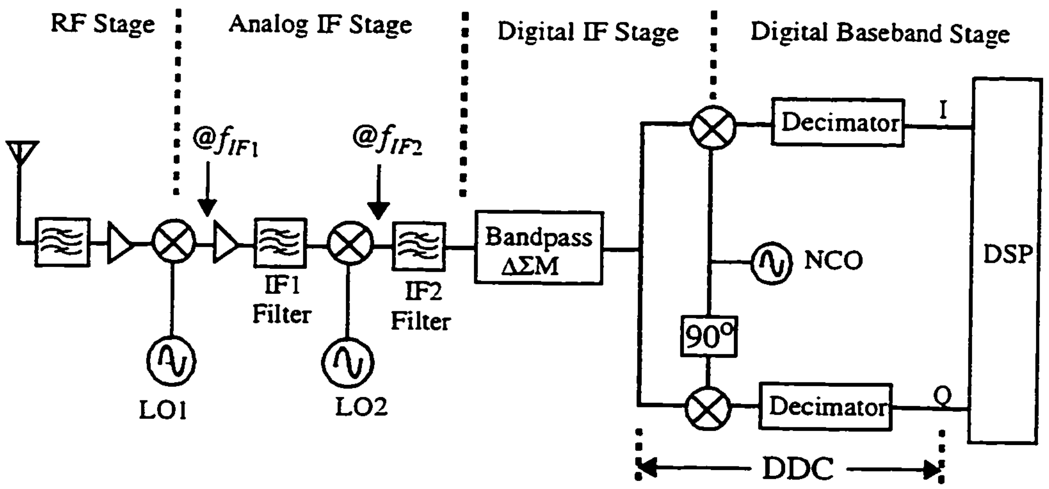


Figure 2.3 A superheterodyne radio receiver with IF digitization

The interferers may be at high levels after the SAW filter but can be removed by subsequent high performance digital filters. According to the Nyquist sampling criterion, the sampling rate must be at least twice the signal bandwidth but not its absolute frequency. Therefore, the sampling rate can be lower than f_{IF2}. The sampling rate f_s is commonly chosen such that the signal at f_{IF2} is aliased by the IF sampling process down to a digital IF frequency of $\frac{1}{4}f_s$ for odd k and $\frac{3}{4}f_s$ for even k. Hence, the sampling rate should be

$$f_s = \frac{4f_{IF2}}{2k-1} \qquad (2.1)$$

where k is an integer [Cons83], [Meyr95]. The outputs of the NCO become simple sequences {1,0,-1,0,...} and {0,1,0,-1,...} respectively. The spectrum is phase-reversed when k is even. The simplest implementation is when $k = 1$ and it relaxes requirements for sampling aperture jitter and IF2 filter. Undersampling occurs when $k \geq 2$ [Gros91].

The receiver dynamic range will be greatly limited by the SFDR of ADCs. A bandpass $\Delta\Sigma$ ADC is a good candidate for this application due to its inherent linearity and high SFDR [Schr89], [Thur95]. In Figure 2.3, the ADC may be replaced by a $\Delta\Sigma$ based frequency discriminator which directly digitizes the phase of the IF2 signal [Bear94]. This approach is suitable for a phase-modulated signal, such as signals using Gaussian Minimum Shift Keying (GMSK) and QPSK modulation.

B. Wideband Receivers

The idea of the wideband receiver is to share the common RF / IF front end, including a wideband ADC, to demodulate multiple frequency channels at the same time in the DSP stage. Thus, this architecture is suitable for basestation receivers. Figure 2.4 shows the architecture where all the analog local oscillators (LO1 and LO2) are fixed, and a DDC is used to tune to the desired channel. The ideal wideband receiver should accommodate simultaneously all, or a large fraction, of the downlink band (824~849 MHz and 1850~1910 MHz for North America cellular and PCS, respectively).

Wideband ADCs and low power DDCs are critical in this architecture. The wideband ADC should have a high enough conversion rate (e.g., 65 MHz for a 20 MHz band) and a sufficient SFDR (e.g., > 80 dB for IS-54 in North America [Wepm95]). DDCs may be

SFDR and SNR (> 60 dB, typically 80 dB). Double-sampling SC implementation of a ΔΣ modulator improves SNR by (6M+3) dB in an M^{th} order modulator almost for free. However the achievable SNR is limited to < 60 dB by capacitor mismatch. Solutions should be worked out to remove the limitation. In the following section, general ΔΣ modulation techniques and the state-of-art of double-sampled ΔΣ modulation techniques are reviewed.

2.2.1 Overview of Discrete-Time ΔΣ Modulation

Quantization of amplitude and sampling in discrete time are at the heart of all digital modulators. If the quantization error is treated as white noise having equal probability in the range $\pm\Delta_q/2$, its rms value is given by [Cand92],

$$e_{rms} = \frac{\Delta_q}{2\sqrt{3}}, \tag{2.2}$$

where Δ_q is the quantization level spacing. The spectral density of the quantization noise is given by [Cand92],

$$E(f) = e_{rms}\sqrt{\frac{2}{f_s}},\ 0 \le f < f_s/2. \tag{2.3}$$

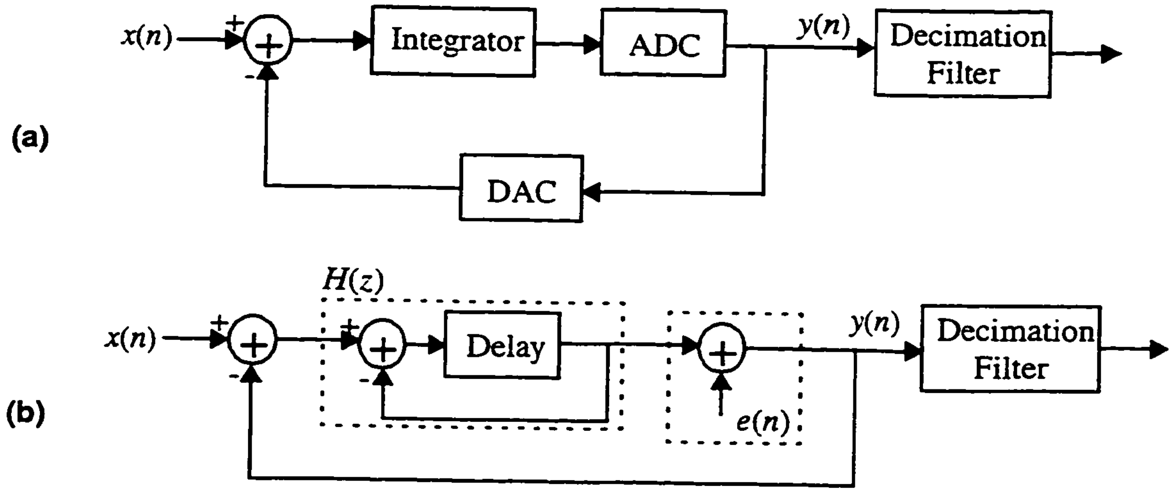


Figure 2.6 (a) A generic delta-sigma ADC and (b) its linear model

A block diagram of a generic $\Delta\Sigma$ modulator is depicted in Figure 2.6 (a) where $x(n)$ is an analog sampled input and $y(n)$ is the digital output of the $\Delta\Sigma$ modulator [Cand92], [Aziz96]. The input to the circuit is fed to the quantizer via an integrator, and the quantized output is fed back and subtracted from the input. This feedback forces the average value of the quantized signal to track the average input. Any difference between them accumulates in the integrator and is eventually corrected by feedback. Note that errors in the Digital-to-Analog Converter (DAC) may directly contribute to the output and degrade SNR. Therefore a one-bit quantizer is often preferred in a $\Delta\Sigma$ modulator. In this implementation, the integrator can be a continuous-time or discrete-time circuit. A SC integrator is widely used in this configuration. The linearized sampled-data model is shown in Figure 2.6 (b), where the integrator has gain $H(z)$ and the white quantization noise is $e(n)$. The output in the z-domain can be expressed as,

$$Y(z) = H_X(z)X(z) + H_E(z)E(z) , \qquad (2.4)$$

where $z = e^{j2\pi(f/f_s)}$ and f_s is the sampling rate. The signal and noise transfer functions are given by $H_X(z) = \frac{H(z)}{1+H(z)}$ and $H_E(z) = \frac{1}{1+H(z)}$ respectively.

In a lowpass $\Delta\Sigma$ modulator, $H_X(z)$ and $H_E(z)$ are a lowpass (or allpass) filter and a highpass filter respectively. The output quantization noise is shaped by $H_E(z)$ in such a way that most of the energy resides outside the desired band. A decimator is used to remove the out-of-band noise, downsample the data rate, and increase the word length.

If the integrator in Figure 2.6(b) has gain $H(z) = z^{-1}/(1-z^{-1})$, then $H_X(z) = z^{-1}$ and $H_E(z) = (1-z^{-1})$. This is a first-order $\Delta\Sigma$ modulator. The spectral density of the shaped modulation noise $N(z) = H_E(z)E(z) = (1-z^{-1})E(z)$ can be expressed as,

$$|N(f)| = \left|E(f)\,(1-e^{-j2\pi(f/f_s)})\right| = 2e_{rms}\sqrt{\frac{2}{f_s}}\sin\left(\frac{\pi f}{f_s}\right) \qquad (2.5)$$

$\Delta\Sigma$ ADC is the ideal candidate for the narrowband IF-digitization radio receiver. A bandpass $\Delta\Sigma$ ADC has also been demonstrated in an RF-digitization radio receiver [Gao97]. $\Delta\Sigma$ ADCs are popular due to the following facts [Cand92], [Aziz96], [Nors97]:

- they are tolerant of analog component variations and hence can achieve higher resolution and higher linearity (or high SFDR) compared to Nyquist ADCs.
- they alleviate the tough requirement for an anti-aliasing filter due to oversampling.
- they permit easy integration with other digital CMOS components since they do not require trimming and thus are suitable for standard CMOS VLSI processes.

2.2.2 Double-Sampled $\Delta\Sigma$ Modulation

In digital radio, a low power lowpass $\Delta\Sigma$ ADC in a baseband-digitization receiver is critical and a low power, high speed bandpass $\Delta\Sigma$ ADC is critical for an IF-digitization receiver, especially for a handset. Power consumption increases linearly with clock rate only up to a critical rate in a CMOS $\Delta\Sigma$ modulator. Above this rate, power consumption increases quadratically with the clock rate [Malo95]. A general practice for low power would be to keep the clock rate as low as possible.

Parallelism allows a high OSR at a low clock and hence is power efficient in the quadratic case. Several techniques using parallelism have been proposed to reduce power consumption or increase the bandwidth of interest [Hurs90], [Aziz93], [Khoi93], [Galt95]. However, all those techniques suffer SNR loss due to mismatch between signal paths. Double-sampling is one of these techniques and will be focused in this work. We review the state-of-art of mismatch mitigation methods in double-sampled modulators.

A. Conventional Double-Sampled $\Delta\Sigma$ Modulation

Double-sampling increases the effective sampling rate by a factor of 2 or reduces the clock rate (and the speed requirement for the op-amps) by a factor of 2. Only capacitors and switches are doubled. Double-sampling imposes no extra requirements for the op-amps. Hence it is an efficient technique for low power applications. The double-sampled SC circuit for filtering applications was proposed in 1980 [Choi80]. A single-sampled SC circuit samples the input signal in one phase and transfers the charge to the output on another. Hence, it wastes op-amp on one of two phases. A double-sampled SC circuit samples the input on both phases, therefore doubling the sampling rate. Note that the double sampling is different from an "N-path"technique ($N = 2$) [Greg86]. In a double-sampled SC circuit, the op-amp is shared on both phases while two op-amps are required in the two-path case and each path has its own op-amp.

In a CMOS analog circuit, the bias current required to get adequate settling increases linearly with clock rate in the weak inversion region and quadratically in the strong inversion region. This is different from CV^2f formula for the digital circuit which describes saturated switching. Hence, there is a critical rate, above which power consumption increases quadratically [Malo95]. For the quadratic case, a power consumption comparison of three techniques mentioned above is listed in Table 2.1 where P_s is the power consumption of a single-sampling SC circuit at sampling rate f_s. As can be seen, double sampling is the most power efficient technique.

Table 2.1 Power consumption comparison in different SC techniques

Sampling rate	Single sampling	Two-path	Double sampling
f_s	P_s	$P_s/2$	$P_s/4$
$2f_s$	$4P_s$	$2P_s$	P_s

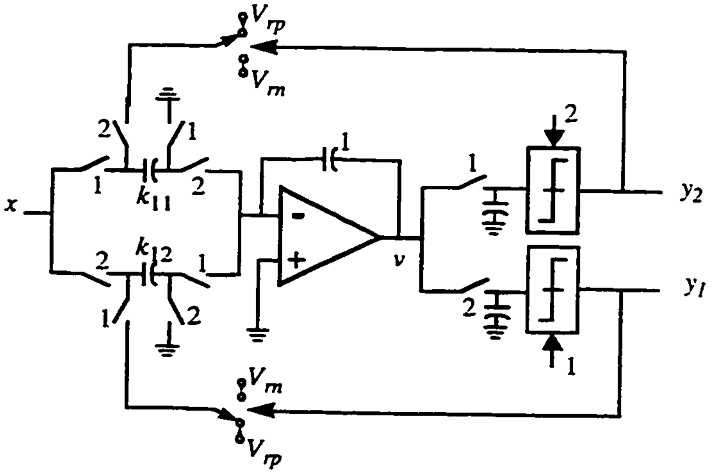


Figure 2.8 A first-order double-sampled SC delta-sigma modulator

with the adding of a pair of capacitors. Three input capacitors are connected to the op-amp during every sample period and hence limit the speed. Additionally, this scheme needs a complex finite state machine (FSM) to generate clocks which are required by the feedback loops. This makes the design more complex and consumes extra power.

An improvement to the above scheme called individual-level averaging switching scheme was proposed in [Than97]. This method is quite similar to the above scheme. The improvements over the above scheme are: (1) three capacitors connected to the op-amp during every sample period are reduced to two; (2) the FSM circuits are simplified. The drawback of using FSM circuits still has not been eliminated.

We address the problem in Chapter 3 of this thesis [Yang94], [Yang96c]. We identify that mismatch in the first feedback integrator dominates. When this integrator is replaced by a double-sampled bilinear integrator, we obtain first-order shaping of mismatch error.

More recently, a double-sampled ΔΣ using differential bilinear integrators has been reported [Send97]. A fully floating double-sampling differential bilinear integrator is constructed by a rearrangement of the sampling capacitors. In the scheme, all integrators are replaced by bilinear integrators. Although the mismatch problem is solved, the SNR is lost by 5 dB in this architecture compared to a negligible degradation in [Yang96c].

In summary, ΔΣ ADCs are used in radio receivers due to their high resolution and high SFDR. The double-sampling SC technique is power efficient (x4 savings compared to single-sampling case). However the achievable SNR is limited by mismatch in signal paths to 12-bit resolution. Several techniques to mitigate the mismatch effect have been reviewed.

Double-sampling lowpass ΔΣ ADCs can be used in digital baseband digitization receivers. Following the ΔΣ modulator is a decimator or a DDC, which is reviewed below together with a proposed solution for low power consumption.

2.3 Decimation and Digital Downconversion

Decimators are key components in modern digital radio receivers, as was described in section 2.1. A ΔΣ modulator is always followed by a decimator. In an IF-digitization receiver, a DDC is used to separate the I and Q signals as well as to decimate the oversam-

It is stated in [Hoge81] that this is of no consequence if the following two conditions are met: (1) The filter is implemented with two's complement arithmetic or other number systems which allows "wrap-around" between the most positive and most negative values, and (2) The word-length is equal to or exceeds the maximum magnitude expected at the output of the CIC decimator.

The CIC decimator is very efficient in these senses: 1) it uses no multipliers, 2) it needs no storage elements for coefficients, 3) it has a regular structure, 4) and it allows a wide range of rate-change.

To design such a decimator, word-length analysis is required. We use two's complement number representation. Assume the word-length of the input data is $(B_{in}+1)$, the maximum word-length at the output of CIC decimator is $(B_{max}+1)$, where [Hoge81]

$$B_{max} = \lceil N log_2 R \rceil + B_{in}, \tag{2.7}$$

where $\lceil x \rceil$ is the smallest integer not less than x. Note that the word-length increment in B_{max} relative to B_{in} is almost proportional to the number of stages N and $log_2(R)$. For a large value of rate-change R, this increment may be large. Not only is $(B_{max}+1)$ the word length at the filter output, but it is also the maximum word length for all stages of the filter. In designing a CIC decimator, a straightforward way is to use $(B_{max}+1)$ bits internally in integrators and differentiators. However, truncation or rounding may be used at each integrator or differentiator stage to reduce the register length [Hoge81].

There are two considerations when we specify the aliasing attenuation requirement in a radio receiver: quantization noise aliasing and the interference aliasing. The aliasing attenuation is determined by the number of stages N and downconversion factor R. The choices of N and R are made suitably to provide acceptable passband characteristics.

As noted in Figure 2.10, in the vicinity of kf_s/R, $k = 1, 2, ...,$ the quantization noise within a bandwidth of $2f_0$ will be folded into the baseband, where f_0 is the desired bandwidth. Candy has shown in [Cand86] that N should be at least 1 plus the order of the $\Delta\Sigma$

modulator in order to prevent excessive aliasing of quantization noise power from entering baseband.

Interferers located at $kf_s/R \pm f_0$ are folded into the desired band after decimation. The least aliasing attenuation occurs at $f_s/R - f_0$. The aliasing attenuation is determined by $OSR = (f_s/R)/(2f_0)$. The interferers can be 60 dB higher than the desired signal in a typical radio system. The attenuation of anti-aliasing filter plus the attenuation provided by the decimator should meet the requirement.

In-band droop is another consideration. The largest droop occurs at f_0 and is determined by the number of stages N and downconversion factor R. The droop is compensated by a low-rate (typically Nyquist rate) amplitude correction filter.

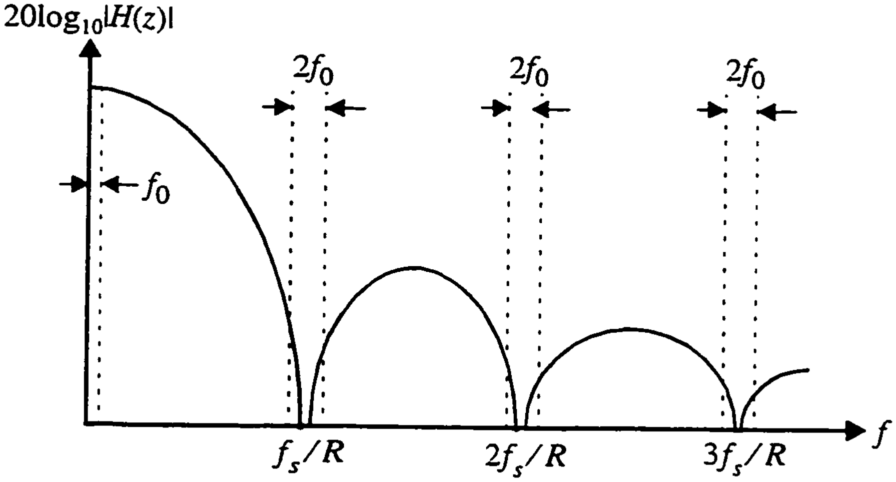


Figure 2.10 Frequency response of a CIC decimator

2.3.2 Decimation and DDC

A DDC consists of a digital mixer and decimation as shown in Figure 2.3. It downconverts a digitized IF signal coming from a bandpass ΔΣ modulator [Schr90] or other ADCs. Therefore, the key component is still the decimator.

Usually, multi-stage decimators are employed to complete the decimation [Croc83] and the first one is often the CIC decimator due to its full sampling rate. The output rate of the CIC decimator is often chosen to be four times the Nyquist rate [Cand86]. Thus, another decimation by 4 is needed to bring down the data rate to the Nyquist rate, which is often accomplished by two halfband decimators [Bran94], [Nors97].

A. Power Consumption and GHz-Rate Operation

CIC decimators are simple circuits for decimating $\Delta\Sigma$ modulated signals. However, they require clocking 3~4 accumulators (typical bit-width of 16~24 bits) at full speed and therefore dominate power consumption and limit clock rates. This becomes more critical for low power consumption or GHz-rate decimation.

To give an idea of the power consumption by the full-rate cascaded integrators, we take an example from a decimation chip reported in [Bran94]. This chip is optimized for low-power consumption and is for a second-order $\Delta\Sigma$ modulator. In the implementation, a conventional three-stage CIC decimator was used, followed by two polyphase halfband filters (18 and 110 taps respectively) and one droop correction filter (8 taps). The internal word-length in the input cascaded integrators is 20-bit. The full-rate integrators consume 16 % of the total power when the input oversampling rate is 11.3 MHz and OSR = 256 [Bran94]. The downsampling ratio is programmable. If the OSR increases to 512, the integrators may consume as much as 38 % of the total power[1]. In the implementation, the polyphase technique which may be exploited to reduce the power consumption was only used in low-rate halfband filters. Considerable power was still consumed by CIC integrators which operated at a higher rate. If simple IIR filters are used to replace long-tap halfband filters, the power consumption taken by integrators out of the total chip power will be higher than reported.

It is very difficult to design 20-bit cascaded integrators at GHz rates with reasonable power consumption. A second-order bandpass $\Delta\Sigma$ modulator at 3.8 GHz with its center frequency at 950 MHz has been developed in a 0.5 μm bipolar process [Gao97]. It achieves an SNR of 56 dB over a 200 KHz bandwidth. A 4 GHz-rate bandpass $\Delta\Sigma$ modulator has been reported [Ragh97] and its noise notch is programmable from 0~70 MHz. The SNR achieved is 92 dB within a 200 KHz bandwidth. A 3.2 GHz second-order lowpass $\Delta\Sigma$

[1] Assume the power consumption of the integrators increases proportionally to the clock rate change.

modulator implemented in InP Heterojunction Bipolar Transistor (HBT) technology was reported in [Jens95]. The SNR is 55 dB over signal bandwidth of 50 MHz. To decimate the second-order bandpass $\Delta\Sigma$ modulator, one needs a two-stage CIC decimator. The internal word-length growth for the 3.8 GHz cascaded integrators is 24-bit when the data is downsampled to 800 KHz. For the second-order lowpass $\Delta\Sigma$ modulator, we need a three-stage CIC decimator. Three cascaded 10-bit accumulators at 3.2 GHz are needed. In either case, one can see the difficulty in implementing such high-rate accumulators without consuming considerable power.

B. Multi-Stage CIC Decimation: Any Advantage?

As a general rule, multi-stage FIR decimators save dramatically in terms of the number of taps in each stage compared to single-stage counterparts [Croc83]. Take an example from [Croc83] where we need to decimate a signal with a sampling rate of 10 KHz by a ratio of 100 to a rate of 100 Hz. A single-stage FIR decimator needs 5080 taps. In a two stage design, the first decimator downsamples the signal by 50 and the second by 2. The required taps are 263 and 110 for the first and second decimators respectively. The savings are 13x and 8x in terms of taps and computation (multiplications).

Multi-rate design using CIC decimators has been discussed in [Chu84] where multiple cascaded CIC decimators are used instead of a single CIC decimator. As a result, the word-length re-growth in each CIC decimator is reduced since the downsampling ratio in each decimator is decreased (see equation (2.7)). However, the power / hardware saving results for FIR decimators do not apply to the multi-stage CIC decimator, as will be demonstrated below.

Assume we have a second-order $\Delta\Sigma$ modulator with OSR = 256 and the downsampling ratio is 256/4 = 64. Using a single CIC decimator, the minimum internal word-length is $2+3\log_2(64) = 20$ bits [Cand86]. For comparison, we use two cascaded CIC decimators where each downsamples by a factor of 8. The minimum internal word-length in the first CIC decimator is $2+3\log_2(8) = 11$ bits and the second is $11+3\log_2(8) = 20$ bits. The power saving is quite small (1~5 %) in comparison to the total power of the whole decimation chip[1]. However, the two-stage CIC decimation design is more complicated in terms of hardware since we need six extra 11-bit adders. Compared to the multi-stage FIR decimation design, the gain with the multi-stage CIC decimation design is insignificant.

To the author's knowledge, there is no publication discussing multi-stage CIC decimation for $\Delta\Sigma$ modulators. The above explanation possibly is the reason. Also multi-stage CIC decimation does not help much in GHz-rate decimation since it is still difficult to implement 11-bit accumulators at 3.2 GHz.

C. Time misalignment

One problem associated with the narrowband DDC is time misalignment which is demonstrated in Figure 2.11. To simplify the implementation, decimation by two is performed in each of the I and Q channels by eliminating the zero-valued multipliers. This decimation causes the sampling images to coincide with the double-frequency terms produced in the mixing process [Saul90], [Thie90]. It therefore eliminates the need for lowpass filtering in this simple decimation. The simplified DDC is shown in Figure 2.11(b). However, this simplification causes a time misalignment between I and Q channels as shown in Figure 2.11(c), where the sample I_i is paired with Q_i (i = 1, 2,...). The I and Q samples ideally should be taken at the same instant.

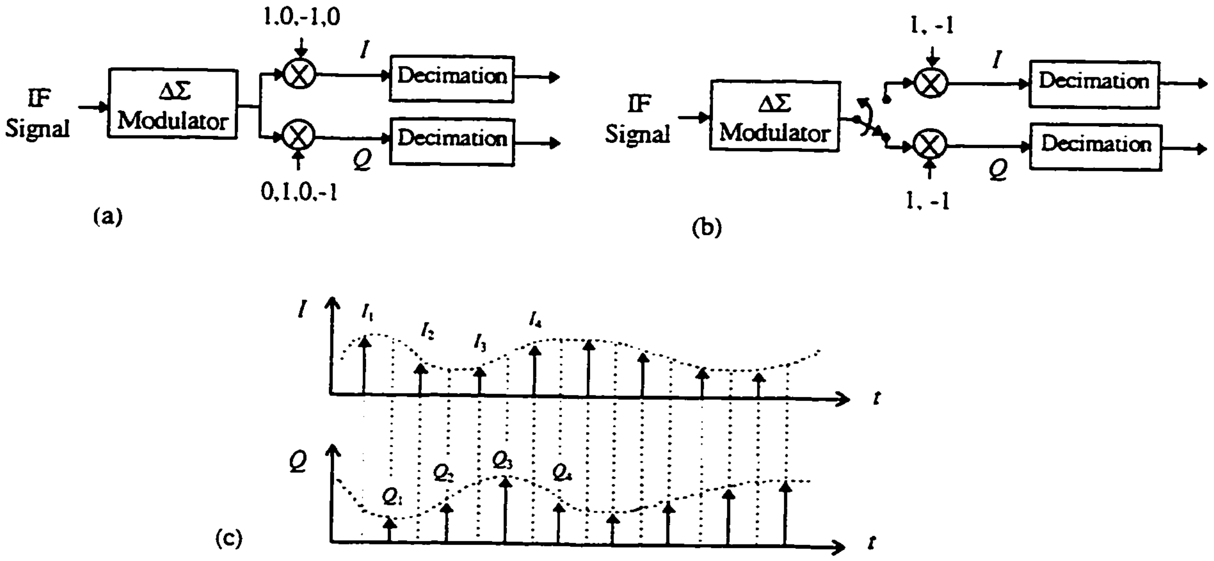


Figure 2.11 Time misalignment in a narrowband DDC

[1] Assume the complexity of accumulator is proportional to the bit width.

integrators in CIC decimators often dominate power consumption and limit clock rates. Unlike a multi-stage FIR decimator, a direct application of multi-stage design to a CIC decimator is not a solution. A proposed solution is to combine multi-stage design with polyphase techniques.

The multi-stage polyphase CIC decimators can be used with double-sampled $\Delta\Sigma$ modulators to achieve low power in baseband digitization receivers. They also can be used in IF digitization receivers to achieve low power or high speed (e.g., for GHz $\Delta\Sigma$ modulators).

Following the $\Delta\Sigma$ ADC is a timing recovery circuit. In the following section, we review all-digital methods for re-timing and show how to simplify the re-timing circuit in an oversampled receiver design which requires a high timing resolution by moving the re-timing function into a modified CIC decimator.

2.4 All-Digital Approaches to Symbol Timing Recovery

Symbol timing recovery is critical for reliable data detection in modern digital communications [Lee90], [Skla88]. The purpose of symbol timing recovery is to synchronize the timing samples to the symbols of the received data signal. There are several ways to recover the symbol timing and in general, they can be categorized as [Lee90], [Gard93]:

- pure analog recovery, shown in Figure 2.12 (a).
- mixed (analog-digital) recovery, shown in Figure 2.12 (b).
- all-digital recovery, shown in Figure 2.12 (c).

The first two methods require a Voltage Controlled Oscillator (VCO) to create a synchronized timing clock to drive an ADC. Those two methods require the subsequent circuits to run at this recovered clock. This creates difficulties in DSP and controller timing.

To take advantage of low cost and high quality digital techniques (hardware and software), it is desirable that the whole timing recovery circuit be implemented digitally [Asch89], [Pokl92], [Meyr95]. In some circumstances, only digital recovery is possible since sampling cannot be synchronized to the received signal. Examples include: (1) multi-channel signals in a wideband receiver, (2) multi-user signals in a CDMA (Code Division Multiple Access) receiver.

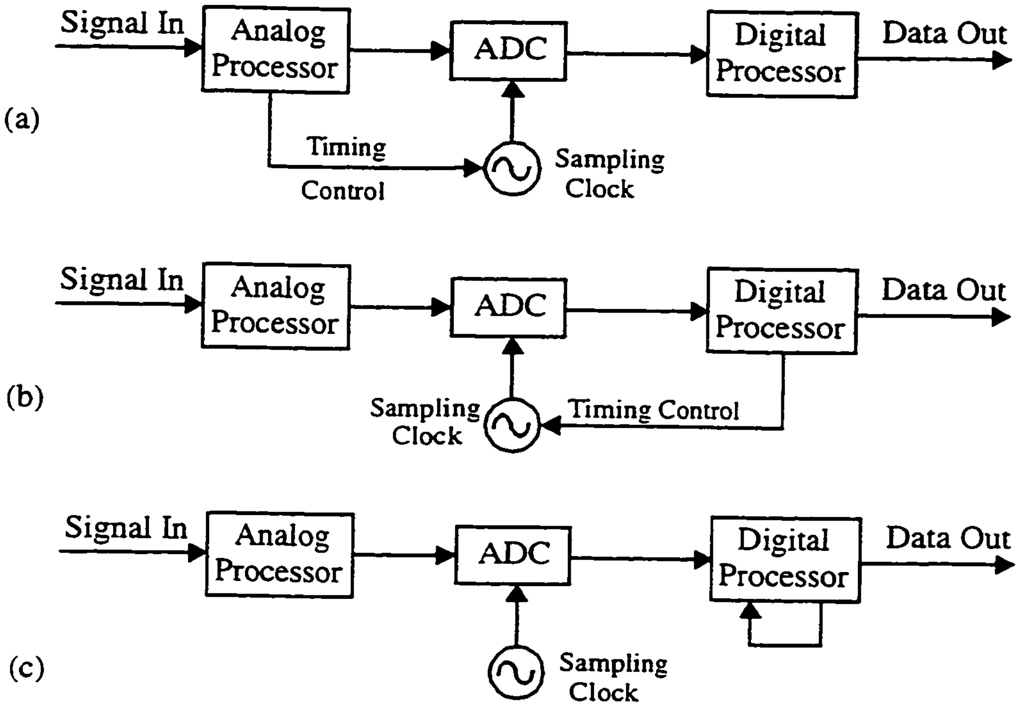


Figure 2.12 Three categories for timing recovery: (a) analog method, (b) mixed method, and (c) all-digital method

The all-digital method uses a free-running clock and the timing recovery activity takes place in a digital processor. The key in this method is a timing adjustment device (or a re-timing circuit) which usually implements a fractional delay [Croc83], [Laak96]. There are three techniques for all-digital implementation [Laak96], [Soll90], [Yang96b], including the decimation method in this thesis. They are described in the following sections.

A. Directly Processing Oversampled Signals

A brute-force method to implement re-timing is to directly process oversampled signals. In some modulation techniques such as BPSK/QPSK, the time resolution of 8x oversampling is adequate. In the demodulation stage, one can directly process an 8x oversampled signal digitally and then choose one of 8 available samples per symbol to adjust the timing [Soll90]. This method is quite simple to implement, avoiding any effort to obtain the re-timing samples. The disadvantage is that one has to deal with the high rate data stream and a high speed DSP must be used to process the incoming 8x oversampled signal. A dedicated high speed ASIC may be required but at the expense of consuming more power.

It is costly to directly process > 8x oversampled signals and hence this method is not suitable for modulation schemes which require a high timing resolution (e.g., better than 1/32 symbol interval) like 64-level or higher Quadrature Amplitude Modulation (QAM).

B. Interpolating Sampled Signals

The interpolation method for symbol timing recovery [Haou87], [Gard93], [Laak96] shown in Figure 2.13 is traditionally used for all-digital implementation where the sampling rate is less than 8x symbol rate. In the figure, a free-running clock samples the ADC and the timing is fine-tuned by a digital processor via controlling the coefficients in the FIR interpolator.

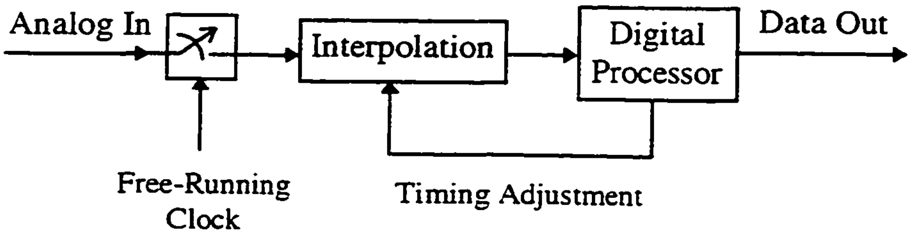


Figure 2.13 Interpolation method for timing recovery

Timing adjustment is done by digitally re-sampling the input digital signal. The key in this technique is a re-timing device called an interpolator which approximately realizes a fractional delay. The performance (e.g., image rejection, jitter) of this technique is determined by the filter types and the number of taps. As well as the conventional FIR lowpass filter design methods [Croc83], many other design methods for such an interpolator have been proposed [Laak96]. To simplify the implementation, an efficient alternate to a separate interpolator is to combine it with a matched data filter [Gott94].

To implement a timing resolution T_{sy}/M_p, we need $M_p/2$ or $M_p/4$ subfilters for a sampling rate $2f_{sy}$ or $4f_{sy}$ respectively, where f_{sy} is the symbol interval. By selecting the signal from one of these subfilters, we can implement a fractional-delay adjustment. The number of taps in each of these subfilters depends on the requirements (e.g., image rejection). For

This method is very efficient since we use a modified existing decimator to do re-timing at a little extra cost. The hardware savings are small for BPSK/QPSK since interpolators are simple. For a high-order QAM system, however, our method offers the fine resolution needed for good performance at a much lower cost than interpolation. Additionally, the complexity of this technique is independent of the timing requirements and modulation schemes unlike the interpolation method.

There is another method which also takes advantage of oversampling [Abou94]. The timing phase is adjusted by directly changing the oversampling clock. The clock is used to sample a ΔΣ modulator. This method falls into the mixed type in the category described in Figure 2.12. The decimation following the ΔΣ modulator is fixed.

In summary, all-digital symbol timing recovery methods are favored in modern digital radio receiver design. We have reviewed three methods and shown pros and cons of each method. For a high-order QAM system, however, the proposed decimation method offers the fine resolution required at a much lower cost than interpolation method.

The proposed decimation re-timing method can be used in any receiver type discussed for symbol timing recovery.

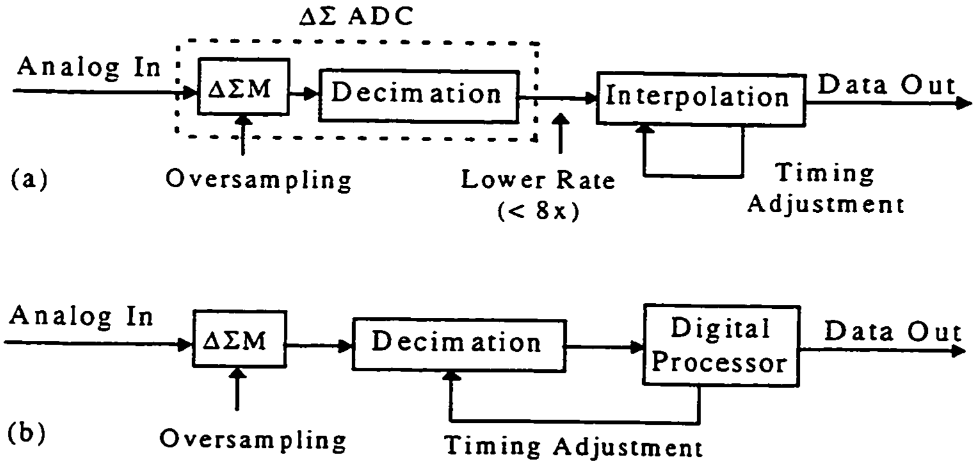


Figure 2.14 All-digital timing recovery for delta-sigma modulated oversampled signals (> 64x): (a) interpolation and (b) decimation methods

Chapter 3
Double-Sampled Delta-Sigma Modulators

One should distinguish between a clock rate f_c and an effective sampling rate f_s. In a single-sampling case, both are the same ($f_s = f_c$) while in a double-sampled modulator, the effective sampling rate is twice the clock rate ($f_s = 2f_c$). The effective OSR (EOSR) is defined as the ratio of the effective sampling rate to twice the signal bandwidth.

The double-sampling technique is an efficient way to increase the EOSR by 2 in a $\Delta\Sigma$ modulator without imposing extra requirements on the clock rate, op-amp settling time and DC gain [Burm96], [Choi80]. For M^{th}-order lowpass or $2M^{th}$-order bandpass double-sampled $\Delta\Sigma$ modulators, approximately ($6M$+3) dB of SNR improvement can be gained.

Double-sampling is also an efficient way for low power applications [Send97]. In a CMOS $\Delta\Sigma$ modulator, power consumption increases linearly with clock rate only up to a critical rate, above which it increases quadratically [Malo95]. By using double-sampling, we are allowed to reduce the clock rate by a factor of 2 for a power improvement between 2 and 4 yet keep the performance.

The achievable SNR, however, is severely limited by the capacitor mismatch of SC circuits [Hurs90] in a lowpass double-sampled $\Delta\Sigma$ modulator. As will be shown, to mitigate the mismatch errors, several methods have been proposed [Ribn91], [Hurs92], [Yang94], [Than97], [Send97]. Their advantages and disadvantages are discussed in section 2.2. To better understand the mechanism of the effect of mismatch, quantitative analysis is required and has proven difficult [Hurs90] because of the complicated feedback structure.

We address these problems in [Yang94], [Yang96c] and this chapter. In Section 3.1, quantitative analyses of the effects of gain mismatches and non-uniform sampling are given. We identify that mismatch in the first feedback integrator dominates. In Section 3.2, a novel double-sampled SC bilinear integrator is proposed and analyzed. In Section 3.3, novel lowpass double-sampled SC $\Delta\Sigma$ modulators which use the bilinear integrator in the

[Hurs90]. It degrades the achievable SNR dramatically. A qualitative analysis of the effect of gain mismatch has been addressed in [Hurs90]. To better understand the mechanism of the mismatch effect, a quantitative analysis is necessary and was proven difficult [Hurs90]. In this section, quantitative analyses of mismatch and non-uniform sampling are given.

The analysis presented here is more difficult than that for a double-sampled SC filter [Riji91] because quantization noise and input signals must be treated separately in a double-sampled $\Delta\Sigma$ modulator.

3.1.1 Lowpass Delta-Sigma Modulators

Nonideal lowpass $\Delta\Sigma$ modulators are first analyzed in this subsection and the results will be extended to a bandpass case in the next subsection.

A. z-transform for two interleaved signals

Assume there are two discrete-time signals $x_1(i)$ and $x_2(i)$, $i = 0, 1, 2, \ldots$ sampled at a clock rate f_s. A third signal is formed by interleaving x_1 and x_2 in such a way that

$$x(n) = \begin{cases} x_1\left(\frac{n-1}{2}\right), n \text{ is odd} \\ x_2\left(\frac{n}{2}\right), n \text{ is even} \end{cases} \tag{3.1}$$

The effective sampling rate for signal x is twice the clock rate f_c. One can derive the relation in the z-domain by using polyphase decomposition [Vaid93],

$$X(z^{1/2}) = X_1(z)z^{-1/2} + X_2(z), \tag{3.2}$$

where $X(z^{1/2})$, $X_1(z)$ and $X_2(z)$ are the z-transforms of signals x, x_1 and x_2 with respect to the clock rate f_c; respectively. $z = e^{j\omega T_c}$, where $T_c = 1/f_c$ and ω is the signal frequency in radians. Note that notation $z^{1/2} = e^{j\omega T/2}$ denotes the z-transform with respect to the effective sampling rate $f_s = 2f_c$. Here the lower case and upper case are used to represent variables in the time and the z-domain, respectively. Thus in the z-domain, the interleaved signal $X(z^{1/2})$, which operates at twice the clock rate, can be obtained from the individual signals $X_1(z)$ and $X_2(z)$ as described in (3.2).

B. Gain Mismatch

A conventional first-order double-sampled SC $\Delta\Sigma$ modulator was described in Section 2.2 and is depicted again in Figure 3.1 for convenience. In the figure, y_1 and y_2 are outputs sampled on ϕ_1 and ϕ_2 respectively. The positive and negative reference voltages are V_{rp} and V_{rn}. In a practical implementation, capacitor mismatch between the two paths k_{11} / k_{12} in Figure 3.1 is responsible for gain mismatch.

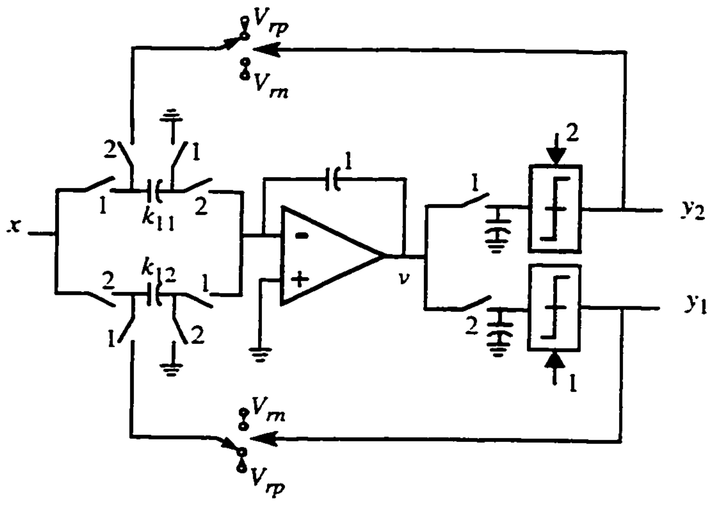


Figure 3.1 A first-order double-sampled SC delta-sigma modulator

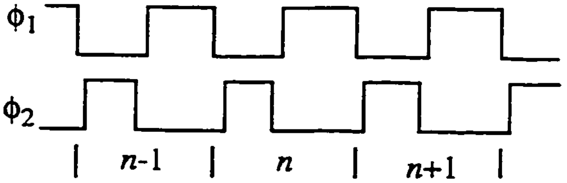


Figure 3.2 A non-overlapping sampling clock scheme

In the following analyses, it is assumed that the only nonideality in the modulator is gain mismatch. In Figure 3.1, the mismatched gains can be expressed as,

$$k_{11} = \left(1 + \frac{1}{2}\delta_1\right),$$

$$k_{12} = \left(1 - \frac{1}{2}\delta_1\right),$$

where δ_1 represents the capacitor mismatch error. It is also assumed that two non-overlapping clocks are used, which is shown in Figure 3.2. In double sampling, there are two samples in a clock cycle. Assume that phase ϕ_2 leads phase ϕ_1 at clock instant n. This means that the clock ϕ_2 samples the signal before the clock ϕ_1 does. As a result, the samples obtained are $x_2(0), x_1(0), x_2(1), x_1(1),..., x_2(n), x_1(n)$, etc. For the SC double-sampled modulator in Figure 3.1, the signals driving the comparators on phases ϕ_1 and ϕ_2, in the z-domain, can be written as,

$$V_1 = \frac{k_{11}X_1z^{-1} + k_{12}X_2}{1-z^{-1}} - \frac{k_{12}Y_1 + k_{11}Y_2}{1-z^{-1}}, \tag{3.3}$$

$$V_2 = \frac{k_{11}X_1z^{-1} + k_{12}X_2z^{-1}}{1-z^{-1}} - \frac{k_{12}Y_1z^{-1} + k_{11}Y_2}{1-z^{-1}}, \tag{3.4}$$

where subscripts 1 and 2 refer to the voltages being sampled on phases ϕ_1 and ϕ_2, respectively. X and Y are the input and output signals, respectively. Note that the independent variable, z, was dropped from all the functions for convenience. Since ϕ_2 leads ϕ_1 at a given instant n, we have

$$Y_1 = V_2 + E_1, \tag{3.5}$$

$$Y_2 = V_1z^{-1} + E_2, \tag{3.6}$$

where E_1 and E_2 are the quantization noises created by the comparator on phases ϕ_1 and ϕ_2 respectively. The noises are assumed to be white [Hurs92]. By interleaving signals Y_1 and Y_2, a new signal Y can be easily obtained by substituting (3.5) and (3.6) into (3.2),

$$Y = Xz^{-1} + (1 - z^{-1/2})E + \frac{1}{2}(\hat{X}z^{-1/2} + \hat{Y})\delta_1, \tag{3.7}$$

where

$$X = X_1 z^{-1/2} + X_2,$$

$$Y = Y_1 z^{-1/2} + Y_2,$$

$$E = E_1 z^{-1/2} + E_2,$$

$$\hat{X} = X_1 z^{-1/2} - X_2,$$

$$\hat{Y} = Y_1 z^{-1/2} - Y_2.$$

Note that signals Y_1 and Y_2 operate at clock rate f_c and the interleaved signal Y operates at the effective sampling rate $f_s = 2f_c$.

In (3.7), the first term is the desired signal. The second term is the error introduced by the quantization noise and is first-order shaped. The last term is an extra error E_{m1} introduced by mismatch which is,

$$E_{m1} = \frac{1}{2}(\hat{X}z^{-1/2} + \hat{Y})\delta_1 \ . \tag{3.8}$$

The term is not noise-shaped. Similar analysis applies to the conventional second-order double-sampled $\Delta\Sigma$ modulator depicted in Figure 3.3. The mismatched gains can be expressed as,

$$k_{11} = \frac{1}{2}\left(1 + \frac{1}{2}\delta_1\right),$$

$$k_{12} = \frac{1}{2}\left(1 - \frac{1}{2}\delta_1\right),$$

$$k_{21} = \frac{1}{2}\left(1 + \frac{1}{2}\delta_2\right),$$

$$k_{22} = \frac{1}{2}\left(1 - \frac{1}{2}\delta_2\right),$$

where δ_1 and δ_2 represent the capacitor mismatch errors. The factor 1/2 is a conventional gain scaling to reduce signal dynamic ranges for op-amps [Bose88]. The output in the desired band can be obtained as (see Appendix A.1),

$$\begin{aligned} Y \equiv & X z^{-3/2} + 4\left(1 - z^{-1/2}\right)^2 E + \frac{1}{2}\left(\hat{X} z^{-1/2} + \hat{Y}\right) z^{-1} \delta_1 \\ & + \left(\hat{U} z^{-1/2} + \hat{Y}\right)\left(1 - z^{-1/2}\right)\delta_2 \end{aligned} \tag{3.9}$$

where $\hat{U} = U_1 z^{-1/2} - U_2$. U_1 and U_2 are the second integrator inputs on phase ϕ_1 and ϕ_2.

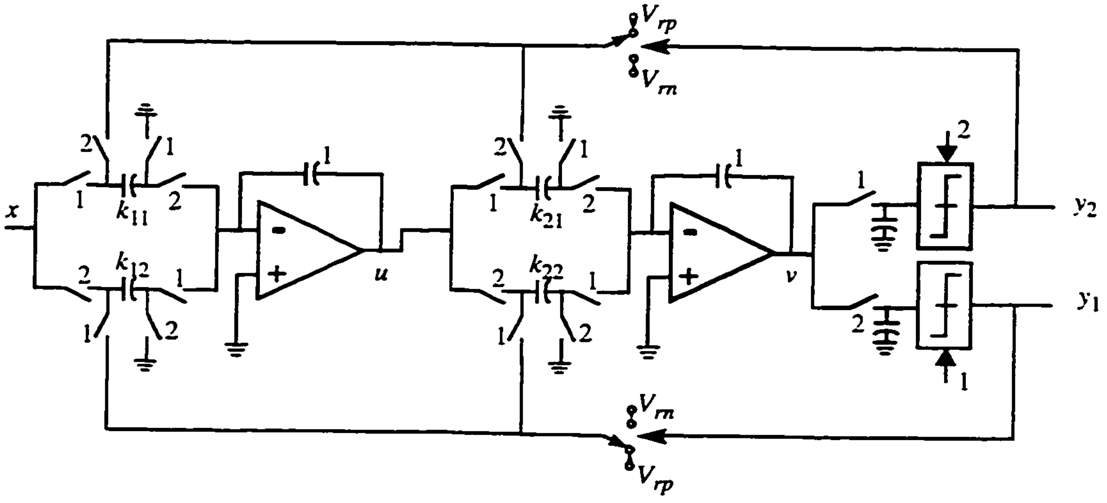


Figure 3.3 A second-order double-sampled SC delta-sigma modulator

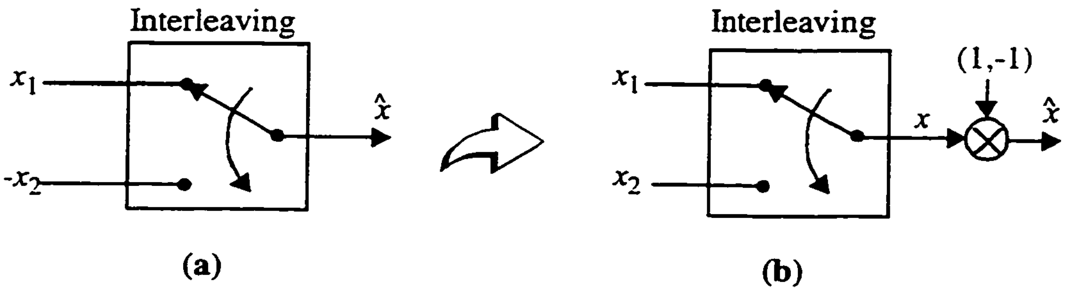
In (3.9), the first term is the desired signal. The second term is the quantization noise and is second-order shaped. The third term is an extra error E_{m1} introduced by mismatch in the first integrator which is,

$$E_{m1} = \frac{1}{2}(\hat{X}z^{-1/2} + \hat{Y})z^{-1}\delta_1 . \tag{3.10}$$

The last term in (3.9) is an extra error E_{m2} introduced by mismatch in the second integrator, that is, $E_{m2} = (\hat{U}z^{-1/2} + \hat{Y})(1 - z^{-1/2})\delta_2$ which is first-order shaped. In an oversampling system, $(1 - z^{-1/2})$ is small and so the unshaped term E_{m1} due to the first integrator dominates. Improving SNR requires us to focus on the first integrator. This will be discussed in Sections 3.2 and 3.3.

C. Interpretation

Note from (3.8) and (3.10) that the errors introduced by mismatch of the first integrators in the first- and second-order ΔΣ modulators are similar except for one more delay z^{-1} in (3.10). The error introduced is proportional to the mismatch error δ_1 and contains two terms $\hat{X}$ and $\hat{Y}$. Observing the result from two interleaved signals in (3.1) and (3.2), one can interpret $\hat{X} = (X_1 z^{-1/2} - X_2)$ as the output of two time-interleaved discrete-time signals X_1 and $-X_2$ as shown in Figure 3.4(a).



(a)

(b)

Figure 3.4 Mismatch is a mixing process

In the time-domain, $\hat{x}$ is used to denote $\hat{X}$. Equivalently, $\hat{x}$ can be seen as a process of mixing two interleaved signals x_1 and x_2 clocked at f_c (the input signal x at $2f_c$) with a periodic sequence {1, -1, 1, -1,...}. This process is shown in Figure 3.4(b). Actually, the periodic sequence is a discrete cosine signal with its frequency f_C, clocked at the effective sampling rate $2f_C$. In the z-domain, $\hat{X}$ is the result of mixing the signal X with a carrier signal whose radian frequency is π. Similarly, $\hat{Y}$ is the result of mixing the signal Y with a carrier signal whose radian frequency is π.

The spectral translation from X to $\hat{X}$ is shown in Figure 3.5(a). It is noted that X is shifted by π in frequency to the position which is higher than π. It turns out that a foldover image is obtained. As discussed in Chapter 2, the output Y contains both the input X and the shaped quantization noise. The spectral translation from Y to $\hat{Y}$ is depicted in Figure 3.5(b). Note that quantization noise in $\hat{Y}$ is shaped by a high pass filter.

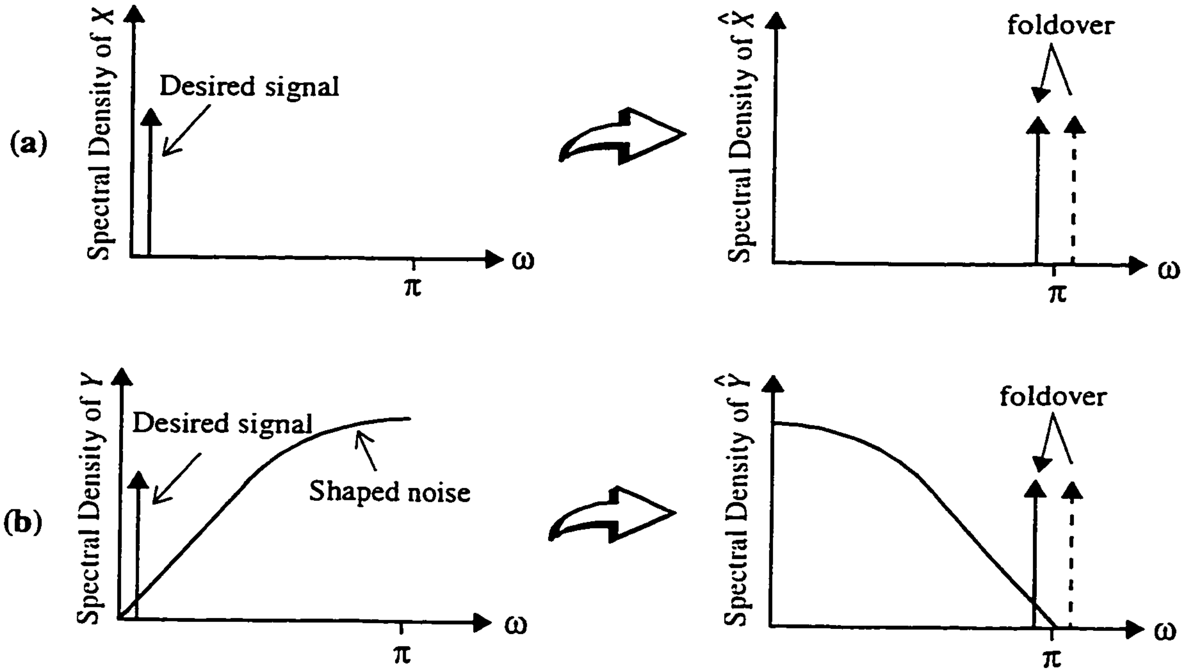


Figure 3.5 Spectral translation due to mismatch

Observing (3.8) and (3.10), one can see that there are three basic errors introduced by mismatch in the first integrator. They are:

(1) *Signal error*: This error is translated from the desired signal due to path mismatch. It is contained in both $\frac{1}{2}\hat{X}\cdot\delta_1$ and $\frac{1}{2}\hat{Y}\cdot\delta_1$. Path mismatch translates a small portion of the

desired signal by an amount of about $\frac{1}{2}\delta_1$ to the location around π. The signal error shown in Figure 3.6 will be filtered by the subsequent decimation filter and therefore is harmless.

(2) *Interference aliasing error*: Any interferer around π will be aliased to the desired band, attenuated by $\frac{1}{2}\delta_1$ as depicted in Figure 3.6. Hence the antialiasing filter should be designed to have sufficient attenuation around π in order to meet the required SNR. This is in contrast to a single-sampling modulator. Before the signal at around π is aliased to the desired band, it is attenuated by 40 to 60 dB for $\delta_1 = 0.1 \sim 1\%$. The cost for the antialiasing filter should not be an issue since the wider transition band due to oversampling.

(3) *Noise error*: This error appears in the term $\frac{1}{2}\hat{Y} \cdot \delta_1$. Path mismatch translates the quantization noise energy around π by an amount of $\frac{1}{2}\delta_1$ to baseband, as shown in Figure 3.6. The in-band noise floor increases due to the noise error. Hence the achievable SNR is limited. This is the most harmful term.

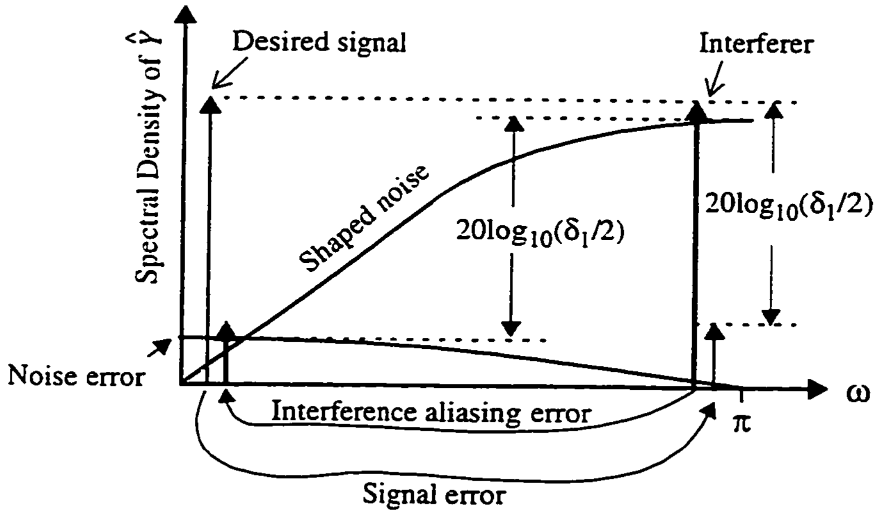

Spectral Density of $\hat{Y}$
Desired signal
Interferer
Shaped noise
$20\log_{10}(\delta_1/2)$
$20\log_{10}(\delta_1/2)$
Noise error
ω
Interference aliasing error
π
Signal error

Figure 3.6 Effects due to mismatch

In summary, mismatch creates images of noise and input signal. In a lowpass ΔΣ ADC, the image of the quantization noise located at the clock rate f_c degrades SNR, often limiting double-sampled ADC to 10-bit. The image of the signal is out of the band.

D. Non-Uniform Sampling

Non-uniform sampling occurs when an exact 50% duty cycle in a clock generated by a clock generator can not be attained. We show in this section that non-uniform sampling only create an image to the input signal and does not affect the SNR.

As shown in Figure 3.7, the falling edge of ϕ_2 at instant n lags that of ϕ_1 at instant $(n-1)$ by $\frac{1}{2}(1+\alpha)T$, and leads that of ϕ_1 at instant n by $\frac{1}{2}(1-\alpha)T$, where T is the clock period and α is the clock phase error relative to $T/2$ caused by non-uniform sampling. In Figure 3.1, the inputs to the comparators on phases ϕ_1 and ϕ_2, in the z-domain, can be written as,

$$V_1 = \frac{X_1 z^{-1} + X_2 z^{-\alpha/2}}{1-z^{-1}} - \frac{Y_1 + Y_2 z}{1-z^{-1}}, \tag{3.11}$$

$$V_2 = \frac{X_1 z^{-1} z^{\alpha/2} + X_2 z^{-1}}{1-z^{-1}} - \frac{Y_1 z^{-1} z + Y_2}{1-z^{-1}}, \tag{3.12}$$

where

$$z^{\pm\alpha/2} = e^{\pm j\alpha\omega T_c/2} = \cos\left(\left(\frac{1}{2}\alpha\omega T_c\right) \pm \sin\left(\frac{1}{2}\alpha\omega T_c\right)\right). \tag{3.13}$$

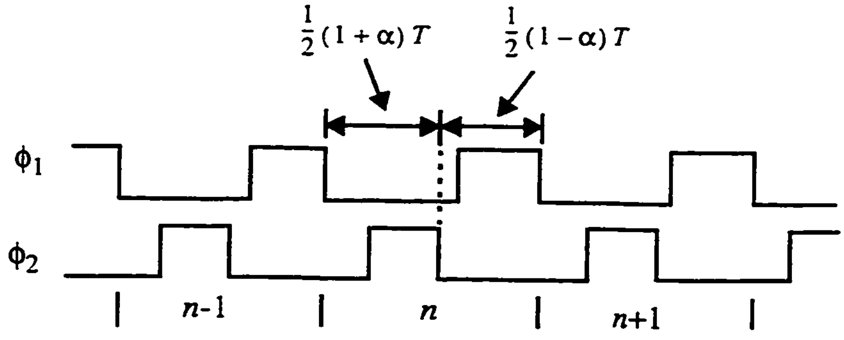


Figure 3.7 Non-uniform sampling clocks

By substituting (3.11), (3.12) and (3.13) into (3.5) and (3.6), and then substituting the results into (3.2), then we can obtain the output in the first-order modulator,

$$Y = \cos\left(\frac{1}{2}\alpha\omega T\right)z^{-1}X + (1 - z^{-1/2})E + \frac{z^{-1}}{1 + z^{-1/2}}\sin\left(\frac{1}{2}\alpha\omega T\right)\hat{X}. \tag{3.14}$$

For the second-order ΔΣ modulator in Figure 3.3, the output in the desired band can be obtained as (see Appendix A.1),

$$Y \cong \cos\left(\frac{1}{2}\alpha\omega T\right)z^{-3/2}X + 4(1 - z^{-1/2})^2E + \frac{z^{-3/2}}{1 + z^{-1/2}}\sin\left(\frac{1}{2}\alpha\omega T\right)\hat{X} \tag{3.15}$$

Notice from (3.14) and (3.15) that the error introduced by non-uniform sampling is different from that by gain mismatch. Non-uniform sampling only introduces signal error and it does not affect the quantization noise. In a practical case $\alpha \ll 1$, the signal error may contains the following error terms:

- For the signal within the band of interest, $\omega T \ll 1$ (due to oversampling), $\cos\left(\frac{1}{2}\alpha\omega T\right) \approx 1$ and $\sin\left(\frac{1}{2}\alpha\omega T\right) \approx \frac{1}{2}\alpha\omega T$. As explained in the previous subsection, the signal is translated to the location around π to create an error term. For a ΔΣ ADC, this error will be added to the quantization around π which will be filtered out by a subsequent decimation and lowpass digital filter.
- As explained in the previous subsection, any signal around π will be attenuated and aliased to the desired band. The attenuation is approximately,

$$\left|\frac{z^{-3/2}}{1 + z^{-1/2}}\sin\left(\frac{1}{2}\alpha\omega T\right)\right| \cong \frac{1}{2}\alpha\pi \tag{3.16}$$

Hence the only concern is the interferer located around π which degrades the SNR. This can be mitigated by designing an antialiasing filter before the modulator. The filter should be designed to have sufficient attenuation around π in order to meet the required SNR. When the interferer around π is aliased to the desired band, it is attenuated by an amount defined in (3.16) which typically is 30 to 50 dB for $\alpha = 0.1 \sim 1\%$. If we design a 16-bit ΔΣ data converter, the antialiasing filter may need to provide 45 to 65 dB attenuation at π. The cost for the antialiasing filter should not be an issue because of the relaxed filter transition band (oversampling).

To reduce the error introduced by non-uniform sampling, either a high EOSR or a clock with a high precision 50% duty cycle should be used, as noted from (3.16). To achieve a high-precision clock, one can first create a clock with twice the required frequency and then use a frequency divider (by 2) to generate the required clock.

3.1.2 Bandpass Delta-Sigma Modulators

A bandpass ΔΣ modulator can be obtained by applying a low-to-bandpass transformation of $z \rightarrow -z\frac{z+a}{az+1}$, $-1 < a < 1$ [Nors97], to a lowpass counterpart. With this transformation, there is no change on the mixing effects caused by nonidealities, as can be seen from (3.7), (3.9), (3.14) and (3.15).

It has been noted in a lowpass double-sampled ΔΣ modulator that path mismatch and non-uniform sampling affect the signal and noise performance. In a lowpass case, the effects are harmless to the signal assuming that the signal rolls off before π in mismatch and non-uniform cases. Mismatch introduces extra noise in the band of interest. However, this is not the case for a bandpass double-sampled ΔΣ modulator. The signal error and noise error in such a modulator are analyzed as follows:

(1) *Signal error*: Assume that the desired signal is located at ω_1. From the analysis in last section, it is known that mismatch translates the desired signal to a frequency of

$$\omega_2 = \pi - \omega_1, \tag{3.17}$$

attenuated by around δ_1. Hence the spurious signal produced by mismatch is an image of the desired signal with respect to a radian frequency of $\pi/2$, attenuated by an amount of δ_1, as shown in Figure 3.8(a). Consider a bandpass ΔΣ whose center frequency is one-fourth of the effective sampling rate (that is, the radian center frequency $\omega_0 = \pi/2$). The spurious signal is an attenuated image of the desired signal with respect to the center frequency ω_0.

There are two cases:

- For a narrowband receiver where the signal is centered at $\omega_1 = \pi/2$ (double-sided signal), the image will be folded to itself when the signal is downconverted to baseband. The spectrum of the image is reversed. In this case, mismatch is harmless since the image is 40 ~ 60 dBc with a typical mismatch error of 0.1 ~ 1 %.
- If the signal is not centered ($\omega_1 \neq \pi/2$) such as in a wideband receiver, any interferer located near ω_2 will create an attenuated image near the desired signal frequency ω_1. This error is often harmful since the interferer may be > 60 dBc for some wireless standards such as TDMA IS-54 in North America.

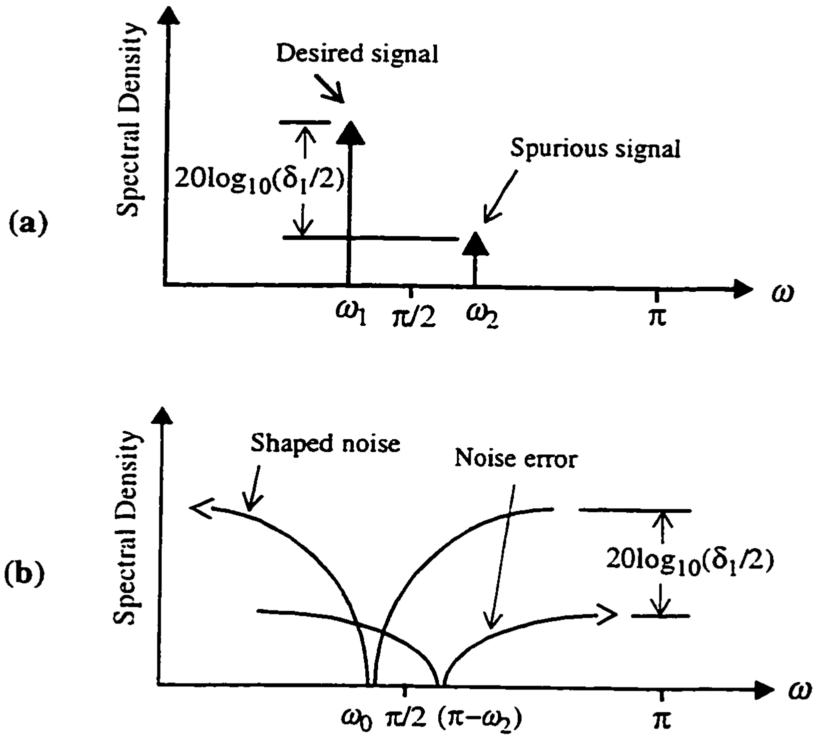


Figure 3.8 Errors generated by mismatch in a bandpass modulator: (a) the signal error and (b) the noise error

(2) *Noise error*: Any noise energy around the frequency of $\pi - \omega_0$ will be translated to the band of interest centered at ω_0, where ω_0 is the center frequency of a bandpass $\Delta\Sigma$ modulator. In general, a small portion of out-of-band noise energy at $\pi - \omega_0$ will be folded into the band of interest and will degrade the SNR as shown in Figure 3.8(b). For a band-

pass $\Delta\Sigma$ whose center frequency is one-fourth of the effective sampling rate ($\omega_0 = \pi/2$), the noise will be self-folded by an amount of $\delta_1/2$, which is harmless.

3.2 A Novel Double-Sampling Technique

The double-sampled integrator used in the first stage of the feedback loop in Figure 3.1 and Figure 3.3 is a backward Euler integrator. This integrator has the drawback of introducing an error when there is capacitor mismatch. The error will directly contribute to the modulator output, hence limiting the achievable SNR. The backward Euler integrator is depicted in Figure 3.9(a), where y_1 and y_2 are the signals from the $\Delta\Sigma$ modulator output through two DACs. Considering capacitor mismatch, we can express the two gains as,

$$k_{11} = 1 + \frac{1}{2}\delta_1,$$

$$k_{12} = 1 - \frac{1}{2}\delta_1,$$

where $\frac{1}{2}\delta_1$ represents the relative path mismatch error. The output can be obtained as,

$$U = -\frac{1}{1 - z^{-1/2}}Y + \frac{1}{2}\frac{\hat{Y} \cdot \delta_1}{1 - z^{-1/2}}. \tag{3.18}$$

Note that the second term is the error introduced by mismatch. The error can be modeled as shown in Figure 3.9(b). As can be seen, the error $\frac{1}{2}\delta_1\hat{y}$ directly contributes to the input of the integrator, hence introducing an error in the output spectrum of the double-sampled $\Delta\Sigma$ modulator.

Note from (3.9) that the error introduced by mismatch in the second integrator is first-order shaped. This means that any error presented to the input of the second integrator will be first-order shaped. If the error $\frac{1}{2}\delta_1\hat{Y}$ presented in the input of the first integrator can be moved to the output or, say, to the input of the second integrator, a first-order shaped error will be obtained.

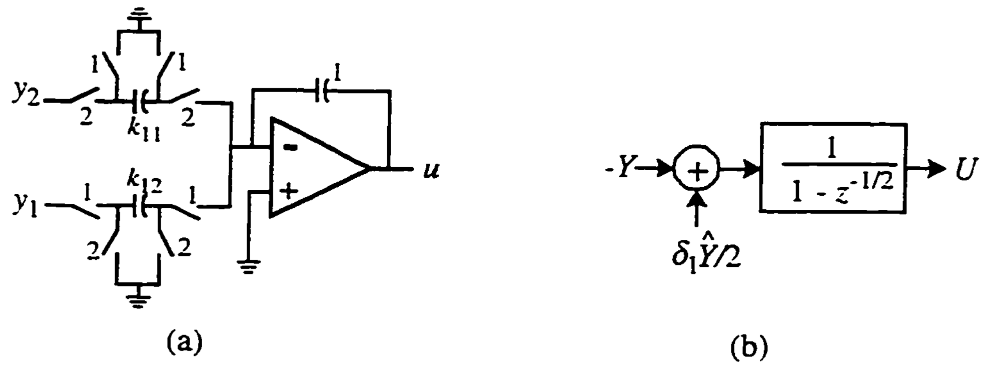


Figure 3.9 A double-sampled backward Euler integrator and its mismatch error model

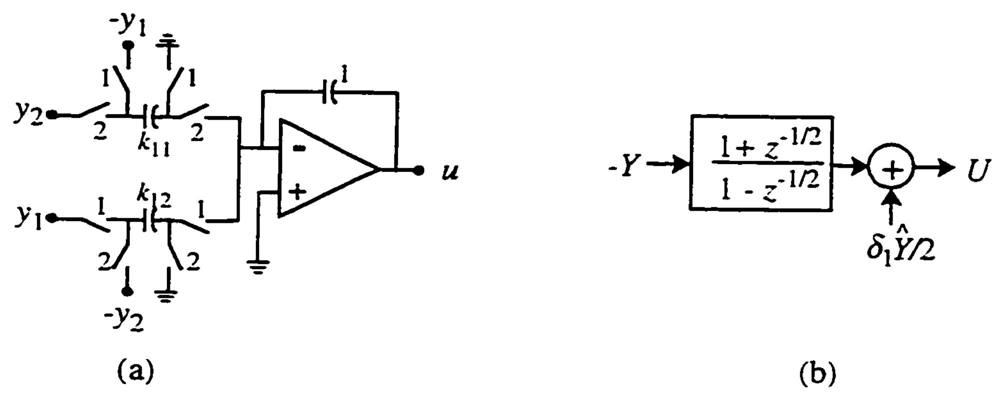
A double-sampled bilinear integrator[1] shown in Figure 3.10(a) can be used to serve this purpose. In the proposed double-sampled integrator, the feedback signal (y_1 or y_2) is sent to both paths. This is contrary to the backward Euler integrator where each path only processes one feedback signal. As a result of the fact that both capacitors are used on both phases, the error introduced by mismatch is partially cancelled. We will show that it is differentiated. The differentiated error is then integrated. Hence, the error bypasses to the output without being processed. The output of the double-sampled bilinear integrator can be derived as,

$$U = -\frac{1+z^{-1/2}}{1-z^{-1/2}}Y + \frac{1}{2}\hat{Y} \cdot \delta_1 . \tag{3.19}$$

The error model is depicted in Figure 3.10(b). Note that the error $\frac{1}{2}\delta_1\hat{Y}$ introduced by mismatch is output-referred.

Now we are in a position where we are able to construct a double-sampled ΔΣ modulator to reduce the mismatch effect. The first backward Euler feedback integrator can be replaced with the proposed bilinear integrator.

[1] A single-sampled bilinear integrator was proposed in [Rahi78].



(a) (b)

Figure 3.10 A double-sampled bilinear integrator and its mismatch error model

3.3 Novel Double-Sampled Lowpass Delta-Sigma Modulators

The mismatch error in the novel double-sampled bilinear integrator is moved to the output due to the differential operation. By using the bilinear integrator as the first feedback integrator, novel double-sampled ΔΣ modulators which are insensitive to mismatch can be obtained. We show in this section that the mismatch errors are first-order shaped.

3.3.1 First-Order Modulator

A novel first-order double-sampled ΔΣ modulator is shown in Figure 3.11 which is obtained by replacing the feedback integrator with a bilinear integrator.

The error due to mismatch appears at the integrator output and hence is first-order shaped when input-referred. In practice, the input path does not need a bilinear integrator since the signal error around π introduced by mismatch can be removed by digital filters.

The linear model for the modulator is shown in Figure 3.12. Note that the strobed comparator introduces a half-period delay in contrast to a full-delay in a single-sampling ΔΣ modulator. This model is used to derive the coefficients in the modulator. The design goal is to choose parameters in order for the signal and noise transfer functions to be low-pass and high-pass types respectively, as described in Section 2.2. One of the choices for the parameters is discussed in Appendix A.2, from where we have,

$$k_2 \cdot g_1 \cdot g_2 = \frac{1}{3}, \tag{3.20}$$

$$k_1 \cdot g_1 = \frac{2}{3}, \tag{3.21}$$

where g_1 and g_2 are the gains of the two-level comparator and two-level DAC respectively. Gains g_1 and g_2 could be arbitrary due to the use of a one-bit quantizer. As a consequence, coefficients k_1 and k_2 can be any values. This freedom allows us to optimize the modulator as will be discussed in section 3.4. Often gain g_1 determines the quantization noise level (rms value) [Arda87] and gain g_2 defines the dynamic range of the input signal x. A conventional modulator would use equal gains $k_1 = k_2$, but in our case the term $(1 + z^{-1/2})$ due to the bilinear operation has a gain of 2 at DC. Therefore, $2k_1 = k_2$ is often true. The gains g_1 and g_2 are set to 1 for simplicity. Thus, $k_1 = \frac{2}{3}$ and $k_2 = \frac{1}{3}$.

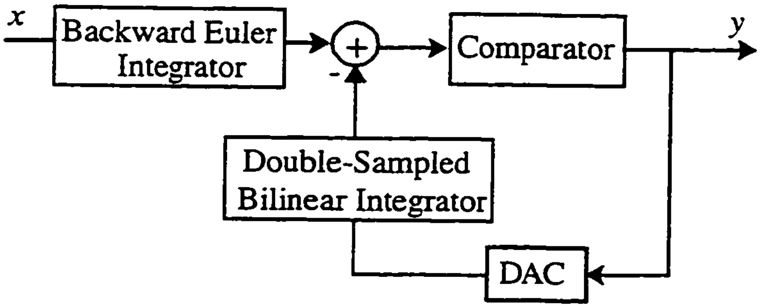


Figure 3.11 A novel first-order double-sampled delta-sigma modulator

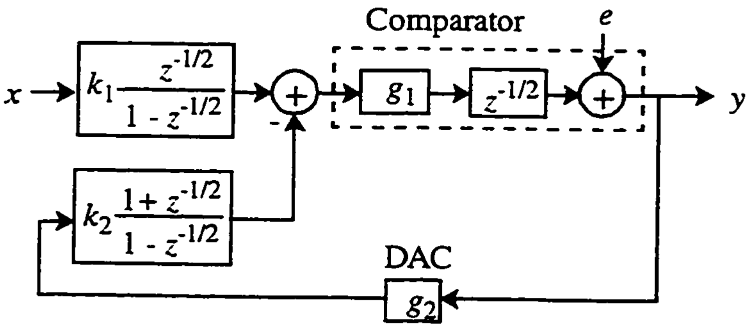

Comparator
e
x
$k_1 \frac{z^{-1/2}}{1 - z^{-1/2}}$
+
-
g_1
$z^{-1/2}$
+
y
$k_2 \frac{1 + z^{-1/2}}{1 - z^{-1/2}}$
DAC
g_2

Figure 3.12 A linear model for the novel first-order modulator

$$k_{21} = \frac{1}{3}\left(1 + \frac{1}{2}\delta_2\right)$$

$$k_{22} = \frac{1}{3}\left(1 - \frac{1}{2}\delta_2\right).$$

where δ_i ($i = 1, 2, 3$) are the relative mismatch errors. The signal transfer function is a low-pass type and therefore the modulator output Y in the desired band is approximately as,

$$Y \equiv Xz^{-1} + \frac{3}{2}(1 - z^{-1/2})E + \frac{1}{2}\tilde{X}z^{-1}\delta_1 + \frac{1}{4}z^{-1}(1 - z^{-1/2})\tilde{Y} \cdot \delta_2. \tag{3.22}$$

The error introduced by mismatch is first-order shaped as can be seen from (3.22). Any quantization noise or interferer around f_c will be attenuated by mismatch error and also first-order shaped before it is folded back to the desired band. Since mismatch δ_2 is 0.1% ~ 1% in a typical CMOS technology, the error introduced by mismatch is much smaller than the first-order shaped quantization error. Hence, the error introduced by mismatch in the proposed first-order double-sampled modulator is negligible.

B. DAC Reference Mismatch

It is to be noted that mismatch between two reference voltages V_{rp} and V_{rn} in Figure 3.13 will not introduce a DC component in the output. Assume,

$$V_{rp} = \Delta V_r + V_r \text{ and}$$

$$V_{rn} = \Delta V_r - V_r,$$

where

$$V_r = \frac{V_{rp} - V_{rn}}{2} \tag{3.23}$$

$$\Delta V_r = \frac{V_{rp} + V_{rn}}{2} \tag{3.24}$$

The common-mode voltage ΔV_r is canceled due to double sampling. The effective reference voltages are the differential-mode voltages V_r and $-V_r$. This allows us to use single-ended references in a practical realization.

3.3.2 Second-Order Modulator

A novel second-order double-sampled $\Delta\Sigma$ modulator employing the proposed bilinear integrator is depicted in Figure 3.14. The first feedback integrator is replaced since it is a dominant contribution to SNR degradation.

The errors introduced by mismatches in integrators in Figure 3.14 appear at the output of the first integrator and hence are first-order shaped in the output. As with the first-order modulator, the input feedforward path does not need a bilinear integrator since the signal error around π introduced by mismatch can be removed by digital filters.

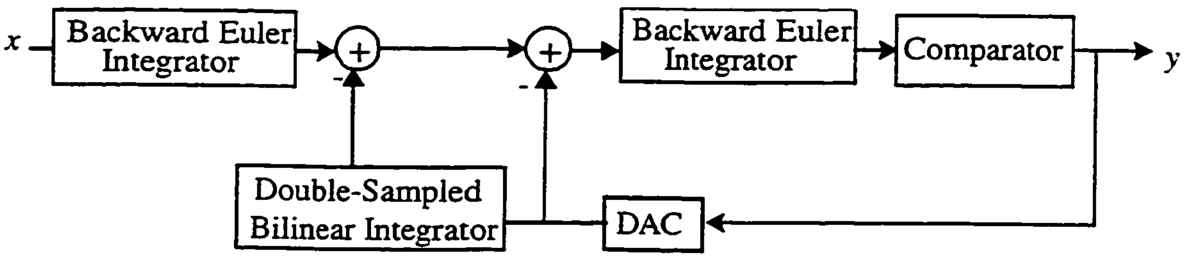

x
Backward Euler Integrator
+
-
+
-
Backward Euler Integrator
Comparator
y
Double-Sampled Bilinear Integrator
DAC

Figure 3.14 A novel second-order double-sampled delta-sigma modulator

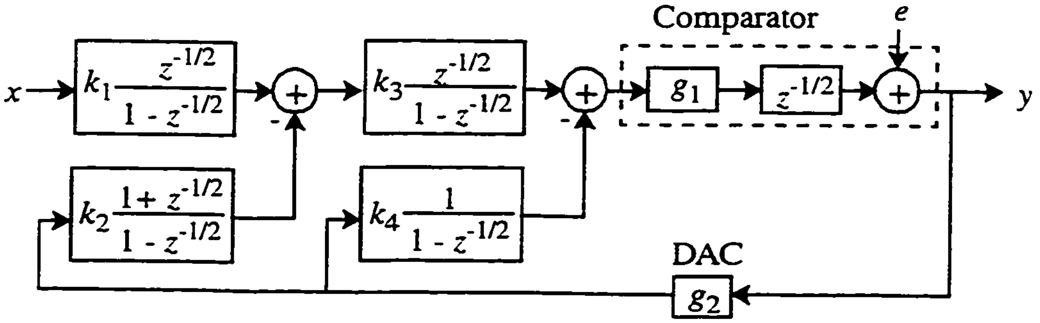

Comparator
e
x
$k_1 \frac{z^{-1/2}}{1 - z^{-1/2}}$
+
-
$k_3 \frac{z^{-1/2}}{1 - z^{-1/2}}$
+
-
g_1
$z^{-1/2}$
+
y
$k_2 \frac{1 + z^{-1/2}}{1 - z^{-1/2}}$
$k_4 \frac{1}{1 - z^{-1/2}}$
DAC
g_2

Figure 3.15 A linear model for the novel second-order modulator

The linear model for the modulator is shown in Figure 3.15. This model is used to derive the coefficients in the modulator. To achieve second-order shaping of quantization noise and a unity signal gain, the following should be satisfied (see Appendix A.3),

$$5k_2k_3 = k_4 \tag{3.25}$$

$$k_1 = 2k_2g_2 \tag{3.26}$$

where k_1, k_2, k_3, and k_4 are integrator gains. There are five unknowns in two equations. In order to share the same circuit (capacitors and switches) for the two integrators in the implementation, we can have $k_3 = k_4$. From (3.25), $k_2 = \frac{1}{5}$. Since gains g_1 and g_2 can be arbitrary, hence k_1 and k_3 can be any values. This freedom allows us to optimize the modulator as will be discussed in section 3.4. Often gain g_1 determines the quantization noise level [Arda87] and gain g_2 defines the dynamic range of the input signal x. A conventional modulator would use equal gains $k_1 = k_2$, but in our case, the term $(1 + z^{-1/2})$ due to the bilinear operation has a gain of 2 at DC. Therefore $2k_1 = k_2$ is often true. The gains g_1 and g_2 are set to 1 for simplicity. Hence, $k_1 = \frac{2}{5}$. Let $k_3 = k_4 = \frac{5}{6}$.

A SC circuit for the double-sampled ΔΣ modulator can be obtained by replacing the feedback integrator in Figure 3.3 by the bilinear integrator shown in Figure 3.10. The circuit is shown in Figure 3.16. The feedback and feedforward paths share the same second integrator as shown in the figure. An important feature of the circuit is it inherent pipelined mode. The input signals (x, v_{y1} and v_{y2}) propagate towards the output in a pipelined way.

The principle of this pipelined structure is as follows: On phase ϕ_1, capacitor k_{11} takes charge from input x. Capacitor k_{12} delivers charge to the first op-amp output and also to charge capacitor k_{31}. Capacitor k_{32} delivers charge to the second op-amp output which is sent to the top comparator input. The top comparator starts to compare with a reference based on the input. At the end of ϕ_1 or at the begin of ϕ_2, the output y_2 will be latched. In the feedback, capacitor k_{21} discharges and capacitor k_{22} takes charge from v_{y1} which is controlled by y_1. Output y_1 is latched by the lower comparator on phase ϕ_1. On phase ϕ_2, a similar operation is performed.

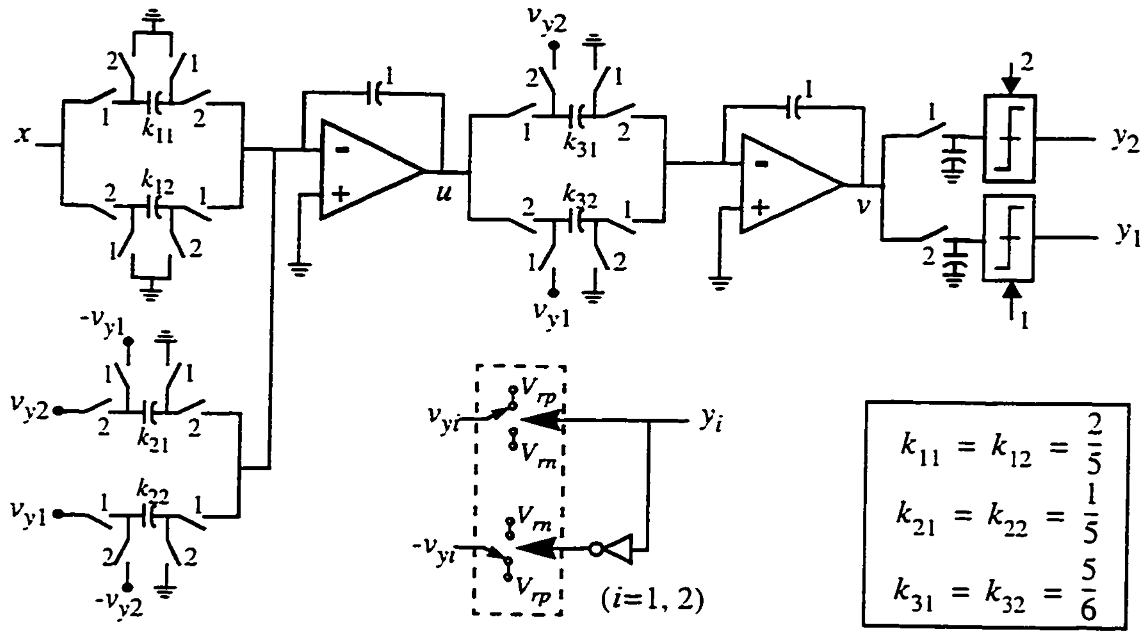


Figure 3.16 A novel second-order SC double-sampled delta-sigma modulator

A. Gain Mismatch

To analyze the mismatch effect, we express the capacitor coefficients in Figure 3.13 as,

$$k_{11} = \frac{2}{5}\left(1 + \frac{1}{2}\delta_1\right),$$

$$k_{12} = \frac{2}{5}\left(1 - \frac{1}{2}\delta_1\right),$$

$$k_{21} = \frac{1}{5}\left(1 + \frac{1}{2}\delta_2\right),$$

$$k_{22} = \frac{1}{5}\left(1 - \frac{1}{2}\delta_2\right),$$

$$k_{31} = \frac{5}{6}\left(1 + \frac{1}{2}\delta_3\right),$$

$$k_{22} = \frac{5}{6}\left(1 - \frac{1}{2}\delta_3\right),$$

where δ_i ($i = 1, 2, 3$) are the relative mismatch errors. The signal transfer function is a lowpass type and therefore the modulator output Y in the desired band is approximately as,

$$Y \cong Xz^{-3/2} + 3\,(1 - z^{-1/2})^2 E \qquad . \tag{3.27}$$
$$+ \frac{1}{2}\tilde{X}z^{-3/2} \cdot \delta_1 + \frac{1}{4}z^{-1}\tilde{Y}\,(1 - z^{-1/2})\,\delta_2 + \frac{5}{4}\,(\tilde{U}z^{-1/2} + \tilde{Y})\,(1 - z^{-1/2})\,\delta_3$$

Note that first-order shaping is achieved for the error introduced by mismatch. Compared to the conventional double-sampled version in (3.9), 6 dB per octave of rejection of mismatch noise is gained in this novel second-order modulator. The mismatch error will be further investigated in section 3.4, where the reduced mismatch requirements are given. For typical OSR and mismatch error δ_1, the mismatch noise becomes negligible.

To get an idea of how well the bilinear integrator works, the second-order double-sampled ΔΣ modulator shown in Figure 3.16 was simulated with Matlab [Matl92] and SWITCAP2 [Suya92]. The power spectral comparison for conventional and novel double-sampled ΔΣ modulators in shown in Figure 3.17. Here, the "conventional" double sampling ΔΣ modulators are referred to those in Figure 3.1 and Figure 3.3.

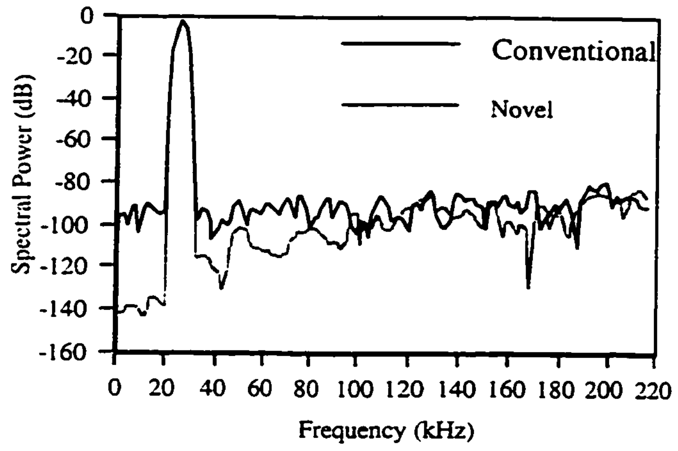


Figure 3.17 Spectral comparison between the conventional and novel second-order double-sampled delta-sigma modulators, EOSR = 128

In the simulations, the EOSR is 128 and the mismatch error is 0.4 %. The input signal is a f = 25.6 KHz sinusoid with an amplitude of -3.74 dB. The band of interest is 51.2 KHz. It can be seen that the noise floor for the novel one is much lower due to first-order shaping of mismatch, especially within the frequency band of interest. The difference can be as small as 10 dB or as large as 40 dB at frequencies below 100 kHz. In the simulation, a minimum 4-term window was used, which can achieve a minimum sidelobe as low as -98.17 dB [Nutt94].

B. DAC Reference Mismatch

It is to be noted that mismatch between two reference voltages V_{rp} and V_{rn} in Figure 3.16 will not introduce any offset. The mismatch is cancelled by the bilinear operation of the integrator as discussed in last section.

C. Stability Consideration

Through simulations, it is found that the output signals of the two integrators have signal swings like those of the single-sampling second-order ΔΣ modulator [Hurs92]. They are bounded to certain ranges. These ranges are determined by the capacitor ratios. The input level to the second integrator can be adjusted by changing k_{31} and k_{32}. The dynamic range optimization is discussed in 3.5.

3.3.3 Higher-Order Modulators

Higher-order (> 2) ΔΣ modulators can be implemented either as single-loop [Lee87] or cascaded [Long88], [Uchi88] structures. We focus on the cascaded cases where a second order modulator is cascaded with a first- or a second-order modulator to create a third- or fourth-order modulators respectively.

The first-stage second-order modulator will be the novel double-sampled ΔΣ modulator shown in Figure 3.16. The second-stage modulator can be any conventional double-sampled ΔΣ modulator since the mismatch effect is second-order shaped. A third-order double-sampled ΔΣ modulator is depicted in Figure 3.18 where coefficients $k_1 = 0.4$, $k_2 = 0.2$, $k_3 = k_4 = 5/6$, $k_5 = k_6 = 0.5$, $k_7 = 1.45$, $k_8 = 0.35$. A conventional first-order double-sampled ΔΣ modulator is used to cascade with the novel second-order modulator. The SC circuit for the third-order modulator can be easily obtained from Figure 3.18 and the expression for the mismatch noise can be obtained from (3.7) and (3.27).

(see Appendix A.3),

$$N_{conven} \cong 2.3\delta\,(EOSR)^{-1/2}e_{rms} \qquad (3.28)$$

where δ is the capacitor mismatch error and e_{rms} is defined in (2.2). For the novel second-order double-sampled $\Delta\Sigma$ modulator shown in Figure 3.14, the rms noise power N_{novel} in the desired band aliased from the noise at f_c due to mismatch is (see Appendix A.3),

$$N_{novel} \cong 7.9\delta\,(EOSR)^{-3/2}e_{rms} \qquad (3.29)$$

The noise terms introduced by mismatch which are given in (3.28) and (3.29) degrade the achievable SNR. The SNR degradation in the conventional and novel $\Delta\Sigma$ modulators as a function of gain mismatch is shown Figure 3.19. In the simulation, the op-amp DC gain is 60 dB and the EOSR is 128. The input signal is a f = 25.6 KHz sinusoid with a peak amplitude of -6 dB. The band of interest is 51.2 KHz. In the figure, one can see the analytical results are in good agreement with the simulation results.

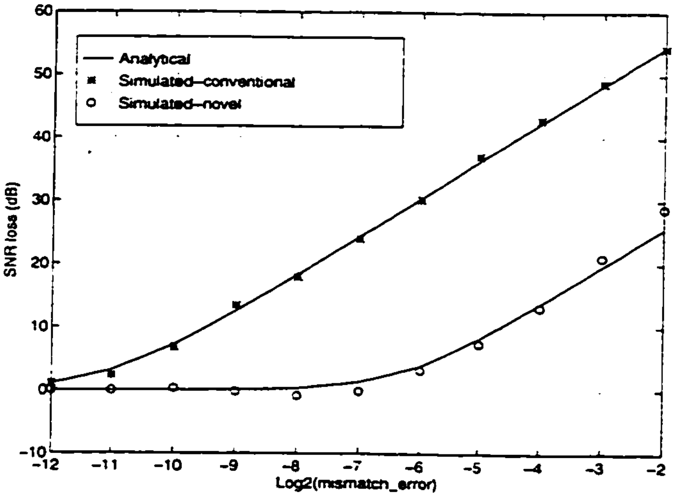


Figure 3.19 SNR degradation versus mismatch error in second-order delta-sigma modulators, EOSR = 128

As shown in the figure, the sensitivity to mismatch has been substantially reduced in the novel modulator due to the first-order shaping of the mismatch error. For example, there is about 3 dB degradation in SNR with 1.5 % of mismatch error (i.e., log2(mismatch_error) = -6) in the novel structure in contrast to 30 dB SNR degradation in the conventional one. With 0.4 % mismatch (i.e., log2(mismatch_error) = -8), the conventional modulator loses 18 dB. This loss exceeds what we can gain from double-sampling which is 15 dB.

Note that there is 6 dB SNR degradation for every doubling of the mismatch error in the conventional version. In the novel structure, the noise in the desired band is dominated by the second-order shaped quantization noise when the mismatch error is less than 1.5 %. Beyond 1.5 %, the mismatch error begins to dominate the total output and the SNR degrades 6 dB for every doubling of the mismatch error as can be seen from Figure 3.19. The distance between two parallel lines is 31 dB. The distance in dB can be obtained from dividing (3.28) by (3.29) as $20\log 10(EOSR) - 10.7$ dB, which is the maximum SNR improvement achievable in the novel modulator compared to the conventional version.

The slopes of the quantization noise power and the mismatch noise power are different. There is a cross point for those two noise power curves. For an M^{th}-order double-sampled $\Delta\Sigma$ modulator, the slope of the quantization noise power is $(6M + 3)$ dB/octave. The mismatch noise power is first-order shaped and its slope is 9 dB/octave + $20\log_{10}(\delta)$. When EOSR increases, the mismatch noise power eventually will exceed the quantization noise power at a cross point. The cross point is determined by the EOSR, the order of modulator and the mismatch error.

To illustrate this, the quantization noise and the mismatch noise in the novel second-order double-sampled modulator are plotted Figure 3.20, where the mismatch error is 1.5 %. The mismatch noise is from (3.29). As can be seen, the slopes for both errors are different (15 dB/octave and 9 dB/octave for quantization noise and mismatch noise respectively). At a lower EOSR, quantization noise dominates the total noise while mismatch noise dominates at a higher EOSR. They are across at $EOSR \cong 128$ (i.e., $\log 2\,(EOSR) \cong 7$) as shown in the figure.

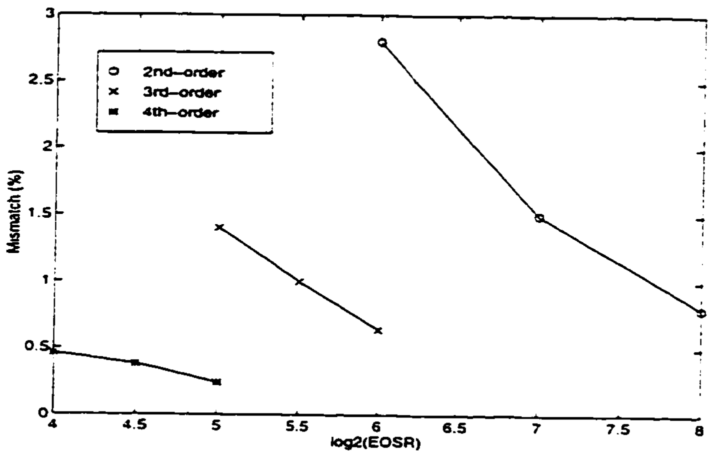


Figure 3.21 3dB-mismatch versus the EOSR in novel double-sampled delta-sigma modulators

3.5 Implementation of a Second-Order Double-Sampled Modulator

The main focus of this chapter is dealing with an architecture level solution to combating mismatch effects. To prove the idea from the circuit level, in a recent collaboration with Ash Swaminathan [Swam97], an implementation of a second-order double sampling $\Delta\Sigma$ modulator in 0.25 μm SOI (silicon-on-insulator) technology has been submitted for fabrication. All circuits are designed by Ash to work at a power supply of 0.9 volts. The architecture is exactly the one shown in Figure 3.16. Circuit optimization to reduce non-idealities [Nors97] has not been a major concern.

The technology we chose is quite aggressive. Due to the process complexity, the chip fabrication has had several consecutive failures and we have been waiting for the chip for more than one year. Hence, this section provides HSPICE simulations to verify the developed architecture and no test result is provided for further verification.

A. Optimization of Coefficients

There is a freedom to choose coefficients in double-sampled $\Delta\Sigma$ modulators. For the first-order double-sampled $\Delta\Sigma$ modulator, coefficients k_1 and k_2 can be any values due to arbitrary gains of the one-bit quantizer. For the second-order double-sampled $\Delta\Sigma$ modulator, there are five unknowns in two equations (3.25) and (3.26). Since g_1 and g_2 can be arbitrary, hence k_1, k_2, k_3 and k_4 can be any values.This freedom allows us to optimize the $\Delta\Sigma$ modulator in terms of the dynamic ranges of each integrator input, SNR, etc. By simulation, we optimize the integrator output ranges. As a result, we obtain $k_1 = \frac{2}{5}$ and $k_3 = k_4 = \frac{1}{2}$. With this choice, the outputs of the first and second integrators remains bounded within (V_{rp}, V_{rn}). The simulated distributions of the integrator output signs levels are shown in Figure 3.22 where the horizontal axis is normalized to the DAC reference level defined in (3.23). These coefficients can avoid integrator output clipping which may introduce in-band noise.

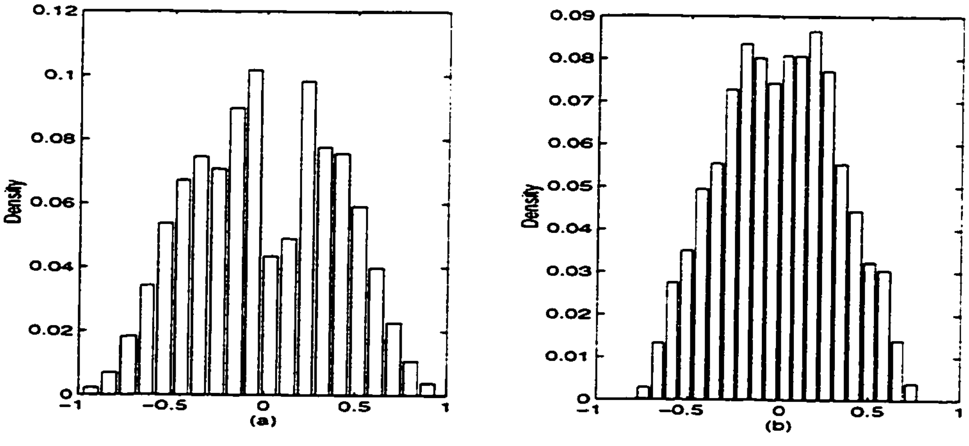


Figure 3.22 Integrator output level distributions

B. Circuit Design and Simulation

The design of circuits for a $\Delta\Sigma$ modulator has been well documented in chapters 7 and 11 of [Nors97] and in [Greg86]. The key components include the op-amp, comparator, clock generator. Single-ended integrators are designed.

The transconductance op-amp used in the implementation is shown in Figure 3.23 [Park86]. The circuit is an inverter. The circuit is a cascode op-amp and has a high output impedance. The *bias*1 and *bias*2 are used to bias the cascode. The advantage of this circuit is its simplicity and hence is suitable for low power applications. The open loop dc gain is about 40 dB for this circuit.

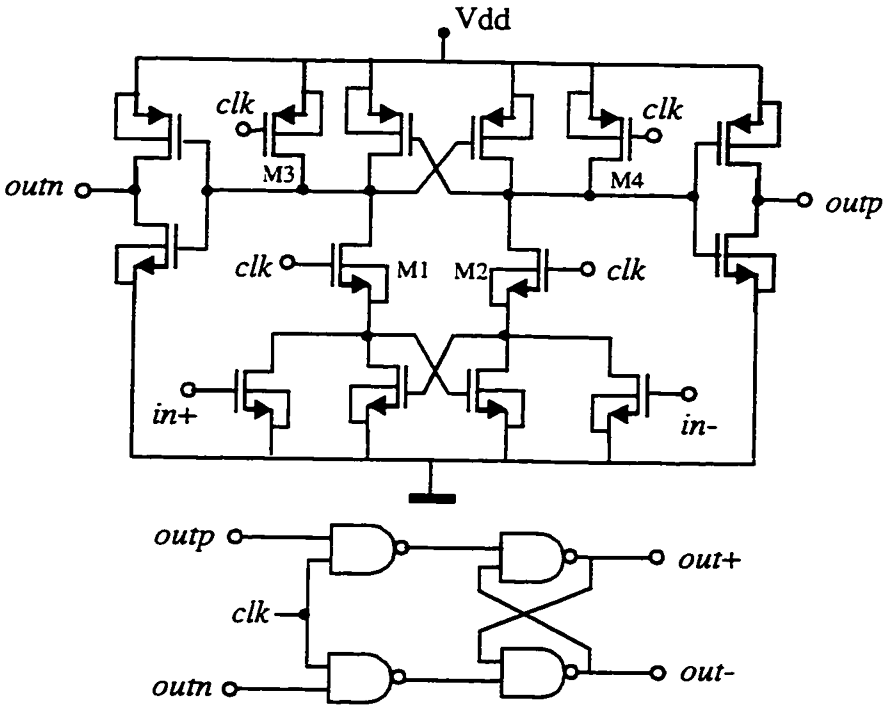


Figure 3.24 A regenerative feedback comparator

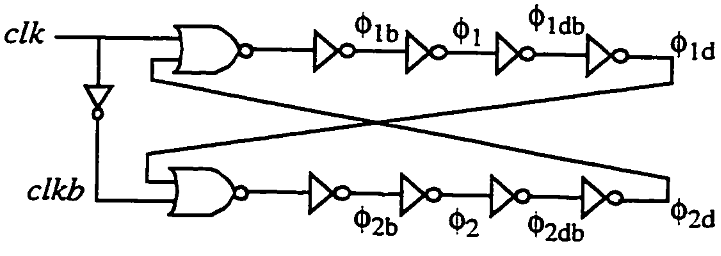


Figure 3.25 A clock generator

The effects of the circuit nonidealities in a general sense other than those discussed in 3.1 on the performance of a ΔΣ modulator have been discussed by many papers [Nors97], [Bose88]. These nonidealities may include integrator nonidealities (leak, gain error, bandwidth, slew rate), comparator hysteresis, nonlinearities, $1/f$ noise, clock jitter, etc. The results also apply to the double-sampled case. Therefore, they are not discussed here.

The circuit level simulation of the designed double-sampled ΔΣ modulator has been simulated with HSPICE [HSPI91] and the output spectrum is plotted in Figure 3.26. In the simulation, 4096 points were obtained. The clock rate is 50 MHz and hence the effective sampling rate is 100 MHz. The fundamental signal frequency is 501 KHz. Therefore, the EOSR is 100. The simulated SNR is about 81 dB. The estimated power is 1 mW at 50 MHz clock rate compared to a low power second-order ΔΣ modulator consuming 2.5 mW at a 4 MHz clock rate [Rabi97]. The power saving is achieved by the combination of the double-sampled technique and a low power SOI technique.

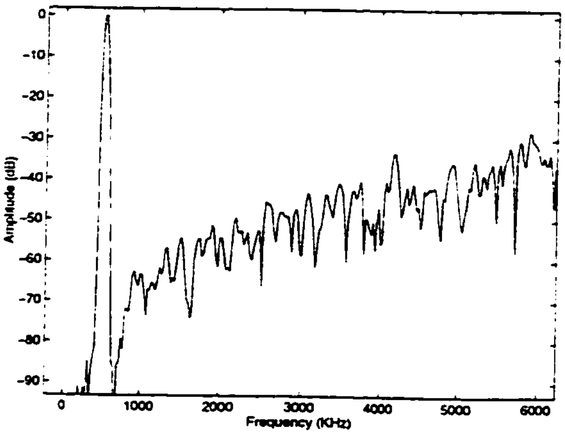


Figure 3.26 Spectrum from HSPICE simulation

3.6 Summary

In this chapter, we have provided a simple practical solution to mitigate the mismatch effect in lowpass double-sampled ΔΣ modulators. The novel modulators can be used in digital baseband digitization receivers discussed in Chapter 2. We have shown that SNR degradation is negligible compared to the improvement that can be obtained by double-sampling. To better understand the mechanism of the capacitor mismatch effect on the performance, we have conducted a quantitative analysis.

We have shown the mismatch in the first feedback integrator is a dominant contribution to the error and it creates images of quantization noise and input signal. Non-uniform sampling only create an image to the input signal and it does not affect quantization noise. In a lowpass ΔΣ ADC, the image of the quantization noise located at the clock rate f_c degrades SNR, often limiting double-sampled ADC to 10~12-bit resolutions. The image of the signal is out of the band. In a bandpass modulator, the image of an interferer may be in the band and may degrade the SNR.

We have developed new architectures for double-sampled lowpass $\Delta\Sigma$ modulators where the first feedback backward Euler integrator is replaced by the novel bilinear integrator. These double-sampled modulators are quite insensitive to gain mismatch since a first-order shaped mismatch error is achieved. With the new architecture, we reduce the SNR loss to less than 3 dB with a typical mismatch of 0.1 % ~ 0.5 %.

The higher order novel doubled-sampled $\Delta\Sigma$ modulators need tighter mismatch requirements. We have provided design guidelines on the mismatch requirements for second, third- and fourth-order modulators and have shown the mismatch causing 3-dB SNR loss is achievable in a typical CMOS technology (the worst case is 0.25 % for a fourth-order modulator with EOSR of 32). Therefore, the novel architectures make the double-sampled $\Delta\Sigma$ modulation practical.

An implementation of a novel second-order double-sampled second-order $\Delta\Sigma$ modulator for the low power application has been reported. The simulated SNR is 81 dB when EOSR is 100 and the estimated power is 1 mW at 50 MHz clock rate compared to a low power second-order $\Delta\Sigma$ modulator consuming 2.5 mW at 4 MHz clock rate [Rabi97].

Chapter 4
Design of Multi-Stage Polyphase CIC Decimators

Decimators are key components in modern digital radio receivers. In a lowpass ΔΣ modulator based radio architecture, they are used to decimate the high speed stream. They are also the major components to build DDCs for bandpass ΔΣ modulated signals. Basically there are two requirements for such a circuit -- low power and high speed.

Usually, the first decimation in a multi-stage decimation chain is a CIC decimator due to its simplicity [Cand86], [Nors97]. However, the full-rate part of the CIC which often consists of 3 or more accumulators with a typical width of 16~24 bits dominates power consumption and limits clock rate. This is true for both low power ΔΣ decimation [Bran94] and GHz-rate ΔΣ decimation [Gao97], [Jens95]. As explained in section 2.3, multi-stage CIC decimators do not help much in achieving low power and GHz-rate ΔΣ decimation.

Another problem is the time misalignment which occurs when the bandpass ΔΣ modulated signal is downconverted by a DDC at half the sampling rate [Saul90]. Time misalignment creates sidebands at the output.

Until recently, no publications tried to address the above problems except for misalignment [Rice82], [Rade84], [Pell92]. One reason, from the author's view, is that there is possibly a lack of researchers who understand the power issues and are simultaneously familiar with ΔΣ modulators, DSP and digital receivers.

A solution to the above problems is to combine multi-stage CIC decimators with polyphase techniques. With the techniques, we can decompose the high-rate signal into multi-phase signals which are processed at a much lower rate. There are many considerations in designing such multi-stage polyphase CIC decimators. In this chapter, we provide not only a solution to decimation for low power and GHz-rate ΔΣ modulators, but, more importantly, a design method to a multi-stage polyphase CIC decimator.

This chapter is organized as follows. In Section 4.1, multi-stage polyphase architectures for CIC decimators are derived. Design considerations are discussed in detail in Section

4.2. The aliasing requirements are discussed to determine the order of the first CIC decimator. A design scheme for polyphase components without employing multipliers and adders, budget for word-length, design procedure, etc. are also detailed. In Section 4.3, decimation for double-sampled and GHz-rate $\Delta\Sigma$ modulators is discussed to demonstrate the advantages of the multi-stage polyphase CIC decimators in low power and GHz-rate applications. An implementation of a polyphase DDC is given in Section 4.4, which provides a concrete example for design considerations. A summary is given in Section 4.5.

This chapter provides the following new contributions:

- We show that the combination of multi-stage CIC decimators with polyphase techniques is a solution to low power and GHz-rate decimation for $\Delta\Sigma$ modulators as well as the removal of time-misalignment in DDCs.
- Provide methods to determine the polyphase components to simplify them and meet aliasing attenuation requirement, to design polyphase components without multipliers and adders and save 2~3 bits, and to choose the word-length in multi-stage polyphase CIC decimators.
- Provide a design procedure for multi-stage polyphase CIC decimators.
- A FPGA implementation of a multi-stage polyphase DDC at 100 MHz is detailed. We show a 5x power saving can be made by using the new design compared to the conventional multi-stage DDC.

4.1 Multi-Stage Polyphase Architectures

CIC decimators were reviewed in Section 2.3. There are some problems with this single-phase architecture:

- It is difficult to implement a CIC decimator for lower power or GHz-rate $\Delta\Sigma$ modulators because of its full-rate cascaded wide word-length accumulators.
- There are multi-phase signals at the output of a parallel delta-sigma modulator (e.g., double-sampled one).
- Time misalignment exists in a DDC which consists just of a decimator.

A conventional decimation method using FIR or single CIC decimator is not a solution. Although Chu in [Chu84] used multiple CIC decimators as a general rule to design filters, the above two issues have not been addressed and resolved, as discussed in Section 2.3.

We propose to combine polyphase techniques with multi-stage CIC decimators to address these problems. However, there are two issues when we develop the polyphase architecture from Figure 2.9. Firstly, the CIC decimator is in neither a direct-form nor a transpose direct-form structure, which is normally used to derive a polyphase architecture [Croc83], [Vaid93]. Secondly, there is no method in the literature to design such a multi-stage polyphase CIC decimator. The first issue is addressed below and the second issue will be discussed in the next section.

Consider a multi-stage architecture consisting of two cascaded CIC decimators shown in Figure 4.1 (a). The first and second decimators are N_1-stage and N-stage, respectively. For generality, the numbers of stages of these two CIC decimators may be different. The first and second decimation ratios are R_1 and R_2 respectively. The total downsampling ratio R is $R = R_1R_2$. The transfer functions for these two CIC decimators are:

$$H_1(z) = \left(\frac{1-z^{-R1}}{1-z^{-1}}\right)^{N1} \tag{4.1}$$

$$H_2(z_1) = \left(\frac{1-z_1^{-R2}}{1-z_1^{-1}}\right)^{N} \tag{4.2}$$

where $z = e^{j\omega T}$, $T = 1/f_s$. Note that (4.1) - (4.2) are expressed with respect to the input sampling rate f_s. With respect to sampling rate f_s/R, we define $z_1 = z^{R1}$. The whole transfer function of these two decimators is,

$$H(z) = \left(\frac{1-z^{-R1}}{1-z^{-1}}\right)^{N1}\left(\frac{1-z_1^{-R2}}{1-z_1^{-1}}\right)^{N} \tag{4.3}$$

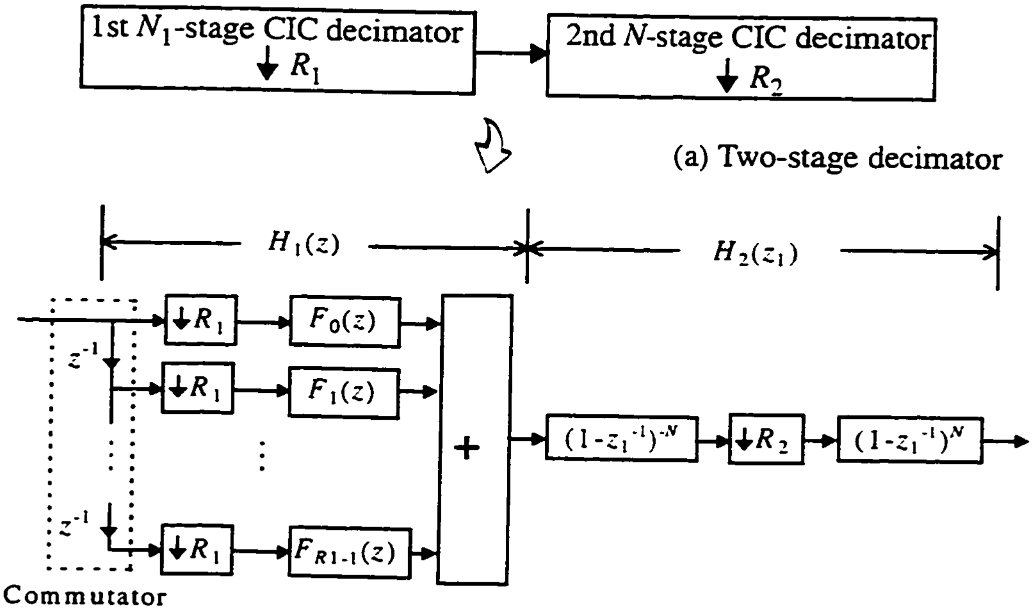
Since the first decimator operates at the full sampling rate f_s, it dominates the power consumption and limits the circuit speed. A solution is to decompose the first CIC decimator into a polyphase filter. It is not straightforward to decompose the first CIC decimator by equation (4.1) because it is in neither a direct-form nor a transpose direct-form. To achieve polyphase decomposition, equation (4.1) needs to be expanded as,

$$H_1(z) = (1+z^{-1}+\ldots+z^{-(R1-1)})^{N1}. \tag{4.4}$$

Then by applying polyphase decomposition techniques [Vaid93] to (4.4), we can obtain a polyphase form as,

$$H_1(z) = \sum_{i=0}^{R1-1} z^{-i} F_i(z^{R1}) , \qquad (4.5)$$

where $F_i(z^{R1})$, $i = 0, 1, \ldots R_1 - 1$, are polyphase components which operate at $1/R_1$ of the input sampling rate, f_s/R_1 . The polyphase architecture for a CIC decimator is shown in Figure 4.1 (b). The filter shown in the figure is called a multi-stage polyphase CIC decimator since it is derived from two decimators.



(a) Two-stage decimator

(b) Two-stage polyphase

Figure 4.1 Multi-stage polyphase architecture of a CIC decimator

Note that the multi-stage polyphase CIC decimator shown in Figure 4.1 (b) has the following features: (1) it accepts multi-phase input signals, and (2) it operates at a reduced rate f_s/R_1 except for the input interface commutator which may be implemented by a array of D flip-flops as will be discussed in section 4.2.2.

There are four design parameters and they are R_1, R_2, N_1 and N. Since the second CIC decimator operates at a lower rate, there is no significant power issue in the choices of R_2

and N. The requirements on N_1 conflict for the complexity and aliasing attenuation. Other design issues to be considered are simple implementation of polyphase components without multipliers and even without adders, the maximum word-length required in each stage, etc. We focus on N_1 and R_1 in the following design considerations.

Note that the idea can also be applied to interpolation. The polyphase architecture of a CIC interpolation filter can be derived similarly. A commutator is used in the output stage to interleave polyphase signals to form a high-speed output.

4.2 Design Considerations

There are many considerations in designing a multi-stage polyphase decimator:

- One major concern in design is to determine the minimum number of stages N_1 in the first polyphase CIC decimator. The complexity of polyphase components is determined by N_1 as can be seen from (4.4) and a smaller N_1 is preferred. The noise and interference rejection is determined by N_1 which favors a large N_1. A trade-off should be made to compromise between these two.
- Another factor is the downsampling ratio R_1. A large R_1 makes the polyphase components complicated but reduces the clock rate of the circuits (and saves power).
- Simplification of the implementation of polyphase components reduces the word-length in each stage and hence further reduces the power consumption.
- Budgeting the word-length in each stage is necessary to minimize the required word-length and hence reduce the power consumption.

We discuss these issues in the following subsections, particularly bullets 1, 2 in Section 4.2.1, bullet 3 in Section 4.2.2, bullet 4 in Section 4.2.3. The design method is summarized in Section 4.2.4.

4.2.1 Aliasing Attenuation and Droop

Conventionally, it is recommended that $N_1 = M + 1$, i.e., one greater than the modulator order M [Cand86]. We show that N_1 can be reduced by 1 or 2 for many practical high speed $\Delta\Sigma$ modulators whose SNR is not quantization noise limited, hence reducing the complexity of polyphase components and saving more power.

A. Aliasing Attenuation

In designing a conventional CIC decimator like that shown in Figure 2.9, there are two considerations when we determine the aliasing attenuation (or the number of stages N). One is the presence of tone interferers that can alias in-band, located within a bandwidth $2f_0$ of multiples of f_s/R, where R the is downsampling ratio. Another is quantization noise in the same bands. These two aliasing terms in the image bands will be folded and accumulated into the baseband. They will cause SNR degradation.

Conventionally, the number of stages required in a CIC decimator are $(M + 1)$, where M is the order of a lowpass ΔΣ modulator [Cand86]. This guideline applies only for quantization noise aliasing. For example, the ideal SNR for a second-order ΔΣ modulator is about -95 dB for OSR = 128. According to [Cand86], the required N is 3. This CIC decimator gives about 50 dB attenuation before an interferer located at $f_s/R - f_0$ folds into baseband. If, for example, this interferer has the same amplitude as the signal, an interferer with -50 dBc appears at baseband. The SNR then is reduced to about 45 dB. To reduce the SNR degradation, we can either design an anti-aliasing filter having 45 dB attenuation at $f_s/R - f_0$ or increase N to a larger number such as $N = 5$. This is especially significant for radio applications where a number of interferers can be nearby.

The same principle can be applied in designing a multi-stage polyphase CIC decimator. However, we have three parameters to optimize instead of one for a required downsampling ratio $R = R_1R_2$. These design parameters are R_1, N_1 and N. Since the second CIC decimator operates at a lower rate, we can afford to have a sufficiently large N. Hence, only R_1 and N_1 are considered here.

Tone Interference Rejection

As a general rule, the design target is to minimize N_1 to simplify polyphase components. N_1 is important in the polyphase CIC decimator design since it determines the aliasing attenuation. The aliasing terms are located in the vicinity of multiples of f_s/R_1 within a bandwidth of $2f_0$. The minimum attenuation A_{min} is at $f_1 = f_s/R_1 - f_0$ which is,

$$A_{min} = \left(\frac{\sin(R_1 \pi f_1 T)}{R_1 \sin(\pi f_1 T)}\right)^{N1} \tag{4.6}$$

within the band of interest will fall into the desired band and will accumulate at baseband. There is an extra noise power folded into the desired band by decimation.

Conventionally we require $N_1 = M + 1$ in order to mitigate the impact of the aliased noise [Cand86]. In designing a multi-stage polyphase CIC decimator, the complexity is determined by N_1. Hence, a minimum N_1 is desirable. We will show that in a typical case, reducing N_1 to $N_1 = M$ has only a 6 dB penalty over the ideal case. This is often insignificant, for example, when thermal noise limits are 10 dB above the ideal quantization limit.

For an M^{th} order lowpass ΔΣ modulator, its noise density in the z-domain can be expressed, in general, as $N(z) = H_E(z) E(z)$, where

$$H_E(z) = (1 - z^{-1})^M \tag{4.8}$$

and $E(z)$ is the quantization noise whose spectral density is defined in (2.3). This shaped noise spreads over the oversampling frequency and has high energy at high frequencies. With this modulator, the rms noise in the desired band as a function of modulator order M is given approximately by [Cand92],

$$P_0(M) \cong \frac{\pi^M}{\sqrt{2M+1}} \left(\frac{1}{OSR}\right)^{M+0.5} e_{rms} \tag{4.9}$$

Assume $N_1 = M - d$, where d is a positive integer. The conventional design takes $d = -1$. The noise at the output of the first polyphase CIC decimator is $E_{d3}(z) = H_E(z) (H_1(z) / R_1^{N_1}) E(z)$ which is,

$$E_{d3}(z) = \left(\frac{1 - z^{-R_1}}{R_1}\right)^{M-d} (1 - z^{-1})^d E(z) \tag{4.10}$$

Note that $E_{d3}(z)$ is the noise after being downsampled to f_s/R_1. This is shown in Figure 4.3. The first term in (4.10) is,

$$\left(\frac{1 - z^{-R_1}}{R_1}\right)^{M-d} = \left(\frac{1 - z_1^{-1}}{R_1}\right)^{M-d}$$

which is approximately an $(M - d)^{th}$ order noise shaping function within the band of interest after being downsampled.

The second noise term in (4.10) is defined as $E_{d1}(z) = (1-z^{-1})^{d}E(z)$ which is d^{th}-order noise shaped. When it is downsampled, it becomes $E_{d2}(z)$. The high frequency energy of this noise will be folded back to the Nyquist band when it is downsampled to f_s/R_1. The total noise power is[1]

$$Power = \int_{0}^{f_s/2} \left|E_{d1}(e^{j2\pi(f/f_s)})\right|^2 df = \frac{(2d)!f_s}{(d!)^2}E^2(f) \tag{4.11}$$

where $E(f)$ is the spectral density of the quantization noise defined in (2.3). Therefore, the power spectral density at the lower rate f_s/R_1 is approximately,

$$E_{d2}(f_1) \cong \frac{\sqrt{(2d)!R_1}}{d!}E(f_1) \ , 0 \le f_1 < \frac{f_s}{2R_1} \tag{4.12}$$

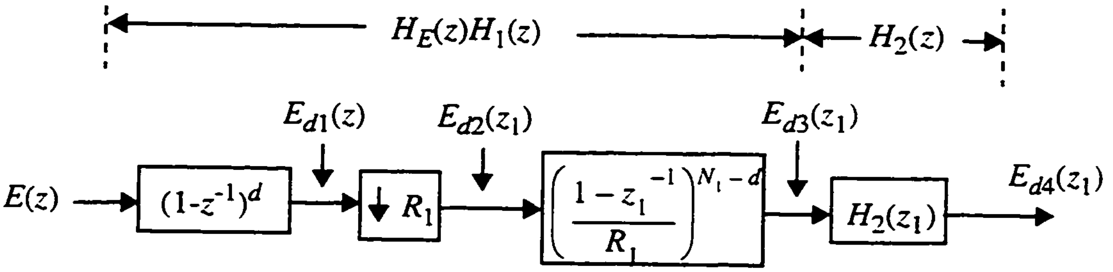


Figure 4.3 Decimation for the lowpass shaped noise

Since we choose $N = (M + 1)$ in the second CIC decimator $H_2(z)$ due to its lower rate, the noise power increase in the desired band is negligible [Cand86]. Hence, the noise power increase is only introduced by the first polyphase decimator. The rms noise power P_4 in the desired band at the output of a multi-stage polyphase CIC decimator is approximately,

$$P_4 \cong \frac{\sqrt{(2d)!R_1}}{d!}P_0(M-d) \tag{4.13}$$

[1] $\left|E_{d1}(e^{j2\pi(f/f_s)})\right| = 2^{d}(\sin(\pi f/f_s))^{d}E(f)$,
where $E(f) = e_{rms}\sqrt{2/f_s}$, $0 \le f < f_s/2$.

where 0! = 1.

Compared to the ideal achievable noise power defined in (4.9) for an M^{th}-order lowpass $\Delta\Sigma$ modulator, the noise power increase in the desired band is,

$$NoisePowerInc = 20\log 10\left(\frac{P_0(M)}{P_0(M-d)}\right) - 20\log 10\left(\frac{\sqrt{(2d)!R_1}}{d!}\right) \tag{4.14}$$

Consider two special cases:

(1) $d = 0$, the noise increase is $10\log 10(R_1)$ dB.

(2) $d = 1$, the noise increase is 20log10(OSR) + 10log10(R_1) - 6, dB.

Quantization Noise Rejection in Bandpass Case

As described in Section 2.2.1, a $2M^{th}$-order bandpass $\Delta\Sigma$ modulator can be obtained by replacing z^{-1} with $-z^{-2}$ from a lowpass counterpart. The noise transfer function is,

$$H_E(z) = (1+z^{-2})^M.$$

The noise notch of this bandpass modulator is located at $f_s/4$ and therefore a sequence {1,0,-1,0,...} can be used to downconvert the bandpass signal to lowpass where decimation occurs. The noise transfer function at baseband is,

$$H_E(z) = (1-z^{-2})^M \tag{4.15}$$

The OSR is reduced by 2 compared to its lowpass counterpart in (4.8). That is, the required OSR doubles in the bandpass case if we need the same SNR performance.

The spectrum of the bandpass modulated signal is symmetrical with respect to $f_s/4$. So is the signal after being mixed down. Therefore the noise power in the desired band does not increase even if we downsample the signal by 2 without filtering. This can be seen from equation (4.15) which becomes $H_E(z_1) = (1-z_1^{-1})^M$, where $z_1 = z^2$, after downsampling by 2. The spectral density of the quantization noise doubles. As a result, the noise power in the desired band remains the same.

When we choose N_1 equal to the modulator's order P in the first decimator of the multi-stage polyphase decimator, the noise increase is $10\log 10(R_1)$ - 3 dB. The SNR loss is 3 dB less in the bandpass case compared to that in the lowpass case.

Quantization Noise Rejection in Practical Cases

It is valuable to choose $N_1 = N - 1 = M$ since the polyphase components can be much simpler (see Table 4.2 to compare $N_1 = 2$ and 3). For example, the complexity is reduced by 2 when N_1 decreases from $N_1 = 3$ to 2 for a typical $R_1 = 4$ or 8. As a result, the noise power in the desired band increases by 6 and 9 dB (or, 3 and 6 dB) in a lowpass case (or, a bandpass case) respectively, relative to the ideal $\Delta\Sigma$ modulator. In an actual implementation of a $\Delta\Sigma$ modulator, however, the achievable SNR might be degraded due to circuit nonidealities (e.g., thermal noise, clock jitter, etc.) in comparison to the ideal SNR. This is especially true for a high speed modulator. For example, the reported SNR is 75 dB in a fourth-order two-path bandpass $\Delta\Sigma$ modulator with the effective OSR 200 [Ong97]. The SNR degradation is > 20 dB compared to the ideal case. In this case, we may not degrade the actual achievable SNR when choosing $N_1 = M$.

It seems that $d \geq 1$ is not practical since it reduces the order of the noise shaping by d. However, we may operate well below the ideal SNR for a very high speed $\Delta\Sigma$ modulator such as for those at GHz rates [Gao97]. In these cases, a lower order of polyphase CIC decimator may be used.

B. Passband Droop

The droop of the first polyphase decimator can be expressed as,

$$A_{droop} = \left(\frac{\sin (R_1 \pi f_0 T)}{R_1 \sin (\pi f_0 T)} \right)^{N1} \tag{4.16}$$

Figure 4.4 shows the droop at the cutoff frequency f_0 for different combinations of N_1 and R_1. Note that the droop is dependent of *IOSR* and slightly of N_1 and R_1. Since *IOSR* in the first polyphase filter is much larger than that in the second CIC decimator, the droop introduced in the first polyphase is negligible in comparison with that in the second filter. As an example, consider a system that needs over 70 dB of aliasing attenuation. One can choose N_1 to be 2 for $IOSR \geq 32$ or to be 3 for $IOSR \geq 8$. The droops due to the polyphase filter are around -0.05 dB for both cases.

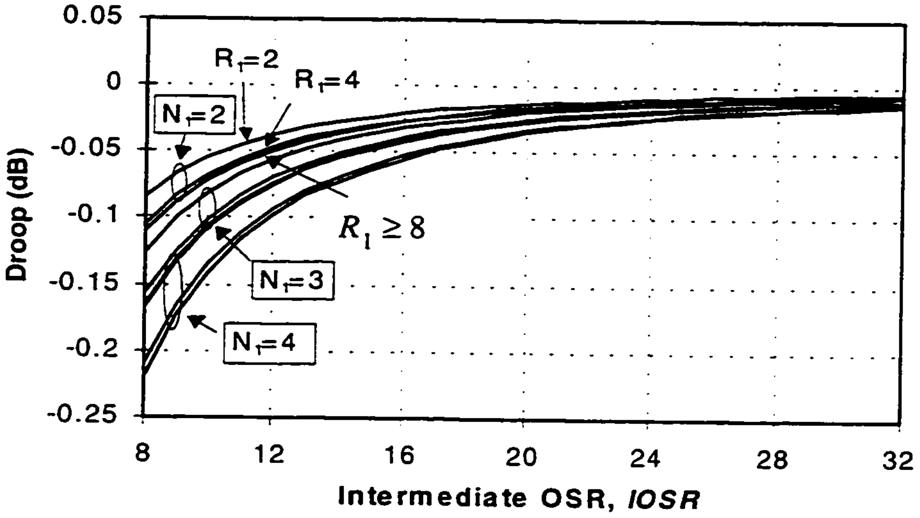


Figure 4.4 Droop at cutoff frequency versus *IOSR*

4.2.2 Polyphase Components and Commutators

To make the multi-stage polyphase CIC decimator attractive, simple design of polyphase components is the key. So is the high speed commutator. We propose a scheme to design the polyphase components, which can save 2~3 bits. This is extremely important since this also reduces the subsequent CIC decimator by the same amount of bits.

A. Polyphase Components

It is important to remain multiplierless in a multi-stage polyphase CIC decimator as in the original CIC decimator. The polyphase components are simple for cases where $N \leq 3$. Table 4.2 lists polyphase components for $N = 2$ and 3 which are derived from (4.4) and (4.5). In the tables, $R_1 = 2, 4$, and 8 are considered (note: R_1 can be any positive integer, not necessarily a power of 2). The notation $(c_0, c_1, \ldots)$ in the tables represents a polyphase component $(c_0 + c_1 z^{-1} + \ldots)$. The polyphase components and the integrators in the second CIC decimator operate at f_s/R_1 which is much lower than the input rate f_s. The design of a multi-stage polyphase CIC decimator to meet a given specification is described later. Note that the polyphase components in Table 4.2 are also suitable for an interpolator.

Table 4.2 Polyphase components for $N_1 = 2$ and $N_1 = 3$

N_1	2			3		
R_1	2	4	8	2	4	8
$F_0(z)$	(1, 1)	(1, 3)	(1, 7)	(1, 3)	(1, 12, 3)	(1, 42, 21)
$F_1(z)$	(2, 0)	(2, 2)	(2, 6)	(3, 1)	(3, 12, 1)	(3, 46, 15)
$F_2(z)$		(3, 1)	(3, 5)		(6, 10, 0)	(6, 48, 10)
$F_3(z)$		(4, 0)	(4, 4)		(10, 6, 0)	(10, 48, 6)
$F_4(z)$			(5, 3)			(15, 46, 3)
$F_5(z)$			(6, 2)			(21, 42, 1)
$F_6(z)$			(7, 1)			(28, 36, 0)
$F_7(z)$			(8, 0)			(36, 28, 0)

Note: $(c_0, c_1, \ldots)$ means that the component is $(c_0 + c_1 z^{-1} + \ldots)$.

A straightforward way to implement the polyphase components needs multipliers and adders. This can be seen from the polyphase components listed in Table 4.2. It is well-known that a filter's coefficients can be represented efficiently by a Canonical Signed-Digit (CSD) form [Lim83]. The CSD representation is a generic method to simplify hardware for a filter design. A CSD code represents coefficients as sums and differences of several powers of two. Since power-of-two multiplications can be obtained almost for free in a dedicated circuit implementation, the use of a CSD representation results in a substantial reduction in hardware complexity. As a result of CSD coding, a coefficient can be realized with a small number of adders. The CSD implementation needs only adders. However, these adders operate at the next highest rate in a polyphase CIC decimator after the commutator. For example, as many as 6 adders (some of them are 8~bit) are needed to implement a polyphase component $(15, 46, 3)$. For a ΔΣ modulated signal which is one or a few bit wide, there is still room to simplify the implementation.

A Proposed Implementation Scheme

When the bit-width of the ΔΣ modulated data stream is narrow, full adders are not necessary because only a small number of different outputs are possible. For the (15, 46, 3) case, only eight different values ±64, ±58, ±34 and ±28 can occur. A 3-bit ROM (read-only memory), or an equivalent logic circuit can replace the adders.

Let the i^{th} polyphase component $F_i(z)$ be,

$$F_i(z) = c_{i_0} + c_{i_1}z^{-1} + \dots + c_{iM_L}z^{-M_L}, i = 0, 1, \dots (R_1\text{-}1) \tag{4.17}$$

Hence, the output $w_i(n)$ of the i^{th} polyphase component can be expressed as,

$$w_i(n) = c_{i0}y_i(n) + c_{i1}y_i(n-1) + \dots + c_{iM_L}y_i(n-M_L) \tag{4.18}$$

where $y_i(n)$ is the output of the ΔΣ modulator.

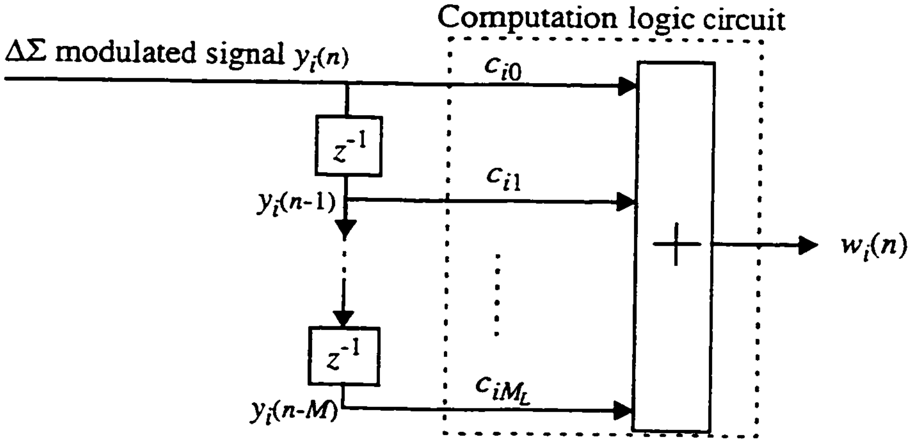


Figure 4.5 A polyphase component implemented by a computation logic circuit

Since the output of a ΔΣ modulator is either "1" representing +1 or "0" representing -1, a computation logic circuit can be designed to implement the polyphase component as shown in Figure 4.5. The inputs to the computation circuit are $\{y_i(n), y_i(n\text{-}1), \dots, y_i(n\text{-}M_L)\}$ and they are logical values (either "1" or "0"). Hence we save 1 bit compared to the 2-bit 2's complement input in a conventional implementation. The output $w_i(n)$ is in 2's complement which is often required by the subsequent CIC decimator.

To design the computation circuit, a truth table may be required. The inputs are $c_{i0}c_{i1}...c_{iM}$ and the output $w_i(n)$ which is represented in 2's complement notation. As can be seen from the polyphase components in Table 4.2, $w_i(n)$ are often even numbers. Thus, $w_i(n)$ may be scaled to reduce the bit-width. As will be demonstrated in section 4.4, the bit-width can be further reduced by 1~2 bits. This is important since it also reduces the required bit-width in the subsequent CIC decimator, saving more power. A detailed design of polyphase components is given in the design example described in Section 4.4.

In summary, the advantages of the implementation scheme are:

- It needs less bit-width, often 2~3 bits less. The "real" 1-bit input saves 1-bit in the proposed design scheme and scaling of $w_i(n)$ further reduces 1~2 bits. Reducing bit-width in polyphase components reduces the required bit-width in the subsequent adders for the combining of polyphase components and in the second CIC decimator by 2~3 bits.
- No adders are needed and only a simple computation logic circuit is required for a polyphase component. The simplicity will be demonstrated in a design example in Section 4.4. The adders are required in combining polyphase components.

B. High Speed Commutators

Another important issue in designing a polyphase filter is to demultiplex the incoming signal to multi-phase signals and align the signals in phase. These signals are then sent to polyphase component. This is accomplished by a commutator which may be constructed with D flip-flops controlled by appropriate clocks. This part is critical in terms of high speed operation since it operates at a full-rate f_s. Figure 4.6 shows a commutator which demultiplexes incoming signal into R_1-phase signals. The output frequency f_1 is f_s/R_1. Note that the D flip-flops in the first column operate at the input rate. For a high-speed input rate such as GHz-rate, they would typically require high speed CMOS dynamic devices [West94] or silicon bipolar circuits [Stou93]. Although the sampling rate for the second column is f_s/R_1, the requirements such as setup time, hold time and delay are the same as those defined in the first column. This is because the D flip-flops in the second column are allowed a duration of $1/f_s$ to set up their output signals. The output signals are aligned in the second column of D flip-flops. These flip-flops are clocked at the same rate as the comparator in the ΔΣ modulator, and may have a similar circuit implementation. It should, therefore, generally be possible to implement them in the same technology. It may be an option to integrate the commutator into the ΔΣ chip.

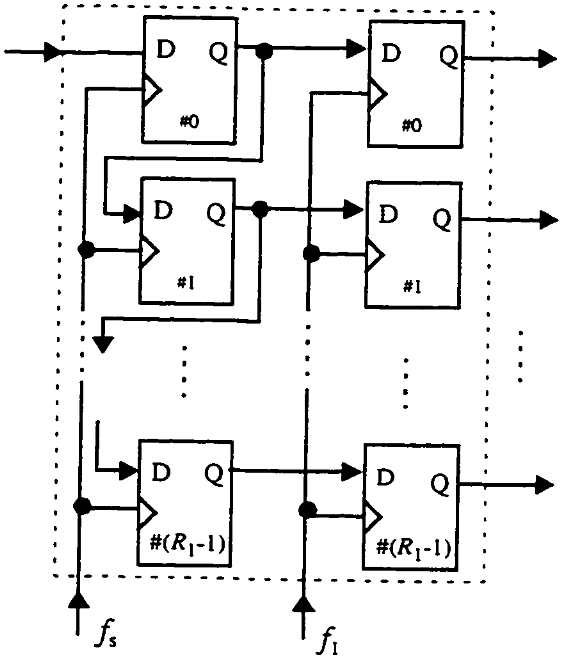


Figure 4.6 A polyphase commutator

4.2.3 Word-Length Budget

The word-length budget is important in digital design and is to specify the minimum bit-width in each stage without degrading the required SNR. Low power consumption is achieved by minimizing the bit-width in each stage. The word-length growth and truncation/rounding are discussed.

A. Word-Length Growth

Assume that two's complement number representation is used. The maximum word length growth B_{pi} at the output of the i^{th} phase signal in Figure 4.1 is

$$B_{pi} = \lceil \log 2(F_i(1)) \rceil - k_s = \lceil (N_1 - 1)\log 2(R_1) \rceil - k_s, \tag{4.19}$$

where $i = 1, 2, \ldots, (R_1 - 1)$ and k_s is the bit-width saving due to the proposed implementation scheme as discussed in section 4.2.2 and is often 2 or 3. The maximum output word-length growth is therefore $B_{pmax} = \lceil N_1 \log 2 (R_1) \rceil - k_s$.

The growth of word length in the polyphase CIC decimator is shown in Figure 4.7. Assume the word-length of the input data is $(B_{in} + 1)$. The maximum word length at the first polyphase filter output is $(B_{m1} + 1)$, where

$$B_{m'1} = \lceil N_1 \log 2 (R_1) \rceil - k_s + B_{in} \tag{4.20}$$

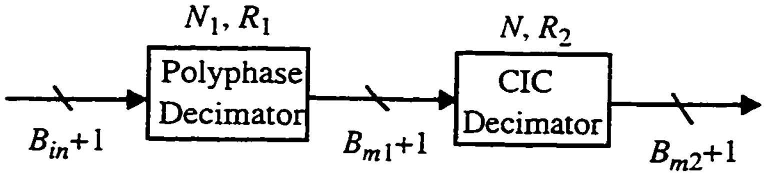


Figure 4.7 Word-length growth in a polyphase CIC decimator

Since the first polyphase decimator output is the input of the second CIC decimator, the maximum word length of the input data to the second CIC decimator is $(B_{m1} + 1)$. Therefore, the maximum word length at the second CIC decimator output is $(B_{m2} + 1)$, where

$$B_{m2} = \lceil N \log 2 (R_2) \rceil + B_{m1} . \tag{4.21}$$

Because $R_2 = R / R_1$, the above expression can be written as,

$$B_{m2} = \lceil N log_2 R \rceil + B_{in} - (\lceil N log_2 R_1 \rceil - \lceil N_1 log_2 R_1 \rceil) - k_s . \tag{4.22}$$

Note from (4.22) that the final maximum word length for a polyphase CIC decimator is shorter than that of Hogenauer's by $(\lceil N log_2 R_1 \rceil - \lceil N_1 log_2 R_1 \rceil) + k_s$ bits. This is due to both the two-stage architecture and the new scheme for implementing polyphase components. For a typical case where $N_1 = 2$, $N = 3$, $R_1 = 4$ and $k_s = 2$~3, we save 4~5 bits.

B. Truncation and Rounding

Assume that the required output word length is $B_{out} + 1$. Since the polyphase CIC decimator consists of two cascaded CIC decimators, the word length at the output of the first polyphase filter can be limited to $B_{out} + 1$ bits without losing accuracy. The number of lower bits discarded at the output of the first polyphase filter is,

$$\begin{cases} B_{m1} - B_{out}, & B_{m1} > B_{out} \\ 0. & B_{m1} \le B_{out} \end{cases}$$

After truncation or rounding, the word length of the input data to the second CIC decimator becomes $B_{in2} + 1$, where $B_{in2} = MIN(B_{m1}, B_{out})$. In general, $B_{out} \le B_{m2}$. The number of lower bits discarded at the output of the second polyphase filter is,

$$\begin{cases} B_{m2} - B_{out}, & B_{m2} > B_{out} \\ 0. & B_{m2} \le B_{out} \end{cases}$$

In a CIC decimator, truncation or rounding may be used at each filter stage to reduce register widths significantly. For truncation and rounding in each filter stage of the second CIC decimator, see [Hoge81] for detail.

In a multi-stage polyphase interpolator, assume that the required output word length is $B_{out} + 1$. Truncation or rounding may be applied after the first CIC interpolator. The number of bits that can be discarded is,

$$B_{in} + \lceil N log_2 R_2 \rceil - B_{out}.$$

4.2.4 Design Procedure Summary

The design procedure for the multi-stage polyphase CIC decimators shown in Figure 4.1 can be summarized as follows:

1- Partition the downsampling ratio between R_1 and R_2 for a known downsampling ratio R such that $R = R_1 R_2$. The choice of R_1 (typically 4) is up to the designer. The considerations may be the complexity of polyphase components and the operation speed of the circuits.

2- Determine N_1 to meet the required SNR. The considerations include the excess noise (see eq.(4.14)) and interference (see Figure 4.2). The trade-off is the complexity of polyphase components and aliasing attenuation. For detail, see Section 4.2.1.

3- Obtain polyphase components from Table 4.2 or derive them from equation (4.4). Synthesize them using the proposed scheme shown in Section 4.2.2.

4- Determine N for the second CIC decimator. It usually equals one plus the order of the $\Delta\Sigma$ modulator. It may increase for a higher interferer. Refer to Section 4.2.1.

5- Budget the word-length in each stage according to Section 4.2.3.

4.3 Decimation for Two Delta-Sigma Modulators

Two typical applications of the proposed decimation for ΔΣ modulators are considered here. One demonstrates low power consumption for a double-sampled ΔΣ modulator, and the other demonstrates the possibility of decimation for GHz-rate ΔΣ modulators.

A. Decimation for Double-Sampled $\Delta\Sigma$ Modulators

A novel double sampling ΔΣ modulator which can be used in baseband digitization receivers was described in Chapter 3. The technique can increase the effective oversampling ratio by a factor of 2. This technique can be used to either increase the OSR or to reduce the power consumption. To achieve these, it makes sense to operate the subsequent decimator at the clock rate rather than at the effective sampling rate (twice the clock rate). The outputs of a double sampling ΔΣ modulator are two-phase signals at the clock rate f_c. With the polyphase CIC decimator, the two-phase signals can be directly processed instead of as one interleaved double-speed signal at $2f_c$. This is shown in Figure 4.8 with $R_1 = 2$.

For $N_1 = 4$ and 5 cases, the polyphase components are $\{F_0$: (1, 6, 1), F_1: (4, 4, 0)$\}$ and $\{F_0$: (1, 10, 5), F_1: (5, 10, 1)$\}$ respectively. They are simple. The tone rejection can be made sufficient by increasing N_1 as seen from Figure 4.2.

The advantages are obvious: (1) lower power consumption or (2) relaxation of circuit timing requirements.

4.4 FPGA Implementation of a DDC at 100 MHz

In this section, a DDC at 100 MHz consisting of the multi-stage polyphase CIC decimator is implemented with an FPGA. The objectives are:

- to provide a concrete example of the design techniques proposed.
- to demonstrate the proposed solution for high speed operation with an FPGA.
- to compare power consumption between the multi-stage polyphase CIC and conventional multi-stage CIC decimators.

4.4.1 System and Circuit Design

A. Architecture

A DDC employing a polyphase CIC decimator is designed to downconvert IF signals coming out of a fourth-order bandpass ΔΣ modulator, as depicted in Figure 4.9(a). The analog input to the bandpass ΔΣ modulator may be a QPSK IF signal with a center carrier frequency f_{IF} = 25 MHz. The sampling rate is 100 MHz and the symbol rate f_{sy} = 195.3125 KHz. The bandwidth of this fourth-order bandpass ΔΣ modulator is $2f_{sy}$. The OSR is 128 which is defined as the ratio of the sampling rate to twice the symbol rate. The DDC consists of one mixer and two decimators as shown in the dashed box of Figure 4.9 (b). Two DDCs are needed for quadrature demodulation. Following the DDCs are halfband decimators to further downsample to twice the symbol rate. The DDC operates at 100 MHz. With a Xilinx 3000 series FPGA [XILINX], it is impossible to downsample the 100-MHz data signal using a straightforward CIC decimator. The proposed multi-stage polyphase CIC decimator is implemented here.

The detailed decimation procedure to downconvert the 100 MHz IF signal to baseband is shown in Figure 4.9 (b). Since the sampling rate is four times the IF frequency, the oversampled signal is translated to baseband by mixing with a simple sequence {1,0,-1,0,...} in the I channel and {0,1,0,-1,...} in the Q channel. The digital signal is downsampled to a sampling rate of 1.5625 MHz by a multi-stage polyphase CIC decimator. The final sampling rate of 390.625 KHz is twice the symbol rate f_{sy} = 195.3125 KHz.

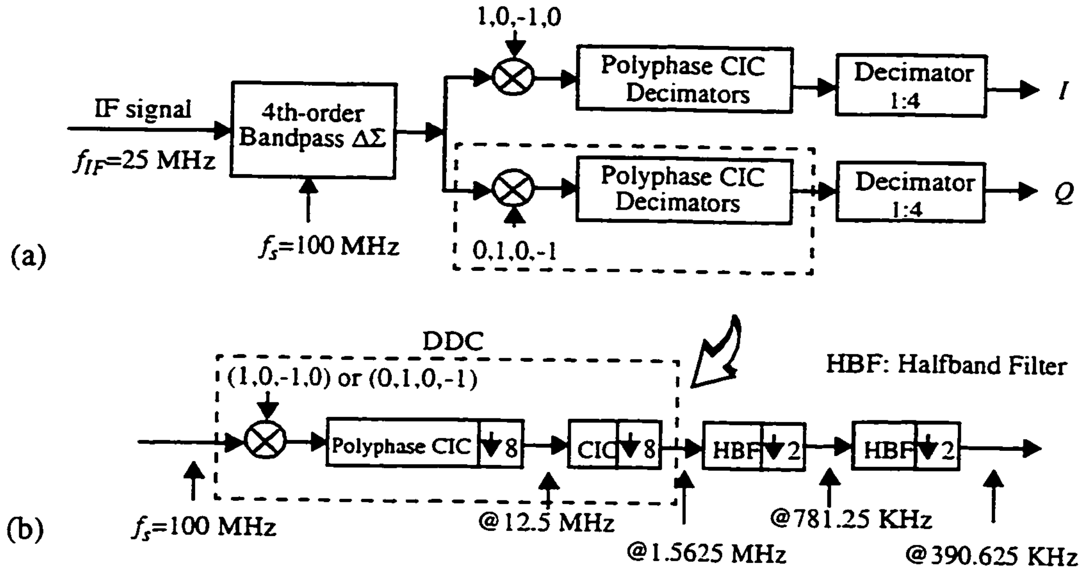


Figure 4.9 A digital IF narrowband quadrature demodulation

B. Design Considerations

Since this FPGA cannot directly handle 100 MHz, a multi-stage polyphase CIC decimator is used where $R_1 = 8$ is chosen to reduce the processing rate to 12.5 MHz. Since $R = 64$, $R_2 = 8$. The DDC block diagram for I and Q is shown in Figure 4.10. Note that the sequences {1,0,-1,0,...} and {0,1,0,-1...} are incorporated into polyphase components. The final halfband lowpass decimators are not implemented.

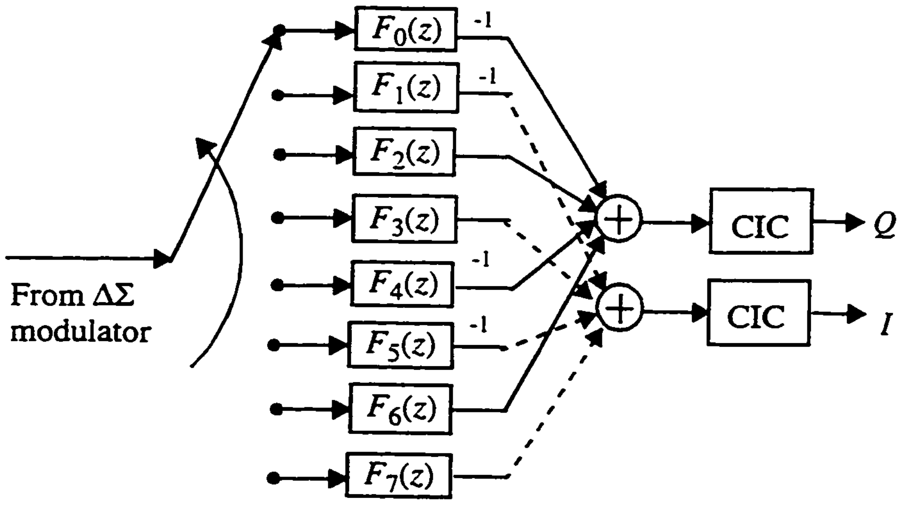


Figure 4.10 An DDC using a multi-stage polyphase CIC decimator

According to [Cand86], $N = 3$ is chosen in the second CIC decimator which makes the aliasing noise contribution negligible. To simplify the polyphase components (and save more power), we choose $N_1 = 2$. From Section 4.2.1, the SNR loss is 6 dB with respect to the ideal SNR achievable by a fourth-order bandpass ΔΣ modulator. Since the OSR = 128, the ideal achievable SNR is about 86 dB. At the output of this DDC, the SNR in the desired band is around 80 dB. This is suitable for a ΔΣ modulator which is thermal noise limited to < 80 dB.

Since the signal at the output of the first polyphase CIC decimator is lowpass, the definition of IOSR is the ratio of the first downsampling rate f_s/R_1 to twice the symbol rate $2f_{sy}$. IOSR is $2\mathrm{OSR}/R_1$ for the bandpass case since $\mathrm{OSR} = f_s/(4f_{sy})$. Hence IOSR = 32. As can be noted from Figure 4.2, the aliasing attenuation is slightly above 70 dB. The droop due to the first polyphase filter is around -0.01 dB as can be seen in Figure 4.4.

C. Circuit Designs

The polyphase components in Figure 4.10 can be obtained from Table 4.2. When a ΔΣ modulated signal is single-bit (either "1" or "0" in logic), a simple logic circuit can be used to implement polyphase component as described in Section 4.2.2.

The polyphase components combined with the mixing signal (1 or -1) for both quadrature and in-phase channels are as follows:

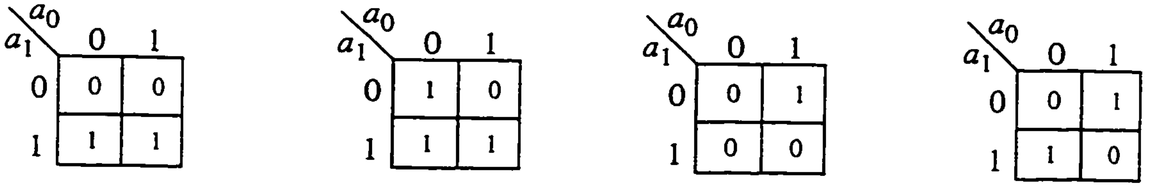
Q: $\{-F_0(z) = -1 - 7z^{-1}, F_2(z) = 3 + 5z^{-1}, -F_4(z) = -5 - 3z^{-1}, F_6(z) = 7 + z^{-1}\}$.
I: $\{-F_1(z) = -2 - 6z^{-1}, F_3(z) = 4 + 4z^{-1}, -F_5(z) = -6 - 2z^{-1}, F_7(z) = 8\}$.

The input signals to each adder in each polyphase component are a_0 and a_1, where a_0 is the current input and a_1 is the delayed input. The output is $b_0 b_1 \ldots b_i$, $i = 3$ or 4, where b_0 is the MSB. For a sample polyphase component $-F_0(z) = c_0 + c_1 z^{-1}$, where $c_0 = -1$ and $c_1 = -7$, The truth table in Table 4.3 can be obtained.

Note that $-F'_0(z)$ is the intermediate result and $-F_0(z)$ is the scaled result. As stated earlier, the scaling is used to further reduce the bit-width by 1 bit. In the first row, since the input is 00 (0 representing -1), the result is 8. So are the results for other cases. After scaling by 0.5, we obtain the third column. The output in two's complement is obtained in the fourth column. From Table 4.3, the Karnaugh maps can be derived for b_0, b_1, b_2 and b_3, which are shown in Figure 4.11. The synthesized circuit is depicted in Figure 4.12.

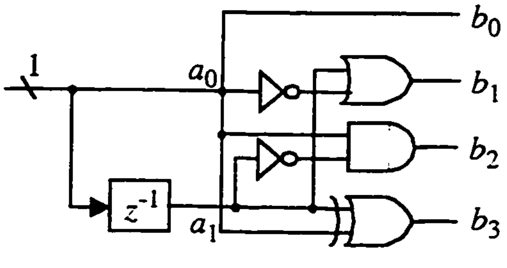
Table 4.3 Truth table for $-F_0$(z)

a_0a_1	$-F_0(z)$	$-F_0(z)$	$-F_0(z)$ in 2's-Complement $b_0b_1b_2b_3$
00	8	4	0100
01	-6	-3	1101
10	6	3	0011
11	-8	-4	1100



$b_0 = a_1$ $b_1 = \bar{a}_0 + a_1$ $b_2 = a_0\bar{a}_1$ $b_3 = \bar{a}_0 a_1 + a_0\bar{a}_1$

Figure 4.11 Karnaugh maps for $-F_0(z)$



(a) $-F_0(z)$

Figure 4.12 Circuit of polyphase component

For polyphase components $-F_i(z)$, $i = 1, 4, 5$, similar results can be obtained. The synthesized circuits are shown in Appendix B. As can be seen, the circuits for polyphase components are quite simple for a one-bit ΔΣ modulated signal, requiring only 36 logic gates in total. Note that 4-bit output is required for $-F_0(z)$, $F_2(z)$, $-F_4(z)$ and $F_6(z)$; and 3-bit for $-F_1(z)$, $F_3(z)$, $-F_5(z)$ and $F_7(z)$. Therefore, outputs I and Q require 6 and 5 bits respectively. Using a conventional method to implement the first decimator, the outputs (I and Q) require 8 bits. Hence, we save 2~3 bits.

After adding two groups of polyphase components shown in Figure 4.10, the word-lengths for the I and Q paths at the outputs of polyphase decimators are 5 and 6 bits respectively.The minimum word-length in the integrators of the second CIC decimator without incurring any distortion is 15 bits[1]. For a 80 dB performance, a 14-bit accumulator (ACC) is sufficient. The 14-bit ACC works at 12.5 MHz and hence a pipelined structure is used. The detailed circuits are shown in Appendix B.

D. FPGA Chip

The digital downconversion circuits are implemented on a Xilinx XC3159A [XILINX]. The XC3100A is a performance-optimized relative of the XC3000 and XC3000A families. Some features are:

- 50 ~ 80 MHz system clock rates,
- 190 to 325 MHz guaranteed flip-flop toggle rates, and
- 1.75 to 4.1 ns logic delays.

The FPGA schematic diagram is shown in Appendix B.2. The circuit block diagram is shown in Figure 4.13, where a multiplexer is used to swap between *I* and *Q* channels, controlled by signal *SELECT*. This arrangement is made so as to share a 3-stage CIC decimator between the *I* and *Q* channels and makes a one-chip solution possible. D flip-flops before the multiplexer are used to pipeline the results from two consecutive additions. All circuits up to the accumulator ACC are clocked by an input clock *CLK* of 12.5 MHz.

[1] $\log 2(R_2^{\,N}) + B_{in} + 1 = \log 2(8^3) + 6 + 1 = 15$.

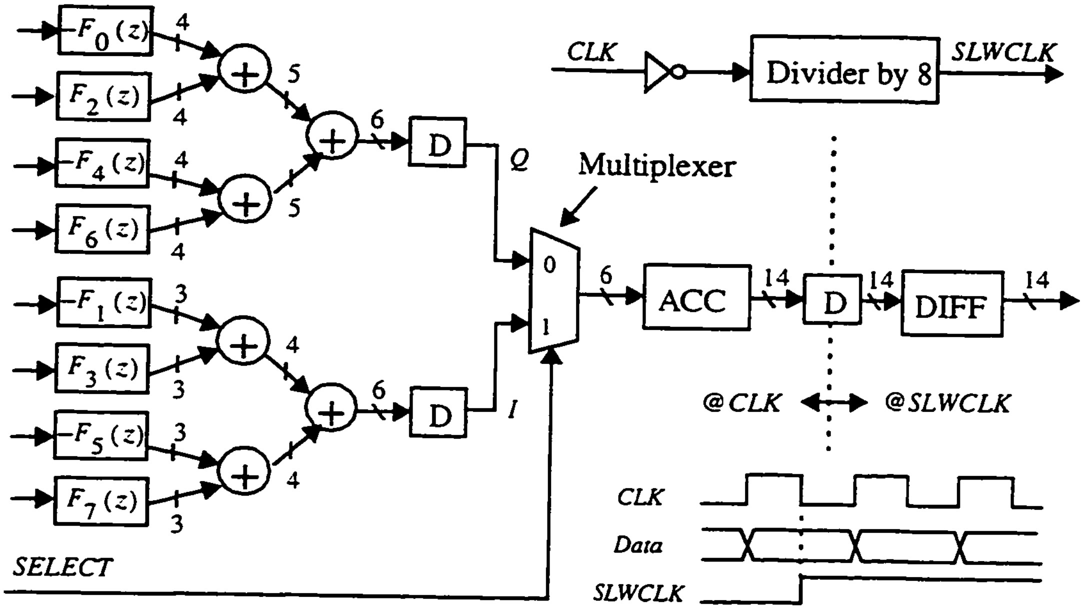


Figure 4.13 FPGA architecture for a polyphase DDC

E. Power Consumption Comparison

For comparison, a conventional DDC architecture is shown in Figure 4.14. In this architecture, two cascaded CIC decimators are used. The first is a two-stage CIC decimator with a decimation ratio of 8 and the second is a three-stage with a decimation ratio of 8. The transfer function is exactly the same as the polyphase one. However, this multi-stage implementation needs two more bits than the proposed scheme (see Figure 4.13) and two 8-bit accumulators operating at the full input rate (100 MHz).

The number of gates used in the FPGA implementation is shown in Table 4.4. The estimated power dissipation is also listed in the table. The estimation is based on [Baza97] for 3.3-V power supply in a 0.5 µm CMOS technology and 50% switching activity. The gate counts and power dissipation estimation in Figure 4.14 are also listed in Table 4.4. One can see that the total gate count in the polyphase architecture is two thirds of that in the conventional architecture. The power dissipation in the polyphase architecture is about one fifth of that in the conventional architecture. Note that pipelined architectures are used in the first CIC decimator and the ACC of the second CIC decimator.

These power savings are obtained even without using lower power supplies on the low-speed circuits, a technique which would further exploit the reductions in clock rate to save power [Chan95]. If the device delay is considered, the polyphase architecture is advantageous since the main circuits operate at a much lower rate (12.5 MHz). The only circuits operating at 100 MHz are D flip-flops implementing the commutator. The conventional architecture needs two 8-bit 100 MHz accumulators. The polyphase architecture allow a lower power supply. If this factor is counted, the polyphase architecture can achieve even lower power dissipation than that listed in the table.

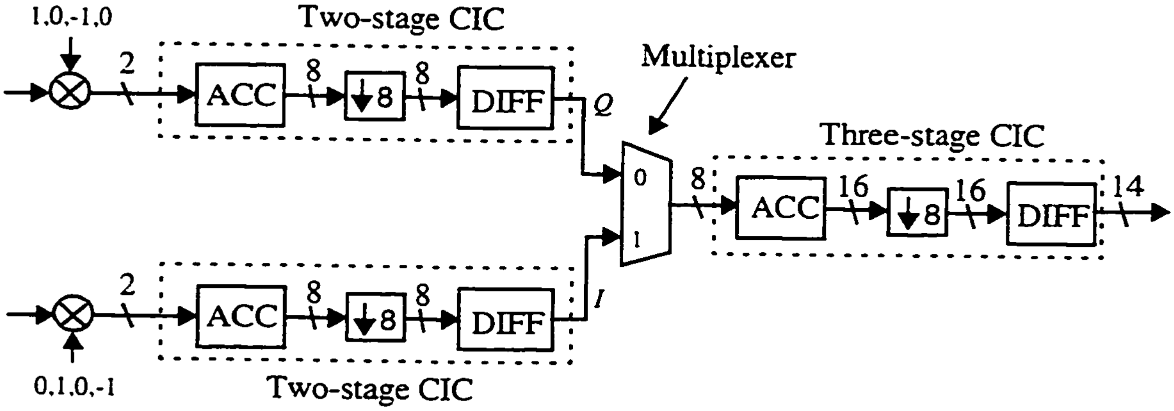


Figure 4.14 A conventional DDC architecture

Table 4.4 Gate counts and power estimation in DDCs

Operation frequency (MHz)	Polyphase DDC architecture		Conventional DDC architecture	
	#'s of gates	Estimated power dissipation (mW)	#'s of gates	Estimated power dissipation (mW)
100	8	0.4	316	15.8
12.5	636	4	852	5.3
1.5625	308	0.2	352	0.3
Total	952	4.6	1520	21.4

4.4.2 Test Results

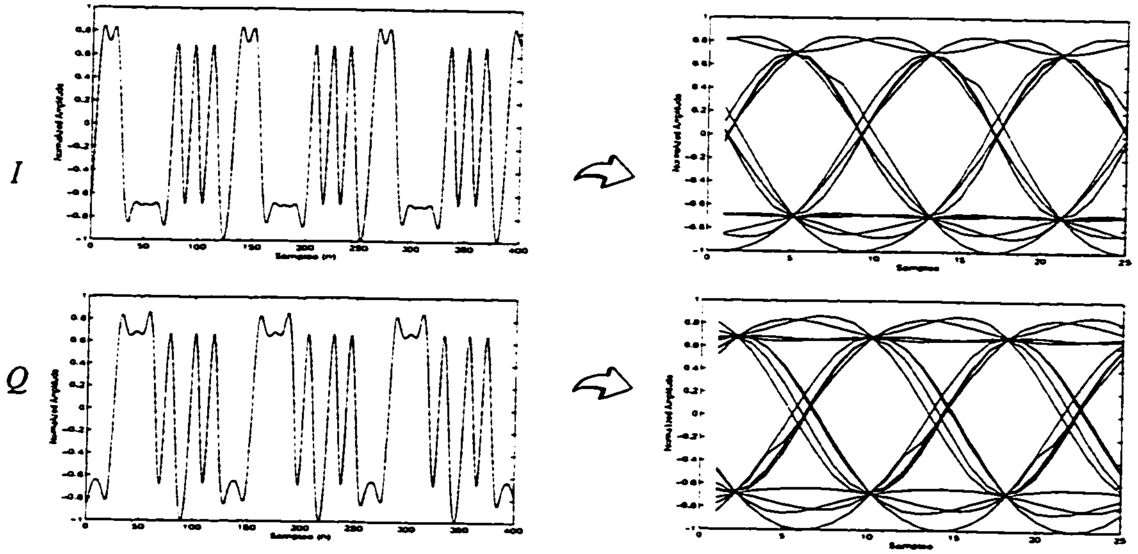
The chip has been tested. The test setup is described in Appendix B.2. Two results were measured and they are reported as follows:

SNR Test

The data is generated using a fourth-order bandpass ΔΣ modulator simulated in SPW. A tone signal at $f_s/4$ is used to test the SNR and hence the data is a repeated sequence {1, 0, -1, 0,..}. The tone level is 6 dB down from full scale and the calculated SNR of the bandpass modulated data in the desired band is 86.4 dB.

The data at the output of this chip is captured and analyzed. The measured SNR in the desired band is 80.1 dB which implies 6.3 dB degradation compared to 6 dB from analysis in 4.2.1. Since we did not implement two half-band filters and the data is taken from the output of the multi-stage polyphase, the SNR in the whole band is measured and it is 56 dB.

Eye Diagram



(a) Waveforms

(b) Eye diagrams

Figure 4.15 The measured *I* and *Q* signals at the outputs of the DDC chip: (a) the *I* and *Q* waveforms, and (b) their eye diagrams

The measured eye diagrams are plotted in Figure 4.15 (a) and (b), respectively. Note that they are normalized to the maximum samples. Since the output rate of the DDC is eight times the symbol rate, the time resolution is good enough to do the timing phase recovery.

One can see from Figure 4.15(b), the "eye opening" defined as the distance from the threshold (which is zero) to the closest trace at the sampling instant is at its widest. This is because the SNR is quite high (56 dB).

4.5 Summary

This chapter not only has demonstrated a solution to decimation for low power and GHz-rate $\Delta\Sigma$ modulators, but, more importantly, provided a design method for multi-stage polyphase CIC decimators in these applications.

Designing decimators for low power and GHz-rate $\Delta\Sigma$ modulators is difficult since the full-rate cascaded integrators typically have 16~24 bits. We have provided a solution to this problem by combining multi-stage decimators and polyphase techniques.

We have shown how to simplify design by reducing the number of stages in the first polyphase CIC decimator from $(M + 1)$ to M, where M is the $\Delta\Sigma$ modulator order. This saves $\log_2(R_1)$ bits, where R_1 is the first downsampling ratio. In a high speed $\Delta\Sigma$ modulator, its achievable SNR may be limited by thermal noise, clock jitter, etc. The SNR loss is insignificant with respect to the actual SNR achievable by a high speed modulator. We have also shown how to save word-length further by 2~3 bits by a new design scheme for polyphase components, and how to determine the required word-length in each stage.

We have also shown two important special cases. The multi-stage polyphase CIC decimators can be used with double-sampled $\Delta\Sigma$ modulators to achieve low power in baseband digitization receivers. They also can be used for low power or high speed applications in IF digitization receivers. The polyphase architecture makes GHz-rate decimation practical by reducing the peak rate at which adders need to clock and the word-length of the fast accumulators.

We demonstrated the proposed design principle by implementing a DDC at 100 MHz in an FPGA for high speed operation. We have shown a power reduction by a factor of 4~5 compared to a conventional multi-stage CIC decimator. The measured output SNR is 56 dB (80.1 dB in the desired band) and the eye opening has a negligible loss.

These power savings are obtained even without using lower power supplies on the low-speed circuits, a technique which would further exploit the reductions in clock rate to save power [Chan95]. Lowering the power supply, the speed of circuits are reduced. Only multi-stage polyphase CIC decimators provide this trade-off.

proposed in Section 5.3 to eliminate the spurious transient signals. Performance analyses including jitter and SNR bound introduced by interferer mixing with jitter are provided in Section 5.4. Simulation and experiment results are given in Section 5.5. An FPGA implementation of the proposed symbol timing recovery for a BPSK modulated signal is described in Section 5.6. Summary of this chapter is provided in Section 5.7.

This chapter provides the following new contributions:

- We propose a scheme to move the re-timing function into a modified decimator in an oversampled receiver.
- We show that simply shifting this clock phase is not a solution since it produces large "glitches" at the output. The glitches settle out after N samples, where N is the stages in CIC decimator. We eliminate them by using a proposed dual-differentiator timing phase adjustable CIC decimator.
- We have shown a good fit between theory and simulation. We show that the performance of this technique is determined by the transmitter and receiver clock rate difference α and OSR: (a) for a small $\alpha < \frac{1}{7OSR}$, the rms timing jitter is dominated by OSR and its slope is -3 dB per octave of OSR. The slope of SNR bound is -3 dB per octave of α or 3 dB per octave of OSR. (b) for a large $\alpha > \frac{4}{7OSR}$, the rms timing jitter and SNR bound are dominated by the frequency difference α and their slopes are 3 dB and 6 dB per octave of α respectively.
- One experiment demonstrates that the method works in the presence of alternate interferer and in conjunction with carrier recovery. The SNR bound is within 1.5 dB of consistency with the estimate.
- An FPGA chip for symbol timing recovery is designed and implemented. The stability of the technique is proved. The eye opening has been measured and it contributes to an Eb/No degradation bound of about 0.75 dB.

5.1 Principle of Timing Adjustment by Decimating Oversampled Signals

In an interpolation method, an interpolator is used to estimate samples between existing samples and it is implemented by a fractional delay filter [Laak96]. This method is suit-

Another way is to use an adjustable-delay decimator with a fixed re-sampling clock. In this method, the signal is re-sampled at a fixed re-sampling clock and the re-sampling phase is adjusted by changing the signal path delay. The signal path delay is adjusted instead of the re-sampling clock directly being changed.

In both methods, the finest timing phase which can be adjusted is one sampling period $T = 1/f_s$ and the maximum delay needed to be adjusted is one symbol period. The first method of adjusting re-sampling clock is considered here and a preliminary study of the second is given in Appendix D.The model decimator used to study the timing adjustment is as follows: The $\Delta\Sigma$ modulator is followed by decimation. The decimation can be split into three stages [Cand86]. A modified CIC decimator is used to re-sample the signal and also downconvert the incoming sampling rate to the first clock rate f_1 which is eight times the symbol rate f_{sy}. Following the decimator is a halfband decimator [Good77] which further downconverts the clock rate by 2 to a clock rate f_2. Then a data filter may be used to shape the received signal pulse in order to satisfy the ISI-free condition [Skla88] and also downconvert the clock rate to a clock rate f_3. The final clock rate f_3 is twice the symbol rate and may be used in subsequent fractionally-spaced equalizers [Lee90].

For signals with excess bandwidth factor β, the sampling rate must exceed $(1 + \beta)/T_{sy}$. A sampling rate of $2/T_{sy}$ is theoretically sufficient for any signal with $\beta < 1$. OSRs for lowpass and bandpass cases are defined as follows,

$$T = \begin{cases} \dfrac{T_{sy}}{2 \cdot OSR}, & \text{for a lowpass } \Delta\Sigma \text{ data converter} \\ \dfrac{T_{sy}}{4 \cdot OSR}, & \text{for a bandpass } \Delta\Sigma \text{ data converter} \end{cases} .$$

Note that the relation between the clock period T and symbol rate T_{sy} is different for these two modulators. To unify the definition, we use

$$T = \frac{T_{sy}}{2 \cdot OSR} \tag{5.8}$$

in the discussion throughout this thesis.Therefore, for a given SNR, the bandpass can operate at half the OSR. Note that the finest timing phase adjustment is better than 1% of

denoted as $R(m)$, where $m = 0, 1, 2,...$ denotes the m^{th} sample at the lower rate f_1 as shown in Figure 5.2. The re-sampling process with a variable downconversion factor R is non-uniform re-sampling. The timing diagram for such a process is depicted in Figure 5.3. Assuming f_s is a constant (or a slowly time-varying variable which can be treated as a constant for several consecutive symbol intervals), one can establish the relation between the re-sampling rate $f_1(m)$ and the incoming sampling rate f_s as $f_1(m) = f_s/R(m)$. This relation can be rewritten in terms of their respective sampling intervals as,

$$T_1(m) = R(m)T, \tag{5.9}$$

where $T_1(m) = 1/f_1(m)$ is the re-sampling interval. Hence the timing phase can be advanced or retarded by adjusting $R(m)$ and the finest timing phase that can be adjusted is T. Note that $R(m)$ is time-varying with an average value $\overline{R}$ that tracks drift between local and transmitted clocks. Equation (5.9) should be seen as an instantaneous relation for these two rates.

A. A Timing Recovery Loop

Figure 5.4 shows a block diagram for timing recovery loop that incorporates the above technique. Note that only one path is shown for the sake of simplicity and the diagram can easily be extended to the quadrature case. Also shown in the figure are the clock rate relationships among all blocks, which are in agreement with those discussed in Section 5.1.

Note that the timing recovery loop shown in the figure is basically a Phase-Locked Loop (PLL). It represents one of many architectures and is used only to illustrate the proposed decimation technique for timing recovery.

The principle of the operation is as follows. The timing error detector is a timing phase detector which calculates the timing phase error. The halfband and data filters are low pass filtering. The controller is a NCO which is used to re-sample the oversampled input signal. The timing-phase adjustable decimator is a re-sampler since its input is highly oversampled and its output operates at a much lower rate. This re-sampler behaves much like an "ADC" with a controlled sampling clock in the mixed analog and digital re-timing method shown in Figure 2.12(b). Thus, the proposed timing recovery loop resembles that in Figure 2.12(b). However, this "ADC" accepts a highly oversampled digital signal. The difference is that the ADC in Figure 2.12(b) is replaced by an adjustable-timing-phase decimator. Hence, the proposed method is all-digital.

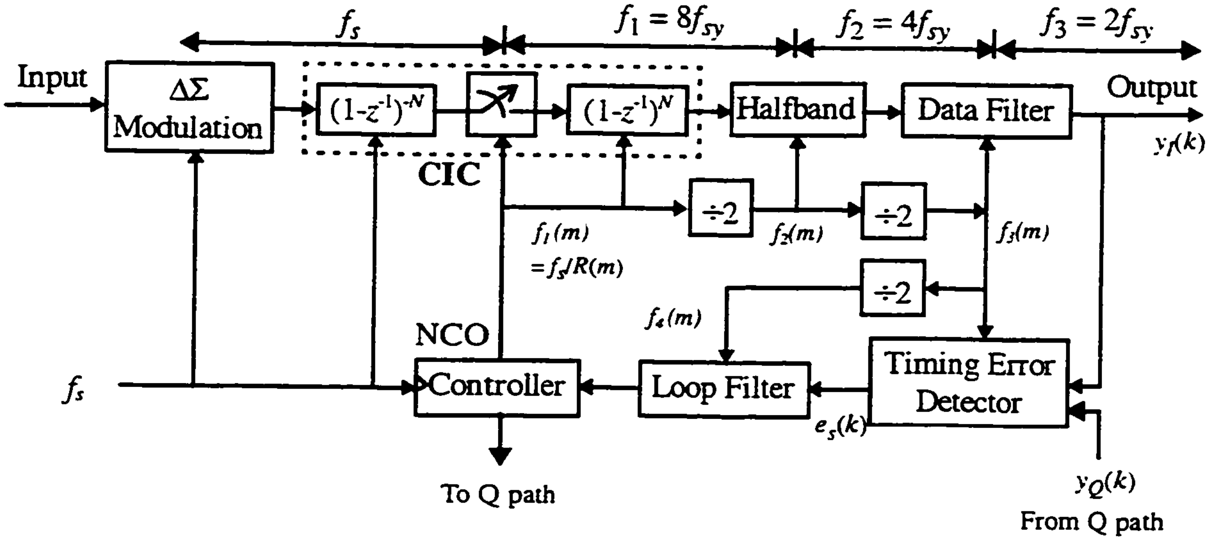


Figure 5.4 Block diagram of symbol timing recovery using an adjustable-timing-phase CIC decimator

Note that the proposed method is independent of the timing error detection algorithm. There are several timing error detection algorithms. Two representatives are the one-sample-per-symbol [Muel76] and the two-sample-per-symbol [Gard86] algorithms [Cowl94]. Here, we use the second one. The timing error signal $e_s(k)$ at instant k is written as,

$$e_s(k) = y_I\left(k-\frac{1}{2}\right)(y_I(k) - y_I(k-1)) + y_Q\left(k-\frac{1}{2}\right)(y_Q(k) - y_Q(k-1)) , \quad (5.10)$$

where y_I and y_Q are the outputs of the I and Q channels, respectively. Note that y_I and y_Q are clocked at twice the symbol rate and $e_s(k)$ is at the symbol rate. This algorithm is also suitable for M-ary modulation including BPSK. The algorithm used here is good for detecting data signals with alternating 0's and 1's. If a proper loop filter is used for this algorithm, it also works well with random data signals.

The loop filter works at the symbol rate and outputs a smoothed timing error in each symbol interval. The controller, clocked at the oversampling rate f_s, adjusts the timing phase by changing the downconversion factor R. This may be implemented by controlling a variable counter.

Assume that the average downconversion factor $\bar{R}$ is close to an integer constant R_0. There are basically two ways to adjust the timing phase:

1) One-sampling-interval adjustment: The method is to advance or retard one sampling interval T_s at one time. This can be realized by reducing or increasing the downconversion factor by 1. Therefore a variable divider with ratios counter $(R_0 - 1)/R_0/(R_0 + 1)$ can be used, driven by a very simple controller.

2) Multi-sampling-interval adjustment: The controller calculates exactly how many sampling intervals it needs to advance or retard in order to minimize the timing error. This may require changing the downconversion factor by a step greater than 1, that is, $R(m) = R_0 + \delta$, where δ is an integer.

Note that T_{sy} and T are determined separately by the transmitter and local clocks, respectively. The goal of using an adjustable-timing-phase CIC decimator is to track drift between local and transmitter clocks. Therefore $R(m)$ should keep being adjusted to track the drift with an average value equal to,

$$\bar{R} = \frac{T_{sy}}{8T}. \tag{5.11}$$

There are eight samples at the output of CIC decimator within one symbol. Therefore, one simple way to adjust the timing phase is shown in Figure 5.5, where the timing phase adjustment occurs at the last sample of the eight CIC output samples in a symbol. Note that the average f_4 is equal to the symbol rate. In other word, the timing phase of f_1 is adjusted just before a new symbol sample is created. In this way we can simplify the controller by avoiding multiple adjustments in one symbol interval. The timing phase adjustment is δT in one symbol interval, where the integer δ is defined as,

$$\delta = \begin{cases} > 0, & \text{advance} \\ < 0, & \text{retard} \\ = 0, & \text{retain} \end{cases} \qquad (5.12)$$

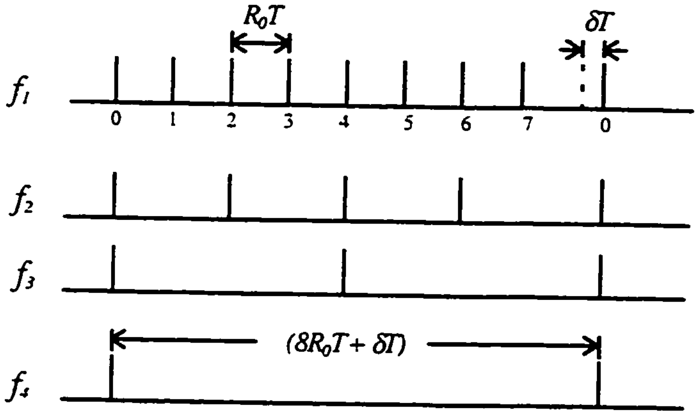


Figure 5.5 Timing diagram in the timing phase adjustment

B. Delays in the Timing Recovery Loop

The timing recovery loop in Figure 5.4 is a feedback loop, and delays in the loop affect its stability [Lee90]. The total loop delay is the sum of delays in the loop components. These components include N cascaded differentiators in the CIC decimator, the halfband filter, data filter and those introduced by the timing error detector, loop filter and controller. Since the clock in the loop is changed slightly due to timing phase adjustment, the loop delay varies slightly. Since the feedback delays introduced by the timing error detector, loop filter and controller are dependent on the algorithms, the average total forward delay is considered and can be derived below. The worst case is very similar.

The transfer function of the cascaded differentiators is $(1-z^{-1})^N$, of which the average delay is $E[NT_1/2]$, where T_1 is a variable. Assume that the halfband and data filters are FIR linear filters with N_h and N_d taps, respectively. The average delays introduced by these two filters are $E[(N_h-1)T_1/2]$ and $E[(N_d-1)T_2/2]$, respectively. Using $E[T_1] = \frac{1}{8}T_{sy}$, we can approximately obtain the average total forward delay as,

$$D_f \equiv (N_h + 2N_d)\,T_{sy}/16. \tag{5.13}$$

The dominant delays will be the halfband and data filter since they operate at relatively low rates. In practice, the first term in (5.13) introduces about 1 symbol interval delay, and the second has a delay of 2 – 3 symbol intervals. The total forward delay would therefore be around 3 – 4 symbol intervals.

C. Design of Timing Recovery Loop

Two parameters important for the loop design are the sensitivities of the error detector and the NCO. A variable counter often implements the NCO.

For tracking analysis, the error detector sensitivity is the slope of the detector around zero. It depends on the timing error detection algorithm. The sensitivity can be obtained by analyzing the s-curves of the error detection algorithms. The sensitivities for that in (5.10) are different for a 101010... data and a random input data [Erup93].

Changing the input to the NCO by δ, the timing phase is adjusted by δT for every symbol. The re-sampling clock is changed by $T_{sy}/(2\mathrm{OSR})$ and hence the sensitivity of the NCO is 1/(2OSR).

The loop can be designed and analyzed using classical PLL theory [Gard79]. The loop filter is often a first-order lead-lag filter. Two coefficients of this filter are determined by the required loop bandwidth and the damping factor. The loop bandwidth is determined by the natural frequency and damping factor. The impact of loop delay on the stability has been discussed in [Lee90]. In general, the loop delay should be much less than the inverse of the transmitter and receiver clock difference. Detailed considerations are outside the scope of the thesis and please see [Gard79].

5.3 Practical Adjustable-Timing-Phase Decimators for Timing Recovery

A severe problem occurs by simply shifting the re-sampling clock in the CIC decimator. After timing adjustment, the N registers in the cascaded differentiators of Figure 5.2 need NT_1 periods to "forget" the values with the old timing phase. During these periods, the CIC decimator goes to a transient state where "glitches" (or spurious transient signals) appear at the CIC decimator output. These glitches can be seen in the simulation result shown in Figure 5.6. They are harmful to timing error detection.

A simple explanation for glitches is that we are making a timing change at the input to an N^{th}-order differentiator (in Figure 5.2), which is very sensitive to small timing changes at the output of cascaded lossless integrators.

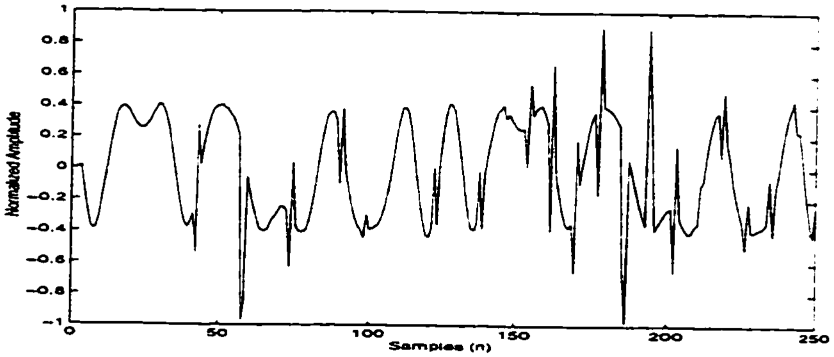


Figure 5.6 Spurious transient signal created by clock adjustment

5.3.1 Spurious Transient Signals Created by Timing Phase Adjustment

To better understand the spurious transient signals, we conduct the following analysis. We show that the glitches last for N samples at rate f_1 whenever there is a timing phase adjustment, where N is the numbers of stages in the CIC decimator.

In Figure 5.2, the output $y(t_m)$ of the N-stage CIC decimator at time instant t_m can be written as,

$$y(t_m) = \sum_{k=0}^{N} c_k y_N(t_{m-k}), \tag{5.14}$$

where index m denotes the m th sample of the CIC decimator output at time instant t_m, c_k's are the coefficients in the impulse response of the N-cascaded differentiators and $y_N(t_{m-k})$ is the input sample of the first differentiator at time instant t_{m-k}. For the case where the timing phase is adjusted by δT at time instant t_{m0} as shown in Figure 5.7, we have $R(m0) = R_0 + \delta$. Therefore,

$$y(t_{m_0+l}) = \begin{cases} \sum_{k=0}^{N} c_k x_N\left(t_{m_0-1} + (l-k)R_0T\right), & l < 0 \\ \sum_{k=0}^{l} c_k x_N\left(t_{m_0-1} + (l-k)R_0T_s + \delta T\right) & \\ + \sum_{k=l+1}^{N} c_k x_N\left(t_{m_0-1} + (l-k)R_0T\right), & 0 \le l < N \\ \sum_{k=0}^{N} c_k x_N\left(t_{m_0-1} + (l-k)R_0T + \delta T\right), & l \ge N \end{cases} \tag{5.19}$$

Note that the CIC decimator outputs $y(t_{m_0+l})$, $0 \le l < N$, calculated from (5.19) contains spurious transient samples. As can be seen from (5.19), the samples before and after the timing adjustment coexist in the differentiators for $0 \le l < N$. This is due to the N-sample propagation delay in the N cascaded differentiators.

The ideal output sample for $0 \le l < N$ should be,

$$y'(t_{m_0+l}) = \sum_{k=0}^{N} c_k x_N(t_{m_0-1} + (l-k)T + \delta T), 0 \le l < N. \tag{5.20}$$

Therefore, spurious transient error signals can be obtained by substracting (5.19) from (5.20). These transient error signals are $\Delta y(t_{m_0+l}) = y(t_{m_0+l}) - y'(t_{m_0+l})$, $0 \le l < N$, and they can be expressed as,

$$\Delta y(t_{m_0+l}) = \sum_{k=l+1}^{N} c_k \Delta x_N(t_{m_0-1} + (l-k)R_oT), 0 \le l < N, \tag{5.21}$$

where

$$\Delta x_N(t) = x_N(t) - x_N(t + \delta T) . \tag{5.22}$$

Two observations can be obtained from (5.21) and (5.22):

(1) The error introduced by nonuniform re-sampling spans N samples (clocked at f_1) starting from the beginning of the timing phase adjustment.

(2) For a lowpass FIR filter, the error given in (5.21) is small since the difference

tor. Waveforms at the outputs of channel 0, channel 1 and the multiplexer are shown in Figure 5.10, together with the multiplexer control signal *Ctrl*. From the figure, one can see the spurious transient signals (in channel 0 and channel 1) introduced by the timing adjustment and the 2-sample span for the spurious signals. The spurious transient signals are removed and a very clean signal is obtained at the output of the multiplexer.

Re-timing causes phase jitter, which can mix large interferers in-band. We will analyze this in the following section.

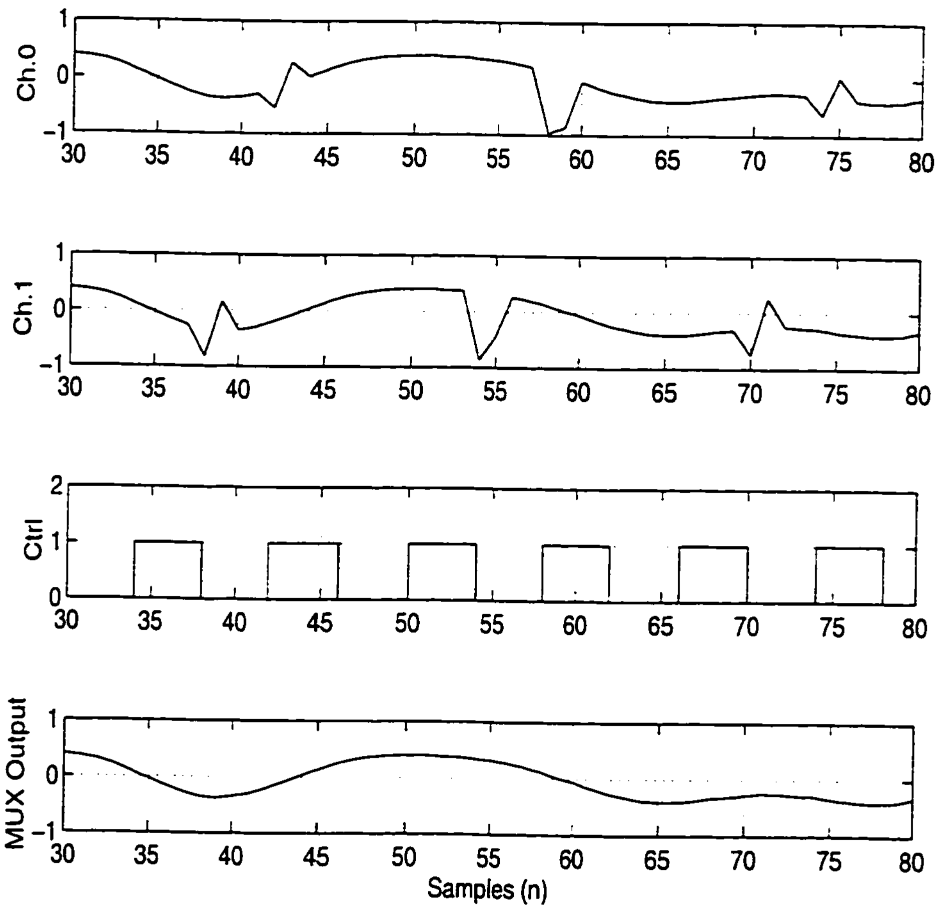


Figure 5.10 Simulated waveforms

5.4 Performance Analysis

The finite timing resolution of the dual-differentiator CIC decimator can degrade bit-error (BER) in two distinct ways: by causing the slicer to miss the widest eye opening, or by reciprocal mixing in which nearby interferers are brought in-band by phase noise of re-timing. We estimate the magnitude of these two effects in this section.

For typical oversampling ratios, we might have a resolution $T = T_{sy}/64$ which would have negligible loss due to mis-timing the eye [Fran81] for a modulation scheme like QPSK. In this type of system, where errors as large as $\pm T_{sy}/8$ are tolerable, even a fairly simple interpolator will suffice, and so our hardware savings are small. For high-order QAM, however, our system offers the fine resolution needed for good performance at a much lower cost than interpolation (see Section 2.4). We will derive the mean and variance of the timing jitter introduced by our method which can be used along with a system error budget and the analyses of [Fran81] to design a system.

For the case of reciprocal mixing, we derive a general formula that limits SNR degradation as a function of OSR, frequency error, channel separation and interferer strength. We show a good fit between theory and simulation, and present curves that can be used for system design in the specific practical case of an alternate-channel interferer.

Our method saves more when the required timing resolution increases. The SNR bound due to reciprocal mixing can be reduced by using analog or digital prefiltering in the interpolation method while analog prefiltering is required in our method.

5.4.1 Mean and Variance of Timing Jitter

Timing jitter is introduced when local timing is not synchronized with received symbol timing (i.e., transmitter timing). The difference between the transmitter and receiver timing, relative to the receiver timing, is defined as,

$$\alpha = \frac{|T_{sy} - T_{rsy}|}{T_{sy}} \qquad (5.24)$$

where T_{sy} and T_{rsy} are the transmitter and receiver symbol periods respectively. Ideally, they are the same. In reality, they are different since they use different clocks and the receiver timing can drift from the transmitter timing. In the proposed timing adjustment method, the timing is adjusted in the CIC decimator.

The normalized timing error is defined as,

$$e_{\tau}(n) = \frac{\tau_e(n) - \tau(n)}{T_{sy}}. \tag{5.25}$$

where $\tau(n)$ and $\tau_e(n)$ are the actual and estimated delays at time instant nT_1, respectively.

To simplify the analysis, we consider a case where the receiver rate is higher than the transmitter rate and the receiver timing needs to be advanced. Results for the slow-receiver case are also presented. In the analysis, we assume ideal timing error detection. Since the timing resolution is T, we adjust timing when the error exceeds $\pm T/2$. We only make timing decision once per symbol.

Figure 5.11 plots the accumulated timing error against time. Since the receiver clock rate is slightly higher than the transmitter clock, the timing phase error keeps growing (from **a** to **b** in Figure 5.11). When this timing error, as detected by the timing error detector, reaches the threshold $T/2$ (at point **b**), a timing phase adjustment $hT_{sy} = \delta T$ is made. Note that the minimum δ is 1. The timing error is reduced after the adjustment (at point **c**). Since we adjust the timing phase only once per symbol, the timing error may exceed the threshold until it is adjusted at the end of the symbol[1]. This is shown in Figure 5.11 in the segment from **c** to **d** where sample phase #6 reaches the threshold but drift continue until the next sample phase #0. Therefore, the maximum timing error $e_{\tau max}$ may be greater than $T/2$ and the minimum $e_{\tau min}$ may be less than $-T/2$. This overshoot is only significant when the receiver and transmitter clocks differ substantially. The curve of the timing phase error is a saw-tooth as shown in the figure. The phase error is a quasi-periodic signal due to overshoot. The average period T_p can be written approximately as,

$$T_p = \begin{cases} \frac{T_{sy}}{2\alpha OSR}, \alpha \leq \frac{1}{2OSR} \\ T_{sy}, \alpha > \frac{1}{2OSR} \end{cases}. \tag{5.26}$$

The closer the transmitter clock to the receiver clock rate, the longer the period. If both clocks are the same, the timing error becomes constant (only a phase shift).

[1] It could in theory interpolate timing estimation and adjust during symbol. Hardware both for the controller and decimator would be more complex.

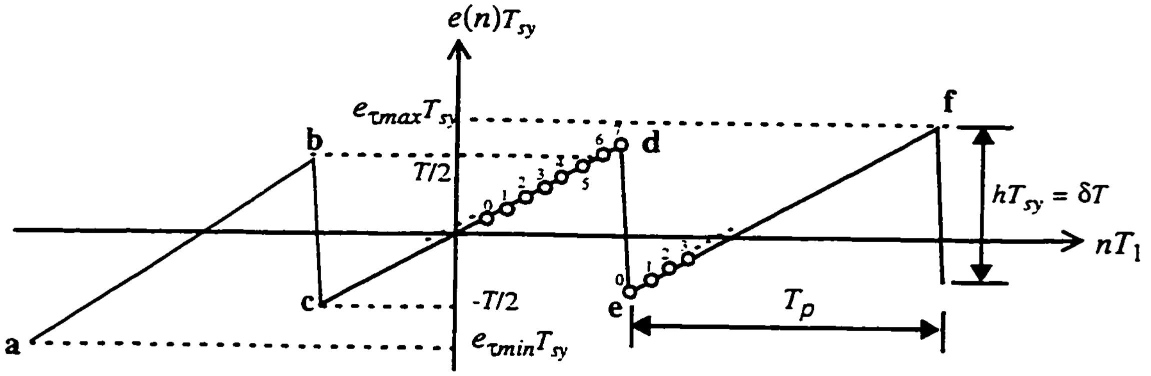


Figure 5.11 Saw-tooth timing phase error

For this fast-receiver case, we cannot "undershoot" and the minimum residual timing error is expressed as,

$$e_{\tau min}T_{sy} = -\frac{1}{2}T, \tag{5.27}$$

which is half the oversampling period.

The maximum timing error occurs at sample phase #7, and includes overshoot. The maximum error is,

$$e_{\tau max}T_{sy} = \left(\frac{7}{8}\alpha T_{sy} + \frac{1}{2}T\right), \text{ where } \alpha \neq 0. \tag{5.28}$$

Since $T_{sy} = 2 \cdot OSR \cdot T$, $e_{\tau min}$ and $e_{\tau max}$ can be written as,

$$e_{\tau min} = -\frac{1}{4OSR} \tag{5.29}$$

$$e_{\tau max} = \left(\frac{7\alpha}{8} + \frac{1}{4OSR}\right) \tag{5.30}$$

We estimate the mean of timing jitter as follows. The saw-tooth phase timing error oscillates between $(e_{\tau min}, e_{\tau max})$ with its mean value changing slightly. For a small $\alpha \ll \frac{2}{7OSR}$, the overshoot above $T/2$ is negligible and hence the timing error distribution is uniform between($T/2$, $-T/2$). For a larger α, it is pessimistic to approximate the distribution as uniform between $(e_{\tau min}, e_{\tau max})$ since the distribution densities between ($T/2$, $e_{\tau max}$) and ($-T/2$, $e_{\tau min}$) are less than that between ($-T/2$, $T/2$). Therefore, we can use the uniform distribution to estimate the timing jitter bound conservatively.

The normalized mean m_τ of the timing jitter is approximately,

$$m_\tau = \frac{7\alpha}{16} \tag{5.31}$$

Since the timing error is quasi-periodic, we can integrate the energy in the discrete tones to obtain the variance. The timing phase adjustment hT_{sy} needs to be known. In the proposed scheme, hT_{sy} must be an integer multiple δT of the oversampling period and has the following bounds,

$$T \le hT_{sy} \le T + \frac{7}{8}\alpha T_{sy} \tag{5.32}$$

Therefore, we have following bounds for h,

$$\frac{1}{2OSR} \le h \le \frac{1}{2OSR} + \frac{7}{8}\alpha \tag{5.33}$$

Note that for small enough α, $hT_{sy} = T$, that is $h = \frac{1}{2OSR}$.

The normalized variance of the timing jitter can be derived[1] and it is approximately,

[1] For a periodic saw-tooth timing phase error signal, the Fourier transform is

$$X(f) = \sum_{k=-\infty}^{\infty} r_k \delta\left(f - \frac{k}{T_p}\right), \text{ where,}$$

$$r_0 = 0 \text{ and } r_k = j\frac{(-1)^k h}{2k\pi}, k \ne 0.$$

Therefore, the variance of the timing error signal is

$$power = 2\sum_{k=1}^{\infty} r_k^2 = \frac{h^2}{2\pi^2}\sum_{k=1}^{\infty}\frac{1}{k^2} = \frac{h^2}{12}.$$

$$\sigma_\tau^2 = \frac{1}{12}h^2 \tag{5.34}$$

There are two other cases: where the receiver rate is higher than the transmitter rate in case II and where the relative rates between receiver and transmitter vary in the range ($-\alpha_{max}$, α_{max}) in case III. The mean and variance values of the timing jitter for these three cases are listed in Table 5.1. Note that case III is closer to reality.

Table 5.1 Mean and variance values of timing jitter

Case	$e_{\tau min}$	$e_{\tau max}$	m_τ	σ_τ^2	Note
I	$-\frac{1}{4OSR}$	$\left(\frac{7\alpha}{8}+\frac{1}{4OSR}\right)$	$\frac{7\alpha}{16}$	$\frac{1}{12}h^2$	$T_{sy} \geq T_{tsy}$
II	$-\left(\frac{7\alpha}{8}+\frac{1}{4OSR}\right)$	$\frac{1}{4OSR}$	$-\frac{7\alpha}{16}$	$\frac{1}{12}h^2$	$T_{sy} \leq T_{tsy}$
III	$-\left(\frac{7\alpha_{max}}{8}+\frac{1}{4OSR}\right)$	$\left(\frac{7\alpha_{max}}{8}+\frac{1}{4OSR}\right)$	0	$\frac{1}{12}h^2$	The relative rates of receiver and transmitter vary

The timing error is a measurement of how far the sampling is from the optimal point. One way to look at the resulting performance degradation is to use a so-called total energy of jitter, $\sigma_\tau^2 + m_\tau^2$ [Lee90], [Kim97].

To confirm these estimates, simulations have been done for timing jitter performance of the proposed timing recovery scheme in Case I. The variance σ_τ^2 versus α as a function of OSR is shown in Figure 5.12 and total energy ($\sigma_\tau^2 + m_\tau^2$) in Figure 5.13. The upper bound on h of (5.33) is used in the formula of Table 5.1. Note that $\sigma_\tau^2 + m_\tau^2$ is a few dB higher than σ_τ^2.

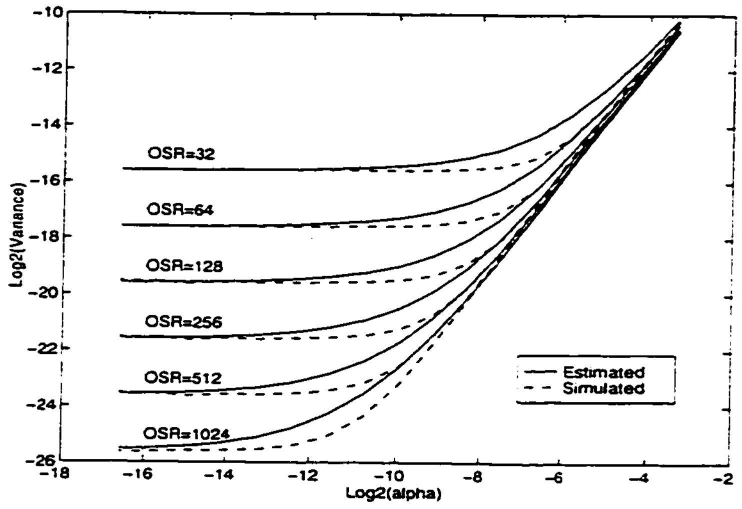


Figure 5.12 Timing jitter variance versus alpha as a function of OSR

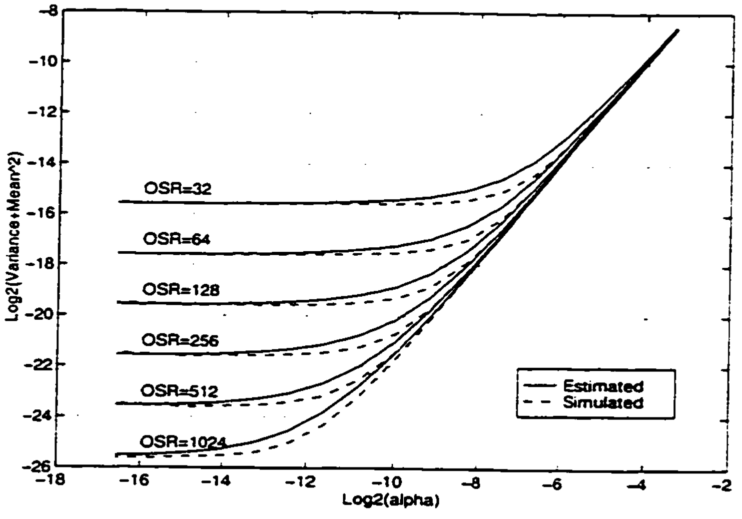


Figure 5.13 (Variance + mean^2) versus alpha as a function of OSR

The following observations can be obtained from Table 5.1, equation (5.33), Figure 5.12 and Figure 5.13:

- The mean value of timing jitter introduced by the proposed timing adjustment scheme is proportional to α for case I and II and independent of OSR. In other words, systematic timing bias comes from overshoot. The mean is zero in case III.

- For a small $\alpha < \frac{1}{7OSR}$, both σ_τ^2 and $(\sigma_\tau^2 + m_\tau^2)$ are dominated by the oversampling resolution error or OSR. Decreasing the timing resolution by 2 by decreasing OSR by 2 will quadruple both σ_τ^2 and $(\sigma_\tau^2 + m_\tau^2)$. In the case, the extra timing error introduced by the clock difference α is negligible.
- For a large $\alpha > \frac{4}{7OSR}$, both σ_τ^2 and $(\sigma_\tau^2 + m_\tau^2)$ are dominated by the transmitter and receiver clock rate difference α. Doubling α will quadruple both σ_τ^2 and $(\sigma_\tau^2 + m_\tau^2)$.
- For $\frac{1}{7OSR} < \alpha < \frac{4}{7OSR}$, where neither term dominates, the analytical value is pessimistic due to a pessimistic estimate of h.

5.4.2 SNR Bound due to Tone interferer

We have looked at the mean and variance of jitter as they could affect the slicer in a digital radio modem. It is also desirable to reject a tone interferer with digital filtering. In a narrowband receiver, this can result in cost reduction for the IF SAW filter. We show that the jitter will introduce phase noise (spurs) by mixing with the tone interferer. The spurs can easily limit the achieved SNR of the desired signal since this tone signal is sometimes up to 80 dB above the desired signal.

To characterize the SNR lower bound, we assume the tone interferer is a sinusoidal signal. This tone signal may represent the alternate channel signal in a TDMA system or an AMPS signal in the CDMA system. The tone interferer at the output of the dual-differentiator adjustable-timing-phase decimator, considering the timing error $e_\tau(n)$, can be described as,

$$y(n) = A \sin\left[2\pi f_0 (nT_1 + e_\tau(n) T_{sy})\right], \tag{5.35}$$

where A is the amplitude of the interferer, $f_0 \le \frac{f_s}{2 \cdot OSR}$ is the offset frequency of the tone interferer from the desired signal and $T_1 = RT$. Due to oversampling, $2\pi f_0 T_{sy} e_\tau(n) \ll 0$.

Equation (5.35) can be rewritten approximately as,

$$y(n) \cong A\sin(2\pi f_0 T_1 n) + 2A\pi f_0 T_{sy} e_\tau(n) \cdot \cos(2\pi f_0 T_1 n) \tag{5.36}$$

The second term is the noise introduced by mixing the interferer with the timing jitter, which is unwanted.

The timing error $e_\tau(n)$ in Figure 5.11 is almost a sawtooth with average period T_p. The timing error $e_\tau(n)$ is only quasi-periodic due to overshoot, but we can use a pure sawtooth for analysis. The sawtooth has a line spectrum, while the true quasi-periodic signal has a continuous spectrum with peaks around the lines of the sawtooth model. The sawtooth model estimates integrated energy of each peak.

The Fourier transform of the sawtooth model of $e_\tau(n)$ is discrete and its lines have amplitude (see the footnote of page 119),

$$|r_k| = \begin{cases} 0, k = 0 \\ \frac{h}{2k\pi}, k \neq 0 \end{cases} \tag{5.37}$$

Hence, the noise introduced by this timing jitter model is spurs with discrete spectral lines mixed with the interferer, is depicted in Figure 5.14. We also can see the mixing effect from (5.36). Note that the skirt extends all the way from the interferer to the desired band. It is also aliased repeatedly by being sampled at f_1.

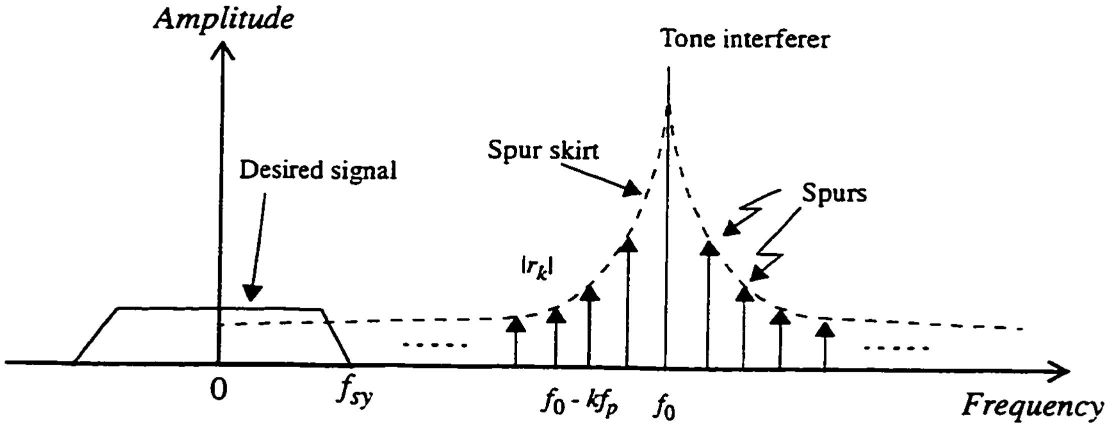


Figure 5.14 Phase noise of a tone interferer due to timing adjustment

Spurs in the desired band degrade SNR. To estimate SNR degradation, we need to know the total spur power of $e_\tau(n)$ in the desired band. We can calculate the total spur power of $e_\tau(n)$ by integrating elements in (5.37) which fall into the desired band. Assume the bandwidth of interest is equal to f_{sy}. The spur power of timing error $e_\tau(n)$ in the desired band is then approximately,

$$N_s \cong 2 \sum_{k = K_1}^{K_2} \left|r_k\right|^2 \tag{5.38}$$

where spurs numbered from K_1 to K_2 fall in the signal band $(0, f_{sy})$ and the factor of 2 accounts for the interferer image at $-f_0$. The two limits in (5.38) are defined as,

$$K_1 = \lceil (f_0 - f_{sy}) T_p \rceil \text{, and}$$

$$K_2 = \lfloor f_0 T_p \rfloor .$$

where $\lfloor x \rfloor$ represents the largest integer value not larger than x. Substituting (5.37) into (5.38) results in,

$$N_s \cong \frac{h^2}{2\pi^2} \sum_{k = K_1}^{K_2} \frac{1}{k^2} \tag{5.39}$$

Assume that the power of the tone interferer is A_i dB higher than the desired signal. We define an SNR lower bound as the desired signal power to the spur power. The SNR bound in dB can be obtained from (5.36) and (5.39) as,

$$SNR_{limit} = -A_i - 10\log 10 (4\pi^2 f_0^2 T_{sy}^2 N_s) . \tag{5.40}$$

In the actual case, $h = \frac{1}{2OSR}$ for small α. Substituting this into (5.39) and then the result into (5.40), the SNR bound in dB can be written as,

$$SNR_{bound} = 20\log 10\left(\frac{f_s}{f_0}C(K_1, K_2)\right) - A_i, \tag{5.41}$$

where $C(K_1, K_2) = \left(2\sum_{k=K_1}^{K_2}\frac{1}{k^2}\right)^{-\frac{1}{2}}$.

Note that SNR degradation is a function of OSR, frequency error α, channel separation f_0 and interferer strength A_i.

In wireless communications, we are often concerned with alternate-channel interferers for which $f_0 \cong 2f_{sy}$. For $\alpha < \frac{1}{20 \cdot OSR}$, $C(K_1, K_2)$ is approximately[1],

$$C(K_1, K_2) \cong 2\sqrt{\left(\frac{1}{K_1} - \frac{1}{K_2}\right)} \tag{5.42}$$

The achievable SNR bound was simulated to verify the estimate provided in (5.41) and the results are shown in Figure 5.15 and Figure 5.16.

The simulation conditions are as follows. The tone interferer is located at $2f_{sy}$ which models an alternate-channel interferer. The desired and tone interferers are of equal amplitude, that is, $A_i = 0$ dBc: correcting for a practical A_i is straightforward addition (equation (5.41)) A second-order lowpass delta-sigma modulator and a 3-stage dual-differentiator CIC decimator are used.

The SNR bound for OSR = 256 versus α is shown in Figure 5.15. Note that the estimated SNR bound is slightly worse (< 5dB) than the simulated values when the frequency

[1] When $K_2 > K_1 > 10$, we have approximately,

$$\sum_{k=K_1}^{K_2}\frac{1}{k^2} \cong 2\left(\frac{1}{K_1} - \frac{1}{K_2}\right)$$

where $K_1 = (f_0 - f_{sy})T_p = \frac{1}{2\alpha OSR}$. Therefore $\alpha < \frac{1}{20 \cdot OSR}$.

difference α is larger than $\frac{4}{7OSR}$; the estimate is conservative when overshoot is significant.

A comparison of estimated SNR bounds versus α for OSR = 64, 128, 256, 512 and 1024 is shown in Figure 5.16. Note that the SNR bound is a function of two variables: OSR and α. Some observations and explanations from studying (5.41), Figure 5.15 and Figure 5.16 are:

- For a small $\alpha < \frac{1}{7OSR}$: Increasing α by 2 will decrease the SNR bound by 3 dB and therefore the slope is -3 dB/Octave. The reason is that increasing α by 2 will: (a) decrease T_p by 2 for $\alpha \le \frac{1}{2OSR}$ in (5.26), (b) hence reduce K_1 and K_2 in (5.38) by 2.
- For a small $\alpha < \frac{1}{7OSR}$: Every doubling of OSR will improve the SNR bound by 3 dB and hence the slope is 3 dB/Octave. When OSR is doubled, f_s is doubled and 6 dB is gained in (5.41). This 6 dB is offset by losing 3 dB since T_p is reduced by 2 in (5.26).
- For a large $\alpha > \frac{4}{7OSR}$, the SNR bound is dominated by the frequency difference α. Every doubling of α will degrade the SNR bound by 6 dB and hence the slope is 6 dB/Octave. In this case, $T_p \equiv T_{sy}$ and $h \equiv 7\alpha/8$. The SNR versus α has a "corner" at $\alpha = \frac{4}{7OSR}$ which is $\log 2(\alpha) \cong -9$.

We summarize the above observations as follows:

- For a small $\alpha < \frac{1}{7OSR}$, the slope of SNR bound is -3 dB per octave of α or 3 dB per octave of OSR.
- For a large $\alpha > \frac{4}{7OSR}$, the slope of SNR bound is dominated by the frequency difference α. The slope is 6 dB per octave of α.

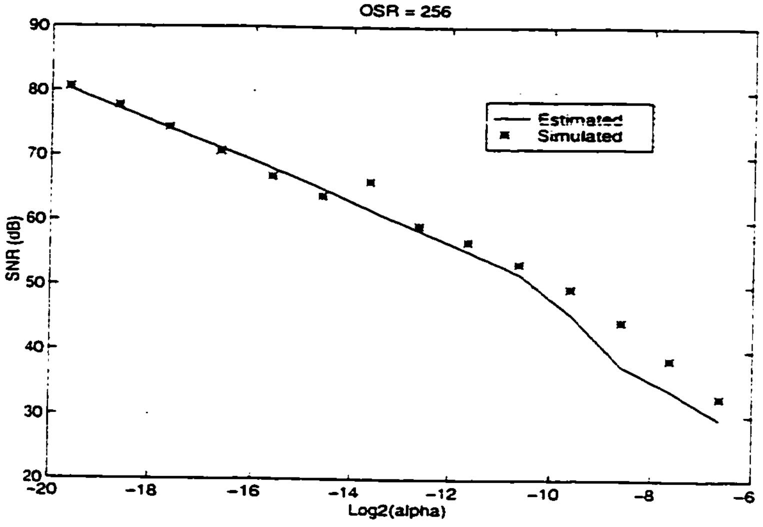


Figure 5.15 SNR versus alpha for OSR = 256. A 0 dBc tone located at alternate channel

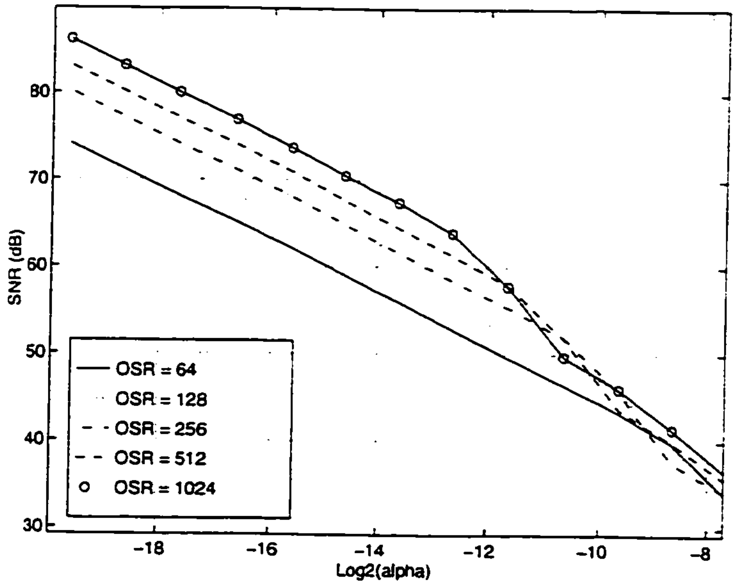


Figure 5.16 SNR versus alpha as a function of OSR. A 0 dBc tone located at alternate channel

Note that Figure 5.15 and Figure 5.16 are useful when we design a receiver. To achieve a required SNR at the detector, OSR and α should be considered together with the tone rejection filtering requirement. This is considered in the next subsection.

5.4.3 System Design Considerations

There are many considerations in estimating performance degradation. Timing jitter and tone-introduced noise are determined by several important parameters α, OSR, and the number of stages of the CIC decimators.

With crystal oscillators, very stable and accurate clocks can be generated. In general the frequency offset between the far end transmitter and local receiver clocks is small, typically with α less than 0.02% (i.e., $\log 2(\alpha) \equiv -12$) [Haou87]. With OSR = 256, we have the following estimates for timing jitter and SNR:

From Figure 5.13, the rms timing jitter is about $2^{-21.5/2} \equiv 0.006$. With this rms timing jitter, the contribution to Eb/No degradation is insignificant for BPSK/QPSK. From [Skla88], the Eb/No degradation for the BPSK signal in additive white Gaussian noise is less than 0.5 dB when the rms timing jitter is 3%. Every doubling of OSR improve rms jitter by a factor of 2 for a small α.

From Figure 5.15, the SNR with a 0 dBc interferer is 58 dB. This receiver would tolerate an interferer at +46 dBc while maintaining a 12 dB SNR reliable detection. Every doubling of OSR improves the SNR by 3 dB. In system design, interferer rejection is partition between digital components and analog IF filters (e.g., SAW filters) preceding the delta-sigma modulator.

5.5 Simulation and Experiment

The purpose of this section is twofold:

- to verify the validity of the proposed method by simulation in ideal conditions.
- to verify the performance of the technique when there is an alternate channel interferer (+35 dBc) using real data collected from a real system including of a bandpass $\Delta\Sigma$ chip.

A. Simulation Results

In the following SPW simulation [SPW], ideal channel conditions are assumed, that is, there is no fading and no additive white Gaussian noise. The performance of the timing loop is evaluated. As discussed in the previous sections, the performance of the proposed timing recovery circuit depends on several factors: adjustable-timing-phase decimator, the timing error detector, loop filter, loop delay, etc.

In the simulation, the timing error detector is from equation (5.10) and the loop filter is a first-order lead-lag filter (see equation (C.1) in Appendix C).

The transmit IF signal data is generated by the structure shown in Figure 5.17, where each block is implemented as a function in SPW. In the figure, a QPSK source generates a pair of I and Q data at a rate of f_{sy}, sampled at $4f_{sy}$. The I and Q signals go to a pair of root raised cosine (RRC) pulse shaping filters clocked at $4f_{sy}$. The rolloff factor of the RRC filters is 0.5. Then a pair of timing-drift filters is used to simulate the timing drift between a transmitter and a receiver. After being interpolated by a factor of $R_I = 128$, the I and Q signals at $512f_{sy}$ are mixed with quadrature LO signals. Interpolation is necessary to simulate an analog signal. The carrier frequency of the LO signals is $128f_{sy}$ which is 1/4 sampling rate. The IF signal is obtained by adding I and Q signals.

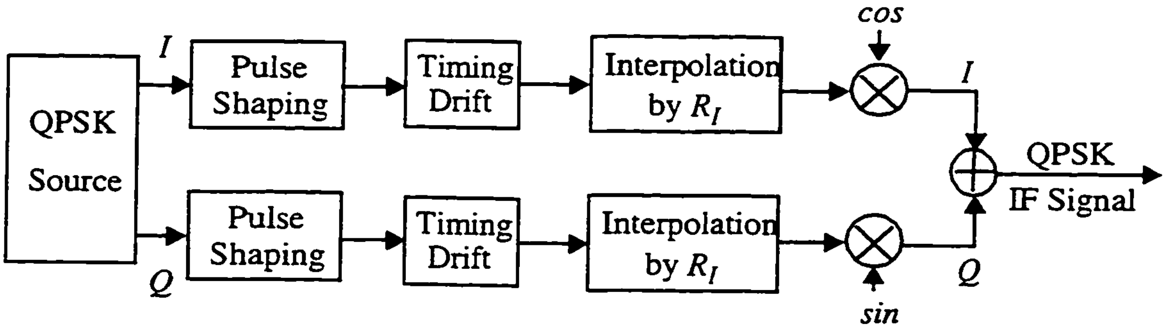


Figure 5.17 A block diagram for generating an IF QPSK signal in SPW

The IF signal from the transmitter is sent to a receiver similar to that shown in Figure 5.8 where the lowpass ΔΣ modulator is replaced by a bandpass version. A 4th-order bandpass ΔΣ modulation is used, where the OSR is 256. The sequences of {1,0,-1,0,...} and {0,1,0,-1,...}are used as the cosine and sine LO signals, respectively, to translate the bandpass IF signal to baseband. A 3-stage dual-differentiator adjustable-timing-phase CIC decimator is used to perform timing recovery.

Two cases are tested here:

- Case I: the QPSK data is a training sequence with alternating 1's and 0's, that is, {(1,1), (0,0), (1,1), (0,0,...}.
- Case II: the QPSK data is random.

The phase shift error and frequency drift error are simulated for each case. The initial phase shift is approximately a quarter of a symbol period and the local clock frequency is $\alpha = 0.1\%$ fast relative to the transmitter sampling rate.

The simulation results are plotted in Figure 5.18 for case I and Figure 5.19 for case II. The timing error signals e_s's from (5.10) are shown in Figure 5.18 and Figure 5.19, measured at the loop filter output and normalized to the maximum output values of y_Q and y_I. The thresholds for the timing error signals used in the controller to advance or retard the timing is 0.05 and 0.2 in case I and case II respectively. Due to some delays introduced in creating the transmitter signal and demodulating the received signal, the actual output and timing error start from around $100T$ as can be noted in the figures.

The constellation scatter plots are shown in the figures where the data were taken starting from $400T$. The clean scatter plots in the figure are due to the ideal channel conditions used in the simulation. The constellations points are slightly large which are due to the following nonideal conditions:

- The loop delay is about 3 symbol intervals. This delay causes extra timing adjustment and hence introduces larger timing jitter.
- The bandwidth of loop filter is not optimized.
- Gardner's timing error detection algorithm is susceptible to inter-symbol interference (ISI) for the mid-points $y_I(k\text{-}1/2)$ and $y_Q(k\text{-}1/2)$ in (5.10) when the rolloff factor is not equal to 1.

This simulation demonstrates that the dual-differentiator timing adjustment techniques operates properly in a timing recovery loop.

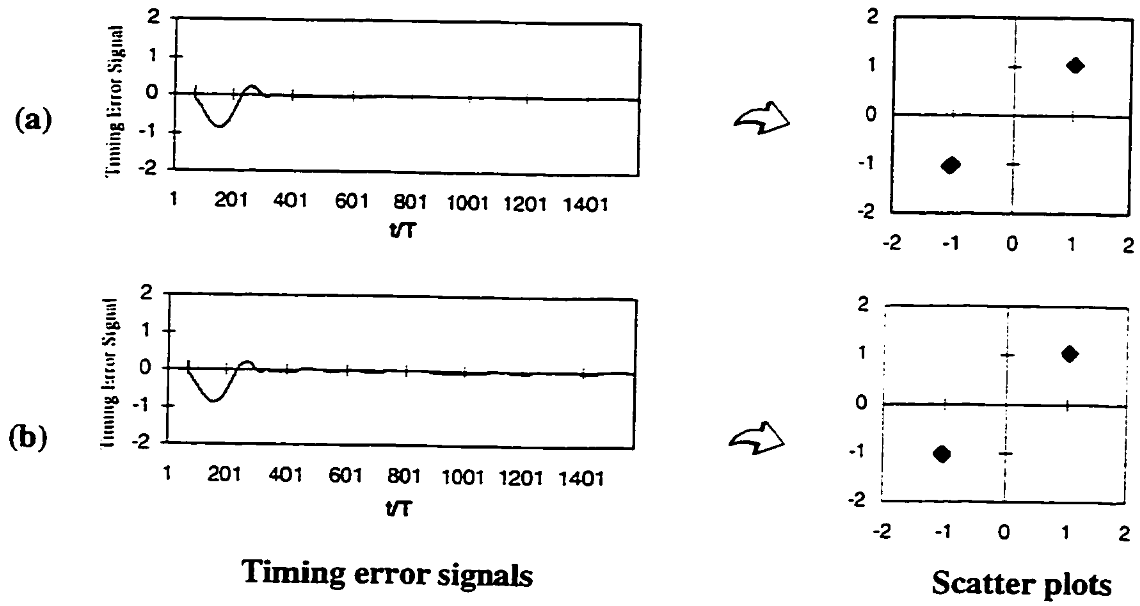


Figure 5.18 Timing error signals (left) and scatter plots (right) for training sequence: (a) phase shift and (b) frequency drift

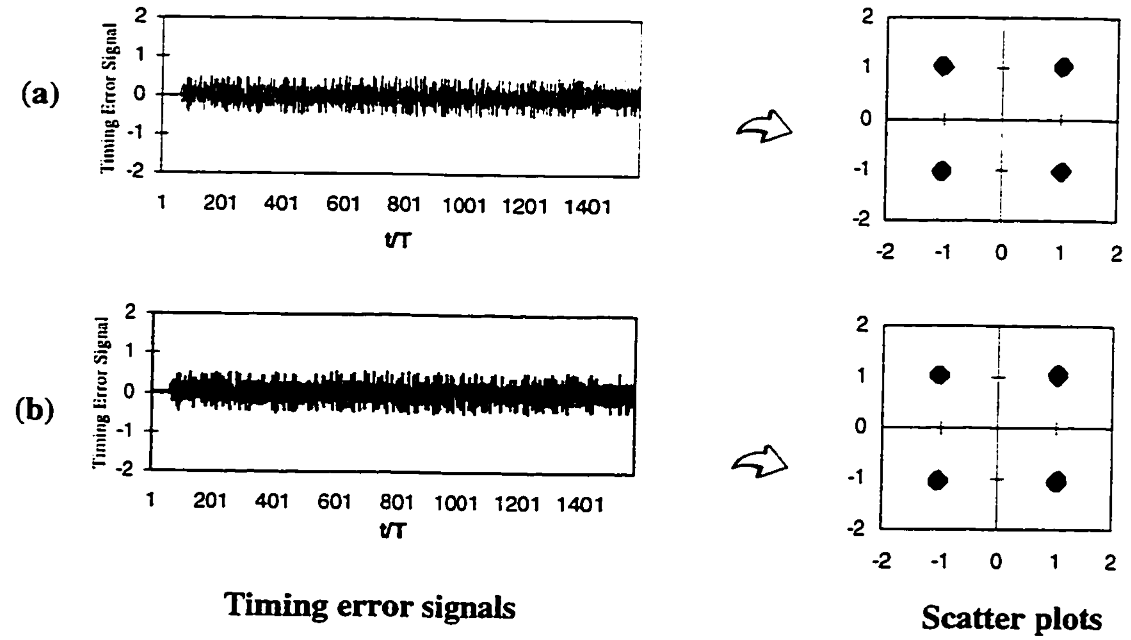


Figure 5.19 Timing error signals (left) and scatter plots (right) for random data: (a) phase shift and (b) frequency shift

B. Experimental Results

An off-line experiment has been conducted to further verify the proposed method in the presence of an interferer and with real $\Delta\Sigma$ noise. The setup for the experiment is shown in Figure 5.20, where an IF QPSK system is tested. The QPSK data is generated by a PC and is sent to a waveform generator (Rohde&Schwarz Dual Arbitrary Waveform Generator - ADS) which generates a repeating sequence of 512 symbols. The pulses go to a signal generator which upconverts the signal to a 10 MHz IF QPSK signal.

At the receiver end, a second-order bandpass $\Delta\Sigma$ modulator from Singor and Snelgrove [Sing95] clocked at 40 MHz is used to digitize the IF signal.This bandpass $\Delta\Sigma$ modulator provides about 55 dB SNR in a 200 KHz bandwidth. In the setup, a pulse generator (HP 8131A) provides a 40 MHz clock to the bandpass $\Delta\Sigma$ modulator and also provides a reference signal to lock the signal generator (Rohde&Schwarz Signal Generator - SHMU). The symbol rate for the QPSK is 100 KHz. The rolloff factor for the root raised cosine pulse filter is 0.5. The digitized 1-bit signal is captured by a logic analyzer (HP 16500A Logic Analyzer) controlled by GPIB and stored in a PC. Then data is transferred to a SUN Workstation on a floppy disk.

SPW is used for data processing. A 3-stage dual-differentiator adjustable-timing-phase CIC decimator is used to perform timing recovery. Since the demodulation implemented in SPW uses coherent detection, carrier recovery is required to synchronize the local LO frequency (frequency and its phase) with the received carrier. This is realized with a simple 4th-power phase estimator [Lee90].

Two cases are tested in this experiment. In one case, there is no interferer. In another case, there is an interferer located 400 KHz away from the center frequency, which is the "alternate channel" for a GSM system. The interferer level is +35 dBc.

The scatter plots for both cases are shown in Figure 5.21 (a) and (b), respectively. The eye opening loss is about 1.1 dB in the first case and is about 1.7 dB in the second case. This extra 0.6 dB is introduced by the SNR loss due to phase noise (spurs) created by the timing adjustment.

There are three sources causing the eye opening loss in the interferer-free case:

- The author found after this work that the data from the signal generator is shaped by a raised cosine filter instead of root raised cosine filter. Hence ISI was introduced. This is the main reason for the eye opening loss.

- The difference between the internal clock of the signal generator and the pulse generator clock. Although they are locked, the instabilities of the two clocks make a difference. The author was told after the test that the pulse generator is poor.
- The simple phase estimation is used in the simulation and the phase error contributes to the loss.

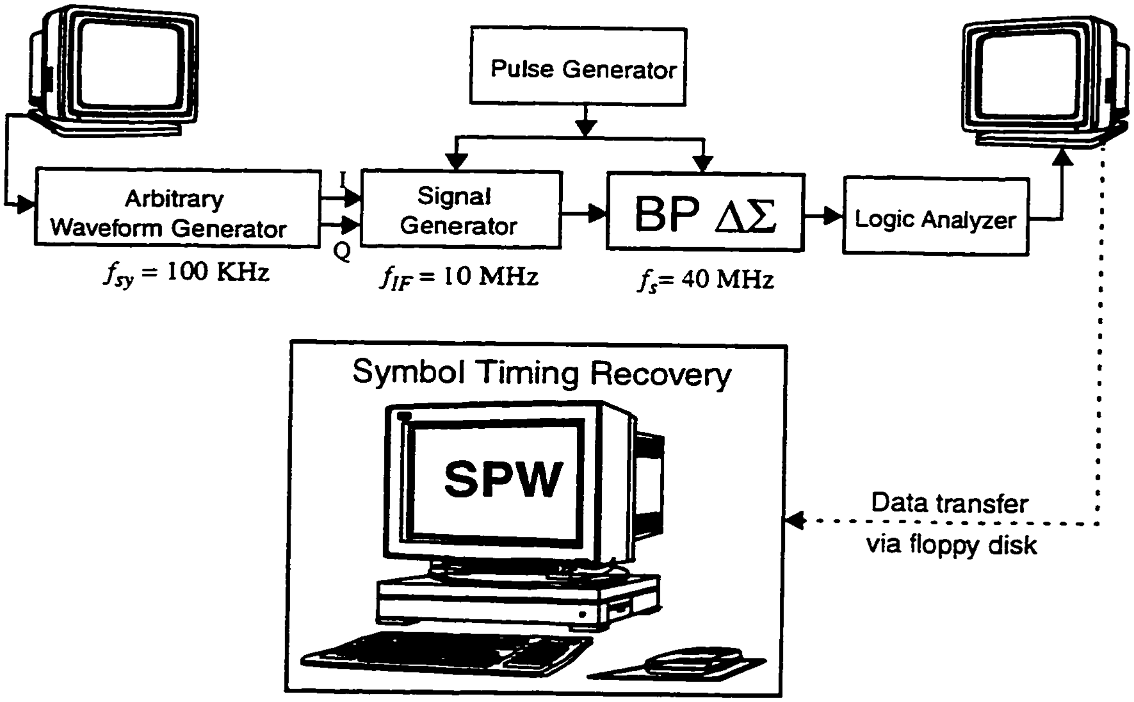
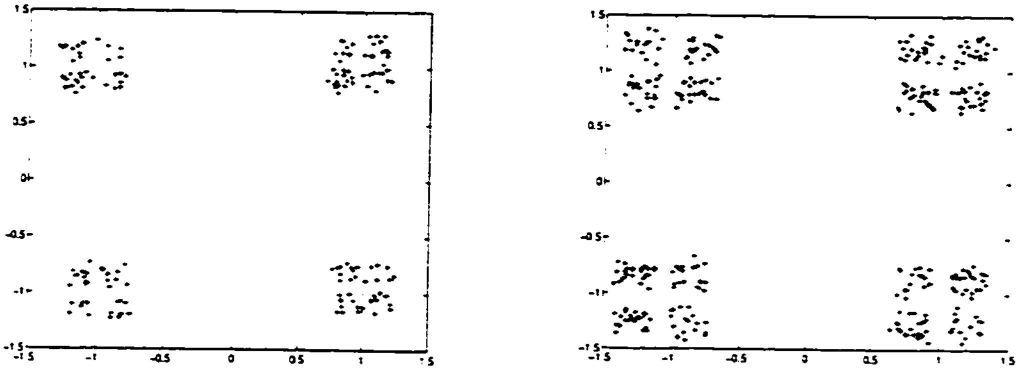


Figure 5.20 Experiment setup for timing recovery of an QPSK IF system

When there is an interferer, spurs due to re-timing contribute to further eye opening loss. The clock instability was estimated to be about 0.1 %. The OSR is 200. The SNRs are about 12.8 dB and 9.1 dB for the first and second cases respectively. Hence the SNR degradation is about 3.7 dB due to interferer mixing and the derived SNR bound relative to the desired signal level is 11.5 dB. From Figure 5.16, the SNR bound for a 0 dBc interferer is about 48 dB. Hence the SNR bound relative to the desired signal level is 48-35 = 13 dB. As we can see that the estimated SNR bound is 1.5 dB better than the measured.

This test demonstrates that the method works in the presence of practical channel impairments and in conjunction with carrier recovery. It is within 1.5 dB of consistency with Figure 5.16.



(a). No interferer

(b). With +35 dBc tone interferer located at offset +400 KHz

Figure 5.21 Output scatter plots for two different cases

5.6 FPGA Implementation of Symbol Timing Recovery for BPSK

In this section, an FPGA implementation of symbol timing recovery using the proposed method is described. The objectives are:

- to demonstrate that the hardware of the proposed technique is simple.
- to validate the performance of the proposed technique for symbol timing recovery.
- to verify the stability of the timing recovery loop for long run.

5.6.1 Architecture and Circuit Design

A. Architecture

The timing recovery loop architecture is shown in Figure 5.22 where the incoming signal is a BPSK modulated IF signal with a center frequency of 10.24 MHz. A second-order bandpass $\Delta\Sigma$ modulator clocked at f_s = 40.96 MHz is used to digitize the IF signal. The sampling rate f_s is four times the IF center frequency. The symbol rate is 160 KHz. Therefore the OSR is 128.

A multi-stage polyphase technique is utilized to save power. The first stage is a 4-phase polyphase CIC decimator (P4 DDC in the figure) and re-timing occurs at the second stage which is a second-order dual-differentiator CIC decimator. P4 DDC, clocked at f_{s0} = 10.24 MHz, downconverts the digital IF to baseband and downsamples the sampling rate by 4. Its architecture is shown in Figure B.2(b) (only one channel is required for BPSK). Since the dual-differentiator CIC decimator is clocked at f_{s0}, the minimum timing phase adjustment is $f_{s0}T_{sy}$ = 1/64 symbol interval which is sufficient for BPSK.

The timing in the 2-stage dual-differentiator CIC decimator is controlled by f_1 and f_{1a}. These clocks are obtained by dividing f_{s0} in two 7/8/9 variable counters as shown in the figure. The input rates to the halfband filter, data filter and timing error detector are f_1, f_2, and f_3 respectively. The output sampling rates for those components are f_2, f_3, and f_4 respectively. Clocks f_2, f_3, and f_4 are generated in the box Clk_Gen1 by dividing f_1 by 2, 4, and 8 respectively. The operating principle of the timing recovery block diagram in Figure 5.22 is the same as that in Figure 5.8.

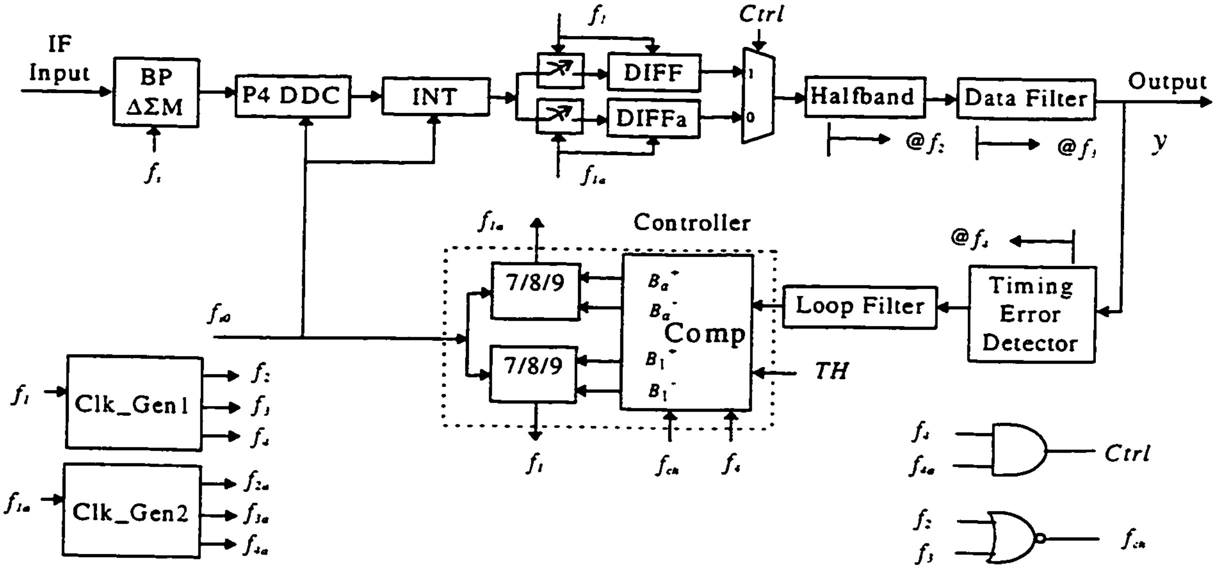


Figure 5.22 A block diagram for the timing recovery via decimation

The purpose of the symbol timing recovery circuit is to synchronize the local timing with the transmitter timing. Thus, E[f_3] = $2f_{sy}$ and E[f_4] = f_{sy}.

The controller here consists of one comparator and two 7/8/9 variable counters. The timing diagram is depicted in Figure 5.23, where a timing adjustment of $1/f_{s0}$ is made in the middle of the symbol for channel f_{1a} and at the end for f_1.

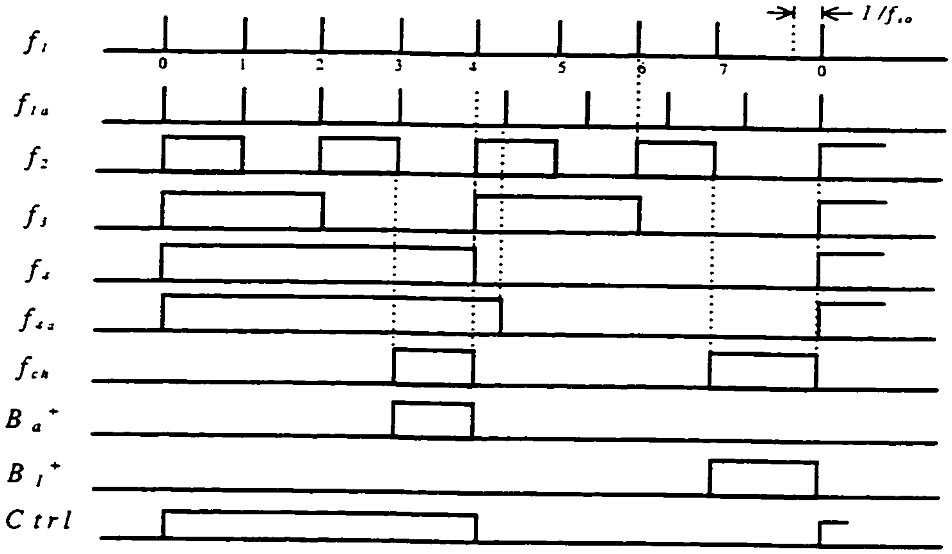


Figure 5.23 Timing diagram in the timing recovery circuit

The output of the loop filter is compared to a threshold *TH* by comparator Comp to decide whether to advance or retard the timing phase. Signal f_{ch} is used to enable the signals B_a^{+}, B_a^{-} in the middle of the symbol and B_1^{+}, B_1^{-} at the end. The rule controlling the timing phase is as follows:

- If (loop filter output > *TH*), then timing phase advance is needed. In this case, $B_a^{+} = B_1^{+} = 1$ and divide-by-9 counters are selected. Both B_a^{+} and B_1^{+} are gated by f_{ch} and f_4 to ensure that only one adjustment is made within one symbol period for clocks f_{1a} and f_1, respectively.
- If (loop filter output < -*TH*), timing phase retard is needed. In this case, $B_a^{-} = B_1^{-} = 1$ and divide-by-7 counters are selected. Both B_a^{-} and B_1^{-} are also gated by f_{ch} and f_4.
- If (-*TH* < loop filter output < *TH*), no timing phase adjustment is needed. In this case, $B_a^{+} = B_1^{+} = B_a^{-} = B_1^{-} = 0$ and divide-by-8 counters are selected.

Clock f_{4a} is created via the box Clk_Gen2 by dividing f_{1a} by 8. The generation of *Ctrl* and f_{ch} is also shown in the figure.

In the design, 8-bit data in two's complement form are used in the data path such as in the halfband filter, data filter, timing error detector, and loop filter. The circuits in the design are given in Appendix C. The SPW fixed point simulation tool was used to perform simulation to verify the validity of the circuit shown in Figure 5.22. The input data was generated by the circuit shown in Figure 5.17.

B. FPGA Chip Gate Counts

Gate counts in the implemented FPGA circuit and power dissipation estimation in a 3.3-V 0.5 μm CMOS technology [Baza97] are given in Table 5.2. A 50 % gate activity is assumed. The total number of gates used is 2321. Also listed are the proportion of gate counts to the total. One can see that three filters (dual-differentiator CIC, halfband and data) use most of the gates (20.6%, 20.0% and 36.2% respectively). The estimated power dissipation is only 1.5 mW. This can be credited to the polyphase architecture in the first decimator and the low complexity of the proposed timing recovery circuit. A combination of the multi-stage polyphase CIC decimator and the dual-differentiator CIC decimator can achieve more power savings.

Note that the two cascaded integrators in the dual-differentiator CIC decimator consume 58.2 % power. The power consumed by extra circuits due to the dual-differentiator structure (one differentiator, one 7/8/9 counter, and half comparator) in the timing loop is less than 10 %. The proposed re-timing technique is power efficient.

Table 5.2 Gate counts and power estimation in the timing recovery circuit

Subcircuits	Clock rate (MHz)	Gate counts		Power estimation	
		#'s of gates	Proportion (%)	Power dissipation (μW)	Proportion (%)
Commutator	40.96	4	0.2	81.9	5.2
	10.24	4	0.2	20.5	1.3
P4 DDC	10.24	4	0.2	20.5	1.3
Dual-differentiator decimator	10.24	178	7.7	911.4	58.2
	1.28	300	12.9	192	12.3

Table 5.2 Gate counts and power estimation in the timing recovery circuit

Subcircuits	Clock rate (MHz)	Gate counts		Power estimation	
		#'s of gates	Proportion (%)	Power dissipation (µW)	Proportion (%)
MUX	1.28	25	1.0	16	1.0
Halfband filter	0.64	464	20.0	148.5	9.5
Data filter	0.32	840	36.2	134.4	8.9
Timing error detector	0.16	192	8.3	15.4	1.0
Loop filter	0.16	168	7.2	13.4	0.9
Comparator	0.16	110	4.7	8.8	0.6
7/8/9 counters	0.16	32	1.4	2.6	0.0
Total		2321	100.0	1565.4	100.0

5.6.2 Test Result

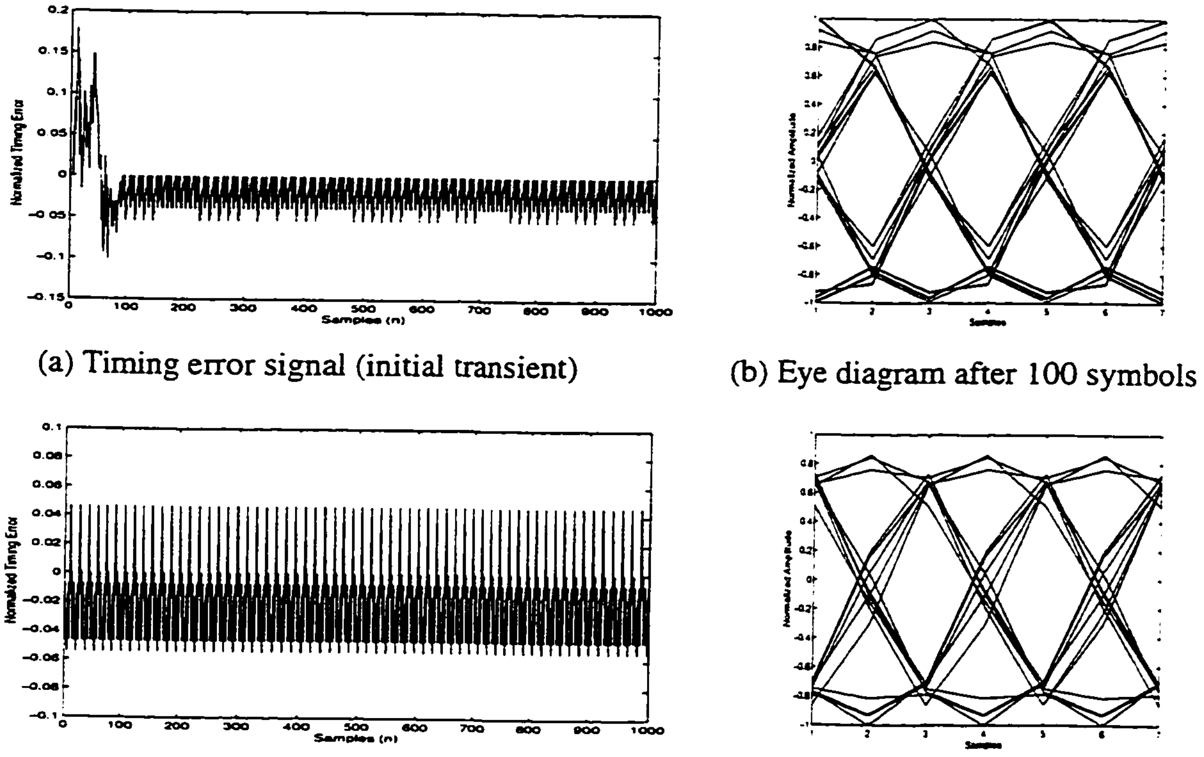
The chip has been tested. The test setup is described in Appendix C.10. The threshold for the comparator was set to a binary 00001000 which corresponds to 8/127 = 0.063. The test results are shown in Figure 5.24, where timing phase error signals e_S's and eye diagrams are given. The results in Figure 5.24 (a) and (b) were captured immediately after resetting the chip. Note that the samples are at twice the symbol rate. The timing recovery circuit took about 100 samples to converge as can be seen from Figure 5.24 (a). One can see the transition of timing phase adjustment. The eye diagram shown in Figure 5.24 (b) is plotted after 100 symbols. Note that the samples in Figure 5.24 (b) and (d) are at twice the symbol rate.

A slightly different result was obtained after a long run, as shown in Figure 5.24 (c) and (d). It is interesting to note that two converged timing errors are locked with different limit cycles although they are within the prescribed range [-0.063, 0.063].

In the figure, the mean value of the eye opening is 0.7 and the samples of the eye opening is distributed between (0.8, 0.6). The rms loss of eye opening is about 0.75 dB which is the bound for Eb/No degradation [Jeru94].

This Eb/No degradation bound of 0.75 dB measures the performance of the whole timing recovery loop. The loss may be from the timing error detection, loop delay, non-optimized design, etc. The design has not been optimized for parameters such as the loop bandwidth and threshold. Better results can be obtained by optimizing the parameters.

The FPGA chip was run overnight to verify the stability of the circuit. Stable operation was observed.



(a) Timing error signal (initial transient)

(b) Eye diagram after 100 symbols

(c) Timing error signal after convergence showing a different limit cycle

(d) Eye diagram after convergence during the limit cycle of (c)

Figure 5.24 Measured results: timing errors and eye diagrams

5.7 Summary

In this chapter, we have shown how to realize symbol timing recovery in a ΔΣ modulator based receiver by moving the re-timing function into the decimator. To combine timing recovery and decimation into one function, a modified timing-adjustable CIC decimator

Chapter 6
Conclusions and Future Work

In this thesis, we have developed three techniques which can substantially reduce power consumption for portable applications: a novel double-sampling technique for improving the SNR attainable in power-efficient $\Delta\Sigma$ modulators; a combination of polyphase and multistage techniques to minimize power in high-rate decimation; and a re-timing decimation technique to avoid the need for an interpolator in timing recovery circuits that follow the $\Delta\Sigma$ ADC. The novel double-sampling lowpass $\Delta\Sigma$ ADCs can be used in digital baseband digitization receivers. The multi-stage polyphase CIC decimators can be used with double-sampled $\Delta\Sigma$ modulators to achieve low power in baseband digitization receivers. They can also be used in IF digitization receivers to achieve low power or high speed (e.g., GHz $\Delta\Sigma$ modulators). The re-timing decimation for symbol timing recovery can be used in both baseband and IF digitization receivers to achieve low power operation. These techniques are summarized as follows.

6.1 Double-Sampling Techniques

Double-sampling is an efficient technique for low power applications, especially in a SC $\Delta\Sigma$ modulator. By using double-sampling, we can ideally improve the SNR by ($6M$+3) dB where M is the modulator order or we can maintain the same SNR while decreasing the clock rate by 2 to reduce the power consumption. However, capacitor mismatch limits the achievable SNR. We have shown by quantitative analyses and simulations that the SNR loss is significant. For example, the SNR loss is 18 dB for 0.4 % mismatch in a second-order double-sampled modulator with EOSR = 128. This limits the achievable SNR to 10~12 bits.

We have provided a practical solution to the double-sampled $\Delta\Sigma$ modulators. We have shown that the dominant mismatch problem turns out to be in the feedback path to the input integrator. By using a novel bilinear integrator feedback in this critical path, we are able to mitigate the mismatch effect by a first-order noise-shaping term. In typical examples, this reduces the SNR loss due to mismatch from about 30 dB in the conventional circuit to less than 3dB and hence makes double-sampling practical.

We have demonstrated that double-sampled second-, third-order and fourth-order $\Delta\Sigma$ modulators with novel double-sampling techniques are insensitive to capacitor mismatch. The SNR degradation is negligible compared to the improvement ($6M$+3) dB that can be obtained by double-sampling with a typical mismatch range of 0.1~0.5 %. The mismatch requirement for the fourth-order modulator is slightly tighter, requiring 0.25~0.4%.

We have demonstrated the low power consumption by implementing a second-order double-sampled modulator in 0.25 μm SOI technology and the simulated SNR is 81 dB with an OSR of 100. The estimated power is 1 mW with a 0.9 V power supply. The techniques are also suitable for other battery-powered applications such as in hearing aids.

6.2 Multi-Stage Polyphase CIC Techniques

Following the sigma-delta modulator in a radio is a decimator, which does simple filtering at the high (oversampled) rate. The high-speed part of this circuit, which might typically consist of three or four 16~24-bit accumulators, often dominates power consumption and limits clock rates. We have shown that it is not a solution to simply use a multi-stage CIC decimator since full-rate accumulators still need 8~11 bits. A solution is to combine multi-stage CIC decimators with polyphase techniques, which also provides a solution to the time misalignment in a DDC.

We have demonstrated not only a solution for decimation in low-power and GHz-rate $\Delta\Sigma$ modulators, but, more importantly, provided a design method for a multi-stage polyphase CIC decimator in these applications. The applications of the new design are wide, and can be in any low power decimators and DDCs as well as GHz-rate decimators. The new design allows further power savings by reducing power supplies on the low-speed circuits, a technique which would further exploit the reductions in clock rate to save power.

We have shown how to determine the polyphase components by considering tone interference rejection and noise power aliasing rejection as well as by trading off SNR with complexity. We have shown that it may be possible to simplify design by reducing the required number of stages in the first polyphase CIC decimator by one, thus saving $\log_2(R_1)$ bits, where R_1 is the first downsampling ratio. In a high speed $\Delta\Sigma$ modulator, its achievable SNR may be limited by thermal noise, clock jitter, etc. The SNR loss due to reducing CIC

Timing jitter and the SNR bound are determined by two factors: α (the transmitter and receiver clock rate difference) and OSR. The effects are: (a) for a small $\alpha < \frac{1}{7OSR}$, the rms timing jitter is dominated by OSR and its slope is -3 dB per octave of OSR. The slope of SNR bound is -3 dB per octave of α or 3 dB per octave of OSR. (b) for a large $\alpha > \frac{4}{7OSR}$, the rms timing jitter and SNR bound are dominated by the frequency difference α and their slopes are 3 dB and 6 dB per octave of α respectively.

We have shown the validity of the proposed method by simulation, experiment and in a real-time FPGA implementation. One experiment demonstrates that the method works in the presence of alternate interferer and in conjunction with carrier recovery. The SNR bound is within 1.5 dB of consistency with the estimate. The low complexity and stability of the technique was demonstrated. The chip is able to do re-timing for a Binary Phase Shift Keying (BPSK) IF signal. The eye opening has been measured in the real-time implementation and it contributes to an Eb/No degradation bound of about 0.75 dB.

The applications of the re-timing technique can be in any receiver employing a $\Delta\Sigma$ modulator. The modulation schemes can be QAM which may require tighter timing jitter for 64 or higher QAM. The complexity of the proposed technique does not increase with a tighter timing jitter requirement, whereas that of the interpolation method does [Laak96].

6.4 Future Work

There are some issues which need further considerations. They are listed as follows:

1- In a double-sampled bandpass $\Delta\Sigma$ modulator, capacitor mismatch creates images to the interferers. These images degrade the achievable SNR. Using the proposed double-sampled bilinear integrator, we may build bandpass $\Delta\Sigma$ modulators which are insensitive to capacitor mismatch. Modulator architectures need to be investigated. Ash Swaminathan has obtained a preliminary result [Swam97].

2- Wideband DDCs implemented by multi-stage polyphase decimators require polyphase NCOs as shown in Figure B.3 to achieve low power consumption and high speed operation. Investigation of such polyphase NCO architectures is necessary to make them possible. Building a FPGA wideband DDC chip to downconvert a multi-channel IF signal digitized by a wideband ADC may be useful.

3- The timing error may be reduced by interpolating the $\Delta\Sigma$ modulated signal or by modulating the control word for the NCO by a $\Delta\Sigma$ modulator. The latter was addressed in [Rile93] to reduce the close-in phase noise in a fractional-N frequency synthesizer. The $\Delta\Sigma$ modulated bit stream may be used to control the NCO and the energy at the high frequency will be filtered by the close-loop filter.

4- A preliminary design of a different structure, the delay-adjustable CIC decimator is given in Appendix D. Further research on the performance and limitation of such decimators is required

5- In a smart antenna, several RF signals are received by respective antennas. A wideband receiver is required for each RF chain to select a desired channel and downconvert it to an IF or a baseband digital signal [Mito95], [Kenn95]. The desired IF or baseband digital signals from respective antennas then go to a digital beamformer where noise and co-channel interference are suppressed and the desired signal is enhanced. The wideband DDC consisting of the proposed polyphase CIC decimator may be used in a wideband receiver. The proposed delay-adjustable CIC decimator may be used to replace an interpolator in a digital beamformer to adjust phase [Prid79]. Oversampled signal processing techniques for smart antennas need further investigation in terms of performance and limitation.

Appendix A
Equations for Double-Sampled Delta-Sigma Modulators

The derivations of equations for double-sampled $\Delta\Sigma$ modulators presented in Chapter 3 are given in this appendix.

A.1 Derivations of Eqs. (3.9) and (3.15)

The exact expression on the left sides of (3.9) and (3.15) should be,

$$Y(1+3(1-z^{-1/2})) = Y+3(1-z^{-1/2})Y, \tag{A.1}$$

which is approximately equal to Y within the band of interest.

A.2 Coefficients in Figure 3.13

More generally, gains of g_1 and g_2 are included to account for the comparator (two level ADC) and the two-level DAC, respectively. The equation describing Figure 3.13 in the ideal case can be readily obtained as,

$$\left(\frac{1+(k_2g_1g_2-1)z^{-1/2}+k_2g_1g_2z^{-1}}{(1-z^{-1/2})}\right)Y = k_1g_1\frac{z^{-1}}{1-z^{-1/2}}X+E. \tag{A.2}$$

The design goal is to find the parameters in order to make the signal and noise transfer functions be low-pass and high-pass types respectively as described in Section 2.2. The left side of (A.2) can be expressed as,

$$\frac{k_2g_1g_2(1-z^{-1/2})^2+(1-3k_2g_1g_2)(1-z^{-1/2})+2k_2g_1g_2}{(1-z^{-1/2})} \tag{A.3}$$

To achieve the lowpass and highpass filtering for the signal and noise transfer functions, we can arbitrarily choose k_2, g_1 and g_2 as can be seen from (A.3). To achieve a unity signal gain, we have

$$k_1 = 2k_2g_2 \tag{A.4}$$

There are many choices for parameters. If we set $k_2 \cdot g_1 \cdot g_2 = 1/3$ and $k_1 \cdot g_1 = 2/3$, a better lowpass function can be obtained and (A.2) can be rewritten as,

$$\left(\frac{2+(1-z^{-1/2})^2}{3(1-z^{-1/2})}\right)Y = \frac{2}{3}\frac{z^{-1}}{1-z^{-1/2}}X+E. \tag{A.5}$$

Within the band of interest, the following is obtained approximately,

$$Y \cong Xz^{-1}+\frac{3}{2}(1-z^{-1/2})E. \tag{A.6}$$

A.3 Coefficients in Figure 3.16

In Figure 3.16 in the ideal case, the following relationship between the input and output can be obtained as,

$$\left(1+\frac{k_4g_1g_2z^{-1/2}}{1-z^{-1/2}}+\frac{k_2k_3g_1g_2z^{-1}(1+z^{-1/2})}{(1-z^{-1/2})^2}\right)Y = k_1k_3g_1\frac{z^{-3/2}}{(1-z^{-1/2})^2}X+E. \tag{A.7}$$

The left side of the above equation can be rearranged as,

$$\left(\frac{1+(k_4g_1g_2-2)z^{-1/2}+(1+k_2k_3g_1g_2-k_4\cdot g_1\cdot g_2)z^{-1}+k_2k_3g_1g_2z^{-3/2}}{(1-z^{-1/2})^2}\right)Y. \tag{A.8}$$

If we set

$$5k_2k_3 = k_4 \tag{A.9}$$

interval delay is taken into account in the polyphase components. For instance, the signal to polyphase component $F_1(z)$ in Figure B.2 (a) has been delayed by a sampling interval. Similarly in Figure B.2 (b), the signal to polyphase component $F_i(z)$ ($i = 1, 2, 3$) has been delayed by i sampling interval(s). Therefore, the I and Q signals from the downconverter consisting of a polyphase CIC decimator have been aligned in time.

B. Wideband Digital Downconversion

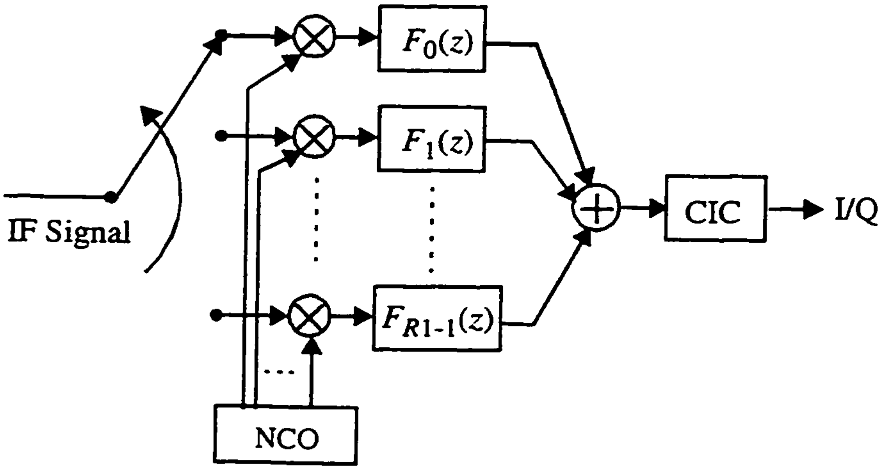


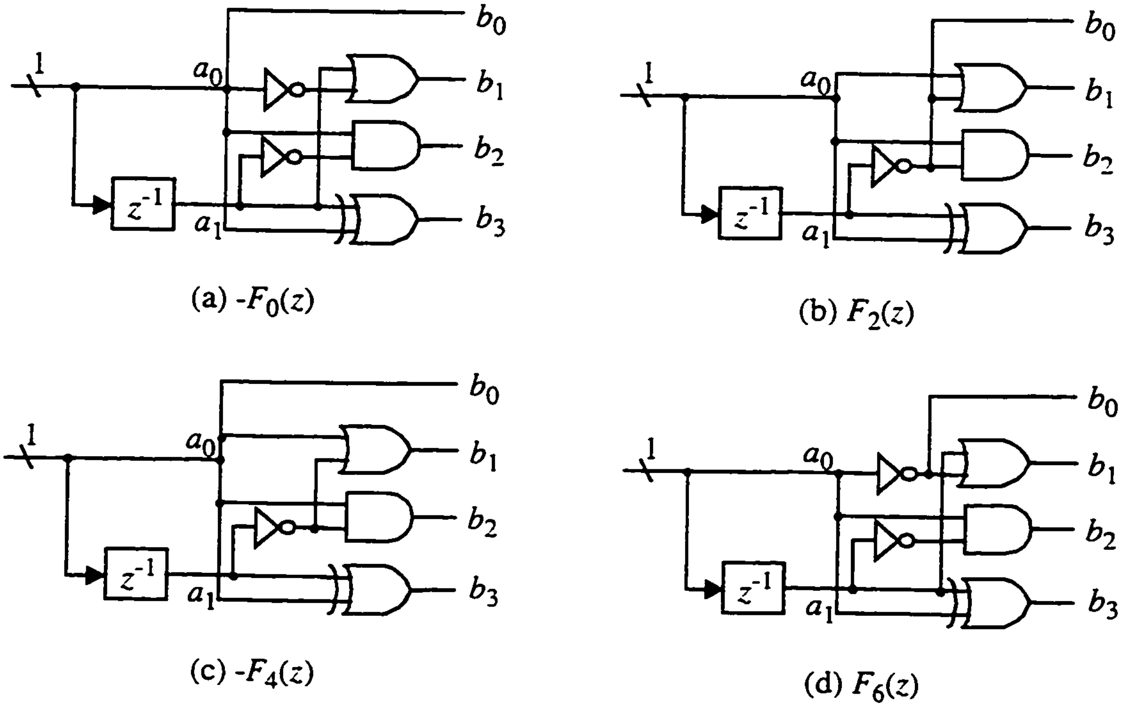
Figure B.3 A wideband DDC based on a polyphase CIC decimation filter

A wideband DDC is used to select a desired narrowband signal from a wideband IF signal digitized by a wideband ADC. A digital mixer is necessary in a DDC. In the circuit design, a multiplier is used to implement a mixer. Since the sampling rate is high, it is advantageous to reduce the processing rate of a multiplier by using several low-rate multipliers. This is required either by low power consumption or the high speed hardware realization such as in an FPGA implementation. The polyphase CIC decimation filter can be used to achieve this, and the derived architecture is shown in Figure B.3, where the digital input IF signal is split into R_1 phases. The polyphase technique can be also used in the NCO [Tan95] which is split into R_1 phases. Hence, the operating rate of the multipliers is reduced by a factor of R_1. The use of a polyphase NCO together with a polyphase CIC decimator has advantages in high speed operation or low power consumption.

B.2 Circuits Design for 100 MHz DDC FPGA

A. Polyphase Component Circuits

The synthesized circuits for polyphase components $-F_i(z)$, $i = 0, 2, ..., 7$, are shown in Figure B.4 and Figure B.5. The circuits are obtained following the design procedures shown Section 4.4.1.



(a) $-F_0(z)$

(b) $F_2(z)$

(c) $-F_4(z)$

(d) $F_6(z)$

Figure B.4 Circuits of polyphase components for the Q channel

B. A 3-Stage CIC Decimator

A CIC decimator consists of 3 accumulators and 3 differentiators. Since the accumulators operate at the high rate, a special design is required.

A 2-bit accumulator is shown in Figure B.6, where *CI* and *IN* are 1-bit carry-in and 2-bit input respectively. *CO* and *OUT* are 1-bit carry-out and 2-bit output respectively. A 14-bit pipelined accumulator consisting of seven 2-bit accumulators is depicted in Figure B.7, where a delayed *CO* from the previous 2-bit ACC is the input to the next *CI*. To align the output, the seven 2-bit inputs require appropriate delays. A cascaded 3-stage pipelined 14-bit accumulator is shown in Figure B.8.

The design for cascaded differentiators in a CIC decimation filter is very simple since the incoming rate is further reduced by a factor of 8 and is 1.5625 MHz. The adder used here is taken from the circuit library in a design tool and is a carry-save adder [West94].

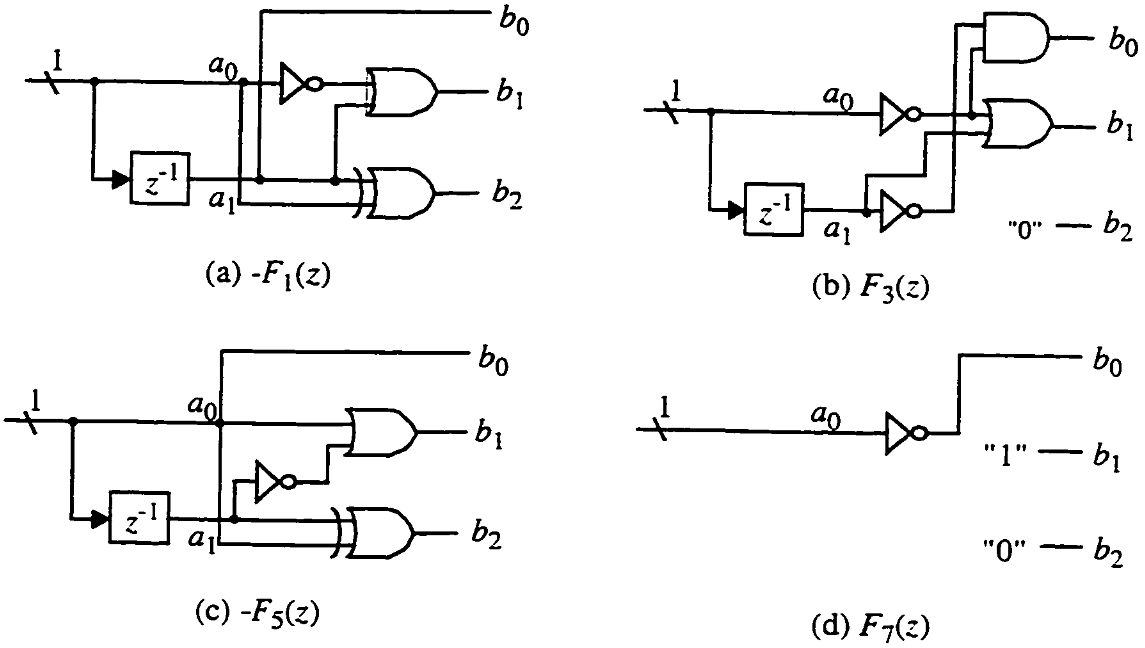


Figure B.5 Circuits of polyphase components for the I channel

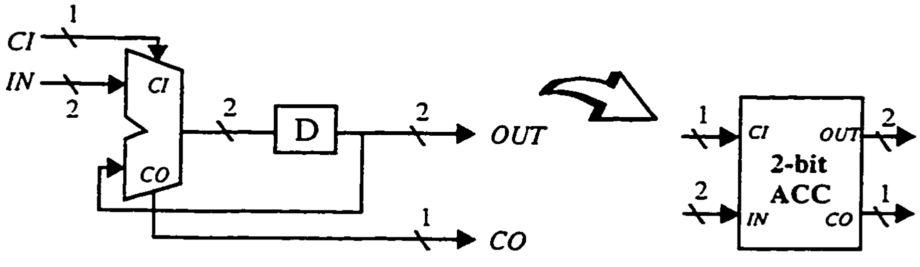


Figure B.6 A 2-bit accumulator and its symbol

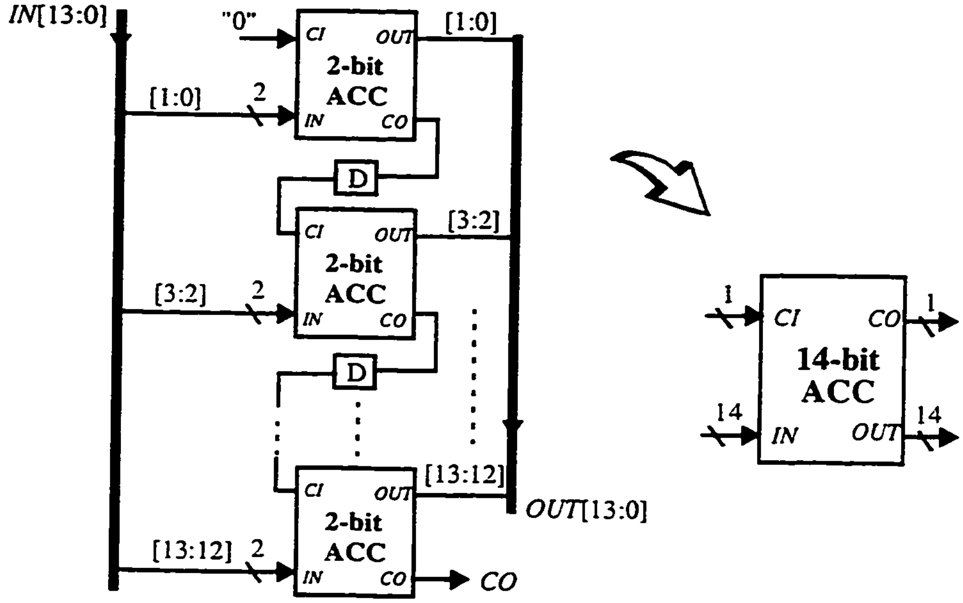


Figure B.7 A 14-bit pipelined accumulator and its symbol

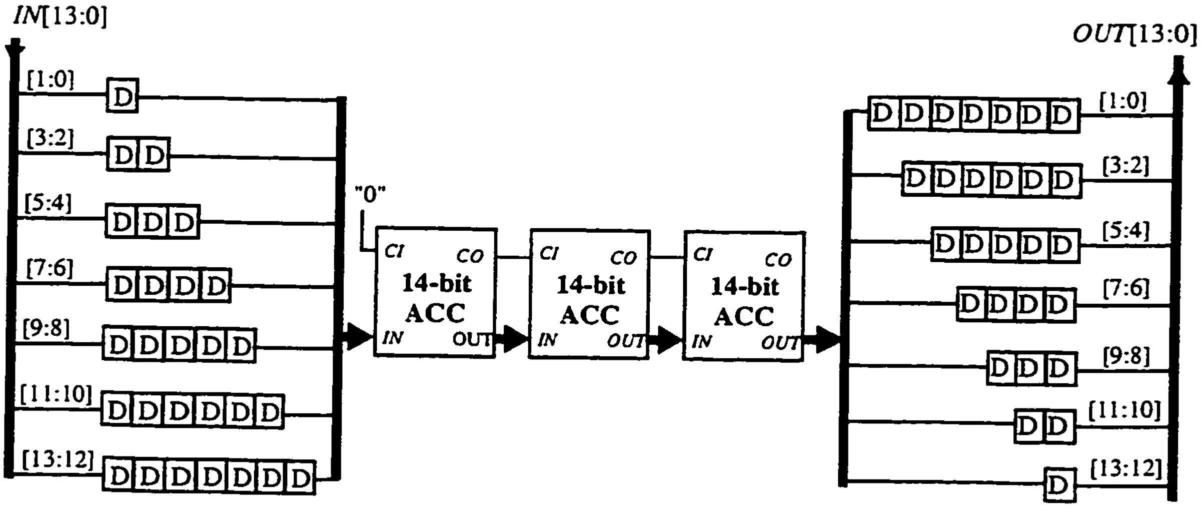


Figure B.8 A cascaded 3-stage, 14-bit pipelined accumulator

C. FPGA Schematic Diagram

The circuit to generate a slow clock *SLWCLK* of 1.5625 MHz is also shown in Figure 4.13, where an inverter is inserted between *CLK* and the divider. This is done to make the rising edge of *SLWCLK* correspond to the falling edge of *CLK*. This is to guarantee the setup and hold time when *SLWCLK* is used to downsample the signals, as shown in the figure. With this *SLWCLK*, the data is sampled in the middle of a high speed pulse.

The schematic diagram for the whole polyphase DDC is shown in Figure B.9. One can see that there are eight data input pads and one clock input pad on the left edge. The triangle symbols next to them are buffers. Eight D flip-flops are used to synchronize the input signals. The 8-phase signals then go to eight polyphase components. One can see there are 3 adders in each channel. The rest of the circuits implements a 3-stage CIC decimator.

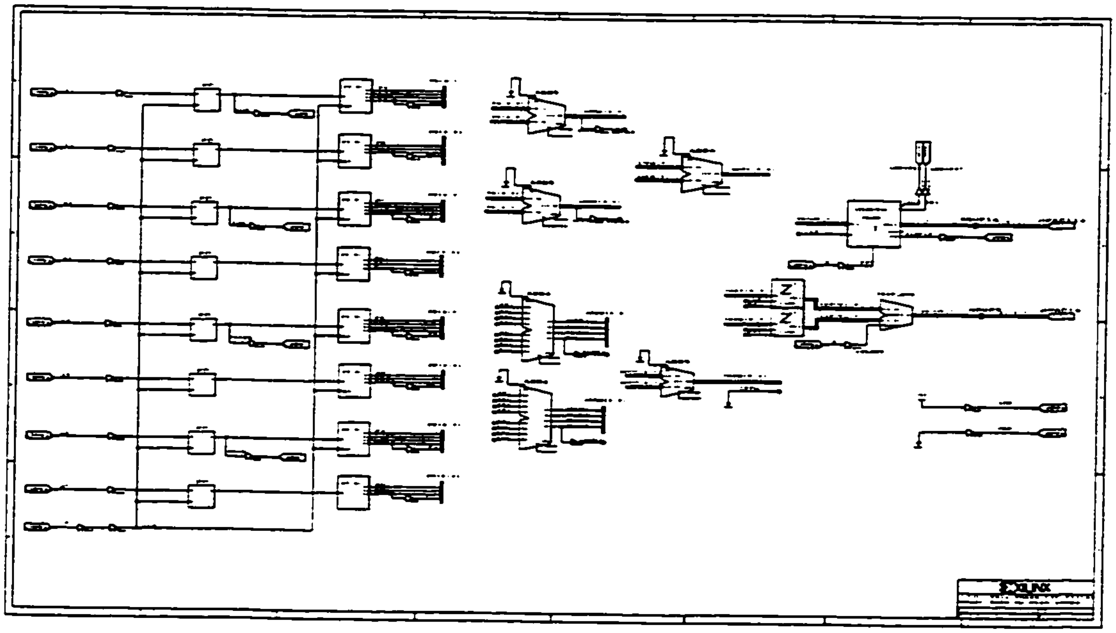


Figure B.9 Schematic diagram for the proposed polyphase DDC

D. Test Setup

The test setup for the DDC chip is shown in Figure B.10. The signal source is a data generator which outputs data (n_1 = 8 bits) to the circuit under test (CUT) at a rate of $Clk1$ = 12.5 MHz. The clock rate $Clk1 = CLK$ in Figure B.10. The data generator HP8180A has a memory of 8 kbits. The data was generated by SPW. In the SPW model, a fourth-order bandpass ΔΣ modulator was used to create a 1-bit ΔΣ modulated bit stream. This 1-bit data was then downloaded to the data generator in such a way that eight parallel bits representing 8 phases are sent out simultaneously. The download is controlled by a GPIB interface via a PC. This method of generating 8-phase data from the data generator removes the need for a demultiplexer, and allows the XC3159A chip to accept a sampling rate *Clk*1. The chip output data (n_2 = 14 bits) together with a clock of *Clk*2 = 1.5625 MHz are sent to an HP16500A logic analyzer. This logic analyzer can store up to 4M bytes of data. The clock rate *Clk*2 = *SLWCLK* in Figure B.10. The output data is saved and then transferred to a personal computer via GPIB ports.

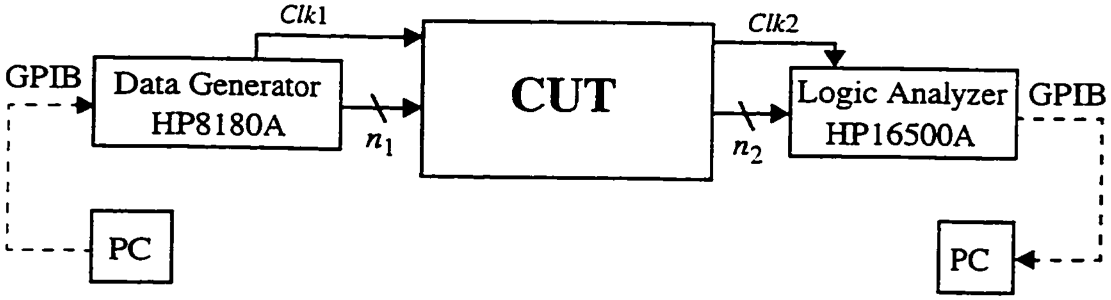


Figure B.10 Test setup for the CUT

The data generator HP8180A can only store 1024 bits for each of eight ports. The output of each port represents one phase of the signal. There are 8192 bits for the eight ports. This allows us to send out 16 different symbols at T_{sy} = 512 *T*. The symbol data repeats after 16 symbols and are designed as follows:

I data = {-1, +1, -1, +1, -1, -1, +1, +1, +1, -1, -1, -1, -1, -1,- 1, +1}.

Q data = {-1, -1, -1, -1, +1, +1, +1, +1, +1, -1, +1, -1, -1, +1,-1, +1}.

that presented in Section B.2.

C.3 A Halfband Filter

A halfband filter is used after the timing-phase-adjustable CIC decimation filter to further downsample the sampling rate by 2. The input sampling rate to the filter is 1.28 MHz and the output is therefore 640 KHz. Nine such simple filters are listed in [Good77], where filter F4 is chosen to fulfil this task. This filter is a 7-tap FIR filter. The coefficients before and after normalization to 32 are as listed in Table C.1. The normalized values are used for implementation. Note that only half the coefficients of this halfband are shown due to its symmetry. The frequency response of the halfband filter is shown in Figure C.2(a), where aliasing attenuation is around 37 dB. The circuit for the halfband filter is depicted in Figure C.3(a) where a polyphase architecture is used. The input data is demultiplexed into two lower-rate streams (the sampling rate is divided by 2). The implementation of a coefficient in a shift-and-addition manner is shown in Figure C.3(b).

Table C.1 Coefficients in halfband and RRC filters

	Halfband Filter		RRC Filter	
Coeffs.	Values	Normalized by 32	Values	Normalized by 32
h(0)	-3	$-2^{-4} - 2^{-5}$	0	0
h(1)	0	0	0	0
h(2)	19	$2^{-1} + 2^{-4} + 2^{-5}$	0	0
h(3)	32	2^{0}	-1	-2^{-5}
h(4)			-2	-2^{-4}
h(5)			-2	-2^{-4}
h(6)			4	2^{-3}
h(7)			17	$2^{-1} + 2^{-5}$
h(8)			30	$2^{0} - 2^{-4}$
h(9)			36	$2^{0} + 2^{-3}$

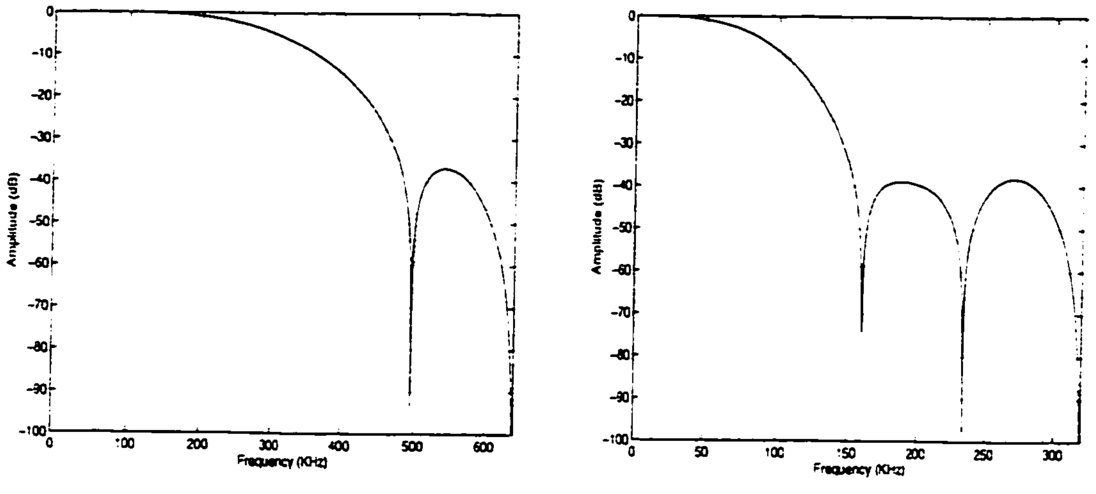
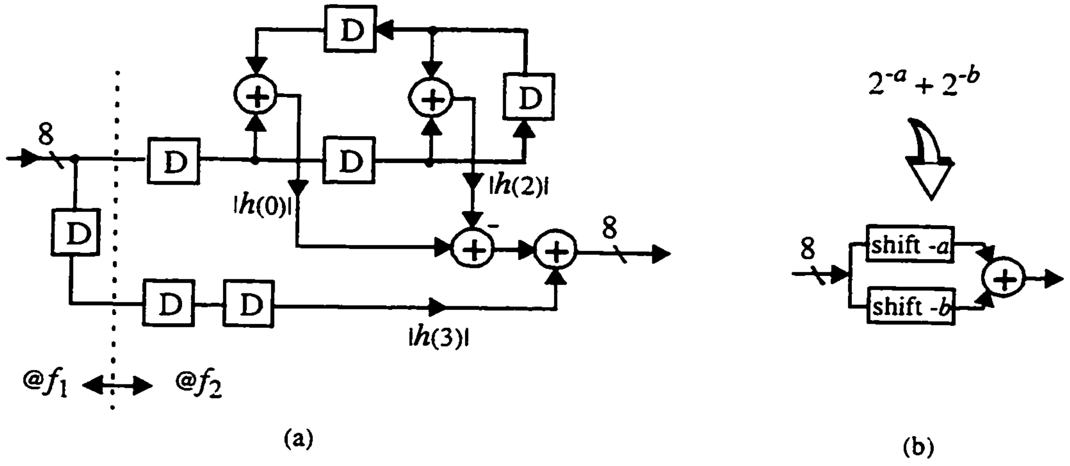


Figure C.2 Frequency responses of (a) the halfband filter and (b) the RRC filter



(a)

(b)

Figure C.3 (a) The halfband filter and (b) its coefficient implementation

C.4 A Root Raised Cosine Filter

The data filter used in the timing recovery circuit is a root raised cosine (RRC) filter. The filter is a receiver matched filter. Together with an identical filter on the transmitter side, a Nyquist inter-symbol-interference (ISI) free filter is achieved. This filter does two jobs: (1) to downsample the input sampling rate of 640 KHz to an output rate of 320 KHz (twice the symbol rate), and (2) to shape the signal to meet the Nyquist ISI-free criterion.

A 19-tap RRC filter is designed. The rolloff factor is 0.5. The coefficients before and after normalization to 32 are as listed in Table C.1. The normalized values are used for implementation. Note that only half the coefficients are shown due to their symmetry. The frequency response of the halfband filter is shown in Figure C.3(b), where the aliasing attenuation is around 38 dB. The circuit implementation is similar to that in Figure C.2 and therefore is not shown here.

C.5 A Timing Error Detector

The algorithm of the timing error detection for a QPSK signal is from [Gard86]. For a BPSK signal, the algorithm becomes, $e(k) = y\left(k-\frac{1}{2}\right)(y_d(k) - y_d(k-1))$, where $y(.)$ is the sampled baseband signal and $y_d(.)$ is a hard-decision based on y (either 1 or 0). The circuit realizing the timing error is shown in Figure C.4.

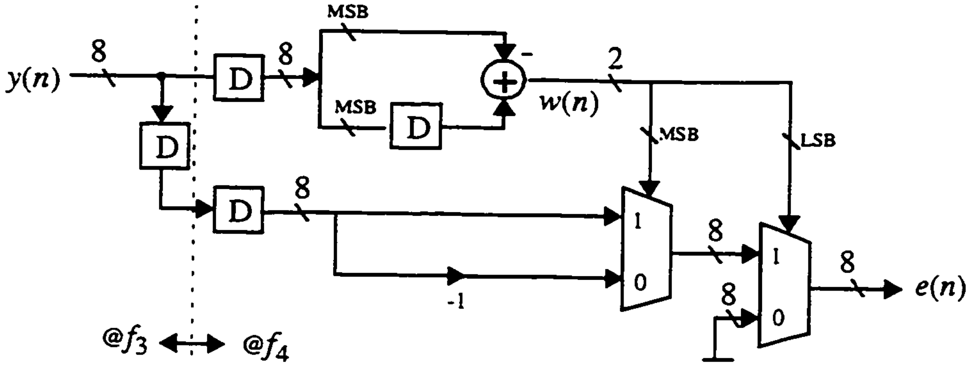


Figure C.4 Circuit for the timing error detector

Since y is in two's complement, we have the most significant bit (MSB) of y:

$MSB(y(k)) = 0$, if $y(k) >= 0$

$MSB(y(k)) = 1$, if $y(k) < 0$

A signal $w(k)$ is obtained by substracting $MSB(y(k\text{-}1))$ from $MSB(y(k))$, as shown in Figure C.4, namely, $w(k) = MSB(y(k\text{-}1)) - MSB(y(k))$. One can have,

$w(k) = 0$, if $MSB(y(k\text{-}1)) = MSB(y(k))$, that is $y_d(k) - y_d(k\text{-}1) = 0$.

$w(k) = 1$, if $MSB(y(k)) = 0$ and $MSB(y(k\text{-}1)) = 1$, that is $y_d(k) - y_d(k\text{-}1) = -1$.

$w(k) = -1$, if $MSB(y(k)) = 1$ and $MSB(y(k\text{-}1)) = 0$, that is $y_d(k) - y_d(k\text{-}1) = 1$.

Hence in two's complement, we have

$w(k) = 00$, $e(k) = 0$,

$w(k) = 01$, $e(k) = -\,y(k - 1/2)$,

$w(k) = 11$, $e(k) = +\,y(k - 1/2)$.

C.6 A Loop Filter

The loop filter with quantized coefficients is given by,

$$H(z) = \frac{1}{8}\frac{1+z^{-1}}{1-0.875z^{-1}}. \tag{C.1}$$

This filter has not been optimized for the timing loop. The design criterion is to compromise between performance and complexity. The loop filter is shown in Figure C.5.

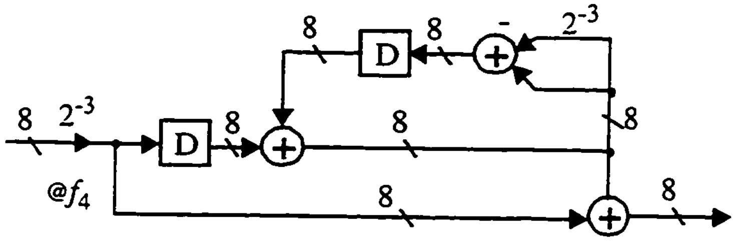


Figure C.5 Circuit for the loop filter

C.7 A Comparator

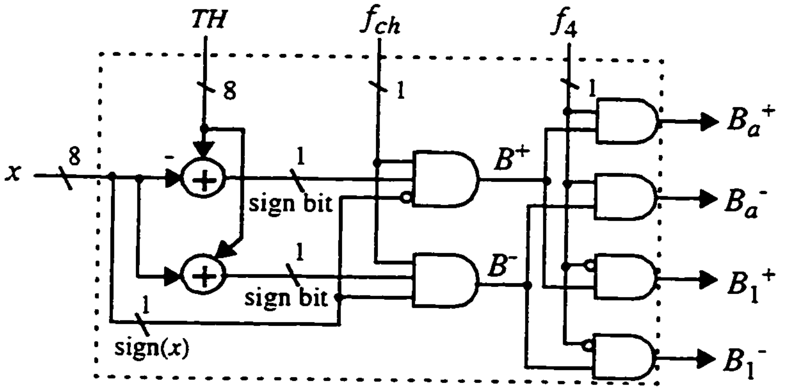


Figure C.6 Circuit for the comparator

The comparator circuit is shown in Figure C.6. The outputs of the comparator are used to drive the 7/8/9 counter.The comparator is used to compare the input data x with the threshold value $TH > 0$ according to the following rules:

C.9 FPGA Schematic Diagram

The timing recovery circuit using the decimation method was implemented with Xilinx XC3159A. The schematic diagram for the whole circuit is shown in Figure C.8.

As shown in the figure, the front-end is a 4-phase DDC which only takes two phases of the input signal (the first and the third) for the BPSK signal. Following two clock shaping D flip-flops are two polyphase components ($-F_1(z)$ and $F_3(z)$). Those two-phase signals are added by a 4-bit adder. Under the adder are two 7/8/9 variable counters, below which are a cascaded integrator and a cascaded dual-differentiator. One can see that the multiplexer follows the two differentiators. On the bottom, there are the halfband filter, root raised cosine filter, timing error detector, loop filter and comparator.

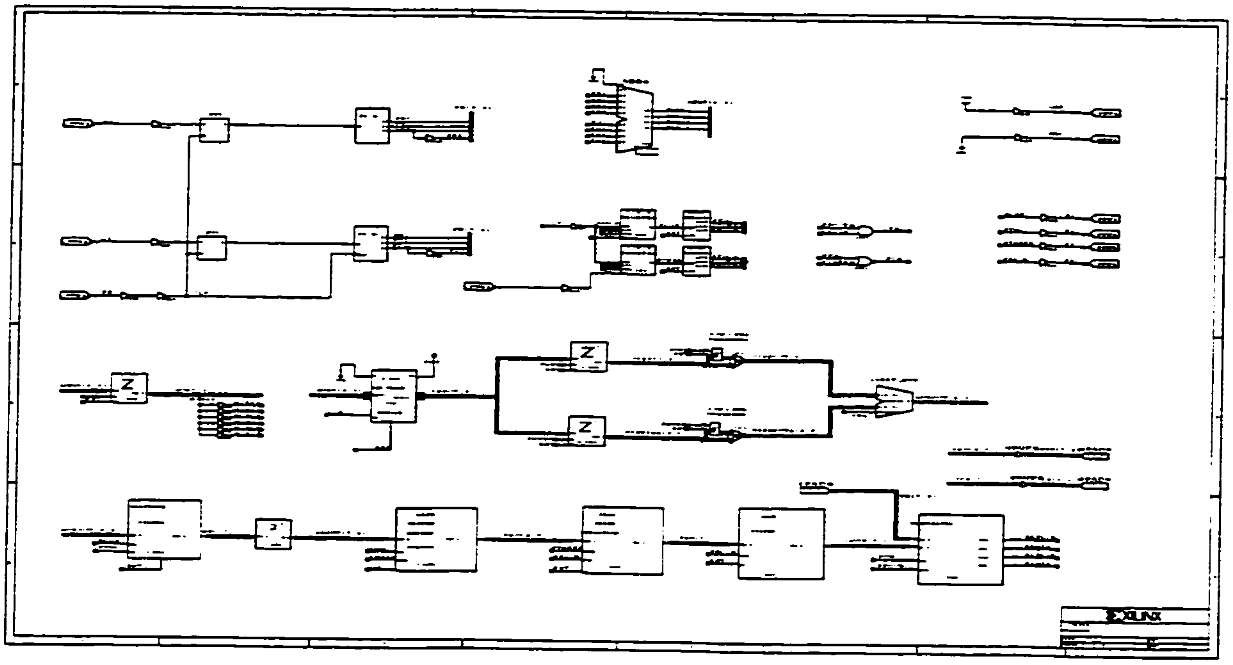


Figure C.8 Schematic diagram for the proposed timing recovery Circuit

C.10 Test Setup

The test setup for the symbol timing recovery chip is shown in Figure B.10. The signal source is a data generator which outputs data (n_1 = 4 bits) to the CUT at a rate of $Clk1$ =

10.24 MHz. The input clock rate $Clk1 = f_{s0}$ in Figure 5.22. The data was generated by SPW. The circuit shown in Figure 5.17 together with a second-order bandpass $\Delta\Sigma$ modulator was used to create a 1-bit $\Delta\Sigma$ modulated bit stream. This 1-bit data was then downloaded to the data generator in a way that four parallel bits representing four phases are sent out simultaneously. The output data ($n_2 = 8$ bits) clocked at $Clk2 = 320$ KHz was sent to a logic analyzer. The mean value of the output rate $Clk2 = f_3$ in Figure 5.22 is twice the symbol rate f_{sy}. The output data was saved and then transferred to a personal computer via a GPIB port.

The data generator can store 1024 bits for each of four ports. The output of each port represents one phase of the signal. There are 4096 bits for the four ports. This allows us to send out 16 different symbols at $T_{sy} = 512\ T$. The symbol data repeats after 16 symbols and is designed as follows:

16 symbols = {+1, -1, +1, -1, -1, -1, +1, -1, -1, +1, +1, +1, -1, +1, +1, -1}.

Appendix D
Adjustable-Delay Re-Timing CIC Decimators

As mentioned earlier in Section 5.1, another way for re-timing is to use an adjustable-delay decimator with fixed re-sampling clocks. In this method, the re-sampling phase is adjusted by changing signal path delays instead of the re-sampling clock. In this appendix, a brief description is given for such a decimator. Further investigation is required.

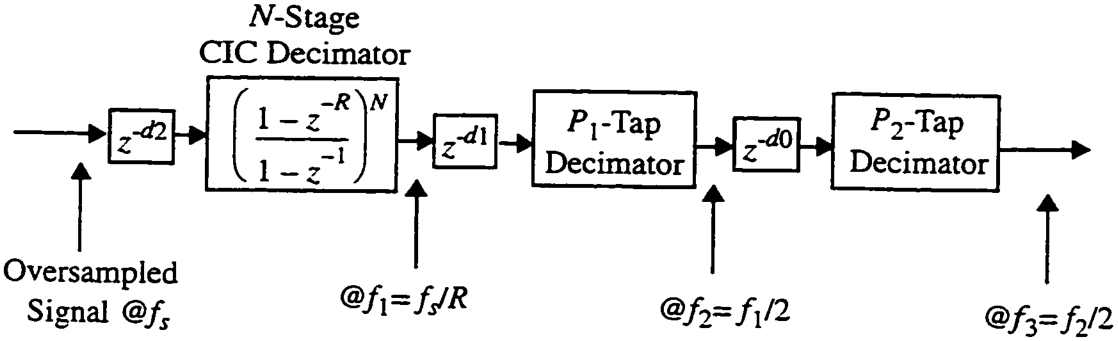


Figure D.1 An adjustable-delay CIC decimator

An adjustable-delay decimator based on a CIC decimator is shown in Figure D.1. This decimator decimates an oversampled signal and meanwhile adjusts the signal delay. Note that all the clocks are fixed while in the method discussed in Chapter 5 the re-sampling clock is varied. In this method, the signal path delays are adjusted to realize the timing phase adjustment. The delay adjustment is realized by controlling integer delays d_0, d_1, and d_2 before each decimator.

An N-stage CIC decimator downsamples the input signal by a fixed downconversion factor R, that is, $f_1 = f_s/R$. The rate at P_1-tap decimator output is $f_2 = f_1/2$. The rate at P_2-tap decimator output is half the rate of f_2. The nominal value of f_3 is designed to be twice the nominal value of f_{sy}. In the actual case, they are different due to the drift between the transmitter and local receiver clocks. The signal delay that can be adjusted is,

References

[Abid95] A.A. Abidi, "Low-power radio-frequency ICs for portable communications," *Proc. of the IEEE*, vol. 83, no. 4, pp. 544-569, Apr. 1995.

[Abou94] T. Aboulnasr, et al, "Characterization of a symbol rate timing recovery technique for a 2B1Q digital receiver," *IEEE Trans. on Communications*, vol. 42, no. 2/3/4, pp. 1409-1414, Feb./Mar./Apr. 1994.

[Arda87] S.H. Ardalan and J.J. Paulos, "An analysis of nonlinear behavior in delta-sigma modulators," *IEEE Trans. Circuits and Systems*, vol. CAS-34, pp. 593-603, June 1987.

[Asch89] G. Ascheid, et al, "An all digital receiver architecture for bandwidth efficient transmission at high data rates," *IEEE Trans. on Communications*, vol. 37, no. 8, pp. 804-813, Aug. 1989.

[Aziz93] P. Aziz et al, "Multiband sigma-delta modulation," *IEE Electronics Letters*, pp. 760-762, Apr. 29, 1993.

[Aziz96] P.M. Aziz, et al, "An overview of sigma-delta converters," *IEEE Signal Processing Magazine*, pp. 61-84, Jan. 1996.

[Bain95] R. Baines, "The DSP bottleneck," *IEEE Communications Magazine*, pp. 46-54, May 1995.

[Baza97] S. Bazarjani, M. Snelgrove, et al, "1 V mixed-signal circuits in a 0.5 μm CMOS technology,"*1997 IEEE Int. Symp. on Circuits and Systems*, Hong Kong, June 1997.

[Bear94] R. D. Beards and M.A. Copeland, "An oversampling delta-sigma frequency discriminator," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signa Processing*, vol. 41, no. 1, pp. 26-32, Jan. 1994.

[Bose88] B.E. Boser and B.A. Wooley, "The design of sigma-delta modulation analog-to-digital converters," *IEEE J. of Solid-State Circuits*, vol. 23, pp. 1298-1308, Dec. 1988.

[Bran94] B. Brandt and B. Wooley, "A low-power area-efficient filter for decimation and interpolation," *IEEE J. of Solid-State Circuits*, pp. 679-687, June 1994.

[Brod92] R.W. Brodersen, A.P. Chandrakasan and S. Sheng, "Technologies for personal communications," *Proc. 1992 VLSI Circuits Symp.*, pp. 5-9, 1992.

[Brow95] C. Brown, "RF research takes two paths," *Electronic Engineering Times*, pp. 35, 38, Sept. 4, 1995.

[Burm96] T. Burmas, et al, "A second-order double-sampled delta-sigma modulator using additive-error switching," *IEEE J. of Solid-State Circuits*, vol. 31, no. 3, pp. 284-292, Mar. 1996.

[Cand86] J.C. Candy, "Decimation for sigma delta modulation," *IEEE Trans. on Communications*, vol. COM-34, pp. 72-76, Jan. 1986.

[Cand92] J.C. Candy and G.C. Temes, *Oversampling Delta-Sigma Data Converters: Theory, Design and Simulation*. IEEE Press, New York, 1992.

[Chan95] A.P. Chandrakasan and R.W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. of the IEEE*, vol. 83, no. 4, pp. 498-523, April 1995.

[Ches94] D. Chester and G. Phillips, "Use DSP filter concepts in IF system design," *Electronic Design*, pp. 80-143, July 11, 1994.

[Cheu91] P.Y.K. Cheung and E.S.K. See, "A comparison of decimation filter architectures for sigma-delta A/D converters," *Proc. of the 1991 IEEE Int. Symp. on Circuits and Systems*, pp. 1637-1640, May 1991.

[Choi80] T.C. Choi and R.W. Brodersen, "Considerations for high-frequency switched-capacitor ladder filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-27, no. 6, pp. 545-552, June 1980.

[Chu84] S. Chu and C.S. Burrus, "Multirate filter designs using comb filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-31, no. 11, pp. 913-924, Nov. 1984.

[Cons83] V. Considine, "Digital complex sampling," *IEE Electronics Letter*, vol. 19, no. 16, pp. 608-609, 4th Aug. 1983.

[Coy92] R.J. Coy, et al, "HF-based radio receiver design based on digital signal processing," *Electronics and Communication Engineering Journal*, pp. 82-90, April 1992.

[Cowl94] W. G. Cowley and L.P. Sabel, "The performance of two symbol timing recovery algorithms for PSK demodulators," *IEEE Trans. on Communications*, vol. 42, pp. 2345-2355, June 1994.

[Croc83] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[Erup93] L. Erup, F.M. Gardner and R.A. Harris, "Interpolation in digital modem — Part II: Implementation and performance," *IEEE Trans. on Communications*, vol. 41, no. 6, pp. 998-1008, June 1993.

[Fran81] L.E. Franks, "Synchronization subsystems: analysis and design," in *Digital Communications: Satellite/Earth Station Engineering*, edited by K. Feher, pp. 294-327, 1981.

[Galt95] I. Galton and H. Jensen, "Delta-sigma modulator based A/D conversion without oversampling," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 12, pp. 773-784, Dec. 1995.

[Gao97] W. Gao and M. Snelgrove, "A 950 MHz second-order integrated LC bandpass ΔΣ modulator," *1997 Symposium on VLSI Circuits*.

[Gard79] F.M. Gardner, *Phaselock Techniques*. 2nd edition, New York: Wiley, 1979

[Gard86] F.M. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers," *IEEE Trans. on Communications*, vol. 34, pp. 423-429, May 1986.

[Gard93] F.M. Gardner, "Interpolation in digital modem — Part I: Fundamentals," *IEEE Trans. on Communication*, vol. 41, no. 3, pp. 502-508, Mar. 1993.

[Good77] D.J. Goodman and M.J. Carey, "Nine digital filters for decimation and interpolation," *IEEE Trans. on Acoustics, Speech, Signal Processing*, vol. ASSP-25, no. 2, pp. 121-126, Apr. 1977.

[Gott94] A. Gottscheber, et al, "Combined interpolator filter for timing recovery in a fully digital demodulator," *Proc. 1994 IEEE Int. Conf. on Communications*, pp. 1467-1471, 1994.

[Greg86] R. Gregorian G.C. Temes, *Analog MOS Integrated Circuits for Signal Processing*. New York: Wiley, 1986.

[Gros91] R. Groshong and S. Ruscak, "Undersampling techniques simplify digital radio," *Electronic Design*, pp. 67-78, May 23, 1991.

[Haou87] A. Haoui, H.-H. Lu and D. Hedberg, "An all-digital timing recovery scheme

[Muel76] K.H. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. on Communications*, vol. 24, no. 5, pp. 516-531, May 1976.

[Nors97] S.R. Norsworthy, R. Schreier and G.C. Temes, *Delta-Sigma Data Converters*. IEEE Press, New York, 1997.

[Nutt94] A.H. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 84-91, Feb. 1986.

[Ong97] A.K. Ong and B.A. Wooley, "A two-path bandpass ΣΔ modulator for digital IF extraction at 20 MHz,"*Digest of 1997 IEEE International Solid-State Circuit Conference*, pp. 212-213, 1997.

[Padg95] J.E. Padgett, C.G. Gunther and T. Hattori, "Overview of wireless personal communications," *IEEE Communications Magazine*, pp. 28-41, Jan. 1995.

[Park86] C.S. Park and R. Schaumann, "A high-frequency CMOS linear transconductance elements," *IEEE Trans. on Circuits and Systems*, vol. 33, pp. 1132-1138, Nov. 1986.

[Pell92] L. E. Pellon, "A double Nyquist digital product detector for quadrature sampling," *IEEE Trans. on Signal Processing*, vol. 40, no. 3, pp. 1670-1681, July 1992.

[Pokl92] J.J. Poklemba and F.R. Faris, "A digitally implemented modem: Theory and emulation results," *Comsat Technical Review*, vol. 22, no. 1, pp. 149-159, Spring 1992.

[Prid79] R.G. Pridham and R.A. Mucci, "Digital interpolation beamforming for low-pass and bandpass signals," *Proc. of the IEEE*, vol. 67, no. 6, pp. 904-919, June 1979.

[Rade84] C.M. Rader, "A simple method for sampling in-phase and quadrature components," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 20, no. 6, pp. 821-824, Nov. 1984.

[Rabi97] S. Rabii and B.A.Wooley, "A 1.8-V digital-audio sigma-delta modulator in 0.8-μm CMOS,"*IEEE J. of Solid-State Circuits*, vol. 32, no. 6, pp. 783-795, June 1997.

[Ragh97] G. Raghavan, et al, "A bandpass ΣΔ modulator with 92 dB SNR and center

[Sing95] F.W. Singor and W.M. Snelgrove, "Switched-capacitor bandpass delta sigma A/D modulation at 10.7 MHz," *IEEE J. of Solid-State Circuits*, vol. 30, pp. 184-192, Mar. 1995.

[Skla88] B. Sklar, *Digital Communication: Fundamentals and Applications*. New Jersey: Prentice-Hall, 1988.

[Soll90] N.R. Sollenberger and J.C.-I. Chuang, "Low-overhead symbol timing and carrier recovery for TDMA portable radio systems," *IEEE Trans. on Communications*, vol. 38, no. 10, pp. 1866-1892, Oct. 1990.

[SPW] *SPW User's Manual*, Cadence Design Systems, 919 E. Hillsdale Blvd., Foster City, CA 94404, USA.

[Stou93] C.L. Stout and J. Doernberg, "10-Gb/s Silicon Bipolar 8:1 Multiplexer and 1:8 Demultiplexer," *IEEE J. of Solid-State Circuits*, vol. 28, no. 3, pp. 339-343, Mar. 1993.

[Suya92] K. Suyama and S.C. Fang, *Users' Manual for SWITCAP2*, Version 1.1, Columbia University, New York, 1992

[Swam97] A. Swaminathan, private communications, Jan. 1997.

[Tan95] L.K. Tan and H. Samueli, "A 200 MHz Quadrature digital synthesiser/mixer in 0.8 μm CMOS," *IEEE J. of Solid-State Circuits*, vol. 30, no. 3, pp. 193-200, Mar. 1995.

[Thie90] T.E. Thiel and G.J. Saulnier, "Simplified complex digital sampling demodulator," *IEE Electronic Letter*, vol. 29, no. 7, pp. 419-421, 29th Mar. 1990.

[Thur95] A.M. Thurston, "Sigma-delta IF A-D converters for digital radios," *GEC J. of Research*, vol. 12, no. 2, pp. 76-85, 1995.

[Than97] C.K. Thanh, et al, "A second-order double-sampled delta-sigma modulator using individual-level averaging," *IEEE J. of Solid-State Circuits*, vol. 32, no. 8, pp. 1269-1273, Aug. 1997.

[Uchi88] K. Uchimura, et al, "Oversampling A-to-D and D-to-A converters with multi-stage noise shaping modulators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1899-1905, Dec. 1988

[Vaid93] P.P. Vaidyanathan, *Multirate System and Filter Banks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.