INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Performance Evaluation of 12 and 14-Bit Converter Technology for Software Radio Applications

by

Borislav St. Todorov

B. Eng. (Electrical Engineering), Technical University, Varna, Bulgaria

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

Master of Engineering in Electrical Engineering

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario K1S 5B6, Canada

May 16th, 2000

© 2000, Borislav St. Todorov

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

Our file *Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-57742-2

Canada

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

Performance Evaluation of 12 and 14-Bit Converter Technology for Software Radio Applications

Submitted by Borislav Todorov, B.Eng.
in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering

Professor S. A. Mahmoud

Dean, Faculty of Engineering and Design
Thesis supervisor

Professor R. A. Goubran

Chair, Department of Systems and Computer Engineering

ABSTRACT

For many years, radio design has relied entirely on analog circuitry to tune, downconvert and even demodulate the desired signal. Flexible only to a small degree and often with a cost many-fold, compared to alternative digital designs, this has been acknowledged from the early days of digital research to be the main drawback in commercial applications development.

In the past few years, though, the digital radio and digital receiver have become increasingly popular due to the advances in semiconductor components technology. High performance A/D converters, ultra-fast data transfer interfaces and cost-effective, powerful DSPs are now readily available to facilitate an entirely digital radio design.

This thesis focuses on investigating advanced converter technologies with respect to applications in the area of wireless communications and their ability to become an integrated part of the digital radio design. For this purpose, experimental efforts concentrate on measuring and evaluating many key performance parameters of interest to the Communications Industry such as Signal-to-Noise Ratio (SNR), Spurious-Free Dynamic Range (SFDR), Analog Input Bandwidth, Channels Cross-Talk Ratio etc. Performance evaluation is based on measurements with currently available digital radio converter technologies from reputable companies like Inter-active Circuits and Systems Ltd. and ©Analog Devices.

In the final chapters, the experimental results are related to the receiver requirements of the Wireless Communications Industry. To be more specific, several wireless standards are looked at in scope and conclusions are made with respect to the experimentally derived converter performance parameters. The analysis is based on two types of commonly encountered receiver shortcomings - dynamic range and noise limitations. It is acknowledged that the ADC spurious free dynamic range limitations are by far the most challenging in the realization of today's software radio receivers. In this context, several novel architectures aimed at reducing the SFDR requirements are discussed and evaluated. Finally, the Thesis concludes with overall conclusions and proposals for future research in this field.

ACKNOWLEDGMENTS

I would first like to offer my appreciation to Professor Samy Mahmoud, my thesis supervisor, for making time in a busy schedule to guide me towards our final goal with professional advice, personal interest and financial support. I greatly appreciate his dedication to help me complete this research.

Special thanks to the entire staff at ICS Ltd., who provided the much needed hardware support and great working atmosphere.

I also appreciate the help I received in every facet of this project from Naren Mehta and Dave Sword of the Department of Systems and Computer Engineering.

Needless to say, I am grateful to my parents and my dear wife for giving me emotional support and encouragement. There are no words to describe what you mean to me. All of my accomplishments, past, present, and future, are a direct result of your guidance, your support, and your love.

Contents

Abstract	iii
Acknowledgments	iv
Table of Content	v
List of Figures	vii
List of Tables	ix
List of Symbols	x
List of Abbreviations	xii
Chapter 1. Introduction	1
1.1 Motivation	1
1.2 Approach	3
1.3 Organization of this Report	4
Chapter 2. Background	6
2.1 Software Radios	6
2.2 ADC Essentials	8
2.3 The ICS-650 and ICS-152 Acquisition boards	15
Chapter 3. Experimental Setup	18
3.1 Defining the Experiments	18
3.2 Hardware design and considerations	24
3.3 Experimental setup evaluation	27
Chapter 4. Details of Software Modules	32
4.1 Software Framework	32
4.2 SFDR and SNR Software description	42
4.3 Matlab specific and other modules description	43
Chapter 5. Experimental Results	46
5.1 SFDR-related	46
5.2 SNR-related	58
5.3 Input Bandwidth-related	60
5.4 Channels Crosstalk-related	61

Chapter 6. Performance Evaluation	63
6.1 Sources of impairments and results accuracy	63
6.2 ICS-650 and ICS-152 comparison	68
Chapter 7. Relating the experimental results to today's Wireless	
Standards specifications	70
7.1 GSM support evaluation	70
7.2 CDMA support evaluation	82
7.3 Selected 3G standards support evaluation	86
Chapter 8. Wideband receiver dimensioning in a multistandard	
environment	92
8.1 The ADC challenge	92
8.2 Digital front-end architectures aimed at reducing the	
ADC requirements	94
Chapter 9. Conclusions and Future Work	103
9.1 Summary	103
9.2 Suggested areas for improvements	105
9.3 Conclusions from the experiments	106
9.4 Proposals for future research	107
List of References	109
A 1. ICS-650 Technical Specifications	111
A 2. HP8644A and HP3335A Specifications	112
A 3. Programming Code	113
1. Spurious Free Dynamic Range source code	113
2. Signal-to-Noise Ratio source code	120
3. Analog Input Bandwidth source code	121
4. Average Noise Floor source code	122

List of Figures

1-1	Common Multicarrier/Multichannel Radio Receiver Architecture	2
1-2	Digital-Design Software Radio	2
1-3	Experimental Testbed	4
1-4	Software Subsystem	4
2-1	Software Radio Architecture	6
2-2	Mathematical Model for Quantization Error Analysis	9
2-3	Block Diagram of 2nd order low pass SDM with linearized quantizer	11
2-4	Noise-shaping characteristics of SDM	13
2-5	ICS-650 Block Diagram	16
3-1	Error resulting from reference clock jitter	22
3-2	Error value depends on the frequency of the sampled signal	23
3-3	Experimental Setup	25
3-4	Sinusoidal to TTL-level Interface Buffer	26
3-5	ICS-650 Typical Noise Performance	29
3-6	ICS-650 Samples Averaged Noise Performance	29
3-7	AD6640 FFT-plot @ 2.2 MHz and 65 Msps Sampling Rate	30
3-8	ICS-650 performance with input signal	31
4-1	The Software Framework	32
4-2	Data Capture Module Architecture	33
4-3	Windows Host software Layout for ICS-650 SDK	33
4-4	Functional description of the FFT module	35
4-5	Harmonics Filtering Module Functional Architecture	36
4-6	Unfiltered Signal FFT-plot	37
4-7	Filtered Signal FFT-plot	37
4-8	Signal Characteristics	38
4-9	Aliasing Characteristics of Sampled Systems	39
4-10	Aliases Removal Module Algorithm	41
4-11	Interaction Between the SFDR Testing Software	42

5-1	FFT plots from measurements @ 60 Msps and PSD = -1dBFS	47
5-2	FFT plots from measurements @ 60 Msps and PSD = -10dBFS . . .	48
5-3	FFT plots from measurements @ 60 Msps and PSD = -21dBFS . . .	49
5-4	SFDR performance for 60 Msps and -1dBFS input signal	50
5-5	SFDR performance for 60 Msps and -10dBFS input signal	51
5-6	SFDR performance for 60 Msps and -20dBFS input signal	51
5-7	SFDR performance for 40 Msps	52
5-8	SFDR Performance with Amplitude @ 60 Msps	53
5-9	SFDR Performance with Amplitude @ 40 Msps	53
5-10	FFT plots @ 40 Msps and PLL bypassed. PSD = -1 dBFS	54
5-11	FFT plots @ 10 Msps and PLL bypassed. PSD = -1 dBFS	55
5-12	FFT plots from measurements @ 70 Msps with ICS-152	56
5-13	SFDR performance for 70 Msps and various signal amplitudes . . .	57
5-14	SNR performance for 60 Msps and various signal amplitudes	58
5-15	SNR performance for 70 Msps and various signal amplitudes	59
5-16	Analog Input Bandwidth for 60 Msps and various signal levels . . .	60
5-17	ICS-650 Channels Crosstalk performance at 60 Msps	61
5-18	ICS-152 Channels Crosstalk performance at 70 Msps	62
6-1	Pulse skew increases overall jitter in the system	65
6-2	Typical DAC transfer functions	66
7-1	Signal variety at the BS receiver	72
7-2	Common digital receiver architecture	73
7-3	CDMA Uplink Signal	82
7-4	IS-95 Open Loop power control range	83
7-5	UMTS Wideband carrier structure	89
7-6	An illustration of the bandwidth-dynamic range trade-off	90
8-1	Companded PCM with typical compression characteristic	95
8-2	Dynamic range suppression with largest signal prediction	96
8-3	Alternative front-end architecture employing subband processing	98
8-4	High/Low gain receiver architecture	99
8-5	Cell regions in the proposed High/Low gain architecture	100

List of Tables

1. Current ADC sampling rates specifications	14
2. ICS-650 and ICS-152 performance comparison	68
3. ICS-650 and ICS-152 noise limited sensitivity	80

List of Symbols

ΔE_b	Energy per bit difference
b	Number of converter bits of resolution
B	Bandwidth of interest
$e_q(n)$	Quantization error
E_b	Energy per bit
f	Frequency
$f_{carrier}$	Carrier frequency
f_{max}	Maximum frequency
f_N	Fundamental (signal) frequency
f_s	Sampling frequency
F_s	Sampling frequency
k	Boltzmann's constant
mV	millivolts
mW	miliwatts
N_0	Noise variance
P_a	Available power from the source
P_r	Power applied to a resistor
P_{AV}	Average power
$P_{NOISE_{RF}}$	RF noise power
P_{PN}	Phase noise power
P_{QN}	Quantization noise power
P'_S	Minimum required received RF signal power
P_{TN}	Thermal noise power
P_{TOT}	Total power
PG	Processing gain
R	Resistor value in Ohms
R_b	Bit rate
R_{EDGE}	Bit rate for EDGE system

R_{GSM}	Bit rate for GSM system
R_{TDMA}	Bit rate for TDMA system
SNR_{req}	Required signal-to-noise ratio
σ_x	Input signal variance
$t_{j,rms}$	Clock jitter value
T	Temperature in Kelvins
$x(n)$	Sampled discrete-time signal
$x_a(t)$	Analog input signal
$x_a(nT)$	Periodically sampled discrete-time signal
$x_q(n)$	Output quantized signal
V	Voltage
V_{max}^+	Largest positive voltage
V_{max}^-	Smallest negative voltage
V_n	Noise voltage
V_{rms}	Root mean square voltage
V_{FS}	Full-scale voltage
$V_{NOISE_{ADC}}$	ADC noise voltage
$V_{NOISE_{RF}}$	RF noise voltage
V_{pp}	Converter full scale range
V_{TOTAL}	Total rms voltage
W	Watts of power

List of Abbreviations

3G	Third Generation
3OP	Third-Order Products
A/D	Analog-to-Digital
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
Amp	Amplifier
AMPS	Advanced Mobile Phone System
API	Application Programmer's Interface
ASIC	Application-Specific Integrated Circuit
BCCh	Broadcast Channel
BF	Baseband Frequency
BPF	Band Pass Filter
BS	Base Station
BW	Bandwidth
CCCH	Common Control Channel
CDMA	Code Division Multiple Access
CH1	Channel 1
CH2	Channel 2
CIR	Carrier-to-Interferer Ratio
C/N	Carrier-to-Interference
dB	Decibels
dBc	Decibels, relative to carrier
dBm	Decibels, relative to 1 mW
dBFS	Decibels, relative to Full-Scale
D/A	Digital-to-Analog
DAC	Digital-to-Analog Converter
D-AMPS	Digital-Advanced Mobile Phone System
DC	Direct Current
DCS-1800	North American upband GSM standard
Demod.	Demodulator
DECT	Digital European Cordless Telephone
DFT	Discrete Fourier Transform
DNL	Differential Nonlinearity

DSP	Digital Signal Processing
E-GSM	Extended GSM
FFT	Fast Fourier Transform
FIFO	First-In First-Out
FPDP	Front Panel Data Port
FPGA	Field-Programmable Gate Array
GHz	Giga Herz
GPS	Global Positioning System
GSM	Groupe Speciale Mobile
Gen.	Generator
HP	Hewlett Packard
IC	Integrated Circuit
ICS	Interactive Circuits and Systems Ltd.
ICS-152	The 14-bit receiver board
ICS-650	The 12-bit receiver board
IF	Intermediate Frequency
IM	Intermodulation
INL	Integral Nonlinearity
IS-95	Wireless CDMA standard
ITU	International Telecommunication Union
KB	Kilobytes
kHz	Kilo Herz
L	Order of the Sigma-Delta Modulator
LNA	Low Noise Amplifier
LPF	Low Pass Filter
LSB	Least Significant Bit
Meg	Megabytes
MB	Megabytes
MHz	Mega Herz
MS	Mobile Station
MSPS	Mega Samples Per Second
NF	Noise Figure
NTF	Noise Transfer Function
OS	Operating System
PA	Power Amplifier
PC	Personal Computer

PCS	Personal Communication System
PCS-1900	European upband GSM standard
PCI	Peripheral Controller Interface
PLL	Phase Locked Loop
PSD	Power Spectral Density
rms	root mean square
RACH	Random Access Channel
RF	Radio Frequency
Rx	Receiver
RSL	Reliable Signal Level
SDCCH	Stand-alone Dedicated Control Channel
SDK	Software Development Kit
SDM	Sigma-Delta Modulators
SFDR	Spurious-Free Dynamic Range
S/N	Signal-to-Noise
SNR	Signal-to-Noise Ratio
SQNR	Signal-to-Quantization Noise Ratio
SRFFT	Split-Raddix FFT
STF	Signal Transfer Function
SWR	Software Radio
TDMA	Time Division Multiple Access
Tx	Transmitter
U	Oversampling Ratio
UMTS	Universal Mobile Telecommunications System
W-CDMA	Wideband CDMA
WPA	Wideband Power Amplifier

Chapter 1

Introduction

The current age of technological innovation has led to an explosion in the field of communications. Vastly improving standards and technologies have allowed users of wireless communication systems to have virtually unrestricted mobility around the world. However, current communications systems possess a number of drawbacks, including early obsolescence of devices, lack of flexibility, and incompatibilities between standards.

Digital-design devices, on the other hand, can greatly improve the flexibility of the system to support various standards by implementing the different accessing schemes and modulation methods in platform interchangeable software modules without the need of maintaining several different hardware sets. The ultimate goal in any approach considered so far is to move the sw/hw boundary as close to the front end antenna as physically possible. In many cases this is the A/D converter in the receive path and correspondingly the D/A converter in the transmit path.

Few questions naturally arise; What level of performance can we expect from these devices today, and is the currently available technology capable of implementing seamless and reliable digital radio operation?

The goal of this thesis is to provide answer to the above questions by performing various performance parameters measurements on a state of the art digital radio receiver as well as the necessary conclusions in relation to that.

1.1 Motivation

For years, radio designs have focused on the use of special-purpose hardware to perform all of the radio functionality from the RF (Radio Frequency) down to the BF (Baseband Frequency) processing. With the recent boom in Wireless Communications this has become highly inefficient. Today's mobile communications systems employ various multichannel and/or multi-carrier accessing schemes. With traditional architecture, this means separate hardware sets for each channel/carrier processing link as depicted in Figure 1-1.

The cost of such system is proportional to the number of BF hardware sets needed to cover the proposed capacity. On the other hand, any system reconfiguration will require different sets of Low Pass Filters and Oscillators in order to successfully downconvert the new channels to baseband. While this architecture provides some degree of flexibility, one cannot alter such parameters as modulation techniques, channel bandwidth, or coding schemes without a major hardware redesign. The cost of such a change would be enormous, as users would have to replace their handsets and wireless providers would have to replace their base stations.

Today in North America there are many cellular standards, including D-AMPS, CDMA and GSM, fighting for control of the market. It is also often the case, where one service provider maintains continuous operation on number of standard at the same time. Each standard comes with its own mobile phones and base station infrastructure, which means that the cost of operating all those incompatible standards is enormous.

Figure 1-1 Common Multicarrier/Multichannel Radio Receiver Architecture

Digital-design radio receiver has the potential to solve this problems by providing both savings in infrastructure costs and flexibility of operation. The widely accepted approach so far, is to move the hw/sw boundary as close to the front end antenna as physically possible. The optimal solution would be for the digitization of the incoming signal and the analog conversion of the outgoing signal to occur just after the antenna BPF. Unfortunately the A/D and D/A converters and processors are not yet capable of operation at such high sampling rates and it is unlikely that they will be in the near future. A solution to this problem is to use hardware frequency translation to an IF (Intermediate Frequency) band and then sample the signal with a wideband A/D converters. Figure 1-2 depicts the common software radio architecture.

Figure 1-2 Digital-Design Software Radio

By comparing Figures 1-1 and 1-2 it is easy to see that software radios eliminate the need of analog LPF and Oscillator sets for each channel. In fact, the whole analog BF Section functions can now be implemented in software or DSP hardware, including tuning, modulation, coding, accessing schemes etc. System reconfiguration can be done with ease, since only a few components need to be interchanged. Implementing better accessing schemes and functionalities can be accomplished “on-the-fly” by simply loading the appropriate software without enduring extensive system shut-down times and customer frustration.

1.2 Approach

Now it is time to give a most important statement that seems to be overlooked too often when considering software radios:

Software runs on Hardware!!!

For this reason, the focus of this research lies on investigating the hardware performance requirements of software radios with respect to applications in the area of mobile communications.

To be able to benefit from all the software processing advantages of the digital radio design, one needs first to bridge the gap between the analog signal format (common for the air link propagation interface) and the digital signal format (best suited for computerized processing). This, in almost all cases, is accomplished by the digital radio’s ADC in the receive path and DAC in the transmit path, thus making the converters, by far, the most important components in any digital radio design.

Converters alone, though, are not capable to deliver their best performance without proper interfacing. Because of this, the majority of any digital radio hardware circuitry is designed to facilitate the converters operation and do not distort their manufacturers’ specifications. How well this is accomplished will often determine the fitness of the device to successfully digitize and preserve the information content of the signal.

Realizing that Analog-to-Digital and Digital-to-Analog conversion are of critical importance for the whole system functionality, the thesis objective was to investigate the performance of currently available converter technologies as an integrated part of a fully functional digital radio, with respect to the mobile communications standards requirements.

After some preliminary research, contact was established with ICS Ltd., which in turn provided the much needed hardware for these investigations. To obtain accurate results, all experiments were carefully designed with respect to the most important parameters of interest to the wireless communication sector. Two high performance Hewlett Packard generators were used for signal and clock generation. Signal was then digitized and data captured to be transferred via the PCI bus into the OS memory.

Figure 1-3 Experimental Testbed

In non-real time, data was subjected to specially developed for the purpose software processing, including 16K FFT routine, data filtering routine and various parameter calculation and visualization routines. Both C/C++ and ©Matlab programming environment were employed in the software development to take advantage of speed plus flexibility features of C/C++ and programming simplicity of ©Matlab. Emphasis was put on evaluating SNR and SFDR as being one of the most important performance parameters, even though some experiments addressed such issues as Crosstalk between channels, Analog Input Bandwidth of the device, performance in the absence of applied signal etc. Experiments were made across the Nyquist BW for several sampling rates as well as for various input signal power levels.

Figure 1-4 Software Subsystem

1.3 Organization of this Report

First, Chapter 2 provides some background on relevant subjects. It starts by introducing software radios, followed by ADC-related issues and detailed explanation of the hardware used for the evaluation measurements. Chapter 3 describes the experiments in detail as well as the challenges currently faced in the hardware design of the experimental setup. It concludes with evaluation of the setup to provide unbiased and meaningful results.

The right-most block in Figure 1-3, pictured in greater detail in Figure 1-4, is the software subsystem and is addressed in Chapter 4. Chapter 5 presents the experimental results in a form of various plots and figures for each experimental category. Chapter 6 summarizes the experimental work by first outlining all sources of impairments known to be existing during the tests. Based on that it then evaluates the credibility of the experimental results and concludes with comparison of the 12 and 14-Bit converter technologies.

Much weight is put on the final chapters as being closely related to the motivation for this research. In Chapter 7, several mobile communications standards are closely looked at, and evaluated against the results obtained from the experiments with the ICS hardware. The final goal being to determine whether or not it is feasible to support them with today's converter technologies.

Chapter 8 acknowledges the current ADC technology dynamic range limitations as being one of the problematic issues in the successful design of software radio applications. As a result, few novel front-end architectures aimed at reducing the receiver' dynamic range requirements are discussed and evaluated.

Finally, Chapter 9 describes the overall performance of 14-Bit versus 12-Bit, suggested areas for improvements as noticed from the experiments, conclusions from the experimental work and provides guidelines for any future research.

Chapter 2

Background

This chapter provides general background on topics related to this thesis. Section 2.1 describes the general principles behind the concept of software radios. Section 2.2 refreshes the reader's knowledge about basic ADC parameters and provides some categorized technical specifications parameters of today's converters. Section 2.3 explains in details ICS-650 and ICS-152, the digital receiver board devices that were actually tested in the experimental work.

2.1 Software Radios

Over the past decade wireless communication technology has made a seemingly unstoppable march from the analog to the digital domain. As this move occurs, researchers are implementing more of the radio functionality in software - a trend which is leading to a device that has been coined the "software radio". Software radios promise to deliver greater flexibility through programmability.

Software control of radio functionality would allow quick and easy alteration of such parameters as channel bandwidth, frequency range, and modulation type. The benefits of such control are apparent when one considers the astonishing multitude of communications standards currently in place. Just a few of the presently used infrastructures are the Advanced Mobile Phone System (AMPS), Groupe Speciale Mobile (GSM), and Code Division Multiple Access (CDMA). Software radios could potentially be able to communicate within more than one system. In addition, new technologies could be applied without requiring expensive changes in the existing hardware infrastructure.

The basic architecture of a software radio includes a multi-band antenna and RF conversion, wideband analog-to-digital (A/D) and digital-to-analog (D/A) converters, and the implementation of data processing functions in programmable processors [1]. Figure 2- 1 illustrates this conceptual approach.

Figure 2-1 Software Radio Architecture

Since the receive and transmit portions are essentially the reverse of each other, let us examine the receive direction only. The antenna takes in the signal, filters out unwanted frequencies, and passes it along to the RF downconversion hardware. This circuitry shifts the frequency from the actual received band down to an Intermediate Frequency (IF) band, which is low enough to be sampled by a wideband A/D converter. The digital data output of the A/D converter is then fed into a digital processor, which performs the required processing and delivers the data, in the form of audio, video, or some other form, to the user. Hence the processor actually has access to the entire frequency band of interest, rather than only being able to work with a single station at a time.

This diagram demonstrates a distinguishing feature of software radios. The A/D and D/A converters are placed as close to the antenna as possible. In other words, the optimal solution would be for the digitization of the incoming signal and the analog conversion of the outgoing signal to occur just after the antenna and filter. Unfortunately, A/D/A converters and processors are not yet capable of operation at such high sampling rates, and so the RF conversion hardware is necessary to bring the signal down to more reasonable frequencies.

The simplicity of Figure 2-1 masks many challenges currently faced in the design of software radios. To achieve good wideband performance, the antenna must operate with low loss through multiple frequency ranges. The RF conversion hardware must offer a reasonable signal-to-noise ratio over this wide band as well. In addition, processing power capability is not always sufficient to handle the data processing required in real time [1].

Still, these difficulties are outweighed by the inherent possibilities of software radios, and one would certainly expect the growth of this new radio architecture to continue. For functions which are too complex for implementation in today's DSPs, application-specific processors and field programmable logic devices can be used. The design is modular and the interface specifications employ standards, which will allow upgrades to be made easily as the technology advances.

The goal is to use wide-bandwidth A/D technology and software-oriented approach to wireless communications to vastly extend the reach of software-based systems by sampling wide bands of data and performing the processing of the samples in application software. This approach provides tremendous flexibility for these software-based systems, as system parameters can be changed easily and inexpensively.

2.2 ADC Essentials

Software radios highly depend on the availability of an ADC that can efficiently sample the signal. Hence we devote a main part of this chapter to familiarize the reader with the basic parameters of these components. The quality of an ADC depends primarily on three parameters:

2.2.1 Introduction

Full-scale range:

The term full-scale range is used to describe the range of an A/D converter for bipolar signals. It can be found to be the difference between the largest positive and the smallest negative levels that the converter can physically handle.

$$V_{pp} = V_{max}^+ - V_{min}^- \quad (1)$$

For example the full-scale input of ICS-650 is $\pm 1V$. This makes the full-scale range $V_{pp} = 2V$. As a consequence smaller full-scale range imposes the need of higher quality LNA (Low Noise Amplifiers) and filters to scale the rms dynamic range of the input signal to fit the one of the ADC.

Analog Input Bandwidth:

This is usually the analog input frequency at which the spectral power of the fundamental frequency (as determined from the FFT analysis) is reduced by 3dB. It is usually referenced as $(DC \div f_{max})$ in MHz. For example the ICS-650 input signal bandwidth is around 200MHz when the 2 ADC channels are used in the process of conversion of a single analog signal.

Dynamic range:

This is the most important ADC parameter and it is associated with the number of bits that are used to encode the quantized signal at the output of the converter. In practice it is expressed in terms of the SFDR and SNR achievable when operating at the highest sampling rate, often accounting for various factors as Distortion from harmonics, Spurious Noise e.t.c.

The Dynamic range or as it is often called The Resolution of an ADC is linearly dependent on the number of bits used to encode the signal, and as such its impact on the quality of the software radio system is immense. Often it is the only parameter of interest that will influence the choice of using a particular converter and will be looked at closely in the following chapters.

2.2.2 Quantization Errors and Analysis

The majority of research in this topic models this type of error as noise that is added to the original (unquantized) signal. There are two types of such noise:

Overload Noise - This is the noise when the input falls outside the range of the quantizer.

Granular Noise - If the input analog signal is within the range of the quantizer, the quantization error resulting from the finite precision representation is bounded in magnitude to be less than 1/2 of the quantization level and the resulting noise is called granular noise.

In the scope of this thesis it can be assumed to a grate extend that the dominating factor in the evaluation of the quality of an ADC will be its quantization error $e_q(n)$ performance due to the granular noise, since the overload noise can be avoided by scaling the input signal so that its dynamic range falls within the range of the quantizer.

The mathematical model for the quantization error $e_q(n)$ is shown bellow:

Figure 2-2 Mathematical Model for Quantization Error Analysis

The most widely used expression for the Signal to Quantization Noise Ratio (defined as $SQNR = x_q(n)/e_q(n)$) of a given converter, by using deterministic approach is found to be:

$$SQNR(dB) = 1.76 + 6.02 \times b \quad (2)$$

Almost every ADC manufacturer uses this formula as a starting point in the technical specifications for its device, regarding its Dynamic range (Resolution) performance. Then the effects of distortion, spurious noise e.t.c. are accounted for to formulate more realistic Dynamic range specifications. The above formula implies that the SQNR increases approximately 6 dB for every bit (b) added to the word length, that is for each doubling of the quantization levels.

Similar result can be derived using a statistical approach as sown by J. G. Proakis in its "Digital Signal Processing" textbook [2]:

$$SQNR(dB) = 6.02 \times b + 16.81 + 20 \log \frac{\sigma_x}{V_{pp}} \quad (3)$$

Here the SQNR is not only dependent on the wordlength used for encoding, but also on the ratio of the range V_{pp} of the ADC to the variance σ_x of the input signal. For example, if we assume that the signal is Gaussian distributed and the range of the quantizer extends from $-3\sigma_x$ to $3\sigma_x$ (i.e., $V_{pp} = 6\sigma_x$), which simply means that less than 3 out of every 1000 input signal amplitudes would result in an overload on the average, then the equation becomes:

$$SQNR(dB) = 6.02 \times b + 1.25 \quad (4)$$

The result is very similar to the one found by using the deterministic approach and that explains the wide use of [2] in the ADC specifications.

2.2.3 Oversampling as a source for SQNR boost.

In software radios wireless applications it is common to sample the entire bandwidth of interest prior to any processing. In most cases this bandwidth is on the order of few tens of MHz, while the channel bandwidths are much smaller. To alleviate the computational resources limitations of today's processing tools, downconversion is usually performed in the digital domain. This not only helps to process the signal in real time, but also improves the SQNR [3].

For example the uplink GSM operation band is approximately 25 MHz, while every physical channel is only 200 KHz wide. With this in mind, a translation to IF in the vicinity $\{1 \div 26\}$ MHz and a following sampling with ICS-650 at 65 MHz, the oversampling ratio U for each GSM carrier can be easily calculated to be:

$$U = \frac{\text{Nyquist BW}}{\text{Channel BW}} = \frac{(65\text{MHz}/2)}{0.2\text{MHz}} = 162.5 \text{ times} \quad (5)$$

According to equation (5) in [3], the in-band quantization noise reduces by 3 dB with every doubling of U . Thus, for most of the physical channels the improvement from oversampling will be significant.

In the following, two basic types of ADC are investigated.

2.2.4 Flash-ADCs

Let V_{pp} be the maximum amplitude range (peak-to-peak-voltage), b the number of bits in the quantizer and U the oversampling ratio. Digitizing a signal with amplitude having Normal distribution (zero mean, variance σ_x^2), the SNR in the frequency-band-of-interest is [4]:

$$SNR_U|_{dB} = 10.8 + 6.02 \times b + 10\log U + 20\log \frac{\sigma_x}{V_{pp}} \quad (6)$$

We observe from equation (6) that:

- 1). The SNR rises with 6.02 dB per additional bit of quantizer resolution.
- 2). The SNR rises with 3.01 dB per doubling the oversampling ratio, equivalent to 1/2 bit more quantizer resolution.

This leads to an equivalent oversampling resolution gain of:

$$b = \frac{1}{2} \log_2 U \text{ bits.} \quad (7)$$

2.2.5 Sigma-Delta Converters

Sigma-Delta converters are noise-shaping ADCs consisting of the Sigma-Delta modulator (SDM), being the AD interface, and the decimator, being a filter reducing the sample rate. Since the part where the AD conversion actually takes place is the modulator, we part the modulator from the decimator and analyze the SDM only. For simplicity we stick to one of the classic 2nd order SDMs employing double-integration, sketched in Figure 2-3 [5]. The noise transfer function (NTF) and the signal transfer function (STF) of this modulator are:

$$NTF(z) = \frac{Y(z)}{N(z)} = (1 - z^{-1})^2 \quad STF(z) = \frac{Y(z)}{X(z)} = z^{-1} \quad (8)$$

While the input signal to the SDM is filtered by an all pass function, the quantization noise is filtered by a high pass function. For estimating the power spectral density of the quantization noise at the output of the modulator we evaluate the NTF on the unit circle and calculate the SNR within the U^{th} fraction of the total frequency band (P_x being the signal power and N_Q the quantization noise power):

$$SNR_U|_{dB} = 10\log \frac{P_x}{N_Q} + 10\log \frac{\pi}{\int_0^{\frac{\pi}{U}} 16 \sin^4 \left(\frac{\omega'}{2} \right) d\omega'} \quad (9)$$

The first term in equation (9) represents the SNR of the quantizer without noise shaping, and the second term represents the enhancement of the SNR by the noise shaping using Sigma-Delta modulation.

Figure 2-3 Block Diagram of 2nd order low pass SDM with linearized quantizer

Applying the 3rd order Taylor series expansion to the sine function in equation (9) and generalizing for SDMs of L^{th} order with sinusoidal noise shaping the SNR is [6]:

$$SNR_U|_{dB} \approx 20 \log \frac{\sigma_x}{V_{pp}} + 6.02b + (2L + 1)10 \log U + 10.8 + 10 \log \frac{2L + 1}{\pi^{2L}}, \quad U \geq 4 \quad (10)$$

We observe that the SNR rises

- 1). With 6.02 dB per additional bit of the quantizer resolution, and
- 2). With $6(L+1/2)$ dB per doubling the oversampling ratio, equivalent to $L+1/2$ bit quantizer resolution.

We have made the analysis using a low pass SDM. However, a low pass prototype SDM can be transformed into a band pass SDM having equivalent characteristics of noise-shaping and dynamic range. A comprehensive analysis of SDMs can be found in [6] and the references given there.

2.2.6 Flash or SDM?

The whole idea behind the concept of mobile communications software radios is to cut down on hardware costs and improve the flexibility of the system to support various standards in platform interchangeable software. This imposes very strict requirements on the converter components employed in the system architecture.

As it is seen from the SDM analysis section, they have the benefit to improve on SNR much more rapidly than the Flash-type converters - $6(L+1/2)$ dB per doubling the oversampling ratio for SDM, compared to 3.01 dB for Flash. But SDM are rarely implemented with more than 1-Bit quantizer resolution. Rather, they compensate with high oversampling ratios. As an example we analyze a 2nd order noise-shaping SDM that meets the requirements of a GSM-GPS-DECT receiver, requiring 80 dB SNR within a band of 200 kHz., Digitizing a bandwidth of 2 MHz and using equation (10) with $2\sigma_x/V_{pp} = 1/3$ for Normally distributed amplitudes signals, we get:

$$SNR_U|_{dB} \approx 50 \log U - 11.7, \quad U \geq 4 \quad (11)$$

From here, the required oversampling ratio is derived to be greater than 68, translating to 13.6 MHz converter sampling rate. Such devices are readily available from number of distributors, but in the scope of a fully functional 125-channel, 25 MHz bandwidth GSM standard, with approximately 30 channels per cell, this will mean that separate SDM be employed for each channel. It is imperative also to note that, even though one might be tempted to reuse some of the SDM modulators to cover more bandwidth, this will result in degradation in the SNR for those channels due to the noise-shaping nature of SDM. This is illustrated in Figure 2-4, where the power spectral density (PSD) of the quantization noise of an SDM is plotted (dotted) along with the GSM-mask (interference levels, dashed) and the minimum received signal (solid). The zero-dB line represents the white quantization noise floor of Flash ADCs.

From this figure and the above analysis, it is seen that SDMs are not an appropriate choice for software radio design for several reasons:

- 1). Because of the noise shaping property of SDMs, a number of such devices are needed to cover the proposed capacity of the mobile system, which is in sharp contrast with the software radio concept to cut down on hardware costs.
- 2). If only one SDM is to be employed, it has to operate at very high sampling frequencies or to be implemented with much higher order noise shaping filter structure - both requirements unattainable with today's technology.
- 3). In the field of mobile communication, SDMs are a tailored solution because of their high complexity and increased processing requirements - again, in disagreement with the software radio perspective of simplicity and flexibility.

Figure 2-4 Noise-shaping characteristics of SDM

Flash ADCs, on the other hand, are much more fit to fulfil the software radio requirements. Unlike SDM, Flash ADCs rely on increased quantizer bit-resolution as primary source for converted signal quality. They have fairly simplified architecture and constant performance characteristics across the frequency range.

Resolution	Sampling Rate (MSPS)		
	Analog Devices	Maxim	Harris
10 bits	<u>200</u> (400 MHz BW)	<u>40</u>	<u>60</u> (250 MHz BW)
12 bits	<u>125</u> (500 MHz BW)	<u>30</u>	<u>10</u> (100 MHz BW)
14 bits	<u>80</u> (200 MHz BW)	<u>22</u>	<u>5</u> (100 MHz BW)
16 bits	<u>n/a</u>	<u>n/a</u>	<u>n/a</u>

Table 1: Current ADC sampling rates specifications

Table 1. outlines some of today's "state of the art" devices performance. The guideline specifications were taken directly from the companies web sites as of May 14, 1999. All of the above converters are commercially available and most of them are designed especially to facilitate the "software radio" concept of having access to the entire RF band of operation. There is also a great deal of research towards higher sampling rates for all resolutions and as the experience have shown the sampling speed doubles every 3 to 5 years depending on the needs of the industry.

As it can be seen from Table 1., even without exact technological details and specifications, it is truly attainable to sample the entire spectrum of operation of most of the Mobile Communication standards in operation today. Consider some of them:

- AMPS/IS-54/IS-136 - They all use identical spectrum allocation of 12.5 MHz per operator. If sampled at baseband or close to baseband IF, most of the above devices satisfy the Nyquist requirement even with highest resolution.
- GSM/E-GSM - With spectrum allocation of 25MHz per GSM operator and 35MHz for EGSM all Analog Devices will do the work easily.
- IS-95/W-CDMA/CDMA-2000 - With channel reuse ratio of 1 and BW/Ch. = 1.25 MHz all of the above converters can support several IS-95 channels at the same time. For the broadband versions most of the above devices suffice.

Nyquist criteria is a necessary, but not sufficient requirement when choosing ADC devices for software radios. Parameters, such as SNR and SFDR, are of much grater importance for wireless applications, as it will be seen in the final chapters. Nevertheless, Flash ADCs by far are the most suitable converters for mobile communications software radio applications. Not surprisingly, most of the hardware available today is implemented with such devices and the majority of research in this field concentrates on improving the resolution and dynamic range of Flash ADCs.

2.2.7 Flash ADC SNR Performance Expectations

We try to analyze now what level of theoretical performance can be expected from Flash ADCs in terms of SNR. This parameter estimation is important for the proper design of a digital radio-based application since it is the factor that indirectly affects the proposed dynamic range coverage of the system.

Using the more conservative equation (4) with no oversampling gain and Normal signal distribution we estimate the theoretical SNR performance to be:

- **12-Bit Flash ADCs** $\Rightarrow SQNR(dB) = 6.02 \times 12 + 1.25 = 73.5 \text{ dB}$
- **14-Bit Flash ADCs** $\Rightarrow SQNR(dB) = 6.02 \times 14 + 1.25 = 85.5 \text{ dB}$

Of course, other factors such as harmonics distortion, integral and dynamic nonlinearity and external circuitry thermal noise will limit the above theoretically derived parameters. Most manufacturers try to account for these effects in their devices' technical specifications by means of measurements or sophisticated simulations. For example the specified typical SNR for AD6640 (the ADC used on the ICS-650) with analog input @ -1dBFS and 15.5 MHz is around 67.7 dB. This is almost 1 bit of resolution less than the theoretically expected.

Nevertheless, from the above equations it is seen that there is more or less a stable improvement in terms of SNR with increased bit-resolution converters.

2.3 The ICS-650 and ICS-152 Acquisition boards

The ICS-650 is a 2 channel, 12bit, 65 MHz, PCI analog input board. It includes a 2 MegaSample "swing buffers" for ADC data storage. For applications in Communications, an optional tuner daughter card can be added in order to demodulate (real or complex demodulation) any user-programmed signal band and reduce the output rate, thus facilitating the implementation of a fully functional digital radio receiver.

The ICS-152 is an acquisition board, similar to the ICS-650, with the only major difference that it employs a 14-Bit resolution ADC capable of operation at 70 MHz. Detailed technical specifications for those boards can be found at ©ICS Ltd. home page <<http://www.ics-ltd.com/>> and in Appendix 1.

In the scope of this thesis, a short description of the main blocks is now presented to the reader for a brief reference and introduction to the following chapters. Figure 2-5 shows a simplified block diagram of the ICS-650 board. The board uses 2, 12-bit ADCs (Analog Devices AD6640) to provide simultaneous sampling at rates of up to 65 MHz/channel for two channels. ADC data is buffered either in the 8 KSample FIFO (First-In First-Out memory) when data is read out via the FPDP, or the dual-ported memories when data is read out via the PCI bus.

Figure 2-5 ICS-650 Block Diagram

The sampling clock and the trigger can be either internal or external. The internal ADC clock is user programmable in steps of approximately 4 kHz. The ADCs can be operated in a number of different modes. In the continuous mode, data is continuously supplied to the selected interface upon application of a trigger signal until the acquisition is disabled. In capture mode, a fixed number of samples are acquired upon each application of the trigger.

The ICS-650 can be configured to pre-store samples before the trigger and acquire a programmable number of samples following the trigger (to a maximum of 524,288 samples/channel), or initiate conversion upon each application of the trigger and acquire a programmable number of samples (again, to a maximum of 524,288). To allow fast transfer of ADC data, the ICS-650 board includes a PCI interface capable of in excess of 132 MByte/s when using burst mode bus cycles, and a 32-bit front panel data port (FPDP) interface. The PCI interface uses the V³ V360EPC chip to support a master/target burst transfer interface. The ICS-650 board can generate PCI bus interrupts at any user programmed interval (number of samples acquired). All power requirements of the ICS-650 are satisfied with standard PCI bus voltages.

The ICS-650 board uses 2, 12-bit ADCs (AD6640), which sample synchronously at frequencies of up to 65 MHz using either a programmable on-board clock or an external frequency source. The minimum sampling rate for the ADCs is 5 MHz, but the ICS-650 provides decimation of the ADC data stream by a factor of up to 32, making the effective minimum sampling rate 156.25 kHz. Decimation is accomplished by storing one out of every N samples where N is programmable from 1 to 32. The full scale input signal level is $\pm 1 V$, with an input impedance of 50 ohm. The outputs of the two ADCs are stored as a single, concatenated 24-bit word in the FPD FIFO or in the PCI swing buffer. Prior to transfer over the PCI bus, each 12-Bit sample in the 24-Bit word is converted to 16-Bit integer to take advantage of the 32-Bit wide PCI bus architecture. The conversion is done in the following way: Each 12-Bit sample is shifted 4 bits to the left and then the four least significant bits (LSB) are zero-filled. Note that this conversion is not destructive and there is no need to convert the samples back to their original 12-Bit representation for further processing, providing the software accounts for the shift in magnitude.

As mentioned earlier, the ICS-152 shares the same design architecture with the ICS-650, except for the difference in converter resolution and maximum sampling rate. For this reason any references to ICS-650 will also apply to the ICS-152, except in those cases where the differences will be mentioned explicitly by the author.

Chapter 3

Experimental Setup

This chapter covers the general aspects of the experimental setup. Section 3.1 contains the experimental definitions as well as details about hw/sw requirements and measurements algorithms. Section 3.2 describes the main points of consideration during the hardware design phase and presents in details the whole evaluation platform for the measurements. The goal of the final section is to evaluate the credibility of the experimental testbed.

3.1 Defining the Experiments

Our main target was to measure some of the key performance parameters of interest to the mobile communication industry. The general assumption for all measurements was that the ICS board design does not significantly degrade the performance specifications of the ADC used on the board (Analog Devices AD6640). There are many definitions to be found relating to those parameters, but to be consistent with the hardware under investigation, our decision was to follow the definitions given by Analog Devices in their technical specifications data sheets. They are as follows:

3.1.1 Spurious-Free Dynamic Range

The ratio of the rms signal amplitude to the rms value of the peak spurious spectral component. May be reported in dBc (i.e., degrades as signal level is lowered), or in dBFS (always related back to converter full scale).

Experimental details:

To be measured:

- In a phase-locked, non-windowed experimental environment (Coherent Sampling of the applied analog input signal).
- With low phase noise, externally supplied clock source.
- For several different sampling rates, with emphasis in the vicinity of F_s max (65MSPS).
- For a number of analog input frequencies and amplitudes, with emphasis on signals at 12.5MHz, 25MHz and 30MHz - for the reason to be related to the needs of the wireless industry.

To be used:

- "ICS Simple Test" data capture software in combination with 16K FFT routine.
 - Harmonics removal and aliases filtering software modules.
 - Parameter calculation and visualization software.
-

Algorithm:

- I. For each individual measurement capture 0.5 Mb of data via the ICS-650 acquisition board.
- II. Segment into 32, 16Ksamples independent data modules.
- III. For each module perform 16K FFT, harmonics filtering and aliases removal.
- IV. Derive individual SFDR parameter by subtracting the Highest Spur magnitude value from the Signal value (in dB).
- V. Average over the 32 SFDR values to obtain statistically independent result.

Some points to consider:

- The spectral purity of the test equipment!
- Proper clock supply termination on the ICS-650 board.
- How the input signal level affects the SFDR performance?
- Is the generated analog input signal falling on a Bin?

3.1.2 Signal-to-Noise Ratio (without Harmonics)

The ratio of the rms Signal amplitude to the rms value of the sum of all other spectral components, excluding the first five harmonics and dc.

Experimental details:To be measured:

- In a phase-locked, non-windowed experimental environment (Coherent Sampling of the applied analog input signal).
- With low phase noise, externally supplied clock source.
- For several different sampling rates, with emphasis in the vicinity of F_s max (65MSPS).
- For a number of analog input frequencies and amplitudes, with emphasis on signals at 12.5MHz, 25MHz and 30MHz - for the reason to be related to the needs of the wireless industry.

To be used:

- "ICS Simple Test" data capture software in combination with 16K FFT routine.
- Harmonics removal and aliases filtering software modules.
- Parameter calculation and visualization software.

Algorithm:

- I. For each individual measurement capture 0.5 Mb of data via the ICS-650 acquisition board.
 - II. Segment into 32, 16Ksamples independent data modules.
 - III. For each module perform 16K FFT, harmonics filtering and aliases removal.
 - IV. Sum all spectral components in magnitude squared, excluding dc, signal and first five harmonics.
 - V. Derive individual SNR parameter by subtracting the square root of the sum from the Signal value (in dB).
 - VI. Average over the 32 SFDR values to obtain statistically independent result.
-

Some points to consider:

- The spectral purity of the test equipment!
- Proper clock supply termination on the ICS-650 board.
- How the input signal level affects the SNR performance?
- Is the generated analog input signal falling on a Bin?

3.1.3 Analog Input Bandwidth

The analog frequency at which the spectral power of the fundamental frequency (as determined by the FFT analysis) is reduced by 3 dB.

Experimental details:To be measured:

- In a phase-locked, non-windowed experimental environment (Coherent Sampling of the applied analog input signal).
- With low phase noise, externally supplied clock source.
- For several different sampling rates, with emphasis in the vicinity of F_s max (65Msps).
- For a number of analog input signal levels.

To be used:

- "ICS Simple Test" data capture software in combination with 16K FFT routine.
- Parameter calculation and visualization software.

Algorithm:

- I. Start with relatively close to dc signal, falling on a Bin.
- II. For each input frequency point capture 0.5 Mb of data via the ICS-650 acquisition board.
- III. Segment into 32, 16Ksamples independent data modules.
- IV. For each module perform 16K FFT and derive signal value in dB.
- V. Average over the 32 SFDR values to obtain statistically independent result.
- VI. Continue up in frequency until the averaged signal value has dropped by 3 dB relative to the averaged signal value at the starting frequency point.
- VII. Derive the Analog Input Bandwidth parameter by subtracting the starting frequency point from the final frequency point.

Some points to consider:

- The spectral purity of the test equipment!
 - Proper clock supply termination on the ICS-650 board.
 - How the input signal level affects the Analog Input Bandwidth performance?
 - Is the generated analog input signal falling on a Bin?
-

3.1.4 Channels Crosstalk

The ratio of the rms Signal amplitude in Channel 1 to the rms value of the spectral component amplitude at the same frequency location in Channel 2 and vice versa.

Experimental details:

To be measured:

- In a phase-locked, non-windowed experimental environment (Coherent Sampling of the applied analog input signal).
- With low phase noise, externally supplied clock source.
- For several different sampling rates, with emphasis in the vicinity of F_s max (65Msps).
- For a number of analog input frequencies and amplitudes, with emphasis on signals at 12.5MHz, 25MHz and 30MHz - for the reason to be related to the needs of the wireless industry.

To be used:

- "ICS Simple Test" data capture software in combination with 16K FFT routine.
- Parameter calculation and visualization software.

Algorithm:

- I. Start with relatively close to dc input signal @ -6dBFS, in Channel 1.
- II. Capture 0.5 Mb of data via the ICS-650 acquisition board in Channel 2.
- III. Segment into 32, 16Ksamples independent data modules.
- IV. For each module perform 16K FFT and derive Crosstalk value in dB.
- V. Average over the 32 SFDR values to obtain statistically independent result.
- VI. Continue up in frequency to cover the Nyquist range.

Some points to consider:

- The spectral purity of the test equipment!
- Proper clock supply termination on the ICS-650 board.
- How the input signal frequency affects the Crosstalk performance?
- Is the generated analog input signal falling on a Bin?

Some of the terminology in the above definitions may need clarification. For this purpose we provide brief explanation for those of most importance in understanding the experiments.

Coherent Sampling - To obtain accurate representation, analog signals need to be sampled coherently. In a sampled system, a reference clock is usually used to sample the input signal. We limit our discussion to *periodic* or *uniform sampling*, which is the type of sampling used most often in practice. This is described by the relation

$$x(n) = x_a(nT), \quad -\infty < n < \infty \quad (12)$$

where $x(n)$ is the discrete-time signal obtained by "taking samples" of the analog signal $x_a(t)$ every T seconds. Now, consider the effect of reference clock jitter on the sampling system.

Figure 3-1 Error resulting from reference clock jitter

Any sample-clock-aperture uncertainty (the sample-to-sample variation in the encoding process) results in an error in the sampled signal accuracy. The statistical characteristics of this error depend not only on the clock jitter value ($t_{j_{rms}}$), but also on the slew rate of the sampled analog signal. At a given frequency, f , the maximum slew rate occurs at the zero crossing and equals the peak signal amplitude times $2\pi f$. Therefore sampling of high frequency signals demands very stable clock supply to the ADC circuitry.

Figure 3-2 Error value depends on the frequency of the sampled signal

This is illustrated in Figure 3-2, where the resulting errors from the same jitter value are different for different frequencies.

As in mixing, a local oscillator (in this case a, sampling clock) multiplies the input signal. Because the multiplication in the time domain is equal to convolution in the frequency domain, the spectrum of the sample clock is convolved with the spectrum of the input signal. Any aperture uncertainty, which appears on the clock as wideband noise, also shows up as wideband noise in the sampled spectrum. Because the ADC is a sampling system, this spectrum is periodic and repeated around the sample rate. The wideband noise degrades the ADC's noise performance. You can determine aperture uncertainty, which limits the theoretical SNR for an ADC, by[7]:

$$SNR = -20\log(2\pi F_{ANALOG} \times t_{j_{rms}}) \quad (13)$$

where F_{ANALOG} is the analog input frequency and $t_{j_{rms}}$ is the aperture uncertainty.

IF sampling typically requires an aperture uncertainty of less than 1 psec to achieve required noise performance. In terms of phase accuracy and intersymbol interference, the effects of aperture uncertainty are small. In worst case-scenario of 1psec rms uncertainty, the phase uncertainty, or error, is 0.09 rms at an IF of 250 MHz. This value is acceptable even for a demanding specification, such as GSM. Standard TTL/CMOS clock oscillator modules can provide 0.7 psec rms jitter, while low-noise modules can provide better numbers.

Signal falling on a Bin - When using FFT analysis of sampled analog signal, the signal spectrum is evaluated at equidistant points (Bins), the number of which depends on the FFT-length. Only frequencies coinciding with those spectral points can be represented accurately. Any signal frequency, not exactly at one of these points, when evaluated by the FFT analysis, results in an interpolated value depending on its neighboring Bins in the spectrum. The greater the difference between the spectral values of these neighboring points, the greater is the resulting error. Therefore in the scope of our experiment, where 16K FFT was routinely used to analyze 16 Ksamples of data, only frequencies of

the form $N \times \left(\frac{F_s}{16K} \right) = N \times \left(\frac{F_s}{16384} \right)$ Hz, where $N = 1, 2, \dots, 16384$ can be accu-

rately represented. For example if $F_s = 60 \times 10^6$ Hz, applying analog input signal with frequency 10 MHz will be inaccurately represented by the FFT analysis, while applying signal with frequency 10.078125 MHz will result in accurate representation. This requirement doesn't impose any restriction on the analog signals in field applications since the FFT can be made of any suitable length as well as signals can be downconverted to appropriate frequencies during the mixing stages prior to conversion

3.2 Hardware design and considerations

In order to achieve the best experimental setup possible, a considerable amount of related literature was researched and evaluated. Sources of information in particular were Application Notes and Data Sheets from semiconductor companies' web sites, selected papers from related periodic publications etc. To be compliant with the experimental definitions, few points needed consideration:

1). Results will be dependent on the spectral purity of the generated analog input signal.

Therefore we needed a very high performance signal generator with specification parameters, exceeding the ones specified for the boards under testing. Especially critical were the spectral purity and phase noise of the generator.

2). Results will be dependent on the stability of the external clock source.

Therefore we needed very stable continuous clock supply source to drive the ADC circuitry on the boards.

3). Results will be dependent on the synchronization of the input signal and the supplied clock.

Therefore we needed a simple and efficient way to synchronize the two sources and achieve coherent sampling.

After carefully evaluating the above requirements, it was decided to use the setup shown in Figure 3-3.

Figure 3-3 Experimental Setup

Two high performance Hewlett Packard signal generators were used to supply the analog input signal and clock needed to drive the boards. Detailed technical specifications for both generators can be found at Hewlett Packard's web site, while the most important ones are provided in Appendix 2.

Since HP8644A has better spurious performance, it was used to supply the analog input signal, while the HP3335A was used as external clock supply source. The two generators were synchronized using the 10MHz Sync signal (standard for all HP generators) supplied from the HP8644A. The HP3335A was locked in-phase to the HP8644A via standard, 50 Ω BNC connector cable. This way it was possible to achieve coherent sampling, which eliminates the need to use a windowing function (such as Bartlett or Hamming) on the captured signal data.

ICS-650 and 152 accept standard TTL-level voltages to drive their external clock supply circuitry. To interface the standard sinusoidal signal output of the HP3335A with the TTL-level compatible clock input of the ICS boards, a custom-made buffer device was also employed. The details of this buffer (shown as "Clock" on Figure 3-3) are illustrated on Figure 3-4.

Figure 3-4 Sinusoidal to TTL-level Interface Buffer

In this Figure, there are two IC that need further clarification. The TL3016C is a ultra-fast, low-power, precision comparator from Texas Instruments. It is designed for a typical 7.8 ns fast conversion, which in terms of frequency means a 128 MHz duty cycle operation. This nearly twice exceeds our requirements of supplying 65 MHz to the boards.

The other IC is simply a line driver, which is also used on the ICS boards ADC clock supply circuitry. In our case it was used as internal buffer to smooth any residual discrepancies due to the comparator as well as for proper line termination.

The platform of choice for computational evaluation was Windows NT 4.0. It has readily build internal support for the PCI architecture and well debugged software drivers for all of the ICS boards under measurements.

Finally, MS Visual C/C++ and Matlab software developing and visualization environments were used to build the software modules needed for all measurements.

3.3 Experimental setup evaluation

Before any measurements can be done, our setup needed evaluation. In our setup we had 4 different pieces of hardware. The HP8644A was rented out and came pre-calibrated with parameter specifications guaranteed from the renting company. Therefore we eliminated it as a possible source of impairments, which would undermine the credibility of the measurements. The other HP generator was borrowed from Carleton University and as such was with questionable accuracy. The same applied for the custom-made clock buffer device, which was assembled at ICS prior to the measurements. Even though these two devices were checked carefully against any flows in their performance on several oscilloscopes at ICS, and seemed to comply to their design specifications, we needed to evaluate their performance in the scope of the experimental setup.

For this purpose we measured the ICS-650 performance with no signal applied at its CH1 and CH2 inputs for several different sampling rates, using the same setup outlined in Figure 3-3. An indication of a good experimental setup performance would be to achieve similar noise characteristics as the ones, specified from Analog Devices for AD6640 - the ADC used on the board. With no signal applied, the available thermal noise power for a given bandwidth and temperature of operation is derived from[7]:

$$P_{TN} = kTB, \text{ in Watts} \quad (14)$$

For a sample rate of 60 Msps, the Nyquist bandwidth B is 30 MHz. This, with average room temperature $T = 300$ Kelvin and Boltzmann's constant $= 1.38 \times 10^{-23}$ J/K, gives a noise power of:
 $P_{TN} = 1.38 \times 10^{-23} \cdot 300 \cdot 30 \times 10^6 = 124.2 \times 10^{-15} \text{ W}$, or -99 dBm. Because in our experiment we often use dBFS (decibels, related to Full-Scale), we relate these two values.

Full Scale (V_{FS}) for AD6640 (and for ICS-650) is 1V. With P_r being the power applied to a resistor, the relation is $P_r = V_{rms}^2 / R$. V_{rms} represents the root mean square of the applied voltage and is equal to $0.707V_{FS}$. The ICS-650 input impedance is 50Ω . Then for P_r we have:
 $P_r = 0.707^2 / 50 = 0.01 \text{ W}$. This, when converted to dBm equals 10 dBm. Therefore we have the relation:

$$\text{Full Scale} = 10 \text{ dBm} \quad (15)$$

Using [15], the available thermal noise power of -99 dBm is equivalent to -109 dBFS in terms of Full Scale. This is a short deviation from our main discussion, but is needed for understanding the setup evaluation, since almost all our plots and calculations are based on dBFS values.

Figure 3-5 shows a typical noise performance for ICS-650, with externally applied clock source and no input signal. The plot is taken from an experiment at 60 Msps sampling rate. As an estimate, the average noise value is seen to be at around -105 dBFS.

To be more accurate, we averaged over a number of 16K data captures. Figure 3-6 illustrates the average noise performance at 60 Msps over 32, 16K data captures. With average noise value of $P_{AV} = -103.414 \text{ dBFS} = 4.5562 \times 10^{-10} \text{ mW}$, we have total integrated noise power of:

$$P_{TOT} = 8191 \times P_{AV} = 8191 \times (4.5562 \times 10^{-10} \text{ mW}) = -64.28 \text{ dBFS} \quad (16)$$

This value represents all noises, thermal (P_{TN}) and quantization (P_{QN}) within the ADC, and it comes close to the minimum SNR value of 64 dB specified from Analog Devices for AD6640.

Figure 3-5 ICS-650 Typical Noise Performance

Figure 3-6 ICS-650 Samples Averaged Noise Performance

For comparison, we provide a similar FFT-plot, taken directly from Analog Devices AD6640 technical specifications data sheet - see Figure 3-7.

Figure 3-7 AD6640 FFT-plot @ 2.2 MHz and 65 Msps Sampling Rate

The plot is from measurements with applied input signal, but this has very little influence on the spectral noise characteristics. On this plot we see similar noise performance, as the one on Figure 3-5, with noise floor at around 100 dBFS (not taking the spurs into account) and fairly flat spectrum through the Nyquist frequency range.

To see how the experimental setup will handle input signals, we made some evaluations with signals at different frequencies and close to full-scale amplitudes. A typical performance with input signal applied in CH1 is shown on Figure 3-8. The noise floor is again at around -100 dBFS, with the highest spurs well detained below the 90 dBFS level. The signal is clearly outlined and appears as a single spectral line in the FFT plot. Its amplitude of -1dBFS is properly represented in the graph. We see a great resemblance between Figures 3-7 and 3-8, which is another indication of the ability of our experimental setup to provide accurate results. Similar to the shown on Figure 3-8, performance was noticed through the entire frequency range and smaller amplitudes.

We provide much more extensive coverage in the following chapters, but for the time being it is possible to come to the conclusion that our experimental setup is well fit to be employed in accurate ICS-650 measurements and performance evaluation.

Figure 3-8 ICS-650 performance with input signal

Chapter 4

Details of Software Modules

4.1 Software Framework

Figure 4-1 shows the main software blocks which perform the data processing in our experiments. The Data Capture module is part of the software that comes with the ICS board. The captured data then is put through the 16K FFT module, based on Split-Radix FFT Algorithm by Sorensen, Heideman and Burrus[8]. To avoid influence from harmonics and aliases, Harmonics Filtering and Aliases Removal modules are used. Finally, the processed data is ready for actual parameter calculations, which are implemented with measurements-specific modules. All of the above-described modules were developed in C/C++ software environment with the main benefit of having fast execution and flexible software support for our measurements. The visualization and presentation module was again measurements-specific and implemented as Matlab script files. Since there was no real-time processing constraint imposed on the measurements, our software was designed to operate in linear fashion to avoid any extra complications from parallel fashion-like or “layered” processing. For reference the exact coding details for all software modules are provided in Appendix 3. Here we only discuss the parts of the software modules that are relevant to understanding the results of this research.

Figure 4-1 The Software Framework

4.1.1 Data Capture Module

The data capture module is implemented with the help of the ICS-650 API (Application Programmer's Interface) software library that comes with the board. The purpose of this module is to initialize the ICS-650 to the supplied via the API functions parameters, to acquire the specified amount of data and to transfer it from the on-board storage buffers into the system's memory space via the PCI-bus - refer to Figure 4-2.

Figure 4-2 Data Capture Module Architecture

Note that in order for this module to operate successfully, it needs to have the device driver properly installed on the operating system as well as the API library software. The way these different pieces of software interact is illustrated in Figure 4-3.

Figure 4-3 Windows Host software Layout for ICS-650 SDK

4.1.2 16K FFT Module

As mentioned earlier, the FFT software module implements Split-Radix FFT (SRFFT) algorithm, originally derived by Duhamel and Hollmann. The theoretical basis for the FFT module was Reference 8. The idea behind the SRFFT is to use radix-2 index map for the even-indexed terms and radix-4 map for the odd-indexed terms. The basic definition of the DFT

$$C_k = \sum_{n=0}^{N-1} x_n W^{nk} \quad (17)$$

with $W = e^{-j2\pi/N}$ gives

$$C_{2k} = \sum_{n=0}^{N/2-1} [x_n - x_{n+N/2}] W^{2nk} \quad (18)$$

for even index terms, and

$$C_{4k+1} = \sum_{n=0}^{N/4-1} [(x_n - x_{n+N/2}) - j(x_{n+N/4} - x_{n+3N/4})] W^n W^{4nk} \quad (19)$$

and

$$C_{4k+3} = \sum_{n=0}^{N/4-1} [(x_n - x_{n+N/2}) + j(x_{n+N/4} - x_{n+3N/4})] W^{3n} W^{4nk} \quad (20)$$

for the odd index terms. This results in an L -shaped “butterfly” which relates a length- N DFT to one length- $N/2$ DFT and two length- $N/4$ DFTs with twiddle factors.

An analysis of the arithmetic complexity showed that the SRFFT is a significant improvement over the Cooley-Tukey FFT and is probably optimal in the sense of minimum floating point arithmetic. It has an efficiency exceeding a radix-8 FFT, a size comparable to a radix-4 FFT, and the flexibility of a radix-2 FFT.

The provided FORTRAN example program in the referenced paper was translated into C/C++ code and adapted to our experimental requirements. The FFT module accepts 3 parameters:

- A reference pointer to the starting location of the captured data in the system’s memory.
- The size of the FFT (in our case equal to the captured data samples - 16K).
- The type of the FFT - with real or complex data.

The FFT module handles the complex nature of the Fourier Transform by operating separately on two types of data - Real and Imaginary, each represented by real-valued numbers. Once called, it initializes the captured data into two operational arrays - X and Y . For real captured data FFTs, the Y array is initialized with all 0 values. It then operates recursively, in-place, on these two arrays to derive the spectral contents F_n of the captured data in the final instances of the X and Y arrays. Graphically, the functions of the FFT module are illustrated on Figure 4-4. In all measurements, the real-valued analog signal was captured via CH1 on the board, while the contents of CH2 were discarded for real-data FFTs.

Figure 4-4 Functional description of the FFT module

4.1.3 Harmonics Filtering Module

To avoid influence in the accuracy of the measured parameters, harmonics filtering module was employed in the signal processing chain. Its main purpose was to eliminate the harmonics distortion effects due to the imperfect harmonics specifications of the HP8644A signal generator. Software-based filtering has the benefit of being fairly simple in implementation, while in combination with a FFT procedure prior to the filtering application. In the same time it provides very flexible and accurate control over the filter parameters. This was very important in our measurements, where the demands were in excess of 100 dB passband/stopband difference. Also by

implementing the filtering operation in software, we eliminate one more external factor, which would otherwise add thermal noise and unknown degree of parameters uncertainty, if an analog device was used for the purpose prior to applying the signal to the board's inputs.

The Harmonics Filtering module operates on the real and imaginary spectral components arrays, produced from the FFT module. It first computes the magnitude of each spectral component using $PSD_n = X_n = \sqrt{X_n^2 + Y_n^2}$, for $n = 1 \dots N$, while assigning each newly computed PSD value to the corresponding array X index location. Then it searches through the array X and finds the signal Bin location, which normally has the greatest PSD value. Harmonics would appear in locations on the spectrum, which are multiples of the signal Bin location. By using this, the next step is to find the harmonics locations and either eliminate the harmonics influence completely or set their PSD value to be equal to the computed average noise floor value. This is illustrated on Figure 4-5.

Figure 4-5 Harmonics Filtering Module Functional Architecture

Since our setup provided very clear signal resolution, this scheme worked pretty well in combination with the FFT module. To illustrate this we provide two FFT-plots from the measurements. Figure 4-6 shows the FFT-plot of the signal prior to the harmonics filtering operation, while Figure 4-7 shows the signal spectrum after the filtering module. As it is seen from the Figures, the spectrum is preserved in its original shape and only the spectral lines of the harmonics are modified.

Figure 4-6 Unfiltered Signal FFT-plot

Figure 4-7 Filtered Signal FFT-plot

4.1.4 Aliases Removal Module

Similar to the harmonics effects on the measurements accuracy, the aliasing effects need to be considered. Aliases are those pseudo-spectral components, which result from the inherent modulo- 2π properties of the Fourier Transform. In a sampled system, only frequency components in the range $\{-\pi, \pi\}$ can be uniquely represented, since the sine and cosine functions employed in the transform between time and frequency domains are periodical with period 2π . Any frequency components that fall outside the fundamental frequency range $\{-\pi, \pi\}$, are folded into this spectrum and affect the purity of the recovered signal.

As with the input signal harmonics, the aliasing effects in our setup were due to the absence of filtering on the generated analog input signal prior to its application to the board's sampling subsystem. In fact, most of the aliased components in our case were higher order harmonics produced from the HP8644A signal generator, folding into the Nyquist spectrum of the board's ADC. Even though not so crucial for the SNR measurements, the effects of the harmonics and aliases severely restrict the accuracy of the SFDR parameter measurements. Let's take the example of trying to digitize a 2.03MHz sine wave with a $1V_{pp}$ output. In the frequency domain the fundamental should appear at 2.03MHz as shown in Figure 4-8.

Figure 4-8 Signal Characteristics

The fundamental of a pure 2.03MHz tone should appear as an impulse in the frequency domain at 2.03MHz. In a sampled system noise terms are produced near the sampling frequency called aliases. These aliases are related to the fundamental in that they are located at $\pm f_N$ around the sampling frequency as shown in Figure 4-9. So for a 2.03MHz fundamental and a 10MHz sampling rate, an alias term is created at 7.97MHz and 12.03MHz. A $(\sin(X))/X$ function shaping is also induced when sampling a signal. Aliases continue up through the frequency spectrum repeating around the sampling frequency and its harmonics (i.e., $2f_S$, $3f_S$, $4f_S$). Aliases will also be produced from the non-filtered harmonics of the input signal in the same manner as with the Fundamental frequency.

Since the highest spur (either harmonically or non-harmonically related) with respect to the fundamental will define the dynamic range of the system, it is used to determine the Spurious Free Dynamic Range of the converter. Now it is clear that if the aliases are not removed prior to calculating the SFDR, the last will result inaccurate since the Highest spur can be an alias.

Figure 4-9 Aliasing Characteristics of Sampled Systems

Aliases and harmonics are not problematic issues in real applications. They can be either removed with appropriate analog filtering prior to sampling the signal or, as shown here, in post-sampling software processing, since their locations are well known in advance. This justifies the harmonics filtering and aliases removal approach in our experimental setup, which would otherwise render our experimental results inaccurate.

The aliases removal module operates on the X array, produced after the processing done in the harmonics filtering module. Based on the Signal Bin location index, it calculates the aliases locations and deals with them accordingly. This is illustrated in Figure 4-10. At the starting point we have an array with 16 K elements and a known Signal Bin location. First we check if the aliases will fall onto the signal and/or harmonics locations on the spectrum. This will happen when the size of the FFT is a power-of-two multiple of the Signal Bin location, or in terms of frequencies, when the sampling rate is a power-of-two multiple of the sampled signal.

If for example one samples a 1.75MHz analog input signal with 14MHz sampling rate ($14\text{MHz} = 8 \times 1.75\text{MHz} = 2^3 \times 1.75\text{MHz}$), then from the above mentioned property of the aliases to occur at $\pm f_N$ around the sampling frequency f_S , we will have a Fundamental at 1.75MHz and harmonics at $2f_N = 3.5\text{MHz}$ and $3f_N = 5.25\text{MHz}$. For real signals, the resulting FFT spectrum is symmetrical around $f_s/2$, or the Nyquist Frequency and this will be the folding point. So for 7MHz highest representable, the aliases from the fundamental will be at:

$$7\text{MHz} - 1.75\text{MHz} = 5.25\text{MHz} \text{ and } 7\text{MHz} + 1.75\text{MHz} = 8.75\text{MHz}$$

For the first harmonic at:

$$7\text{MHz} - 3.5\text{MHz} = 3.5\text{MHz} \text{ and } 7\text{MHz} + 3.5\text{MHz} = 10.5\text{MHz}$$

And for the second harmonic at:

$$7\text{MHz} - 5.25\text{MHz} = 1.75\text{MHz} \text{ and } 7\text{MHz} + 5.25\text{MHz} = 12.25\text{MHz}$$

But these are the exact locations of the Fundamental and falling within Nyquist harmonics. Therefore with harmonics filtering, prior to aliases removal module, we have already eliminated the aliasing effects and there is no need at all to exercise aliases removal operation for those special cases, where the sampling frequency is a power-of-two multiple of the Fundamental frequency.

Figure 4-10 Aliases Removal Module Algorithm

For all other cases we proceed with the aliases removal operation. A loop is entered, where, based on the indexing scheme, the aliases are set in amplitude to be equal to the precomputed Average Noise Value (-103.4 dBFS for the ICS-650), so they would not affect the measured parameters.

4.2 SFDR and SNR Software Description

4.2.1 SFDR

The SFDR software has all the software framework functionality described in the previous paragraph. In its core lies the actual SFDR parameter calculation procedure. It is fairly simple to implement and derives the parameter according to the definition outlined in paragraph 3.1, Chapter 3. In all measurements, the maximum amount of data for a single capture was acquired. This for ICS-650 and ICS-152 is 0.5MB, which is equal to 32 times 16 KB of data. This way for every single SFDR parameter measured, an average over 32 sets of 16 KB of data value was obtained, to avoid statistical fluctuations to influence the results in any way.

The SFDR software comprised of two different parts. The first part was implemented on MS Visual C/C++ SDK in a form of a Windows NT Console Application. It included Data Capture, 16K FFT, Harmonics Filtering, Aliases Removal and SFDR Calculation subroutines. Its execution produced a file with two, 32-elements arrays containing the signal and correspondingly Highest Spur values for each individual 16K set of the 0.5 Meg captured data.

The second part of the SFDR software was implemented as a Matlab script file, which was responsible for calculating the average SFDR parameter, based on the data produced from the Console Application, and visualizing the results on the screen. The console application was executed from within Matlab environment by a call to the system kernel. The interaction between the two software parts is illustrated on Figure 4-11.

Figure 4-11 Interaction Between the SFDR Testing Software

The actual SFDR parameter calculation was done in the following way: For each individual 16 K data set produced from the aliases removal subroutine, the Signal magnitude value was saved in a temporary array, based on the Signal Bin location. Then, the Signal value in the original array was set to zero and the next highest in amplitude value was searched for. This value would

constitute the Highest Spur value and was saved in another temporary array. Working this way for all 32 sets of 16 K data, finally, the two temporary arrays containing the Signal and Highest spur values were normalized, converted to dB values and saved in a file, which was loaded into Matlab. In Matlab environment the two sets of data were averaged and then subtracted from, to produce the final value for the SFDR parameter, in accordance with the definition outlined in Chapter 3.

4.2.2 SNR

The SNR software, similarly to the SFDR, has all the software framework functionality described in paragraph 4.1. In its core lies the actual SNR calculation procedure, which derives the parameter according to the definition outlined in paragraph 3.1, Chapter 3. Again, in all measurements, the maximum amount of data for a single capture was acquired (0.5MB = 32 times 16 KB of data). This way for every single SNR parameter measured, an average over 32 sets of 16 KB of data value was obtained, to avoid statistical fluctuations to influence the results in any way.

The software was implemented in a similar fashion as with the SFDR, with the only difference in the actual SNR calculation subroutine. Calculations were done with the use of the following equation:

$$SNR = Signal - \sqrt{\sum_{i=1}^{8191} ArrayX_i^2 + ArrayY_i^2}, \text{ in dBFS} \quad (21)$$

For each individual 16 K data set produced from the aliases removal subroutine, the Signal magnitude value was saved in a temporary array, based on the Signal Bin location. Then, the Signal value in the original array was set to zero and all components were summed in magnitude. The result from this operation was then squared to produce the rms Noise value, which was saved in another temporary array. Note that the effects of the harmonics were excluded by the harmonics filtering routine and the dc component was also excluded from the sum according to the definition for the SNR parameter outlined in Chapter 3. After normalization, and conversion to dB values, the Signal and Noise temporary arrays were saved to a file, which was loaded into Matlab. There, the two sets of data were averaged and then subtracted from, to produce the final value for the SNR parameter.

4.3 Matlab specific and other modules description

In this paragraph we present some of the Matlab script files that were used in combination with the C/C++ developed software as well as outlines of the software for the other experiments that were made with the ICS boards. Due to the simplicity of the Matlab files they are presented in their original version with comments on the side.

4.3.3 FFT plots - related

```
clear;           // Clears the current variables
!EFFT;          // System call for execution of the FFT plot Windows NT Console Application
load data01.dat; // Loads the data file produced from the Console Application
x = 0:30/8192:(30-(30/8192)); // Establishes the x-axis in frequency (0 through 30MHz)
```

```
%x = 0:8191;      // Or alternatively the x-axis in Frequency Bin format
plot(x,data01);    //
xlabel('Frequency in MHz');    // Visualization
ylabel('Power Spectral Density in dBFS');    // and
grid;              // Plotting
```

4.3.1 SFDR - related

```
clear;           // Clears the current variables
!ESFDR;          // System call for execution of the SFDR Windows NT Console Application
load data01.dat; // Loads the data file produced from the Console Application
for i=1:32,      // Establish a loop
    signal(i) = data01(i,1);    // Read the Signal values
    spur(i)   = data01(i,2);    // Read the Highest Spur value
end             // End of loop
S = 0;          // Initialize to zero
SP = 0;         // Initialize to zero
for i=1:32,     // Establish a loop
    S = S + signal(i);    // Accumulate the Signal values
    SP = SP + spur(i);    // Accumulate the Highest Spur values
end             // End of loop
SFDR = ((S/32) - (SP/32));    // Compute the average SFDR parameter
x = 0:31;       //
plot(x,signal,'o:',x,spur,'x:');    //
xlabel('Samples Index');    // Visualization
ylabel('Signal versus Worst Spur in dBFS');    // and
Title(['Average SFDR = ',num2str(SFDR),' dB']);    // Plotting
grid;           //
```

4.3.2 SNR - related

```
clear;           // Clears the current variables
!ESNR;           // System call for execution of the SNR Windows NT Console Application
load data01.dat; // Loads the data file produced from the Console Application
for i=1:32,      // Establish a loop
    signal(i) = data01(i,1);    // Read the Signal values
    noise(i)  = data01(i,2);    // Read the Noise value
end             // End of loop
S = 0;          // Initialize to zero
N = 0;          // Initialize to zero
for i=1:32,     // Establish a loop
    S = S + signal(i);    // Accumulate the Signal values
    N = N + noise(i);    // Accumulate the Noise values
end             // End of loop
SNR = ((S/32) - (N/32));    // Compute the average SNR parameter
x = 0:31;       //
plot(x,signal,'o:',x,noise,'x:');    //
```

```
xlabel('Samples Index');           // Visualization
ylabel('Signal versus Integrated Noise in dBFS'); // and
Title(['Average SNR = ',num2str(SNR),' dB']);    // Plotting
grid;                                           //
```

4.3.4 Analog Input Bandwidth

The software developed for these measurements consisted of only one part. The Windows NT Console Application included Data Capture, 16 K FFT and Signal Average modules. Harmonics filtering and aliases removal operation were not needed for these type of measurements, since they do not affect the accuracy of the results. Again, 0.5 Megs of data were captured and transferred into system memory via the PCI-bus. Then each 16 K data set was put through 16 K FFT after which the Signal Bin location was found, based on magnitude comparison between all spectral values. The Signal values were saved in a temporary array and after the last 16 K set of data was processed, the 32 elements Signal array was averaged over to produce a final value for the measured signal amplitude. This value was then normalized, converted to dB and printed on the screen. The actual measurement cycle would proceed according to the following scenario:

- 1). For a preset sampling rate, measurements started with a constant amplitude, close to dc in frequency, analog input signal.
- 2). The Console Application was executed to produce an average value for the Signal amplitude, which was recorded.
- 3). Moving up in frequency, the Console Application was executed number of times, until the produced average Signal amplitude was 3dB below the first recorded one.
- 4). The Analog Input Bandwidth in this case would be the difference in frequency between the final and starting frequencies of the analog input signal.

4.3.5 Channels Crosstalk

Similarly to the Analog Input Bandwidth, the Channel Crosstalk software consisted of a single Windows NT Console Application. In fact, the same application was used for these measurements with a different experimental setup. Input signal was applied to CH2 of the board, while CH1 was left undriven. After capturing 0.5 Megs of data, the application worked on CH1 data with CH2 data discarded. The measurements were done in the following way:

- 1). For a preset sampling rate, measurements started with a constant amplitude (-1 dBFS), close to dc in frequency, input analog signal.
 - 2). The Console Application was executed to produce an average value for the Crosstalk signal amplitude in CH1, which was then recorded.
 - 3). Similar measurements were done for equidistant in frequency input signals to cover the whole Nyquist Range of the board for the current sampling rate.
 - 4). The Channels Crosstalk value in this case would be the difference in amplitude between the analog input signal applied to CH2 and the corresponding recorded amplitude on the spectrum in CH1.
-

Chapter 5

Experimental Results

The general assumption for all measurements was that the ICS boards design does not significantly degrade the performance specifications of the ADC used on the board (Analog Devices AD6640). Two key parameters were the subject of the ICS-650 initial evaluation - Spurious Free Dynamic Range (SFDR) and Signal to Noise Ratio (SNR) of the device. Later, results were obtained for Analog Input Bandwidth and Channels Crosstalk parameters. Evaluation was also done in a similar fashion for the ICS-152, 14-Bit board.

5.1 SFDR-related

The SFDR parameter was measured in a phase-locked, non-windowed experimental environment (i.e. coherent sampling of the applied analog signal), according to the definitions, outlined in Chapter 3. Measurements were made over the whole frequency range of the boards as well as for most sampling rates and signal levels. The following sections of this report present the essential results obtained.

5.1.1 ICS-650 SFDR performance

SFDR was measured in the following way. First, data was captured in 32 sets of 16K samples (total size 0.5MB). Then a 16K FFT was performed on each block of data, followed by filtering of signal harmonics and aliases removal. After normalization and conversion to decibel levels, the SFDR parameter was derived by subtracting the amplitude of the highest spurious signal found from the amplitude of the fundamental signal. Finally, results were averaged over the 32 data sets.

We start with several FFT plots obtained from the experiments. They show the device performance over the Nyquist frequency range, as determined from the particular sampling rate used.

Figure 5-1 is a collection of 6 FFT plots from the measurements with 60 Msps sampling rate. The signal level was kept constant at -1dBFS during all captures. It can be seen that:

- 1). The analyzed signal, as expected, consist of a single spectral line and is at the appropriate amplitude level (-1dBFS) for all measured input frequencies.
 - 2). The analyzed noise floor of the board is relatively flat over the Nyquist frequency range, except in a close proximity of the signal.
 - 3). Spurs are randomly distributed along the spectrum and in amplitude, but seem to increase in density and amplitude with the frequency of the analyzed signal
 - 4). There is a non-uniform, increased noise distribution around the analyzed signal, which seems to get worse with frequency.
-

Figure 5-1 FFT plots from measurements @ 60 Mps and PSD = -1dBFS

The last-mentioned point is in sharp contrast with the expectation of the board's performance. For this reason several other measurements were done at lower sampling rates to test the consistence of this performance shortcome. They showed that this phenomenon was in effect

present, regardless of the sampling frequency on the board's ADC.

Next we performed the same measurements with smaller input signal amplitudes. Figures 5-2 and 5-3 illustrate this.

Figure 5-2 FFT plots from measurements @ 60 Msp/s and PSD = -10dBFS

Figure 5-3 FFT plots from measurements @ 60 Msps and PSD = -21dBFS

From these plots, one can note, that lower signal amplitudes resulted in improved overall ISC-650 board performance over the whole Nyquist range. The increased noise distribution around the analyzed signal is almost non-visible and the same can be said about most of the

higher amplitude spurs seen in Figure 5-1. To be more accurate, they are still present, but lower in amplitude and in effect hidden into the noise floor of the device. The exact cause of this phenomenon was found to be related to the Phase Locked Loop (PLL) used on most of the ICS boards. Proof of that will be given at the end of this paragraph, but for now we proceed with the SFDR measurements results.

As one can expect from the above Figures, the SFDR performance will closely match the dependence of the noise distribution around the analyzed signal with frequency. The reason for this is, that the noise distribution in most cases will determine the Highest Spur value, which is used in the SFDR calculations.

The first measurements presented in this paragraph are from the experiments with 60 Msps sampling rate on the board. Figure 5-4 is the result from 31, equidistant frequency point measurements with -1dBFS input analog signal amplitude.

Figure 5-4 SFDR performance for 60 Msps and -1dBFS input signal

The solid line on the above plot represents the SFDR parameter with frequency. The performance is frequency selective and degrades as we move up in frequency. This is a potential problem for any software radio design, based on this device, since the software radio concept demands uniform performance throughout the frequency range. But, as noted earlier, this is the result of the under-designed, in this case, PLL on the board, which is the main source of the increased noise distribution around the analyzed signal. On the next page are similar plots for signal amplitudes -10 dBFS and -20 dBFS respectively.

Figure 5-5 SFDR performance for 60 Msps and -10dBFS input signal

Figure 5-6 SFDR performance for 60 Msps and -20dBFS input signal

As in the case with the FFT plots from Figures 5-1, 5-2 and 5-3, the SFDR plots show an improvement in terms of uniform performance over the Nyquist range, but with the cost of decreased Spurious Free Dynamic Range values.

Some measurements, made with 40 Msps, are also presented in Figure 5-7 below. They show similar device performance.

Figure 5-7 SFDR performance for 40 Msps

Finally, measurements were made to investigate the SFDR performance with amplitude of the input analog signal for several different frequencies. For this purpose measurements were made with 60 Msps sampling rate on the board, for three frequencies - close to dc at 2.34375 MHz, around the middle at 12.65625 MHz and close to Nyquist at 25.3125 MHz. Figure 5-8 has the details for each measurement. The decrease in value for the SFDR parameter was relatively steady along the 80dB reference line, except for the parts with high amplitudes and higher frequencies of the input signal. This again is in accordance with the earlier results, where dependence with frequency was established due to the on-board PLL design. Similar plots are also shown for 40 Msps sampling rate on Figure 5-9 for two input signal frequencies - 4.375 MHz and 18.75 MHz.

Figure 5-8 SFDR Performance with Amplitude @ 60 Mps

Figure 5-9 SFDR Performance with Amplitude @ 40 Mps

To conclude, we base our evaluation on the results from the measurements with 60 Mps sampling rate for the reason, that converter technologies are usually operated at their full capacity.

Overall, the following numbers and conclusion points can be established:

- 1). *Worst-case Spurious Free Dynamic Range for a full-scale input analog signal was around 70 dB.*
- 2). *SFDR decreased consistently with decreasing input signal amplitude, except for signals with relatively large amplitudes and higher frequencies.*
- 3). *To a great extend, the above-mentioned shortcomings in the device performance were the result of the on-board PLL.*

The last point needs clarification and here we provide the promised in the beginning proof. Having noticed the anomaly with the increased noise floor around the analyzed signal in the very early stages of the measurements, we consulted some of the ICS-650 design engineers about a possible cause for this phenomenon. They suggested that we make some measurements with the on-board PLL bypassed and see if there will be any difference in the performance.

Bypassing the PLL is fairly easy, since the IC has the feature integrated in its design. Unfortunately, without the PLL, the on-board ADC clock rate is one-half of the supplied to the board externally clock signal. With our clock driving (HP3335A) generator range of 200 Hz to 81 MHz, this meant that measurements with sampling rates up to 40 Msps can only be made.

In the following pages we provide some FFT plots taken from these measurements. We start first with the 40 Msps results. On Figure 5-10 are some FFT plots with signals at 2.1875 Mhz, 4.375 MHz, 6.25 MHz and 13.75 MHz accordingly.

Figure 5-10 FFT plots @ 40 Msps and PLL bypassed. PSD = -1 dBFS

From the FFT plots on Figure 5-10 we note, that the increased noise distribution around the signal is now not present. We also note that there is an increase in the spurs density and magnitude value in comparison with the FFT plots from the normal experiments (with the PLL active). This can be explained with the fact, that without the PLL buffering functions, generator phase noise and clock jitter affect the quality of the recovered signal in much stronger fashion, comparing with the occasions when the PLL is actively engaged and locked onto the clock signal.

On the other hand the increased noise distribution with frequency (when the on-board PLL is active) can be explained with the inability of the PLL to lock itself exactly onto the supplied clock signal from the generator for a relatively long period of time. PLL devices usually work in two stages - signal acquisition and signal tracking. With higher signal frequencies, it becomes much more difficult for the PLL to track the signal with accuracy. This will result in a narrowband jitter process, which will also have a narrowband effect on the recovered signal spectral distribution due to the convolution of the two spectra in the frequency domain (as explained in Chapter 3, Coherent Sampling section).

When the PLL is bypassed, the narrowband jitter process is no longer present and the device has to deal with the much broader phase noise distribution of the generator. The resulting effect is evident from the FFT plots on Figure 5-10, where the spurs are much higher in amplitude and density across the entire spectrum. This was observed to be the case, regardless of the converter sampling rate, and is illustrated on Figure 5-11 with several plots @ 10 Msps and no PLL.

Figure 5-11 FFT plots @ 10 Msps and PLL bypassed. PSD = -1 dBFS

5.1.2 ICS-152 SFDR performance

SFDR for the 14-Bit converter hardware was measured in exactly the same fashion as for the 12-Bit board (ICS-650). Due to the architecture similarities and use of the same type of PLL component on both boards, the increased noise distribution around the analyzed signal with frequency was noticed to be present in ICS-152 as well. To compare the overall performance of the two boards, some FFT plots from measurements at 70 Msps are presented in Figure 5-12.

Figure 5-12 FFT plots from measurements @ 70 Msps with ICS-152

It is easy to see the improved noise floor performance over the 12-Bit board (compare with Figure 5-3). The SFDR gain, as expected, is less than the theoretically derived 6 dB per added bit resolution, but still clearly noticeable and relatively uniform across the Nyquist range. Considering the above-mentioned PLL issue, we expected the SFDR performance of the ICS-152 to get worse with frequency and similar to the one of the ICS-650. Figure 5-13 presents some graphical results derived from the ICS-152, SFDR measurements.

Figure 5-13 SFDR performance for 70 Msps and various signal amplitudes

The SFDR performance for ICS-152 improves in terms of uniformity over the Nyquist range with lower signal amplitudes. The average overall gain when comparing to the 12-Bit board is around 7 dB for all signal levels and frequencies. This is not a satisfactory value for a 2-Bits increase in resolution. The difference in gain is due to the fact that spurs, by nature, are random phenomena, depending on many system parameters. Indication of this is the deviation, from the theoretically expected, straight line performance with frequency - best seen in the top-left graph in Figure 5-13. One can expect more uniform and better results in terms of SNR performance due to the integration of the noise terms. Large deviations from the average noise floor (as are the spurs) will now play non-dominant role, since those are rather small percentage, and will be easily absorbed in the integration.

5.2 SNR-related

The SNR parameter was measured in a phase-locked, non-windowed experimental environment (i.e. coherent sampling of the applied analog signal), according to the definitions, outlined in Chapter 3. Measurements were made over the whole frequency range of the boards as well as for most sampling rates and signal levels. Since the FFT-type analysis performance was already presented in section 5.1, here we include only the essential graphical results from the SNR parameter measurements for both boards.

5.2.1 ICS-650 SNR performance

SNR was measured in the following way. First, data was captured in 32 sets of 16K samples (total size 0.5MB). Then a 16K FFT was performed on each block of data, followed by filtering of signal harmonics and aliases removal. After normalization and conversion to dB levels, the SNR parameter was derived by subtracting the integrated noise amplitude value from the amplitude of the fundamental signal. Finally, results were averaged over the 32 data sets for each point.

Figure 5-14 SNR performance for 60 Msps and various signal amplitudes

Figure 5-14 shows the SNR performance of ICS-650 over the Nyquist range for three different signal amplitudes. The SNR parameter deviation across the range is smaller and levels off

much faster, compared to the SFDR performance for the same board. This, as stated earlier, is due to the integration of all the noise terms components, when deriving the SNR parameter value. The effects of the highest spurs, being only a small percentage of all the noise terms, are easily absorbed into the integrated noise value. This is rather important observation, since it shows that SNR is much more flexible and controllable parameter than SFDR. One can improve on SNR in various ways - by extracting and processing smaller signal bandwidths, deliberately downconverting the desired signal bandwidths to an appropriate spectral locations (close to dc, where the device performance is better) or even with increasing sampling rates and/or converter resolution.

5.2.2 ICS-152 SNR performance

SNR for the 14-Bit converter hardware was measured in exactly the same fashion as for the 12-Bit board (ICS-650). Figure 5-15 outlines the ICS-152, SNR performance across the Nyquist range and several different signal levels at 70 Mps sampling rate on the board.

Figure 5-15 SNR performance for 70 Mps and various signal amplitudes

When compared to the ICS-650 performance (Figure 5-14), the 14-Bit board shows similar SNR characteristics, with the only major difference being the performance gain from the increased converter resolution. This is not unusual, since both boards share similar architecture and hardware components. In general, the SNR parameter uniformity improves with small amplitudes, but much faster than the SFDR.

5.3 Input Bandwidth-related

Input Bandwidth characteristics are of great importance to the communication industry for the fact that in some cases it is much more beneficial to perform bandpass sampling (in terms of Intermediate Frequencies), rather than the usual downconversion to baseband and sample operation. If the analog input signal range is from dc to $F_S/2$, then the amplifier and the filter combination must perform to the specification required. However, if the signal is placed in the third Nyquist zone (F_S to $3F_S/2$), the amplifier is no longer required to meet the harmonic performance required by the system specifications, since all harmonics would fall outside the passband filter. For example, the passband filter would range from F_S to $3F_S/2$. The second harmonic would span from $2F_S$ to $3F_S$, well outside the passband filter's range. The burden then has been passed off to the filter design, provided that the ADC meets the basic specifications at the frequency of interest. In many applications, this is a worthwhile trade-off since many complex filters can easily be realized using SAW and LCR techniques alike at these relatively high IF frequencies.

Due to these reasons, we analyzed the board's Input Bandwidth performance in the following way (as outlined in Chapter 3, Section 3.1.3.). To start with, we used close to dc in frequency analog input signal. It was digitized by the board and data captured in quantities of 0.5 MBytes. After processing each 16 KByte block of data, the signal amplitude value in dBFS was extracted and averaged over the 32 sample sets. The averaged value was recorded as a starting point and then we proceeded in a similar fashion until an average value, approximately 3dB below in amplitude, was recorded. Then, the analog Input Bandwidth is easily derived by subtracting the starting signal frequency from the frequency of the signal which produced the final amplitude value. Figure 5-16 illustrates the ICS-650 measurement results for several signal amplitude levels.

Figure 5-16 Analog Input Bandwidth for 60 Msps and various signal levels

Unlike the SFDR and SNR parameters, the board's Analog Input Bandwidth parameter is not affected from underdesigned hardware components like PLLs or Local Oscillators. As seen from Figure 5-16, the performance is identical for the three different signal amplitude measurements. An average estimation for this parameter was derived to be around 95 MHz per channel, regardless of the input signal level. This makes the total (complex signal sampling) Input Bandwidth twice that - 190 MHz. Note that the drop in amplitude with frequency is non-linear. In fact it is less than 1 dB up to 60 MHz, which more than twice doubles most of today's mobile communications operational bandwidths. Almost identical results were obtained for the 14-Bit board, with slightly better overall Analog input Bandwidth parameter value - 195 MHz.

5.4 Channels Crosstalk-related

Channels Crosstalk is also a very important parameter since it is directly related to the quality of the recovered signal, when both channels on the converter are employed in the system operation. To benefit from the full capacity of the digital receiver it is desirable to use both channels on the board in the process of digitizing the desired signal. There are many ways in which the full system capacity can be utilized, but in any case, there will be an unavoidable mutual interference between the 2 channels since they coexist in a single piece of hardware and share most of the hardware's resources and functionalities. This interference is often related to as Channels Crosstalk and is an indication of how the processed signal in one of the channels affects the quality of the recovered signal in the other channel.

To measure the Channels Crosstalk parameter, we applied an analog input signal at one of the channels and in the same time captured data from the other channel, which was undriven and short-circuited to avoid influence from any parasitic voltages on the board. Figure 5-17 shows the results of this experiment at 60 Msps.

Figure 5-17 ICS-650 Channels Crosstalk performance at 60 Msps

For these measurements, each data point on the above graph was derived by averaging 5, independently captured, crosstalk data values. The worst-case recorded channel interference separation was around 74 dB. This is almost equal to the worst-case spurious performance of the device, thus making the Channels Crosstalk performance an attractive area for improvement.

It is interesting to note that most of the frequency data points, that produced low channel separation levels, coincide with the “power of two” signal bin locations on the spectrum. For example, the worst-case Channels Crosstalk parameter was recorded for signal at 15 MHz (Bin 4096), with points at 7.5 (Bin 2048) and 22.5 (Bin 6144) MHz being close in values. This can be explained with the fact, that in these cases (as explained in Paragraph 4, Section 4.1.4), all or most of the aliased components will fall onto the recovered signal spectral location. Due to the nature of the filtering and aliases removal modules, used in our measurements, to operate on the derived from the FFT routine spectrum, it is impossible to separate signal from aliases in this case. The aliased components will then boost the PSD in the evaluation location, which at these small levels will be quite noticeable. To combat this effect, proper digital or analog filtering prior to the FFT is required (opposed to the post-FFT approach used in these measurements).

The Channels Crosstalk parameter for both boards under investigation improved in a linear fashion as amplitude level of the applied analog signal decreased and was noticed to be relatively independent from issues like underdesigned PLLs or Local Oscillators. Results for the 14-Bit board are illustrated in Figure 5-18.

Figure 5-18 ICS-152 Channels Crosstalk performance at 70 Msps

There is a visible improvement of around 6 dB. Apart from this difference, both boards showed identical performance of having their Channels Crosstalk parameter close in value to the worst-case SFDR, which got better with smaller signal amplitudes.

Chapter 6

Performance Evaluation

Before any conclusive results can be established from this investigative work, there is a strong need for a thorough understanding of all factors that may have affected the research credibility. The purpose of Section 6.1 is to disclose all known sources of impairments during the experiments and to extract an estimate of the results credibility. Finally, Section 6.2 compares the two receivers in terms of their converter-specific performance.

6.1 Sources of impairments and results accuracy

Knowing the sources of impairments is important in any “to-be-trusted” experiment. In fact, often experiments begin with a careful analysis of all factors and elements of uncertainty, that may undermine the credibility of the results and the true device performance. While every effort is directed to eliminate their effects from the very beginning, some may still be present throughout the experiments. Only by having a thorough knowledge and understanding of the impairments sources, one can judge the extend of their presence on the investigated device performance and account for their influence when deriving the final results of the work.

In our case, sources of impairments were found to exist from various factors. By a first approximation they can be categorized in several subcategories:

6.1.1 Generator-specific.

As stated in Chapter 3 (Section 3.2), two HP signal generators were used for signal and clock generation in order to drive the boards during the experiments. Being physical devices themselves, they have their performance limitations. The main sources of impairments from the HP generators were the non-ideal spectral purity performance and phase noise characteristics. Even though they are interrelated, each has a different effect on the receiver under testing.

Consider the HP8644A, used for signal generation. Its harmonics level specifications are < -25 dBc (see Appendix 1), which is somehow an upper bound, but still, it demanded harmonics filtering to derive the SFDR and SNR parameters. With the post-FFT filtering approach used in our software blocks, harmonics and folded into the Nyquist aliases related to them, were quite high in amplitude and required removal. In some cases, where aliases happened to fall onto the analyzed signal bin location (as explained in Chapter 5, Section 5.4), aliases removal for this particular location was not possible and thus the signal value was affected, depending on the phase/magnitude characteristics of the affecting aliases. This was well known in advance, but was considered to have a minor influence on the measurements. It primarily affects the fundamental and only to a small degree. For example if the fundamental is at -1 dBFS and the alias is at -60 dBFS, then when we sum them in magnitude we have:

$$(-1\text{ dBFS}) \pm (-60\text{ dBFS}) = 9\text{ dBm} \pm (-50\text{ dBm}) = 7.9\text{ mW} \pm (0.00001\text{ mW}) \approx -1\text{ dBFS}$$

Even several aliases with same phase/magnitude characteristics would not affect the fundamental in such a way as to influence more than a fraction of a dB impairment to the derived parameters.

The spurious performance of HP8644A (-105 dBc) was considered to be far better than what's expected from the devices under investigation and was excluded as a source of impairments on the supplied signal.

The next factor is the generator's phase noise (-136 dBc/Hz @ 1GHz). Phase noise is a nonuniform distribution around a local oscillator and usually improves as you move farther away from the tone. Since we have a sampled system we evaluate the phase noise at preset, equidistant frequency points, the number of which depends on the FFT length. With a conservative estimate of a flat, -136 dBc phase noise generator performance we have integrated phase noise power of:

$$P_{PN} = 10\log 8192 - 136 = -96.87\text{dB} \quad (22)$$

As seen, this is not a potential impairment factor, since the above value is few tens of dB smaller than the total noise power of the receiver (-64.28dBFS, as derived in Chapter 3, Section 3.3). Again, taking on account that phase noise improves with distance from the fundamental and that the actual -136 dBc/Hz value is specified for 1GHz (way up in frequency, compared to the usual generated signal frequency range of up to 30 MHz in our experiments), we considered that to be a non-degrading issue. This is not a problem in real applications where the signal is not synthesized from a generator and where one would normally use smaller bandwidth channel processing. If for example we process a GSM channel (200 kHz bandwidth), then:

$$P_{PN} = 10\log\left(\frac{200 \times 10^3}{30 \times 10^6} 8192\right) - 136 = -118.63\text{dB} \quad (23)$$

Now we move our attention to HP3335A, used for generating the clock signal. Here we deal with the same sources of impairments such as harmonics, spurs and phase noise, but they have different effects on the devices performance. As explained in Chapter 3, our setup involved sinewave-to-TTL compatible voltage levels transformation buffer and a consequent line driver. Further down in the link there was the on-board PLL. Each component added jitter, while regenerating the clock signal. The effects of the non-ideal harmonics and spurious generator specifications were in fact eliminated from the comparator by generating a new, TTL-level compatible signal, based on the properties of the supplied sinusoidal signal. The effects of the phase noise, though cannot be eliminated.

Comparators are linear devices without any embedded logic that could track changes in state. Any phase noise on the sinewave translates into jitter onto the produced TTL-level signal. Moreover, comparators have their own subculture of critical specifications, which may affect the signal quality. Propagation delay (the lag between the ideal sampling instant and the actual sampling time), is not a factor in communications applications because the sampling action is repeti-

tive. Pulse skew (defined to be the difference between the propagation delay of the high-to-low and low-to-high transitions), on the other hand, can contribute to increased jitter, since sampling instants at the ADC are commanded by either rising or falling pulse edges. Although skew is statically distributed, or time-invariant, you also must contend with the dynamically distributed, or time-variant jitter. In time-variant jitter, the signal edge varies from one cycle to the next, thus increasing the static delays and broadening the uncertainty in clock-edge arrival time at the ADC (see Figure 6-1). This automatically results in sample instant errors, which grow in magnitude with higher signal frequencies as discussed in Chapter 3, Section 3.1. (See Figures 3-1 and 3-2). The resulting errors distort the recovered signal spectrum in a form of more spurs and increased noise distribution around the analyzed signal location. This affects both SFDR and SNR parameters, especially at higher input signal frequencies.

Figure 6-1 Pulse skew increases overall jitter in the system

6.1.2 Receiver-specific.

The receiver boards contained several sources of impairments, which contributed to the already present sources from the generators. In terms of the supplied analog signal, there were several stages of impedance matching, prior to the actual encoding from the ADC on the board. Due to imperfect components physical characteristics and specification tolerances, ideal impedance matching is impossible and some of the signal energy was lost in the form of thermal noise into the signal channelization hardware on the board. This affected the visualization tools precision, but had little effect on the derived system performance. Losses were none-frequency selective and affected the rest of the spectral components in much the same way as the fundamental.

The supplied clock signal had to undergo an impedance matching and phase recovery prior to being distributed to several units on the receiver, including the ADC clock input. Proper termination, again was not an issue, since the TTL standard allows variation in voltage levels to a much greater scale than what usually resulted from the imperfect impedance match on the board's clock input.

More attention is directed towards the next stage in the supplied clock signal channelization hardware - the on-board PLL. Its primary goal is to derive an estimate of the supplied to the unit clock signal and on the basis of that estimate to generate a clock signal for distribution to various units along the link. As explained in Chapter 5, Section 5.1.1, the on-board PLL was found to be the main source of impairments in the experiments. It was responsible for the nonuniform increased noise distribution around the analyzed signal, along with creating more spurs throughout the spectrum. Its effects on the recovered spectrum are strongly interrelated with the supplied analog signal frequency. The higher the analog signal frequency, the worse the device performance. Higher frequency signals have increased slew rates and result in greater sample magnitude errors in a sampled system, where jitter is present on the encode clock. This is best illustrated in Figure 3-2. Also, with less time-domain evaluation points per period for higher frequencies signals, the jitter effects are much more pronounced, when compared to lower frequency signals. One can judge about the extend of the on-board PLL related impairments from the performance degradation in the SFDR plots in Chapter 5, Section 5.1.1. With signal at full scale, this degradation was recorded to be close to 18 dB at 60 Msps sampling rate on the ADC clock.

6.1.3 Converter-specific.

We begin our converter analysis with the traditional differential nonlinearity (DNL) and integral nonlinearity (INL) parameters. DNL indicates the worst case deviation from an ideal step size between adjacent codes along the transfer curve. Increases in DNL, which generally occur with higher input frequencies, appear as an increase in quantization noise and a rise in the converter's noise floor. Typical value for AD6640 is $\pm 0.5LSB$, while the maximum and minimum specified values are double that.

INL measures worst case deviation from the ideal, straight-line converter transfer function (as drawn through the transfer-function endpoints or as a best-fit straight line through the data points). Such deviations, or bends in the straight-line transfer function (see Figure 6-2), result in the converter's generating harmonics and spurious frequencies that were not present in the original waveform being digitized. Typical value for AD6640 is $\pm 1.25LSB$.

Figure 6-2 Typical DAC transfer functions

6.1.4 Results accuracy

Relative to the on-board PLL, all other impairment sources can be considered to have limited impact on the accuracy of the acquired performance results. Nevertheless, the accuracy of the results needs to be estimated to allow for safety margins in any subsequent discussions. While proof that the experimental setup is capable of delivering accurate results was presented throughout this work, it is important to have a numerical estimate of the deviation from the actual performance caused by the physical imperfections of the testing apparatus.

Since the PLL-related impairments overwhelmingly dominate the performance degradation, we exclude them from the composite estimate for the following reasons:

- The PLL component in the case of the ICS-650 and ICS-152 boards was clearly unable to deliver the performance needed for such high speed applications. Many alternative design solutions are possible, the simplest of them being to eliminate the component altogether.
- In practice, multicarrier digital radio applications, the ADC must share its dynamic range with other RF carriers. For example, if eight carriers have equal power, each signal should have dynamic range not more than one-eighths of the total range, or -18 dBc if you consider peak-to-peak signals. As shown in Chapter 5, the increased noise distribution around the analyzed signal was almost non present at these levels and hence can be easily avoided.

With this on mind, our estimates can be broken down into several components, that are closely related to each impairment factor outlined in the previous sections.

1). The effect of non-ideal harmonics specifications of the signal generator

It has been shown that folding within Nyquist aliases, resulting from the non-ideal harmonics specifications of HP8644A, affect the fundamental in cases where they happen to fall onto the signal bin location, but only to a fraction of a dB. Therefore we estimate this source of impairments to be no more than 0,25 dB.

2). The effect of generator phase noise on the analog signal

The phase noise on the generated analog signal was shown to be smaller than the thermal noise of the system. Its integrated value was around 30 dB below the total noise present in the system, as derived in Chapter 3, Section 3.3. This makes it a non-dominating source with a conservative estimate of 0,25 dB.

3). The effect of jitter on the clock signal

This is the clock jitter, resulting from the non-ideal, HP3335A phase noise performance. Exact phase noise characteristics could not be established with great accuracy. We allow for a 2,0 dB degradation.

4). The effect of comparator pulse skew and buffer propagation delays

As discussed, pulse skew broadens the system jitter PDF. Propagation delays instabilities also add to this value. Our estimate is around 1,0 dB rms jitter added to the one already present from the generator.

5). The effect of ADC nonlinearities

Every added bit increases resolution with 6dB. DNL of $\pm 0.5LSB$ means $\pm 3dB$ small-signal error range and almost no large-signal errors. The same calculations apply for INL. With 32 averaged points for every measurement, we estimate this source to have less than 1,0 dB contribution on the final results.

In summary, we conclude that errors, resulting from the above listed impairment sources do not exceed 2 dB with respect to the experimentally derived performance for both receivers.

6.2 ICS-650 and ICS-152 comparison

To compare the performance of ICS-650 and ICS-152, it is best to summarize the results from Chapter 5 in a conclusive table.

#	Parameter	Conditions	ICS-650	ICS-152	Goals	Notes
1)	SFDR - Spurious free dynamic range	Signal at $\leq -1dBFS$ up to 30 MHz	$\leq -86dBFS$	$\leq -93dBFS$	$\leq -90dBFS$	At 60 Msps for ICS-650 and 70 Msps for ICS-152
2)	SNR - Signal to noise ratio	Signal at $\leq -1dBFS$ Nyquist BW	$\leq 60 dB$	$\leq 68 dB$	$\geq 75dB$	Excluding dc and first 5 harmonics
3)	Analog input bandwidth	Analog signal at -3 dB point	$\leq 95 MHz$	$\leq 98 MHz$	$\leq 100 MHz$	Per channel, for real signal processing
4)	Channels Crosstalk	One driven at -1dBFS the other monitor	$\geq 74 dB$	$\geq 80 dB$	$\geq 70 dB$	At 60 Msps for ICS-650 and 70 Msps for ICS-152

Table 2: ICS-650 and ICS-152 performance comparison

The presented results in the above table refer to the best performance obtainable during the measurements. This is especially valid for both SFDR and SNR parameters, which characteristics throughout the Nyquist frequency range were strongly affected by the on-board PLL issue. The numbers under the "Goals" column were taken from a work sheet and reflect the communication industry proposed needs in terms of digital receivers.

Starting from the SFDR parameter, we see an improvement of around 7dB for the 14-Bit board, compared to the one obtained from the 12-Bit converter. This improvement is based on results from close to dc analog input signals, where the boards have their best performance, but in general, this value was consistent throughout the frequency range. The improvement of 7dB is 5dB short of the theoretically expected 12dB for 2 added bits of resolution. Taking on account,

that at the time of measurements, both 14-Bit ADC and the ICS-152 hardware around it were still in a preliminary stage, this is not a surprise. Note that, with the ICS-152, the proposed -90dBFS SFDR value (specified in the “Golas” column), is accomplished, while the one for ICS-650 is very close to meet this criteria.

The Signal-to-Noise Ratio parameter for the 14-Bit board also presents an improvement over the 12-Bit one. The actual value being around 8dB is derived from the difference in both boards performance. Like the SFDR, the SNR improvement was consistent throughout the frequency range. The 60dB value for ICS-650 reflects its best performance and is 4dB short of the board’s integrated noise value with no input signal (64.28dB), as derived in Chapter 3, Section 3.3. This is solid proof that the total noise in the system increases in the presence of applied analog signals. Part of this increased noise is due to the non-ideal signal generator specifications, like harmonics and phase noise, while the rest is due to the signal energy leakage from the imperfect sampling system (generator clock signal jitter, PLL instability, etc.). This is close to our estimate from the previous section of 5% results accuracy.

The goal value of 75dB is not accomplished with either board. We wish to note, that from the worksheet it was taken from, it was not clear for what bandwidth this value was specified. This required SNR is above the specified SNR capabilities for the AD6640 (used in ICS-650) and just matches the AD6644 (used in ICS-152), SNR specifications. The amount of bandwidth, over which the noise is integrated, when deriving SNR makes a difference. With smaller bandwidths we have less noise present and consequently the SNR improves as we saw in Section 6.1.1 (see equations (22) and (23)).

The Analog Input Bandwidth parameter was similar for both boards, with only a minor improvement of around 3dB for the 14-Bit board. As mentioned earlier, this parameter was derived for a single channel, real processed signal. To utilize the system capacity, one would normally perform complex signal processing, using both channels, thus making the discussed parameter twice the value specified in Table 2. Both boards meet the proposed goal of 100 MHz input bandwidth per channel, which makes them an ideal choice for bandpass sampling with all its benefits as discussed in Chapter 5, Section 5.3.

Channel Crosstalk for ICS-152 was found to be on average 6dB better than the one for ICS-650. This was mainly due to the increased dynamic range and better spurious performance of the 14-Bit board. Nevertheless, both boards provided very good channels isolation and meet the proposed 70dB channels separation goal parameter.

As an overall performance, ICS-152 meets or exceeds the proposed goal parameters with the exception of the SNR, while ICS-650 comes at a very close distance to achieving this goal. The improvement in performance of the 14-Bit board over the 12-Bit one, is evident, even though not as theoretically expected for the 2 bits increase in resolution. Both devices performance was steady throughout the experiments and well handled by the operating system. With the exception of the PLL issue, the receivers under investigation showed uniform performance throughout their frequency range and for different signal levels and mode of operation.

Chapter 7

Relating the experimental results to today's Wireless Standards specifications

Chapter 7 reflects the motivation behind this research and evolves around the issues presently faced in the integration of the software radio concept in today's mobile communication systems. Several wireless standard specifications are looked at closely to help obtain a detailed analysis of the current software radio technology ability to be supported commercially. For the purpose, the standards requirements are related to the experimental results, while taking on account each standard distinguishable features and differences. Several detailed examples are also provided to help understand the main limitations of software radios in the mobile communications environment. Section 7.1 deals with the GSM standard support and its up-band derivatives DCS-1800 and PCS-1900. Section 7.2 details the CDMA standard in general, with emphasis on IS-95. Section 7.3 concludes this chapter with evaluation of selected 3rd generation systems software radio support.

7.1 GSM support evaluation

The Global System for Mobile Communication (GSM) is undoubtedly the most successful second generation digital mobile radio system. It remains an evolving standard with many additional features, high service quality and ability to support data in various formats and rates. All these valued features have their impact on the system architecture. GSM is often being described as one of the most demanding standards in radio history. Its high band versions DCS-1800 and PCS-1900 are no exception, since they share the same GSM architectural design, while having to operate in worse propagation environment. We begin our discussion with the core GSM technology. It is fairly well documented and can serve as a starting point in the evaluation.

7.1.1 The Base Station Receiver

The term mobile communication in general means communication, which takes place between two mobile users, communicating over the common air interface via means of hardware transceivers. In today's mobile communication systems there are two transceiver types - the Base Station transceivers and the Handset transceivers. Each type has different capabilities, while serving the same purpose of transmitting and successfully receiving the communication signal. In the initial phase of this research a decision was made to base our work on the common Base Station Receiver architecture due to the following reasons:

- 1). High resolution, high speed converters still have large power consumption (710 mW typical for AD6640). Taking on account the surrounding components consumption, this will make the ADCs practically unusable at this time in software radio receiver user handsets, since this would limit severely the device talktime/standby parameters.

- 2). Another serious constraint is the computational power required for a seamless operation. Such DSPs are fairly expensive even in large quantities and consume a lot of power and space - two very limited resources in the handset design. While custom-tailored solutions with ASICs or FPGAs are still possible, they are not flexible enough, thus contradicting with the main benefit of a software radio design.
- 3). User handset transceivers usually operate on much smaller bandwidths compared with their BS counterparts. Straightforward application of the software radio concept would result in overdesign and would not be cost-effective at this time.
- 4). Due to limited physical resources and low-price marketing strategies, user handsets operate in much more relaxed regimes, thus relying on the system to compensate for their performance limitations. For example most GSM transmitters employ class "C" amplifiers, which have poor linearity. This results in reduced dynamic range, intermodulation products, non-uniform frequency performance etc. All these has to be compensated by smart processing at the BS.

On the other hand, Bases Station equipment is perfect candidate for software radio implementation. Most of the limiting for the user handset equipment factors, such as space, power and intense computational platform dependence, here are not an issue. Moreover, the BS transceivers normally operate in multicarrier/multichannel regimes, often serving a number of customers at the same time. With a software radio design, as discussed in Chapter 1, Section 1.1., the savings in hardware could be quite substantial. Last, but not least, is the benefit of increased flexibility. New accessing schemes and modulation/coding parameters can be updated with ease, simply by software uploads, and where necessary, by hardware modules interchange at the site - tasks much more difficult to perform on users handsets where bureaucracy and service cost is proportional to the number of subscribing customers in the area. Due to the above-mentioned reasoning, all subsequent evaluations will be related to the common BS receiver architecture.

7.1.2 GSM-standard receiver specifics

The GSM system operates in two paired bands: 890-915 MHz for uplink transmission and 935-960 MHz for downlink transmission. This translates to a bandwidth of 25 MHz in each direction. Therefore sampling rates, grater than 50 Msps are a must for any digital radio design implementation, with a more practical numbers being in the region 10 to 20% more. This requirement is easily met by today's high resolution converter technologies, which now operate in the 65-70 MHz range. This simple requirement, though, is only the beginning. Parameters such as receiver dynamic range and receiver sensitivity are much more important.

The Front End receiver performance is assessed in terms of a reference sensitivity level. This is the RF signal level at the receiver input assuming a specific propagation channel, for which the required error rate performance is achieved. For the hand portable receivers, the reference sensitivity level is -102 dBm, and for all other types of mobile and base station receivers this level is -104 dBm. The receiver blocking and intermodulation characteristics are specified assuming static propagation conditions for both wanted and unwanted signals, and a wanted RF signal level

at the receiver input equal to the reference sensitivity level. It is required that a mobile station receiver performance does not degrade by more than 3 dB when an interfering continuous wave signal is applied to the receiver at a level of -23 dBm or less, and that the frequency spacing between the wanted and unwanted signals are not less than 800 kHz and not more than 45 MHz. The requirements for the base station receivers are 10 dB higher, e.g., the level at which the 3 dB degradation in performance occurs is set at -13 dBm. In realistic applications this interfering signal will most likely be an adjacent channel, thus setting the upper bound for the BS receiver. From here it is easily established that the Front End dynamic range is derived to be:

$$(-13\text{dBm}) - (-104\text{dBm}) = 91\text{dB} \quad (24)$$

When receiving European digital cellular Global System for Mobile Communications (GSM) signals, a receiver at the BS must digitize signals between -104 dBm and -13 dBm (-23 dBm for the DCS-1800 standard). In other words, the receiver must digitize accurately in the presence of other signals (Figure 7-1) over a dynamic range of 91 dB, not counting overhead for equalization.

Figure 7-1 Signal variety at the BS receiver

GSM is designed to operate at carrier-to-interference ratios of 10-12 dB. Once the modulation is recovered, advance techniques equalize and decode the channel, separate signaling from traffic information, recreate the voice sounds etc. With this on mind, the SFDR of the converter and analog front end must exceed 100 dB relative to full scale (dBFS).

When a base-station receiver receives signals from nearby cellular phones, SFDR is very important. It indicates how strong signals interfere with adjacent channels. Figure 7-1 shows that front-end nonlinearity causes strong signals to produce the largest spurious signals (spurs). When a receiver must simultaneously process strong signals and weak ones that originate near the cell fringes, the strong signals cause spurs that can mask the weaker signals. SFDR is also important because it shows how the receiver performs as the signal approaches the noise floor.

The European GSM cellular-communications standard is among the most demanding applications for digital receivers. The dynamic-signal range is very wide and the receiver must introduce almost no in-band spurious signals.

7.1.3 Receiver noise performance expectations

This section estimates the expected level of performance based on the GSM-specific receiver requirements. We use the common digital receiver architecture depicted in Figure 7-2. It comprises a series of stages between the antenna and the Digital Receiver's ADC, which feeds the data to the DSP.

Figure 7-2 Common digital receiver architecture

First, assume that the receiver is noise-limited, with no in-band spurs that would otherwise limit performance. This assumption is reasonable, because you can choose the RF/IF components and provides a starting point for determining performance limits. The second assumption is that the IF bandwidth is 25 MHz. Thus, a 25-MHz band is translated downward in frequency and applied to the A/D converter input. To start the analysis, we must calculate the noise at the antenna. This noise is not the one in a 50Ω resistor, but is the atmospheric thermal noise in a 25-MHz band. Because a properly matched antenna appears resistive to the receiver front end, we can use the following well-known equation to determine the noise voltage across the matched input terminals:

$$V_n^2 = 4kTRB \quad (25)$$

where k is Boltzmann's constant ($1.38 \times 10^{-23} \text{ J/K}$), T is the temperature (in degrees Kelvin), R is the resistance (in ohms), and B is the bandwidth (in hertz). The power available from the source (in this case, the antenna is the source) is:

$$P_a = \frac{V_n^2}{4R} \quad (26)$$

When we substitute the previous equation, the above equation simplifies to:

$$P_a = kTB \quad (27)$$

Thus, in this case, the available noise power from the source is independent of impedance for nonzero and finite-resistance values.

From equation (27), the available noise power (when evaluated at 300K and 25MHz BW) is $103.5 \times 10^{-15} W$, or -99.85 dBm. This noise becomes the background noise at the receiver input. Remember that this noise is distributed across the entire 25-MHz band.

The next portion of the receiver is the RF section, which consists of a band-pass filter, low noise amplifier and a mixer. Follows the IF section with a postamplifier and a low-pass filter. The RF/IF architecture in general can be different from the one depicted in Figure 7-2. For example it may consist of several mixing stages (each with its own amplifying and filtering subsystem), prior to the digital receiver hardware. Nevertheless, this has little impact on the analysis, since all details can be hidden in the combined noise figure (NF) and gain of the front end section. If you realize that most of the noise degradation occurs in the receiver's first few stages, you can use traditional methods to calculate the RF/IF processing stages' overall NF and apply different design techniques to minimize it.

From a study of several manufacturers' wideband-component data books, we can assume that the combined noise figure is 8 dB and the total loss through the chain is 10 dB. Good low-noise amplifiers and microstrip filters can achieve such performance. Next, we must calculate the needed amplifying gain prior to the receiver input. Because the radio must simultaneously process multiple signals, automatic gain control (AGC) should not be used. Using an AGC would reduce RF sensitivity and cause the radio to drop the weaker signal. If the digital receiver hardware used is ICS-650 or ICS-152, it has a 50Ω standard input impedance. This translates to $10mW$ or equivalently $10dBm$ full scale input (as discussed in Chapter 3, Section 3.3.) in terms of power delivered to the receiver. Working with this information, we calculate that the RF/IF gain needed is 33 dB by:

$$\text{Full Scale receiver range} = \text{maximum signal power level} - \text{losses} + \text{RF/IF gain} \quad (28)$$

or, with the above numbers:

$$10dBm = -13dBm - 10dB + \text{gain} \quad (29)$$

Therefore, we need gain of 33 dB, which may be distributed between the amplifiers in the chain. When we apply the noise figure and conversion gain to the noise, the total noise power at the receiver input becomes:

$$-99.85dBm + 8dB + 33dB = -58.85dBm \quad (30)$$

The next step is to calculate how the converter hardware degrades the receiver noise. Although the calculation seems difficult, the simplest method is to convert both the S/N ratio and the receiver noise into rms voltages and obtain the total rms noise by determining the vector sum. First, assume that the receiver hardware is ideal and does not degrade the converter's performance. Working with the AD6640 (the ADC used on ICS-650) specifications, we have a typical SNR of 68 dB. With a $2V_{pp}$ input range, the ADC's noise voltage can be derived from:

$$V_{NOISE_{ADC}} = 0.707 \times 10^{-SNR/20} \quad (31)$$

With the above numbers, the voltage is calculated to be $0.2815mV$. The corresponding voltage for AD6644 (the ADC used on ICS-152) is $0.1257mV$, for a typical specified SNR of 75 dB. These voltages represent all noises, thermal and quantization, within the ADC. Similarly, converting -58.85 dBm to watts and then rms voltage yields:

$$P_{NOISE_{RF}} = 0.001 \times 10^{-58.85/10} = 1.303 \times 10^{-9} W \quad (32)$$

$$V_{NOISE_{RF}} = \sqrt{R \times P_{NOISE}} = 0.255mV \quad (33)$$

The total rms noise voltage will then be:

- for AD6640 --> $V_{TOTAL} = \sqrt{V_{NOISE_{ADC}}^2 + V_{NOISE_{RF}}^2} = 0.38mV \quad (34)$

- for AD6644 --> $V_{TOTAL} = \sqrt{V_{NOISE_{ADC}}^2 + V_{NOISE_{RF}}^2} = 0.2843mV \quad (35)$

These values are the total noise in the ADC, including receiver and quantization noise.

To have more realistic numbers we can substitute in the above equations with the true, derived from the measurements results. With the best SNR performance values, as specified in Table 2, we have the following performance expectation results:

- for ICS-650 --> $V_{TOTAL} = 0.7517mV \quad (36)$

- for ICS-152 --> $V_{TOTAL} = 0.38mV \quad (37)$

7.1.4 Conversion gain and sensitivity

These noise voltages contribute to the overall receiver performance in the following way. Assume that only one RF signal is present in the receiver bandwidth. The S/N ratio is then calculated to be:

- for ICS-650 --> $20\log(SIG/NOISE) = 20\log(0.707/0.0007517) = 59.47dB$.
- for ICS-152 --> $20\log(SIG/NOISE) = 20\log(0.707/0.00038) = 65.39dB$

If the sampled RF signal is the usual, 200kHz bandwidth GSM carrier, then this is an over-sampling application, and the actual signal bandwidth is much less than the sample rate. In this case only 400 Ksps, or twice the signal bandwidth, is adequate. Anything beyond this rate results in oversampling. Oversampling is very important because it allows for improving the S/N ratio in the digital domain.

In any digitization, the faster the signal is sampled, the lower the noise floor, because noise spreads out over more frequencies. (The total integrated noise remains constant, however.) This noise floor follows the equation:

$$\text{Noise Floor} = 6.02 \times B + 1.76 + \log(F_s/2) \quad (38)$$

where B is number of converter resolution bits. This equation represents the level of the quantization noise within the converter and shows the relationship between noise and the sampling rate, FS. Each time the sampling rate doubles, the effective noise floor improves by 3 dB.

Although increasing the sampling rate produces some improvements, one does not achieve the full effect of processing gain until the signal is decimated and filtered. The A/D converter bandwidth, which is the bandwidth that the A/D converter effectively digitizes, is always one half of the sampling rate. All of the sampled noise is distributed in this bandwidth, even if the A/D converter operates in an undersampling mode. In our case (with 65Msps sampling rate for ICS-650 and 200kHz single GSM carrier bandwidth) only 0.2/32.5 of the noise passes through the filter. Because sampling folds all signals and noise around a frequency equal to half of the sampling rate, the noise bandwidth is 32.5 MHz. Expressed in dB, the process gain for each carrier is:

- for ICS-650 --> $-10\log(200\text{kHz}/32.5\text{MHz}) = 22.11\text{dB}$
- for ICS-152 --> $-10\log(200\text{kHz}/35\text{MHz}) = 22.43\text{dB}$

When we account for the processing gain, the effective S/N ratio for a given signal is:

$$\text{SNR} = 6.02 \times B + 1.76 + \log(F_s/(2 \times BW)) \quad (39)$$

Remember that processing gain increases SNR because the gain spreads the noise bandwidth, after which we apply a narrowband filter. We can use the calculated from the previous page SNR instead of the number of bits:

- for ICS-650 --> $\text{SNR} = 59.47 + 10\log(F_s/(2 \times BW)) = 81.58\text{dB}$
- for ICS-152 --> $\text{SNR} = 65.39 + 10\log(F_s/(2 \times BW)) = 87.82\text{dB}$

These final noise calculations involved processing gain. As discussed earlier, processing gain results from eliminating all noise (and signals) outside the band of interest. In essence, decimation and FIR filtering digitally apply narrowband filtering to the wideband data.

In multicarrier applications, the ADC must share its dynamic range with other RF carriers. For example, if eight carriers have equal power, each signal should be no more than one-eighth of the total range, or -18 dB ($20\log(1/8)$ in terms of power), if we consider peak-to-peak signals. However, because the signals are normally not in phase with one another among receivers (because handsets are not phase-locked), the signals rarely if ever align. Therefore, a peak that is less than -18 dB of the total range satisfies our requirements. In addition, only two signals normally align at once, and, because these are modulated signals, we need to reserve only 3 dB (5 to

6 dB for a conservative design) for head room. In the event that signals do align and cause the converter to clip, the clipping occurs for only a fraction of a second before the circuitry clears the overdrive condition. Thus, headroom and multicarrier operation use a total of 21 dB of S/N ratio. The total S/N ratio in the single channel's multisignal environment is then:

- for ICS-650 --> $SNR = 81.58dB - 21dB = 60.58dB$
- for ICS-152 --> $SNR = 87.82dB - 21dB = 66.82dB$

These values are the equivalent carrier-to-noise (C/N) ratio in a multichannel environment.

The minimum C/N ratio for adequate demodulation depends on the modulation scheme. For GSM it was mentioned earlier to be between 10 and 12 dB. If the minimum C/N ratio is 10 dB, the input signal cannot be so small that the remaining S/N ratio is less than 10 dB. Thus, the signal level can fall 50.58 dB from its present level of 21 dB below the A/D converter's full scale, or 71.58 dB below the A/D converter's full scale. Because the A/D converter has a full-scale range of 10 dBm, the signal level at the A/D converter's input is then -61.58 dBm. The needed RF/IF path amplifying gain was estimated to be 33 dB, relative to the antenna. With this value, the noise-limited sensitivity is calculated to be:

- for ICS-650 --> -94.58 dBm, referred to the antenna
- for ICS-152 --> -100.82 dBm, referred to the antenna

These final sensitivity estimates show that the ICS-152 performance is very close to the GSM standard specification limit of -104 dBm lowest signal amplitude recovery at the BS, while the ICS-650 comes 10 dB short of meeting this criteria. The difference in sensitivity between the two receivers is due to the lower A/D converter S/N ratio. Other possible sources of reduced sensitivity include higher analog-chain noise and effects of aperture jitter. If the design requires more sensitivity, one can increase the gain in the RF/IF stages. However, noise figure depends on gain, and an increase in gain may adversely affect noise performance from additional gain stages. Another solution is higher input impedance values design in order to lower full scale power delivered to the receiver, thus avoiding high amplifying gains.

Increasing sensitivity can also be accomplished by dividing the processed channel bandwidth to a subset of smaller bandwidths and processing each one in parallel fashion, thus increasing the S/N ratio. For example, if the 200 kHz GSM channel bandwidth is divided into 5-40 kHz smaller bandwidths, the expected processing gain will be $-10\log(40kHz/32.5MHz) = 29.1dB$. This is an improvement of 7dB on the average, but with the cost of increased computation.

An important conclusion from this section is that noise performance in general is dominated by the receiver's ADC quality. Both increased resolution and sampling bandwidth improve the device characteristics. This is evident from the improved sensitivity estimate for ICS-152 over the one for ICS-650 and is mainly due to the increased SNR of the 14-Bit converter. Nevertheless, noise limitation are flexible and can be controlled with various techniques and alternative designs.

7.1.5 Spurious performance expectation

A noise-limited example does not adequately demonstrate the true limitations of a receiver. Other limitations, such as spurious-free dynamic range (SFDR), are more restrictive than SNR and noise. Referring back to the experimentally derived performance in Table 2, the ICS-650 has SFDR of -86 dBFS, or -76 dBm (because full scale is 10 dBm). Also, we can assume that a tolerable carrier-to-interferer ratio (CIR) is 14 dB. Based on this assumption, the minimum signal level is -72 dB full scale (-86+14) or -62 dBm. With the 33 dB RF/IF gain from the example in the previous section, at the antenna, the signal level is -95 dBm (-62-33). This figure is comparable to the noise limited sensitivity estimate for ICS-650 from the previous page. For a single carrier radio applications this may seem like a reasonable SFDR value, but when we consider the multi-carrier characteristics of the GSM, like in the noise-limited example from the previous section, the signal level at the antenna becomes -74 dBm (-95+21) - way short of the 91 dBm (-104+13) dynamic range, specified for the GSM signals at the BS receiver. The corresponding figures for ICS-152 are 7 dB better, but still fall short of meeting the 91 dB requirement. Therefore, single or multitone SFDR would limit receiver performance long before noise limits it.

Unlike noise, spurs are much harder to deal with. They are the result from converter static nonlinearity (both DNL and INL) and are usually component-specific. Their amplitude is proportional to the amplitude of the digitized signal, proof of which are the results and discussions from Chapter 5. One way to reduce the effects of the spurs is then to control the signal strength at the entry of the ADC. This technique, though, has one major drawback. It does not in any way change the spurious properties of the device. Instead, it hides the spurs into the noise floor, thus reducing the potentially available SFDR.

The other best known approach to control spur levels is the injection of narrowband dither into the wideband signal prior to the actual conversion in the ADC. Adding dither can greatly improve SFDR. The addition of out-of-band noise can improve SFDR well into the noise floor [9]. Although the amount of dither is converter-specific, the technique applies to all ADCs in which static differential nonlinearity limits performance, rather than ADC problems, such as slew-rate limiting. In simple terms, dither works by randomizing the coherent spurious signals that are generated within the ADC. Since no energy is lost during this process, the dither simply causes the spurs to appear as additional noise in the floor of the converter. This additional noise appears as a slight increase in the average noise floor of the converter. The trade-off one makes by using out-of-band dither is that all internally generated spurious signals can be removed, but there is a slight degradation in the overall SNR of the converter. In practical terms, this degradation amounts to much less than 1 dB of sensitivity loss, which results in a typical noise-limited example and is much better than the SFDR-limited example.

We need to point out two important things about dither. First, in a multicarrier receiver, one can expect that none of the channels are correlated. If this expectation is true, the multiple signals serve as a self-dither source for the receiver channel. If this expectation is false, there is need to add dither when signal strengths are weak. Second, the noise contributed from the analog front end alone is insufficient to dither the ADC. Practical dither values for the converters used in the

ICS receivers are between -30 and -40 dBm. In comparison, the analog front end provides only -58.85 dBm of noise power, far from what is need to achieve optimum performance.

GSM is a dynamic application, where slow frequency hopping and channel relocation is done "on the fly". Using dither in such environment necessitates for a smart dither bandwidth and spectrum allocation scheme which will prevent signal and dither to coincide.

The RF section also contributes to the receiver's spurious performance in several ways. Unless one takes steps to limit them, spurs that originate in the RF section disrupt the receiver's performance. Dither does not affect these spurs and they must be addressed separately. Care must be taken when choosing RF/IF components such as mixers, filters and amplifiers in order to keep these spurious signals at bay. They result mainly from third order products of the form:

$3OP = nf_i + kf_j$, where n, k, i, j are some integers, for which the frequency products fall within the spectrum of interest.

7.1.6 DCS-1800 and PCS-1900 performance expectation

Having considered the core GSM standard so far, few words need to be said about its upband derivatives DCS-1800 and PCS-1900. These two standards are very similar to GSM, except for the operational spectrum allocation. Their deployment resulted from the concept of moving up to the higher frequency bands, in order to obtain more frequency spectrum. In Europe, for example, only 2x25MHz is available for GSM-900 services, whereas a band of 2x75MHz is reserved for DCS-1800. Measurement results have shown that the path loss in urban areas is approximately 10 dB higher at 1800 MHz compared to 900 MHz. This is the main reason behind the reduced BS receiver dynamic range requirements for the upband standards.

The DCS-1800 standard specifies the level at which the 3 dB degradation in performance occurs as -23 dBm (10 dB lower than GSM-900), while the sensitivity level of -104 dBm is preserved due to the same considerations as with the GSM standard (required error rate performance assuming a specific propagation channel, typical atmospheric and interference noise conditions, signal quality etc.). With these signal bounds, the dynamic range for DCS-1800 and PCS-1900 is:

$$(-23dBm) - (-104dBm) = 81dB \quad (40)$$

Since DCS-1800 and PCS-1900 share the same GSM-900 architecture, all aspects mentioned so far in the GSM performance evaluation will in general hold true for the upband standards. With 10 dB reduced dynamic range and following the same calculations, as in the previous sections, the resulting SNR numbers for ICS-650 and ICS-652 are:

- for ICS-650 --> $20\log(SIG/NOISE) = 20\log(0.707/0.00109242) = 56.22dB$.
- for ICS-152 --> $20\log(SIG/NOISE) = 20\log(0.707/0.000879) = 58.11dB$

With oversampling gain and 21 dB headroom, the reference values at the antenna are:

- for ICS-650 --> -91.33 dBm, referred to the antenna
- for ICS-152 --> -93.54 dBm, referred to the antenna

There are few conclusions to be made from this simple evaluation. First we observe lower margin in performance difference between the two boards (6.24 for GSM-900 evaluation compared to 2.21 for DCS-1800/PCS-1900 evaluation). The second point that grabs our attention is the evident drop in noise-limited sensitivity performance at the antenna. Both discrepancies can be attributed to the high amplifying gain needed in the RF/IF stages of the receiver. To bring the highest possible signal amplitude to the receiver full scale in the case of GSM-900 we needed amplifying gain of 33 dB. For the DCS-1800/PCS-1900, this gain figure is 43 dB due to the 10 dB lower highest possible signal amplitude. These high gains, in term, caused the background noise to be boosted up in power along with the signal of interest long before the actual digitization takes place. With such high gains, the background noise becomes dominant and affects the overall SNR of the receiver in a nonlinear fashion according to the rms noise terms vector combining formula (equation (34) or (35)). This is especially true for higher resolution ADCs, where the benefit of having lower converter noise floor is lost due to the much higher amplitude values of the amplified antenna background noise.

The unusually high gains for both examples were due to the intrinsic input impedance design of the ICS receivers. At the time of writing, primarily market for these types of digital receiver hardware is the research community. Therefore the ICS design considerations were to provide a standard, 50Ω input interface, compatible with the majority of signal generator hardware and connecting cables (such as BNC-type connectors). The actual ADCs components have much higher input impedance. For example AD6640 has typical Differential Input Resistance value of $1k\Omega$, which makes the ADC full scale input -3 dBm. If we base our figures on receiver design which utilizes this property, with the so far considered examples we will see a reduction in the needed RF/IF gain of 13 dB. Then reworking the calculations will give us the results for $1k\Omega$ input impedance in Table 3.

	Noise limited sensitivity for 50Ω input impedance		Noise limited sensitivity for $1k\Omega$ input impedance	
	<u>GSM-900</u>	<u>DCS-1800/PCS-1900</u>	<u>GSM-900</u>	<u>DCS-1800/PCS-1900</u>
ICS-650	-94.58 dBm	-91.33 dBm	-95.08 dBm	-104.834 dBm
ICS-152	-100.82 dBm	-93.54 dBm	-103.255 dBm	-111.932 dBm

Table 3: ICS-650 and ICS-152 noise limited sensitivity

These numbers look much more promising and are close to or exceeding the GSM sensitivity requirements (except for ICS-650, GSM-900 value). The increase in performance difference between the two boards, compared with the higher gain examples (50Ω input impedance), is also evident - 8.175 dB for GSM-900 evaluation and 7.098 dB for DCS-1800/PCS-1900. Note how close these values are to the difference in SNR for the two boards from Table 2 (8 dB). The lower

figure for DCS-1800/PCS-1900 is in accordance with the 10 dB higher gain in the RF/IF stage (meaning 10 dB higher noise values, accounting for the drop in performance difference) compared to the GSM-900 example. The improvement for GSM-900, comparing with the 50 Ω input impedance example, is only a few dB, considering the quite substantial reduction in preconversion gain. This is due to the fact that the RF noise was still not dominant. The actual gain benefit here is entirely from the reduction in RF noise voltages in the combining equations. On the other hand, the same improvement for DCS-1800/PCS-1900 is 13.5 dB for ICS-650 and 18.4 dB for ICS-152. This can be explained with the smaller dynamic range of the upband standards. In the 50 Ω example, this fact was masked by the quite high RF/IF gain needed to bring the signal at full scale. The RF noise became dominant and diminish the benefit of the smaller dynamic range.

Another aspect in the DCS-1800/PCS-1900 support evaluation is the receiver spurious performance. As with the GSM-900 example, we may assume that the RF/IF stage does not create significant spurious activity due to third order intermodulation products. In this case tweaking the RF/IF configuration or the input impedance will not have any effect of the receiver's spurious performance, since all spurs are created within the ADC. Due to the smaller dynamic range of the upband standards (81 dB), the spurious characteristics will be less limiting. From the GSM-900 example we had spurious free dynamic range at the antenna of 74 dBm for ICS-650 and 81 dBm for ICS-152 correspondingly. We now see that ICS-650 is only 7 dB short of meeting the dynamic range requirement, while ICS-152 exactly satisfies it.

To conclude, the DCS-1800/PCS-1900 evaluation shows that digital receiver support for these standards in terms of noise and spurious limitations is possible with the tested ICS-152 converter technology without any additional spur controlling and noise reduction techniques. We must note though, that this will only be achieved with careful RF/IF section design in terms of component noise figures, tolerable losses, input impedance etc. ICS-650 comes close in performance expectation and is potentially able to support the above cellular standards with some form of spurs controlling technique.

7.1.7 Summary

There are many other factors that would otherwise limit the receiver performance (such as ADC clock jitter and phase noise for example), but they can be avoided with proper hardware design. In the scope of the experimentally derived results, noise and spurious limitations are the most common digital receiver limitations. When applied to the GSM architecture, the spurious limitation seems to be the one of great importance, unless care is taken to control the spur levels. This is also the main motivation for the active research and transition towards higher resolution technologies. The noise in some cases can be also a limiting factor, especially when working with large bandwidths and higher RF/IF gains. We must note, though, that straightforward application of the software radio concept will not accomplish a successful solution in general.

As we have seen from the measurements, there is a visible improvement in performance of the 14-Bit receiver over the 12-Bit one. This was also in effect with the GSM evaluation, where the noise limiting requirement was satisfied with the 14-Bit architecture, while the spurious was close to or being met even without additional spur controlling techniques. Nevertheless, GSM support with current converter hardware is practically possible, but remains a tailored solution with many unknown parameters.

7.2 CDMA support evaluation

The worldwide acceptance of the CDMA as a mobile communications standard is mainly due to its theoretically proven qualities of coverage increase and flexibility over the existing alternative standards. While the CDMA principles were known for a long time and used intensively in the military for slightly different purposes, it was not until Qualcomm Inc. pioneered its well known IS-95 standard that the CDMA concept achieved its mature state and commercial success. Due to its high popularity and wide acceptance throughout North America, we devote this section to evaluate its support in software radio.

7.2.1 Reverse Link Signal characteristics

CDMA is a single carrier standard. To cover the proposed capacity and accessing scheme requirements it needs to have a wideband implementation. As with the majority of other wireless standards, the most critical part of the system functionality is the reverse signal link (from the mobiles to the BS receiver). This is especially true for the CDMA, where mutual user interference and non-synchronization is part of the system design (Figure 7-3).

Figure 7-3 CDMA Uplink Signal

All users transmit on the same frequency with the same signal bandwidth. This means that signals at the BS add up in power and are received as a single frequency carrier. Due to the different distance of each user from the BS (as well as different propagation conditions) signals arrive at the BS with different power levels. This is well known as the near-far effect and is the main motivation behind the user terminals power control scheme, as envisioned in the IS-95 standard.

In the reverse direction, the handset power is controlled by a three-level algorithm to ensure that all handset signals are received at the base station with the same power level. The purpose of the reverse link open loop power control is to compensate for distance attenuation and shadowing. It adjusts the initial access channel transmission power of the mobile station and compensates large abrupt variations in the pathloss attenuation. The mobile station determines an esti-

mate of the pathloss between the BS and MS by measuring the received signal strength at the mobile using an AGC circuitry, which gives a rough estimate of the propagation loss for each user. The basic formula by which, this is implemented is:

$$P_{TRANSMIT} = k/P_{RECEIVED} \quad (41)$$

In more simple form this equation can be rewritten as:

$$\text{Mobile Transmit Power (dBm)} = -73 \text{ dB} - \text{received power from BS (dBm)} \quad (42)$$

Parameters in this formula are transmitted on the synch channel and received from the mobile in the access message. Figure 7-4 illustrates the recommended power control range characteristics.

Figure 7-4 IS-95 Open Loop power control range

Once the average power is adjusted to the required level, the closed loop power control takes place. Its purpose is to compensate for rapid fluctuations of power due to motion and to equalize the power received from all mobile units. IS-95 specifies the control range to be ± 24 dB in steps of 1 dB. The composite dynamic range for open and close loop power control is specified to be ± 32 dB for mobile stations operating in band class 0 (800 MHz), and ± 40 dB for mobile stations operating in band class 1 (1900 MHz) [22]. The typical standard deviation of the power control error due to the closed loop is on the order of 1.1 to 1.5 dB.

7.2.2 CDMA receiver performance expectation

To derive an estimation of the CDMA receiver performance we can use the same arguments as with the GSM evaluation. Due to the three different levels of power control in the IS-95 design the dynamic range of the received at the BS composite signal is much smaller than the dynamic range of the GSM signal. According to the outlines in the end of the previous subsection, it is save to consider the dynamic range of the IS-95 signal to be 67 dB for mobile stations operating in band class 0, and 83 dB for mobile stations operating in band class 1 (including the typical standard deviation of the power control error due to the closed loop). These values fit well in the range of SFDR receiver performance from Table 2 in Chapter 6.

Although both receivers can accommodate the IS-95 signal dynamic range, there are few other consideration points that need to be mentioned:

- 1). The CDMA composite signal has a noise-like shape and is unlikely to create large spur levels. This in term could improve the SFDR performance of the receivers in much the same way as dithering does.
- 2). Since user differentiation is based on code, there is no need of C/I margins (the effective E_b/N_o is obtained with processing gain). This also helps improve the receiver's SFDR.
- 3). IS-95 is a single carrier standard (it operates at the top of the AMPS/IS-54 band with allocated bandwidth of 1,23 MHz). Therefore AGC can be used to relax the SFDR requirements. AGC reduces the possibility of in-band overdrive and allows for maximum signal gain prior to data conversion, thus relaxing the dynamic range requirements on the A/D converter.

These arguments lead us to believe, that IS-95 software radio application will not be SFDR-limited, as was the case with GSM applications. Nevertheless, it requires a careful RF/IF design in order to achieve best performance. For example one might want to downconvert the IS-95 bandwidth of interest as close to baseband as possible prior to digitizing or analog-filter the unwanted signals. RF/IF gain also needs to be designed with care. Weather AGC or fixed gain is used in the stages prior to the ADC, the signal level shouldn't drop much below a predetermined value. For most of the time the system logic will demand that the composite signal be at minimal, but sufficient in terms of service quality, power level. This poses a threat to the signal being buried in the converter noise, which may be much higher than the usual atmospheric noise and interference tolerable levels.

All this makes a strong case that noise issues will be of much greater importance for CDMA type of applications in software radio. To convince ourselves, it may be a good idea to analyze a typical IS-95 application in terms of noise-limiting aspects. We start with receiver noise calculations. Assume that the receiver used is ICS-650, and that the bandwidth of the front end is the receiver's ADC Nyquist bandwidth. Although the allocated IS-95 bandwidth is only 1.23 MHz, for example, assuming Nyquist bandwidth simplifies the calculations. Using the assump-

tions, a sample rate of 65 Msps gives Nyquist bandwidth of 32.5 MHz. Using equation (27), the background noise at the receiver input is -98.71 dBm and distributed across the entire 32.5 MHz bandwidth. Due to the much smaller IS-95 signal bandwidth, we can expect processing gain (whether from analog/digital filtering or decimation) of 14.22 dB. This will bring down the level of in-band noise to -112.93 dBm.

Next, we try to determine the minimum reliable IS-95 signal level. For this purpose, the SNR requirement can be turned into a specification for the minimum required received RF signal power, denoted P'_S . In the case of a single CDMA user:

$$SNR_{req} = \left(\frac{E_b}{N_0} \right)_{req} = \frac{P'_S}{N_0 R_b} = \frac{P'_S}{N_0 W} \cdot \frac{W}{R_b} = \frac{P'_S}{N_0 W} \cdot PG \quad (43)$$

$$P'_S = (SNR_{req} \cdot N_0 W) \div (PG) \quad [44]$$

or in decibel form:

$$P'_S(dBm) = SNR_{req}(dB) + N_0(dBm/Hz) + W(dBHz) - PG(dB) \quad [45]$$

Using typical values of the system parameters we have:

$$P'_S(dBm) = 7.0 - 173.83 + 60.9 - 21.1 = -127.03 dBm \quad [46]$$

We now take on account factors such as noise figure of the receiver, one-cell margin for 90% reliability and handoff gain. Then, the reliable signal level (RSL) is:

Minimum required signal P'_S	-127.03 dBm
+ Receiver noise figure	5.00 dB
+ One-cell margin for 90% reliability	10.25 dB
- Handoff gain	4.09 dB
RSL	-115.87 dBm

What this simple calculation shows is that the reliable signal level is below or at best comparable to the background noise level (-112.93 dBm). Even with an average of 12 active users per cell, the composite received signal $(-115.87 + 10 \log 12 = -105.08 \text{ dBm})$ is not far above the background noise level.

The implications of this fact, related to the software radio concept, is that converter's own quantization noise could be a potential interferer if the received CDMA signal is not amplified enough prior to digitization. To avoid this we now have two choices: First one is to use high amplifying gain in the RF/IF stage to boost the received composite signal level well above the converter's quantization noise level so it will be a non-dominant factor in the noise combining equation (equations (34) or (35)). For example, applying gain of 40 dB to 12-user composite signal will result in -65.08 dBm RSL prior to digitization. With oversampling gain of

$$U = 10\log(32.5/1.23) = 14.22\text{dB} \quad (47)$$

the effective signal level will be -50.86 dBm. This is well above the ICS-650, 60 dB SNR from Table 2 and will be more than sufficient to cast aside the effects of converter's quantization noise. But high gains in the RF/IF stage are a last resource, since they increase the overall noise figure of the receiver. Apart from this, high gains in this case will result in a reduced signal dynamic range in general.

A more elegant choice is to use AGC and improved power control algorithms. Automatic gain control will maintain the received signal power close to an optimum for the current situation level (depending on the power control algorithms in the system, number of active users, minimum SNR required for the specified service quality etc.). It will also prevent the converter from overloading and losing the signal altogether.

With the presented discussion so far it is evident that CDMA software radio support is much more realizable with the commercially available converter technologies. This is primarily due to its relatively small signal dynamic range and operational bandwidth compared to other cellular standards. It should be mentioned, though, that realizing a CDMA standard like IS-95 in software radio does not result in great benefits since it is a single carrier standard with a common RF/IF stages. Therefore there are no real savings due to hardware reduction, which is the main benefit of a software radio application. Software radio implementation with converter technologies like ICS-650 or ICS-152 will be overdesigned in this case, since many Sigma-Delta ADC technologies are available for this amount of bandwidths and are better suited for CDMA applications than the Flash-type converters due to their quantization noise shaping properties.

7.3 Selected 3G standards support evaluation

The primary requirement for the Third-Generation Mobile Communications System is to establish a "Global System". The radio transmission technology of the third-generation mobile communication system should be able to support multimedia, personal as well as intelligent functions. Specifically, the system should meet the requirements recommended by ITU-R such as advanced multi-rate services up to 2Mbps and a quality comparable to that of the fixed communication networks. At the same time, the system should aim to realize a simple cell structure, easy channel management, high subscriber capacity and low transmit power.

7.3.1 Evolution towards third generation

From a radio access perspective, adding 3G capabilities to the existing 2G systems mainly means supporting even higher bit rates. Possible scenarios depend on spectrum availability for the network operator. Depending on the spectrum situation, two different migration scenarios must be supported, namely:

- Refarming of existing spectrum bands.
- New or modified spectrum bands.

To support the above scenarios, two 3G radio access building blocks have been identified:

- 1). EDGE uses high-level modulation (such as 8-PSK) in 200 kHz TDMA and is based on plug-in transceiver equipment, thereby enabling the migration of existing bands in small spectrum chunks.
- 2). Universal Mobile Telecommunications Service (UMTS) is a new radio access network based on 5 MHz Wideband-CDMA (W-CDMA) and optimized for efficient support of 3G services. UMTS can be used in both new and existing spectra.

With the EDGE modulation scheme 8-PSK, the E_b/N_0 distributions will be lowered, due to higher gross bit rates. In the GSM case, assuming a full-rate speech coder, speech coverage requires an E_b/N_0 of 6 dB. For TDMA the requirement is 15.7 dB. In case of EDGE, the difference ΔE_b will be [23]:

$$\Delta E_b = \frac{R_{GSM}}{R_{EDGE}} = \frac{271}{3 \times 271} = -4.8dB, \text{ with respect to GSM.}$$

$$\Delta E_b = \frac{R_{TDMA}}{R_{EDGE}} = \frac{48.6}{3 \times 271} = -12.2dB, \text{ with respect to TDMA.}$$

The IS-136 TDMA North America Standard community envisions a three-step strategy for evolving into a 3G standard. It consist of enhancing the voice and data capabilities of the 30 kHz channels, while adding the evolved GSM 200 kHz carrier for 384 kb/s data (EDGE), and adopting a wider 1.6 MHz bandwidth TDMA carrier for 2Mb/s data. New 8-PSK modulation can improve the capabilities of the 30 kHz carrier together with new timeslot formats [24].

While the EDGE concept has the benefits of easy and seamless transition towards the 3G concept, it inherits the main drawbacks of the TDMA systems - network complexity, higher power levels, resource-limited capacity etc. Due to this reason, ITU considers the UMTS concept as being more viable in the long run. The main objectives and technical challenges in UMTS are briefly described next:

- UMTS spectrum - It is an objective of UMTS to exploit the frequency bands identified for FPLMTS: 1,885 MHz - 2,025 MHz and 2,110 MHz - 2,200 MHz, including the bands: 1,980 MHz - 2,010 MHz and 2,170 MHz - 2,200 MHz identified for the satellite component. UMTS standards may be applied in frequency ranges currently not identified for UMTS and should provide higher spectrum efficiency than the existing systems.
- UMTS bit rate services - It is an objective of UMTS to provide high bit rate services up to initially 2 Mbit/s including support of user bit rates up to 144 kbit/s with complete wide-area mobility and coverage, and up to 2 Mbit/s in situations of local mobility and coverage. Some medium bit rates in the order of 500 kbit/s should also be provided. It is not expected that high bit rate services will be available in all environments.
- UMTS coverage - In the case of UMTS, the overall goal is the capability to provide the services in a single integrated system, allowing users access regardless of the particular environment in which they happen to be. Provision of services via handheld, portable, vehicular-mounted, movable and fixed terminals, in all radio environments, based on single-radio technology is expected.
- Compatibility, wherever possible, with second-generation mobile networks e.g. GSM/DCS, while providing speech and service quality at least comparable to the current fixed public network.

The basic UMTS wideband carrier is configured in the three dimensions - time, frequency and spreading (Figure 7-5). All three axes need not be exploited operationally at all times. For instance, FRAMES Multiple Access (FMA) incorporates both spread and non-spread carrier characteristics.

Figure 7-5 UMTS Wideband carrier structure.

In the FMA1 mode, for example, the carrier spacing B is equal to 1.6 MHz, while in the FMA2 mode, B could be equal to 4.8 MHz, i.e., three times the carrier spacing as in the former mode. The carrier spacing is always a multiple of the GSM 200 kHz carrier raster. Both modes will allow for a dual carrier bandwidth selection (3.2 and 9.6 MHz respectively) to make the bandwidth selection flexible depending on the service and/or environment.

7.3.2 The Bandwidth-Dynamic range trade-off

Selecting the digitization bandwidth, so that channels of only one standard are received simultaneously offers an advantage, that can be derived from the fundamental characteristics of mobile communications signals, and could be named "The Bandwidth-Dynamic range trade-off". It is not a characteristic of the signals themselves, but results from the typical adjacent channel interference and blocking characteristics defined by the different standards. Given a fixed digitization bandwidth, the dynamic range of a mobile communications signal diminishes as the channel bandwidth increases. The narrower the channels, the more channels are present in the fixed digitization bandwidth. But more channels mean more adjacent channel interferers and eventually an increasing dynamic range. This behavior is shown on Figure 7-6, where the typical frequency masks of different mobile communications standards are given.

Figure 7-6 An illustration of the bandwidth-dynamic range trade-off.

The channel of interest is centered around f_{carrier} . While the adjacent channel interferer power depends on the very standard, the in-band dynamic range δ of the channel of interest is supposed to be of the same order in all three cases. For digitization bandwidth of 5 MHz, for example, the signal comprises 25 channels in the case of GSM. Only three 1.23 MHz IS-95 channels fit the same 5 MHz bandwidth in practice and in the case of UMTS it is just one channel. Hence the GSM channel of interest has 24 interfering adjacent channels, while the IS-95 channel of interest has only two adjacent interferers. The UMTS channel is not interfered with by adjacent channels. Consequently, the latter signal has minimum dynamic range, while the GSM signal has the highest dynamic range of the three signals.

7.3.3 The Software radio and 3G concepts.

The ongoing research directed towards 3G systems evolution has produced number of radio transmission technologies and accessing schemes. Unfortunately, none of them has yet been standardized. The general trend, though, is to move to higher frequencies and wider bandwidths in order to accommodate the proposed large bit rates. Until then, we can only try to estimate how the software radio concept will fit in this new wireless environment.

The bandwidth-dynamic range trade-off suggests that a very wide signal dynamic range is only necessary in relatively narrow bandwidths, where the channels of interest lie. This property greatly relaxes the requirements on the receiver's dynamic range coverage with relation to the 3G systems. In accordance with the general trend, they are expected to employ large bandwidths and to implement some form of CDMA accessing, combined with some form of advanced power control mechanism. In Section 7.2, some aspects of the software radio concept with respect to CDMA were discussed and the general conclusion was that CDMA is much more practically realizable in software radio, due to its smaller signal dynamic range properties. The same reasoning can be applied for the UMTS concept, which will be based on CDMA/TDMA technology. The 3G channel bandwidths will unlikely exceed 15 MHz, which with today's high-resolution converter technologies operating in the 100 MHz zone is more than adequate for satisfying the Nyquist criteria.

On the other hand, due to the CDMA nature combined with advanced power control mechanism, we can expect 3G systems to have very small signal dynamic range. This in general means less spurious emission because of the noise-like characteristics of the received CDMA signal and less constraints in terms of recoverable signal dynamic range at the converter. Therefore it is unlikely that a 3G system will be SFDR-limited even with the present state of the converter technologies capabilities.

Referring back to the noise-limited examples with respect to a 3G system it is clear that noise-related issues will be much more critical in this case. The increased signal and digitization bandwidths will result in higher background noise levels. Also, due to the power control mechanism needed to reduce the mutual interference in the system, the composite signal at the receiver front end may require large RF/IF amplifying gains prior to digitization in order to avoid resolution-limited convertor quantization noise to become a destructive interferer. These limitations result in very strict power amplifiers requirements. A Wideband Power Amplifier (WPA) amplifies the composite signal, which has a greatly varying envelop and thus is extremely sensitive to non-linearities. The WPA has to use a feed-forward compensation technique for suppression of undesired IM in order to get an acceptable efficiency of the power supply. The WPA also imposes additional demands for multilevel modulation, multicarrier and CDMA systems as well as low voltage requirement - power efficiency is very important.

Chapter 8

Wideband receiver dimensioning in a multistandard environment

This chapter is the logical continuation of the analysis made in Chapter 7. It identifies the limited dynamic range performance of today's ADCs as being the most challenging element in the development of Software Radios. This is especially true in multistandard/multicarrier wireless environments such as some of the proposed 3G systems. In order to circumvent this problem and to relax the ADC requirements, several SWR front-end architectures are proposed and examined in terms of their achievable performance.

8.1 The ADC challenge

The need for software-defined radios raises a number of technical challenges, which play a significant role in the development of third-generation personal communication systems (PCS). Some of the most important technical challenges deal with analog-to-digital conversion and digital signal processing technologies, which in many cases cannot provide the advanced hardware needed to support the demanding PCS telecommunication services.

It is straightforward to realize that the hardware platform is a most prominent enabling component of a software radio terminal. Of special interest is the very part of the terminal, where analog signal processing is replaced by digital signal processing, namely the *digital front-end*. It has been established to be one of the most critical functionalities of the software radio terminal - largely due to the combination of wide bandwidth and high dynamic range of the signals to be processed.

Typical characteristics of mobile communication signals - fading, shadowing, and so on (caused by RF signal propagation characteristics) - in conjunction with potentially strong blocking and interfering signals (due to coexistence of several transmit signals) lead to very high dynamic range. Such "narrowband signals at wideband reception" yield a dynamic range far above that with which conventional terminals have to cope. For instance, a wideband signal in the GSM band has a dynamic range in the vicinity of ~100 dB.

Our measurements and evaluation showed that with current state of the art, 14-bit ADC technology it is possible to derive SFDR performance in excess of 90 dB. As it was mentioned before, this is just enough to theoretically cover the GSM standard BS receiver dynamic range requirements of 91 dB (the difference between the maximum nominal error rate defined blocking signal level -13 dBm and the reference sensitivity defined useful input signal level -104 dBm).

Contemporary GSM base stations, on other hand, regularly exhibit sensitivity which surpasses the specified performance. Although the exact figures are not public information, it appears safe to assume -108 dBm reference sensitivity presently and -110 dBm in the nearest future. Hence, the required dynamic range becomes indeed a figure in the vicinity of ~100 dB.

Another very important issue to mention is the expected drop in achievable SFDR performance when such converter technologies are employed in field applications. The dynamic specifications in ADC data sheets are optimum figures that manufacturers obtain by testing with optimum test setups. They demonstrate what a device is capable of doing, but it's difficult to realize such performance in practice. All this can be said about our measurements and evaluation, where the main goal was to obtain best achievable device performance in the lab. Under realistic conditions, though, there could be substantial difference. Few reasons for that may be the following:

- SFDR measurements were performed in a coherent sampling fashion. In practice this is almost impossible to realize as there is no way to perfectly synchronize the received signal to be “coherent” with the sampling frequency at the receiver.
- Derived ADC performance was through measurements on a single-tone, sinusoidal analog signal. In a realistic multicarrier wireless environment and with signals very much deviating from pure sinusoids (actually being entire spectral bands), one can certainly margin for slight drop in performance due to nonlinearity, as well as harmonic distortion, that can potentially be hidden in the case of single-tone measurements.
- SFDR figures were derived with nominal analog input signals, placed at 1 dB below full scale. In order to optimize spurious performance in practice, though, the ADC input signal may be required to be placed at lower levels (-3 dBFs for example).
- What real-world system has a perfect, jitterless clock; a perfect, no-phase-noise, analog driving source; and perfect, noise-free power supplies?!

From all these reasonings it is obvious that at least 2 more bits of resolution are currently needed in order to safely cover the dynamic range requirements in a typical GSM working environment. Extending to signals of different standards that might appear simultaneously at the receiver, the dynamic range can get even larger. A simple reference to the CDMA support section in the previous chapter can easily convince the reader, that typical IS-95 signal reliable levels could be well below -110 dBm. The same can be said about the majority of other mobile communications standards today and extended towards the emerging 3G PCS systems, primarily based on wideband CDMA technologies.

With the above considerations, it becomes clear that the satisfaction of ADC performance requirements is a real challenge in the development of software radios. This challenge has also been identified by major projects dealing with next-generation PCS/UMTS systems throughout the world: such projects include the Flexible Integrated Radio System and Technology (FIRST) [25], Future Radio Wideband Multiple-Access Systems (FRAMES) and Speakeasy [26]. The ADC challenge proclaims itself especially in the 3G systems development, where the software radio concept is foreseen as the cost-effective and flexible solution that could provide more efficient approach toward the implementation of demanding multistandard third-generation radios. For this reason, SWR are considered one of the essential components of third-generation systems, and therefore have been experiencing ramped-up research and development over the last few years.

In this context, the following section is dedicated to discuss this technical challenge and any potential enabling schemes that could relax the immense ADC technology requirements, thus aiding the implementation of software radio terminals.

8.2 Digital front-end architectures aimed at reducing the ADC requirements

The ADC is a key component of a software radio terminal. As the analog front-end, it has to cope with signals of large bandwidth and high dynamic range. Unfortunately, the ADC technology today is not advanced enough to enable unique, straightforward approach to the software radio front-end architecture. In the meantime, there is a great demand for such applications, which will enable multistandard PCS support on a unified hardware platform. This is especially true in scope of the development of next generation PCS services, where backward standard compatibility and support is a must. Which Service Provider wouldn't favor a network solution that would encompass broadband services with multistandard support together with the ability of implementing new, advanced services without the necessity of additional hardware development costs and updates?!

Therefore, alternative front-end designs, targeted specifically to overcome the ADC technology limitations are of special interest as being the viable solution to a practical software realization today. Below, several approaches, which would circumvent the dynamic range challenges are outlined and evaluated.

8.2.1 Non-linear analog signal processing prior to digitization

One possible method of tackling the ADC challenge would be to control or suppress the signal's dynamic range at the input of the ADC. In this context we could apply, for example, a logarithmic compression of the signal before digitization and subsequently could conduct digital anti-log processing to restore the original signal properties. Much in the same way the voice signals are being routinely digitized today for transmission through the public switched telephony networks, this principal is illustrated below:

Figure 8-1 Companded PCM with typical compression characteristic

Historically, the nonlinear function was first implemented on analog signals using nonlinear devices such as specially designed diodes. Nowadays, with the advance in IC technologies, a resistor array is used to establish a sequence of nonuniformly spaced decision thresholds. A sample value is encoded by successively comparing the input value to the sequence of decision thresholds until the appropriate quantization interval is located. This allows for integration of the nonlinear function and the subsequent linear encoding process into a specialized device (codec), without the need of explicitly using an ADC as part of the system.

Looking back at this approach from the software radio perspective, we see a number of difficulties. First of all, we are bound to use a nonlinear processing device on the analog signal to implement the compressing function prior to digitization. Due to the wideband nature of the signal to be digitized along with high sampling speeds, this specialized devices are very difficult to design and manufacture in big quantities. To minimize errors due to parameter variability and temperature sensitivity, such nonlinear device will have to incorporate a complex compensating logic with very tight control margins. The problem with this approach is that it is extremely expensive and care must be taken not to introduce intermodulation products in the analog signal due to the use of a nonlinear (logarithmic) function.

The other alternative is to drastically adapt the leading edge ADC technology to implement nonuniform quantization. This will involve huge investments in the semiconductor manufacturing infrastructure and years of research and development - something which hardly justifies its goal. Nevertheless, dynamic range reduction by nonlinear processing is a theoretically viable alternative. It is well documented in the literature and has shown to be particularly effective in the case of speech processing. Applied to software radios, however, this method is prone to intermodulation distortion and furthermore is economically expensive.

8.2.2 Dynamic range reduction by largest signal estimation

A further possible approach to suppress the high dynamic range of the signal at the input of the digital front-end ADC was considered in [27]. The idea behind it is to treat the highest received signal as interferer - much in the same way interference cancellation techniques are employed in CDMA baseband processing.

Figure 8-2 Dynamic range suppression with largest signal prediction

This approach is illustrated in Figure 8-2 for a simple case, where there are only two signal components within the wide bandwidth of the receiver, featuring significantly different power levels. In this approach, the ADC does not digitize the received signal, $x(t)$, but instead digitizes the residual signal, $d(t) = x(t) - \hat{x}(t)$. The signal $\hat{x}(t)$ is the analog equivalent of the digital signal produced by the predictor, which aims to predict the largest received signal component. Ideally, therefore, $\hat{x}(t)$ would be a scaled version of the largest received signal, and consequently the residual signal $d(t)$ would feature a considerably reduced dynamic range when compared to $x(t)$. This dynamic range is controllable and depends on the prediction effectiveness. Hence, in the scheme shown in Figure 8-2, dynamic range suppression is accomplished by predicting the largest signal and subsequently subtracting a portion of it from the overall signal.

A key feature of the above technique is the prediction process. Several interesting issues emerging from this process include the prediction efficiency, the processing requirements of prediction, and the effect of signal characteristics on the prediction. The research results of [27] indicate that first, in most cases, the prediction can be quite efficient because the sampling rate is significantly higher than the bandwidth of the predicted signal, and therefore the correlation between a number of successive samples is large. Second, the key factor which determines the prediction efficiency is not the modulation scheme, but the power difference between the blocking and wanted signals. Simulations showed that to decrease the prediction error considerably, the blocking signal must be at least 10 dB stronger than the wanted signal, and the larger the power difference, the better the system performance. Third, some prior information about the blocking signal, such as modulation scheme, carrier frequency, and baud rate, can be exploited to further improve the prediction gain.

Another important parameter of the arrangement shown in Figure 8-2 is the resolution of the digital-to-analog converter (DAC), which generates the analog form of the predicted signal, $\hat{x}(t)$. The resolution requirements of this DAC can be generally high, but in most cases this requirement can be met in practice.

In the above approach of the dynamic range suppression, the achievable SNR can be expressed as:

$$SNR = \frac{E[x^2(t)]}{E[e^2(t)]} = \frac{E[x^2(t)]E[d^2(t)]}{E[d^2(t)]E[e^2(t)]} = G_p \cdot SNR_{QN} \quad (48)$$

where $E[\cdot]$ denotes average value and $e(t)$ expresses the overall quantization noise. SNR_{QN} stands for the quantization noise of the ADC, while the ratio $G_p = E[x^2(t)]/E[d^2(t)]$, referred to as the prediction gain, specifies the overall SNR improvement that is accomplished with the above prediction technique. For example, assuming that prediction reduces the relative power of the blocking signal from 85 dB to 65 dB, the performance requirements of the ADC in Figure 8-2, processing GSM signals, would be reduced (from 100 dB to 80 dB SFDR). This would yield 3 bits resolution improvement, and therefore a considerable performance benefit would be gained and the SWR application could be more feasible in practice. Another advantage of the above technique is the flexibility to be used in different applications (e.g., in voice encoding systems), since there are no restrictions on the modulation scheme of the blocking signal.

On second thought, though, this approach carries many potential pitfalls which could render it practically unrealizable. As we pointed earlier, the success of this dynamic range suppression scheme depends primarily on the prediction effectiveness. In a more realistic environment, with number of both blocking and wanted signals featuring different power levels across the entire spectrum, the prediction unit will have to cope with many channelization/digital-downconversion-to-baseband/correlation operations on a sample-by-sample basis to effectively predict all signals falling within its marginal range of operation. This evaluates to an immense computational load, in addition to the one associated with the post-ADC stage of the signal processing. Even with optimization (like sharing the channelization/digital-downconversion-to-baseband operations with the subsequent stages) this intense processing may turn out to be practically impossible to accommodate at this stage.

Another drawback is the use of a DAC to regenerate the analog version of the predicted signal. With the discussed in the beginning case of a single blocking signal, one might think that a narrowband, sigma-delta DAC can be effectively utilized to regenerate the needed analog correcting signal. But taking on account the wideband nature of a typical SWR application (combined with a number of randomly distributed blocking signals across the entire spectrum), in reality this type of dynamic range suppression scheme requires a wideband DAC capable of operating at speed at least equal to the ADC's sampling speed. Also, to avoid introducing unnecessary errors from the imperfectly regenerated analog version $\hat{x}(t)$ of the predicted signal, it needs to have very good SNR and SFDR performance. This requirement easily translates itself into additional resolu-

tion bits. Such high resolution, wideband devices doesn't come cheap, though. With the already added cost of the predictor hardware, one might start questioning whether alternative designs like subband processing are more cost-effective and viable in the long run.

8.2.3 Parallel subband downconversion and digitization at IF

Perhaps the most straightforward approach to relaxing the ADC requirements would be the division of the whole reception bandwidth into many subbands and the utilization of many parallel ADCs, one for every subband. An example of such a solution is illustrated in Figure 8-3.

Figure 8-3 Alternative front-end architecture employing subband processing

Especially with receivers for umbrella cell base stations, a very high dynamic range is required to cope with different signal strengths in the macro- and micro-cells. The same can be said about GSM base station receiver specifications, which are considered to be among the most demanding cellular standards. Therefore, the presented architecture is particularly attractive for GSM and multi-standard support systems.

In wideband amplification and downconversion, intermodulation and desensitization effects occur due to the nonlinearities of the LNA and mixer. This causes an upper limit of the dynamic range. Assuming the ideal case of a homogeneous signal spectrum, it can be shown that the dynamic range of the analog part of a downconverter increases inversely proportional to the bandwidth (The bandwidth-dynamic range trade-off). Furthermore, it is well known that for a given ADC technology the product of sampling rate and resolution can be considered constant. A reduced sampling rate therefore transforms into an improved resolution. For these reasons it can be advantageous, instead of downconverting the entire receive band (e.g., with $\Delta f = 25\text{MHz}$), to split the band by means of a filter bank into a number of subbands (e.g., of 10MHz width) which are separately amplified, downconverted and digitized. In this approach the full programmability of signal processing is partially lost. Hence, dynamic range is gained by cost of flexibility and added hardware complexity.

It should be noted, though, that this approach requires careful frequency layout planning and smart dynamic frequency assignment schemes in order to conveniently allocate different power levels in each of the processed subbands. This way, the whole signal dynamic range is partitioned into several zones (each one assigned to a separate subband processing track), which in fact implements dynamic range reduction.

One such proposal is considered in [28] and illustrated in Figure 8-4. The principle of the proposed method is providing separate receivers for strong and weak signals. Hence, the receivers need not exhibit the full dynamic range required of the base station.

Figure 8-4 High/Low gain receiver architecture

With this approach, the functionality of the accessing scheme becomes particularly important. For example, the High/Low gain receiver scheme as applied to the GSM system, will include also a narrowband receiver, because the random access channel (RACH) receiver must be capable of capturing mobile stations accesses at both very high and very low levels. Therefore, it must support the entire dynamic range and, due to the assumed component limitations, have narrowband implementation. As a result, a GSM base station according to the proposal will contain three types of receivers: traditional, narrowband receiver for the signaling channel, or a broadcast channel (BCCH) frequency, high gain wideband receiver for distant mobile stations and a low gain wideband receiver for nearby terminals. Each of these receivers will cover a different frequency band. Thus, the detected signal level will be one of the criteria in radio channel allocation for pending or active communication.

As with any cellular system serving mobile users, one consequence of utilizing such approach will be the increase in intracell handover rates, mainly due to user mobility between the High and Low gain zones within the cell (Figure 8-5).

Figure 8-5 Cell regions in the proposed High/Low gain architecture

In order to avoid hard decisions and errors due to undetected signal level variations, it is best to provide for partly overlapping ranges of the High and Low gain receivers. With this topology, an additional feature of level-based intracell, intertransceiver handover must be designed. This is necessary in order to provide service through the correct receivers as the mobiles move across the regions. For example, if User 1 (situated near the cell edge) is moving towards the centre of the cell, its transmitting power level should be dynamically adjusted to lower values as he moves along. Otherwise, there is very strong possibility that at some point in time it could saturate the high gain receiver and corrupt all other signals, currently received at this particular point. The reason for that lies deep down into the nature of GSM systems power control specifications.

Dynamic power control, as provisioned in GSM, has several aims. The most important are reduction of interference to neighboring cells and extension of MS battery life. Adaptive RF power control is mandatory for the mobile stations, but optional for the base stations. This feature reduces cochannel interference while maintaining the quality of the radio channel. It also decreases the power consumption, which is important for hand-held mobile stations. Provisions are made for 16 different power control levels with 2 dB spacing between adjacent steps. The lowest power level for all mobile stations, regardless of their power class is 13 dBm (20 mW). Referring back to User 1, one can easily imagine a situation where its transmitting power has already been reduced to the minimum, 13 dBm level, while it still continues to move towards the centre of the cell. As its distance from the base station reduces, the probability that it will saturate the high gain receiver increases in accordance with the RF propagation laws.

Prevention of ping-pong handovers near the region borders is another reason for employing the partially overlapping topology and it calls for handover margins, or hysteresis. Naturally, the handover thresholds must be located on the overlapping area of the low and high gain receivers. At this point, though, several questions arise:

- What is the required number of low gain carriers in different base station configurations and environments?
- How would the reservation of low gain carriers affect the cell traffic capacity?

Traffic capacity in the low gain receiver can be minimized if the terminal power control can take full effect before a traffic channel is assigned. This would happen during the call setup signaling on the common control channel (CCCH) and stand-alone dedicated control channel (SDCCH), which are located on the BCCH frequency and therefore served by the narrowband receiver. Required low gain receiver capacity is further minimized if the handover region is placed at the upper end of the high gain receiver input range.

With respect to the second question, one can safely assume that reserving low gain carriers will ultimately reduce the cell traffic capacity. This is easily established by taking on account the random nature of the offered traffic within the low gain receiver serving range. It can be assumed that the loss is largest in small base station configurations due to the well known fact that free-space loss is proportional to the distance squared from the point of emission. However, in this case there is substantial overcapacity in the low gain receiver. Hence, a more intelligent scheme would be needed to allocate channels in the low gain receiver to more distant terminals whenever they are available. For this to happen, the necessary condition is that both receiver ranges overlap not only to a certain degree (where intracell handovers could be managed efficiently), but to a more greater extent with the ideal case of maximum overlap. To manage such a system, though, requires very smart and efficient handover and traffic assignment scheme, which is bound to employ some form of prediction or statistical algorithm.

At this point it should be noted that the necessary low gain capacity will be affected by a number of factors. These include the actual terminal distribution, path loss characteristics, cellular network plan, intertransceiver handover margin, and in particular by the wideband receiver dynamic range.

It may be of interest to notice that an alternative solution would be to divide the channel pool in the time instead of the frequency domain. In practice, this could be achieved by applying different receiver gain during different time slots. Now the allocation of time slots for the mobile terminals should be done based on the received signal level. Naturally, a separate narrowband receiver would still be required for the BCCH frequency.

8.2.4 Conclusions

Wideband multicarrier receivers for software radio are an attractive technology. However, due to technological constraints it may not be possible to achieve the required dynamic range. One method to circumvent these limitations is to employ novel system architectures like the ones examined in this section. This doesn't come without a price, though. A number of difficulties and drawbacks can be encountered with these approaches ranging from reduced system flexibility and extra added hardware to high cost and complex radio resource management. Nevertheless, until the ADC technology is capable of supporting the very demanding requirements of software radios, enabling schemes such as those presented here are being considered worldwide and continuously evolve with ongoing research.

Chapter 9

Conclusions and Future Work

In this final chapter, we present the results and conclusions from the thesis work and summarize potential areas for future research in the field of software radios. Section 9.1 summarizes the research work. The next section suggests some areas for improvements and considerations in terms of software radio receiver hardware design. Section 9.3 presents the most important conclusions from the experiments and finally the thesis concludes with outline of the potential areas for research in the field of software radio converter technologies.

9.1 Summary

The main objective of this thesis was to evaluate the performance of commercially available converter technologies for digital radio applications. Not to a lesser extend, the secondary objective was to relate the experimental results to the requirements of the existing wireless standards in terms of their realization in software radio.

Chapter 1 acquainted the reader with the software radio concept in general and disclosed the motivation behind this research. The benefits of such design were analyzed and the approach to achieve the goal of the research was outlined.

In Chapter 2, we presented in brief the essential ADC performance parameters as they relate to the two main types of converters currently used for PCS (Sigma-Delta and Flash ADCs), each of which has its own shortcomings. The two ADC types were discussed and compared against each other in terms of their applicability to the software radio concept. The advantages of the Flash-ADCs were highlighted as being much more suitable to be employed in a digital receiver architecture. Theoretical $SQNR$ performance expectations were derived for Flash-ADCs of 12 and 14 bits resolution respectively with a general conclusion that performance gets better with increased bit-resolutions. In the final section the reader was familiarized with the data acquisition boards used in the experiments.

Chapter 3 dealt with the experimental setup. In order to achieve industry-wide results acceptance and consistency, the experimental parameters definitions were taken directly from the ADCs manufacturer's data sheets and the experimental details were clearly outlined. Section 2 presented the experimental setup and the general considerations motivating this particular hardware configuration. The final section in this chapter had the purpose of evaluating the experimental setup and assure the reader that it is fit to deliver meaningful results.

Chapter 4 was dedicated to the software needed for the performance measurements. After presenting the software framework in the beginning of the chapter, a detailed description of each software module was subsequently discussed, while the exact code details were included in

Appendix 3. Whenever needed, the particular module implementation was backed up with theoretical explanations and experimental data.

Chapter 5 contained the experimental results for both acquisition boards. For the main part of the chapter, they were presented in graphical form to better illustrate their meaning. It was found that the SFDR and SNR performances of both boards were frequency dependent. They got worse as the input signal frequency was swept from dc to Nyquist. Eventually the answer to this performance mishap was found to be deeply rooted into the boards' design and not related to the ADC components quality. Proof of this was presented in a form of experiment, where the main cause for the performance degradation - the onboard PLL, which supplied the ADC encode signal, was physically bypassed. In this case the SFDR and SNR performances were clearly much less frequency dependent. The reasons behind this were analyzed and attributed mainly to a narrowband jitter process resulting from the PLL inability to track the higher encoding frequencies with the needed accuracy. The Analog Input Bandwidth parameter, which is important in band-pass sampling, was found to be similar for both boards and close to the one specified in each ADC component data sheet. The Channel Crosstalk performance results were satisfactory and not a major design flaw issue.

Chapter 6 was the conclusion of the experimental work. Its first section presented all known sources of impairments during the experiments and on that basis derived an estimate of the results credibility. The next section summarized the overall experimental results in a table and made a comparison between the two investigated converter technologies. To have a common ground, it was chosen to refer to the best performance obtained from the experiments. Several conclusion were derived from the experimental work. With the exception of the PLL design flaw, it was found that performance in general increases with higher bit-resolutions and that both receivers showed performance very close to the one specified for the ADC components used on the boards.

In Chapter 7, we related the experimental results to some of the existing wireless standards requirements. The standards chosen were GSM and IS-95 due to their large market share and the fact that they represent the best achievements in terms of the TDMA and CDMA concepts. Several detailed examples were presented to better illustrate some of the challenges and limitations currently faced in implementing those particular standards in software radio. It was found that different standard architectures required different receiver design. For example, the GSM core standard was most likely to be SFDR-limited with current software radio technologies due to its wide dynamic range and multicarrier detection requirements. Its noise performance in general was dominated by the receiver's ADC quality and with careful RF/IF stages design could be excluded as performance-limiting factor. The CDMA, on the other hand, was found to be practically realizable in terms of recoverable dynamic range at the receiver, but needed extra care during the receiver design phase in terms of background noise and interference issues. In the final section, an effort to relate the experimental results to some of the emerging 3G systems requirements was made. Since non of the proposed 3G radio transmission technologies is yet standardized and widely accepted from the industry, the general trends were presented and on that basis the software radio concept application was discussed.

Chapter 8 concludes the research by acknowledging the limited dynamic range performance of today's ADCs to be one of the most challenging elements in the development of Software Radios. This fact stems not only from the analysis made in Chapter 7, but is widely discussed by the Software Radio research community [25 and 26]. Therefore, there is interest in searching architectures, which would circumvent such challenges. In this context some novel architectures aimed at reducing the front-end receiver' dynamic range requirements are discussed and evaluated with Parallel subband processing being the most promising one.

9.2 Suggested areas for improvements

This section intends to outline some of the performance-limiting factors in a typical software radio design hardware, based on the results from the experiments.

The analog-to-digital converter is the most critical component in any digital radio receiver. In order to achieve the best possible performance one needs to operate this device at its best. This is not always easy to achieve, however. Any deficiencies in driving, clocking, grounding, bypassing, and powering the converters can result in severe degradation of high-speed parameters. This was confirmed in the experiments with the ICS boards, where SFDR and SNR dependence with frequency was established.

One of the major sources for ADC performance degradation is the jitter present in the encode clock signal. In our case it accounted for close to 20 dB difference in performance in some cases. Any design that takes this into account should use a clock source that's as jitter-free as possible. This implies that the link from the clock source to the ADC should be very well grounded impedance matched and short in length. Any buffering for clock alignments, PLLs or gates should be avoided, since every logic circuit adds jitter which is hard to eliminate.

Another issue responsible for receiver performance degradation is an improper matched input impedance. One has to account for the antenna impedance and match it to the input impedance of the receiver to avoid reflections. If the analog-input line has a high impedance, it should be positioned away from fast-slewing clock signals and other digital lines to avoid coupled noise.

In some cases, like with the DCS-1800/PCS-1900 evaluation, mismatched receiver and ADC input impedances can be attributed to unnecessary high gains in the RF/IF stage in order to deliver the received signal amplitude to converter full scale. Any amplifiers in the signal path add signal noise, phase noise, and harmonic distortion, which get worse with high gains. In this manner receiver input impedance should be matched or at least made to be close in value to the ADC component impedance in order to keep the needed gains at minimum.

Wherever possible one should also use linear analog power supplies for the receiver power planes. Digital supplies are extremely noisy in some cases and can compromise severely the accuracy of the receiver.

Extra care should be taken when selecting components for the RF/IF stages. Low noise amplifiers are a must, but in some cases distributing the gain between each section can result in better overall noise figure. Third order intermodulation products (3OP) associated with the amplifiers and the mixer are a key component in the RF/IF front end design. Once created, 3OP are hard to eliminate and are not affected by dithering at the ADC. Due to this reason, wide dynamic range signal applications necessitate very good Front End IM performance.

Amplifier linearity is another issue especially in CDMA applications, where the signal does not have the constant envelope found in GSM for example. After the CDMA waveforms are separately combined with the I and Q channels, filtered and used to modulate the cosine and sine carriers, the resulting composite envelope at the BS receiver is characterized by relatively large peaks that occur periodically. This means that the RF/IF amplifier(s) must have very good linearity in order to deliver the waveform without significant distortion.

9.3 Conclusions from the experiments

High resolution convertors for a wide range of frequencies is a major bottleneck for modern communications systems. The work presented in this thesis demonstrated that, in the limits currently achieved by the convertor technologies, it is possible to implement some of today's mobile communications standards in software radio by using state of the art wideband converters. The main accomplishments are outlined below.

- 1). A digital radio receiver test setup architecture using coherent sampling was proposed. The hardware design was backed with experimental data and theoretical evaluation. Such receiver testing scheme has an improved accuracy over the conventional windowed approach and delivers true converter performance evaluation.
- 2). A complete software framework, reflecting the specific needs of digital receiver testing and evaluation was developed. The software modules were designed with extra care in terms of efficiency and meaningful experimental data results interpretation, while the primary goal was to deliver true parameter estimation with as little as possible influence from hardware or software-specific imperfections.
- 3). An extensive set of experimental data with emphasis on ADC critical performance parameters, including SFDR and SNR.
- 4). A detailed software radio support evaluation for the most popular mobile communication standards, based on the experimental data with currently available converter technologies.

With respect to the overall thesis achievements, several conclusions can be made. The experimental results proved the theoretical concept that increased bit-resolution ADC architectures lead to improvement in converter performance with respect to quantization noise and dynamic range of the recovered signal. There is a significant gap, though, between the theoretically expected performance improvement for every additional resolution bit and the experimentally derived data. In the case with the 12 and 14-Bit converter technologies, the shortcome can be estimated to be around $1/3$ of the theoretically 6.02 dB per additional resolution bit.

The most important conclusion that can be established from the experimental work is that both converter technologies performed close to, or in some cases surpassed the general requirements of the Communication Industry in terms of software radio deployment [Chapter 6, Table 2]. This fact may have an important implication on the software radio design/manufacturing and research industries, since it proves that the software radio concept is a viable and practically realizable alternative to the conventional radio architecture. A good reasoning is the software radio support evaluation made for some of today's operational wireless standards in Chapter 7, where full support for GSM and IS-95 was established in terms of noise and dynamic range limitations.

It should be noted, though, that the converter technology today is still limited and straightforward application of the software radio concept as an universal solution to any wireless application will not in general result in a successful implementation. To a great extent, the receiver hardware layout and configuration is still application dependent and even the best converters available will not suffice without proper and knowledgeable design.

9.4 Proposals for future research

There are many ways in which this research could be extended. The investigations that we set to accomplish with our thesis work cover only the bear minimum needed to evaluate the performance of commercially available converter technologies. Much more work needs to be done in the area of software radios before they can become an integrated part of wireless equipment for commercial purposes. Therefore the author's intention is to outline some of the most important aspects in relation to digital receivers.

Spurious levels reduction by dithering

It is a common practice to improve converters spurious performance by dithering. In essence, it is the out of band injection of noise into the analog signal prior to conversion aimed to reduce the effect of converter nonlinearities, which are the major cause of spurious emission [9]. The mechanism of dithering is a vast research topic on its own, but with respect to the contemporary multicarrier/multichannel wireless standard architectures and their support in software radio, the following points are of great importance:

- i). Is the improvement in spurious performance dependent on the dithering frequency, amplitude, bandwidth, PSD and how?
 - ii). Is there a way to maximize the improvements achieved from dithering in any given situation, and how?
 - iii). In a dynamic multicarrier/multichannel wireless application, what will be the implications of dynamically distributing dither in order not to affect any signal of interest?
-

Two-tone and bandpass measurements

The dynamic specifications in ADC data sheets are optimum figures that manufacturers obtain by testing with optimum test setups. They demonstrate what a device is capable of doing, but it's difficult to realize such performance in practice. What real-world system has a perfect, jitterless clock; a perfect, no-phase-noise, analog driving source; and perfect, noise-free power supplies? Also, a single-frequency test is less stressful to converters than is a multiple-frequency test. In fact, an ADC's performance when sampling a full-scale single frequency at the highest frequency of a band-limited signal is typically a good measure of its worst performance for that waveform, since the internal S/H amplifier of the ADC will often experience the most stress.

Due to the above reasons, two or multi-tone measurement results will have much higher weight as an indication of the true receiver performance. An ideal situation will be to have field measurements, where entire cellular bandwidths (each one with its own distinctive carrier/modulation schemes) are evaluated.

It should be noted that two or multi-tone measurements impose severe restriction in terms of evaluation setup hardware. Practical two-tone measurement schemes involve the use of two high performance signal generators together with a very low insertion loss analog mixer for combining the two signals. Together with the rest of the hardware, this amounts to a significant financial budget - not often at the disposals of the average research facility. For multiple-tone setups, the cost is even higher and the only practical way may involve a custom-made solution.

Nevertheless, such measurements are tremendously important since they provide a guideline about the level of achievable performance, given the original converter specifications.

Novel architectures aimed at reducing the software radio front-end dynamic range requirements

Since receiver specifications vary between cellular standards, the available components may not provide sufficient performance for some systems. This is also true in a multistandard operational environments, where signals with different modulation, coding and accessing characteristics may further increase the composite signal dynamic range. Hence, there is a very big potential in the alternative front-end architecture research. Novel designs should follow the general software radio concept of maximum flexibility in mind and should avoid solutions leading to complex implementation and maintenance - something which could potentially outweigh the benefits of having a software radio design in the long run.

LIST OF REFERENCES

- [1] Mitola J., "The Software Radio Architecture", *IEEE Communications Magazine*, pages 26-38, May 1995.
 - [2] Proakis J., "Digital Signal Processing", *Prentice Hall*, 1996.
 - [3] Harris Semiconductor Corp., "A Brief Introduction to Sigma Delta Conversion", *Application Note AN9504*, <<http://www.intersil.com>>.
 - [4] Frerking M., "Digital Signal Processing in Communication Systems", *Van Nostrand Reinhold*, 1994.
 - [5] Candy J., "A Use of Double Integration in Sigma Delta Modulation", *IEEE Communications Magazine*, pages 249-258, March 1985.
 - [6] Azizi P., Sorensen H., van der Spiegel J., "An Overview of Sigma-Delta Converters", *IEEE Signal Processing Magazine*, pages 61-84, Jan. 1996.
 - [7] Brannon B., "Digital-radio-receiver design requires re-evaluation of parameters", *EDN Magazine*, pages 163-170, Nov. 1998.
 - [8] Sorensen H., Heideman M., Burrus C., "On Computing the Split-Radix FFT", *IEEE Transactions on acoustics, speech and signal processing*, pages 152-156, Feb. 1986.
 - [9] Brannon B., "Overcoming converter nonlinearities with dither", *Application Note AN-410*, Analog Devices Inc.
 - [10] Lee J., "CDMA Systems Engineering Handbook", *Artech House*, 1998.
 - [11] Sampei S., "Applications of Digital Wireless Technologies to Global Wireless Communications", *Prentice Hall*, 1997.
 - [12] Steele R., "Mobile Radio Communications", *IEEE Press*, 1992.
 - [13] Coursey C., "Understanding digital PCS; the TDMA standard", *Artech House*, 1999.
 - [14] Glisic S., "Spread spectrum CDMA systems for wireless communications", *Artech House*, 1997.
 - [15] Redl S., "GSM and personal communications handbook", *Artech House*, 1998.
 - [16] Zvonar Z., "GSM: evolution towards 3rd generation systems", *Kluwer Academic Publishers*, 1999.
-

-
- [17] Larson L., "RF and microwave circuit design for wireless communications", *Artech House*, 1996.
 - [18] Yang S., "CDMA RF system engineering", *Artech House*, 1998.
 - [19] Stephens D., "Phase-locked loops for wireless communications; digital and analog implementation", *Kluwer Academic Publishers*, 1998.
 - [20] Rohde U., "Microwave and wireless synthesizers; theory and design", *Wiley*, 1997.
 - [21] Cripps S., "RF power amplifiers for wireless communications", *Artech House*, 1999.
 - [22] Ojanpera T., Prasad R., "Wideband CDMA for Third Generation Mobile Communications", *Artech House*, page 61, 1998.
 - [23] Furuskar A., "EDGE: Enhanced Data Rates for GSM and TDMA/136 Evolution", *IEEE Personal Communications Magazine*, June, 1999.
 - [24] Austin M., "Service and System Enhancements for TDMA Digital Cellular Systems", *IEEE Personal Communications Magazine*, June, 1999.
 - [25] Benett D. et al., "The ACTS FIRST Project and its Approach to Software Radio", *IEE Colloq. Adaptable and Multistandard Mobile Radio Terminals*, pp. 411-16, 1998.
 - [26] Lackey R., Upmal D., "Speakeasy: The Military Software Radio", *IEEE Commun. Mag.*, vol. 33, no. 5, pages 56-61, May 1995.
 - [27] Nie H., Salkintzis A., Mathiopoulos P., "A New Approach for Mitigating A/D Conversion Requirements in Software Radio Base Stations", *IEEE VTC '99*, Amsterdam, 1999.
 - [28] ETSI TC-SMG, "Digital cellular telecommunication system (GSM Radio Access Phase 3) Radio transmission and reception (GSM 05.05)", v. 6.0.0, January 1998.
 - [29] Travis, B., "EDN Hands-On Project: Demystifying ADCs", *EDN Magazine*, March, 1997.
 - [30] Travis, B., "Remystifying ADCs", *EDN Magazine*, October, 1997.
-

ICS-650 SPECIFICATIONS**ANALOG INPUT**

No. of Analog Input Channels:	2
Analog Connector Type:	SMB Coaxial
Input Impedance:	50 Ohm
Full Scale Input:	±1 Volt
Input Signal Bandwidth:	DC - 200 MHz (-3 dB point)
Max. Sampling Rate:	65 MHz/ch., 2 channels simultaneous
ADC Resolution:	12 bits
Sampling:	Rising edge of internal sample clock, rising or falling edge of external clock (programmable).
External trigger:	Sampling occurs on second clock edge following rising edge of external trigger. Trigger must remain high for at least one clock cycle.
Dynamic Range (S/N+D):	>66.5dB @ 65 MHz
Inter-channel Cross-talk:	-67dB
On-board Storage:	1 Msample/channel

GENERAL

PCI bus Interface:	32-bit 33MHz, Master/Target Burst Mode (DMA)
Front Panel Interface:	FPDP 32-bit, 50 MHz
Environmental:	
Operating Temp:	-0° - 50°C (at entry point of forced air)
Storage Temp:	-40° - +85°C
Humidity:	90% non-condensing
Cooling:	Approximately 200 LFM
Power:	+5V @ 2.2A (without tuner module)
	+5V @ 2.9A (with tuner module)

HP8644A

Frequency Specifications

Range:	0.252 to 1030 MHz
Resolution:	0.01 Hz

Spectral Purity Specifications

SSB Phase Noise @ 1 GHz (20 kHz offset):	-136 dBc/Hz
Nonharmonics (>10 kHz offset):	-105 dBc
Harmonics:	< -25 dBc

HP3335A

Frequency Specifications

Range:	200 Hz to 81 MHz
Resolution:	0.001 Hz

Spectral Purity Specifications

Nonharmonics:	>75 dBc
Harmonics:	-40 dBc

Appendix 3

Programming Code

A3.1 Spurious Free Dynamic Range source code

The outlined code structure below is the main skeleton of the executables for all parameter estimations. In the subsequent parts of this appendix (A3.2, A3.3 and A3.4), only the essential parts of the source code that are different for the particular parameter will be given.

```
***** START OF SOURCE CODE *****

/* Definitions & inclusions */
#define ASYNC_READ
/* #define NO_PLL */
#define WIDE 0
#define NARROW 1
#define TUNER_TYPE WIDE
#define REAL 0
#define COMPLEX 1
#define FFT_TYPE REAL
#define FFTSIZE 16384
#define SAMPLES 32
#define SAMPL_R 60.0
#include <windows.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <winioctl.h>
#include <math.h>
#include "api.h"

/* Source, needed to drive the capture board */
/* Globals */
HANDLE hDevice; // handle to the ICS650 device
OVERLAPPED gOverlapped;
HANDLE hEvent;

/* Local prototypes */
void FFT ( int *, int, int );
void saveFile ( double *, double *, int );
void main(void)
{
    ICS650_CONFIG boardCfg;
    ICS650_WIDE_CONFIG WideConfig;
    ICS650_NARROW_CONFIG NarrowConfig;
    /* char choice; */
    double sampleRate = SAMPL_R;
    int *rbuf;
    int temp;
    int decimation = 1;
    int fftType = FFT_TYPE; // Real or Complex FFT
}
```

```

int          acqSize          = FFTSIZE*SAMPLES; // Size of data to capture
ULONG       actualLength, bResult, nBytesToRead;

// Create Event for overlapped structure
hEvent = CreateEvent(NULL, /* no security attribs */
    TRUE, /* manual reset */
    FALSE, /* initial state CLEARED */
    NULL); /* no event object */

// create the handle to the device
hDevice = CreateFile(
    "\\\\.\\ICS650-1",
    GENERIC_READ | GENERIC_WRITE,
    FILE_SHARE_READ | FILE_SHARE_WRITE,
    (DWORD)NULL,
    OPEN_EXISTING,
    FILE_FLAG_OVERLAPPED,
    NULL);

#ifdef ASYNC_READ
#else
    (DWORD)NULL,
#endif /* ASYNC_READ */
    (HANDLE)NULL);

if (hDevice == INVALID_HANDLE_VALUE) {
    printf("***Error: couldn't open a handle\\n");
    return;
}

// Allocate memory for the receiving buffer */
if ( (rbuf = (int*) malloc (acqSize * sizeof(int))) == NULL ) {
    printf("Could not allocate the memory!\\n");
    CloseHandle(hDevice);
    return;
}

// Reset everything on the board */
ics650BoardReset(hDevice);

// Configure the board */
* printf("\\n\\rEnable tuner? (y/n):");
choice = getch();
if (choice == 'y' || choice == 'Y')
    boardCfg.moduleSel = ICS650_ENABLE; /* Enable Tuner module */
* else */
    boardCfg.moduleSel = ICS650_DISABLE; /* Disable Tuner module */
boardCfg.inttrig = ICS650_DISABLE; /* Clear software trigger bit */
boardCfg.enable = ICS650_ENABLE; /* Enable acquisition */
boardCfg.datapath = ICS650_PCI; /* Set output data path to PCI */
boardCfg.diagenable = ICS650_DISABLE; /* Disable diagnostic mode */
boardCfg.mode = ICS650_CAPTURE; /* Set board to the Capture mode */
boardCfg.clksei = ICS650_EXTERNAL; /* Internal/External ADC clock */
boardCfg.trigsel = ICS650_INTERNAL; /* Internal ADC trigger */
* boardCfg.xclkedge = ICS650_FALLING; /* Rising or Falling edge sampling */
* printf("\\n\\rWhat sample frequency (in MHz):");
scanf("%f", &sampleRate); */

#ifdef NO_PLL
if (sampleRate > 25) /* Choose PLL frequency range */
    boardCfg.freqsel = ICS650_HIGH;
else
    boardCfg.freqsel = ICS650_LOW;
#else
    boardCfg.freqsel = ICS650_HIGH;

```

```

#endif /* NO_PLL */
ics650ConfigSet (hDevice, &boardCfg);
ics650ADCClkSet (hDevice, &sampleRate);
temp = acqSize - 1;
ics650AcquireCountSet (hDevice, &temp);
temp = acqSize - 1;
ics650BufferLengthSet (hDevice, &temp);
temp = decimation - 1;
ics650DecimationSet (hDevice, &temp);
if( boardCfg.moduleSel == ICS650_ENABLE ) {
    if(TUNER_TYPE == NARROW) { /* narrow band */
        NarrowConfig.mode           = ICS650_REAL;
        NarrowConfig.filterSel      = ICS650_80;
        NarrowConfig.offset        = ICS650_DISABLE;
        NarrowConfig.flip           = ICS650_DISABLE;
        NarrowConfig.dbGain         = 0;
        NarrowConfig.bandwidth      = 0.8* sampleRate*1000/64;
        /* decimate by 64 works for narrow & wide in complex */
        NarrowConfig.tuningFreq     = sampleRate/16;
        NarrowConfig.sampleRate     = sampleRate;
        /* Configure tuners in both channels*/
        if(ics650NarrowConfig(hDevice, 0, &NarrowConfig)!=OK)
            printf("\nNarrowConfig() returned ERROR!");
        if(ics650NarrowConfig(hDevice, 1, &NarrowConfig)!=OK)
            printf("\nNarrowConfig() returned ERROR!");
        ics650NarrowAccSync(hDevice);
        ics650NarrowSystemSync(hDevice);
    }
    else {
        WideConfig.mode           = ICS650_REAL;
        WideConfig.gsMode         = ICS650_ENABLE;
        WideConfig.offset        = ICS650_DISABLE;
        WideConfig.flip           = ICS650_DISABLE;
        WideConfig.dbGain         = 0;
        WideConfig.bandwidth      = 0.8*sampleRate*1000/64;
        /* decimate by 64 works for narrow & wide in complex */
        WideConfig.tuningFreq     = sampleRate/16;
        WideConfig.sampleRate     = sampleRate;
        /* Configure tuners in both channels*/
        if(ics650WideConfig(hDevice, 0, &WideConfig)!=OK)
            printf("\nWideConfig() returned ERROR!");
        if(ics650WideConfig(hDevice, 1, &WideConfig)!=OK)
            printf("\nWideConfig() returned ERROR!");
        ics650WideGainSync(hDevice);
        ics650WideSync(hDevice);
    }
    /* The board must be reset in order to load internal counters */
    ics650ADCReset (hDevice);
    /* Enable aquisition */
    if (ics650Enable(hDevice) != OK)
    {
        printf("\nAquisition failed!");
    }

    /* Start the aquisition */

```

```

    if (ics650Trigger(hDevice) != OK)
    {
        printf("\nAquisition problem!");
    }
    temp = 300;
    if (ics650WaitADCInt(hDevice, &temp) != OK)
    {
        printf("\nTimed out waiting for ADC interrupt!");
    }
    else
    {
        nBytesToRead          = acqSize * sizeof(ULONG);
#ifdef ASYNC_READ
        /* setup the overlapped structure fields */
        gOverlapped.Offset      = 0;
        gOverlapped.OffsetHigh  = 0; /* ignore */
        gOverlapped.hEvent      = hEvent;
        /* Do read */
        ReadFile(hDevice, (int *)rbuf, nBytesToRead, &actualLength, &gOverlapped);
        bResult                  = WaitForSingleObject(hEvent, 5000 /* msec */);
        if (bResult != WAIT_OBJECT_0)
        {
            printf("\nERR: Read Timed out ");
            printf("\nErr: %d", GetLastError());
        }
#else
        if (!ReadFile(hDevice, (int *)rbuf, nBytesToRead, &actualLength, NULL)) {
            printf("\nRead file failed");
            return(ERROR);
        }
#endif /* ASYNC_READ */
        else
            FFT ( rbuf, acqSize/SAMPLES, fftType );

        /* Close the device */
        CloseHandle(hDevice);
    }

    /* Entry point for the FFT subroutine */
    void FFT (int *buff, int N, int T)
    {
        int          i, j, k, M, N2, C, K, N4, J, IS, ID, IO, I1, I2, I3, I, N1;
        int          Signal_Bin_Index, max_bin, tmp;
        double        R1, R2, S1, S2, S3, XT, temp;
        double        x[FFTSIZE], y[FFTSIZE];           // Arrays for FFT
        double        SIGNAL[SAMPLES], SPUR[SAMPLES];    // Arrays for samples
        double        *ptrS, *ptrSP;                   // Pointers to SIGNAL and SPUR
        extern double tab[FFTSIZE];                     // Phase factors
        ptrS = SIGNAL;
        ptrSP = SPUR;
        M = 0;
        j = 1;
        while(j<N)
        {
            j<<=1;
            M++;

```

```

    }
    for (i = 0; i < SAMPLES; i++)
    {
        if ( T == COMPLEX )
        {
            max_bin = N;
            for (j = 0; j < N; j++)
            {
                x[j] = ((short int) ((buff[i*N+j]>>16) & 0xffff)); // CH1.
                y[j] = ((short int) ((buff[i*N+j]      & 0xffff))); // CH2.
            }
        }
        else
        {
            max_bin = N/2;
            for (j = 0; j < N; j++)
            {
                x[j] = ((short int) ((buff[i*N+j]>>16) & 0xffff)); // CH1.
                y[j] =                                     0.0;      // CH2.
            }
        }
    }

    ***** START OF THE FFT SECTION *****/
    N2=2*N;
    C=1;
    for(K=0;K<M-1;K++)
    {
        N2=N2/2;
        N4=N2/4;
        for (J=0;J<N4;J++)
        {
            IS=J;
            ID=2*N2;
            do {
                for(I0=IS;I0<N-1;I0=I0+ID)
                {
                    I1=I0+N4;
                    I2=I1+N4;
                    I3=I2+N4;
                    R1=x[I0]-x[I2];
                    x[I0]=x[I0]+x[I2];
                    R2=x[I1]-x[I3];
                    x[I1]=x[I1]+x[I3];
                    S1=y[I0]-y[I2];
                    y[I0]=y[I0]+y[I2];
                    S2=y[I1]-y[I3];
                    y[I1]=y[I1]+y[I3];
                    S3=R1-S2;
                    R1=R1+S2;
                    S2=R2-S1;
                    R2=R2+S1;
                    x[I2]=R1*tab[4*J*C]-S2*tab[4*J*C+1];
                    y[I2]=-S2*tab[4*J*C]-R1*tab[4*J*C+1];
                    x[I3]=S3*tab[4*J*C+2]+R2*tab[4*J*C+3];
                    y[I3]=R2*tab[4*J*C+2]-S3*tab[4*J*C+3];
                } //for I0
            } while (I0 < N-1);
        }
    }

```

```

        IS=(2*ID)-N2+J;
        ID=4*ID;
    } while(IS<(N-1)); // do
    } //for J
    C=C*2;
} //for K

***** LAST STAGE OF THE SPLIT-RADIX FFT *****/
IS=0;
ID=4;
do {
    for(I0=IS;I0<N;I0=I0+ID)
    {
        I1=I0+1;
        R1=x[I0];
        x[I0]=R1+x[I1];
        x[I1]=R1-x[I1];
        R1=y[I0];
        y[I0]=R1+y[I1];
        y[I1]=R1-y[I1];
    } //for
    IS=(2*ID)-2;
    ID=4*ID;
} while(IS<N); // do

***** BIT REVERSAL ROUTINE *****/
J=0;
N1=N-2;
for( I=0;I<=N1;I++)
{
    if(I<J)
    {
        XT=x[J];
        x[J]=x[I];
        x[I]=XT;
        XT=y[J];
        y[J]=y[I];
        y[I]=XT;
    } //if
    K=N/2;
    while(K<=J)
    {
        J=J-K;
        K=K/2;
    } //while
    J=J+K;
} //for I

* Start of the SFDR-specific software */
temp = 0.0;
for ( j = 1; j<max_bin; j++ )
{
    x[j] = sqrt(x[j]*x[j] + y[j]*y[j]);           // Assign rms value
    if ( x[j] > temp )
    {
        temp = x[j];
    }
}

```

```

        Signal_Bin_Index = j;
    }

    SIGNAL[i] = temp; // Save Signal value
    k = max_bin.Signal_Bin_Index; // Number of Harmonics
    for ( j = 1; j <= k; j ++ ) // Take care of Harmonics
    {
        x[j]*Signal_Bin_Index] = 0.0; // Set them to zero
    }

    if ( N%Signal_Bin_Index != 0 ) // If aliases don't fall onto
                                   // the harmonics
        for ( tmp = N; tmp != Signal_Bin_Index; tmp -= Signal_Bin_Index) // Take care of Aliases
        {
            if ( tmp < max_bin ) // If they fall within the
                                // Nyquist spectrum
                if ( tmp > 0 ) // 
                    if ( x[tmp] > 1814.0 ) // If magnitude is > than the average
                                            // noise floor
                        x[tmp] = 1814.0; // Make them equal to the average
                                           // noise floor
                else // Else wrap up for more aliases
                    tmp = N + tmp;
        }

    temp = 0.0;
    for ( j = 1; j<max_bin; j++ )
    {
        if ( x[j] > temp )
            temp = x[j];
    }

    SPUR[i] = temp;
} // for i
free (buff);
saveFile ( ptrS, ptrSP, max_bin );
} // FFT

/* Entry point for saveFile subroutine */
void saveFile ( double *array_S, double *array_SP, int q )
{
    int j;
    double normalize_factor;
    char fname[] = "data01.dat";
    FILE *fp;

    printf ("\nEnter file name with extension:");
    scanf ("%s",fname); /*
    if( fp = fopen ( fname, "w")) == NULL)
        printf ("\nCannot open file %s.", fname);
    else
    {
        printf ("\nFile %s opened O.K.", fname); /*
        normalize_factor = 1.0/(0x7fff*q); // 1/((2^15-1)*max bin)

```



```

for (j = 0; j < SAMPLES; j++)
{
    /* Save data */
    fprintf (fp,"%f ", ((log10(( array_S[j]*normalize_factor)))*20.0) );
    fprintf (fp,"%f\r\n", ((log10((array_SP[j]*normalize_factor)))*20.0) );
}
fclose (fp);
}
/***** END *****/

```

A3.2 Signal-to-Noise Ratio source code

Below is the relevant source code for the SNR parameter estimation. After initializing the board, capturing the data and deriving its FFT in the same manner as with the SFDR parameter source code, the SNR section is entered:

```

/* Entry point after the FFT subroutine */
temp = 0.0;
for ( j = 1; j<max_bin; j++ )
{
    x[j] = x[j]*x[j] - y[j]*y[j];
    if ( x[j] > temp )
    {
        temp = x[j];
        Signal_Bin_Index = j;
    }
}

norm_factor = 1.0/(0x7fff*max_bin);
SIGNAL[i] = 20*(log10((sqrt(temp))*norm_factor));
k = max_bin/Signal_Bin_Index;
for ( j = 1; j <= k; j ++ )
{
    if ( j < 6 )
        x[j*Signal_Bin_Index] = 0.0;
    else
        x[j*Signal_Bin_Index] = 3290596.0;
}

if ( N%Signal_Bin_Index != 0 )
{
    for ( tmp = N; tmp != Signal_Bin_Index; tmp -= Signal_Bin_Index )
    {
        if ( tmp < max_bin )
        {
            if ( tmp > 0 )
            {
                if ( x[tmp] > 3290596.0 )
                {
                    x[tmp] = 3290596.0;
                }
            }
            else
            {
                tmp = N + tmp;
            }
        }
    }
}

```

// Exclude DC
//
// Assign magnitude value
// Find Signal location
//
//
//
// 1/((2¹⁵-1)*max_bin)
// Save Signal value
// Number of Harmonics
// Take care of Harmonics
//
// Only first 5 harmonics
// Set them to zero
// Any other harmonics
// Hide them into the noise
//
// If aliases don't fall onto
// the harmonics
// Take care of Aliases
//
// If they fall within the
// Nyquist spectrum
//
// If magnitude is > than the
// average noise floor
// Make them equal to the average
// noise floor
//
// Else wrap up for more aliases
//

```

    }
    //
    //
    //
    NOISE[i] = 0.0;
    for ( j = 1; j<max_bin; j++ )
    // Initialize to zero
    // Exclude DC
    //
        NOISE[i] += x[j];
    // Accumulate in magnitude
    //
    NOISE[i] = 20*(log10((sqrt(NOISE[i]))*norm_factor));
    // Save sample dB value
    //
    for i
    free (buff);
    saveFile ( ptrS, ptrN );
    }
    FFT

    * Entry point for the saveFile subroutine */
void saveFile ( double *array_S, double *array_N )
{
    int j;
    char fname[] = "data01.dat";
    FILE *fp;
    *
    printf ("nEnter file name with extension:");
    scanf ("%s", fname); */
    if((fp = fopen (fname, "w")) == NULL)
        printf ("nCannot open file %s.", fname);
    else
    {
        * printf ("nFile %s opened O.K.", fname); */
        for (j = 0; j < SAMPLES; j++)
        {
            * Save data */
            fprintf (fp, "%f", array_S[j] );
            fprintf (fp, "%f\n", array_N[j] );
        }
        fclose (fp);
    }
}

***** -END- *****/

```

A3.3 Analog Input Bandwidth source code

Below is the relevant source code for the Analog Input Bandwidth (AIB) parameter estimation. After initializing the board, capturing the data and deriving its FFT in the same manner as with the SFDR parameter source code, the AIB section is entered. It is essentially an estimation of the captured signal energy. To derive the AIB it was started with close to dc analog input frequency and moving up in frequency. For every frequency, the energy of the input signal was measured until a value close to 3 dB less in power from the starting value was recorded.

```

temp = 0.0;
for ( j = 1; j<max_bin; j++ )
{
    x[j] = sqrt(x[j]*x[j] + y[j]*y[j]);
    // Assign rms value
    if ( x[j] > temp )
    {

```

```

        temp = x[j];
        Signal_Bin_Index = j;
    }
    SIGNAL += temp; // Save Signal value
} // for i
free (buff);
SIGNAL = SIGNAL/32;
SIGNAL = 20*(log10(SIGNAL*(1.0/(0x7fff*max_bin))));
printf( "\nr%\e\nr", SIGNAL);
} // FFT

***** -END- *****/

```

A3.4 Average Noise Floor source code

Below is the relevant source code for the Average Noise Floor parameter estimation. After initializing the board, capturing the data and deriving its FFT in the same manner as with the SFDR parameter source code, the Average Noise Floor section is entered. Remember that Average Noise Floor parameter was measured with no analog input signal!

```

NOISE[i] = 0.0;
for ( j = 1; j<max_bin; j++ )
    NOISE[i] = NOISE[i] + (sqrt(x[j]*x[j] + y[j]*y[j])); // Assign rms value
}
NOISE[i] = NOISE[i]/max_bin;
} // for i
free (buff);
saveFile ( ptrN, max_bin );
} // FFT
void saveFile ( double *array_N, int q )
{
    int j;
    double normalize_factor;
    char fname[] = "data01.dat";
    FILE *fp;
    if((fp = fopen (fname, "w")) == NULL)
        printf ("Cannot open file %s.", fname);
    else
    {
        normalize_factor = 1.0/(0x7fff*q); // 1/((2^15-1)*max_bin)
        for (j = 0; j < SAMPLES; j++)
        {
            /* Save data */
            fprintf (fp,"%f\n", ((log10((array_N[j]*normalize_factor))*20.0) );
        }
        fclose (fp);
    }
}

***** -END- *****/

```