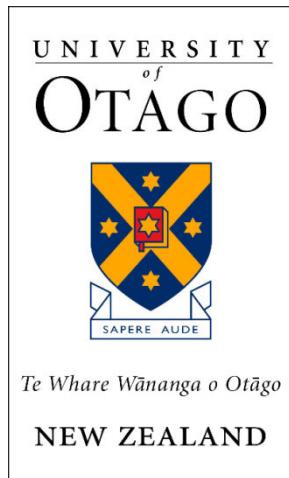


FPGA Based Radio

Ian Scott



A thesis submitted for the degree of

Master of Science

At the University of Otago, Dunedin,
New Zealand

16 May 2011

Abstract

This thesis describes a demonstration *field programmable gate array* (FPGA) based radio intended for high frequency applications. This prototype uses a Xilinx Spartan-3 FPGA to provide many signal processing functions previously executed in a microprocessor or DSP.

The FPGA radio described here has separate receive and transmit functions. Operating frequency and channel bandwidth are software configurable. System test frequency was 13.45 MHz corresponding to a University narrow band frequency shift keying licence.

The system uses three processes – *radio frequency* (RF), *mixed signal devices* (MSD) and FPGA computation. Several Xilinx IP cores are used integrated with Verilog code.

The receive path uses direct *analogue to digital converter* (ADC) signal acquisition using the FPGA system clock at 50 MHz. The transmit path uses a similar approach using a *digital to analogue converter* (DAC) at the same clock rate.

A number of technologies have been used in this implementation. These include *direct digital synthesis* (DDS), *cascaded integrator comb* (CIC) decimation, *digital frequency down conversion* (DDC) to I and Q, complex frequency demodulation, software pulse width modulator DACs with 10 bit resolution and *parameter tuned digital filters* (PTDF).

The system has been successfully built and tested. Receive and transmit channels have behaved as predicted. It is shown that low cost, medium resolution mixed signal devices can achieve land mobile radio communication performance standards. From this, increased MSD bit resolution can be expected to exhibit even higher performance.

Acknowledgements

I would like to thank Dr Tim Molteno for his assistance in this FPGA based radio project. Tim suggested the original direction based on the direct over-sampling approach method for signal acquisition and construction. Since this FPGA Based Radio is intended for general HF communications, over-sampling is well within the capability of current mixed signal devices (MSDs).

I would also like to thank Hilary Lawrence for her critical review and invaluable suggestions for improving this thesis.

Acronyms used in this thesis

Acronym	Text
ACP	Adjacent Channel Power (in dB)
ADC	Analogue to Digital Converter
ALCP	Alternate Channel Power
BER	Bit Error Rate
BJT	Bipolar Junction Transistor
BW	Bandwidth
CIC	Cascaded Integration Comb (filter)
CMR	Common Mode Rejection (amplifier)
DAC	Digital to Analogue Converter
dBm	Decibels relative to 1 mW (50 dBm ↔ 100 Watts)
dBW	Decibels relative to 1 Watt (20 dBW ↔ 100 Watts)
DCM	Digital Control Manager
DDS	Direct Digital Synthesis (for N-Bit word generation)
DFS	Direct Frequency Synthesis (for logic level generation)
EMI	Electro-Magnetic Interference
EVM	Error Vector Magnitude
FIM	Fractional Integrating Multiplier
FSK	Frequency Shift Keying
GBW	Gain Band-Width (linear or product or log of sum)
GMSK	Gaussian Minimum Shift Keying
HF	High Frequency (typical usage is 2 MHz to 30 MHz)
IQ	In-phase Quadrature ($z = I + j \cdot Q$)
IQRF	IQ to RF Modulator
LO	Local Oscillator
MAF	Moving Average Filter
MMIC	Monolithic Microwave Integrated Circuit
MSD	Mixed Signal Device (ADC or DAC)
MSI	Mixed Signal Interface
MSK	Minimum Shift Keying
NCO	Numerically Controlled Oscillator
PCB	Printed Circuit Board
PLL	Phase Locked Loop
PTDF	Parameter Tuned Digital Filter
PWM	Pulse Width Modulation or Pulse Width Modulator
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
RFIQ	RF to IQ Demodulator
RSSI	Receive Signal Strength Indicator
SMD	Surface Mount Device

Table of Contents

Table of Figures.....	7
List of Tables.....	8
Chapter 1 Introduction.....	9
1.2 FPGA Based Radio Project Goals.....	9
1.3 FPGA Based Radio Implementation	10
1.4 FPGA Based Radio Processing Overview	10
1.5 Regulatory Issues	11
2 FPGA Based Radio Hardware Overview.....	13
2.1 System Considerations - Clock Jitter Limits.....	13
3 RF Sub System.....	15
3.1 Receive Path Architecture	15
3.2 Transmit Path Architecture	16
4 Mixed Signal Interface.....	18
4.1 Estimating AD9760 DAC Output Power at Full Scale.....	18
4.2 Predicting AD 9760 DAC Output Signal to Noise Power Density.....	19
4.3 Estimating AD9760 DAC Output Spurious Energy	19
4.4 AD9283 ADC Full Scale Input Power Estimation.....	19
4.5 AD9283 Dynamic Range Estimation.....	21
4.6 Mixed Signal Interface PCB Schematic.....	21
5 Digital FPGA Subsystem	24
6 FPGA Modular Architecture	26
6.1 FPGA Based System Modules	26
6.2 Digital Receiver Architecture.....	26
6.3 Digital Frequency Synthesis.....	29
6.3.1 Potential Digital Frequency Synthesis Options.....	29
6.3.2 Digital Clock Manager– Digital Frequency Synthesis.....	29
6.3.3 Direct Digital Synthesis – Numerically Controlled Oscillator.....	31
6.3.4 Logic Gate Invert and Frequency Divide Method	32
6.3.5 PLL Frequency Synthesiser	32
6.4 Transmit Path	32
6.4.1 Overall Transmitter System	32
7 Receiver FPGA Implementation.....	35
7.1 Receive Frequency Conversion from RF to IQ.....	35
7.2 Decimation Filters	35
7.3 Receive Channel Filter	37
7.4 Second Order Parameter Tuned Digital Filter.....	38
7.5 FM/FSK Demodulator Summary	42
8 Transmitter FPGA Implementation.....	44
8.1 FM/FSK Modulator	44
8.2 Transmit Channel Filter	44
8.3 User Data Interface.....	45
8.4 Level Scaling and Level Shifting	45
9 Measured FPGA Based Radio Performance	47
9.1 Receive CIC and Channel Filter Frequency Response	47
9.2 On Channel Receiver Performance	47
9.3 FPGA Based Receiver Interference Immunity Tests	49
9.3 FPGA Based Transmitter Tests.....	50
9.3.1 Measurement Method.....	50

9.3.2 Transmitter Output Spectrum.....	50
9.3.2 FPGA Based Radio Development and Measurement Location	53
10 Conclusions and Future Development Suggestions.....	55
12 Papers	56
12.1 PTDF Document Presented at ENZCon 2010.....	56
12.2 PTDF Poster Presented First Prize at ENZCon 2010.....	64
12.3 US Patent Cover Page	65
13 Appendix – Detailed Mathematical Derivations	66
13.1 Instantaneous Complex Frequency Interpretation.....	66
13.2 Parameter Tuned Digital Filter (PTDF)	67
13.2.1 PTDF Features.....	67
13.2.2 First Order Parameter Tuned Digital Filter	68
13.2.3 Second Order Parameter Tuned Digital Filter.....	70
14 Appendix B – Verilog Excerpts.....	74
14.1 Top Level Verilog Code.....	74
14.2 Second Order PTDF Verilog Code	75
14.3 10 Bit PWM DAC	75
15 Bibliography	76

Table of Figures

Figure 1 - Converting Analogue Signal Processing to FPGA Implementation	11
Figure 2 - System Overview, showing the Radio-Frequency (RF), Mixed Signal Interface (MSI) and FPGA subsystems.....	13
Figure 3 - Receiver RF Amplifier Using Minicircuits Modules	15
Figure 4 – Transmitter RF Amplifier	16
Figure 5 - Predicting DAC Power into A Resistive Load.....	19
Figure 6 - Estimating Full Scale Input from a Resistive Source	20
Figure 7 - MSD PCB Schematic	22
Figure 8 - PCB Layout, SMD Component Side.....	22
Figure 9 - XC3S1000 System.....	24
Figure 10 - FPGA PCB Layout	25
Figure 11 - FPGA Based Generic Receiver Architecture	26
Figure 12 - CIC Filter Structure	27
Figure 13 - CIC Frequency Magnitude Response with Various N	28
Figure 14 - Complex LO Phase Jitter.....	29
Figure 15 - Statistical Jitter Model.....	31
Figure 16 - Xilinx DDS IP Core.....	31
Figure 17 - Quadrature LO Using Logic Gates.....	32
Figure 18 - FPGA Radio Transmitter.....	32
Figure 19 - Multiplier Symbol	35
Figure 20 - Alias Rejection from a Moving Average Filter.....	36
Figure 21 - Second Order Parameter Tuned Digital Filter.....	39
Figure 22 - Second Order PTDF with Two FIMs and Dual Feedback	39
Figure 23 - Simplified Second Order PTDF	40
Figure 24 - Second Order PTDF Function in Mathcad	40
Figure 25 - Sixth Order PTDF Magnitude Response	41
Figure 26 - Sixth Order PTDF Group Delay	42
Figure 27 - Transmitter Input Data Interface	45
Figure 28 - Receiver Analogue and Digitized Outputs for -10 dBm at ADC Input ...	48
Figure 29 - Receiver Analogue and Digitized Output Waveforms at -114 dBm	48
Figure 30 - FPGA Transmit Spectrum without FSK Modulation.....	51
Figure 31 - Transmitted Spectrum When Modulation Is Applied	52
Figure 32 - Mathcad Simulation for Tx Spectrum with Sinusoidal Modulation	52
Figure 33 - FPGA Based Radio Measurement Area and Equipment.....	54
Figure 34 - ENZCon 2010 PTDF Poster	64
Figure 35 - One Version of FPGA Architecture	65
Figure 36 – Position of The FIM In First Order PTDF	68
Figure 37 - PTDF Frequency Response in dB	72
Figure 38 - PTDF Group Delay in us	73

List of Tables

Table 1 - Unwanted Emissions Limits	11
Table 2 - Emission Category for the University Licence.....	11
Table 3 - Gain and Noise Figure Strategy.....	15
Table 4 - Gain and 1 dB Compression Point Strategy	16
Table 5 - DCM Features and Capabilities (Xilinx)	30
Table 6 - Maximum Allowable Cycle to Cycle Jitter (Xilinx)	30
Table 7 - Maximum Allowable Period Jitter.....	31
Table 8 - Receiver Channel Filter Magnitude Response with Offset Frequency.....	47
Table 9 - Approximate Receiver BER with RF Input Level.....	49
Table 10 - Receiver Co-Channel, Adjacent and Alternate Interference Immunity.....	50
Table 11 - Receiver Blocking and Alias Interference Immunity	50

Chapter 1 Introduction

Software Defined Radios (SDR) use computer CPUs to extract and impart signal information on a RF carrier. Whilst this approach is convenient for programming (e.g. MATLAB), the SDR is limited by its processing speed. Field Programmable Gate Array (FPGA) radio systems benefit from the parallel processing of FPGA systems.

Both approaches are similar in concept. The SDR and FPGA generally use some RF processing combined with digital signal processing techniques. These can use superhet RF-IF architectures or direct RF conversion to I and Q channels. Direct conversion can result in fewer components and complexity. This is desirable especially for superhet approaches that require multiple frequency conversion stages. However some effort can be required to remove I and Q channel errors.

Digital architectures, applied to radio communication systems are gaining popularity over previous hardware intensive approaches. Their programmable architecture offers potential for greater flexibility. This can be advantageous for systems that need to support several modulation standards on a common hardware platform. Additionally, many component manufacturers have moved away from developing hardware processing components. For example, FM demodulation ICs had been the mainstay of many land mobile communication products (NE605, MC3371). Now they become increasingly difficult to source. Similarly, frequency synthesis ICs appear less available. Many factors therefore prompt digital solutions.

The FPGA Based Radio has the additional advantage of being suitable for IC implementation. In contrast the SDR operates on a CPU. The SDR approach is less suitable for single IC conversion. The CPU is also a serial processing engine and will be less capable of wide bandwidth processing than the parallel FPGA Based Radio.

1.2 FPGA Based Radio Project Goals

This project investigates a prototype FPGA Based Radio intended for medium HF operation ~13.45 MHz using FSK. This “proof in principle” platform is intended for experimental purposes and does not represent a direct commercial implementation. The technology explored, in contrast, is directly applicable to commercial applications given subsequent development.

Goals for this FPGA Based Radio are

- To work within the limitations of fixed point numerical processing
- To investigate methods that extend FPGA based radio flexibility
- To develop FPGA signal processing algorithms that lead towards more complex radio systems (e.g. GPS frequency reference using Kalman filter etc.)
- Although less relevant directly, producing multiple versions at some later date had been suggested. Therefore attention to costs and complexity, even in a demonstration platform was applicable.

An additional goal of this masters degree was to gain familiarity with Verilog FPGA programming, especially when applied to radio communication systems.

1.3 FPGA Based Radio Implementation

FPGA based radios can be implemented in many ways. For example, direct conversion may be best suited for FPGA radios operating above 1 GHz. These use *RF to IQ demodulator*[1] (RFIQ) and *IQ to RF modulator* (IQRF) components that translate high signal frequencies to more convenient rates centred at DC. However, components suitable for operation below 1 GHz tend to be losing popularity. In contrast *analogue to digital converters*[2] (ADC) and *digital to analogue converters*[3] (DAC) are plentiful below 1 GHz. These are collectively referred to here as *mixed signal devices* (MSD).

This FPGA based radio is intended for operation below 25 MHz. Its centre frequency is programmed for 13.45 MHz. Eight toggle switches allow this frequency to be selectable in 5 kHz steps from 12.81 MHz to 14.085 MHz with current Verilog code.

A Xilinx Spartan-3 XC3S1000 FPGA[4] is used for numerical processing. This component has 1,000K system gates and 24 dedicated 18 by 18 bit multipliers.

This device is available on a Development board. This FPGA also provides a 50 MHz Temperature Controlled Crystal Oscillator (TCXO) for its system clock and is externally available. This clocks an ADC and DAC mounted on a separate PCB. Well established, cost effective MSDs were selected. The AD9283 8-bit ADC[2] and AD9760 10-bit DAC[3] were considered suitable.

1.4 FPGA Based Radio Processing Overview

The over-sampling approach was used. This provides significant processing gain (~34 dB for channel bandwidth of 10 kHz at a Nyquist rate of 25 MHz). This increases the ADC bit resolution to ~84 dB effective resolution). RF filtering requirements are relaxed allowing simple LC filtering for alias energy removal.

This FPGA radio uses a simple mixed signal interface and minimal RF processing. The bulk of this report therefore focuses on structures implemented in a FPGA (Xilinx Spartan-3 XC3S1000[4]). This platform was used as we have several FPGA-PCBs available with good on board diagnostic support. This is useful for algorithm development that must run in real time. The FPGA Based Radio was characterised with conventional Radio Frequency (RF) measurements.

The project's primary goal is to convert RF signal processing topologies into near equivalent digital topologies. This requires a migration from RF signal processing from hardware to the FPGA domains. An intermediate conversion using MSDs is applied. The general procedure is:

- Development of digital architectures that replicate their analogue counterparts
- Conversion of these architectures to representative software code

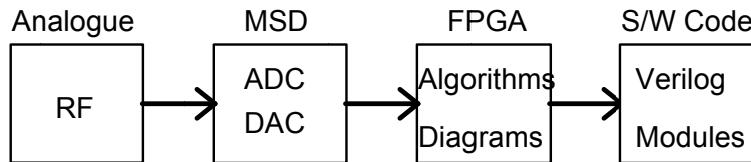


Figure 1 - Converting Analogue Signal Processing to FPGA Implementation

These conversion steps proceed from left to right.

1.5 Regulatory Issues

The University licence is not comprehensive, especially when compared with common land mobile standards. It also does not appear to be self-consistent. When researched[5], the following requirements were revealed.

Table 1 - Unwanted Emissions Limits

Spurious Emissions Limit (peak power)	Frequency Range	Measurement Bandwidth
-56 dBW ¹ (2.25μW) e.i.r.p (59 dBμV/m at 10 metres)	< 150 kHz	1 kHz
	150 kHz to 30 MHz	10 kHz
	30 MHz to 1 GHz	100 kHz
	> 1 GHz	1 MHz

Note 1: dBW ↔ dB relative to 1 Watt, also dBm ↔ dB relative to 1 mW.

Table 1 lists unwanted emissions at -56 dBW i.e. -26 dBm measured in 10 kHz bandwidth. However this reference does not specify the offset measurement frequency. If the measurement offset is ±3 kHz with a measurement bandwidth of 10 kHz then the spurious measurement would extend -2 kHz to +8 kHz on the high side and +2 kHz to -8 kHz on the low side. Clearly this spurious power measurement would contain the allowed transmitted signal.

A minimum feasible measurement offset is ±8 kHz. However this offset implies impossibly sharp spectral energy containment from pass-band to stop-band.

The standard land mobile narrow band specification for 12.5 kHz offsets appears to be more practical. Recent communication with the New Zealand spectral agency has not yet clarified this issue.

The University licence defines the modulation format as 6K00F1DB,

Table 2 - Emission Category for the University Licence

Spectrum Details

Channel	Low(MHz)	High (MHz)	Ref. Freq. (MHz)	Power dBW (eirp)	Emission	Polarisation
	13.447000	13.453000	<u>13.450000</u>	20.0	6K00F1DB	Non-specific

Wikipedia defines this modulation format code as,

6K00	↔	Bandwidth = 6.00 KHz
F	↔	Frequency Modulation FM or FSK
1	↔	Single Channel Modulation, No Sub Carrier
D	↔	Data Transmission e.g. Telemetry or Remote Control
B	↔	Two Condition Data (2-FSK), Fixed Duration and Quantity

This information appears consistent with BW = 6 kHz but does not mention the dB mask normally associated with bandwidth specifications (e.g. @-1 dB, @-3 dB etc.).

2 FPGA Based Radio Hardware Overview

The FPGA radio hardware is implemented as 3 sub-systems with clearly defined physical interfaces

- FPGA processing sub system (the digital radio component)
- Mixed Signal Interface (MSI) sub system (signal acquisition and generation)
- RF processing sub systems (antenna interface)

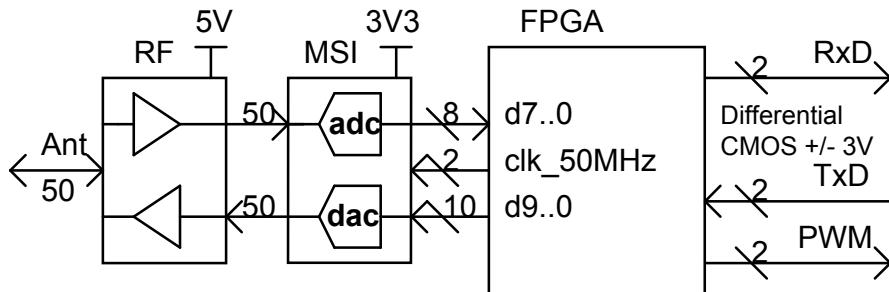


Figure 2 - System Overview, showing the Radio-Frequency (RF), Mixed Signal Interface (MSI) and FPGA subsystems.

The mixed signal and RF blocks represent signal transport mechanisms between the FPGA engine and the antenna port. The design of Low Noise Amplifier (LNA) and RFPA processes is fairly standard practise and has not been overly emphasised here. These will be implemented using connectorized modules from Minicircuits. Replacing each with standard Microwave Monolithic Integrated Circuit (MMIC) gain blocks could lead to a practical PCB. However “proof in principle” is adequately demonstrated with equivalent connectorized modules.

Equally, medium speed and medium resolution ADC and DAC devices are well documented in application notes. It is convenient to fabricate a PCB for these however. The PCB used in this FPGA radio operated as expected. Two diagnostic Pulse Width Modulated (PWM) DACs were created in software and provide 10 bit resolution. These provide analogue FM audio and Receive Signal Strength Indicator (RSSI) diagnostic outputs.

2.1 System Considerations - Clock Jitter Limits

The MSI PCB could however be degraded by clock time jitter τ_j [6]. In high performance systems the clock is generated externally, applied to the MSI interface and then passed to the FPGA. These systems may use 16 bit devices[7] and could require time jitter limits below 0.1 ps to preserve SNR performance. This FPGA receiver uses medium resolution 8-bit ADC devices and 10 ps rms jitter results in comparable SNR.

FPGA devices often use Digital Clock Managers (DCM) for clock distribution. The clock jitter for these modules can exceed 150 ps[8]. This exceeds ADC and DAC requirements and would introduce excessive SNR implementation losses.

The effect of clock jitter on ADC and DAC SNR performance can be estimated[9],[6],[10],[11]. A standard measurement bandwidth at Nyquist is assumed.

$$SNR(\tau_j) = -20 \log_{10}(2\pi f_{in} \tau_j). \quad (1)$$

Setting $f_{in} = 13.45 \text{ MHz}$ and $\tau_j = 10 \text{ ps}$ suggests that the SNR limit caused by jitter is $SNR(\tau_j) = 61.5 \text{ dB}$. This limit exceeds the expected SNR for a 8-bit ADC ($\sim 49.7 \text{ dB}$) by about 12 dB. We conclude that $\tau_j = 10 \text{ ps}$ is tolerable. Equally we conclude that $\tau_j = 150 \text{ ps}$, (typical FPGA DFS), would degrade the ADC performance by 12 dB relative to the predicted value for a 8 bit ADC (again, 49.7 dB). Consequently, considering clock jitter is essential if full predicted ADC SNR performance is to be maintained.

Equation (1) assumes a flat “white” jitter spectrum. In some cases it may impose unnecessarily harsh demands on jitter requirements, especially in cases where the jitter spectrum is significantly different from being white[11]. In this FPGA radio the TCXO jitter spectrum will be relatively flat. It may contain discrete spurious components from the FPGA however, and reciprocal mixing is possible[11]. High performance FPGA based radios will benefit from close attention to clock jitter concerns.

3 RF Sub System

3.1 Receive Path Architecture

Three Minicircuits connectorized gain blocks precede the ADC. These will provide a power gain up to ~ 55 dB. This was expected initially to result in an adequately low noise figure. An intermediate Minicircuits attenuator pad is included at an intermediate point for gain adjustment. In practice, the required amplification will be ~ 50 dB (see chapter 4.4).

The receive line-up uses two cascaded gain modules from Minicircuits followed by an attenuator pad, then a subsequent gain module followed by a low pass filter (LPF). This represents a reasonable approach and suitable modules had been purchased previously. The same topology would be used if a PCB were made using standard MMIC gain blocks. Given this, allocating to design a PCB did not seem justified.

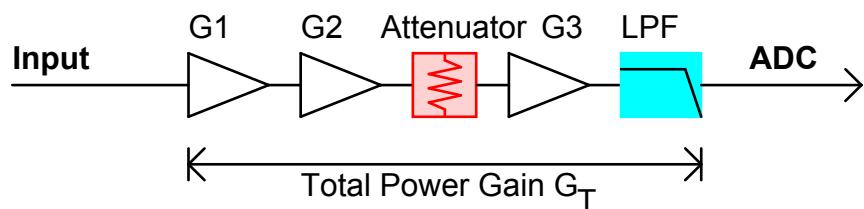


Figure 3 - Receiver RF Amplifier Using Minicircuits Modules

The performance of these modules can be summarised in Table 2 (note – P_{1dB} indicates the output power where 1 dB gain compression occurs)

Table 3 - Gain and Noise Figure Strategy

Parameter	G1	G2	Atten.	G3	LPF	G_T/NF
Model No.	ZX80-4016 ^E	ZFL-1000		ZFL-1000	SLP-10.7	
Power Gain	19.5	18.4	-3dB	18.4	-2.5	50.8
Noise Figure	3.8	6	3dB	6	2.5	3.8
P_{1dB}	17.5 dBm	10.5 dBm	∞	10.5 dBm	∞	8 dBm

Note 1: The NF ~ 3.8 dB exceeds requirements for a HF receiver. Atmospheric noise is typically 30 dB (or more) above thermal noise at 13.45 MHz. Consequently, commercial HF receivers usually aim at NF ~ 30 dB. However this is a demonstration SDR. Also adequate modules were available.

Note 2: The modules were borrowed from a previous GPS project and are not ideally suited to this SDR. However they are adequate. Also the general principles of gain implementation are demonstrated.

Note 3: The SLP-10.7 LPF operates just over its pass-band at 13.45 MHz. It is a low order filter and provides no channel selectivity. Instead its function is to reject the first negative alias (~ 50 dB at 36.55 MHz) from the ADC, clocked at 50 MHz.

Stage G1 dominates the overall system noise figure. Stage G2 provides intermediate power amplification. The attenuator allows fine amplification control. Stage G3 completes the final gain requirement. This has adequate 1dB output power compression (P_{1dB}) commensurate with the ADC input full-scale limit. Finally, the LPF is required to reject noise energy falling on alias responses at the ADC input.

A practical SDR PCB based line-up might include the following

- Input Band Pass Filter to reject potentially strong interferers
- A switched attenuator pad (high / low) to extend dynamic range.

Unlike 1 or 2 bit GPS-like systems, AGC may be inappropriate for FPGA radios. Strong off-channel signals could activate the AGC causing gain reduction to wanted on-channel signals. Avoiding AGC signal modulation effects also requires some care. It appears more sensible to use a switched attenuator that becomes active near ADC overload.

3.2 Transmit Path Architecture

The transmit path amplifies the DAC output to an output power level suitable for transmission. Since Minimum Shift Keying (MSK)[12], Frequency Shift Keying (FSK) and Gaussian Frequency Shift Keying (GMSK) are constant envelope modulation formats there are no in-channel linearity requirements. However, unwanted spurious output harmonic energy needs to be curtailed. It is prudent to limit these both at the DAC output and the final power amplifier output. For example, DAC harmonics and alias energy could cause unexpected frequency combinations when passed through subsequent non-linear amplifiers and appear as spurious products close to the transmitted channel.

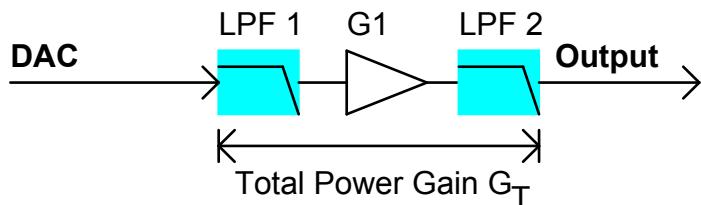


Figure 4 – Transmitter RF Amplifier

Table 4 - Gain and 1 dB Compression Point Strategy

Parameter	LPF 1	G1	LPF 2	G _T
Model No.	SLP-10.7	ZX80-4016E	SLP-10.7	
Power Gain	-2.5 dB	19.5 dB	-2.5 dB	14.5 dB
P _{1dB}	∞	17.5 dBm	∞	15 dBm

Note 1 The transmitter DAC will provide at least -2 dBm RF drive. The corresponding transmitter output power will be ~ 12.5 dBm (15 mW).

In summary, we observe that the FPGA radio requires relatively simple RF processing. The use of broad-band “unconditionally stable” MMIC gain blocks is

relatively common now. These 50Ω devices often use Bipolar Junction Transistors (BJT) with $F_T \approx 70$ GHz and provide drop in gain block solutions from DC to 6 GHz.

4 Mixed Signal Interface

The Mixed Signal Interface (MSI) bridges the RF processing system with the FPGA processing device. It uses two Mixed Signal Devices (MSD) – a DAC for transmit and an ADC for receive.

4.1 Estimating AD9760 DAC Output Power at Full Scale

The DAC generates signals for the transmit RF path. The RF system has a defined input drive power requirement and an output power target. We therefore need to first estimate the DAC output power at full scale into a resistive load. A transmission line balun was used[13] (transformation = 1:1). Shunt resistors are required to restrict the DAC output voltage[3] ($V_{max} = 500$ mV). The output power is therefore equally shared between these resistors and the load termination R_L .

The AD9760 DAC has differential current outputs that can be resistor programmed to supply peak outputs between 2 mA and 20 mA per channel. The 20 mA peak setting reduces the amount of RF amplification required for a given transmitter power. A small benefit in SNR is also obtained. Since half the energy is absorbed in bias resistors and each channel is biased at 10 mA, the resulting differential output current will be $I_{ac} = 10$ mA_{peak}.

The differential DAC outputs need to be converted to signal ended signals (at some point in the transmitter chain). The most convenient point is at the DAC output. There are various methods for achieving this:

- OpAmp Common Mode Rejection (CMR) Topology
- Transformer with N:1 impedance ratio
- Transmission line balun with 1:1 impedance ratio

The OpAmp approach is usually the most familiar method that engineers consider. It has an advantage of operating down to DC. A number of drawbacks exist however:

- An OpAmp operating from single ended supplies below 5 V often have Gain Bandwidth (GBW) < 20 MHz. These would produce severe distortion at 13.45 MHz due to inadequate open loop voltage gain prior to negative feedback.
- A high frequency OpAmp (e.g. 1 GHz) usually requires ± 5 V supplies. These need a dedicated (e.g. capacitor multiplier) producing -5 V given that the FPGA PCB only has +5 V and +3.3 V outputs..
- Input voltage range compliance can be quite restrictive and requires care
- Single balanced-unbalanced CMR stages assume source impedances close to 0 Ω . Also, instability can result if their CMR resistors are too large
- Four CMR resistors are required along with supply decoupling components

Alternatively, transformers or baluns can be used. This FPGA radio adopted a simpler 1:1 transmission line balun, consistent with previous goals preferring simplicity and reasonable costs (if multiple systems are built subsequently). MSI PCB layout was also simplified.

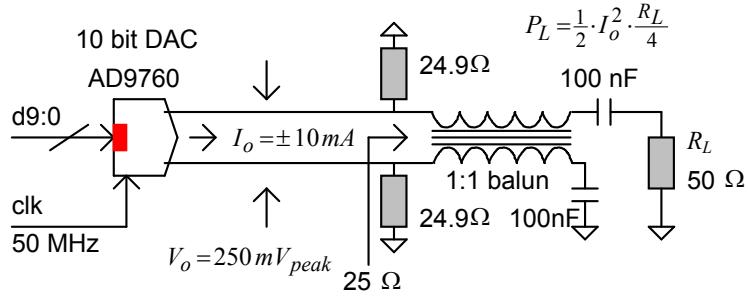


Figure 5 - Predicting DAC Power into A Resistive Load

From the diagram we see that $P_L < 625 \mu\text{W}$ (i.e. $P_L < -2.0 \text{ dBm}$).

4.2 Predicting AD 9760 DAC Output Signal to Noise Power Density

The DAC samples at $f_s = 50 \text{ MHz}$ with bit resolution of $N = 10$. Its output power spectral density will have a best case limit of,

$$SNR_{dB} < 1.76 + 6.02 N + 10 \cdot \log_{10} \left(\frac{f_s}{2} \right). \quad (2)$$

Equation (2) is consistent with standard ADC estimates with an additional linear to dB conversion term added. SNR is used to represent power spectral density for convenience. Note that dB units require some care when power spectral densities are used. Decibels, unlike linear power units do not scale with bandwidth. The best case DAC output power spectral density is therefore $135.9 \text{ dB}_{1\text{Hz}}$ for sinusoidal output. This is appropriate for narrow band FSK as its time domain waveform will appear sinusoidal on an oscilloscope. Also when referred to the 6 kHz channel bandwidth the output SNR will be $\sim 98 \text{ dB}$.

4.3 Estimating AD9760 DAC Output Spurious Energy

The DAC is a sample and hold device. Its output spectrum will exhibit a $\text{sinc}(x)$ function,

$$H(f) = 20 \log_{10} \left(\left| \frac{\sin \left(\pi \frac{f}{f_s} \right)}{\pi \frac{f}{f_s}} \right| \right) \text{ where } f_s = 50 \cdot 10^6 \text{ Hz}. \quad (3)$$

The first zero will occur at $f = 50 \text{ MHz}$ and the attenuation at $f = 13.45 \text{ MHz}$ will equal -1.06 dB . However the output balun will add some extra loss.

4.4 AD9283 ADC Full Scale Input Power Estimation

The AD9283 ADC has differential inputs with a full-scale input range of $\pm 512 \text{ mV}$ [2]. Preceding amplification is single ended. A single ended to differential conversion is needed. Potential options include:

- OpAmp Conversion

- Transformer conversion

Although the OpAmp may appear attractive it has many potential, although not compelling drawbacks

- Single rail OpAmps often have GBW < 20 MHz and have high distortion (a previous survey revealed few devices specified at 13.45 MHz)
- High frequency OpAmps alternatives (1 GHz) often require ± 5 V supplies
- The selected Spartan-3 PCB does not supply -5 V
- Producing -5V could require an additional capacitor multiplier IC
- Further OpAmp bias and feedback resistors would also be needed
- Two OpAmps are needed to produce differential ADC drive
- OpAmps in this class can cost \$10 each = \$20 total

A simpler alternative is to use a transformer[14]. These cost less than \$ 5 and only require two passive components. If a 4:1 version is used, an additional power gain of 6 dB is obtained. Transformers introduce minimal distortion and can have wide bandwidth capability (e.g. the ADC transformer operates from ~2 MHz to ~250 MHz).

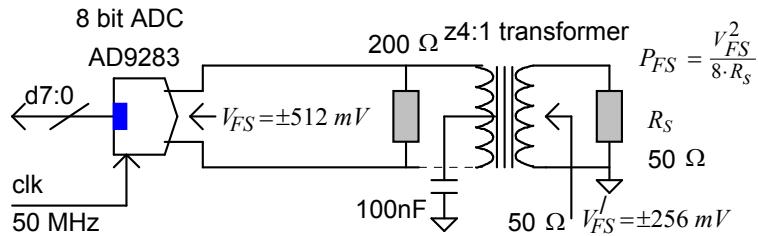


Figure 6 - Estimating Full Scale Input from a Resistive Source

The AD9283 ADC input sensitivity can be expressed as an equivalent noise figure NF. This parameter facilitates system power gain estimation. A convenient NF estimation procedure based on standard principles is,

Full scale input in \pm peak volts	$= \pm 512 \text{ mV}$
Input termination resistance in Ω	$= 200 \Omega$
\Rightarrow Full scale input power in dBm	$= -1.8 \text{ dBm}$
ADC Bit resolution N	$= 8 \text{ bits}$
\Rightarrow ADC SNR in dB	$= 49.9 \text{ dB}$
ADC Sample frequency	$= 50 \text{ MHz}$
\Rightarrow ADC SNR density	$= 123.9 \text{ dB per 1 Hz}$
Thermal noise density	$= -174 \text{ dBm per 1 Hz}$
\Rightarrow ADC Noise Figure (NF)	$= 50.1 \text{ dB}$ (i.e. noise spectral power above thermal spectral noise shown here in dB units)

The procedure predicts RF ADC NF with reasonable accuracy, typically within +2 dB. It is useful for predicting preceding amplification requirements prior to more detailed analysis or device SNR characterization. Further, other noise sources, such as clock jitter, can cause NF degradation and require consideration. To illustrate, if a “noiseless” amplifier with 50.1 dB power gain is placed in front of this example ADC,

the resulting system NF will equal 3 dB. Further, if a real amplifier with $NF = 3$ dB, power gain = 47.1 dB is placed in front, the overall system NF will increase to 6 dB.

4.5 AD9283 Dynamic Range Estimation

It may be asked if 8 bits is “enough”. The ADC dynamic range can be estimated readily. Consider the previous example. The receiver bandwidth is likely to be ~ 6 kHz in this FPGA receiver application.

Consider the following estimation procedure, also based on well understood physical relationships:

Thermal noise density	= -174 dBm per 1Hz
System NF	= 6 dB
=> System noise density	= -168 dBm per 1 Hz
Demodulation bandwidth	= 6 kHz (38 dB)
=> System Noise	= -130 dBm
SNR required for BER = 5 %	~ 10 dB
Receiver sensitivity	= -120 dBm
ADC Full scale range	= -1.8 dBm
Preceding ADC power gain	= 47.1 dB
System full scale input	= -48.9 dBm
=> Direct Dynamic range	= 71.1 dB

This result of 71.1 dB is very respectable despite the use on an 8-bit ADC. It represents an upper limit for dynamic range. Still, even if we hypothesise implementation losses as high as 10 dB the result would still be 61.1 dB.

4.6 Mixed Signal Interface PCB Schematic

The MSI PCB schematic and PCB were drawn on Cadsoft EAGLE. The board was fabricated using the labs’ routing machine. Components were SMD from Digikey.

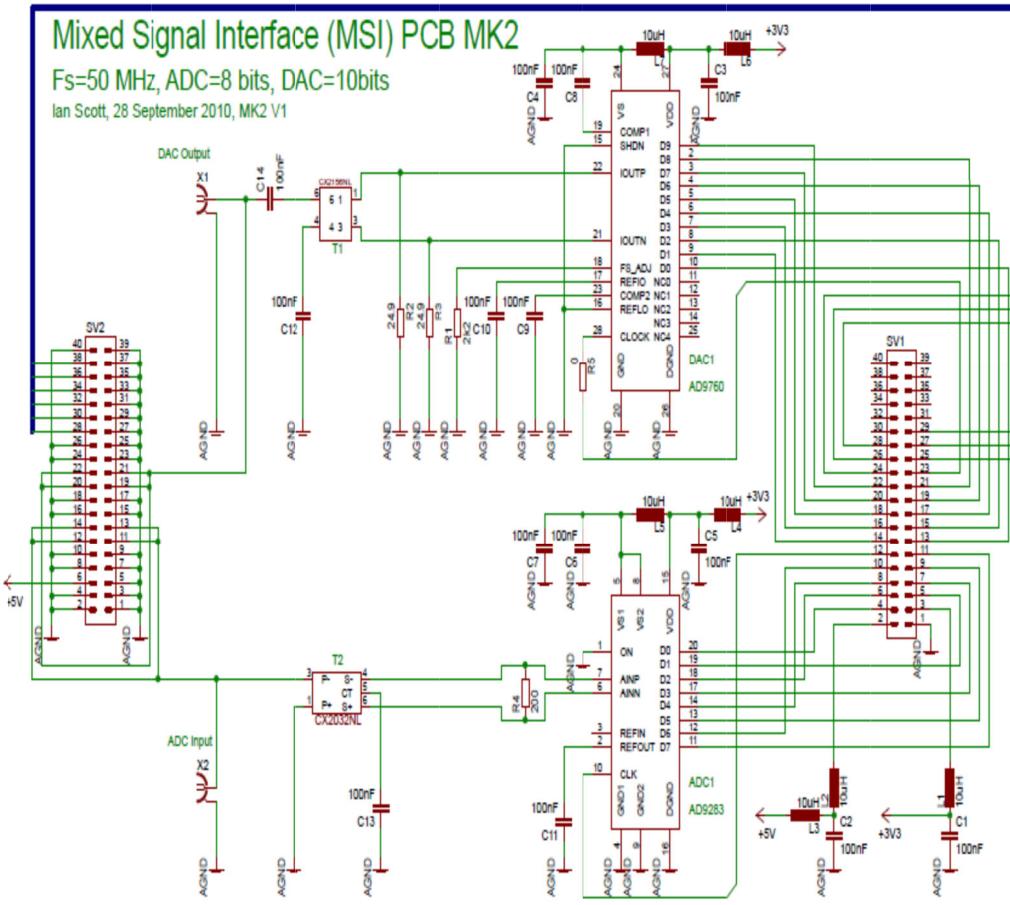


Figure 7 - MSD PCB Schematic

The bold line represents a 6-track bus. The upper symbol represents the DAC and the lower symbol represents the ADC. Balun and transformer appear to the left.
The PCB SMD component side is shown below.

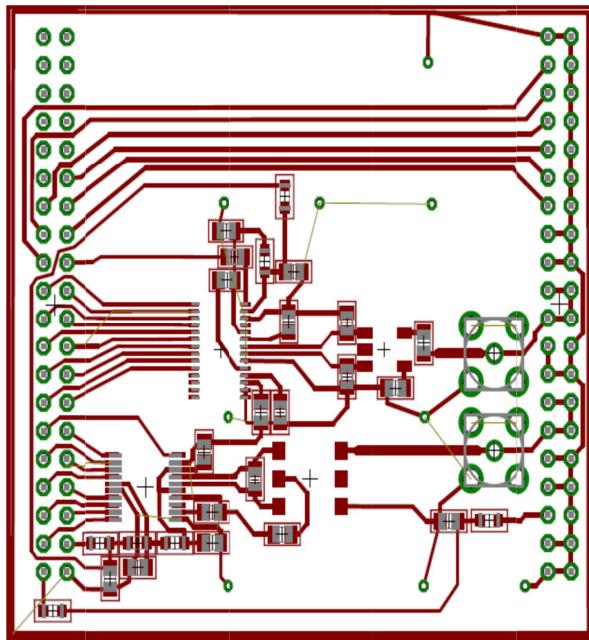


Figure 8 - PCB Layout, SMD Component Side

The PCB layout is shown with the SMD ground plane removed for clarity. SMD components are placed on bottom layer 2. The topside layer uses continuous copper except for connector holes. Wire vias were used to stitch the top and bottom ground planes together to enhance Electromagnetic Interference (EMI) integrity.

The MSI schematic adopts a minimum component count strategy. This is facilitated by the use of an ADC input transformer and a DAC output balun. The RF interface is available at SMA test connectors. It continues to the right on a 40 pin connector intended to connect to a possible RF PCB. The RF impedance is 50Ω .

The MSD data lines operate at 50 MHz and are near equal length with minimum length per device. Using series resistors on the digital lines seemed unnecessary at these moderate frequencies. This was confirmed subsequently. The system operates from a single +3.3V supply. A +5V supply is available for the RF PCB shown lowest on the PCB layout.

There are 6 uncommitted I/O lines left from the FPGA A2 port. These are routed to the RF PCB shown at the top of the PCB. They can also be configured differentially. The B1 port is used for diagnostic 10 bit PWM DACs (implemented in Verilog). Reconstruction filtering is external. This allows the future students to tailor filtering requirements for their application.

Supply decoupling is standardized with 10 uH solid inductors and 100 nF capacitors.

5 Digital FPGA Subsystem

The Xilinx Spartan-3 Rev E PCB (Digikey) houses a XC3S1000 FPGA in a FT256 package[4]. Several PCBs had been ordered for other applications. This device contains 1,000K system gates or 17,280 equivalent logic cells. It has 120K distributed RAM Bits and 432 K block RAM bits. In addition, it has 24 18*18 multipliers, 4 Digital Clock Managers (DCM) and up to 391 user I/O pins.

The PCB contains the following key items:

- 50 MHz Clock TCXO
- Three 40-pin expansion headers (A1, A2, B1)
- Eight slide switches (used to control SDR frequency etc.)
- Four push button switches
- Eight diagnostic LED indicators
- Four character, seven segment alpha numeric LED display (16 bit diagnostic)
- JTAG download cable connector
- XCF02S 2Mbit configuration PROM
- Two 256K × 16 10ns SRAM
- RS232 serial port
- PS/2 port
- VGA port with eight color capability
- 5 Volt unregulated supply input

The PCB is programmed via its JTAG port using a USB to JTAG “platform cable”.

The A2 expansion connector is used for the SDR as this interface has the highest number of free IO pins (26). It is also closest to the FPGA for high speed I/O. However the B2 port is also used for diagnostic, audio and data interfaces.

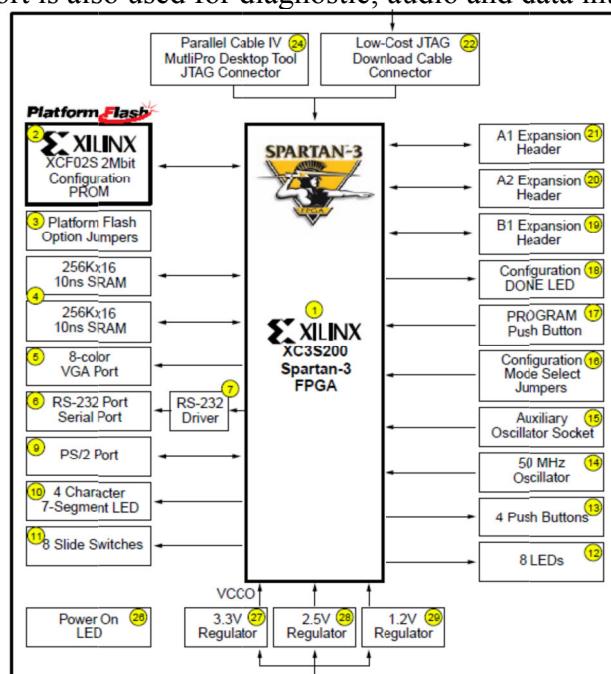


Figure 9 - XC3S1000 System

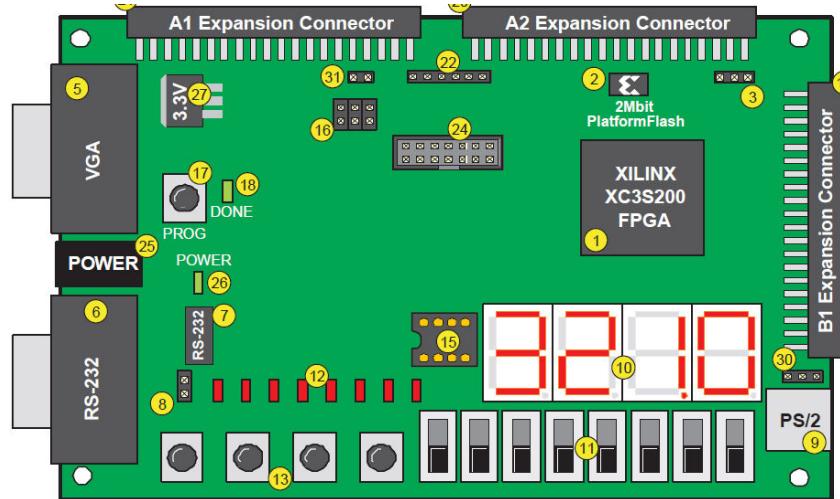


Figure 10 - FPGA PCB Layout

The on board diagnostic LEDs, alphanumeric display and switches were useful for initial RF-software algorithm testing with DC test vectors. To illustrate, all digital filters were implemented with unity processing gain. The ADC and DAC use unsigned integer arithmetic. All other processing needs signed integer arithmetic and use many bit resolution conversions. The FPGA filters were initially tested using the switches as stimulus and the hexadecimal display as a monitor. After DC measurements were secure, dynamic RF testing was conducted using the FPGA.

6 FPGA Modular Architecture

6.1 FPGA Based System Modules

The digital system contains three major systems with several sub systems:

- FPGA Based Receiver Topology
- FPGA Based Frequency Synthesis Topology
- FPGA Based Transmitter Topology

6.2 Digital Receiver Architecture

The FPGA receiver digital processing block diagram is shown below in Figure 11. This structure is generic and the indicated bit resolutions and frequencies can be changed in software to suit alternative applications.

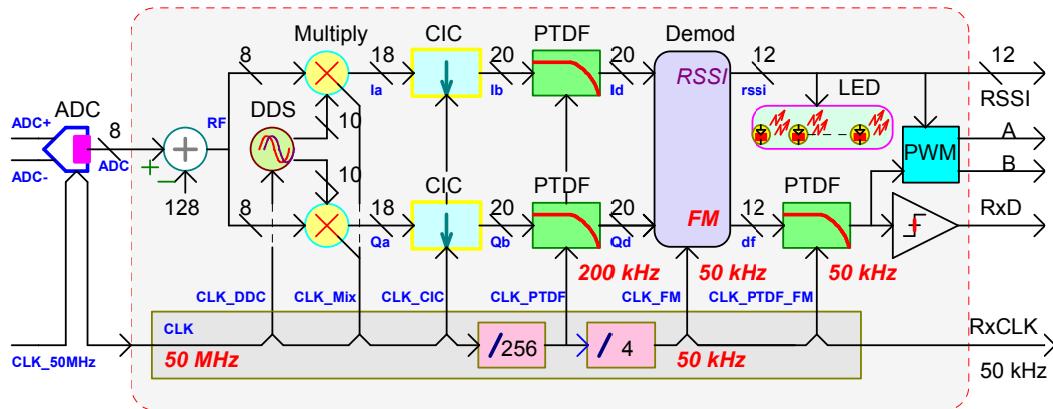


Figure 11 - FPGA Based Generic Receiver Architecture

The RF input signal is digitized with an 8-bit ADC[2]. Its output words are unsigned and range from 0 to 255. These words are converted to signed-digital values by subtracting an offset of 128. The signed values (-128 to 127) are converted to IQ values with two Xilinx multipliers[15] and 10 bit DDS LO[16]. The DDS LO uses higher bit resolution than the ADC to maintain near 8 bit overall resolution.

The multiplier outputs become signed 18 bit numbers. These are passed through IQ Cascaded Integrator Comb (CIC) 256:1 decimation filters[17]. Output bit resolution is increased to allow for processing gain. The full precision equation from Xilinx[17] is,

$$B_{\max} = \lceil N \log_2 RM + B \rceil \quad . \quad (4)$$

In Xilinx' equation N = number of CIC stages, R = decimation ratio (256), M = differential delay (1 or 2) and B represents bit resolution. The bit growth $\Delta B \equiv B_{\max} - B = 48$ for $M = 1$ and $N = 6$. If full precision is actually required, the 18-bit words arriving from the IQ multipliers would balloon up to 66-bits after the CIC decimators. This seems excessive and full precision may not be required.

From a system perspective, the SNR *density* after decimation should equal the SNR *density* before decimation regardless of the number of stages used. This reasoning suggests that $6 N_a + 10 \log\{F_{s_a}/2\} = 6 N_b + 10 \log\{F_{s_b}/2\}$ where the “a” and “b” subscripts refer to before and after bit-resolution and sample rate. It is clear therefore that $\Delta N_{a,b} = \frac{5}{3} \log\{F_{s_b}/F_{s_a}\}$. In comparison to the full precision equation offered from Xilinx, a decimation ratio of 256:1 predicts a bit growth of $\Delta N_{a,b} = 4$.

This system estimate appears more realistic. However its derivation is not rigorous. A practical bit-growth allocation probably lies somewhere in-between. It seemed sensible to use as high a bit resolution as the FM demodulator algorithm would allow. Experimental results suggested that 20 bit output resolution was reasonable, based on the ADC’s 8 bit output, 256:1 decimation and several extra bits to reduce implementation losses. These extra bits are well accommodated in the selected FPGA.

The general CIC filter structure is shown below[17] in Figure 12.

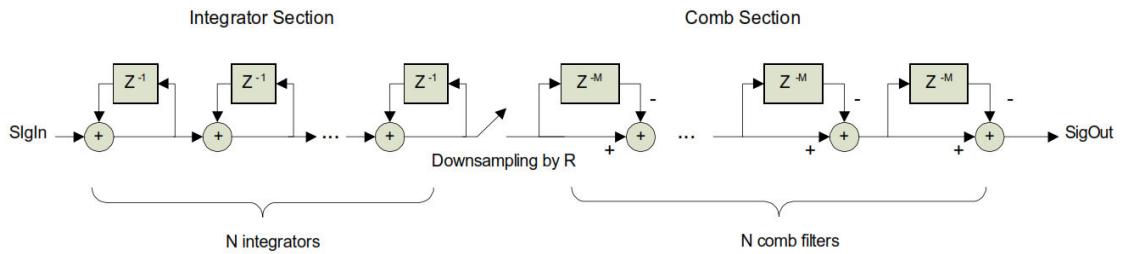


Figure 12 - CIC Filter Structure

This CIC filter has a z-transform equation,

$$H(z) = \left[\sum_{k=0}^{R*M-1} z^{-k} \right]^N$$

$$H(z) = \frac{(1 - z^{-R*M})^N}{(1 - z^{-1})^N} . \quad (5)$$

The typical frequency response for CIC decimation filters[17] is shown in Figure 14.

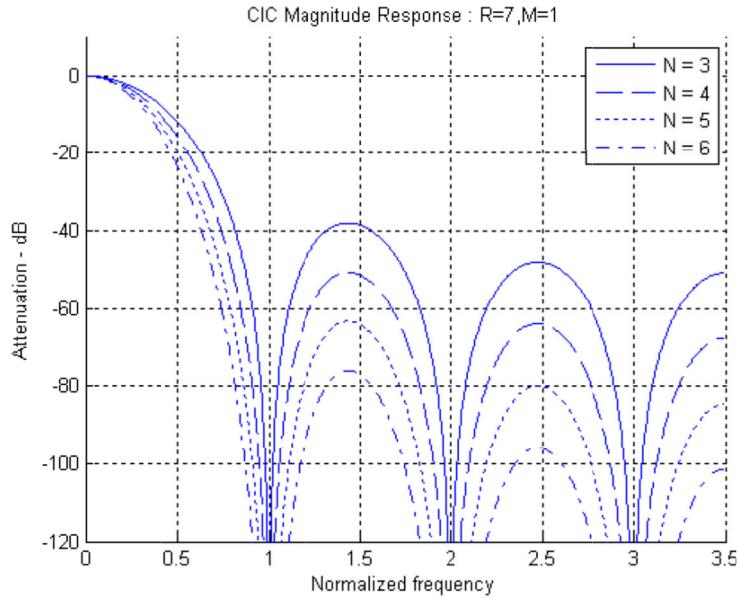


Figure 13 - CIC Frequency Magnitude Response with Various N

This filter provides efficient decimation and interpolation and does not require multipliers. It is however inadequate to used solely as a sharp channel filter as seen above. The channel filter is placed after the CIC decimation filter. This FPGA receiver uses a 4th order Parameter Tuned Digital Filter (PTDF)[18] preceded by a Moving Average Filter (MAF) for alias removal.

The filtered IQ signal is presented to the FM Demodulation Algorithm. The traditional approach applies versions of $\tan^{-1}\{\frac{Q}{I}\}$ depending on which IQ quadrant the signal is in. IQ phase is then estimated and differentiated, producing FM. This approach requires a number of decisions and a simpler algorithm was used based on complex frequency offset estimation[19].

The Verilog module, based on this mathematical representation, uses Xilinx IP multiplier cores[15], a Xilinx IP divider ($R=\frac{P}{Q}$) core[20] and discrete time differentiation. An additional PWM output is provided for RSSI. The RSSI computation uses a Xilinx square root ($\sqrt{\cdot}$) IP core[21] and provides an accurate estimate for IQ magnitude.

The demodulated FM values are low-pass filtered and digitized to 1 bit output data. Two Verilog code based 10 bit PWM DACs are provided for diagnostic purposes. One PWM channel produces 10 bit Received Signal Strength Indicator (RSSI) magnitude and another PWM channel produces 10 bit analogue FM. These additional diagnostics required minimal implementation effort and assisted subsequent testing.

It may be thought that once a signal is digitised, then small time sampling errors within the data should not affect the system accuracy. However in digital signal processing applications this may not be correct. To explain, the real ADC data samples are multiplied by a digital complex Local Oscillator (LO) from a Numerically Controlled Oscillator (NCO). This is produced by Direct Digital Synthesis (DDS) in the FPGA code and has rotating phase. The LO phase directly determines the phase of data in the IQ channels. If the LO phase has sample-to-sample jitter, the “I” and “Q”

channels will also have jitter. Conversely, the digital input can be considered to have created equivalent time jitter with a “perfect” complex LO. This time jitter represents a signal processing acquisition error and is therefore noise, causing SNR degradation (hence the use of 10 bit DDS LO).

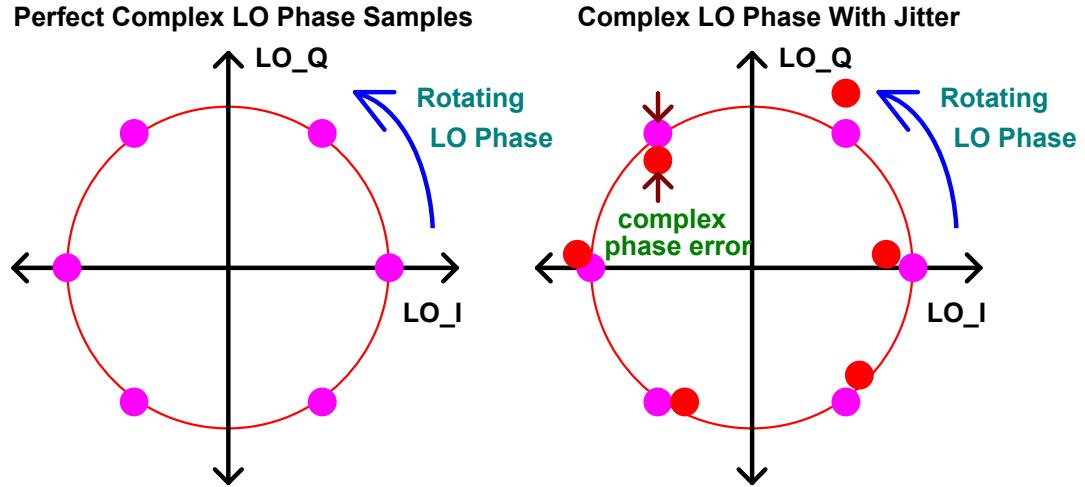


Figure 14 - Complex LO Phase Jitter

Finally the clock distribution strategy deserves some explanation. This FPGA radio uses pipeline-like clock distribution rather than single point star distribution. This was intended to reduce timing errors as the clocks travel with their associated data.

An external clock source is preferred as these can be designed with minimum jitter. The FPGA board can be “noisy” and inject jitter on its internal TCXO. This would be important with high resolution MSDs or high processing frequencies. The clock jitter requirements for an 8 bit ADC operating at 13.45 MHz (10~20 ps) is modest. It appeared reasonable to use the FPGA TCXO clock initially and test performance.

6.3 Digital Frequency Synthesis

6.3.1 Potential Digital Frequency Synthesis Options

The performance of the SDR is critically dependant on the performance of its frequency clocks and digital LO. The relative suitability of potential options requires consideration. Four candidates will be considered:

- Digital Clock Managers (DCM) producing Digital Frequency Synthesis (DFS)
- Direct Digital Synthesis (DDS) - Numerically Controlled Oscillators (NCO)
- Logic Gate Based Invert and Frequency Divide by 2
- Phase Lock Loop (PLL) Frequency Synthesizer

6.3.2 Digital Clock Manager– Digital Frequency Synthesis

The prime frequency ratios between the 50 MHz clock and the 13.45 MHz RF input frequency is,

$$50 \cdot \frac{269}{2 \cdot 2 \cdot 2 \cdot 5 \cdot 5} = 13.45 . \quad (6)$$

The numerator is a prime number (269). This exceeds the division range of the Xilinx DCM[8]. The DCM – DFS frequency ranges are shown in the Xilinx Figure 16 image below.

Table 5 - DCM Features and Capabilities (Xilinx)

Table 1: Digital Clock Manager Features and Capabilities

Feature	Description	DCM Signals
Digital Clock Managers (DCMs) per device	<ul style="list-style-type: none"> • 4, except in XC3S50 • 2 in XC3S50 	All
Digital Frequency Synthesizer (DFS) Input Frequency Range*	1 MHz to 280 MHz	CLKIN
Delay-Locked Loop (DLL) Input Frequency Range*	18 MHz to 280 MHz**	CLKIN
Clock Input Sources	<ul style="list-style-type: none"> • Global buffer input pad • Global buffer output • General-purpose I/O (no deskew) • Internal logic (no deskew) 	CLKIN
Frequency Synthesizer Output	Multiply CLKIN by the fraction (M/D) where M={2..32}, D={1..32}	<ul style="list-style-type: none"> • CLKFX • CLKFX180

The maximum multiplication or division ratio is only 32. The DFS output frequency is defined as,

$$f_{out} = f_{clk} \frac{M}{D} \text{ where } \begin{matrix} M=2..32 \\ D=1..32 \end{matrix} . \quad (7)$$

It is clear that the DFS cannot generate the prime factor 269. However, perhaps it was used in combination with another process for fine adjustment. It is necessary to consider its time jitter performance τ_j as shown in section 2.1.

There are two definitions for time jitter; “cycle to cycle” and “period” jitter. The DFS cycle to cycle jitter table specifications from Xilinx are shown in Figure 17 below.

Table 6 - Maximum Allowable Cycle to Cycle Jitter (Xilinx)

Table 7: Maximum Allowable Cycle-to-Cycle Jitter

Functional Unit	Frequency Mode	
	Low	High
Digital Frequency Synthesizer (DFS)	CLKIN_CYC_JITT_FX_LF ~ ±300 ps*	CLKIN_CYC_JITT_FX_HF ~ ±150 ps*
Delay Locked Loop (DLL)	CLKIN_CYC_JITT_DLL_LF ~ ±300 ps*	CLKIN_CYC_JITT_DLL_HF ~ ±150 ps*

The best cycle-to-cycle jitter limit is $\tau_j = \pm 150 \text{ ps}$

Table 7 - Maximum Allowable Period Jitter

Table 8: Maximum Allowable Period Jitter

Functional Unit	Frequency Mode	
	Low	High
Digital Frequency Synthesizer (DFS)	CLKIN_PER_JITT_FX_LF ~ ±1,000 ps* (±1 ns)	CLKIN_PER_JITT_FX_HF ~ ±1,000 ps* (±1 ns)
Delay Locked Loop (DLL)	CLKIN_PER_JITT_DLL_LF ~ ±1,000 ps* (±1 ns)	CLKIN_PER_JITT_DLL_HF ~ ±1,000 ps* (±1 ns)

The best period jitter limit is $\tau_j = \pm 1000 \text{ ps}$ based on a 7-sigma normal distribution.

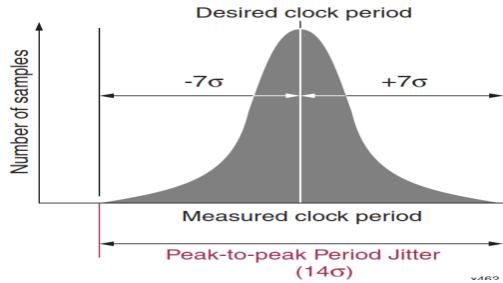


Figure 15 - Statistical Jitter Model

The best time jitter estimate is $\tau_j = 150 \text{ ps}$. Given $f_{in} = 13.45 \text{ MHz}$, $SNR = 38 \text{ dB}$.

Since an 8-bit ADC has $SNR \sim 50 \text{ dB}$, the DFS falls 12 dB short of requirements. Even so, perhaps “special” low jitter DFS sources were available. This DFS output would have single 1 bit resolution. LO harmonics from this square wave source could fold back on alias terms causing jitter. The 1-bit approach would introduce severe SNR degradation unless integer related frequencies are selected. In summary, the DCM approach cannot synthesis the required prime frequency ratio (269), has excessive jitter and only produces a single 1 bit resolution output. It is therefore rejected as a candidate LO source.

6.3.3 Direct Digital Synthesis – Numerically Controlled Oscillator

Direct Digital Synthesis (DDS) is far more promising. This technology is mature and available as a Xilinx IP Core. The DDS core produces extremely clean high resolution sinusoidal and co-sinusoidal outputs that are well matched to FPGA radio applications.

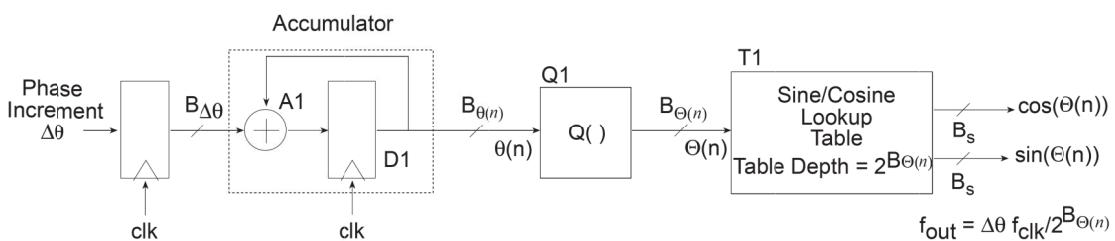


Figure 16 - Xilinx DDS IP Core

Xilinx provides an extremely capable DDS IP core[16] that is compatible with the XC3S1000 FPGA. It can provide spurious outputs exceeding 100 dB below its signal output. This FPGA radio uses two DDS IP cores, one for Rx and the other for Tx. The

Tx frequency is defined by the licence. The Rx can receive on an alternative frequency if desired. This is useful for back-to-back system testing where potential Tx-Rx feedback could be problematic or for occasions where off-channel monitoring might be useful.

6.3.4 Logic Gate Invert and Frequency Divide Method

A common approach used in some RFIQ and IQRF IC's is to use inverters and dividers as shown in Figure 21 below.

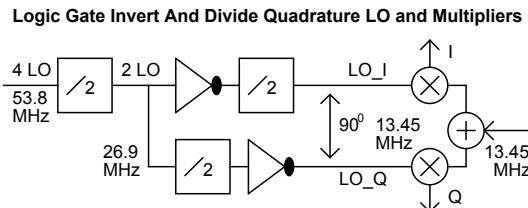


Figure 17 - Quadrature LO Using Logic Gates

This implementation is simple and produces an accurate phase quadrature output. Unfortunately it has no frequency agility. The Spartan-3 TCXO frequency would need to change from 50 MHz to 53.8 MHz. This is feasible. However 53.8 MHz is non-standard. There are no TCXO options from Digikey or other suppliers that produce this output or even any multiple or sub multiple. However an external DDS could be used to generate an accurate external clock at the expense of additional hardware.

6.3.5 PLL Frequency Synthesiser

It is possible to replace the TCXO with a Phase Locked Loop (PLL) frequency synthesiser. However, PLL ICs for frequency synthesis are now uncommon. Few manufacturers appear commercially interested lower volume HF components compared to presumably more lucrative markets above 800 MHz. Other HF components have followed a similar demise; IQ modulator and demodulator ICs and high power RF BJTs. Service replacement components may be obtainable but these do not provide a forward path for FPGA radio technology. In contrast, DDS technology receives extensive investment and will presumably remain supported.

6.4 Transmit Path

6.4.1 Overall Transmitter System

The FPGA transmitter uses DDS for signal generation. Potentially asynchronous input data is over-sampled to reduce jitter. A Moving Average Filter (MAF) is used to restrict transmitted bandwidth prior to modulation.

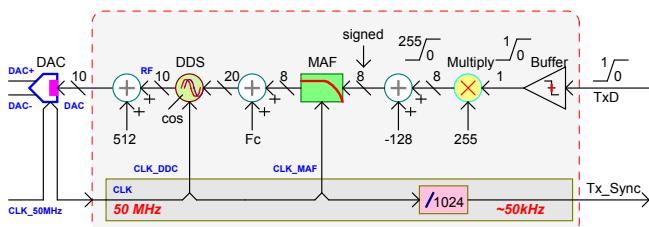


Figure 18 - FPGA Radio Transmitter

Modulation is applied at TxD (1 bit on / off). A buffer is included to ensure that these levels are accurate. Multiplication by 255 converts this data to 8 bit unsigned data. Subtracting 128 then creates signed data ranging from -128 to 127.

The Moving Average Filter (MAF) transforms these abrupt transitions to smooth transitions with intermediate values. This precautionary filter improves adjacent channel power sidebands with minor data distortion. Another offset is applied to control the DDS centre frequency (e.g. 13.45 MHz). The FPGA PCB has 8 slide switches that currently control the frequency in 5 kHz steps.

The DDS incremental frequency step ∂f is[16],

$$\partial f = \frac{f_{clk}}{2^N}. \quad (8)$$

Here N represents the DDS bit-phase resolution. It follows that the DDS output frequency f is programmed by a phase increment integer n where,

$$n = \text{round} \left\{ \frac{f}{f_{clk}} 2^N \right\}. \quad (9)$$

The bit-phase resolution N is determined by defining the maximum FM frequency deviation Δf and required modulation signal to noise ratio SNR (at this frequency deviation Δf). The DDS frequency modulation is a quantized system. Equation 11 predicts the corresponding SNR from a simple ratio,

$$SNR_{dB} \cong 20 \log_{10} \left\{ \frac{\Delta f}{\partial f} \right\}. \quad (10)$$

Note that the conventional term “1.76 dB” is relatively insignificant in this context (DDS bit resolution is extremely flexible) and has been dropped as a nuisance term. Equation 12 now predicts,

$$N = \text{round} \left\{ \frac{\log_{10} \left\{ \frac{f_{clk}}{\Delta f} \right\} + \frac{SNR}{20}}{\log_{10} \{2\}} \right\}. \quad (11)$$

The University licence with modulation format 6K00F1DB states a transmission “bandwidth” of ± 3 kHz. This could be used as an upper limit for Δf . Also, the required SNR is somewhat arbitrary; FSK generally becomes usable when RF SNR exceeds $5 \sim 10$ dB[22]. As previously mentioned, the Xilinx LogiCORE DDS is virtually unlimited for phase bit resolution N . Therefore, we will just adopt a “good” analogue telephone line standard e.g. $SNR \leq 50$ dB . Given,

$$f_{clk} = 50 \text{ MHz}, \Delta f \equiv 3 \text{ kHz}, SNR = 50 \text{ dB} \rightarrow N = 22 \text{ dB}. \quad (12)$$

A more difficult consideration involves adjacent channel power (ACP). The University licence is however unclear on the offset frequency at which this limit applies[5].

A pragmatic approach is to adopt a slightly higher value for N e.g. $N \equiv 24$ bits as the actual value is programmable with little DDS consequence. From equation (8) we therefore predict frequency resolution $\partial f \cong 2.980$ Hz.

Note: The target SNR corresponds to a DAC bit resolution of 8 bits. Adding an extra 2 bits matches the 10-bit DAC used in this FPGA radio.

7 Receiver FPGA Implementation

Receiver design is often more challenging than transmitter design. Whilst transmitters can precisely control signal levels, the receiver has to process wide variations in signal amplitude across a wide range of potentially interfering frequencies.

The FPGA algorithms first translate the desired input signal to IQ channels. This involves two digital multipliers driven by a phase quadrature digital Local Oscillator (LO). This process closely resembles its hardware equivalent. The multiplier outputs represent sampled data at 50 MHz and require decimation to a lower, more convenient rate. Decimation filters follow based on Cascaded Integrator Comb (CIC) filters. The current decimation ratio is 256:1. These filters are effective at decimation but provide relatively poor selectivity characteristics[17]. It is necessary to follow the CIC filters with a channel filter that provides the overall receiver selectivity. Once accomplished, frequency demodulation is appropriate.

7.1 Receive Frequency Conversion from RF to IQ

The digital frequency down-conversion mixers are analogous to analogue double balanced mixers (DBM) that use passive quad-diodes or Gilbert Cell implementations. Two Xilinx LogiCORE multipliers[15] are used. Each is defined with 8-bit input at port “A”, 12 bit DDS quadrature LO at port “B” and corresponding 20 bit output at port “P”. The Xilinx multiplier symbol is shown below in Figure 19.

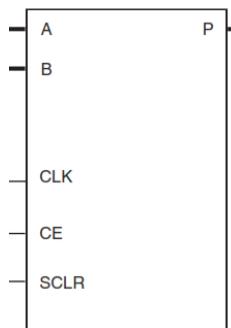


Figure 19 - Multiplier Symbol

Note: Although the Xilinx symbols use labels in capital letters, the Verilog code must identify each entry in lower case.

The DDS LO bit resolution at port “B” must exceed the bit resolution at port “A” to prevent significant SNR degradation. The available bit range is 2 to 64 so this requirement is not problematic. Also, the Cascaded Integrator-Comb (CIC) decimation filters offer significant bit growth compared to the bit resolution at ADC input port “A”. This is explained in chapter 7.2.

7.2 Decimation Filters

The FPGA radio is a multiple-rate system with extensive over sampling compared to the modulation bandwidth (i.e. 50 MHz compared to ~6 kHz). This offers significant

signal processing gain and relaxes hardware requirements. It also allows relatively low bit resolution ADCs to “appear” to have significantly more bits.

To understand this, consider an ADC sampling at 50 MHz. Its effective noise is evenly spread from DC to Nyquist (25 MHz). Decimation to 50 kHz now contains the total noise power from DC to 25 kHz, i.e. 1000 times reduced bandwidth. The spectral noise density is now 1000 times lower whereas the wanted signal power remains unchanged. This ratio corresponds to 30 dB.

The standard formula used for estimating the SNR for a N-bit ADC with sinusoidal input excitation is,

$$SNR \geq 1.76 + 6.02 N \text{ in } dB . \quad (13)$$

Since the decimated SNR has improved by 30 dB, it follows that the bit resolution N has increased by about 5 bits. Therefore, the 8-bit ADC selected in this SDR realization is “acting” like a 13 bit ADC.

It should be appreciated that “decimation” is not the same as simply throwing samples away. Decimation requires interpolation filtering combined with sample reduction.

A corollary to this is that the bit resolution following decimation must increase by a concordant amount, otherwise information is lost and a SNR penalty will occur. FPGA devices allow large bit resolutions without great difficulty. In contrast, DSP devices have fixed bit resolution and may suffer from bit growth. The strategy adopted here is to adopt 18 bit resolution as this is compatible with the 24 18×18 bit multipliers available in the XC3S1000 FPGA[4].

The rate conversion filters can be FIR, CIC or MAF. The CIC and MAF filters avoid multipliers that are required elsewhere. The MAF has the advantage of simplicity and appears adequate. It has a $\text{sinc}(x)$ response. The first zero of the $\text{sinc}(x)$ frequency response is used to remove alias responses prior to decimation (or interpolation). However the MAF provides minimal selectivity so a detailed channel filter is placed at the end of the decimation chain.

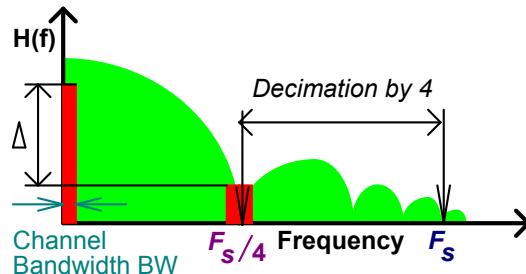


Figure 20 - Alias Rejection from a Moving Average Filter

The alias rejection Δ reduces as the channel bandwidth (BW) increases. This FPGA receiver uses k cascaded CIC filters to ensure that Δ remains high even if wide channel bandwidths are used. These are combined in a single Xilinx IP core. The worst case alias rejection for the cascade is therefore,

$$\Delta_{dB} = 20 k \log\left(\left|\frac{\sin(\pi\Omega)}{\pi\Omega}\right|\right) \text{ where } \Omega \equiv 1 - \frac{f}{F_s} \quad (14)$$

The SDR described here uses a first decimation of 256:1. The intermediate sample rate is therefore $50 \text{ MHz} / 256 = 195.3125 \text{ kHz}$. Let's assume a wide channel bandwidth of $f = \pm 10 \text{ kHz}$. The attenuations Δ for various k become

Order $k = 1$	$\Delta = 25.4 \text{ dB}$
Order $k = 2$	$\Delta = 50.8 \text{ dB}$
Order $k = 3$	$\Delta = 76.1 \text{ dB}$
Order $k = 4$	$\Delta = 101.5 \text{ dB}$

A first order CIC provides only 25.4 dB anti alias rejection. This corresponds to a spurious response in a receiver or a spurious output for a transmitter. A second order CIC filter is superior but perhaps borderline. A third order CIC filter offers 76.1 anti-alias rejection and this is exception (by most radio standards). The forth order CIC is significantly over specified. However the Xilinx CIC IP core offers orders up to $k = 6$ [17]. Consequently this SDR will adopt third or higher order CIC filters for decimation and interpolation. This approach allows easy migration to higher bandwidths in other applications if required.

The filter response can be expressed in the z domain. This is useful as the form translates directly to Verilog script.

$$H(z) = \prod_{k=0}^{K-1} \sum_{n=0}^{N-1} z^{-n}. \quad (15)$$

Here, K represents the CIC filter order and N represents the decimation ratio ($N=256$) in this FPGA receiver.

7.3 Receive Channel Filter

As shown, the CIC provides excellent alias rejection without the need for multipliers. The CIC filter however is somewhat “lazy” as a channel filter. The solution is to use a composite approach; the CIC filter is used for preliminary decimation prior to a more selective “channel filter”.

The FIR filter may represent a first choice. However FIR filters tend to have some drawbacks;

- They may need a significant number of multipliers. For example, a 256 tap FIR may need as many as 256 multiply operations.
- Even if implemented “serially” the upper frequency operation is then limited
- FIR filters often require a very moderate fractional bandwidth Ω i.e. $0.1 < \Omega < 0.5$ to restrict size. The parameter $\Omega \equiv 2 F_c / F_s$.

- Large FIR filters have long throughput delay. This latency adds to “echo” in voice systems especially when cascaded nodes link world areas
- The FIR is not dynamically flexible. If the FIR frequency response needs modification, there is no clear way to modify its coefficients in a tuneable way. This becomes more obscure as the filter size increases.
- The FIR response can however be modified by recomputing its coefficients. However the resulting FIR size may then need readjustment

Sometimes the FIR is replaced with an IIR filter. These can be smaller for a given frequency response mask. This reduces latency but other weaknesses in the FIR filter remain.

Although the FIR filter may be widely used, it appears to be inappropriate for a FPGA radio channel filter. Perhaps FPGA based radios should be flexible where possible.

The Spartan-3 PCB uses a XC3S1000 FPGA device. This has 24 dedicated 18×18 bit multipliers[4]. Significant time-shared multiplier reuse would be needed to implement a 256 tap FIR. Even then, not all these multipliers are available and many are used elsewhere. For example, consider the sample rate to be ~ 200 kHz and that a 256 tap FIR is to be implemented serially. If one dedicated multiplier is multiplexed, then it must operate at $256 * 200$ kHz = 51.2 MHz.

This FPGA radio will use an alternative Parameter Tuned Digital Filter (PTDF) as described in a paper presented to ENZCon 2010[18]. This structure offers the following features:

- It is completely tuneable with 1, 2, 3, or 4 parameters etc.
- The parameters are directly and intuitively related to the frequency response
- Synthesising the parameters is simple – closed form solutions are shown
- The PTDF is can be cascaded to form higher order responses
- The PTDF uses only one multiplier per filter “order”. N=4 uses 4 multiplies
- The PTDF can operate and extremely low fractional bandwidth Ω without penalty. Its size and computational overhead remains constant.

7.4 Second Order Parameter Tuned Digital Filter

The first order PTDF demonstrates the general approach to synthesising these topologies. However it is desirable to have higher order structures. The basic “building block” structure is second order. This structure is shown in Figure 21.

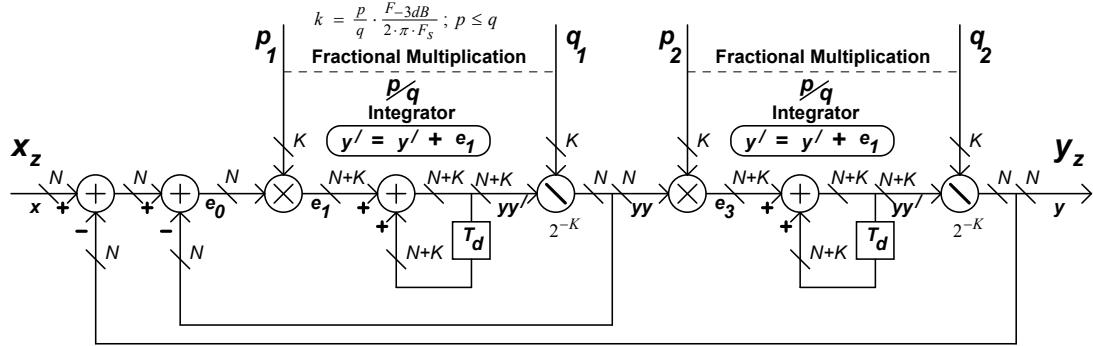


Figure 21 - Second Order Parameter Tuned Digital Filter

This has two negative feedback paths. The inner loop represents the previous first order PTDF. A scaled discrete time integrator follows this. Negative feedback is then applied around the whole filter. The method of summation is of course arbitrary – the drawing shows two separate summations for clarity.

The second order PTDF uses 2 Fractional Integrator Multipliers (FIM). Each FIM contains a multiplication, integration and a power-by-two division. This strategy allows the integrator function to implement scaling ratios between 0 and 1 with fixed point (integer based) arithmetic.

It is worthwhile to note that the order of operations in fixed point arithmetic is important. For example the operation $7/2 = 3$. However the operation $2/7 = 0$.

The FIM components are shown clearly in Figure 22 below.

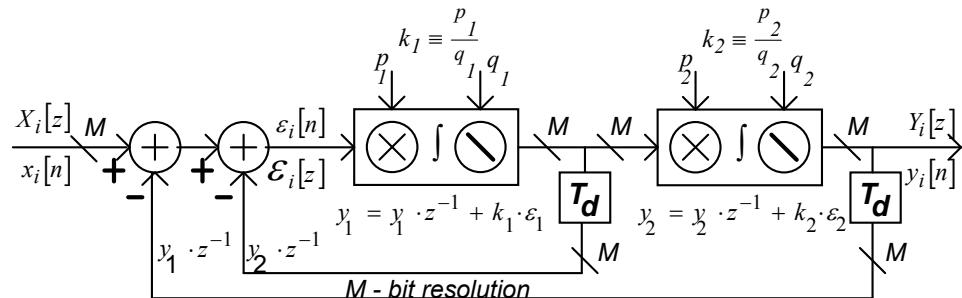


Figure 22 - Second Order PTDF with Two FIMs and Dual Feedback

The explicit representation for each FIM can be omitted to simplify analysis.

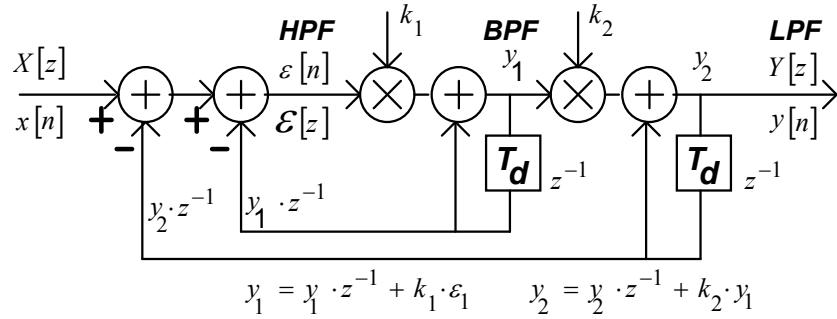


Figure 23 - Simplified Second Order PTDF

The PTDF low pass output has a z-transform model shown in equation 16.

$$LPF_2(z) \equiv \frac{y_2}{x} = \frac{k_1 k_2}{1 + (k_1 k_2 + k_1 - 2)z^{-1} + (1 - k_1)z^{-2}} \quad (16)$$

A small frequency approximation can be made to derive a frequency domain transfer function. The low-pass output is shown below in equation 18.

$$LPF_2(\Omega) \approx \frac{1}{1 + \frac{j}{k_2} \Omega - \frac{1}{k_1 k_2} \Omega^2} \quad (17)$$

Unlike conventional FIR and IIR filters, this second order PTDF provides low-pass, band-pass and high-pass outputs from the same structure as shown in Figure 23.

It is also worthwhile to note the simplicity by which the PTDF can be implemented. The following Mathcad demonstrates this for the second order example, and provides low pass, band pass and high pass outputs from one algorithm.

```
PTDF_2(k1, k2, x) := | Samples<- rows(x)
                        yy<- 0
                        yout<- 0
                        for n ∈ 0.. Samples - 1
                            εε <- x_n - yout
                            ε <- x_n - yy - yout
                            yy<- yy + k1·ε
                            yout<- yout + k2·yy
                            y_{n,0}<- yout
                            y_{n,1}<- yy
                            y_{n,2}<- ε
                        y
```

Figure 24 - Second Order PTDF Function in Mathcad

Note that three outputs occur simultaneously in the columns of y corresponding to low-pass, band-pass and high-pass functions. Equally, the direct z-transform function can be used to predict frequency response.

The second order PTDF is a building block that can be used to create higher order filters. These can be cascaded. The following example shows a cascade of three second order PTDFs. These are offset 5 dB to show their identical shape (otherwise they overlay directly). The blue upper trace predicts these using z-transforms. The lower red trace applies an impulse and uses FFT analysis.

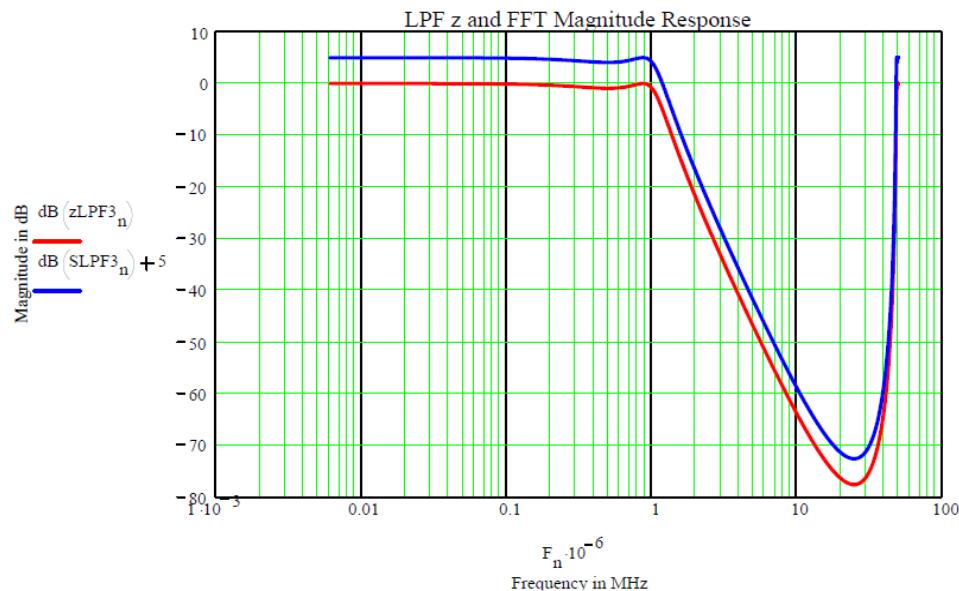


Figure 25 - Sixth Order PTDF Magnitude Response

The PTDF has repeating responses at multiples of the sample rate. It therefore needs a preceding CIC, FIR, IIR or MAF filter. To illustrate, the receive multipliers will provide output energy at $2 F_s$. This unwanted energy would interfere with the demodulator algorithm leading to a distorted output.

The PTDF differential group delay has also been computed by both methods. This is shown in Figure 26 below.

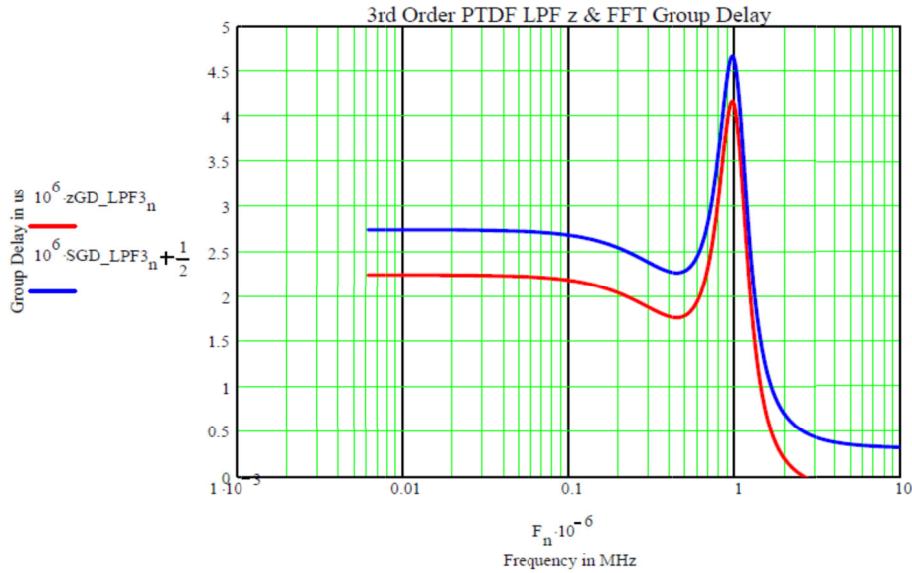


Figure 26 - Sixth Order PTDF Group Delay

A sample rate of $F_s = 100$ MHz has been selected with a pass-band of 1 MHz ($\Omega = 0.02$) to show typical PTDF performance. The resulting differential group delay is not excessive and can be reduced to almost zero by reducing the rate of stop-band attenuation. This is controlled by the ratio of the p_j parameters.

7.5 FM/FSK Demodulator Summary

The method of “instantaneous complex frequency estimation” implemented in this FPGA receiver[19]. A brief summary is listed here although more detailed explanations are included in the Appendix sections.

The complex baseband signal $z(t)$ is defined as,

$$z(t) \equiv I(t) + j Q(t). \quad (18)$$

Then,

$$\log(z) = \log(|z|) + j \phi(z). \quad (19)$$

After some algebra (using \circ to represent differentiation w.r.t. time),

$$\dot{\frac{z}{z}} = \dot{\frac{r}{r}} + j \omega \text{ where } r \equiv |z| \text{ and } \omega \equiv \dot{\phi}. \quad (20)$$

Expressing frequency demodulation δf in Hz rather than radians implies,

$$\delta f = \frac{1}{2\pi} \operatorname{Im}\left\{\dot{\frac{z}{z}}\right\}. \quad (21)$$

Converting to discrete time,

$$\Delta f_n = \frac{1}{\pi \Delta t} \cdot \text{Im} \left\{ \frac{(\Delta I \sum Q + \Delta Q \sum I) + j(\Delta Q \sum I - \Delta I \sum Q)}{(\sum I)^2 + (\sum Q)^2} \right\}$$

with $\sum I \equiv I_n + I_{n-1}$ and $\sum Q \equiv Q_n + Q_{n-1}$
and $\Delta I \equiv I_n - I_{n-1}$ and $\Delta Q \equiv Q_n - Q_{n-1}$

(22)

It is now clear that,

$$\Delta f_n = \frac{F_s}{\pi} \frac{\Delta Q \sum I - \Delta I \sum Q}{(\sum I)^2 + (\sum Q)^2} \quad \text{where } F_s \equiv \frac{1}{\Delta t}. \quad (23)$$

Equation 23 is compact and provides FM demodulation with a single equation. Alternative approaches either use digital implementations of conventional hardware based phase locked loops or IQ phase estimation of $\theta = \tan^{-1} \left\{ \frac{Q}{I} \right\}$ followed by differentiation $\omega = \frac{\partial \theta}{\partial t}$ [23]. In comparison, implementing equation 23 is straightforward and it has no inherent SNR implementation losses (associated with intermediate decisions) during demodulation.

8 Transmitter FPGA Implementation

The transmitter has been optimized for efficient FSK generation. It uses direct DDS modulation. A more general architecture would mirror the receiver architecture in reverse. This would be appropriate for arbitrary modulation format generation e.g. QPSK, QAM or OFDM. This generalization can be implemented in FPGA code at some later date, without hardware changes.

8.1 FM/FSK Modulator

The modulator uses a Xilinx DDS IP Core[16] for carrier frequency generation and direct frequency modulation. The DDS is well suited to this direct application as per previous SNR considerations in chapter 6.4.1.

8.2 Transmit Channel Filter

It is probably worthwhile to discuss some of the advantages associated with the transmit channel filter. This uses a MAF and has several functions:

- Constrains the modulating data bandwidth prior to the DDS
- Limits Error Vector Magnitude (EVM) degradation after filtering
- Introduces minimal amplitude overshoot

The transmitter output spectrum will resemble Minimum Shift Keying (MSK)[12] if the channel filter is omitted. Unlike other modulation formats, MSK is defined by its method of generation rather than on system parameters. Although claimed as “spectrally efficient” this term can be misleading. In crowded spectral environments, being a “good neighbour” is far more important. This requirement is emphasised in the University licence[5]. MSK has high side-lobe energy. The first side-lobe is as high as -25dB. Both filtered FSK and GMSK offer superior side-lobe rejection.

Filtering the incoming data improves Adjacent Channel Power (ACP). There is a trade-off between the extent of filtering and EVM. A common approach in GMSK is to apply a Gaussian low pass filter to the modulating data. The frequency response has a “smooth” roll-off producing smooth data transitions. The ratio of -3dB pass-band frequency to data rate provides a trade-off between ACP reduction and EVM.

Unlike Quadrature Amplitude Modulation (QAM) systems, the channel filter should not exhibit amplitude overshoot. If this occurs, resulting frequency excursions will extend into the adjacent channel and increase ACP. This is what the channel filter was meant to prevent. Consequently, FSK systems use modulation filters that have a smooth roll-off in the frequency domain and minimal amplitude overshoot.

Since FM/FSK is a non-linear modulation format, the output spectrum does not mirror the filtered data spectrum. Even with harsh data filtering, the transmit output spectrum will still mirror familiar FM Bessel sideband behaviour.

Consequently, the primary requirements for the transmit data filter is to provide some spectral containment without producing excessive EVM or modulation overshoot.

Many filter types satisfy these requirements. This FPGA transmitter uses a simple moving average filter (MAF) but Gaussian types can be substituted if required.

8.3 User Data Interface

This FPGA transmitter accepts asynchronous or synchronous input data. The data interface is shown in Figure 27.

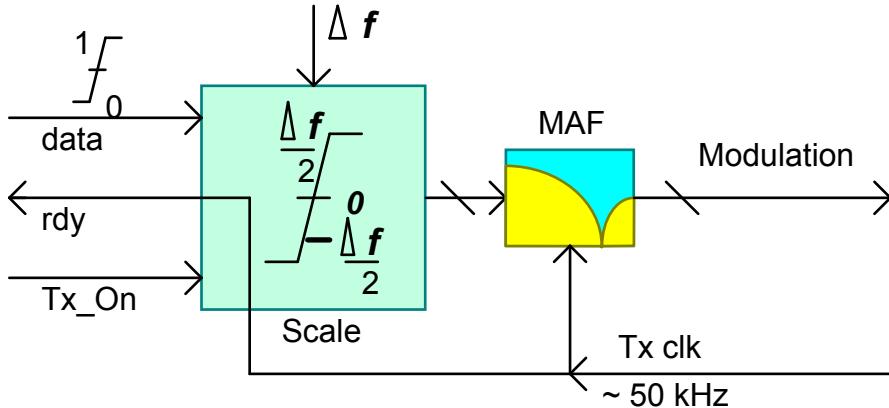


Figure 27 - Transmitter Input Data Interface

The default transmit carrier frequency offset is 0 Hz when inactive. This offset is programmable. When Tx_On is active high, input binary data [0, 1] maps to target transmit frequency offsets $\pm \frac{\Delta f}{2}$. Note that Δf represents frequency deviation in this chapter as opposed to DDS frequency offset used in chapter 6.4.1.

The mapped bit-phase resolution needs to exceed the minimum limit derived earlier as 22 bits. It will therefore be set as 24 bits to allow some margin.

The length M of the MAF determines the degree of data filtering. Multiple MAFs can be cascaded to modify frequency response roll-off. Alternative the FPGA transmitter can replace the MAF with a Gaussian LPF.

The relationship between frequency deviation Δf and transmit data rate is flexible. The FPGA based radio uses non coherent modulation and demodulation so that exact integer frequency relationships are unnecessary. The deviation is defined to be equal to the bit-rate i.e. $(H \equiv \frac{\Delta f}{Bit\ Rate} = 1)$ initially. The value of parameter H is programmable. The deviation also needs to be less than the bandwidth ± 3 kHz. The FPGA radio will use ± 2 kHz as a starting point. This is similar to the deviation used in narrow band FM with 12.5 Hz channel spacing as used in land mobile communications. The expected bit rate will then equal 2 kB/s.

8.4 Level Scaling and Level Shifting

The required peak to peak frequency deviation is determined by the variable Δf applied to the “scale” module. The signed output is applied to the transmit channel filter with a centre offset $\partial f = 0$. For convenience the numerical value and its corresponding target frequency offset will be used interchangeably although they are

of course different. The DDS input is unsigned and has a numerical offset applied as per the previous DDS frequency programming equation $n = \frac{f}{f_{clk}} 2^N$ in equation (8).

9 Measured FPGA Based Radio Performance

9.1 Receive CIC and Channel Filter Frequency Response

The receiver selectivity is limited by the frequency attenuation characteristics of its composite IQ filters. Three filters are cascaded; the CIC decimation filters, a MAF and the PTDF[18] prior to the demodulation algorithm. Additionally, the frequency response should be reasonably flat in the pass-band e.g. ± 3 kHz.

The filter was characterized by applying an unmodulated carrier at the ADC input and measuring the receiver RSSI output. The frequency was varied in 1 kHz steps.

Table 8 - Receiver Channel Filter Magnitude Response with Offset Frequency

Offset kHz	0	± 1	± 2	± 3	± 4	± 5	± 6	± 7	± 8	± 9
RSSI mV	365	365	365	375	375	205	65	15	4	0
RSSI dB	0.0	0.0	0.0	0.2	0.2	-5.0	-15.0	-27.7	-39.2	$-\infty$

As shown, the magnitude response is reasonably flat up to ± 4 kHz and then falls sharply off to below measurement capability beyond ± 9 kHz.

The gradual magnitude roll-off caused by the CIC and MAF filters was compensated by frequency peaking adjusted in the PTDF.

9.2 On Channel Receiver Performance

A test stimulus was presented directly to the receiver ADC at -10 dBm using 1 kHz (2 kB/s) square wave FM modulation with ± 2 kHz deviation. Figure 32 shows the demodulated signal after the PWM DAC. The output data was captured on a Cleverscope CS328A Oscilloscope.

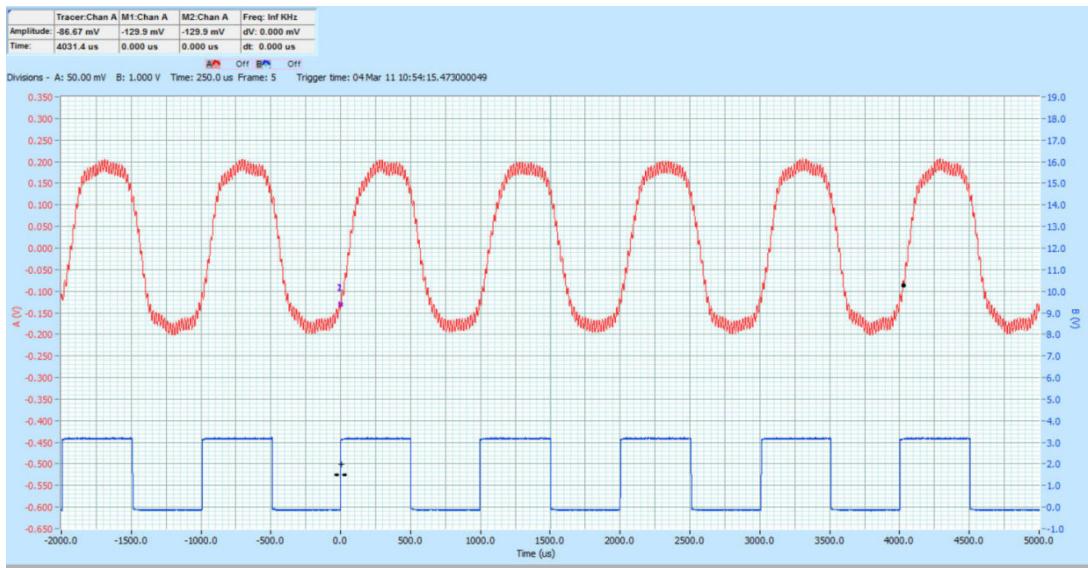


Figure 28 - Receiver Analogue and Digitized Outputs for -10 dBm at ADC Input

The top trace shows the analogue output voltage from the PWM DAC. The waveform exhibits a smooth low pass filtered characteristic without overshoot. Some high frequency PWM leakage is visible due to the simple RC reconstruction filter used for measurement. The Verilog PWM DACs produce significant energy around their sampling frequency (~50 kHz) and a higher order reconstruction filter would be preferable and assist signal observation. The lower trace shows the receiver's digitized output voltage. This has a 50% duty cycle as expected and transitions from 0 Volts to ~3.3 Volts. The waveform was derived from a FPGA comparator placed after a MAF connected to the FM demodulator module in the FPGA (see figure 12).

The RF system was placed in front of the ADC and the same signal modulation was applied at a power level of -114 dBm. This is close to the limit of receiver sensitivity.

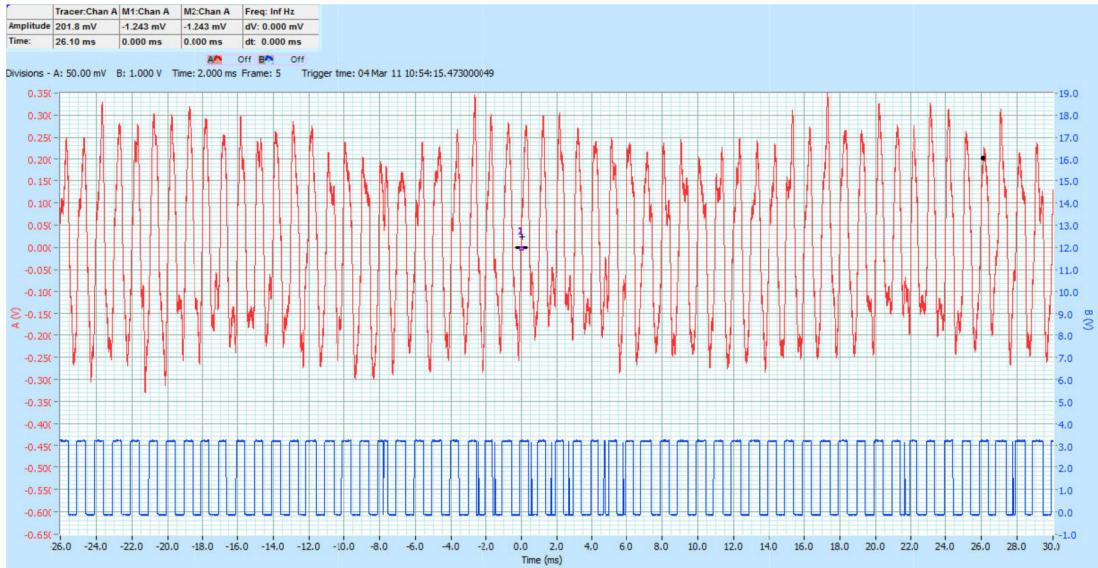


Figure 29 - Receiver Analogue and Digitized Output Waveforms at -114 dBm

The top trace represents the analogue FM output channel and the bottom trace represents the digitized output. There are about 56 ones and 56 alternating zeros in this rudimentary assessment. Also there are approximately 10 glitches that might represent an error. The approximate Bit Error Rate (without error correction) is therefore $\sim 10 / 112$ bits = $\sim 10\%$ at this signal level and modulation.

The average digitized duty cycle remains close to 50%. This indicates reasonable symmetry in the demodulation algorithm in the presence of noise.

The test was then repeated at six power levels and the BER was estimated as before.

Table 9 - Approximate Receiver BER with RF Input Level

Power Level dBm	-120	-117	-114	-111	-110	-108
Approximate BER %	100	50	10	< 1	0	0

The BER shows a sharp transition from complete errors and zero errors. The method is subjective. For example BER = 100 % would actually imply one half the receiver outputs are correct and one half are wrong. In this method BER = 100 % implies that it would be impossible to tell correct decoding from incorrect decoding.

A detailed characterisation would require BER test equipment and a pseudo-random test sequence. This was not available at the time of measurement so an alternative method was used based on equipment that was available. As mentioned previously, the University licence places no requirements on the receiver. However some characterization, however inaccurate, remains useful.

9.3 FPGA Based Receiver Interference Immunity Tests

Two signal generators were applied to a -6 dB combiner. Signal generator A was the wanted signal and B was the interfering source. Generator A had standard modulation at 2 kHz deviation at 1 kHz rate (2 kB/s square wave). Generator B had 2 kHz sinusoidal modulation at 400 Hz rate.

A CS328A oscilloscope was used to monitor the receiver analogue FM output in spectrum analysis mode. The limit of receiver immunity to unwanted interference was defined as

- The 1 kHz component equalled the 400 Hz component for co-channel interferer
- The FFT noise floor increased by ~ 10 dB for other interferers

Generator "A" was set to -108 dBm. This corresponds to -114 dBm at the receiver RF input. The receiver rejection limit was recorded as the difference in level between generator "B" and generator "A" in dB.

The CS328A does not supply a SNR or SINAD output and a suitable measuring device was not available. The results are arguably subjective.

Table 10 - Receiver Co-Channel, Adjacent and Alternate Interference Immunity

Co Channel	Adj. Channel +12.5 kHz	Adj. Channel -12.5 kHz	Alt. Channel +25 kHz	Alt. Channel -25 kHz
13.450 MHz	13.4625 MHz	13.4375 MHz	13.475 MHz	13.425 MHz
0 dB	60 dB	59 dB	71 dB	68 dB

Table 11 - Receiver Blocking and Alias Interference Immunity

Blocking + 1 MHz	Blocking -1 MHz	1 st ADC Alias	2 nd ADC Alias
14.450 MHz	12.450 MHz	36.55 MHz	63.45 MHz
78 dB	57 dB	105 dB	106 dB

From section 4.5 the predicted receiver sensitivity was -120 dBm with a dynamic range of 70.1 dB. Although unsophisticated measurement methods were used, the best case outcomes lay between -114 dBm and -117 dBm. The implementation loss appears to exceed 3 dB. The dynamic range is between 59 dB and 78 dB. One explanation could be that inadequate bit resolution was employed in the FPGA numerical processing. The predicted performance was based on the ADC imperfections alone and did not include implementation losses attributed to FPGA numerical processing. It is reasonable to expect implementation losses from these additional contributors. Further, only rudimentary measurement techniques were available at the time of measurement. Measurement bias could be either favourable or unfavourable.

Additionally the alias responses exceed predictions from chapter 4. This should be expected as the two anti-alias low pass filters from Minicircuits reject these interferers.

A curious observation was observed. When signal “B” was increased 1 dB above the limit stated in table 11, the expected 1 kHz spectral term jumped to 400 Hz. Mechanisms in the FPGA receiver are not evident. It may be possible however that the strong power level from generator “B” deactivated the output of generator “A”.

9.3 FPGA Based Transmitter Tests

9.3.1 Measurement Method

A square wave modulated signal was applied to the receiver ADC input using the previous modulation (2 kHz deviation at 1 kHz rate). The digitized output was then passed to the FPGA transmitter internally. This system processed the data as described previously and applied filtered modulation to its internal DDS. The FPGA output connects to the DAC on the MSI PCB. The RF output was displayed on a spectrum analyser.

9.3.2 Transmitter Output Spectrum

A HP 8596E spectrum analyser was available for spectral analysis. Its reference level was set to -5 dBm with centre frequency = 13.45 MHz and span = 25 kHz. Since this

instrument did not have electronic data transfer a camera was used to record its screen display.

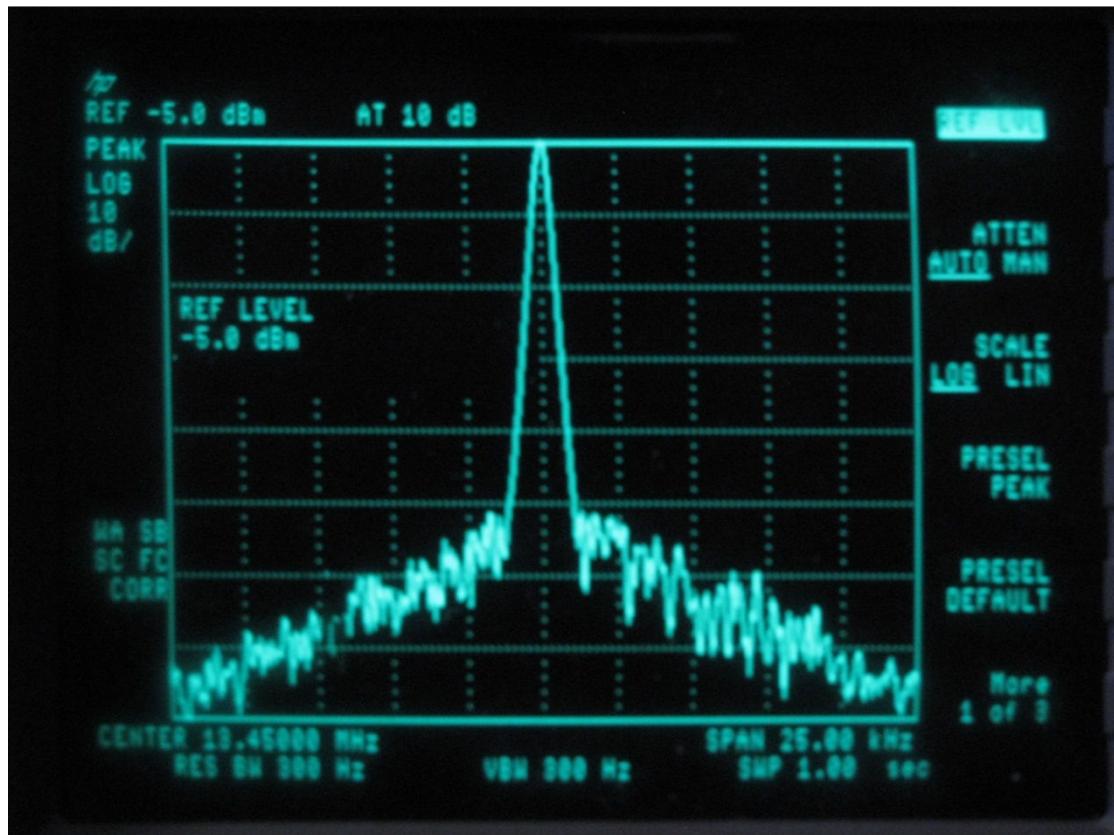


Figure 30 - FPGA Transmit Spectrum without FSK Modulation

The instrument's resolution bandwidth was 300 Hz. The SNR at ± 12.5 kHz in this bandwidth is displayed at 75 dB. This limit is attributed to phase noise in the spectrum analyser. When referred to a bandwidth of 10 kHz (as per the transmitter specification) the measurement limitation with this instrument is ~ 60 dB.

The measured output power was -5 dBm, slightly lower than predicted. This is possibly due to small losses in the DAC balun, The DAC sinc(x) response and possibly a higher DAC bias resistance than optimum.

The FSK modulation was then passed to the FPGA transmitter. This was programmed via Push Button PB0 on the FPGA PCB. Figure 31 displays the transmitted spectrum when modulated.



Figure 31 - Transmitted Spectrum When Modulation Is Applied

As expected, familiar Bessel sidebands appear spaced at the modulation rate ($1 \text{ kHz} = 2 \text{ kB/s}$).

The transmit data was well filtered and resembles a sinusoid. The spectral energy at $\pm 12.5 \text{ kHz}$ is comparable to the unmodulated case suggesting that spectrum analyser phase noise limits this measurement (note that DAC noise is flat).

A Mathcad simulation was then run to compare the ideal spectra with measured data.

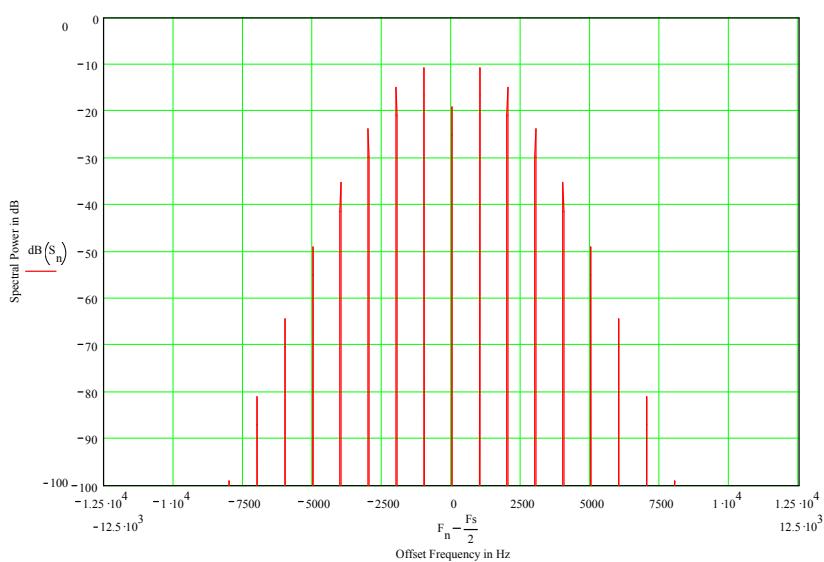


Figure 32 - Mathcad Simulation for Tx Spectrum with Sinusoidal Modulation

The Mathcad spectrum closely matches the measured transmit spectrum but is not impaired by phase noise (spectral lines are discrete) at higher frequency offsets. The Bessel sidebands appear 90 dB below in-band sidebands for an offset frequency exceeding ± 7.5 kHz in both cases. This offset represents the measurement limit for BW = 10 kHz measured at offset frequency = ± 12.5 kHz for the spectrum analyser in use. However the primary comparisons are equivalence in relative Bessel sideband power and frequency spacing for each term.

In summary, actual transmit spectral energy limits are inconclusive due to limitations in measurement equipment. Further the University licence has a vague specification limit. Communication with NZ Radio Spectrum Management (RSM) has not yet resulted in a definitive limit. However, if RSM accepts a proposal to use ± 12.5 kHz as the adjacent channel for spurious measurements, then the theoretical limit for spurious output power, predicted from simulation is 90 dB. This will not be realizable in practice as real data is not sinusoidal, nor are DAC's and FPGA processing numerically exact.

Conversely, if the frequency offset for measurement is interpreted as ± 3 kHz using a measurement bandwidth of BW = 10 kHz then no practical transmission is possible.

Despite such difficulties, the FPGA transmitter appears to work well and may exceed the ability of current equipment to fully assess its performance.

9.3.2 FPGA Based Radio Development and Measurement Location

The Figure 37 shows the equipment used in developing this FPGA based radio



Figure 33 - FPGA Based Radio Measurement Area and Equipment

- Upper left: Rhode and Schwartz SMBV100A signal generator used as interferer B
- Upper middle: Agilent 8446B signal generator used as wanted source A
- Construction on top: Minicircuits RF modules used in a previous project
- Construction, right: FPGA PCB and MSI PCB
- Upper right: Author's notebook used for ISE Webpack, analysis and documentation
- Lower left: HP 8596E Spectrum Analyser

10 Conclusions and Future Development Suggestions

This thesis has presented a generic FPGA based radio architecture based on direct oversampling of RF signals. This demonstration platform integrates RF, mixed signal, FPGA and Verilog programming language technologies. All these interact to some extent. Despite this, the approach taken differentiates each into relatively standalone functions.

The operating frequency is programmable from less to 1 MHz up to Nyquist (25 MHz based on the FPGA PCB 50 MHz TCXO). The design uses direct digital synthesis (DDS) to enable this flexibility. Extremely fine frequency resolution is possible, allowing direct DDS modulation with narrow band FSK or GMSK modulation.

Standard engineering principles are employed in the FPGA radio architecture. Although the demonstration platform is aimed at HF applications, the architecture is not limited to 30 MHz. MSD devices are available with sample rate capability extending beyond 2 GHz. The FPGA based radio has an IQ interface to the external world. Consequently, even Direct Conversion architectures are compatible with the architecture presented here.

This thesis has shown how FPGA radios can be designed and constructed. Physical measurement revealed the limitation of current laboratory equipment. Future research into FPGA based radio technologies would benefit from low phase noise vector spectral analysis equipment and Bit Error Rate analysis tools.

Although the demonstration platform adopts FSK as its preliminary modulation format, other formats are not excluded. Alternative formats require a different software module. Other modules would conceivably remain unchanged. The FPGA transmitter has adopted an optimal architecture aimed specifically for FSK. However the receiver architecture, operated in reverse, will implement a generic transmitter.

Practical application for over the air communication is however hampered by unclear specifications associated with the University's licence. Compliance with regulatory requirements lacks certainty given this deficiency.

Even so, compliance with conventional land mobile narrow band FM approvals appears quite feasible. The MSI adopts low resolution, cost effective MSDs. For example, the 8 bit ADC can readily be upgraded to 10, 12, 14 or 16 bit resolution based on commercially available device technologies.

The FPGA based radio project could be extended to long range world-wide communication using 13.45 MHz. The current licence allows transmit power levels up to 100 Watts. An authenticated and detailed spectral mask is necessary however before direct antenna connections could be guaranteed as being legal.

Alternatively, the FPGA receiver could be used for GPS applications. Several RFIQ devices are available to produce IQ signals directly for the FPGA receiver architecture.

12 Papers

12.1 PTDF Document Presented at ENZCon 2010

Parameter Tuned Digital Filter

Ian R. Scott, T. C. A Molteno

University Of Otago Physics Department, University Of Otago

Dunedin, P.O. Box 56

New Zealand

scoia295@student.otago.ac.nz, tim@physics.otago.ac.nz

Abstract:

Digital filters are widely used in many signal-processing systems. They are divided into FIR and IIR categories. Each category then has multiple implementations. Although referred to as “digitally accurate” these filters remain inflexible. They typically require a large number of coefficients that must be predetermined prior to application. This limits their use in software-defined radios (SDR) that must process many diverse modulation formats simultaneously. Each new modulation format therefore demands human intervention. Even so, determining a common assortment of integer related clock frequencies and filter architectures may not be feasible. We propose an alternate Parameter Tuned Digital Filter or PTDF using integer arithmetic. We show that this filter can provide a flexible alternative to traditional FIR and IIR designs. Few parameters are required to define PTDF’s and determining these parameters is straightforward and intuitive. Once determined, the PTDF’s frequency and time domain behaviours allow simple parameter scaling. These favourable characteristics suggest its inclusion in highly flexible, software defined radios.

Keywords:

SDR, “Digital Filter”, “Integer Arithmetic”, “Nested Loop”, Biquad, Butterworth, Chebychev, Decimation, Interpolation,

1 INTRODUCTION

Previous analogue filters suffered from performance variation caused by component tolerances. In contrast, the digital filter is as stable as its clock sampling frequency F_s . Further they achieve high ratios of stop to pass band rejection that would be impractical for analogue filters. However digital filters have two weaknesses; large numbers of coefficients have to be predetermined by various mathematical methods to implement a target frequency or time response. New coefficients have to be recalculated when requirements change. These new coefficients do not resemble the previous set. Simple scaling approaches are ineffective and human intervention is needed. Further, the fractional bandwidth Φ for a cut frequency F_c defined as $\Phi \equiv 2 \pi^{F_c} / F_s$ has a limited range of feasible values e.g. $\pi/5 < \Phi < \pi$. Low Φ forces excessive filter length. Therefore many coefficients and digital multiplies are needed. This results in excessive use of FPGA resources.

This inflexibility seems counter-intuitive. Many people would expect that a digital filter should be software configurable. Whilst true, human intervention is needed for modifications. Even so, multiple requirements often conflict in applications. For example, a filter designed for wide band service becomes excessively large in narrow band applications. Simply recalculating coefficients is inadequate. Proposing a bank of application specific filters is also impractical – the excessive size of a narrow bandwidth filters may exceed the FPGA resources designed for wide bandwidth applications. Increasing FPGA size might suggest a solution but then additional cost would be incurred. Even if these costs are accommodated, specific clock frequencies are required for each data rate. These may not be integer related or have an unattractively low common factor.

The SDR requires flexibility as a fundamental requirement. Further the SDR is required to process conflicting modulation standards in parallel. Each standard demands specific frequency and time response characteristics. There is no guarantee that a common clock frequency can be found. Even if phase locked loops are used these still require a common integer related reference frequency. Again, even if a common frequency is found, specialist design and potentially expensive hardware is needed for suitable low phase

noise clocks. The generic SDR remains seeking suitable enabling technologies.

The motivation of this document is to propose an alternate Parameter Tuned Digital Filter, or PTDF architecture. The PTDF avoids many FIR and IIR weaknesses. The PTDF operates equally well at low and high values of Φ . Successful operation has been obtained with values of Φ less than 0.001. Further, Φ is linearly scaleable by a single global parameter that we will just call k . In fact,

$$\Phi' = k \cdot \Phi \quad (1)$$

This inherent flexibility simplifies the design of a SDR that could, in principle, process multiple simultaneous modulation formats. In contrast to FIR and IIR filters the PTDF complexity is independent of Φ .

The PTDF is inherently tuneable by k . Its frequency domain transfer function is also continuously tuneable with extra parameters k_1, k_2, \dots, k_N . The number of parameters equals the filter order N . The PTDF emulates conventional analogue filters of the same order N . It has an IIR response and closely emulates Butterworth or Chebychev filter types. Additionally, similar design methods can be employed to design the PTDF based on analogue filter design methods

The PTDF requires little FPGA resource compared to its FIR and IIR counterparts. For example, a $N=4$ Chebychev PTDF only uses four multipliers and parameters. A corresponding IIR requires far more FPGA resource especially at low values of Φ . Even if these extra resources are accommodated, the IIR remains inflexible whilst the PTDF would not. Further, the PTDF can use low precision fixed-point arithmetic. Not all FIR or IIR are well suited to fixed-point arithmetic. We show stable operation on a Spartan 3 FPGA platform programmed with 8-bit resolution.

1.1 First Order PTDF Topology

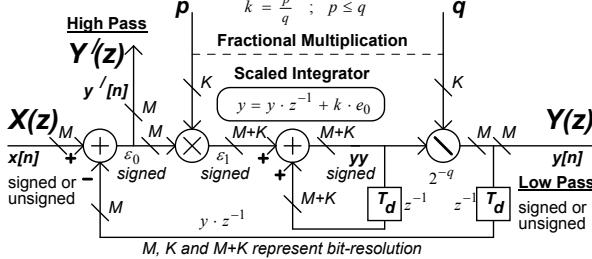
The first order PTDF is simplest. It requires an integrator, integer input scaling, power of two-output division and feedback. Division by two corresponds to an array shift operation and uses minimal FPGA resource.

The PTDF is tuned by parameter p with fixed $q = 2^Q$ where Q is an integer 1, 2, ...N. The parameter k is defined as $k \equiv p/q$. The lowest

value for Φ occurs when $p = 1$ but q can be made arbitrarily large. Therefore Φ can be made arbitrarily small without increasing filter length.

The first order PTDF can be implemented using the following architecture

1st Order Parameter Tuned Digital Filter (PTDF)



This PTDF produces simultaneous high and low pass outputs. It implements fractional multiplication despite using fixed-point arithmetic. We call this fractional multiplying integer a FMI.

1.1.1 First Order PTDF Analysis In The z -Domain

Z-domain analysis reveals

$$LPF(z) \equiv \frac{y}{x} = \frac{k}{1 + (k-1) \cdot z^{-1}} = \frac{k \cdot z}{z + (k-1)} \quad (2)$$

The corresponding high pass response is

$$HPF(z) \equiv \frac{\varepsilon_0}{x} = \frac{z - 1}{z + (k-1)} \quad (3)$$

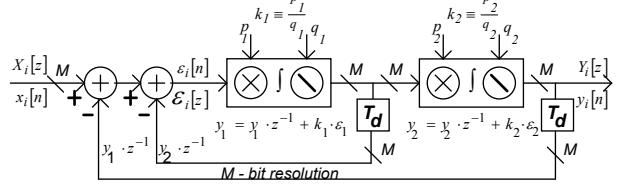
The conventional definition for z is.

$$z \equiv e^{j \cdot 2 \cdot \pi \cdot \frac{f}{f_s}} \quad (4)$$

1.2 Second Order PTDF Topology

Adding an additional integrator and extra feedback path creates a second order PTDF. It therefore has a nested loop feedback structure. The following diagram shows two FIM's. Bit resolution is shown as M .

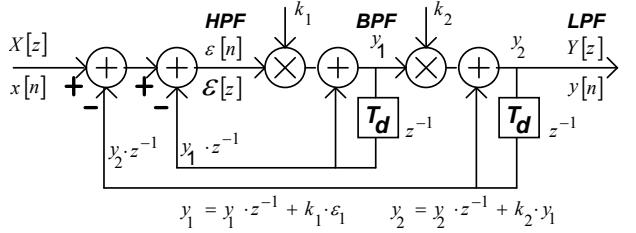
Second Order PTDF Using Two FIMs With Dual Feedback



1.2.1 Z-Domain Analysis For The 2nd Order PTDF

The second order PTDF provides an additional band-pass output where the two integrators are joined.

Simplified Second Order PTDF For Analysis



The z-domain equations used for analysis are

$$\begin{aligned} \varepsilon &= x - y_2 \cdot z^{-1} - y_1 \cdot z^{-1} \\ y_1 &= y_1 \cdot z^{-1} + k_1 \cdot \varepsilon \\ y_2 &= y_2 \cdot z^{-1} + k_2 \cdot y_1 \end{aligned} \quad (5)$$

We can derive the low pass z-domain transfer function from (4) by algebraically eliminating variables.

$$LPF_2(z) \equiv \frac{y_2}{x} = \frac{k_1 \cdot k_2}{1 + (k_1 \cdot k_2 + k_1 - 2) \cdot z^{-1} + (1 - k_1) \cdot z^{-2}} \quad \dots(6)$$

$$BPF_2(z) \equiv \frac{y_1}{x} = \frac{k_1 \cdot (1 - z^{-1})}{1 + (k_1 \cdot k_2 + k_1 - 2) \cdot z^{-1} + (1 - k_1) \cdot z^{-2}} \quad (7)$$

$$HPF_2(z) \equiv \frac{z}{x} = \frac{1 - 2 \cdot z^{-1} + z^{-2}}{1 + (k_1 \cdot k_2 + k_1 - 2) \cdot z^{-1} + (1 - k_1) \cdot z^{-2}} \quad (8)$$

1.3 Low Frequency Analysis

It is convenient to transform equations in z to frequency f especially when Φ is small. Consider the following definition for normalised frequency Ω .

$$\Omega \equiv 2 \cdot \pi \cdot \frac{f}{F_s} \quad (9)$$

The complex exponential in z can be expanded using a Taylor series and approximated for small Ω .

$$z \equiv e^{-j \cdot \Omega} \cong 1 - j \cdot \Omega \quad (10)$$

Substituting (10) into (6) results in

$$LPF_2(\Omega) \cong \frac{1}{1 + j \cdot \Omega \cdot \left(\frac{1 - k_2}{k_2} \right) - \Omega^2 \cdot \left(\frac{1 - k_1}{k_1 \cdot k_2} \right)} \quad (11)$$

Equation (11) represents a second order frequency response approximation based on Ω . The same approach can equally be applied to band and high pass transfer equations (7) and (8).

1.5 Higher Order PTDF.

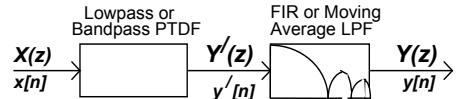
High orders PTDF are formed by cascaded lower order PTDF's. This approach is identical to conventional high order analogue filter design. Although a single high order PTDF could be formed using multiple feedback paths and FMIs it is simpler to multiply lower order PTDF's. The composite z transform becomes

$$H_N(z) = \prod_{n=0}^N H_n(z) \quad (12)$$

1.6 Periodic Frequency Response Removal

The PTDF has a repeating frequency response. For example, characteristics at Ω and $(2 \pi F_s) - \Omega$ are equivalent. In some cases a simple moving average filter can be combined with the PTDF to remove this response.

Removing The PTDF Periodic Frequency Response



This "roofing filter" is required to remove energy above $(2 \pi F_s) - \Omega$ so will not require adjustment.

1.7 Second Order Frequency Response Simulation

1.7.1 Simultaneous LPF, BPF and HPF Outputs

We used Mathcad to predict the frequency response behaviour for a second order PTDF corresponding to equations (6), (7) and (8). The simulation was implemented in time. The Mathcad algorithm, shown below, is simple.

```

PTDF_2(k1, k2, x) := | Samples<- rows(x)
                        yy<- 0
                        yout<- 0
                        for n < 0.. Samples - 1
                            εε <- x_n - yout
                            ε <- x_n - yy - yout
                            yy <- yy + k1·ε
                            yout <- yout + k2·yy
                            y_n,0 <- yout
                            y_n,1 <- yy
                            y_n,2 <- ε
                        y

```

A time domain impulse "x" was used as a stimulus. Three outputs corresponding to low pass, band pass and high pass transfer functions are shown. The final variable y is an $N * 3$ matrix where N represents the number of time domain samples. Parameters $k1$ and $k2$ control the PTDF characteristics.

Matrix y contains individual transient responses generated by this second order

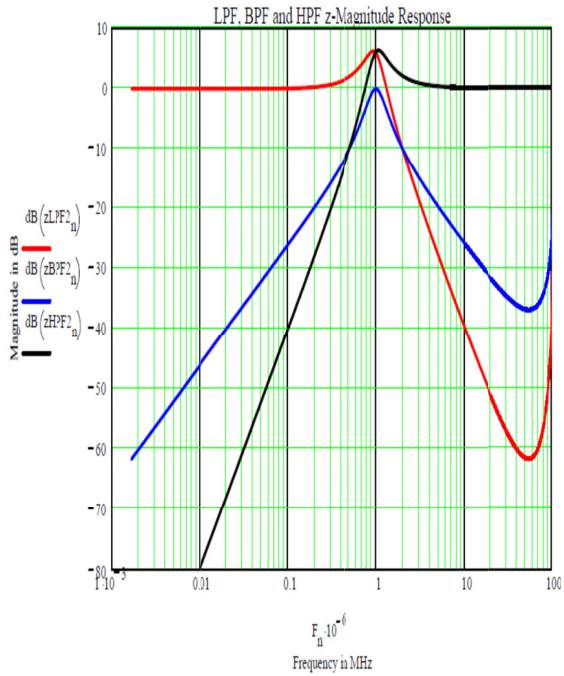
PTDF. These are applied to a FFT function, creating three frequency domain responses. The results obtained were identical to direct z-domain simulation, shown below in Mathcad syntax.

```

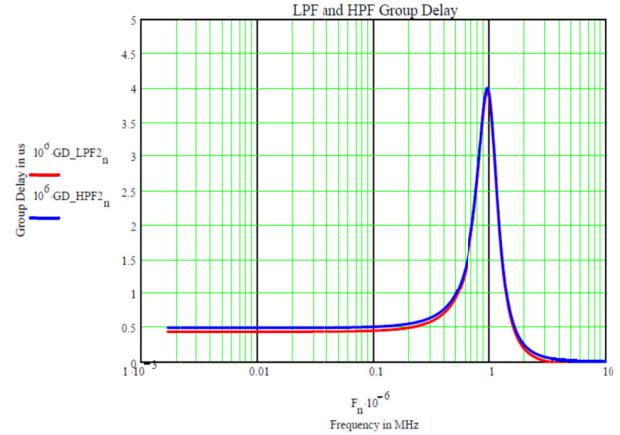
z_PTDF_2Fs,k1,k2,F := | N←rows(F) - 1
                         for n ← 0..N
                           | F_n
                           | 1j·2·π · -----
                           | z← e
                           | D← 1 + (k1·k2 + k1 - 2)·z-1 + (1 - k1)·z-2
                           | PTDFn,0← k1·k2
                           | -----
                           | PTDFn,1← k1·(1 - z-1)
                           | -----
                           | PTDFn,2← 1 - 2·z-1 + z-2
                           |
                           PTDF

```

The following graph shows three frequency domain amplitude responses in dB. We found that both approaches gave identical responses so only one version is shown.



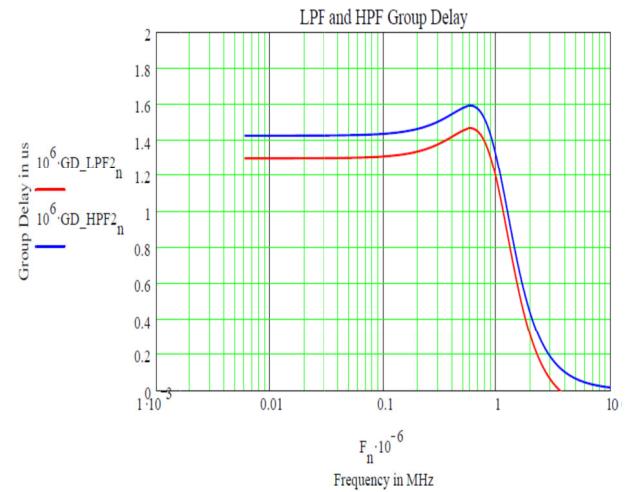
The next graph shows differential group delay for high filter “Q” values. This represents a Chebychev response with high pass band ripple.



The previous graphs used the following parameters

$$Q := 2 \quad k := 2 \cdot \pi \cdot \frac{F_c}{F_s} \quad k1 := \frac{k}{Q} \quad k2 := k \cdot Q$$

Reducing the PTDF Q to 0.7 improves group delay significantly and converts a Chebychev response to a Butterworth version.



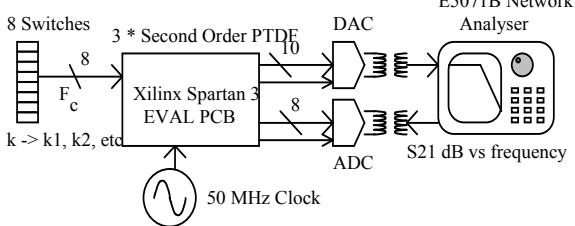
The familiar concern over IIR group delay is perhaps ill conceived based on this second graph using $Q = 0.7$. Low and High pass outputs are shown to be very similar.

2 DISCUSSION

First and Second order PTDF's have been analysed and simulated in Mathcad. We used Verilog to implement both in a Xilinx Spartan 3 Starter Kit. Preliminary tests have been conducted at 50 MHz sample rate with 8 bit DC input test vectors. A dual hexadecimal

alphanumeric display was used to indicate 8 bit output values. We plan to update the test configuration with an 8-bit ADC, 10-bit DAC and ancillary components. This is planned for the next few weeks. However the actual PTDF appears to remain well behaved with this test configuration.

RF Test Configuration For The Sixth Order PTDF



S21 Plot Using Agilent E5071B Network Analyser



Preliminary results are encouraging. The plot above shows a 6th order PTDF frequency response when implemented on the Spartan 3 FPGA. This PTDF used three-cascaded 2nd order PTDF's with a common frequency scaling factor k . An 8-bit 50 MHz ADC was used for signal acquisition and a 10-bit DAC was used to output the filtered result.

The PTDF is motivated by the need for complete parameter defined characteristics needed in a SDR. However it is also well suited for audio applications such as Automatic Level Control (ALC). Such systems require “fast attack-slow decay” gain behaviour. The parameter-controlled bandwidth of the PTDF can be switched between increasing and decreasing audio levels without introducing

transients. When the audio level increases, a high k is used. When the audio level falls k takes on a smaller value. This ALC application is less well suited to FIR and IIR filter types with pre-determined coefficients. Simply switching between two standard filters is likely to introduce ALC transients and show up as audio clicks

Digital systems typically require exact integer ratios between internal sampling frequencies. To illustrate consider a SDR operating with two system bandwidths. The use of fixed cascaded decimator and interpolation filters places high demands on the design engineers. They need to find suitable common multiples of clock frequencies that allow simultaneous signal processing across both channel bandwidths. If a third channel bandwidth is added then finding a 3-way solution becomes exponentially difficult both for the engineer and potentially the digital processing hardware. Given that a generic SDR should not be limited to three simultaneous formats, the conventional approach may be unattractive.

This inflexibility represents the Achilles heel of current digital radio systems intended for SDR. One outcome from this research project is to find digital processing topologies that avoid inflexibility without adding excessive complexity. The PTDF has uncomplicated implementation compared to FIR or IIR approaches. It uses far less FPGA resource. It provides potentially high decimation or interpolation ratios in a single filter. This is because Φ can be arbitrarily small without increasing PTDF size and corresponding FPGA resource. In contrast conventional decimation filters are usually limited to ratios of 2 to 5. To illustrate, consider an ADC sampling at 50 MHz. Also assume that a final demodulation bandwidth need only require a sample rate of 100 kHz. The decimation ratio is therefore 500:1. In principle a single PTDF could be used. The alternative approach might require decimation ratios of 5, 5, 5 and 4 to achieve this overall decimation ratio. Obviously four FIR filters would be needed.

We observe further difficulties that this approach imposes. When we consider a generic SDR, each one of these decimation

filters requires a common clock reference frequency. If finding a common clock reference for a single filter is difficult, extending the requirement to 4 filters may make this task almost impossible.

We do not propose the PTDF as a single solution to a generic SDR. We do propose it as one potential solution of many that require invention and development. Nor do we claim that the PTDF is free of unwelcome characteristics not yet experienced. Further research is needed to assess its complete characteristics. However, it must be said, that any filter has its pro's and con's. The most we can hope for is that the pros outweigh the cons or that a suitable compromise is available.

3 FURTHER WORK

The PTDF is intended for Software Defined Radio (SDR) applications. The author's project is intended to show "proof in principle" for a smaller scale SDR. This project requires both technology innovation and application in a FPGA, combined with suitable RF processing circuits and systems. It is not simply a software programming exercise executed in a FPGA. Some imagination and creativity is required for a successful SDR outcome. Further, divergent disciplines are needed to finish a working prototype SDR for *demonstration*.

A prototype test system is planned for operation at 13.45 MHz but with parameter controlled frequency, bandwidth and data rate. This architecture will be scalable for much higher frequency operation. In this view the current investigation is intended to provide "proof in principle" for the PTDF filter architecture and other software structures with high levels of flexibility that elude conventional approaches.

Other software structures also need refinement for this example SDR. For example, the standard Numerically Controlled Oscillator (NCO) requires a phase accumulator, relatively expensive sine-cosine Look Up Table (LUT). These NCOs often need ancillary enhancement based on dither noise injection and polynomial correction. They can achieve exceptional performance but use significant FPGA resource. Recent investigations reveal that a potentially simpler architecture is feasible by changing the way the NCO is mathematically

implemented. Outcomes may offer further benefit for SDRs.

The PTDF however requires a great deal of further analysis, measurement and refinement. We will investigate optimum bit resolutions based on SNR requirements. Although limit cycles have not yet been observed, critical assessment is required. The integration stages employed by the PTDF may offer great benefit however by introducing fundamental smoothing functions.

Further, the use of cascaded first and second order PTDF requires comparison and evaluation with a single high order PTDF. We will investigate the feasibility of a PTDF with orders much higher than 2. Our interest is in both the comparative performance as well as potential reduction in FPGA resource demand.

The tuneable nature of the PTDF may also offer significant benefit in systems that require ad-hoc adaptability. For example, consider a receiver experiencing a strong interferer close to its reception channel. The PTDF could simply reduce its channel bandwidth to exclude this interferer. Although this would usually increase BER on a "clear" channel, the actual BER on an interfered channel could significantly improve given this strategy. Equally the PTDF could shift its centre frequency to the opposite side of the interferer.

4 CONCLUSIONS

The results to date have been encouraging. Although suggesting potential benefits in flexibility the PTDF may not be an essential component for a SDR, it does appear to overcome many of the weaknesses associated with conventional approaches. Its inherent flexibility and low FPGA resource overhead is attractive. Operation at low Φ may also simplify decimation and interpolation requirements. Its real world introduction in SDR systems requires strategies for optimal placement and design. Further, these *generic* SDRs need to progress beyond "proof in principle" demonstrators to real world products used commercially.

If further analysis and testing remains favourable, the PTDF may well provide a useful addition to traditional FIR and IIR digital filter technologies.

5 REFERENCES

Modelling, Synthesis and Rapid Prototyping with the Verilog HDL – Michael D. Cilletti, University of Colorado, Colorado Springs,

Digitally Removing A DC Offset: DSP Without Mathematics, Ken Chapman – Xilinx, WP279 (v1.0) July 18, 2008

ELEC 358: Electronic Design Techniques, Tim Molteno, Physics Department, University Of Otago, Dunedin, September 3, 2008

http://en.wikipedia.org/wiki/Software-defined_radio

http://www.wirelessinnovation.org/page/Defining_CR_and_DSA

Acronyms

FPGA – Field Programmable Gate Array

SDR – Software Defined Radio

FMI – Fractional Multiplication Integrator

DGD – Differential Group Delay

FIR – Finite impulse Response (filter)

IIR – Infinite Impulse response (filter)

PTDF – Parameter Tuned Digital Filter

Ω - Filter Fractional Bandwidth (f_c/f_s)

Parameter Tuned Digital Filter (PTDF)

Ian R. Scott, Tim C. A. Moltenco

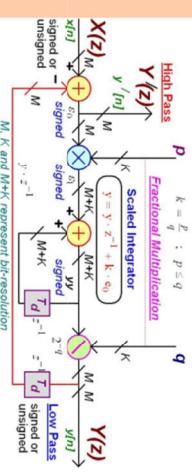
University Of Otago Physics Department

Dunedin, PO Box 56, New Zealand, scott295@student.otago.ac.nz, tim@physics.otago.ac.nz

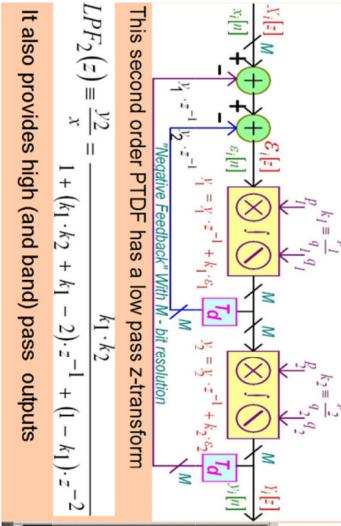
Introduction to the PTDF

Digital filters are increasingly used in many signal processing systems. However they are often fixed wrt frequency. In contrast the PTDF is continuously tunable over a wide range. It has an IIR response similar to Butterworth and Chebychev analog filters

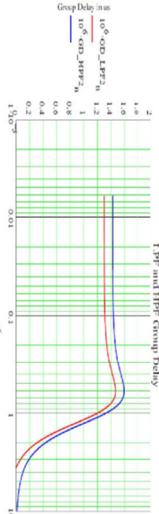
1st Order Parameter Tuned Digital Filter (PTDF)



Second Order PTDF Using Two FIRs With Dual Feedback



Although the PTDF is IIR it can exhibit low "group delay variation" by reducing its "amplitude ripple"



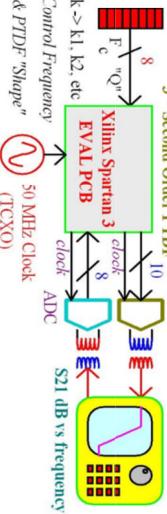
High order PTDF's are formed by cascading lower order sections, similar to analog approaches

$$H_N(z) = \prod_{n=0}^N H_n(z)$$

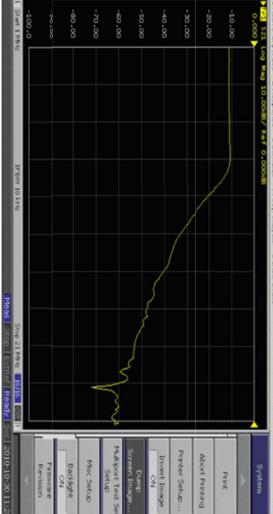
Simulation (e.g. on Mathcad) is also reasonably simple. The code is similar to Verilog.

```
PTDF_2(k1,k2,y) := Samples->rows(x)
yy<-0
yout<-0
for n <= 0..Samples - 1
  xx<-x_n - yout
  e<-x_n - yy - yout
  yy<-yy + k1*e
  yout<-yout + k2*yy
  y_n,0<-yout
  y_n,1<-yy
  y_n,2<-e
```

RF Test Configuration For The Sixth Order PTDF



Here is a 6-th order PTDF low pass response using fixed point arithmetic (sample rate = 50 MHz)



Some modern "cognitive" and "intelligent" radios demand highly flexible operation. Unlike traditional systems these seek to use unoccupied spectrum for communication. The PTDF may be suited to these applications due to its inherent flexibility. However it is likely that traditional CIC, FIR and IIR filters will still play an essential role in these novel structures.

Figure 34 - ENZCon 2010 PTDF Poster

12.3 US Patent Cover Page

US 20060230089A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2006/0230089 A1
 Scott et al. (43) Pub. Date: Oct. 12, 2006

(54) FREQUENCY ESTIMATION

(76) Inventors: Ian Russell Scott, Christchurch (NZ);
 Refik Shadich, Christchurch (NZ);
 William Mark Siddall, Christchurch
 (NZ)

Correspondence Address:
DANN, DORFMAN, HERRELL & SKILLMAN
1601 MARKET STREET
SUITE 2400
PHILADELPHIA, PA 19103-2307 (US)

(21) Appl. No.: 10/546,696

(22) PCT Filed: Feb. 24, 2004

(86) PCT No.: PCT/NZ04/00035

(30) Foreign Application Priority Data

Feb. 24, 2003 (NZ)..... 524369

Publication Classification

(51) Int. Cl.
G06F 1/02 (2006.01)

(52) U.S. Cl. 708/270

(57) ABSTRACT

The present invention relates to a method and hardware for estimating the frequency offset of a signal. The method includes obtaining samples of the signal at at least two instants in time, and utilising the samples in a mathematical equation relating estimated offset frequency to the samples, wherein the mathematical equation is derived based on the premise of a modulating signal with a complex frequency.

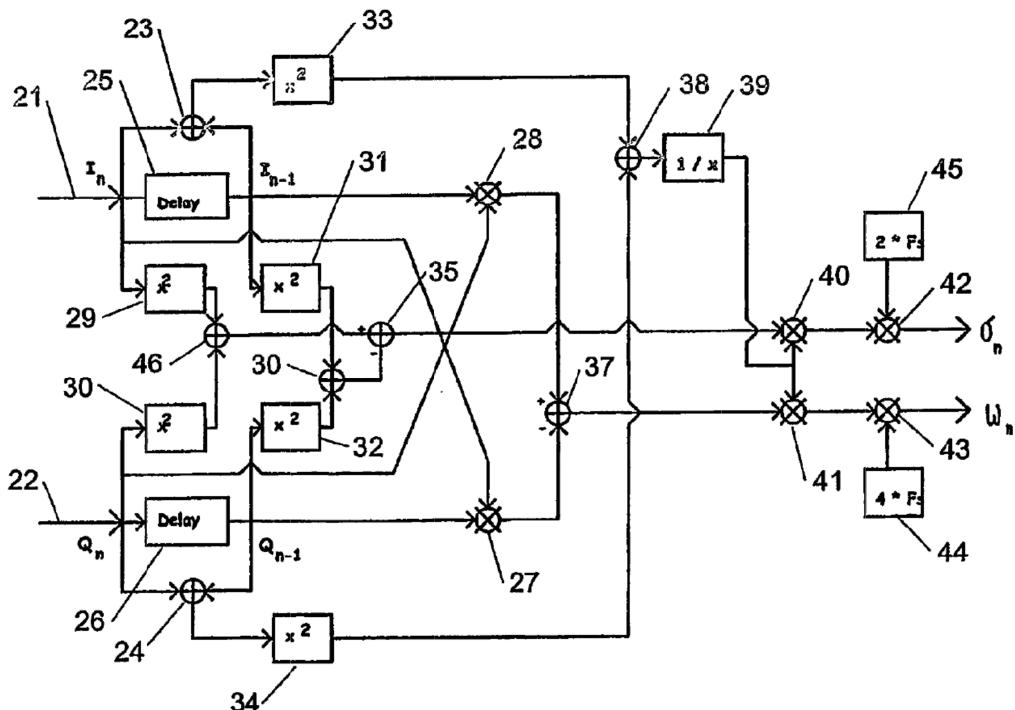


Figure 35 - One Version of FPGA Architecture

Note – A higher resolution version is available on the CD

13 Appendix – Detailed Mathematical Derivations

13.1 Instantaneous Complex Frequency Interpretation

The method of “instantaneous complex frequency estimation”[19] is implemented in this FPGA radio. A definition for a complex baseband signal $z(t)$ is,

$$z(t) \equiv I(t) + j Q(t)$$

Then

$$\log(z) = \log(|z|) + j\phi(z)$$

We differentiate both sides

$$\frac{\partial}{\partial t} \log(z) = \frac{\frac{\partial}{\partial t}|z|}{|z|} + j \frac{\partial}{\partial t} \phi(z)$$

From which it follows that

$$\frac{\frac{\partial}{\partial t}z}{z} = \frac{\frac{\partial}{\partial t}|z|}{|z|} + j \frac{\partial}{\partial t} \phi(z)$$

Now using the “dot” notation for differentiation clarifies the relationship

$$\frac{\overset{\circ}{z}}{z} = \frac{\overset{\circ}{r}}{r} + j\overset{\circ}{\omega} \text{ where } r \equiv |z| \text{ and } \overset{\circ}{\omega} \equiv \overset{\circ}{\phi}$$

We now have a simple frequency demodulation process.

$$\overset{\circ}{\omega} = \text{Im}\left\{\frac{\overset{\circ}{z}}{z}\right\}$$

Expressing frequency demodulation δf in Hz rather than radians reveals

$$\delta f = \frac{1}{2\pi} \cdot \text{Im}\left\{\frac{\overset{\circ}{z}}{z}\right\}$$

Convert this continuous time domain equation to discrete time-domain approximation,

$$\Delta f_n = \frac{1}{\pi \Delta t} \cdot \text{Im}\left\{\frac{z_n - z_{n-1}}{z_n + z_{n-1}}\right\} = \frac{F_s}{\pi} \text{Im}\left\{\frac{z_n - z_{n-1}}{z_n + z_{n-1}}\right\}$$

Now recall the starting definition for z and write

$$\Delta f_n = \frac{1}{\pi \cdot \Delta t} \text{Im}\left\{\frac{(I_n + jQ_n) - (I_{n-1} + jQ_{n-1})}{(I_n + jQ_n) + (I_{n-1} + jQ_{n-1})}\right\}$$

Although the real component of instantaneous complex frequency has its uses, this FPGA radio will focus on the imaginary component representing frequency modulation. We therefore write

$$\Delta f_n = \frac{1}{\pi \Delta t} \operatorname{Im} \left\{ \frac{(\Delta I \sum Q + \Delta Q \sum I) + j(\Delta Q \sum I - \Delta I \sum Q)}{(\sum I)^2 + (\sum Q)^2} \right\}$$

with $\sum I \equiv I_n + I_{n-1}$ and $\sum Q \equiv Q_n + Q_{n-1}$

and $\Delta I \equiv I_n - I_{n-1}$ and $\Delta Q \equiv Q_n - Q_{n-1}$

It is now clear that

$$\Delta f_n = \frac{F_s}{\pi} \frac{\Delta Q \sum I - \Delta I \sum Q}{(\sum I)^2 + (\sum Q)^2} \text{ where } F_s \equiv \frac{1}{\Delta t}$$

This final result is elegant and compact. It is interesting to compare the approach with the traditional method. This requires the following steps.

- Estimate phase from $\arctan(Q / I)$
- Resolve quadrant ambiguity as \arctan is not defined over 0° to 360°
- Phase wrap the estimate to avoid the discontinuity crossing $0^\circ \pm 360^\circ$
- Differentiate the resolved phase-wrapped estimate to produce Δf

This method could be used in the SDR but it does seem clumsy. Further it requires two unnecessary decisions for quadrant ambiguity and axis crossing discontinuity. Both these decisions can introduce error and therefore SNR impairment.

13.2 Parameter Tuned Digital Filter (PTDF)

13.2.1 PTDF Features

This FPGA radio uses a “Parameter Tuned Digital Filter” or PTDF as described in a paper submitted to ENZCon 2010[18],[24]. This structure offers the following features

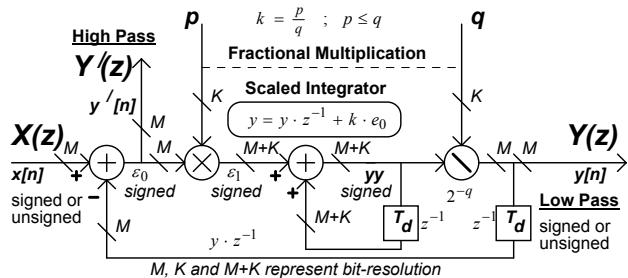
- It is completely tuneable with 1, 2, 3, or 4 parameters etc.
- The parameters are directly and intuitively related to the response
- Synthesising the parameters is simple – closed form solutions are possible
- The PTDF is can be cascaded to form higher order responses
- The PTDF uses only one multiplier per filter “order”. N=4 uses 4 multipliers
- The PTDF can operate with extremely low fractional bandwidth Ω without penalty. Its size is unchanged regardless of the value of Ω

The topology may be novel as far as the author can determine.

13.2.2 First Order Parameter Tuned Digital Filter

The simplest PTDF has order 1 and approximates a RC low-pass and high-pass filter.

1st Order Parameter Tuned Digital Filter (PTDF)



This PTDF consists of a discrete time integrator placed inside a negative feedback loop. The -3dB pass-band frequency is directly proportional to the numerical scaling term $k = p / q$.

The Spartan-3 PCB houses a FPGA suited for fixed-point (integer arithmetic) computation. This is potentially problematic as $0 < k < 1$. Higher values of k cause instability. Negative values also cause instability (the feedback is then positive). The solution is to split k into two components, p and q . The discrete time domain integration is placed in-between these scaling terms. This provides fractional multiplication. It makes sense to call this a Fractional Integration Multiplier (FIM).

The value of q is fixed at some large value that is a multiple of 2, e.g. 4096. The value of p can now range from 0 to 4096. The total value of $k = p / q$ now ranges from 0 to 1 ensuring stability.

Since q is a fixed multiple of 2, the division corresponds to a simple shift operation. This is computationally inexpensive. In practice, Verilog accepts the division directly without complaint.

Also worthy of note is bit resolution. Since the error term is multiplied by p the internal bit resolution must increase. If the input has 8 bits, the error term also has 8 bits. If p has 12 bits, the internal bit size is 20. This is not problematic for a FPGA as bit size is programmable.

Drawing the complete structure can be unwieldy on the page. It makes sense to represent the FIM as a symbol.

Placement Of The FIM In The First Order PTDF

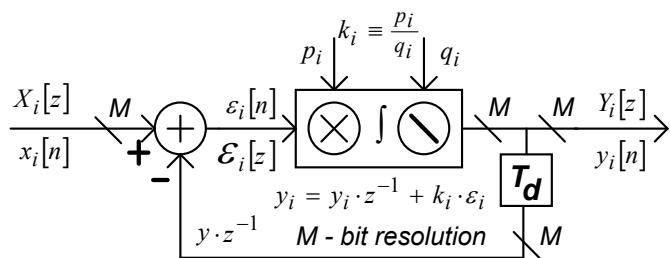
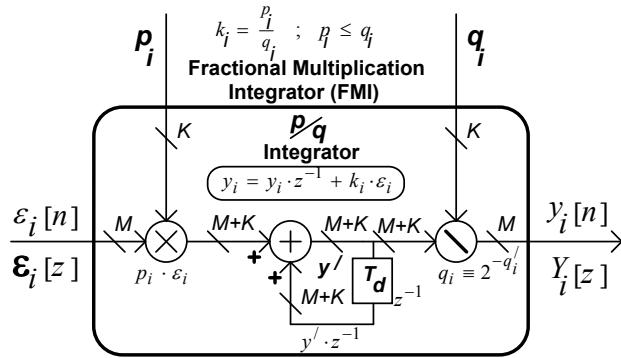


Figure 36 – Position of The FIM In First Order PTDF

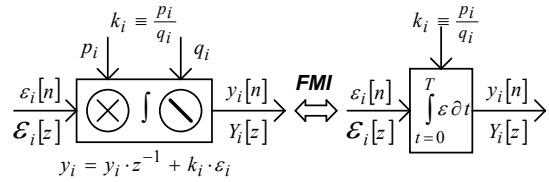
The FIM symbol then defines the following internal structure

Fractional Multiplication Integrator Architecture



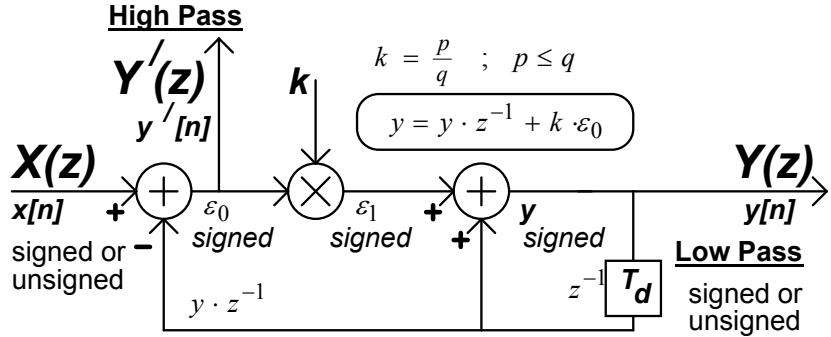
A more compact symbol can be used to simplify the diagram,

Fractional Multiplication Integrator (FMI) Symbols



As long as it is understood that the FIM is inserted for fixed point arithmetic, or neglected for floating point, the topology can be redrawn explicit symbols,

Simplified 1st Order PTDF For Analysis



It is then straightforward to derive the z-transform response, and the associated frequency domain response. For example, the low-pass output has the following transfer function

$$H_z \equiv \frac{y}{z} = \frac{k}{1 + k - z^{-1}} \quad \text{where } z_f \equiv e^{j 2 \pi f \Delta t}$$

$$LPF(z) \equiv \frac{y}{x} = \frac{k}{1 + (k-1)z^{-1}} = \frac{k z}{z + (k-1)}$$

$$HPF(z) \equiv \frac{\varepsilon_0}{x} = \frac{z - 1}{z + (k-1)}$$

When k is small, the pass-band frequency is also small so we can approximate (Taylor series) the exponential in z to obtain a frequency response function

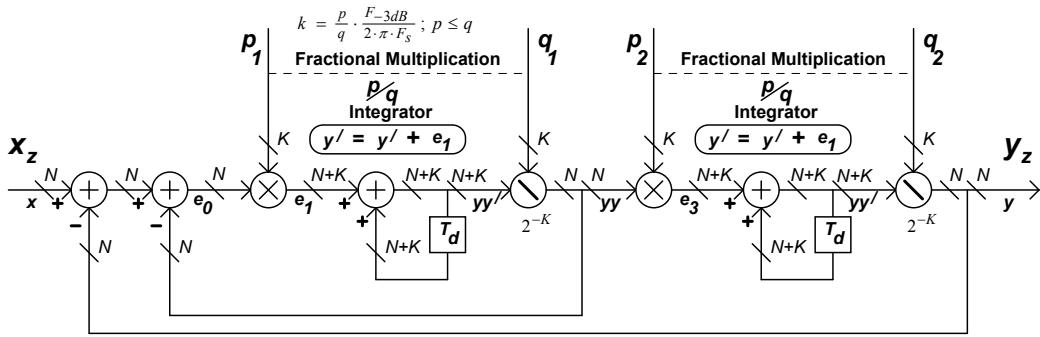
$$LPF(\omega) \approx \frac{1}{1 + j \frac{\omega}{\omega_c}} \text{ where } \omega \equiv 2\pi f \text{ and } \omega_c \equiv k F_s$$

This transfer function is identical to that of a RF low-pass filter.

13.2.3 Second Order Parameter Tuned Digital Filter

The first order PTDF demonstrates the general approach to synthesising these topologies. However it is desirable to have higher order structures. The basic “building block” structure is second order, e.g.

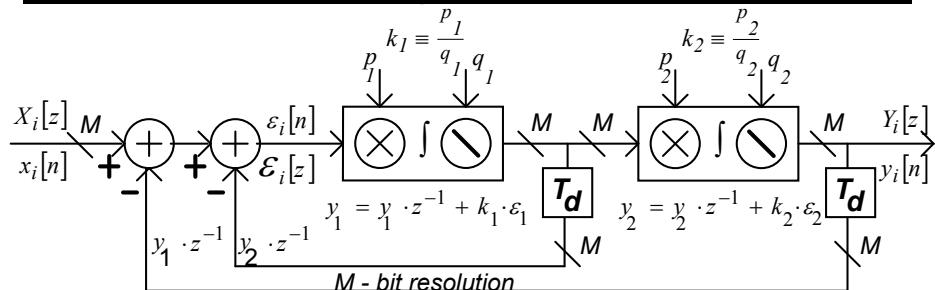
Second Order Parameter Tuned Digital Filter



This PTDF has two negative feedback paths. The inner loop represents the previous first order PTDF. A scaled discrete time integrator follows this. Negative feedback is then applied around the whole filter. The method of summation is of course arbitrary – the drawing shows two separate summations for clarity.

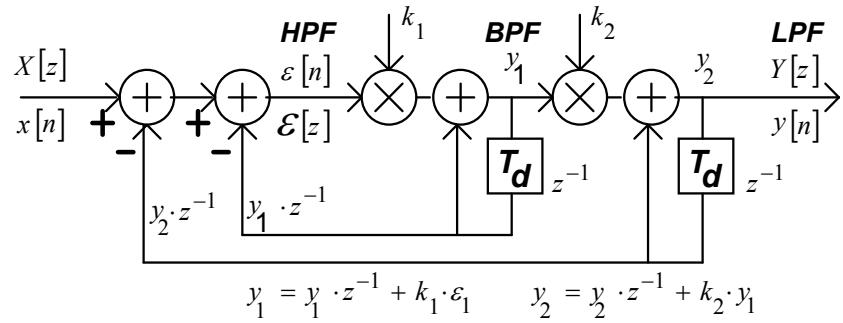
The second order PTDF contains 2 FIMs.

Second Order PTDF Using Two FIMs With Dual Feedback



Explicit representation for each FIM can be omitted. This simplifies analysis.

Simplified Second Order PTDF For Analysis



Having done so, deriving z-transforms for each of the three outputs is relatively straightforward.

$$LPF_2(z) \equiv \frac{y_2}{x} = \frac{k_1 k_2}{1 + (k_1 k_2 + k_1 - 2) \cdot z^{-1} + (1 - k_1) z^{-2}}$$

$$HPF_2(z) \equiv \frac{\varepsilon}{x} = \frac{1 - 2z^{-1} + z^{-2}}{1 + (k_1 k_2 + k_1 - 2) \cdot z^{-1} + (1 - k_1) z^{-2}}$$

$$BPF_2(z) \equiv \frac{y_1}{x} = \frac{k_1(1 - z^{-1})}{1 + (k_1 k_2 + k_1 - 2) z^{-1} + (1 - k_1) z^{-2}}$$

The same small frequency approximation can be made to derive a frequency domain transfer function. The low-pass output is used here as an example

$$LPF_2(\Omega) \equiv \frac{1}{1 + \frac{j}{k_2} \Omega - \frac{1}{k_1 k_2} \Omega^2}$$

It is also worthwhile to note the simplicity by which these PTDF's can be implemented. The following Mathcad demonstrates this for the second order example,

```
PTDF_2(k1, k2, x) := | Samples ← rows(x)
                      | yy ← 0
                      | yout ← 0
                      | for n ∈ 0.. Samples - 1
                      |   εε ← x_n - yout
                      |   ε ← x_n - yy - yout
                      |   yy ← yy + k1·ε
                      |   yout ← yout + k2·yy
                      |   y_{n, 0} ← yout
                      |   y_{n, 1} ← yy
                      |   y_{n, 2} ← ε
                      |
                      | y
```

Again no effort is made to simplify the script as obscuring concepts is best avoided at this stage. Note that three outputs occur simultaneously in the columns of y corresponding to low-pass, band-pass and high-pass functions. Equally, the direct z-transform functions can be used.

```

z_PTDF_2 (Fs, k1, k2, F) := | N←rows(F) - 1
                             | for n ∈ 0..N
                             |    $l_j \leftarrow \frac{F_n}{F_s}$ 
                             |    $z \leftarrow e^{j \cdot 2 \cdot \pi \cdot l_j}$ 
                             |   D ← 1 + (k1·k2 + k1 - 2)· $z^{-1}$  + (1 - k1)· $z^{-2}$ 
                             |   PTDFn,0 ←  $\frac{k1 \cdot k2}{D}$ 
                             |   PTDFn,1 ←  $\frac{k1 \cdot (1 - z^{-1})}{D}$ 
                             |   PTDFn,2 ←  $\frac{1 - 2 \cdot z^{-1} + z^{-2}}{D}$ 
                             |
                             | PTDF

```

The second order PTDF is a building block that can be used to create higher order filters. These are simply cascaded. The following example shows a cascaded of three second order PTDFs. The z domain functions were used to predict the red curve on Figure 37. The direct time domain form was excited with an impulse and converted to spectrum to reveal the blue curve. These are offset 5 dB to show their identical shape (otherwise they overlay directly).

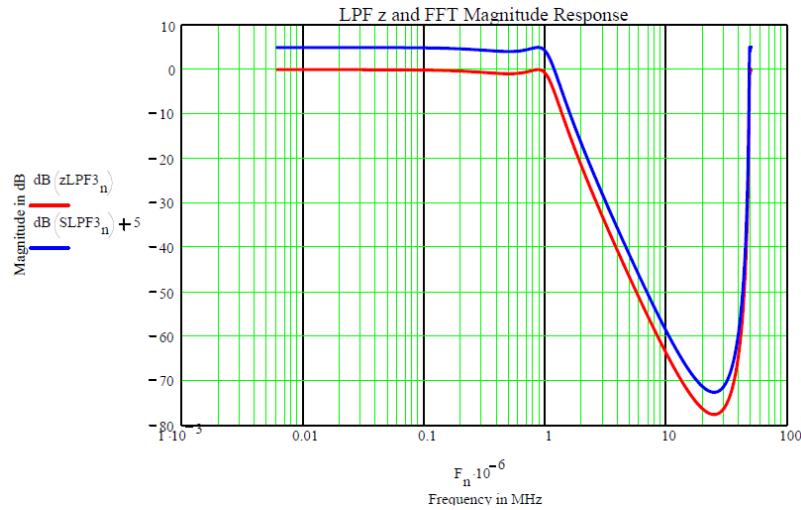


Figure 37 - PTDF Frequency Response in dB

The PTDF has a repeating frequency and needs to be combined with a moving average or FIR filter. Measurement on the receive channel filter suggests it should be before the PTDF. The next image shows a Mathcad plot of group delay with a restricted frequency axis.

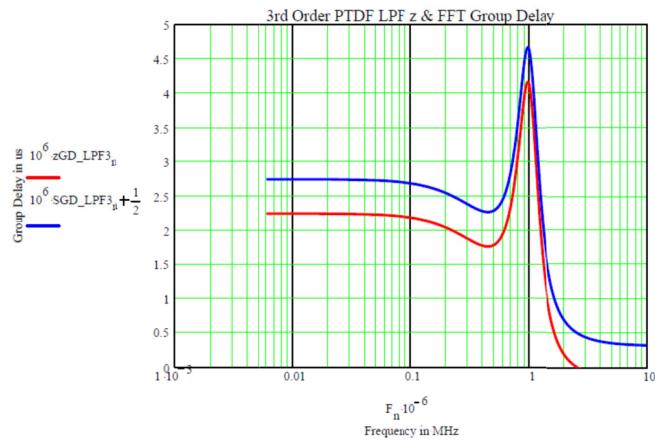


Figure 38 - PTDF Group Delay in us

To illustrate the performance in general, a sample rate of 100 MHz was selected with a pass-band of 1 MHz. The differential group delay is not excessive and can be reduced to almost zero by reducing the rate of stop-band attenuation.

14 Appendix B – Verilog Excerpts

14.1 Top Level Verilog Code

```
module Receiver(CLK_50MHz, PB, Sw, LED, Module, Segment,
    ADC, ADC_CLK, DAC_CLK, DAC, DATA_OUT);
    input CLK_50MHz;
    input [3:0] PB;
    input [7:0] Sw;
    output [7:0] LED;
    output [3:0] Module;
    output [7:0] Segment;
    input [7:0] ADC;
    output ADC_CLK;
    output DAC_CLK;
    output [9:0] DAC;
        output [3:0] DATA_OUT;
    assign ADC_CLK = CLK_50MHz;
    //assign DAC_CLK = CLK_50MHz;
    //
    wire CLK256;
    DivideN divN0(.clk(CLK_50MHz), .N(256), .Q(CLK256));
    DivideN divN1(.clk(CLK256), .N(4), .Q(CLK1024));
    /////////////////////////////////
    //assign Module = PB;
    //assign Segment = Sw;
    //assign DATA_OUT = PB;
    //assign LED = Sw;
    //assign DAC = 510; // + 0 * (ADC - 128); // was 4 *
    ///////////////////////////////
    wire signed [19:0] I, Q;
    RF8_IQ20 rfiq(.clk(CLK_50MHz), .RF(ADC), .I(I), .Q(Q));
    //
    wire signed [19:0] Ia, Ib, Qa, Qb;
    MAF16_20B maf20(.clk(CLK256), .x(I), .y(Ia)); // 4 stage MAF of 16 samples
    MAF16_20B maf21(.clk(CLK256), .x(Q), .y(Qa)); // 4 stage MAF of 16 samples
    //
    wire [11:0] p1, p2, p3, p4; // target BW = +/- 4 kHz
    assign p1 = 267; assign p2 = 1333; assign p3 = 1600; assign p4 = 4000;
    PTDF_LP_4 ptdf0(.clk(CLK256), .p1(p1), .p2(p2), .p3(p3), .p4(p4), .x(Ia), .y(Ib));
    PTDF_LP_4 ptdf1(.clk(CLK256), .p1(p1), .p2(p2), .p3(p3), .p4(p4), .x(Qa), .y(Qb));
    //
    wire [15:0] RSSI;
    wire signed [15:0] FM;
    FM_Demodulator fm0(.clk(CLK1024), .I(Ia), .Q(Qa), .RSSI(RSSI), .FM(FM));
    //
    wire signed [15:0] Filt_RSSI, Filt_FM;
    MAF16_16bit maf1616a(.clk(CLK256), .x(RSSI), .y(Filt_RSSI));
    MAF16_16bit maf1616b(.clk(CLK256), .x(FM), .y(Filt_FM));
    //
    wire d0, d1;
    DataSlicer ds0(.clk(CLK256), .a(FM), .qa_ge_b(d0));
    assign d1 = ~d0;
    //
    wire d2, d3;
    DAC_PWM_10Bit dac2(.clk(CLK_50MHz), .x(Filt_RSSI / 64), .y(d2)); // unfiltered
    DAC_PWM_10Bit dac3(.clk(CLK_50MHz), .x(512 + FM / 64), .y(d3)); // unfiltered
    //
    assign DATA_OUT[0] = d0; // Square +
    assign DATA_OUT[1] = d1; // Square -
    assign DATA_OUT[2] = d2; // RRSI PWM
    assign DATA_OUT[3] = d3; // FM PWM
    //
    // ***** Transmitter Section *****
    //
    //wire signed [15:0] TxDAC;
    Transmitter Tx0(.CLK_50MHz(CLK_50MHz), .Sw(Sw), .PB(PB), .data(d0), .DAC(DAC), .DAC_CLK(DAC_CLK));
);
    //assign DAC =
    //
    // ***** Diagnostics *****
    //
    BarLED8 bl0(.clk(CLK_50MHz), .x(Filt_RSSI), .LED(LED));

```

```

HexDisplay2 hex2(.clk(CLK_50MHz), .x(Filt_RSSI), .DP(PB), .Module(Module), .Segment(Segment) );//  
Endmodule

```

14.2 Second Order PTDF Verilog Code

This Verilog code creates a second order PTDF low pass filter. Parameters p1 and p2 (12-bit) scale the cut frequency up or down whilst their ratio controls the filter Q. Input and output data is 20 bits.

```

module PTDF_LP_2(CLK, p1, p2, x, y); // Second order PTDF
    input CLK; // Input clock
    parameter bits = 20; //Maximum bit resolution = 20
    parameter pbits = 12; // resolution = 12 i.e. 0 < p* < 4095
    input [pbits - 1:0] p1, p2; // 0 < p* < 4095 => 0 < k* < 1
    input signed [bits - 1:0] x;
    output signed [bits - 1:0] y;
    //
    reg signed [bits + pbits + 1:0] e, y1, y3;// 2 bit margin
    reg signed [bits - 1:0] y2, y4, y; //
    //
    always @(posedge CLK)
    begin
        e = x - y2 - y4; //
        y1 = y1 + p1 * e; //
        y2 = y1 / 4096; //
        y3 = y3 + p2 * y2; //
        y4 = y3 / 4096; //
        y = y4; //
    end
    //
endmodule

```

14.3 10 Bit PWM DAC

This module creates a 10 bit PWM DAC from 1-bit input data. The input clock is 50 MHz and the output sample rate is ~24.4 kHz.

```

module DAC_PWM_10(clk, x, y); // 1 bit PWM DAC with 10 bit resolution
    parameter bits = 10; // PWM DAC output resolution
    input clk; //Use 50 MHz FPGA system clock
    input [bits - 1:0] x; // 10 bit input word
    output y; // 1 bit PWM output data (pre LPF)
    //
    reg increment, y; // increment ramps count up (1) then down (0)
    reg [bits - 1:0] count; // count has 10 bit resolution
    always @(posedge clk) // policy is to use positive edge triggers
    begin
        case (count)
            0: if(count == 0) increment <= 1; // increment count up
                 if(count == 1023) increment <= 0; // decrement count down
        endcase
        //
        case (increment)
            1: count <= count + 1; // count increases from 0 to 1023
            0: count <= count - 1; // count decreases from 1023 to 0
        endcase
        //
        if(x >= count) y = 1; // y is high if x exceeds count
        else y = 0; // y is low is x is less than count
    end
endmodule

```

15 Bibliography

1. Devices, A. *AD8347 0.8 GHz to 2.7 GHz Direct Conversion Quadrature Demodulator*. Analog Devices Component Data, 2005. 1-28.
2. Devices, A. *AD9283 8-Bit, 50 MSPS/80 MSPS/100 MSPS 3 V A/D Converter*. Analog Devices Component Data, 2001. 1-12.
3. Devices, A. *AD9740 10-Bit 210 MSPS TxDAC D/A Converter*. Analog Devices Component Data, 2009. 1-32.
4. Xilinx *Xilinx Spartan-3 FPGA Family: Introduction and Ordering Information*. 2009.
5. Management, R.S. and RSM *Radiocommunications (Radio Standards) Notice 2010*. Radio Communications Standards, 2010. 1-8.
6. Maxim *Random Noise Contribution to Jitter - Theory and Practice*. Maxim Application Notes, 2005. 1-13.
7. Maxim *Dynamic Performance Requirements for High-Performance ADCs and RF Components in Digital Receiver Applications*. Maxim Application Notes, 2004.
8. Xilinx *Xilinx LogiCORE Digital Clock Manager (DCM) Module*. Xilinx LogiCORE, 2009. 1-6.
9. Maxim *Aperture Jitter Calculator for ADCs*. Maxim Application Notes, 2009. 1-4.
10. Jones, P. and T. Hofner *Design Challenges for an Ultra-Low-Jitter Clock Synthesizer*. Maxim Application Notes, 2008. 1-8.
11. Arkesteijn, V.J., E.A.M. Klumperink, and B. Nauta, *Jitter requirements of the sampling clock in software radio receivers*. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2006. **53**(2): p. 90-94.
12. Pasupathy, S., *Minimum Shift Keying: A Spectrally Efficient Modulation*. IEEE Communications Magazine, 1979: p. 14-22.
13. Pulse *RF Transformer For Wideband RF Applications*. Pulse Catalog, 2009.
14. Pulse *RF Transformer Low Profile Transformer for Wideband RF Applications*. Pulse Catalog, 2010.
15. Xilinx *Xilinx LogiCORE IP Multiplier* 2009.
16. Xilinx *Xilinx LogiCORE IP DDS Complier*. 2009.
17. Xilinx *Xilinx LogiCORE IP CIC Compiler* 2010.
18. Scott, I.R. and T.C.A. Molteno, *Parameter Tuned Digital Filter*. ENZCon 2010, 2010.
19. Scott, I.R., R. Shadich, and W.M. Siddall, *Frequency Estimation*, in *US Patents Office*, U.P. Office, Editor 2006, Ian Russell Scott: United States. p. 16.
20. Xilinx *Xilinx LogiCORE Divider Generator v3.0*. Xilinx LogiCORE, 2009.
21. Xilinx *Xilinx LogiCORE CORDIC v4.0*. Xilinx LogiCORE, 2009.
22. Anthes, J. *OOK, ASK and FSK Modulation in the Presence of an Interfering signal*. RF Monolithics, 1998. 19.
23. Shima, J.M., *FM Demodulation Using A Digital Radio And Digital Signal Processing*, in *Electrical Engineering* 1995, University of Florida: Florida. p. 81.
24. Scott, I.R. and T.C.A. Molteno *Parameter Tuned Digital Filter (PTDF)*. ENZCon 2010, 2010. 1.