Efficient FM Demodulation by Single Tone Detection for FPGA Implementation

Matthew D'Souza, Michael Chan, Adam Postula
School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, QLD, Australia 4072
{dsouza, mchan, adam}@itee.uq.edu.au

Abstract

This paper presents a software radio receiver architecture that uses a single tone detection technique to demodulate a frequency modulated (FM) signal. In a software radio most of the signal processing algorithms are performed in software or in digital hardware instead of using analogue components. A conventional software radio receiver uses filters to demodulate FM signals. The Fourier Transform can also be used to demodulate FM signals. The presented single tone detection software radio (STDSR) uses the Goertzel Algorithm as a single bin discrete Fourier Transform to detect if a particular frequency is present in an FM signal. The STDSR is used to demodulate a commercial FM radio station audio signal. This paper compares the STDSR with filter based and Fast Fourier Transform (FFT) based software radios in terms of FPGA resource usage and computational efficiency. The STDSR was found to be more efficient to implement than the filter or FFT based software radios. Future work includes improving the audio signal quality and applying the STDSR architecture to other wireless modulation and demodulation methods.

1. Introduction

In recent years, software radios have become feasible for practical implementation in many applications due to the improving logic and digital signal processing (DSP) resource capacity of Field Programmable Gate Array (FPGA) devices. A software radio (SR) as described in [1] is a radio transmitter or receiver in which the main DSP functions such as filtering, signal mixing and frequency conversion are performed in the digital domain by software algorithms. A typical SR physical hardware structure consists of a Radio Frequency (RF) analog front-end connected to a high speed Analog to Digital Converter (ADC).

Software radios were first implemented using DSP microprocessors. DSP microprocessors are similar to conventional processors but contain special instructions for DSP operations. The main disadvantage of using DSP microprocessors is that parallel operations can only be performed using multiple DSP microprocessors. In recent years the logic capacity of FPGAs has increased to a level where it is more advantageous to use FPGAs for SR applications rather than DSP microprocessors. FPGAs are more suited to performing parallel DSP operations in terms of performance than DSP microprocessors.

FPGA manufacturers are now including special DSP slices into the FPGA fabric of their high-end FPGA devices. Examples are the Xilinx Vertix 4 Family [2] and the Altera Stratix II Family [3] that both have DSP slices that consist of multipliers and synchronization blocks. The high sampling rates of ADCs can range from 40MSPS (Mega Samples per Second) to over 100MSPS. Such high sampling rate ADCs can directly sample an RF signal at the intermediate frequency stage which simplifies the analog circuitry of the RF analog front end. An example is the Universal Software Radio Peripheral by the GNU radio project which is used to demodulate commercial Amplitude Modulated (AM) and FM radio stations [4]

This paper introduces a novel and efficient method of using signal tone detection in a software radio to demodulate FM signal transmissions. The presented architecture utilizes a frequency tone detection algorithm known as the Goertzel algorithm to demodulate a commercial FM radio station audio signal. This paper also discusses and compares the computational and FPGA related resources of the different DSP operations used by existing SR architectures for frequency demodulation.

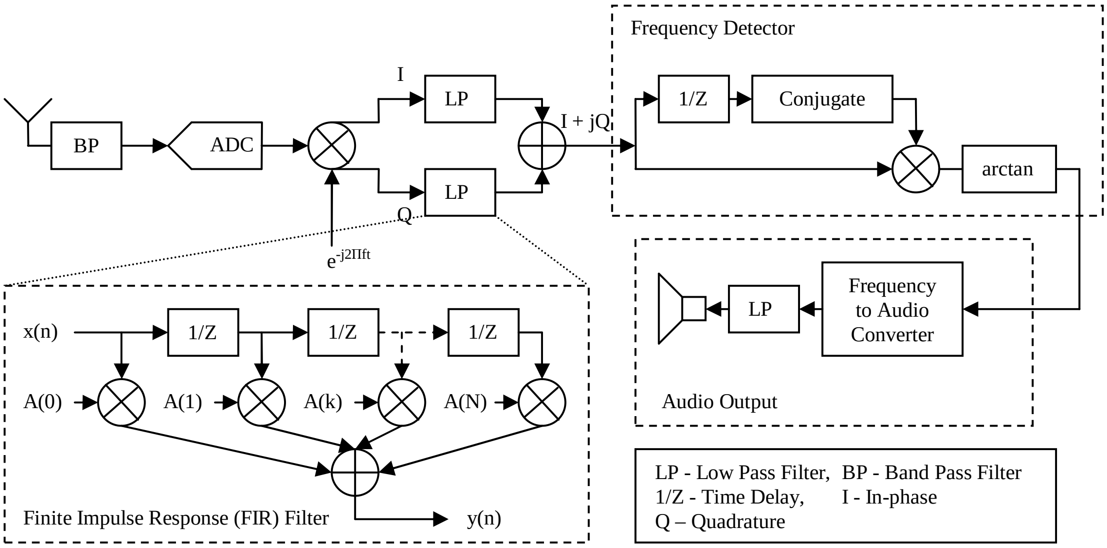


Figure 1. Filter based Software Radio Architecture

Section 2 introduces the common digital signal processing functions used in SRs and other DSP applications. Section 3 presents the architecture of the single tone detection SR (STDSR). Section 4 describes how the STDSR was implemented. Section 5 discusses the computational resources of the discussed SR architectures. Future work is discussed in Section 6 and conclusions are drawn in Section 7.

2. Conventional SR FM Demodulation Architectures

Simple frequency demodulation is performed by calculating the instantaneous frequency of the received signal. A typical spectrum of a commercial FM station broadcast signal can be seen in figure 2. It consists of the left minus right (L – R) stereo audio signal (0 to 16kHz), a stereo pilot tone at 19kHz and the left plus right (L + R) stereo audio signal (30kHz to 46kHz).

A conventional filter based FM SR structure can be seen in figure 1. It consists of an antenna connected to a tunable analog band pass filter. The output of the band pass filter is connected to an ADC. The ADC needs a suitably large frequency bandwidth in order to sample the incoming FM signal. Typically ADCs that can sample at intermediate frequencies, 40MSPS to 125MSPS, have a large bandwidth of 300MHz. Provided the signal of interest is bandwidth limited to less then half the sampling rate then it can be sampled and still contain the transmitted information [5]. This has the added effect of shifting down the signal frequency spectrum by any integer number of the

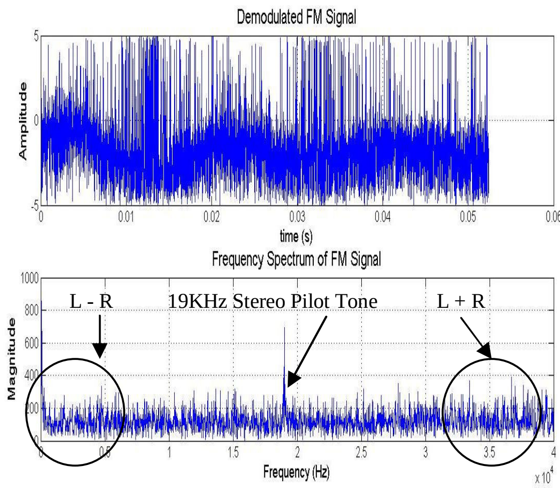


Figure 2. Demodulated FM Signal Waveform and Frequency Spectrum using the STDSR

sampling rate. In figure 1, the signal output of the band pass filter is then quadrature mixed or heterodyned with a locally generated oscillator signal which down converts the frequency of the signal of interest to DC.

The outputs of the quadrature mixer consist of the down converted in-phase (I) and quadrature (Q) signals. The I and Q signals are then passed through a low pass filter which allows only the signal of interest to pass through undistorted. The current frequency of the filtered combined I and Q signals is calculated using a derivative and an arctan operation. The derivative operation can be performed by multiplying the sample with the conjugate of the previous sample.

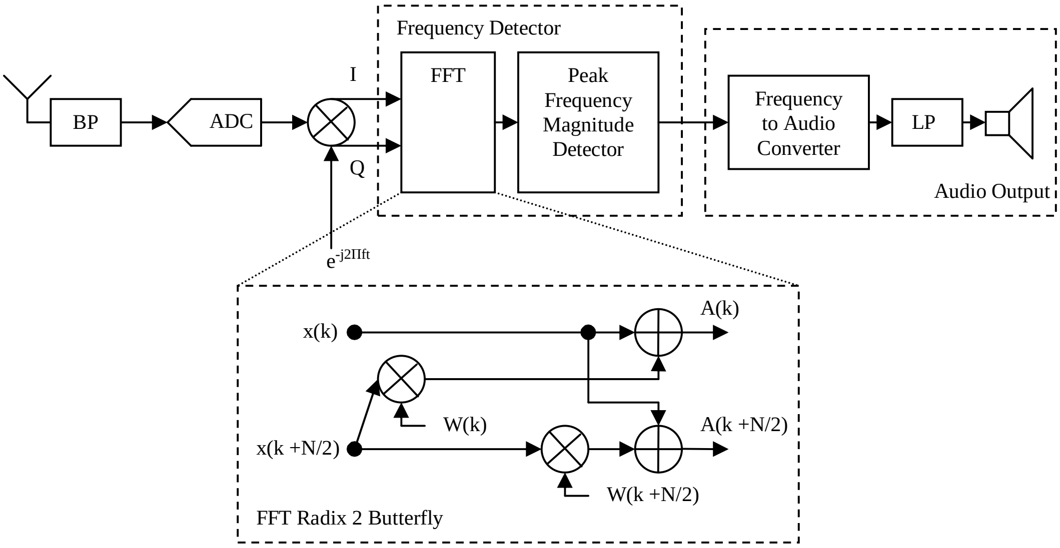


Figure 3. Fast Fourier Transform Based Software Radio Architecture

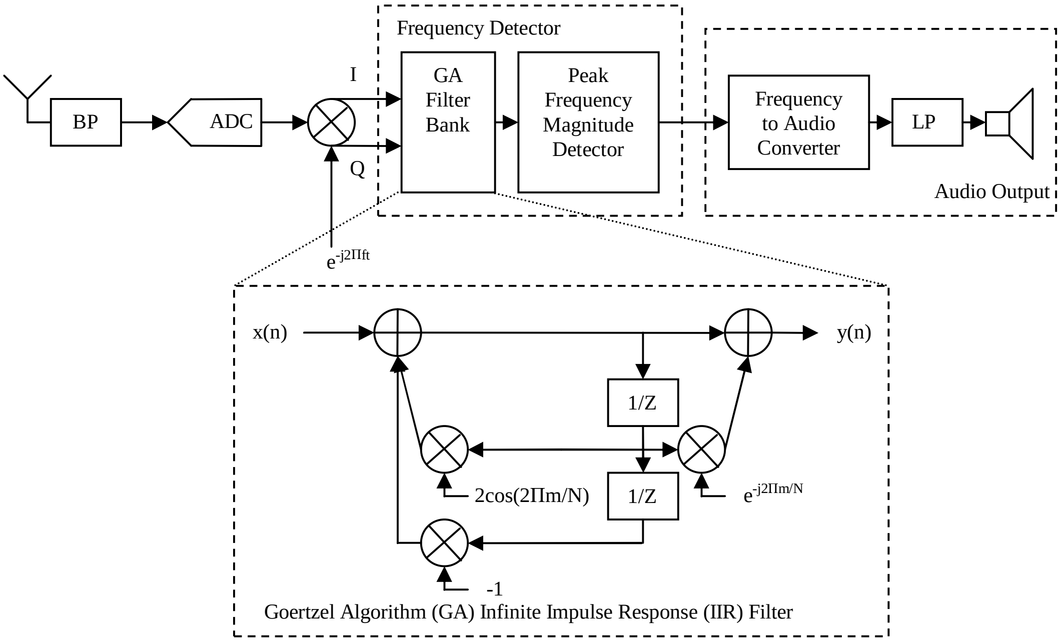


Figure 4. Single Tone Detection Software Radio Architecture

The arctan operation can be performed using the CORDIC algorithm[6]. Once the instantaneous frequency has been calculated, it is converted to an audio signal and is filtered further before being played on a speaker.

An alternate FM SR structure uses the Fast Fourier Transform (FFT) algorithm for frequency detection. This is shown in figure 3 where the frequency detection is performed by the FFT. The FFT calculates the frequency spectrum of the incoming signal. Once the spectrum has been calculated, the tones in the frequency range of interest are used to determine the value of the demodulated frequency. The index of the tone with the biggest magnitude determines the value of the demodulated frequency. The advantage of this method is that the precise filtering is not required. The

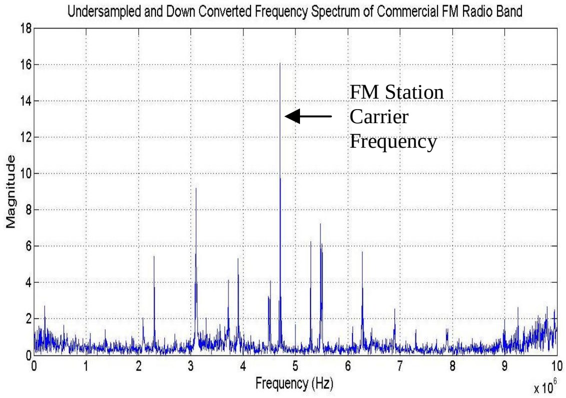
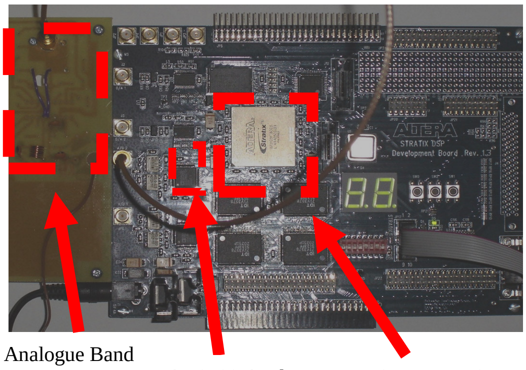


Figure 5. 100MHz to 110MHz Commercial FM Station Frequency Spectrum



Analogue Band Pass Filter with Antenna

AD9433 Analog to Digital Converter

Stratix EP1S25 Device

Figure 6. STDSR Hardware Implementation

FFT is implemented using a butterfly structure seen in figure 3. A disadvantage of using the FFT is that the frequency resolution is determined by the number of points the FFT has to process. Another disadvantage is the time invariance of the FFT. Time localization can be achieved with the FFT by processing a fixed block of samples to provide time localization. To improve the frequency resolution, the number of points processed by the FFT can be increased by zero padding the block of samples.

3. Single Tone Detection SR Architecture

The Single Tone Detection (STD) SR architecture is similar to the architecture of the FFTSR describe in the previous section. The architecture can be seen in figure 4. The STDSR utilizes the Goertzel algorithm (GA) to detect if a particular frequency is present in the incoming signal. The GA can be thought of as a single bin discrete Fourier Transform. The GA filter's transfer function is shown in (1) and is implemented as an infinite impulse response (IIR) filter as shown in figure 4.

$$(1)\qquad H(z)=\frac{1-e^{-j2\pi m/N}z^{-1}}{1-2\cos(2\pi m/N)z^{-1}+z^{-2}}$$

The STDSR uses a bank of GA filters to detect the frequency of the incoming signal. Each GA filter is configured to detect a particular frequency. As with the FFT, the frequency resolution is determined by the number of points to be processed. The GA filter is more computational efficient to use than an FFT if only a part of the spectrum is required. Another advantage is that the number of points that can be calculated by a GA filter is not restricted to a power of two as with the FFT.

4. Single Tone Detection SR Implementation

The STDSR was implemented on an Altera Stratix DSP Board. It contains a Stratix EP1S25 device which has 25560 logic elements, 80 (8x8) embedded multipliers, 1944576 bits of RAM[7]. The Stratix EP1S45 device is connected to two AD9433 ADCs from Analog Devices. Both ADCs can sample at maximum rate of 125MSPS and produce 12-bit signed samples[7]. The RF front end consists of a capacitor and inductor parallel circuit tuned to the 80MHz to 120MHz range. Figure 6 shows the Altera Stratix Board and RF front end. A commercial FM station broadcast at 105.4MHz was attempted to be demodulated. The commercial FM station carrier signals under-sampled at 40MSPS can be seen in figure 5.

Incoming samples were passed through a quadrature mixer that down-converted the sampling rate from 40MSPS to 10MSPS. The quadrature mixer only required add and subtract operations since the incoming sample rate was four times the down-converted sampling rate [5]. The down-converted samples were then divided into sampling blocks of 125 samples. This was done to provide time localization of the detected frequency. This means that the frequency of the incoming signal is detected every 12.5us, which is suitable for demodulating the audio signal (0 – 40kHz) of a commercial FM radio station.

The GA filter bank used to detect the frequency of the incoming signal consisted of sixteen GA filters. Each GA processed 2048 points by zero padding each sampling block with 1923 zeros. This was done to allow the GA filters to estimate the incoming signal frequency to within 5kHz. Each sample was a 24 bit fixed point number which consisted of 14 bits for the fractional part, 9 bits for the magnitude and 1 bit for the sign of the number.

In the context of FM radio demodulation, different FM radio stations can be selected by changing the constants in the GA filter bank. Thus the entire range of FM stations can be accessed with a negligible increase to digital hardware complexity.

5. Analysis of Logic and Computational Efficiencies

Table 1 compares the computational resources of the main DSP units which are used in the three FM SR architectures for frequency detection. Each DSP unit requires resources such as 24 bit multipliers and memory to store 24 bit coefficient numbers. The amount of logic elements used by each DSP unit is also compared. A logic element is the smallest logic unit used by the Altera Stratix EP1S25 device. Table 2 compares the computational units used for frequency demodulation used by the three SR architectures.

Table 1. DSP Units used for Frequency Detection.

SR	DSP Unit	Multiplier used	Memory used (bit)	Logic Elements
Filter	FIR Filter	10	240	1000
FFT	Radix 2 Butterfly	6	1.08M	936
STD	GA IIR Filter Bank	33	1152	13045

Table 2. Computational Units used for Frequency Demodulation.

SR	Unit	Logic Elements
Filter	CORDIC	595
FFT	Comparator	100
STD	Comparator	100

The main DSP unit of the filter based SR is the Finite Impulse Response (FIR) filter. A typical FIR filter structure can be seen in figure 1. The number of multipliers depends on the number of filter coefficients used. For this case, 10 coefficients are sufficient. The radix 2 butterfly structure is the main DSP unit used by the FFTSR. Its structure can be seen in figure 3. A

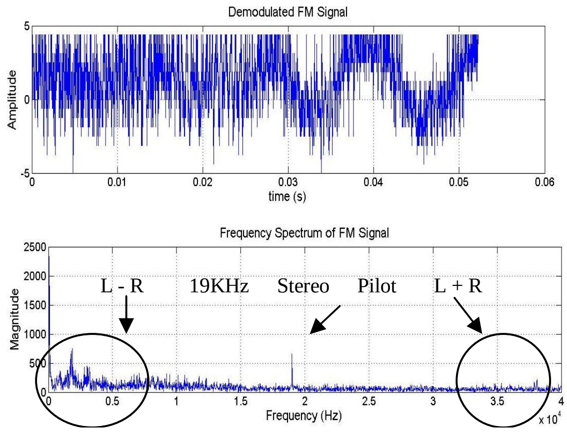


Figure 7. Demodulated FM Signal Waveform and Frequency Spectrum using the Filter based Software Radio

2048 point FFT is used by the FFTSR which requires 2048 complex coefficients for eleven stages. A complex multiplier actually consists of 3 multipliers and 3 adders. The STDSR uses a basic IIR filter structure to implement each GA filter which can be seen in figure 4. The GA filter bank uses one real multiplier per GA filter and one complex multiplier to compute the final stage of each GA filter.

The CORDIC unit is used by the filter based SR to implement the arctan function to calculate the frequency of the incoming signal. The FFT and STD SRs only require a magnitude comparator to determine the frequency of the incoming signal.

5.1 Logic and Computational Usage Comparison

As can be seen in table 1, the number of multipliers used is directly related to the logic element usage. The FFTSR requires the least number of multipliers but needs the most amount of memory to store its coefficients. The filter SR requires fewer multipliers compared to the STDSR but it requires an analogue quadrature mixer to demodulate the FM signal to the same level as that obtained using the STDSR and FFTSR. Figure 7 shows the demodulated FM signal waveform and frequency spectrum using the filter based SR with the same quadrature mixer used by the STDSR. The L - R signal is not as prominent as it is in figure 2.

The FFTSR requires fewer multipliers then the STDSR but it requires substantially more memory and only a fraction of the produced outputs are used. The GA filter does not require the same amount of buffering of the incoming samples as required by the radix 2 FFT. This is also the same for the filter based

SR. Both the STDSR and FFTSR use a simpler frequency magnitude selection process to produce the final audio signal compared to the CORDIC unit used by the filter based SR.

6. Future Work

Future work includes improving the STDSR in terms of audio quality. This involves developing and implementing frequency spectral peak location algorithms. Other future work involves applying the GA filter structure for use with other forms of wireless modulation/demodulation methods such as Quadrature Amplitude Modulation.

7. Conclusion

A software radio architecture that used a single tone detection method to demodulate FM signals is presented in this paper. The single tone detection software radio (STDSR) used the Goertzel Algorithm (GA) to perform a single bin Discrete Fourier Transform to detect if a particular frequency is present in an incoming FM signal. The STDSR is used to demodulate a commercial FM radio station audio signal. The STDSR is implemented on an Altera Stratix DSP development board.

The STDSR implemented the GA as an infinite impulse response (IIR) filter. The STDSR used a bank of sixteen GA filters to demodulate an FM signal. The STDSR use of the GA as an IIR filter also allows it not to buffer incoming samples as required by the FFTSR. The number of samples that can be processed by the GA filter is not limited to a power of two. This means that the GA has a more flexible frequency resolution than the FFT.

The STDSR is better suited for FPGA implementation than the filter based SR because it does not require an analogue quadrature mixer as needed by the filter based SR. The STDSR is found to have a smaller memory usage than the FFTSR. This is because the STDSR only calculates the part of the frequency spectrum required to detect the FM radio station's carrier signal while the FFTSR calculated the entire commercial FM radio spectrum. The STDSR combines the best features of the filter based SR and the FFTSR. Future development of the STDSR involves improving audio fidelity by using various frequency spectral peak location algorithms and to apply the GA filter structure to other wireless modulation and demodulation methods.

8. References

[1] J. Mitola, "The Software Radio Architecture"," *IEEE Communication Magazine*, vol. 33, pp. 26-38, May 1995.

[2] Xilinx, "Virtex-4 FPGAs." www.xilinx.com/virtex4, 2005.

[3] Altera, "Stratix II Devices." www.altera.com/stratix2.

[4] GNURADIO, "GNU Radio - The GNU Software Radio." www.gnu.org/software/gnuradio/, 2005.

[5] R. G. Lyons, *Understanding Digital Signal Processing*, Second ed. Upper Saddle River, New Jersey: Prentice Hall, 2004.

[6] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*. Heidelberg: Springer-Verlag, 2001.

[7] Altera, "Stratix EP1S25 DSP Development Board Data Sheet," Altera, 2003.