Chapter 1
INTRODUCTION

1.1 OBJECTIVE

Recent advancement in semiconductor technology has made it possible to process high speed communication signals in radio systems using as much digital technology as possible. This makes the system very flexible and adaptive. Such a technology is called Software Defined Radio (SDR).The last decade has seen digital radio systems becoming more and more popular replacing analog radio systems for a wide range of applications in wireless telecommunications and broadcasting. We aim to deisgn and implement a software defined radio that performs all the functionalities of mixer and digital down converter in software.

Traditional analog radio receivers and transmitters consist of dedicated analog circuits for filtering, tuning, and demodulating/modulating a specific type of waveform. These hardware based radio systems are inflexible and hard to modify if changes are to be made to its fundamental characteristics such as demodulation/modulation types. To make the system more flexible, SDR technology facilitates implementation of some of these functions in software. This results in reconfigurable software radio systems where changes to its fundamental characteristics can be made simply by modifying the software, whereas a complete hardware based radio system would require hardware modification in order to change these parameters. In Software Defined Radio technology, radio functions are implemented in software running on digital signal processing hardware platforms. The three most commonly used platforms are DSPs (Digital Signal Processor), ASICs (Application Specific Integrated Circuit), and FPGAs (Field Programmable Logic Array). FPGA is able to perform any task by mapping the task to the hardware. Re-configurability is a feature, which enables FPGA to realize any user hardware by changing the configuration data on a chip as many times as needed. FPGAs consists not only look-up tables, registers, multiplexers, distributed and block memory, but also dedicated circuitry for fast adders, multipliers, and I/O processing. FPGA has a capability for implementing highly parallel arithmetic architectures. Hence here we use FPGA as the hardware platform for implementing the radio receiver.

1.2 SCOPE

SDR technology has potential applications in all areas of radio communications and broadcasting. The constant evolution of wireless network standards from 2G to 3G and further to 4G causes problems for subscribers, network operators and equipment vendors. Users are forced to buy new handsets every time a new generation is deployed. The air interface and link-layer protocols differ across different continents, for example, European wireless networks are mainly GSM/TDMA while US networks are predominantly CDMA. This problem has inhibited the deployment of global roaming facilities and thus causes inconvenience to subscribers who travel frequently between continents. SDR technology promises to solve these problems as it enables implementation of radio functions in networking infrastructure and subscriber terminals as software modules running on generic hardware platforms. This relieves the cost and complexity of migrating network from one generation to another since the migration would only involve a software upgrade. A fully implemented SDR will have the ability to navigate a wide range of frequencies with programmable channel bandwidth and modulation characteristics. Furthermore, since radio functions are implemented in software, multiples software modules implementing different standards can co-exist in the equipment and handsets.

1.3 MOTIVATION

Being students of electronics and communication in a premier institution which stresses on exposing students to recent trends in the field, we decided to do a project which would satisfy our desire to be familiar with the latest technology. Software Defined Radio (SDR) is a field in which research and improvements are going on throughout the world. Since implementation of software defined radio in Digital Signal Processor (DSP) was widespread, we decided to move one step ahead and implement SDR on Field Programmable Gate Array (FPGA) platform. The fact that we were to learn a hardware description language in the seventh semester, compelled us to finalise this topic for our project.

Chapter 2

LITERATURE REVIEW

2.1 SDR - Benefits

The benefits of SDR are compelling. For radio equipment manufacturers and system integrators, SDR enables:

- A family of radio products to be implemented using a common platform architecture, allowing new products to be more quickly introduced into the market.
- Software to be reused across radio products, reducing development costs dramatically.
- Over-the-air or other remote reprogramming, allowing ”bug fixes” to occur while a radio is in service, thus reducing the time and costs associated with operation and maintenance.

For Radio Service Providers, SDR Enables:

- New features and capabilities to be added to existing infrastructure without requiring major new capital expenditures, allowing service providers to quasi-future proof their networks.
- The use of a common radio platform for multiple markets, significantly reducing logistical support and operating expenditures.

• Remote software downloads, through which capacity can be increased, capability upgrades can be activated and new revenue generating features can be inserted.
For End Users - from business travelers to soldiers on the battlefield, SDR technology aims to:
• Reduce costs in providing end-users with access to ubiquitous wireless communications enabling them to communicate with whomever they need, whenever they need to and in whatever manner is appropriate.

2.2 SDR Related Technology

SDR can act as a key enabling technology for a variety of other reconfigurable radio equipments commonly discussed in the advanced wireless market.While SDR is not required to implement any of these radio types, SDR technologies can provide these types of radio with the flexibility necessary for them to achieve their full potential, the benefits of which can help to reduce cost and increase system efficiencies: Adaptive Radio Adaptive radio is radio in which communications systems have a means of monitoring their own performance and modifying their operating parameters to improve this performance. The use of SDR technologies in an adaptive radio system enables greater degrees of freedom in adaptation, and thus higher levels of performance and better quality of service in a communications link. Cognitive Radio Cognitive radio is radio in which communication systems are aware of their internal state and environment, such as location and utilization on RF frequency spectrum at that location. They can make decisions about their radio operating behaviour by mapping that information against predefined objectives. Cognitive radio is further defined by many to utilize Software Defined Radio, Adaptive Radio, and other technologies to automatically adjust its behaviour or operations to achieve desired Venn diagram illustrating relationship between associated advanced wireless technologies objectives. The utilization of these elements is critical in allowing end-users to make optimal use of available frequency spectrum and wireless networks with a common set of radio hardware. As noted earlier, this will reduce cost to the end-user while allowing him or her to communicate with whomever they need whenever they need to and in whatever manner is appropriate. Intelligent Radio Intelligent radio is cognitive radio that is capable of machine learning. This allows the cognitive radio to improve the ways in which it adapts to changes in performance and environment to better serve the needs of the end user. These types of radio adaptive radio, cognitive radio and intelligent radio do not necessarily define a single piece of equipment, but may instead incorporate components that are spread across an entire network.

2.3 SDR Hardware Platforms

The hardware aspects of a SDR platform consist of the radio-frequency (RF) parts, communications links to the software-based signal processing elements (mostly a Host-PC). The rest may consist one or more of the following;

•ASICs (application-specific integrated circuits).

•FPGAs (field-programmable gate arrays).

•DSPs (digital signal processors).

•GPPs (general-purpose processors). The ASICs are non-reprogrammable that contradicts the principle of SDR, but still used as a part for special characteristics. The FPGAs provide high computing power due to quasi-parallel processing nature while the DSP and GPPs are essentially serial in operation. The main strengths of DSPs and GPPs are their flexibility and easy configurability [12]. The various SDR hardware solutions (RF Front-End) are available in commercial and academic area, providing potential opportunities in the radio communication industry. Universal Software Radio Peripheral 2 (USRP2): It is a brainchild of Matt Ettus (Ettus Research LLC). The USRP family of products has been nominated Technology of the Year award from the Wireless Innovation Forum, 2010. The USRP2 is a second generation of Universal Software Radio Peripheral, its platform consist Xilinx Spartan-III FPGA and general purpose AeMB processor . The USRP-E100 is its next revision thats a stand-alone system powered by the combination of an ARM Cortex A8 processor and TI C64x+ DSP and a Xilinx Spartan 3A-DSP1800 FPGA. Rice Wireless Open-Access Research Platform (WARP): The wireless open-access research platform of Rice University is a scalable and extensible programmable platform, built for prototyping advanced wireless networks. The Xilinx Virtex-4 FX100 FPGA is used to enable programmability of both physical and network layer protocols on a single platform. Berkeley Emulation Engine 3 (BEE3): BEE3 is new generation of Berkeley Emulation Engine-2 . It is jointly developed by Microsoft Research, UC Berkeley and BEEcube Inc. The platform contains four Virtex-5FPGAs, emulates over RISC processor cores concurrently at a 100 MHz rate. It is useful for most computationally intensive real-time applications, a high-speed multiple FPGA and validation solution. BEE3 is suited as a real-time, real-world prototyping and development platform. Kansas University Agile Radio (KUAR): The KUAR hardware employs a Xilinx Virtex II Pro P30 FPGA along with 1.4 GHz Pentium M processor. It has been promoted through the defense advanced research projects agency (DARPA) next generation (XG) program. The complete system was developed in Simulink, implemented in Xilinx ISE generating the Verilog code from Simulink model(s) using a Modelsim of Mentor Graphics. Small Form Factor Software Defined Radio (SSF-SDR): The Xilinx Inc. in collaboration with Lyrtech and Texas Instruments incorporated a SFF-SDR development platform for developing the handheld and mobile radios . The Xilinx Virtex-IV FPGA, TI DSP TMX320 and TI MSP430 MCU are used in addition with ARM926 embedded processor [20]. Intelligent Transport System (ITS); National Institute of Information and Communications Technology (NICT) of Japan, developed a software-defined radio platform so-called NISTITS. It is specially designed for mobile communication, wireless LAN and digital terrestrial TV . The platform contains Xilinx Virtex-4 FPGA, and two general purposes process (GPP) 430 MIPS (240 MHz). Table II shows a detailed survey of existing SDR hardware platforms and their performance. The USRP2 is a cheaper and fast enough; the Xilinx Spartan-3 FPGA (XC3S2000) contains; 2M system-gates, 46080 equivalent logic cells. The larger FPGA and general purpose AeMB processor allows the USRP2 to be used as a standalone system without a host computer in many cases; the USRP2 is used in the prototype.

Chapter 3

SYSTEM OVERVIEW

3.1 Proposed System

FPGA (Field Programmable Gate Array) is able to perform any task by mapping the task to the hardware. One of the advantages of FPGA is its re-configurability capability that ASIC does not have. Re-configurability is a feature, which enables FPGA to realize any user hardware by changing the configuration data on a chip as many times as needed, they are often programmed with a hardware description language, such as Verilog or VHDL. The logic fabric of today's FPGAs consists not only of look-up tables, registers, multiplexers, distributed and block memory, but also dedicated circuitry for fast adders, multipliers, and I/O processing (e.g., giga-bit I/O) . The memory bandwidth of a modern FPGA far surpasses that of a microprocessor or DSP processor running at clock rates two to ten times that of the FPGA. In addition, FPGA has a capability for implementing highly parallel arithmetic architectures. Some special signal processing algorithms suitable for FPGA architectures have been developed such as distributed arithmetic algorithm. The FIR filtering using distributed algorithm, for example, has the same speed whether the number of filter taps is 1 or 100. This makes it suitable for implementing a high-speed filter with large number of taps.

In summary, with its many advantages, FPGA has become key components in implementing high performance digital signal processing systems. In this project, we will be using FPGA as the targeted hardware platform for implementing Software Defined Radio Receiver.

3.2 Feasibility Study

As mentioned above, in Software Defined Radio technology, radio functions are implemented in software running on digital signal processing hardware platforms. The five most commonly used platforms are general purpose microprocessors (GPP), and graphics processing units (GPU), DSPs (Digital Signal Processor), ASICs (Application Specific Integrated Circuit), and FPGAs (Field Programmable Logic Array). General purpose microprocessors, such as the Intel and AMD devices commonly found in personal computers, are not specialized for any particular application. Therefore, they are very flexible. However, SDR systems using GPPs are often wasteful since these processors are designed for speed and generality rather than power efficiency or mathematical operations. Graphics processing units employ massively parallel architectures that are optimized for vector manipulations and other graphical operations. Such parallel designs are well suited for signal processing, but GPUs are still relatively difficult to program and they are extremely power hungry.

A digital signal processor solve these two problems and also does signal processing by fetching instructions and data from memory, does operations, and stores the results back to memory, just like a regular CPU. The difference between a DSP chip and a CPU chip is that a DSP chip usually has a block that does high-speed signal processing, especially a block called MAC

(Multiply and Accumulate). By calling different routines in memory, a DSP chip can be reconfigured to perform various functions. On the other hand, their narrow focus makes them potentially slow for other applications. ASIC (Application-specific Integrated Circuit) is an integrated circuit that is designed to perform a fixed specific task. Examples of signal-processing specific ASICs are DDC (digital down converter) chip, and digital filter chips. The disadvantage of ASIC is that its functionalities are fixed and thus cannot be changed by the user.

FPGA (Field Programmable Gate Array) is able to perform any task by mapping the task to the hardware. One of the advantages of FPGA is its re-configurability capability that ASIC does not have. Re-configurability is a feature, which enables FPGA to realize any user hardware by changing the configuration data on a chip as many times as needed, they are often programmed with a hardware description language, such as Verilog or VHDL. The logic fabric of today's FPGAs consists not only of look-up tables, registers, multiplexers, distributed and block memory, but also dedicated circuitry for fast adders, multipliers, and I/O processing (e.g., giga-bit I/O) . The memory bandwidth of a modern FPGA far surpasses that of a microprocessor or DSP processor running at clock rates two to ten times that of the FPGA. In addition, FPGA has a capability for implementing highly parallel arithmetic architectures. Some special signal processing algorithms suitable for FPGA architectures have been developed such as distributed arithmetic algorithm. The FIR filtering using distributed algorithm, for example, has the same speed whether the number of filter taps is 1 or 100. This makes it suitable for implementing a high-speed filter with large number of taps.

In summary, with its many advantages, FPGA has become key components in implementing high performance digital signal processing systems.

3.3.4 DIGITAL TO ANALOG CONVERTER

As the ADC may be thought of as the heart of an SDR receiver, in like manner, the DAC may be seen as having the same importance in the SDR. High-performance DACs with digital functionalities provide an increased level of flexibility. Digital-to-analog converter (DAC or D-to-A) is a device that converts a digital (usually binary) code to an analog signal (current, voltage, or electric charge). As per the Nyquist sampling theorem, a DAC can reconstruct the original signal from the sampled data provided that its bandwidth meets certain requirements (e.g., a baseband signal with bandwidth less than the Nyquist frequency).

3.3.5 SOUND CARD

It is the expansion card that allows computer to send audio information to an audio device like speakers or a pair of headphones. It is piece f hardware in a computer that turns digital sound information created by software. Most sound cards use a digital to- analog converter (DAC), which converts recorded or generated digital data into an analog format. The output signal is connected to an amplifier, headphones, or external device using standard interconnects. Digital sound reproduction is usually done with multichannel DACs, which are capable of simultaneous digital samples at different pitches and volumes, and application of real-time effects such as filtering.

3.4 Software Block Diagram

Since the initial development of wireless communications, analog has been the primary technology platform for two-way radios. The majority of radio systems today still run in analog mode. But as digital technology progresses and transforms it is beginning to play a major role in professional two-way radio communications. The disadvantages of analog systems include:

- Typically, only one two-way conversation can occur at a time on each channel.
- Analog systems require hardware receivers and transmitters that are designed to fit the particular transmission. Analog devices can be upgraded with features but not technology.
- No software-driven business applications are available for analog radios.

The advantages of digital systems include:

- More simultaneous talking paths are possible, and information such as unit ID, can be embedded into a single digital radio channel.
- Bandwidth consumption is reduced.

Since a software-defined radio system, or SDR, is a radio communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software we choose digital domain for its implementation.

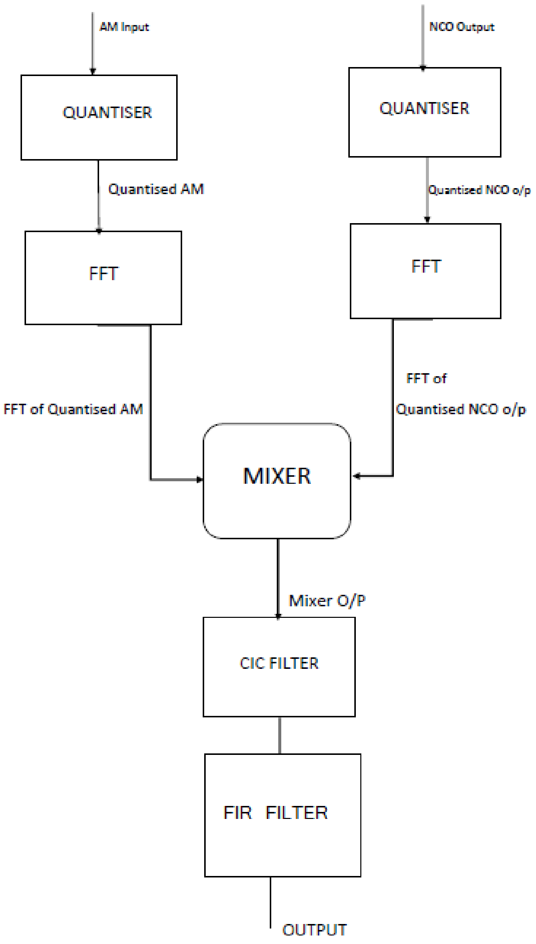


Figure 3.2: Flow Chart[1]

3.4.1 DIGITAL DOWN CONVERTER

A fundamental part of many communications systems is Digital Down Converter (DDC). It is the key component for digital radio. Digital Down Conversion is a technique that takes a band limited high sample rate digitized signal, mixes the signal to a lower frequency and reduces the sample rate while retaining all the information. The main advantage of using an FPGA to implement the Digital Down Converter is the speed, but it also has advantages associated with any digital signal processing system in that once it is defined it is fixed relative to the sample frequency, and will not change with time or temperature. In many cases signal of interest represents a small portion of the bandwidth. A DDC allows the rest of that data to be discarded, allowing more intensive processing to be performed on the signal of interest. It allows the signal to be shifted from carrier frequency down to baseband i.e. 0Hz .DDC performs two essential software radio functions: frequency translation and channel filtering.

The main parts of a DDC are: Numerically Controlled Oscillator (NCO), Digital Mixer, and Decimator. Decimator consists of a Cascaded Integrator Comb (CIC) filter and a FIR low pass filter. Mixer and NCO performs the frequency translation and decimator performs channel filtering.

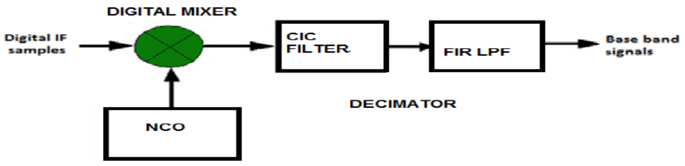


Figure 3.3: Block diagram of DDC[1]

3.4.2 NUMERICALLY CONTROLLED OSCILLATOR

It is also called Direct Digital Synthesizers (DDS).As the name suggests this form of synthesis generates the waveform directly using numerical / digital techniques. A direct digital synthesizer operates by storing the points of a waveform in digital format, and then recalling them to generate the waveform. The rate at which the synthesizer completes one waveform then governs the frequency. The operation can be envisaged more easily by looking at the way the phase progresses over the course of one cycle of the waveform. This can be envisaged as the phase progressing around a circle. The synthesizer operates by storing various points in the waveform in digital form and then recalling them to generate the waveform.

3.4.3 PHASE ACCUMULATOR

The phase accumulator has a counter carry function that allows it to act as a phase wheel. Imagine the sine-wave oscillation as a vector rotating around a phase circle and phase wheel itself has many designated points. Each designated point corresponds to the equivalent point on a cycle of a sine wave. Visualize that as the vector rotates around the wheel, the sine of the angle will generates a corresponding output sine wave. When rotating at a constant speed, one revolution of the vector around the phase wheel is equivalent to one complete cycle of the output sine wave. The contents of the phase accumulator correspond to the points on the cycle of the output sine wave. The phase accumulator provides the equally spaced angular values accompanying the vectors linear rotation around the phase wheel. The PA is a modulo-M counter, each times it receives a clock pulse it will store its increment number. The magnitude of the increment is determined by the frequency tuning word and it effectively sets how many points to skip around the phase wheel and this word forms the phase step size between reference clock updates. The larger the jump size, the faster the phase accumulator overflows and completes one full round of the phase wheel is equivalent to a sine-wave cycle. The resolution of the PA (n-bits) represented by the number of discrete phase points contained in the wheel whereby it determines the tuning resolution of the NCO. The tuning word in the phase register, M determines the speed of overflowing and the amount of phase accumulator is incremented each clock cycle.

3.4.4 PHASE TO AMPLITUDE CONVERTER

The signal converts digital phase input from the accumulator to output amplitude. This output represents the phase of the wave as well as an address to a word, which is the corresponding amplitude of the phase in the LUT. The digital phase information is used to address the ROM. A ROM pointer are implemented which points to the memory location in ROM that corresponds to the appropriate magnitude for its corresponding phase angle to meet the input requirements of the ROM block,. After each clock cycle, the appropriate magnitude of the ROM output is assigned to create a complete sine wave.

3.4.5 DIGITAL MIXER

The Digital Mixer is simply a 2 inputs multiplier which outputs the product of two digital samples. Digital output samples from the A/D are mathematically multiplied with digital samples of a sine (or cosine) signal from the local oscillator. Note that the input samplesfrom the A/D and sine (cosine) samples from the Local Oscillator are generated at the same rate fs (the sampling frequency of the A/D). The multiplication is sample-bysample and thus the mixer produces samples at the same rate of fs. Unlike analog mixer which produces many unwanted mixer products, the digital mixer only produces the sum and the difference frequency signals.The figure illustrates the effect of mixing signal with a local oscillator signal. The mixer shifts the spectrum at the frequency of the local oscillator down to DC.

3.4.6 DECIMATOR

The concept of changing the sampling rates downward to a lower sampling rate is called Decimation. Main parts of a decimator section are CIC filter and FIR low pass filter.

3.4.7 CIC FILTER

The cascaded Integrator Comb (CIC) or Hogenauer filter is a class of FIR filter. It is a multiplier-less filter architecture that is extremely important for implementing area efficient high sample rate changes in Digital Down Converters (DDC). Generally FIR filters are preferred over IIR filters in multirate systems because only a fraction of the calculations that would be required to implement a decimating or interpolating FIR in a literal way actually needs to be done. Since FIR filters do not use feedback, only those outputs which are actually going to be used have to be calculated. Therefore, in the case of decimating FIRs in which only 1 of N outputs will be used, the other N-1 outputs don't have to be calculated. A CIC filter consists of an equal number of stages of ideal integrator filters and comb filters. Its frequency response may be tuned by selecting the appropriate number of cascaded integrator and comb filter pairs. The highly symmetric structure of a CIC filter allows efficient implementation in hardware. These characteristic make CICs very useful for digital systems operating at high rates.

3.4.8 FIR FILTER

Finite Impulse Response (FIR) Filter is one of the primary types of filters used in digital signal processing application. Since the filter does not use feedback the impulse response of the filter is finite. Because of the passband droop, and therefore narrow usable passband, many CIC designs utilize an additional FIR filter at the low sampling rate. This filter will equalize the passband droop and perform a low rate change, usually by a factor of two to eight. Here we use a low pass filter for serving the purpose.

Chapter 4

SYSTEM DESIGN

4.1 Software

Since the DDC hardware is going to need to decimate the input signal by a large factor, the decimation will be broken up into a cascade of three decimating filters. The first filter in the cascade is going to have to operate at a fairly high sample rate. Because of this,the first filter in the cascade was chosen to be a cascaded integrator-comb (CIC) filter. These filters have a low-pass response and are multiplier-free. These filters can also be made to decimate or interpolate. After the CIC filter partially decimates the input signal, a more complicated filter can be implemented to perform the rest of the decimation. Due to this, the next filter in the cascade were chosen to be FIR decimating filter. The theory of each of these types of filters were described in the previous sections.

Performing down conversion of the signal in FPGA enables the use of generic adders and multipliers. Also, initializing the butterfly mixers is done by loading them with .coe file containing AM samples and NCO values.

4.1.1 Design Hierarchy

The Verilog design has several different levels of hierarchy. This section explains each level of the hierarchy in detail. The following sections descend into the design hierarchy supplying descriptions of each module.

The top module is the first block in the processing chain. The 12 bit output is fed to a DAC which is further connected to a sound card. For the reasons discussed earlier, there are no multipliers in the demodulator circuitry. This module assigns the output according to a clock, and reset signals. The top module calls the FIR filter module which in turn further calls CIC decimator module.

The FIR filter is the final step in the digital down conversion of the signal. As mentioned before, after the CIC filter partially decimates the input signal, it gets quantized and then input to the FIR filter. This is so that the FIR filter can maintain full precision throughout the filtering operations. In FIR filter module, the output of the CIC filter module are shifted by 5,4,3,2 and 1 times respectively and added. The MSB bits are added and fed as output from the FPGA.

4.1.2 Butterfly Mixer

The value of butterfly mixer is calculated using multipliers. Every time a clock period is completed , the ticks counter is incremented by one. The calculation is done in 4 stages-one each to find the real part, imaginary part of the value, to delay the output and to sum the values.

4.1.3 Numerically controlled Oscillator

The output is 8 bit fed into the butterfly mixer. Everytime the modulo counter wraps up, the ticks counter is incremented by one.

4.2 Hardware Design Details

4.2.1 RF Front End and ADC9233

After down conversion of AM (550KHz to 1650KHz) using RF Front End, an IF of 455KHz is obtained. A Nyquist rate of 900 kilosamples per second(2*455) is the minimum requirement. Therefore ADC speed of 1000 kilosamples per sec was chosen . We zeroed in on ADC AD9233, which met this requirement . The increased speed is used to compensate for the anti alias roll off. The output of the ADC is buffered before feeding to the input pins of Spartan3 FPGA. Low power is consumed by the ADC,i.e., 395mW at 125MSPS. The 12 bit ADC has a Signal to Noise Ratio of 73dB.

4.2.2 Spartan 3

The Input/Output Block (IOB) of Spartan 3 XC3S400 FPGA provides a programmable, bidirectional interface between an I/O pin and the FPGAs internal logic. IOBs are allocated among eight banks, so that each side of the device has two banks. There are a total of 116 differential I/O pairs and 264 user I/O pins in the FPGA. ADC and DAC interfacing uses 12 pins each. The maximum and minimum voltage levels supplied by the I/O pins in the FPGA for High Speed Tranceiver Logic are1.8V and 1.1 V respectively. The voltage supplied to the FPGA Spartan3e is 2.5V.

4.2.3 ADC - AD9233

Monolithic single 1.8V , 12bit , 80MSPS ADC. Data conversion rate is 1Msps. Clock frequency required is . The voltage supply required is 1.8V .The output of the ADC is buffered is done before fed to the input pins of the Spartan3 FPGA. Low power is consumed at 395mW at 125MSPS. SNR is 69.5dB to 70MHz input. AD 9233 features a separate digital output driver supply to accommodate 1.8V to 3.3V logic families. Hence it was found to be compatible with the FPGA we had selected

The RF Front End stage is to be designed with the receiver tuned to a FM broadcast. The anti-aliasing filter, filters the RF band of 88 MHz to 108 MHz and rejects all the other frequencies that are out of band.ie., if an FM station at 107.9 MHz is desired and the IF frequency is 10.7 MHz, the local oscillator would be tuned to 107.9−10.7 =97.2 MHz.

The signal from the receiver (centered in 10.7 MHz and with 220 KHz bandwidth) is digitalized by the ADC with 14 bit precision. We plan to use 11.875 MHz for the A/D sampling clock CK_{ADC} signal so that after 1:16 decimation we will have an integer number of samples per RBDS bit (625 samples):

4.2.4 Design Example

$$(11.875e6/1187.5)/16 = 625S/bit \tag{4.1}$$

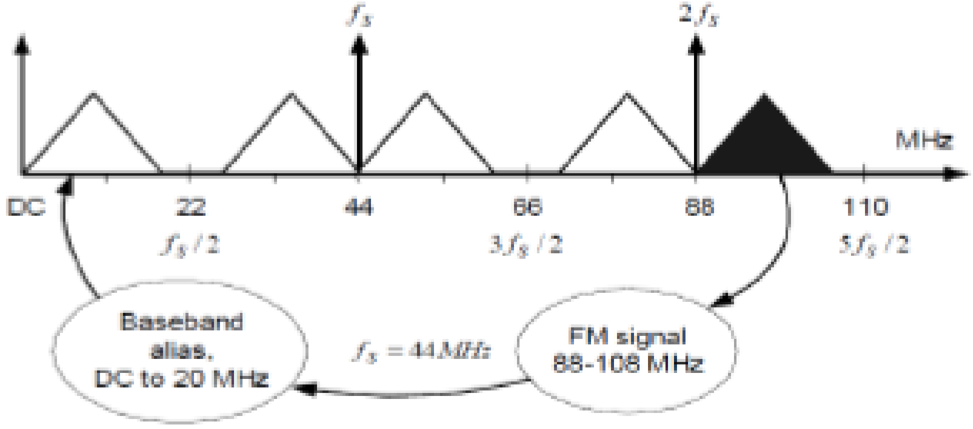


Figure 4.1: Design Example[1]

The fact that the 10.7 MHz signal is subsampled at 11.875 MHz generates a frequency shift, so we have our signal centered at 0 base band signal(typically 1.1875 Mhz). The products are in 14 bits precision and are passed by FIR (finite impulse response) low pass filters to eliminate the images from the mixing. The ADC serves as an input to the next stage, digital mixer. The other input being the signal fed from NCO. The NCO generates a sine signal with frequency same as that of the desired frequency 10.7MHz. For this , we take the 8 MSB bits from the ADC, and used them to address a 64KByte ROM (read only memory) in such a way that the 8 MSB bits of the ROM address correspond to the 8 MSB of sine signal with desired frequency.The value in ROM table corresponds to a sine signal. The ROM generates the phase $\varphi(t)$ of the demodulated signal. To get the frequency we approximate the derivative of the equation as a differences equation,

$$frequency(t) = d\varphi/dt \tag{4.2}$$

generating one sample delay and inverting it (equivalent to multiply for -1) and adding it with the next sample from the ROM. So at the output of the adder we have is the sine signal of desired frequency that is 107.5MHz The mixer is a multiplier that produces sum and difference frequencies, here we require only the difference frequency ie.,F$_{adc}$ - F$_{nco}$. This amounts to 0Hz or DC,thus the desired signal is transferred to baseband. The CIC filter and low pass FIR filter processes the desired signal and feeds them to DAC which converts the signal in digital domain into analog domain, making it suitable for sound card to reproduce the information.

Chapter 5

IMPLEMENTATION DETAILS

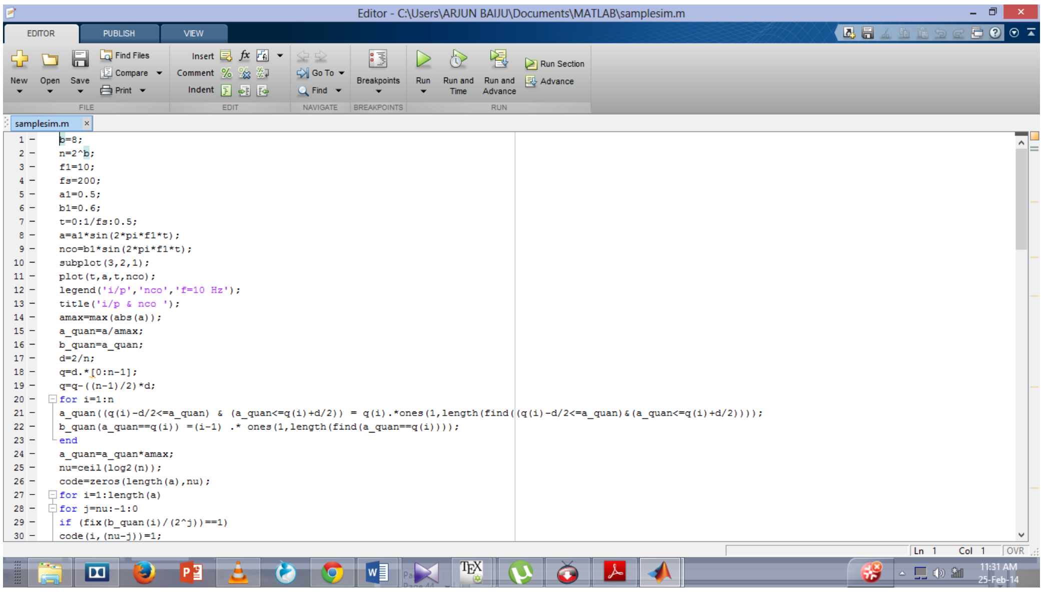
5.1 Platforms Used

5.1.1 MATLAB

The software used for the implementation of this system is MATLAB. MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in an easy to use environment where problems and solutions are expressed in a familiar mathematical notation. Typical use include Math and computation algorithm development, data acquisition modeling, simulation and prototyping data analysis ,exploration and visualization, scientific and engineering graphics, application development, including graphical user interface.

5.1.2 SIMULINK

Simulink is an environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code genera-



```
b=8;
n=2^b;
f1=10;
fs=200;
a1=0.5;
b1=0.6;
t=0:1/fs:0.5;
a=a1*sin(2*pi*f1*t);
nco=b1*sin(2*pi*f1*t);
subplot(3,2,1);
plot(t,a,t,nco);
legend('i/p','nco','f=10 Hz');
title('i/p & nco ');
amax=max(abs(a));
a_quan=a/amax;
b_quan=a_quan;
d=2/n;
q=d.*[0:n-1];
q=q-((n-1)/2)*d;
for i=1:n
a_quan((q(i)-d/2<=a_quan) & (a_quan<=q(i)+d/2)) = q(i).*ones(1,length(find((q(i)-d/2<=a_quan)&(a_quan<=q(i)+d/2))));
b_quan(a_quan==q(i)) =(i-1) .* ones(1,length(find(a_quan==q(i))));
end
a_quan=a_quan*amax;
nu=ceil(log2(n));
code=zeros(length(a),nu);
for i=1:length(a)
for j=nu:-1:0
if (fix(b_quan(i)/(2^j))==1)
code(i,(nu-j))=1;
```

Figure 5.1: Design Example[2]

tion, and continuous test and verification of embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement and test a variety of time-varying systems, including communications, controls, signal processing, video processing and image processing.

KEY FEATURES

•Extensive and expandable libraries of predefined blocks

• Interactive graphical editor for assembling and managing intuitive block diagrams

•Ability to manage complex designs by segmenting models into hierarchies of design components

• Model Explorer to navigate, create, configure, and search all signals, parameters, properties, and generated code associated with your model

•Application programming interfaces (APIs) that let you connect with other simulation programs and incorporate hand written codes

- MATLAB function blocks for bringing MATLAB algorithms into simulink and embedded system implementations
- simulation modes for running simulation interpretively or at complied c-code speeds using fixed or variable step solvers
- Graphical debugger and profiler to examine simulation results and diagnose performance and unexpected behavior in your design
- Full access to MATLAB for analyzing and visualizing results, customizing the modeling environment and defining signal, parameter and test data.

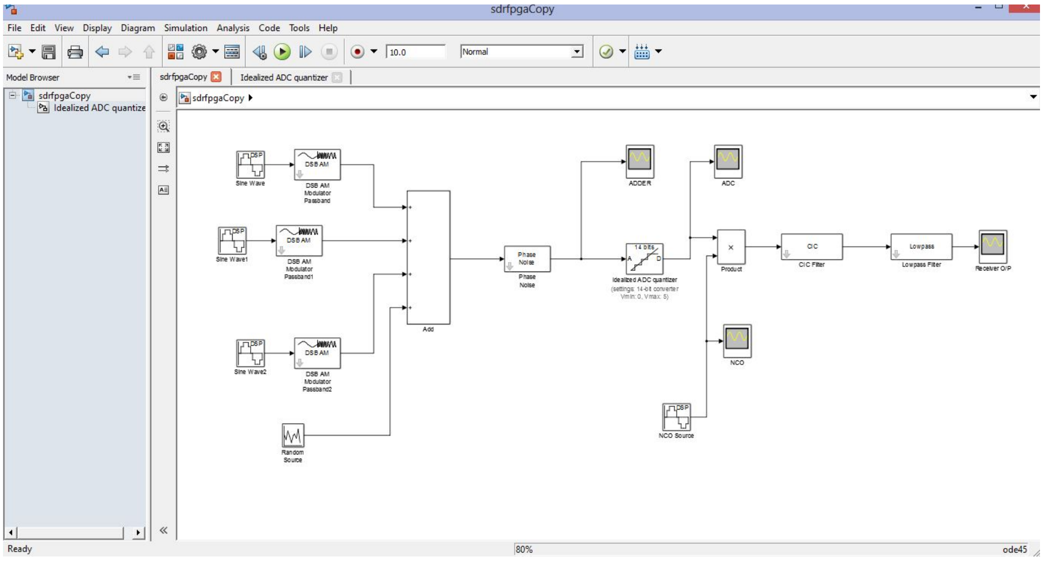


Figure 5.2: Design Example[1]

Chapter 6

RESULTS AND DISCUSSIONS

6.1 MATLAB Simulation Results

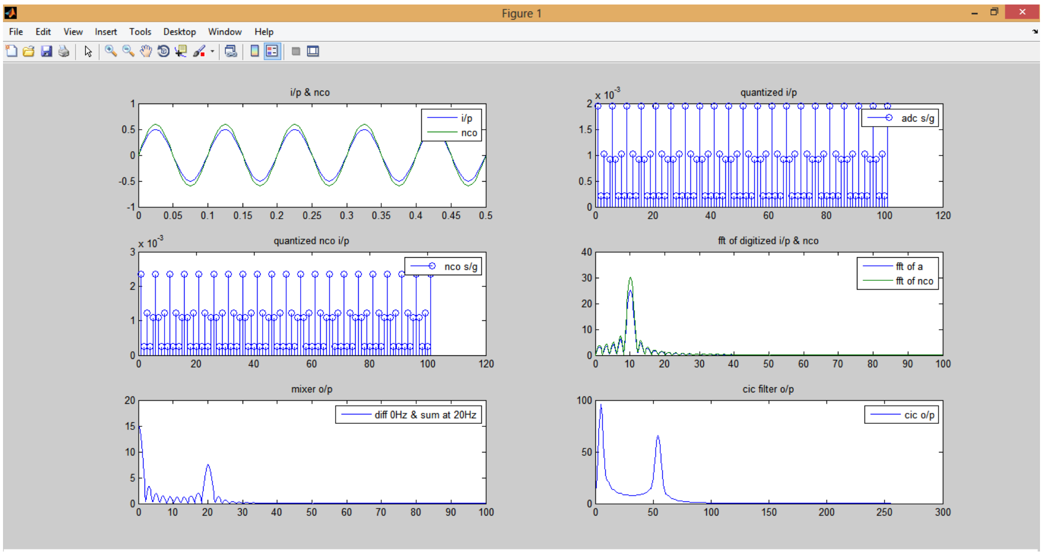


Figure 6.1: Simulation result in MATLAB

Figure shows the MATLAB simulation results of the different blocks.

Input and NCO: The input signal is to be mixed with the output from the

NCO block. Hence, the NCO signal simulated as shown above is a sine wave of frequency 1MHz. The input signal considered for this simulation is also a sine wave, this corresponds to a single audio frequency in the output.

Quantized Input and NCO : The analog signal is sampled and quantized to convert it to digital signal. The sampling frequency was chosen to satisfy the Nyquist criteria. fs$\geq$ 2fm The frequency chosen was 5MSPS.

Mixer Output: The digital mixer accepts the digitized input signal and digitized NCO signal as its inputs. The output consists of two frequencies which are the sum and difference of these two signals. Hence, the output shows a peak corresponding to 0 and 2 MHz.

6.2 SIMULINK Simulation Results

Simulink is an environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement and test a variety of time-varying systems, including communications, controls, signal processing, video processing and image processing.

The three DSB AM modulator used are defined with carrier frequency 10kHz, 11kHz and 12kHz. These are inputted to the Adder. The output of the adder is quantized using Idealized ADC Quantizer.The quantized output is mixed with a local oscillator output tuned to 10kHz.The mixer output is passed through a low pass filter to get the receiver output. The blockdiagram and simulation results are as shown below:

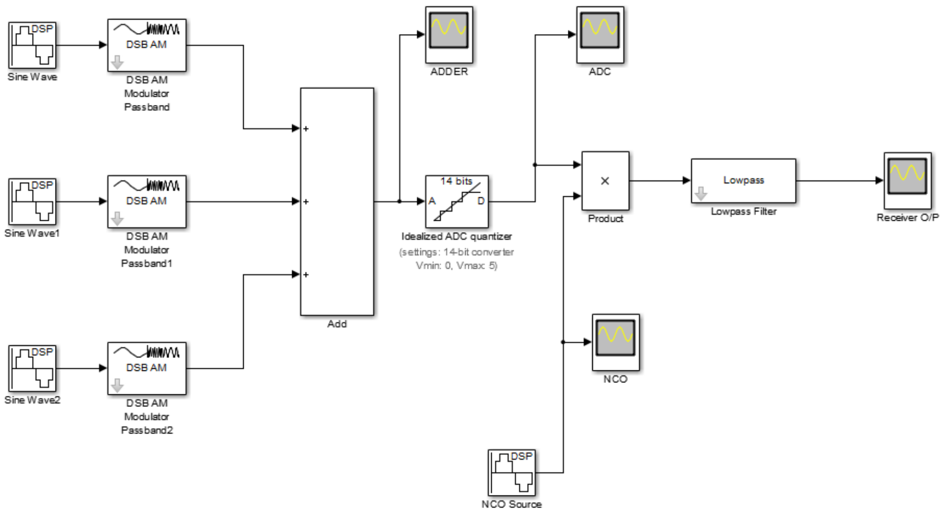


Figure 6.2: Simulation Block Diagram in Simulink

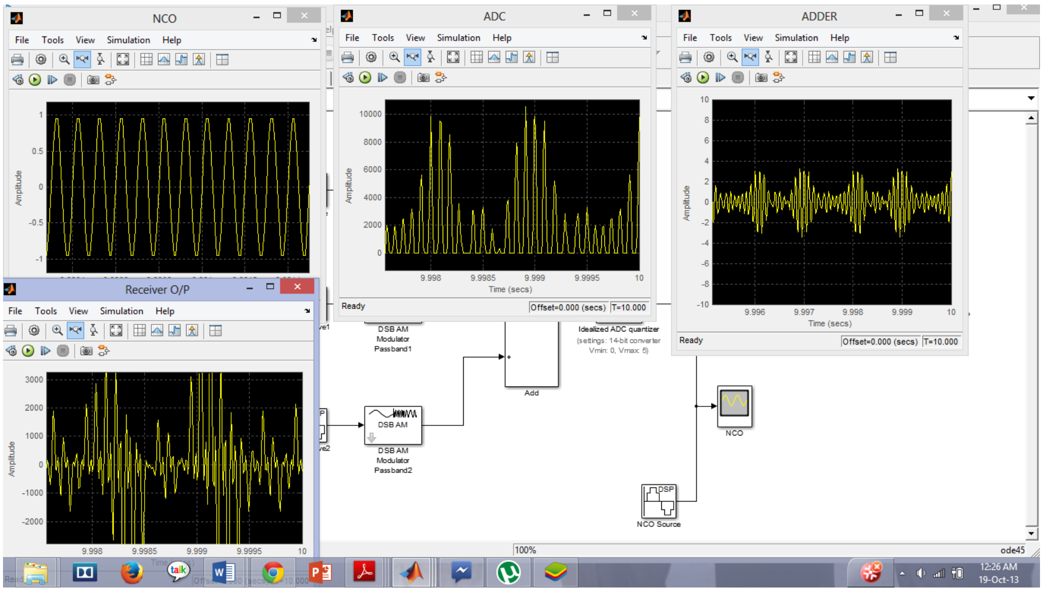


Figure 6.3: Simulation Output in Simulink[1]

6.3 MODELSIM Simulation Results

NCO: As the NCO block produces analog sine signal of the required frequency, ModelSim was reuired to view it.The Verilog code was written with

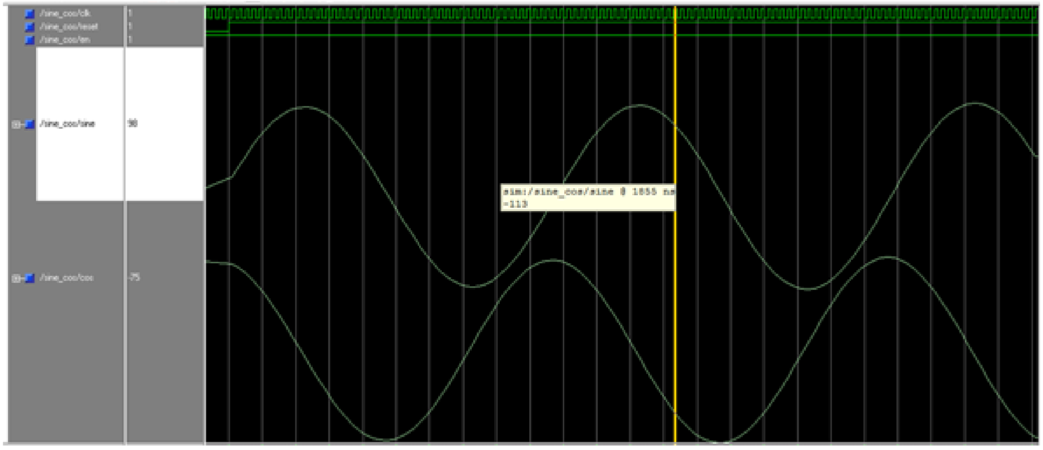


Figure 6.4: Simulation Output in Simulink[2]

2 input pins(clk,reset) and 2 output pins(sine,cosine):

Clk: Input signal set to 20MHz . 40 clock cycles corresponds to one cycle of sine or cosine wave output. Since AM stations correspond to a bandwidth of 500-1500kHz, the clock frequency was set as 20MHz. This will provide an output signal of frequency 500kHz.

Reset : Input signal tied to HIGH.

Sine[11:0]: Obtained sine wave of 500kHz frequency and amplitude 2 Vpp .

Cosine[11:0]: Obtained sine wave of 500kHz frequency and amplitude 2 Vpp.

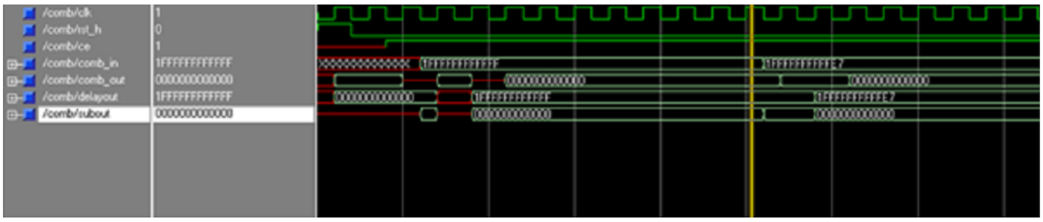


Figure 6.5: Simulation Output in Simulink[3]

The block was coded in Verilog hardware description language and simulated in Modelsim. The input and output signals are:
clk: The clock frequency was set as 20MHz, the same as that for the NCO bock
combin[11:0]: data from the FIR out 12 bit ce control.This controller uses both D cic and D fir to determine how many cycles of the global clock to count before its clock enable signal is output. Note that this controller counts only the global clock, it does not depend on the clock enable controller from the downsample module in the CIC filter.
enable control: This controller serves much the same purpose as that of the start ctrl controller shown in the fig. Namely, it outputs an enable signal to the ce control module signaling it when the first good sample has arrived. The major difference between start ctrl and enable control is that enable control counts the clock enable signal output from the downsampling module in the CIC filter instead of the global clock signal. This controller accounts for the latency introduced by the pipeline registers in the comb section of the CIC filter.
delayout[11:0]:The combin signal has been delayed.
combout[11:0]: The output bit width of the CIC filter and the bit width of the accumulators in the integrator stages.

6.4 Xilinx ISE Results

After verification in Modelsim , the Verilog code was run in Xilinx ISE, which generated the above waveforms: The input 12 bit signal is first converted to 24bit sign extended input. Registers were defined to store values of sample and hold states. The intermediate signals like mask, mod, DUMPON and PERIODN were also generated.

topmoduleop.png

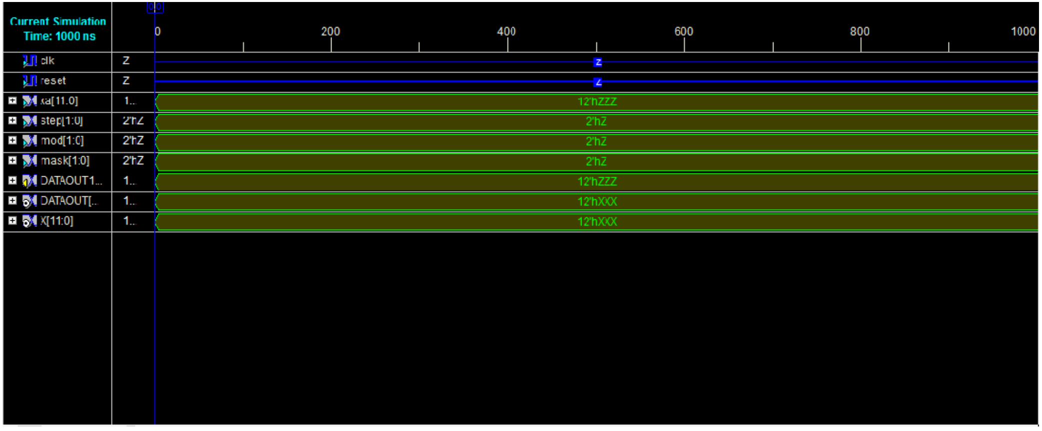


Figure 6.6: Matlab Simulation Result

Clk: input signal set to 20MHz . 40 clock cycles corresponds to one cycle of sine or cosine wave output. Since AM stations correspond to a bandwidth of 500-1500kHz, the clock frequency was set as 20MHz. This will provide an output signal of frequency 500kHz.

Reset input signal tied to HIGH.

Sine: Obtained sine wave of 500kHz frequency and amplitude 2 Vpp

Cosine: Obtained sine wave of 500kHz frequency and amplitude 2 Vpp

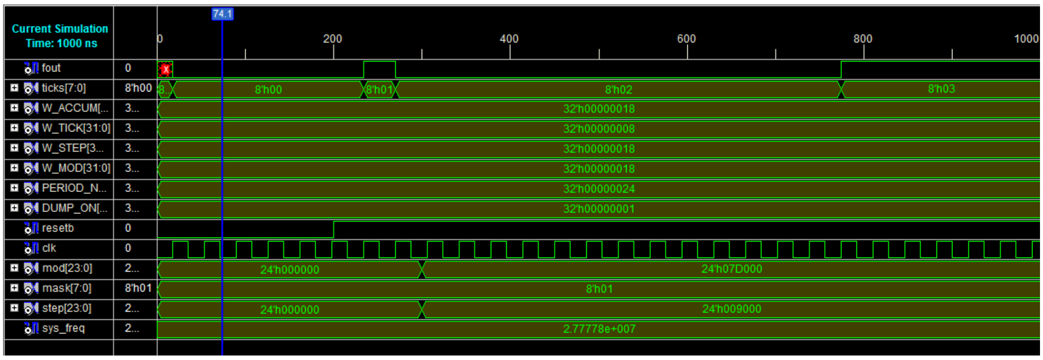


Figure 6.7: Matlab Simulation Result

Step, Mask and mod designed for desired frequency. Modulo and Step values were selected such that they would not be divisible. This is the 12bit output obtained from the top module. The top module is the first as per the coding structure, but the last module as per logic block and also order of completion of execution. Hence the final output is to be extracted from this module. This module calls the FIR module and tranfers the control to it. The control is further passed on to CIC, Mixer and NCO.

6.5 Xilinx Spartan 3 Output

After implementing the design in Xilinx ISE and generating the program file the target device,i.e. Xilinx Spartan 3E XC3S400 FPGA, was configured.The program was loaded into FPGA after performing boundary scan and bypassing PROM. FRC1 was input I/O bank and FRC2 was output I/O bank. The fout signal from the FPGA kit was given to a loudspeaker through a sound card. The audio output corresponding to the selected channel was produced.

Chapter 7

CONCLUSION AND FUTURE SCOPE

A software AM radio receiver was implemented using Xilinx Spartan 3E XC3S400 FPGA. Using this experimental board, AM audio signals were successfully demodulated and fed to a speaker. The block level design was verified by simulations in Simulink and Matlab. The code written in Hardware Description Language was verified using ModelSim . After performing a syntax check using Xilinx ISE, behavioural model simulation was also undertaken before loading the bit file into FPGA kit.

This design is not too large to fill a modern FPGA, so additional processing can be performed on the same device if desired. Summarizing, SDR is a promising technology that facilitates development of multi-band, multi-service, multi- standard, multi-feature consumer handsets.

As the nextstep of the project, implementatation of the RF front end also in software can be undertaken. Also, adding an AM/FM mode facility will display the multi-mode facility of SDR.A larger step would be to implement cognitive radio, yet another field in which a lot of research is in progress all round the world.

Work Division

Idea Development	Ann Mary George, Nithin Eldho Abraham
Feasibility Study	Neethu Susan Jose, Arjun B
Design	Ann MaryGeorge, Neethu Susan Jose
Ciruit Simulation	Arjun B, Nithin Eldho Abraham
Documentation and Preparation of Report	Arjun B, Ann Mary George

7.1 Work division

Bill of Materials

Sl. No	Item	Quantity	Cost
1	Sound Card	1	500
2	FPGA	1	4000
3	Speaker	1	100
4	ADC	1	35
5	DAC	1	35
	TOTAL		4670

7.2 Bill Of Materials

References

[1]*Model-based Software-defined Radio(SDR) Design Using FPGA* Anton S. Rodriguez, Michael C. Mensinger Jr., In Soo Ahn, and Yufeng Lu ,Department of Electrical and Computer Engineering,Bradley University, Peoria, Illinois 61625
[2]R.H. Hosking, *Digital Receiver Handbook: Basic of Software Radio*, Pentek Inc., www.pentek.com/products/GetLit.CFM/
[3]*Modular FPGA-Based Software Defined Radio for CubeSats* by Steven J. Olivieri, May 2011