CHAPTER 1

INTRODUCTION

1.1 Objective of the project

The main objective of our project is to configure an Audio Codec as per the requirements of the Software Defined Radio and implement it to an Frequency Modulator.

The project has been divided into three phases.

The first phase is to configuring the audio codec using VHDL codes and implement it on a SPARTAN6 FPGA.

The second phase is to convert the serial data coming out of the SPARTAN6 FPGA into parallel.

The third stage is to interface the SPARTAN6 FPGA to a VIRTEX4 FPGA to verify FM modulation and demodulation.

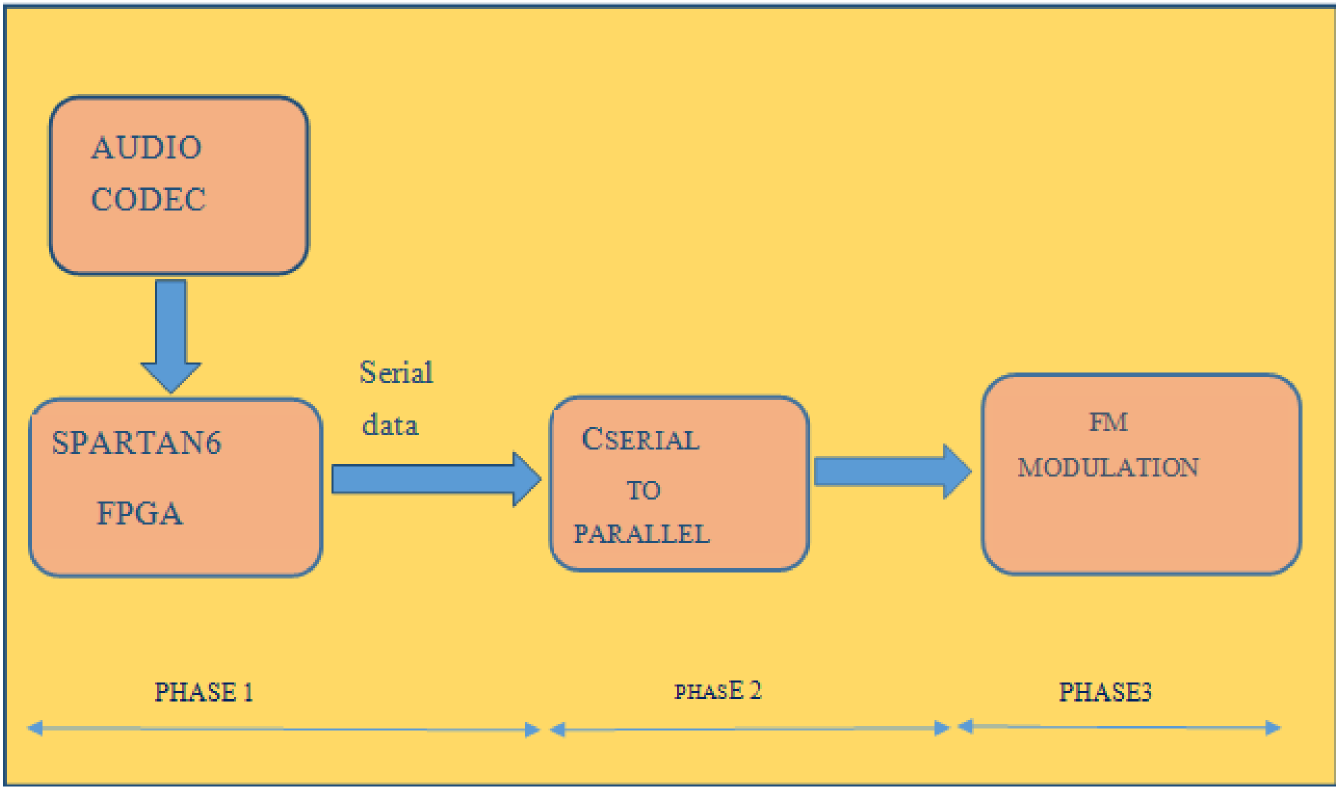
Phase one consists of the following:

- Analysing the data sheet of the TLV320AIC3107 Audio Codec and the required interfaces for programming and accessing the voice data needs to be built on the FPGA using VHDL program.
- Developing an I2C interface on the SPARTAN6 which is used to write configuration data into the registers of the Audio Codec.
- Verifying the protocol on ISIM simulator of Xilinx software
- Hardware implementation and verification of voice loopback

Phase two consist of the following:

- Conversion of serial data to parallel data to be fed into the FM modulator.

  Phase three consists of the following:
- Interfacing the SPARTAN6 FPGA with the VIRTEX4 FPGA to implement FM modulation.



1.2 Software Defined Radio

Software Defined Radio is a radio communication system where components that have been typically implemented in hardware are instead implemented by means of software. A basic SDR system may consist of a personal computer equipped with a sound card, or other analog-to-digital converter, preceded by some form of RF front end. Significant amounts of signal processing are handed over to the general-purpose processor, rather than being done in special-purpose hardware (electronic circuits). Such a design produces a radio which can receive and transmit widely different radio protocols (sometimes referred to as waveforms) based solely on the software used.

Software radios have significant utility for the military and cell phone services, both of which must serve a wide variety of changing radio protocols in real time.

A software-defined radio can be flexible enough to avoid the "limited spectrum" assumptions of designers of previous kinds of radios, in one or more ways including:

- Spread spectrum and ultra wideband techniques allow several transmitters to transmit in the same place on the same frequency with very little interference, typically combined with one or more error detection and correction techniques to fix all the errors caused by that interference.

- Software defined antennas adaptively "lock onto" a directional signal, so that receivers can better reject interference from other directions, allowing it to detect fainter transmissions.
- Cognitive radio techniques: each radio measures the spectrum in use and communicates that information to other cooperating radios, so that transmitters can avoid mutual interference by selecting unused frequencies.
- Dynamic transmitter power adjustment, based on information communicated from the receivers, lowering transmit power to the minimum necessary, reducing the near-far problem and reducing interference to others, and extending battery life in portable equipment.
- Wireless mesh network where every added radio increases total capacity and reduces the power required at any one node. Each node only transmits loudly enough for the message to hop to the nearest node in that direction, reducing near-far problem and reducing interference to others

1.3 Requirements of a Software Defined Radio

The Audio Codec has to be configured as per the specifications of the SDR. The specifications are as mentioned below

- PLL control bit has to be disabled
- BCLK is an input(slave mode)
- WCLK is an input (slave mode)
- Serial data bus uses I2S mode
- Audio data word length is 16bits
- Continuous-transfer mode used to determine master mode bit clock rate, etc..

1.4 Operating principle

The ideal receiver scheme would be to attach an analog-to-digital converter to an antenna. A digital signal processor would read the converter, and then its software would transform the stream of data from the converter to any other form the application requires.

An ideal transmitter would be similar. A digital signal processor would generate a stream of numbers. These would be sent to a digital-to-analog converter connected to a radio antenna.

The ideal scheme is not completely realizable due to the actual limits of the technology. The main problem in both directions is the difficulty of conversion between the digital and the analog domains at a high enough rate and a high enough accuracy at the same time, and without relying upon physical processes like interference and electromagnetic resonance for assistance.

Receiver architecture

Most receivers use a variable-frequency oscillator, mixer, and filter to tune the desired signal to a common intermediate frequency or baseband, where it is then sampled by the analog-to-digital converter. However, in some applications it is not necessary to tune the signal to an intermediate frequency and the radio frequency signal is directly sampled by the analog-to-digital converter (after amplification).

Real analog-to-digital converters lack the dynamic range to pick up sub-microvolt, nano watt-power radio signals. Therefore a low-noise amplifier must precede the conversion step and this device introduces its own problems. For example, if spurious signals are present (which is typical), these compete with the desired signals within the amplifier's dynamic range. They may introduce distortion in the desired signals, or may block them completely. The standard solution is to put band-pass filters between the antenna and the amplifier, but these reduce the radio's flexibility. Real software radios often have two or three analog channel filters with different bandwidths that are switched in and out