PROJECT NO. 25

Hardware Designing and Verification of Three Phase to Sequence Decomposer

A Project Report

Submitted in partial fulfilment of the

requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

Submitted By:

Aishwarya Mittal **Kakarla Uday Kanth Reddy** **Shubham Chowdhary**

Guided By:

Dr. Vishal Kumar

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE- 247667 (INDIA)

APRIL 2018

CANDIDATE'S DECLARATION

We hereby certify that the work which is presented in this project entitled "Hardware Designing and Verification of Three Phase to Sequence Decomposer" in the partial fulfilment for the award of the degree of Bachelor of Technology (Electrical) submitted to the Department of Electrical Engineering, IIT Roorkee, is an authentic record of our own, work carried out during August 2017 to April 2018 under the guidance of Dr. Vishal Kumar, Professor, Department of Electrical Engineering, IIT Roorkee.

Aishwarya Mittal	Kakrala Uday Kanth Reddy	Shubham Chowdhary
En. No. 14115008	En. No. 14115055	En. No. 14115115

This is to certify that above declaration is true to the best of my knowledge.

Dr. Vishal Kumar

Professor

Department of Electrical Engineering

IIT Roorkee

Acknowledgement

We would like to give our sincere thanks to Dr. Vishal Kumar, Professor, Department of Electrical Engineering, for his constant support and guidance during the course of this project. His constant contribution in stimulating suggestions and encouragement helped us in coordinating our project, especially in writing this report.

We also express our sincere gratitude and appreciations to our peers and people who have willingly helped us out with the best of their abilities during the course of this project.

Abstract

In power systems, the analysis of multi-phase systems generally takes place with the help of sequence components. A balanced system can always be analyzed on a per-phase basis, but to account for various other cases such as control and operation of grid-connected devices (thorough positive sequences), protection and control, determining the type and location of a fault appearing in a power Network (thorough negative sequences), sequence component-wise analysis is irrefutable. Also, many applications require extracting sequence components and tracking them in real time.

Hence such systems, (also known as the Synchronization Systems), are needed with better dynamic response. To meet the above needs, recently various Phase Measurement Units (PMUs) have also been developed. In addition, IEEE C37.118-2005 synchrophasor standard has been developed to approve the performance of such devices built or proposed. The tests on the recently developed PMUs show a failure in meeting the dynamic performance standards. This failure was the sole motivation behind the development of the SDSS (Sequence Decomposer Synchronization System).

The design claims not only to pass the high frequency ramp tests, but also to work for isolated power systems, working at very high base frequencies, like the Isolated Aerospace systems (at 360-800Hz). It also claims to have wide frequency range, good dynamic response to transients, work even when harmonic distortions, wideband noise and DC-offsets are present.

Moreover, the filtering has been made frequency adaptive, which led to normalized frequency-based FIR filter designs. This model would split the 3-phase signals to their corresponding sequence components, calculate their corresponding phase, amplitude and frequencies and that to within 2 clock cycle delay. All of this happens independent of the input signal frequency.

Moreover, to improve the speed and accuracy in terms of parallel and digital signal processing, the entire system is realized on FPGA setup.

Table of Contents

List of Figures

List of Tables

List of Symbols

Va = Phase a Input Voltage

Vb = Phase b Input Voltage

Vc = Phase c Input Voltage

Vpos = Positive Sequence Component

Vneg = Negative Sequence Component

Vzero = Zero Sequence Component

Fs = Sampling Frequency

Fin = Fundamental Frequency of Input Signal

k = Downsampling Rate

β = Phase delay between samples after downsampling

N= Number of samples per cycle after downsampling

Chapter 1

Introduction

1.1 Sequence Components

A system of three unbalanced phasors can be resolved in the following three symmetrical components:

1. **Positive Sequence:** A balanced three-phase system with the same phase sequence as the original sequence.
2. **Negative sequence:** A balanced three-phase system with the opposite phase sequence as the original sequence.
3. **Zero Sequence:** Three phasors that are equal in magnitude and phase.

Figure 1.1a depicts a set of three unbalanced phasors that are resolved into the three sequence components mentioned above. In this the original set of three phasors are denoted by Va , Vb and Vc , while their positive, negative and zero sequence components are denoted by the subscripts 1, 2 and 0 respectively. This implies that the positive, negative and zero sequence components of phase-a are denoted by Va1, Va2 and Va0 respectively. Note that just like the voltage phasors given in Fig. 1 we can also resolve three unbalanced current phasors into three symmetrical components.

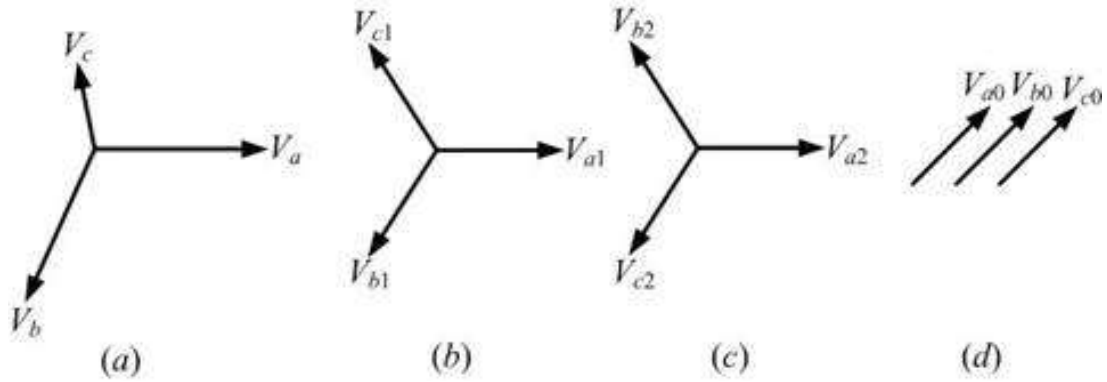

Figure 1.1a: Representation of (a) an unbalanced network, its (b) Positive Sequence, (c) Negative Sequence and (d) Zero Sequence

1.2 Objective

The objective of this project is to develop a hardware model of 3-Phase to sequence decomposer. It will be useful in relaying circuits for faster action on any faults. The model will be implemented on a FPGA board using Verilog HDL. The input analog signal is first converted into digital signal by using ADC (Analog to Digital Converter) which can be done using the inbuilt ADC on FPGA board. A digital circuit is implemented which takes the 3-phase discrete signals as input and gives the discrete values of Positive, Negative and Zero sequence components. Finally, the sequence values can be used for the identification of the unbalanced and balanced faults (L-G, L-L, L-L-G and L-L-L) and sends the control signals to respective relays.

Although the conventional method of calculating sequence components can also be used in digital circuits but it results in phase delay of 240° i.e. delay of 2/3 time periods. eq. 1 and 2 gives conventional method of positive phase sequence calculations. Where T_n is the fundamental frequency period.

$$3V_{a1} = V_a + V_b e^{-j240^0} + V_c e^{-j120^0} - (1.1)$$

$$3V_{a1}(t) = V_a(t) + V_b\left(t - \frac{2T_n}{3}\right) + V_c\left(t - \frac{T_n}{3}\right) - (1.2)$$

Reference [2] outlines the derivation of the Fast Retrieval Technique. For this technique, the sequence components are extracted from a three phase discrete input signal. The sequence components are extracted as a function of the three input phases (Va, Vb, Vc) based on the following discrete time formulas:

$$3V_+[k] = V_a[k] + P_b V_b[k] - Z_b V_b[k-1] - P_c V_c[k] + Z_c V_c[k-1] - (1.3)$$

$$3V_-[k] = V_a[k] - P_c V_b[k] + Z_c V_b[k-1] + P_b V_c[k] - Z_b V_c[k-1] - (1.4)$$

$$3V_0[k] = V_a[k] + V_b[k] + V_c[k] - (1.5)$$

Where Pb, Pc, Zb, and Zc are coefficients dependent on the sampling rate and the fundamental frequency. By maintaining a fixed ratio of the sampling and fundamental frequency, the SCE

coefficients do not have to be adapted, and equations (1.3), (1.4) and (1.5) are easily implemented in hardware. The frequency ratio is kept constant using the ADS rate, which is used to downsample the discretized three phase input. The ADS rate is computed based on the following formula:

$$ADS\ rate = k\ = round\left(\frac{F_s}{N.F_{in}}\right) - (1.6)$$

Where Fs is the A/D sampling frequency, N represents a fixed number of ADS samples recorded in one fundamental cycle and Fin is the fundamental frequency estimation. The ADS rate is fed back into the Adaptive Downsampler block, which generates a downsampled clock denoted as dclk.

So the fast retrieval method with adaptive downsampling gives a fast and frequency adaptive method of sequence components extraction which best suited for our objective.

Chapter 2

Software Tools and Hardware Platform

2.2 Software Tools

2.2.1 MATLAB Simulink

Simulink (Simulation and link) is developed by MathWorks as an add-on with MATLAB. It is a graphical programming language which offers modelling, simulation and analyzing of multi domain dynamic systems under Graphical User Interface (GUI) environment. The Simulink have tight integration with the MATLAB environment and have a comprehensive block libraries and toolboxes for linear and nonlinear analyses. The system models can be so easily constructed via just click and drag operations. The Simulink comes handy while dealing with control theory and model based design. In this project Simulink is used for the development and verification of initial model of sequence decomposer.

2.2.2 Xilinx ISE Design Suite 14.7

Xilinx ISE (Integrated Synthesis Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Xilinx ISE is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. The Xilinx ISE is primarily used for circuit synthesis and design.

2.2.3 ModelSim Simulator

ModelSim is an easy-to-use yet versatile VHDL/(System)Verilog/SystemC simulator by Mentor Graphics. It supports behavioral, register transfer level, and gate-level modeling. ModelSim supports all platforms used here at the Department of Pervasive Computing (i.e. Linux, Solaris and Windows) and many others too. While in ISim (Inbuilt simulator in Xilinx ISE Design Suite) analog signals waveform cannot be visualized ModelSim is perfect tool for visualizing analog signal waveforms.

2.2.4 Xilinx ChipScope Pro

Simulation based method is widely used for debugging the FPGA design on computers. Time required for simulating complex design for all possible test cases becomes prohibitively large and simulation approach fails. For rapid testing, such designs can be loaded on to the target FPGAs and tested by applying test inputs and directly observing their outputs. As the complexity of the design under test increases, so does the impracticality of attaching test equipment probes to these devices under test. The ChipScope Pro tools integrate key logic analyzer and other test and measurement hardware components with the target design inside the FPGA. Computer based software tool communicate with these hardware components and provide a designer robust logic analyzer solution.

2.3 Hardware Platform

2.3.1 Spartan-3E FPGA Evaluation Platform

The Spartan-3E family of Field-Programmable Gate Arrays (FPGAs) is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The Spartan-3E family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs. Each Spartan 3e consists of two inbuilt ADC and DAC.

2.3.2 Virtex-5 LXT FPGA ML505 Evaluation Platform

The hardware platform used for verification of the design on chip is virtex-5 family FPGA from Xilinx Corporation as shown in figure 2.2b.

The Virtex®-5 LXT ML505 is a general purpose FPGA and RocketIO™ GTP development board

- Provides feature-rich general purpose evaluation and development platform
- Includes on-board memory and industry standard connectivity interfaces
- Delivers a versatile development platform for embedded applications

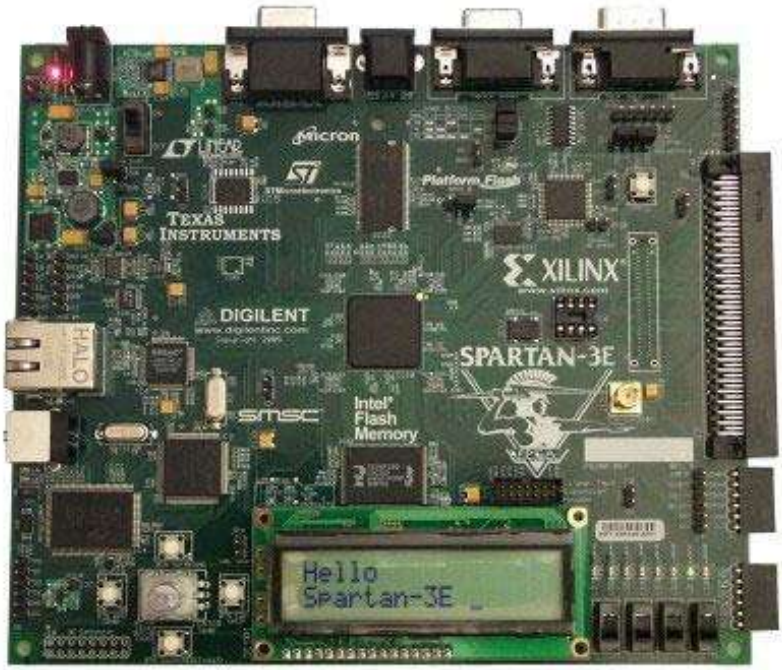

Figure 2.2a. Spartan-3E FPGA Evaluation Platform

Figure 2.2b. Virtex-5 LXT FPGA ML505 Evaluation Platform

Chapter 3

Filtering: Least-Squares and Normalized Frequency

3.1 Normalized Frequency

In general, any filter frequency characteristics is a plot of frequency vs the gain. But, when there is a need to design a frequency adapted filter, clearly such a notion cannot be used. Hence the concept of normalized frequency is used.

Normalized frequency is a unit of measurement of frequency equivalent to cycles/sample. In digital signal processing (DSP), the continuous time variable, t, with units of seconds, is replaced by the discrete integer variable, n, with units of samples. More precisely, the time variable, in seconds, has been normalized (divided) by the sampling interval, T (seconds/sample), which causes time to have convenient integer values at the moments of sampling. This practice is analogous to the concept of natural units, meaning that the natural unit of time in a DSP system is samples.

Some programs (such as MATLAB) that design filters with real-valued coefficients use the Nyquist frequency (fs/2) as the normalization constant. The resultant normalized frequency has units of half-cycles/sample or equivalently cycles per 2 samples.

The following table shows examples of normalized frequencies for a 1 kHz signal, a sample rate fs = 44.1 kHz, and 3 different choices of normalized units. Also shown is the frequency region containing one cycle of the discrete-time Fourier transform, which is always a periodic function.

Units	Domain	Computation	Value
cycles/sample	[-½, ½] or [0,1]	1000 / 44100	0.02268
half-cycles/sample	[-1,1] or [0,2]	1000 / 22050	0.04535
radians/sample	[-π,π] or [0,2π]	2 π 1000 / 44100	0.1425

Table 3.1. Conversions from frequency to normalized frequency

3.2 FIR Designing: Least Squares

The weighted least squares method is implemented in a matlab function “firls”. The logic behind it is discussed here.Let the FIR filter length be L+1 samples, with L even, and suppose we'll initially design it to be centered about the time origin (``zero phase''). Then the frequency response, when there is even symmetry in impulse response and right-shift of L/2 samples(for a linear phase filter), is given on our frequency grid ωk by

$$H(\omega k) = h_0 + 2\sum_{n=1}^{L/2}(h_n \cos(\omega k\, n)) - (3.1)$$

for k = 0,1,2,…,N-1 (digitally)

or in matrix form:

$$\underbrace{\begin{bmatrix} H(\omega 0) \\ H(\omega 1) \\ \vdots \\ H(\omega N-1) \end{bmatrix}}_{d} = \underbrace{\begin{bmatrix} 1 & 2\cos(\omega 0) & \dots & 2\cos\left(\omega 0\left(\frac{L}{2}\right)\right) \\ 1 & 2\cos(\omega 1) & \dots & 2\cos\left(\omega 1\left(\frac{L}{2}\right)\right) \\ \vdots & \vdots & & \vdots \\ 1 & 2\cos(\omega N-1) & \dots & 2\cos\left(\omega N-1\left(\frac{L}{2}\right)\right) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h0 \\ h1 \\ \vdots \\ \frac{hL}{2} \end{bmatrix}}_{h} - (3.2)$$

So, in matrix form filter-design problem is restated as,

Min for h ($||\mathbf{A}h-d||_2^2$) –(9)

The solution for this can be found using gradient descent and hence solving for h as,

$h = [(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T]d, -(10)$

where $[(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T]$ is the left pseudo-inverse of **A**.

The h calculated here is used in FIR least squares filters for designing required filter characteristics.

Chapter 4

The Modules: Their Implementation

4.1 Simulink Model for Sequence Decomposer

We have first implemented a MATLAB Simulink model for finding the value of system parameters and checking the validity of fast retrieval method. Figure 4.1a. gives the Simulink model of sequence decomposer where Va, Vb and Vc are three phase discrete voltage sources. This Simulink model implements 3 major blocks ADS Block, SCE Block and DZCPD block. The working of these blocks is given below.

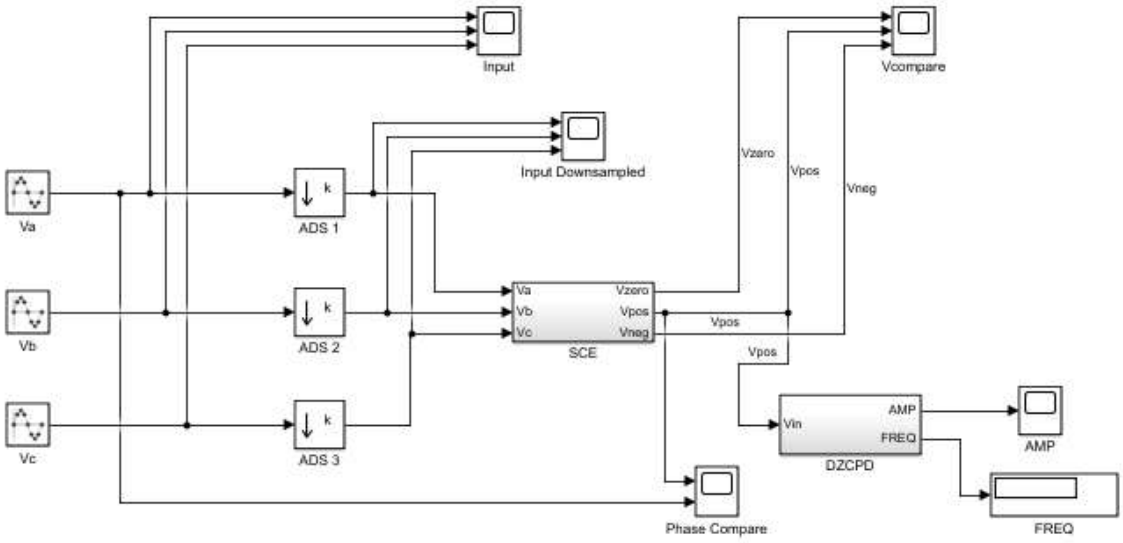

Figure 4.1a. Sequence Decomposer Simulink Model

4.1.1 ADS (Adaptive Down Sampler) Block

ADS Block uses Simulink Downsampler block for downsampling the input sinusoidal samples to get N fixed number of samples per cycles in downsampled waveform. It downsamples the input waveform by the factor k (k=round(fs/(N*fin)), where fs is the sampling frequency and fin is the input frequency, N=32).

4.1.2 SCE (Sequence Component Extractor) Block

Figure 4.1b. gives Simulink implementation of SCE block. SCE block implements the fast retrieval algorithm for extracting sequence components by taking downsampled phase voltages (Va, Vb and Vc) as inputs and generating sequence components (Vpos, Vneg and Vzero) as outputs. The value of constants Pb, Pc, Zb and Zc is calculated from beta (β=2π/N, which is phase delay for 1 sample). Then equation 1.3, 1.4, 5 are implemented using discrete transfer function.

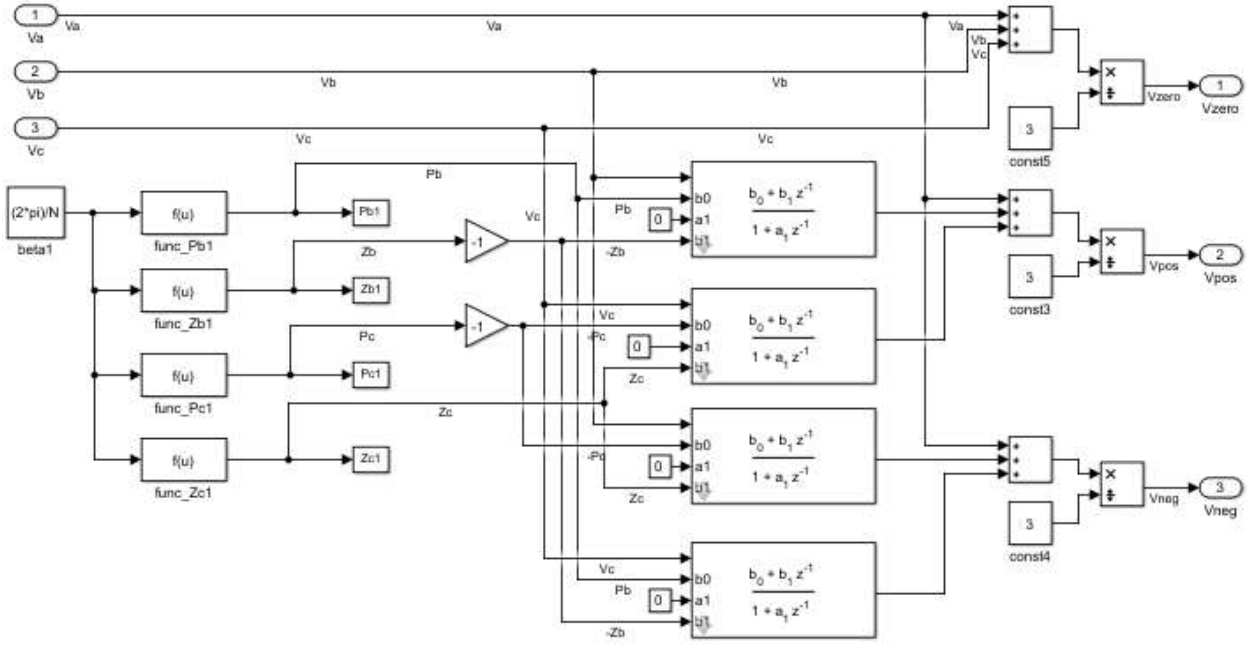

Figure 4.1b. SCE Block

4.1.3 DZCPD (DC compensated zero-crossing and peak detection) Block

Figure 4.1c. gives DZCPD Block in Simulink. It takes a sequence component as input and gives amplitude and frequency as output. Moving maxima and Moving minima Simulink blocks are used to calculate positive and negative peaks. Offset is calculated by taking average of positive and negative peaks and subtracted from input voltage to nullify effect of DC offset.

Figure 4.1d. gives the subsystem for frequency measurement which uses a counter to calculate number of samples between two zero crossing of input waveform to calculate frequency.

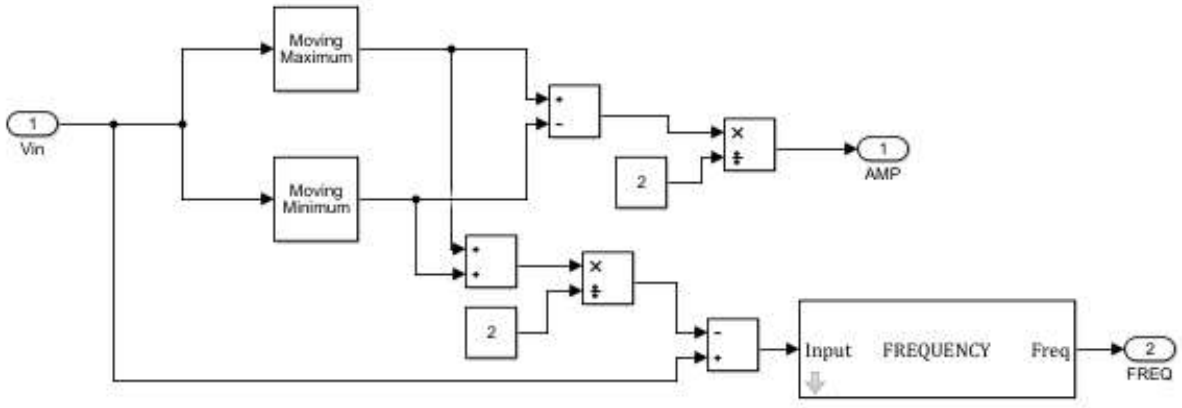

Figure 4.1c. DZCPD Block

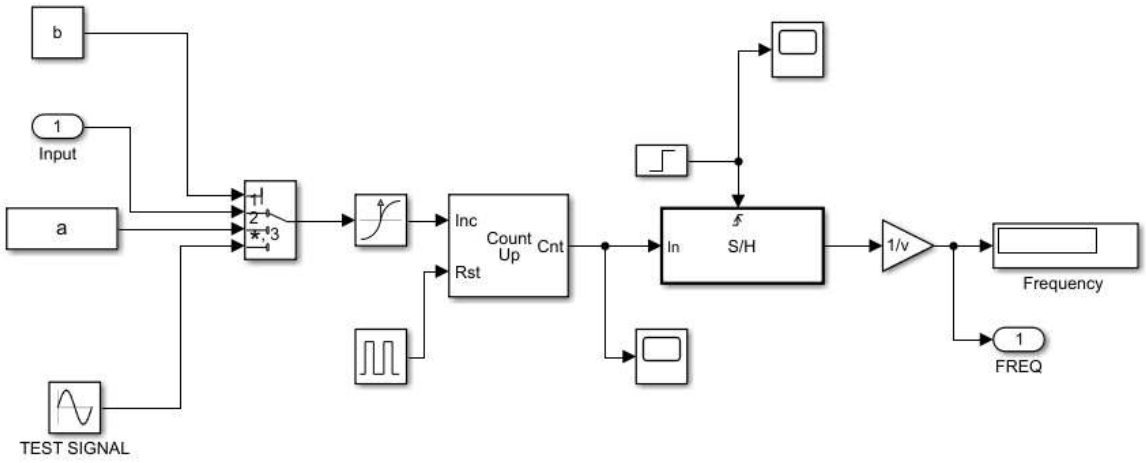

Figure 4.1d. Frequency measurement subsystem

4.2 HDL Implementation of Sequence Decomposer

The HDL model of Sequence Decomposer is implemented in Verilog HDL using Xilinx ISE for Vertex-5 FPGA Board. The Sequence Decomposer module consists of three major Modules. They are:

1. DC Offset Removal Module
2. ADS Module
3. SCE Module
4. DZCPD Module for each sequence output i.e. positive, negative and zero.
5. FIR Filter Module

The parameters are initialized as:

1. Sampling Frequency Fs = 8KHz i.e. Ts = 125µs.
2. No. of samples per cycle after down sampling N = 32.

ADS Rate (Adaptive Down sampling rate k):

ADS tells the no. of samples of A/D converter for which one sample is considered. For example, ADS = 5 implies for every 5 samples taken by the A/D converter one sample has to be considered i.e. the last sample of every 5 samples is considered. In this way the sampling frequency is controlled and helps in maintaining a constant phase difference between the samples even if the input frequency is varied. Now,

$$360/N = k*360*Fin/Fs$$

Therefore,

$$k = Fs/(Fin*N) \quad (3.3)$$

Figure 4.2b gives the RTL schematic of SEQ DECOMPOSER module which describes the flow of signals between the modules. Here DC-offset removal module and FIR filter module connected three times for 3 Input phases i.e. Vphase= Va, Vb and Vc. Similarly DZCPD Module is connected thrice for each sequence component i.e. Vseq= Vpos, Vneg and Vzero.

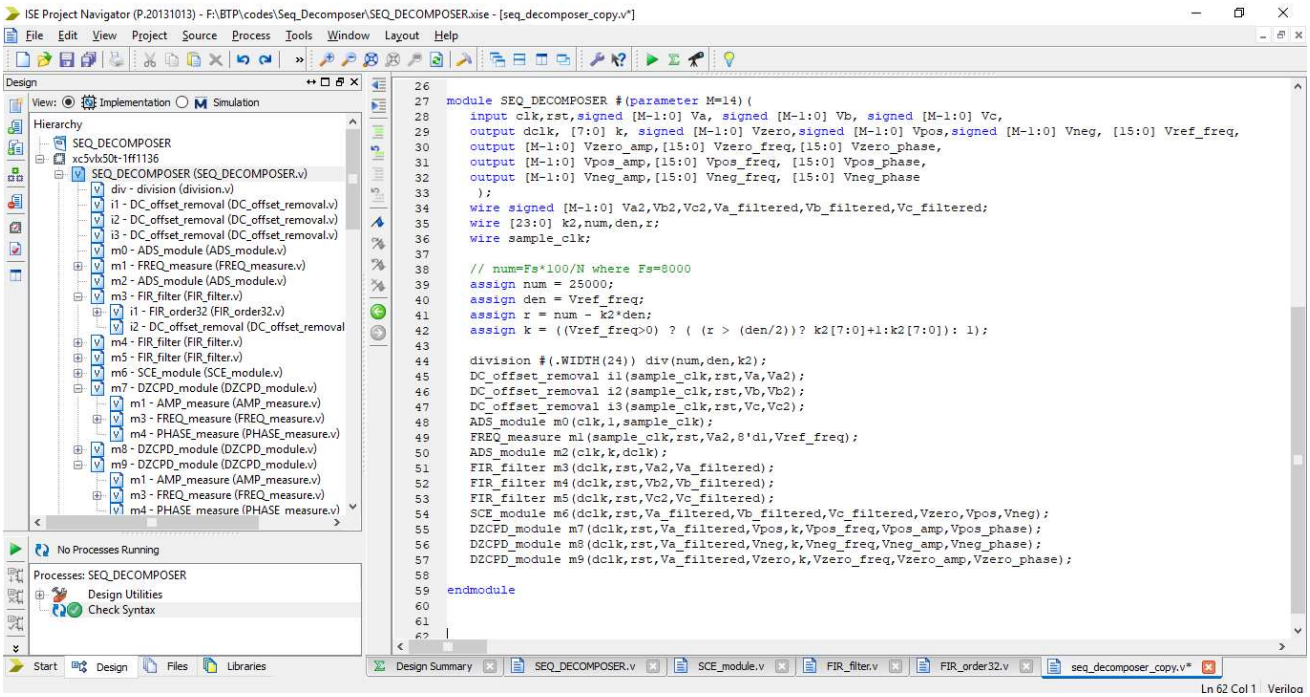

Figure 4.2a. Sequence Decomposer Verilog module

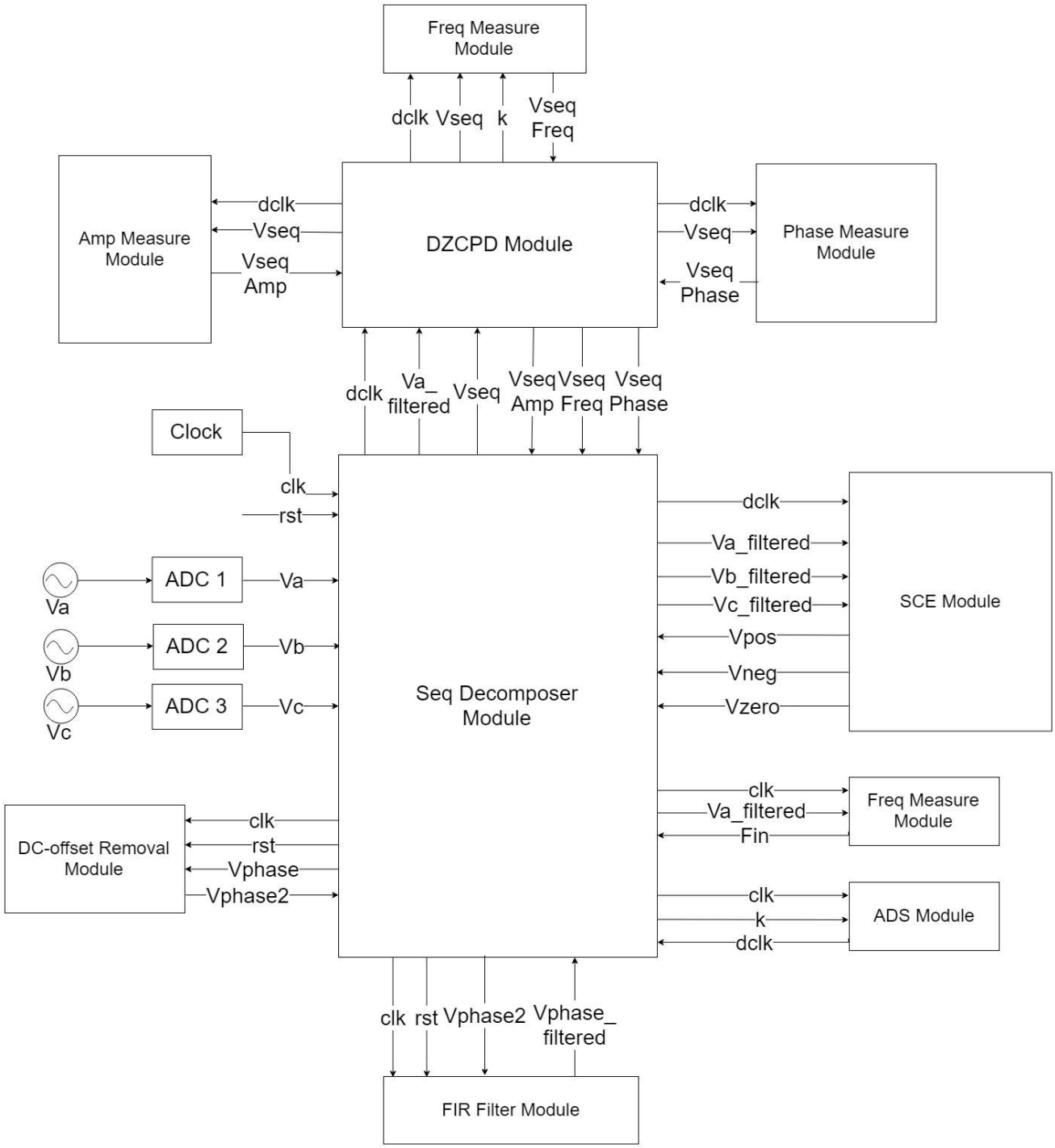

Figure 4.2b. Schematic Diagram of Sequence Decomposer

The inputs and outputs of sequence decomposer are described below:

a. Inputs:

a) Va, Vb, Vc: 3-phase adc samples of 14 bits

b) clk: Input clock

c) rst: reset signal

b. Outputs:

a) k: downsampling rate

b) dclk: downsampled clock

c) Vref_amp: amplitude of reference voltage i.e Va

d) Vref_freq: reference frequency in 100*(frequency in Hz)

e) Vpos: Output positive phase voltage with amplitude Vpos_amp, frequency Vpos_freq, phase Vpos_phase

f) Vneg: Output negative phase voltage with amplitude Vneg_amp, frequenc Vneg_freq phase Vneg_phase

g) Vzero: Output zero phase voltage with amplitude Vzero_amp, frequency Vzero_freq, phase Vzero_phase

The 5 major modules of SEQ_DECOMPOSER module are described below:

4.2.1 DC-offset Removal Module:

DC-offset removal module removes any dc-offset present in the input. It implements a 4 state FSM with states s0, s1, s2, s3. The State diagram is given in figure **4.2c**. Vin is the current input voltage while Vin_prev is input voltage in previous cycle. This module maintains to variables max_val (positive peak voltage in a cycle) and min_val (negative peak voltage in a cycle). The description of each state is given below:

a. s0: The state remains s0 until zero crossing of Vin from –ve to +ve i.e. start of positive half cycle.

b. s1: In state s1 max_val is updated with the +ve peak of input voltage. State goes to next state s2 on zero crossing of input from +ve to –ve.

c. s2: In state s2 min_val is updated with the -ve peak of input voltage. State goes to next state s3 on zero crossing of input from -ve to +ve.

d. s3: In state s4 offset value is calculated as (max_val+min_val)/2 and this offset value is subtracted from input voltage to give the output i.e. Vout=Vin-offset.

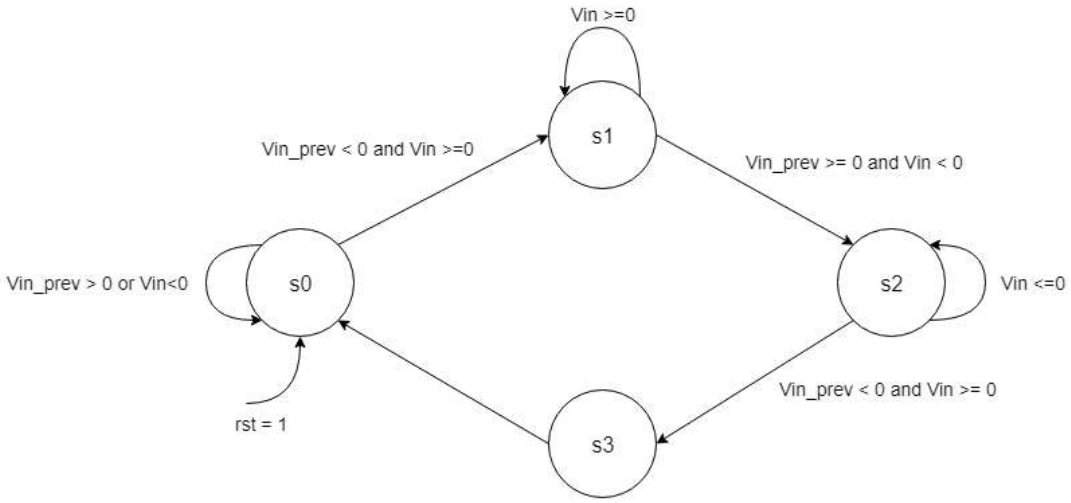

Figure 4.2c. State diagram of DC-offset removal module FSM

4.2.2 ADS Module

The ADS Module downsamples the input clock by ADS rate k to pass fixed number of samples per cycle to SCE Module. The Downsampled rate k is calculated in SEQ_DECOMPOSER module based on input signal frequency k = Fs/(Fin*N) where FREQ_Measure module calculates the frequency of the input(Fin). It implements a counter which changes the value after k cycles so that it produces a downsampled clock.

The inputs and outputs of ADS module are described below:

a. Inputs:

a) clk = Input clock

b) k = Downsampling rate

b. Outputs:

a) dclk = Downsampled clock

4.2.3. SCE Module (Sequence Component Extractor)

SCE Module takes 3-Phase downsampled input signals Va, Vb and Vc as inputs and computes sequence components Vpos, Vneg and Vzero as outputs in just one sample delay using fast retrieval method. The values of constants Pb, Pc, Zb and Zc are fixed for a particular value of N. Three registers Va1, Vb1 and Vc1 stores the value of Va, Vb and Vc at one sample delay. Sequence components are calculated using equations 1.3,1.4 and 1.5 as following:

1. Vpos = (Va*cf+Pb*Vb+Zc*Vc1)- (Zb*Vb1+Pc*Vc)/3000
2. Vneg = (Va*cf+Zc*Vb1+Pb*Vc)- (Pc*Vb+Zb*Vc1)/3000
3. Vzero = (Va+Vb+Vc)*cf/3000

Here cf=1000 is used as the multiplication factor as Pb, Pc, Zb and Zc are multiplied 1000 times to consider 3 significant figures after decimal point. The Inputs and outputs of SCE module are given below.

a. Inputs:

a) clk = Input clock

b) rst = Reset signal

c) Va = Phase a input voltage signal

d) Vb = Phase b input voltage signal

e) Vc = Phase c input voltage signal

b. Outputs:

a) Vpos = Positive sequence component

b) Vneg = Negative sequence component

c) Vzero = Zero sequence component

4.2.4 FIR-Filter Module

Since higher order harmonics may be present in the input phase voltage signal an FIR-Filter (FIR: Finite Impulsive Response) is implemented to remove these higher order harmonics and give the filtered output signal consisting of only fundamental component.

The typical characteristics to be designed looks like a low pass filter at the beginning. Also, with a closer look into a working block diagram of the filter is as follows:

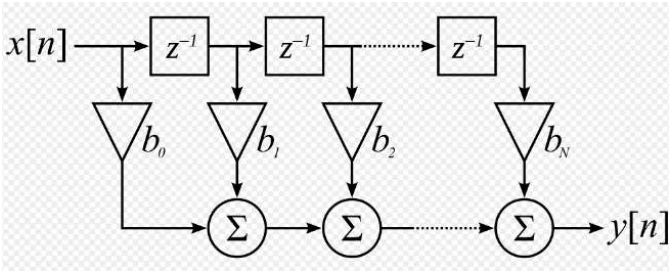

Figure 4.2d: FIR Working Block Diagram

This block model in discrete domain would facilitate the Verilog designing of the filter. Once the above amplitude and phase adapted model is designed, the most crucial part for a frequency independent SDSS would be a frequency adapted network designing. This designing can be facilitated by detecting frequency too. But the input signal includes harmonics too. This problem is solved by using this kind of low pass filter in our network to improve its performance.

This structure in the digital domain is simply a time-based weighted averaging circuit. This can be implemented by a simple Verilog behavioral code, using the following expression:

$$h[n] = \sum_{i=0}^{N} b_i . \delta[n-i] = \begin{cases} b_n & 0 \le n \le N \\ 0 & otherwise \end{cases} - (4.1)$$

The main challenge here is identifying correct h[i] for our FIR filter and the order of the FIR filter. A large order gives a better performance. According to references, an order of 180 would suffice. But, due to limited resources on the board, a 32-order filter was used, which ensured low resources and at the same time, satisfactory performance. There are many ways to design an FIR filter. Here, a "least-squares" method (Chapter 3) is employed to design an FIR, using the MATLAB module "filterDesigner". Various parameters like the Wpass and Wstop are adjusted so as to create a response, where 50Hz (generally used frequency in power systems in India, although such a choice is not at all necessary) frequency gain is quite high as compared to its multiple harmonics. Moreover, in order to have a frequency adaptive designing, we used the concept of normalized frequency (Chapter 3). Hence, the received characteristic is of Normalized Frequency vs the Gain. The MATLAB design and the choice of Wpass and Wstop so as to get specified gain-based performance are shown in figure.

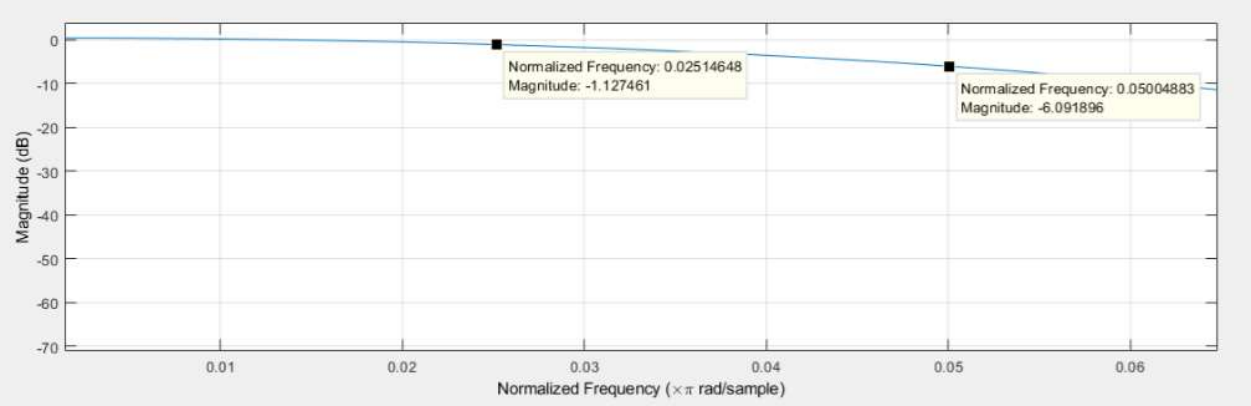

Figure 4.2e. Selecting Wpass and Wstop properly

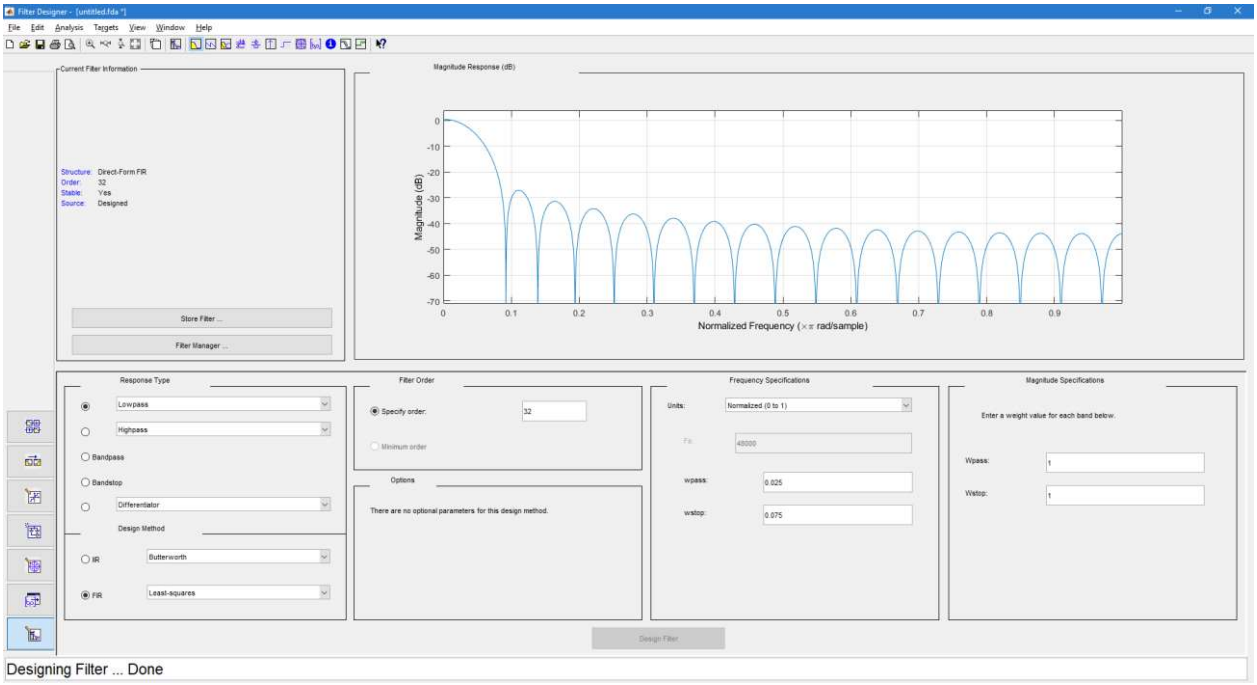

Figure 4.2f. FIR designed characteristics

Now, we follow following steps to use hi in our Verilog code:

1. As described above using FDA toolbox (using the filterDesigner command) in MATLAB FIR Filter can be designed and values of h[i] are calculated
2. We dump the same in a local file.
3. The values are read by the Verilog files in the beginning of the analysis.
4. The weighted sum evaluation is done accordingly.

The code also needs to ensure a generalized approach in terms of:

1. The order of the filter
2. The resolution and of the input signals in terms of the bit-width used.

To simulate the results in Verilog testbench, we created a Lookup Table for sine wave with 32 samples a cycle and then scaled it up to get an appropriate bit resolution, (14-bit input) by multiplying it by suitable constants.

That LUP was read in a module of Verilog to generate a down-sampled clock sampled sine wave, which was used in our simulations later.

The FIR filter for a small order is designed using synchronous D-Flip Flops. When the input arrives, it latches in the D-Flip Flops, and in subsequent clock cycles, the “n” previous values of the input is used to calculate the weighted average of the terms hi*xi.

The assumed bit-width for simulation is 8, but it can be varied as required. Simulation order here is 5. Again, it can be varied as pleased.

But an implementation of a higher order using Gate-Module level coding is a tedious task. Hence, a behavioral approach and Right-Shift registers are used for the same. The generalized code has:

1. Clock, reset and input data stream as the input
2. An FIR output

If the input bit stream is of width “n”, the output bit-width can be calculated by the following expression *data_out_l = ceil(log2((2^data_len)*(order+1)))*

Now a behavioral for loop is used with @always (posedge rst, negedge clk). In every clock negative edge, the right shift action takes place, which allows storing of the most recent input and the last “n-1” inputs. Now, a temporary register is used to calculate ∑(hi*xi). The dataout register stores the final result.

In practice, the output of FIR, had an appreciable DC component, which had to be detected and removed in parallel, to get proper fundamental sine for later calculations.

The inputs and outputs of FIR-Filter Module are listed below.

a. Inputs:

a) clk = Input clock

b) rst = Reset signal

c) Vin = Input voltage signal

b. Outputs:

a) Vout = Filtered output

4.2.5 DZCPD (DC offset compensated Zero crossing detector)

It consists of three modules:

a) AMP measure Module

b) FREQ measure Module

c) PHASE measure Module

a) AMP measure Module

AMP measure Module works in same way as DC-offset removal module using 4 state FSM with states s0, s1, s2 and s3. The State diagram is given in figure **4.2e.** During s1 i.e. positive half cycle max_val is updated with positive peak of input voltage and during s2 i.e. negative half cycle min_val is updated with negative peak of input voltage. Finally in state s3 amplitude is calculated as (max_val-min_val)/2.

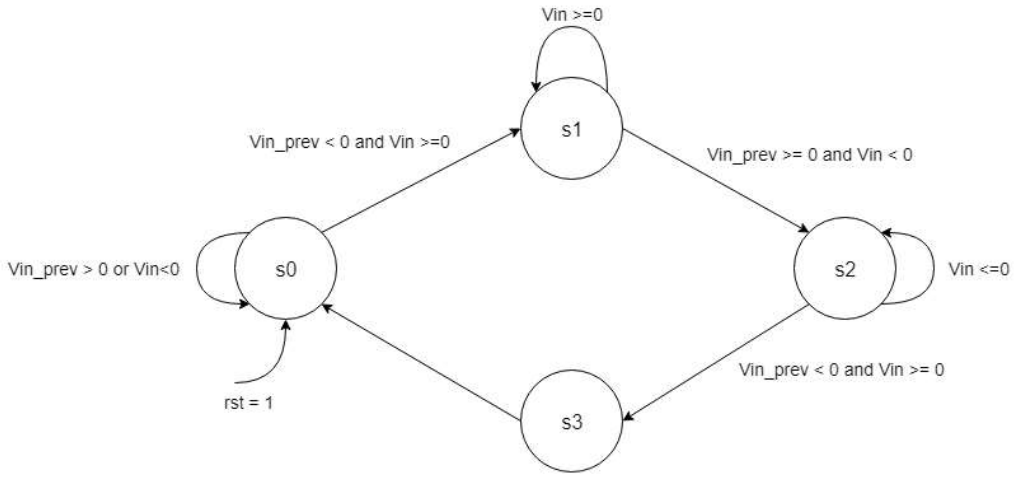

Figure 4.2e. State diagram of AMP measure module FSM

The inputs and outputs of AMP measure Module are described below:

a. Inputs:

a) clk = Input clock

b) rst = Reset signal

c) Vin = input signal i.e Vpos or Vneg or Vzero

b. Outputs:

a) amp = Amplitude of the Positive or Negative or Zero sequence Component

b) FREQ measure Module:

FREQ measure module measures the frequency of input waveform by counting the number of samples between two zero crossing of the inputs signal. It calculates the frequency of input signal in SEQ_DECOMPOSER module and frequency of sequence components in DZCPD module. Freq measure module uses four state FSM with states idle, count1, count2 and done. The state diagram is given in figure **4.2f** We maintain a counter for which the no of pulses in both positive (during state count1) and negative half cycles (during state count2).

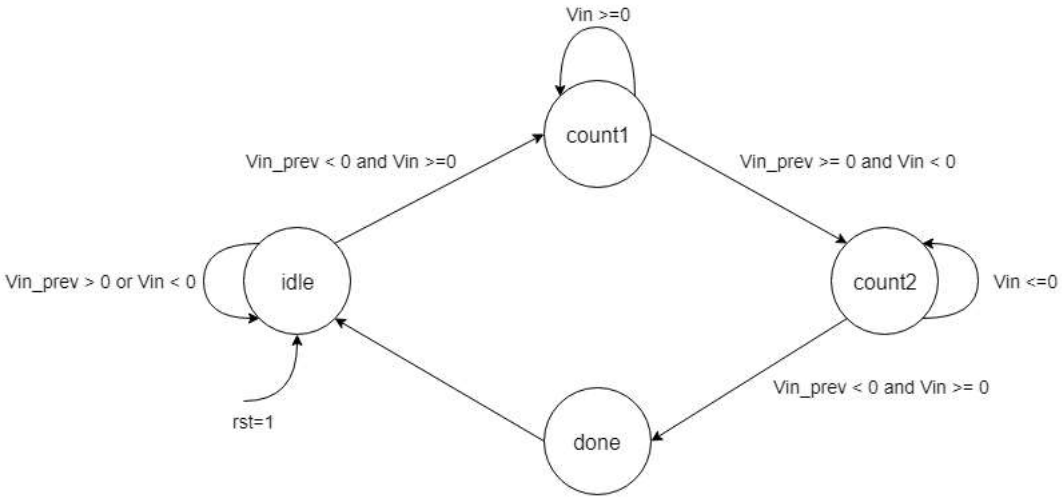

Figure 4.2f. State diagram of FREQ measure module FSM

The inputs and outputs of FREQ measure Module are described below:

a. Inputs:

a) clk = Input clock

b) rst = Reset signal

c) Vin = Input sampled signal

d) k= ADS rate

b. Outputs:

a) freq=(Fs*100)/(counter*k) , where Fs=Sampling Frequency

c) PHASE measure Module:

Phase difference of positive, negative and zero phase sequence components is measured by taking Va_filtered as reference i.e. Vref = Va_filtered. As soon as the Vref signal crosses 0 i.e. from positive half cycle to negative half cycle a counter is started which keeps counting pulses until the zero crossing of sequence components from positive half cycle to negative half cycle. In this way we obtain phase of sequence components.

The module contains four states:

1. Reset: Whenever the reset signal is high the state machine is in reset state
2. Wait1 - The state machine stays in this state till the ref signal crosses its offset value.
3. Wait2: We start a counter and start counting the no. of clock pulses till the corresponding sequence component also crosses its offset value.
4. Result: Then we enter into the result state. Phase of the corresponding sequence component is calculated as phase = 360-((counter%N)*360)/N, where N=32 (No. of samples per cycle)

The state diagram of PHASE measure module FSM is given below:

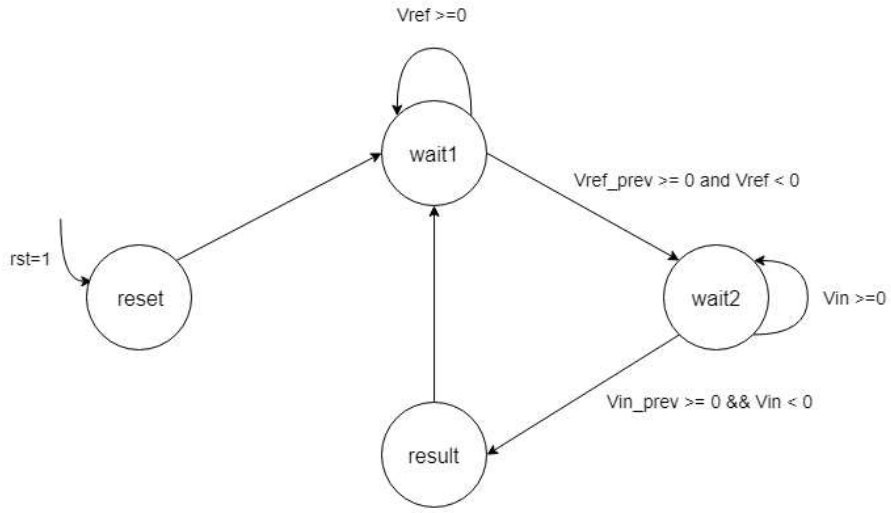

Figure 4.2g. State diagram of Phase measure module FSM

4.3 HIL (Hardware-in-the-Loop) Implementation

The entire project can be seen as a 3 blocked-broad level work:

1. Capturing the analog signal Block (Using the ADCs)
2. Processing the signals Block, using 2 SPARTAN3E, and 1 VIRTEX5
3. Displaying the processed signals Block (Using DACs and CRO)

Figure 4.3a gives the HIL implementation of sequence decomposer. 2 spartan3e boards are used for ADCs, then Vertex 5 board implements the main sequence decomposer module and output positive, negative and zero phase sequence components. Finally, the sequence components are converted in analog signal using DAC so that final output can be viewed on a CRO.

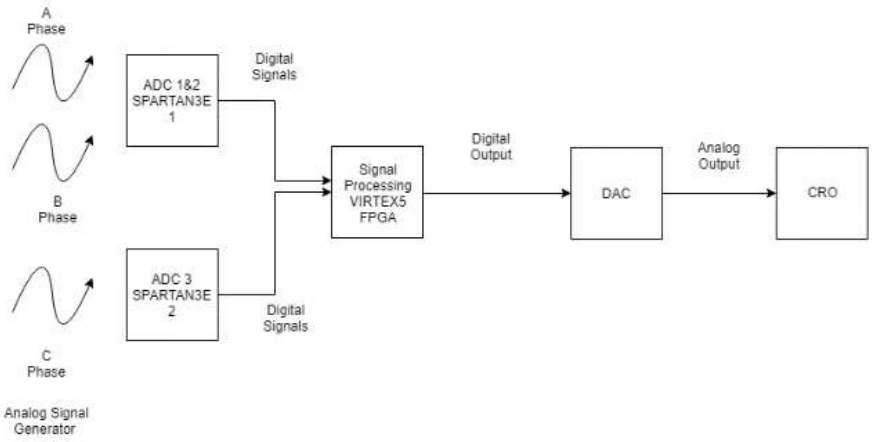

Figure 4.3a. HIL Implementation Diagram

4.3.1 Analog-to-Digital Converters (ADCs)

The real-time application requires capturing the real-world signals from the power system and then processing it digitally. Hence, ADCs will be required for the same. The ADCs work on the principle of sampling and quantization. Any analog signal once fed to an ADC, is sampled at particular intervals of time, and the value captured is quantized to nearest level value.

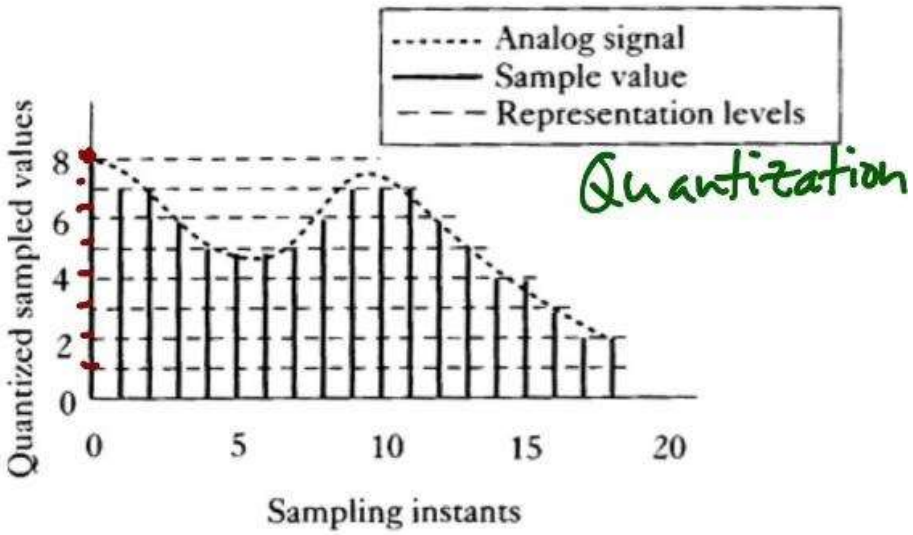

Figure 4.3b. Quantized sampling with 8 representation levels (3 bits per sample)

In this project 3 real-time analog signals are required as inputs. So, 3 ADCs are required for this purpose. Every SPARTAN3E has 2 inbuilt ADCs. So for this sake, we require 2 SPARTAN3Es in order to get 3(>2) ADCs for our signal capturing. Separate Verilog modules to make ADCs in these boards work has been written.

ADC Interfacing on Spartan3E

Table 2 lists the interface signals between the FPGA and the ADC. The SPI_MOSI, SPI_MISO, and SPI_SCK signals are shared with other devices on the SPI bus. The DAC_CS signal is the active Low slave select input to the DAC. The DAC_CLR signal is the active-Low, asynchronous reset input to the DAC.

Signal	FPGA Pin	Direction	Description
SPI_SCK	U16	FPGA→ADC	Clock
AD_CONV	P11	FPGA→ADC	Active-High shutdown and reset.
SPI_MISO	N10	FPGA←ADC	Serial data: Master Input, Serial Output. Presents the digital representation of the sample analog values as two 14-bit two's complement binary values.

Table 4.3. ADC Interface Signals

Digital Outputs from Analog Inputs

The analog capture circuit converts the analog voltage on VINA or VINB and converts it to a 14 bit digital representation, D[13:0], as expressed by Equation.

$$D[13:0] = GAIN \times \frac{V_{IN} - 1.65V}{1.25V} \times 8192 - (4.2)$$

The GAIN is the current setting loaded into the programmable pre-amplifier. The reference voltage for the amplifier and the ADC is 1.65V, generated via a voltage divider shown in Figure 4.3d. Consequently, 1.65V is subtracted from the input voltage on VINA or VINB. The maximum

range of the ADC is ±1.25V, centred on the reference voltage, 1.65V. Hence, 1.25V appears in the denominator to scale the analog input accordingly.

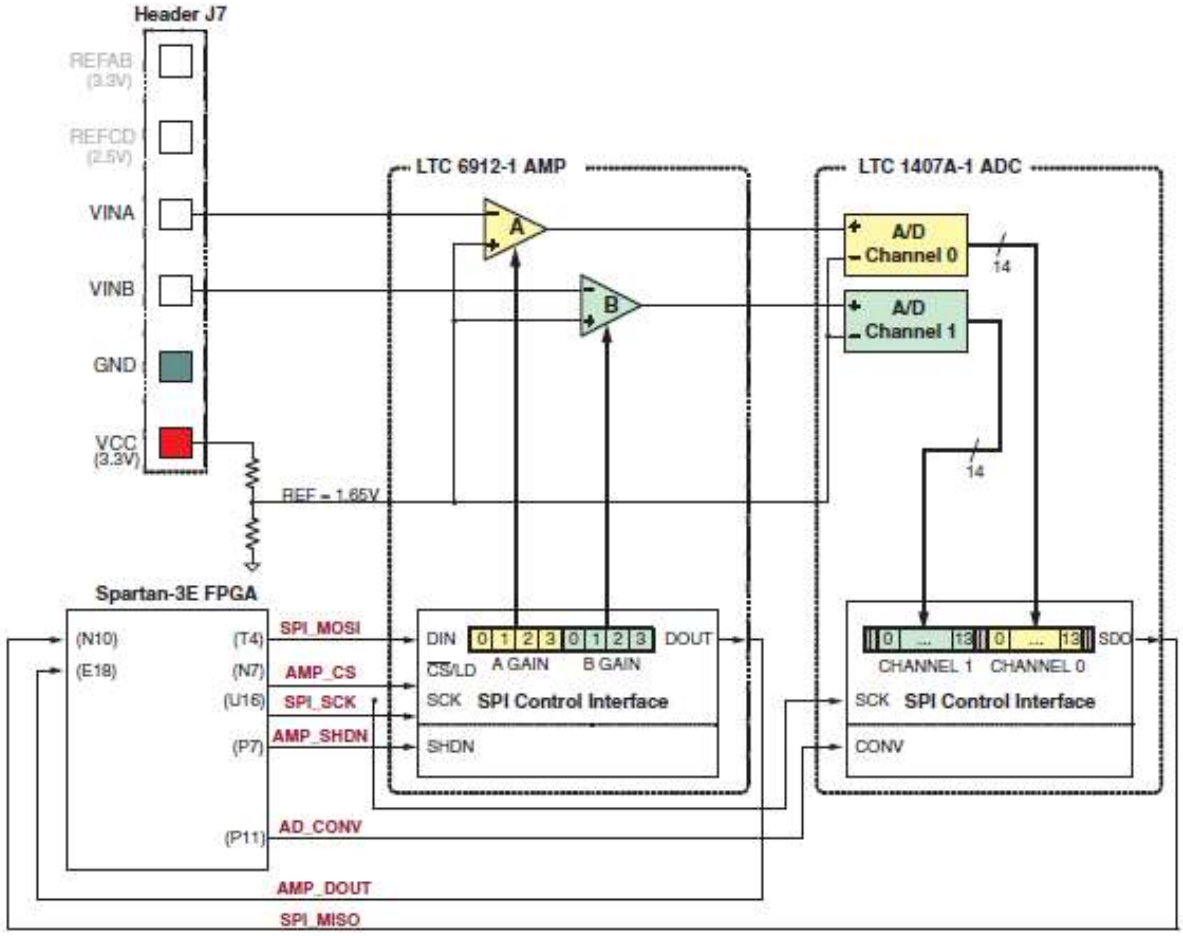

Figure 4.3d. Detailed View of Analog Capture Circuit

SPI Control Interface

Figure 4.3 e and f provides an example SPI bus transaction to the ADC. When the AD_CONV signal goes high, the ADC simultaneously samples both analog channels. The results of this conversion are not presented until the next time AD_CONV is asserted, a latency of one sample. The maximum sample rate is approximately 1.5 MHz. The ADC presents the digital representation of the sampled analog values as a 14-bit, two's complement binary value.

Following this, all the digitized signals are fed to VIRTEX5, where the main blocks for signal splitting are performed.

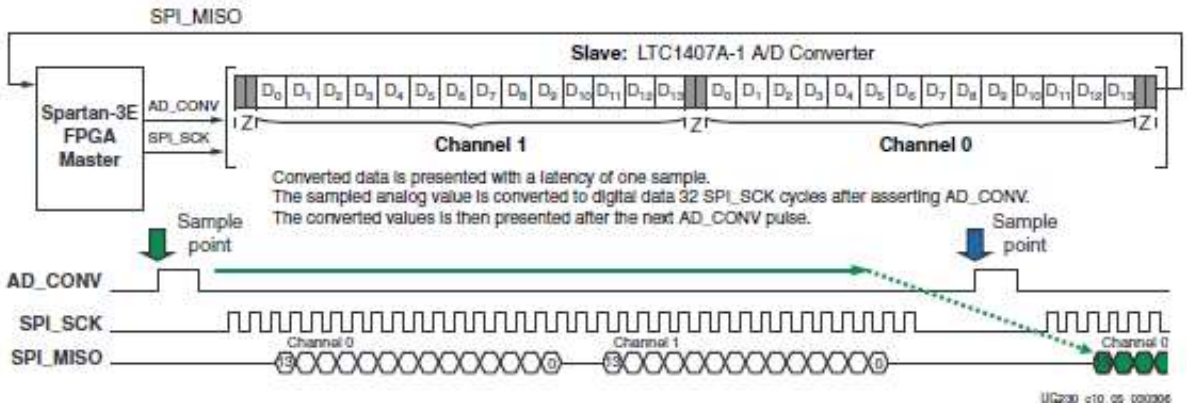

Figure 4.3e. Analog-to-Digital Conversion Interface

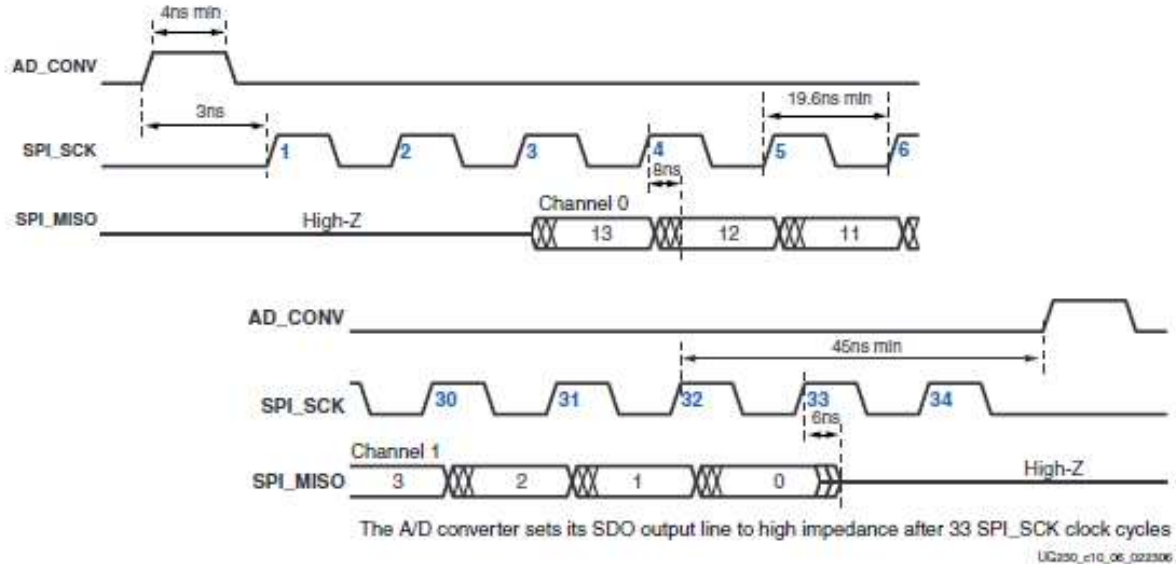

Figure 4.3f. Detailed SPI Timing to ADC

Chapter 5

The Results: Visualizing Outputs

5.1 Verification of Simulink Model

The Simulink model is verified by applying input signals of varying phase difference at fixed amplitude at 50Hz. Output for three cases are given below at balance, equal and unbalanced phase in Figure 5.1a, 5.1b and 5.1c respectively. We can see that satisfying the sequence component formula.

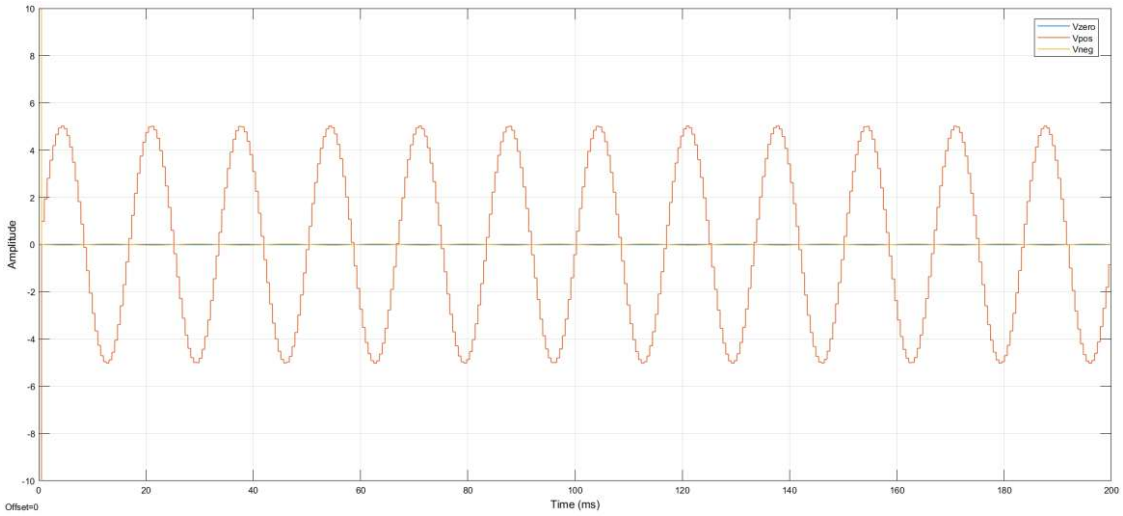

Figure 5.1a. Balanced Three Phase output waveform (Phase A=0°, B=240°, C=120°)

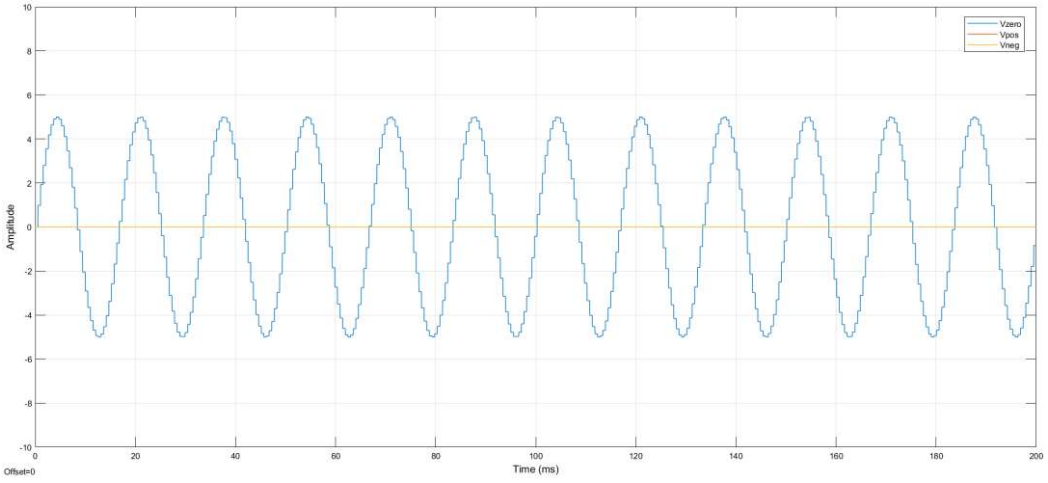

Figure 5.1b. Equal Three Phase output waveform (Phase A=0°, B=0°, C=0°)

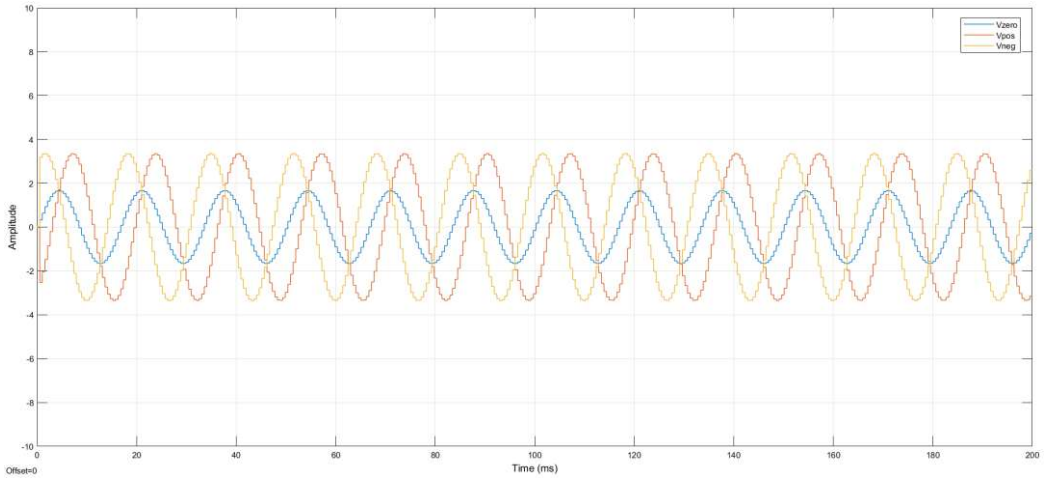

Figure 5.1c. Phase imbalance output waveform (Phase A=0°, B=180°, C=0°)

5.2 HDL Verification of Sequence Decomposer using ModelSim

FIR Filter and Sequence Decomposer HDL model is frequency adaptive so the input frequency is not need to be known priori. 14 bit ADC samples are generated using MATLAB script (with different frequency and phases) and outputs are stored in text files in binary format. Verilog test bench reads the samples from text file one by one. ModelSim is used to display the simulation results in analog format for better visualization of sinusoidal signals. Sequence Decomposer Verilog module is tested with different input frequency and varying phase differences at fixed amplitude.

5.2.1 Output of FIR Filter Module

Figure 5.2a shows the output response of FIR filter with 3rd harmonic present in input waveform. The final output waveform consists of only fundamental frequency component without any harmonics.

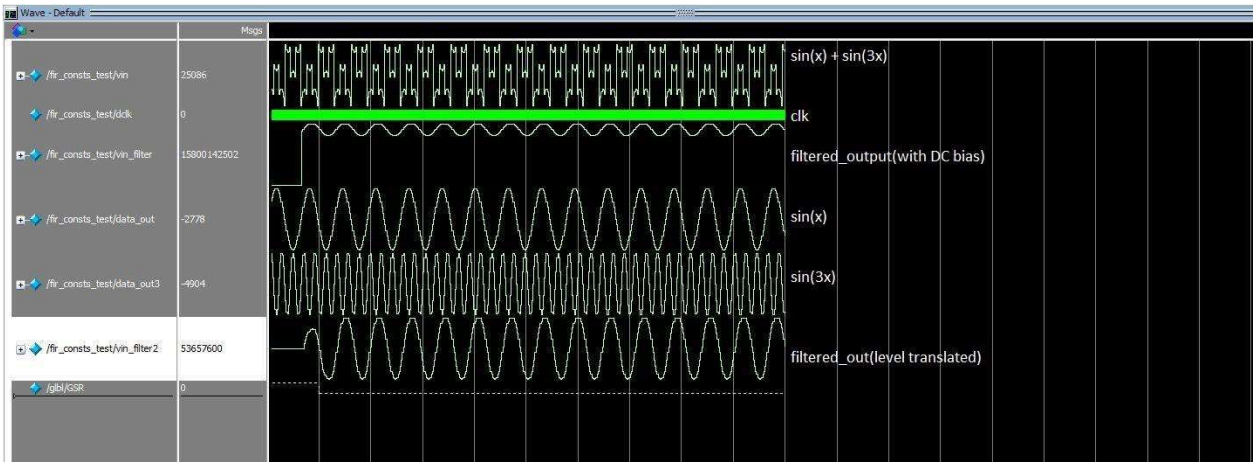

Figure 5.2a. FIR-Filter Simulation Result

5.2.2 Outputs of Sequence Decomposer Module at different phase differences

Figure 5.2b, 5.2c and 5.2d gives the output at 3 different phase differences between Va, Vb and Vc at 50Hz at same conditions as in Simulink. Here first three analog signals in color blue, red and yellow are Va_filtered, Vb_filtered and Vc_filtered respectively. Next three analog signals in color sky blue, green and blue are Vzero, Vpos and Vneg respectively. We can see that both results are matching and satisfying the sequence component formula.

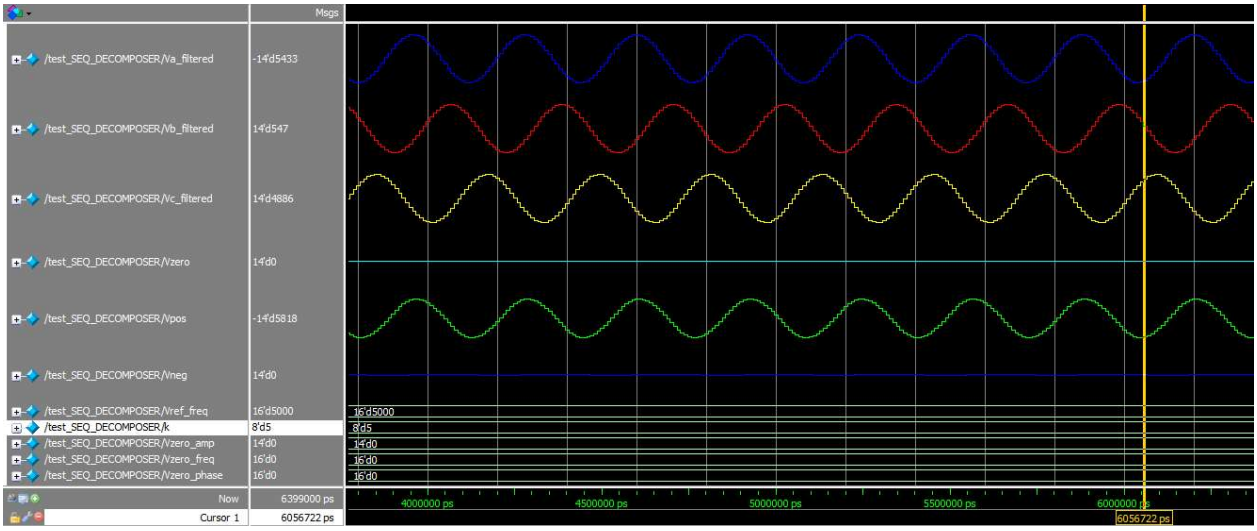

Figure 5.2b. Balanced Three Phase output waveform (Phase A=0°, B=240°, C=120°)

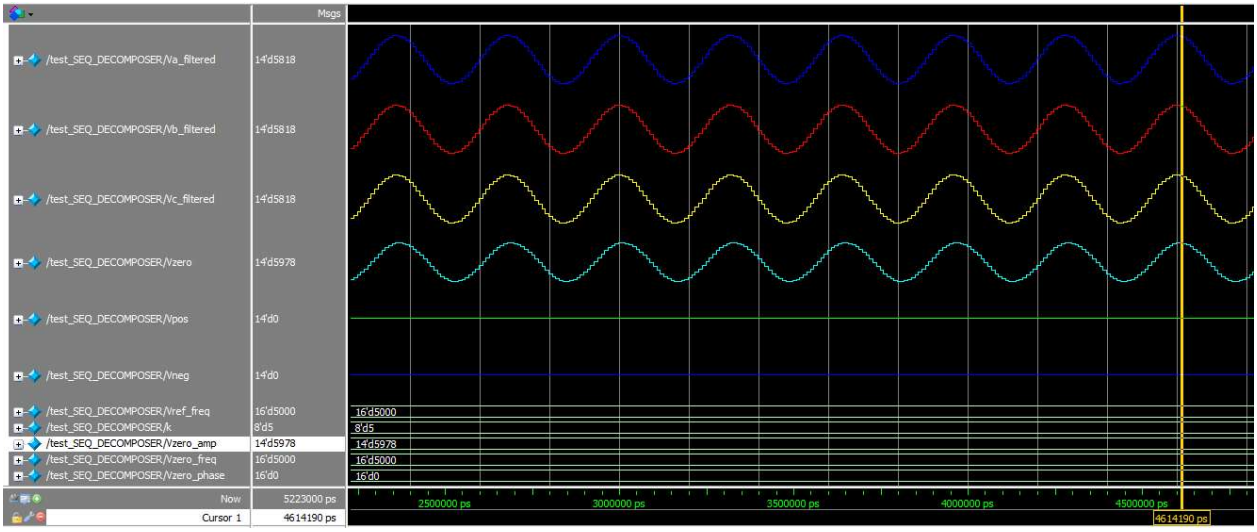

Figure 5.2c. Equal Three Phase output waveform (Phase A=0°, B=0°, C=0°)

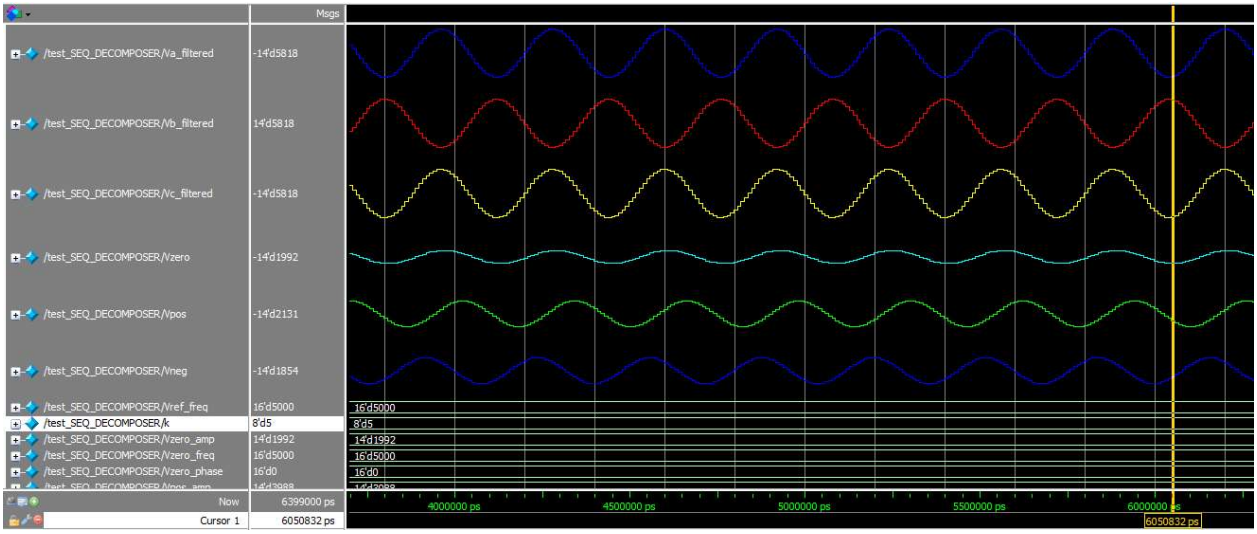

Figure 5.2d. Phase imbalance output waveform (Phase A=0°, B=180°, C=0°)

5.2.3 Outputs of Sequence Decomposer Module at different frequency and balanced three phase

Figure 5.2e and 5.2f gives the balanced three phase output waveform at input frequency of 40Hz and 60Hz respectively. The results are satisfactory and we can say that out sequence decomposer

module is frequency adaptive. For the frequency in range 30Hz - 250Hz this sequence decomposer system will work well.

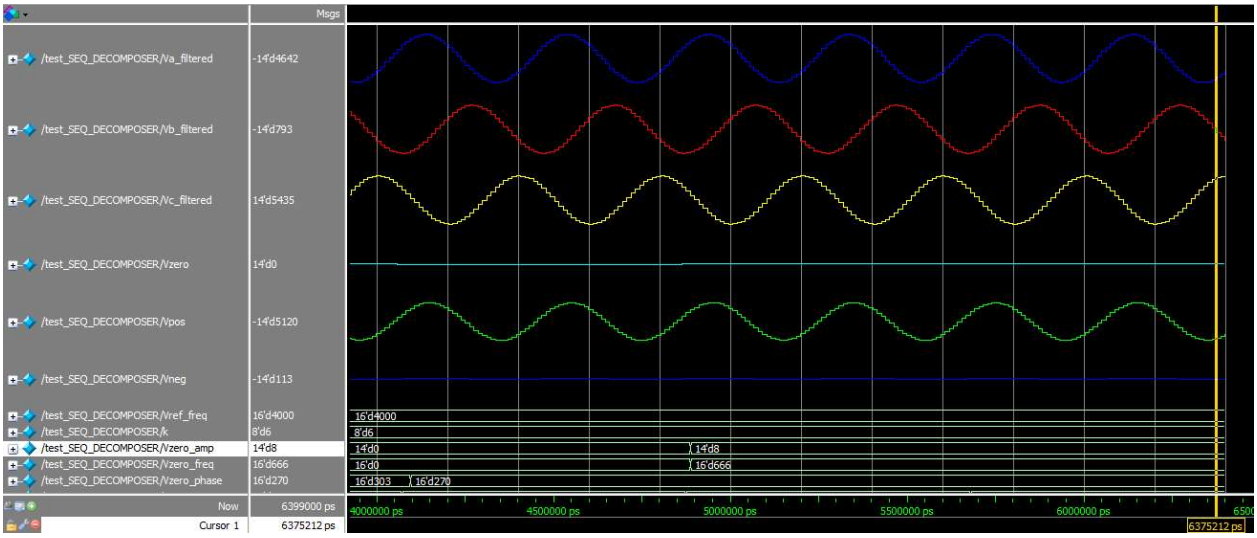

Figure 5.2e. Balanced three phase output waveform at 40Hz

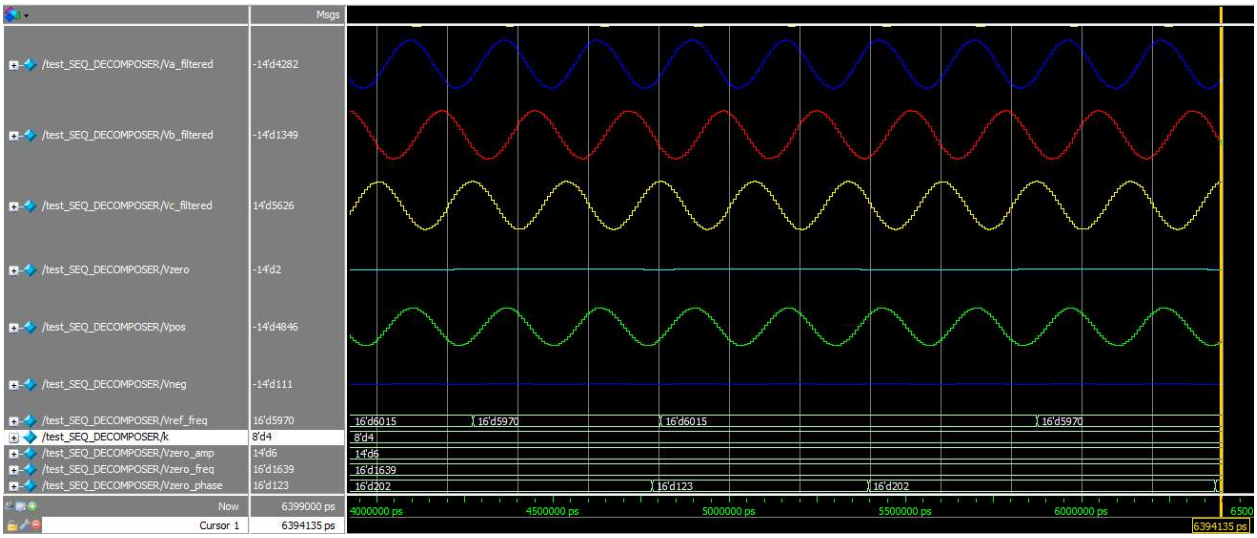

Figure 5.2f. Balanced three phase output waveform at 60Hz

5.2.4 Outputs of Sequence Decomposer Module with harmonic present in input

Figure 5.2g and 5.2h gives the simulation result of sequence decomposer module with 3rd harmonic present in the input signal which is 20% in magnitude of fundamental frequency component. Va, Vb and Vc is balanced three phase supply. Figure 5.2h gives the sequence component in this case so only Vpos is present in output with Vzero having very small amplitude.

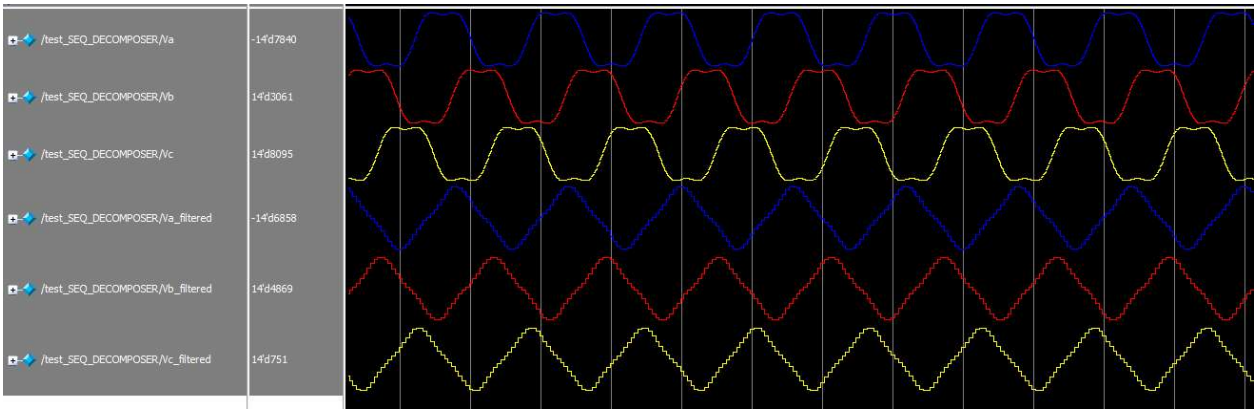

Figure 5.2g. Va_filtered, Vb_filtered and Vc_filtered with 3rd order harmonic present in input

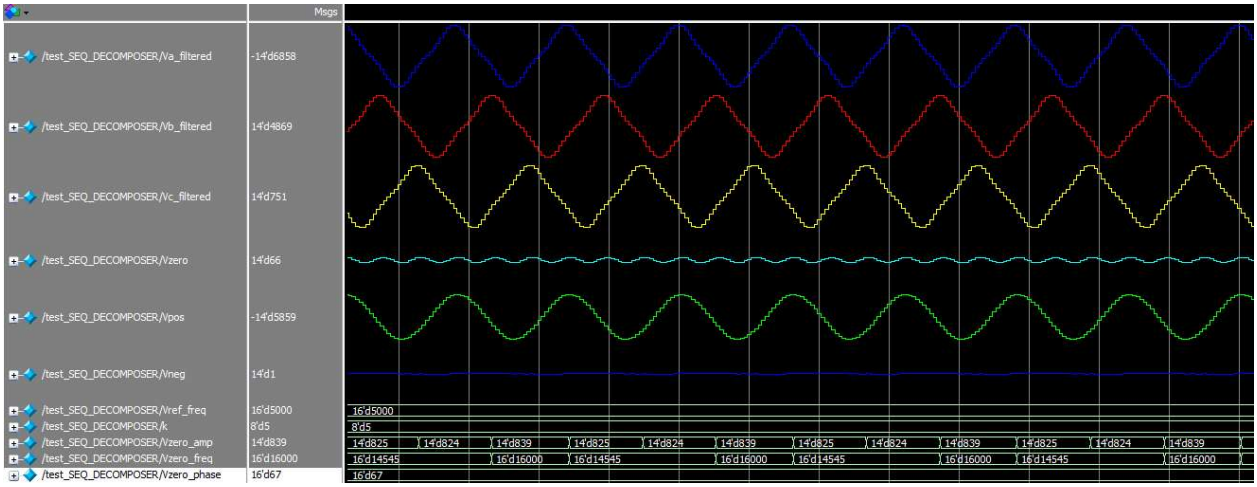

Figure 5.2h. Vpos, Vneg and Vzero waveform with 3rd order harmonic present in input

5.3 Verification of ADC

Figure 5.3a shows the hardware setup for verification of ADC. Working of ADC is verified by giving fixed DC as input and observing the output using Spartan3E LEDs. The Result for different values of input voltages are given in table-5.3.

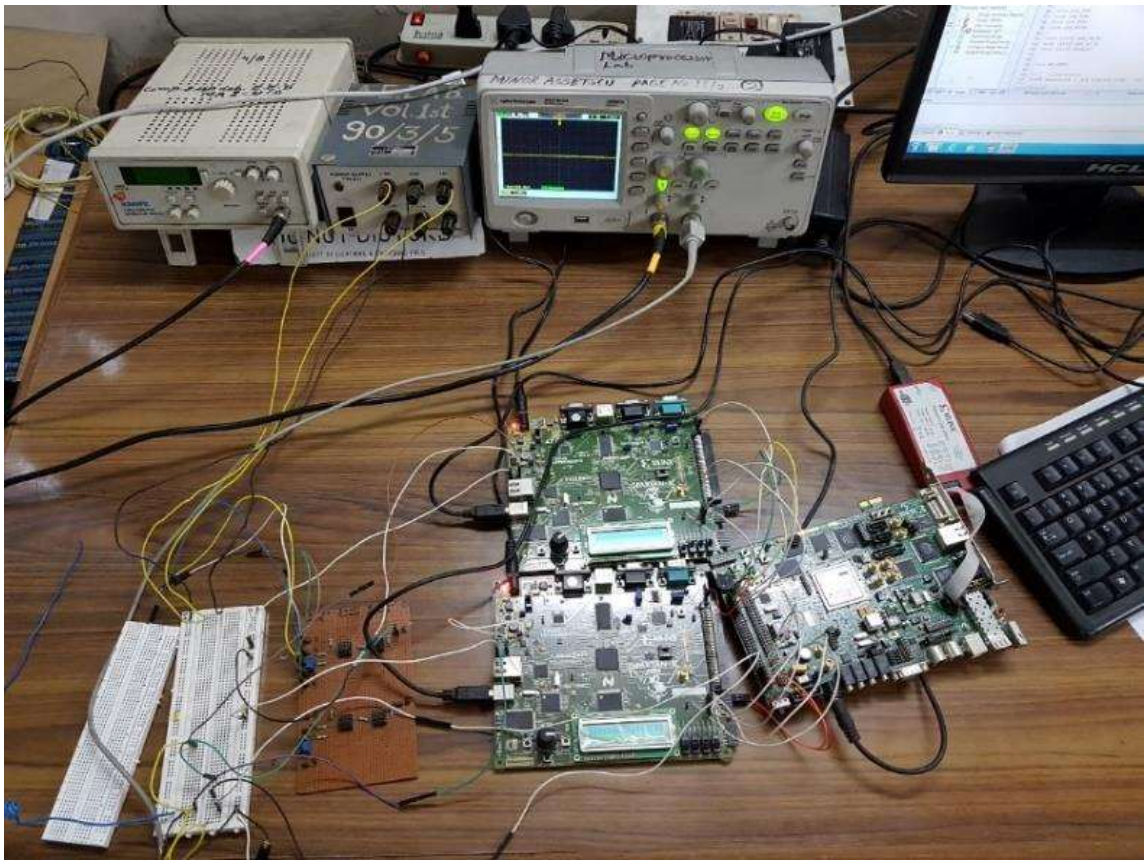

Figure 5.3a: Hardware Setup

Input	Output
+2.9V	$(01\ 1111\ 1111\ 1111)_2$ = +8191
+0.4V	$(10\ 0000\ 0000\ 0000)_2$ = -8192
+2.0V	$(00\ 1000\ 1111\ 0101)_2$ = +2293
+1.0V	$(10\ 1111\ 0101\ 1101)_2$ = -4259

Table 5.3: ADC output result

5.4 FPGA Verification using Xilinx ChipScope Pro

Xilinx ChipScope tool is used for analyzing input and output waveforms of sequence decomposer module. A ChipScope definition file i.e. seq_dec.cdc file is generated with ILA core to view 13 signals which are k, Vref_freq, Va, Vb, Vc, Va2, Vb2, Vc2, Vpos, Vneg, Vzero, Vpos_amp, Vneg_amp and Vzero_amp. Sinusoidal input signal is given using function generator. A RC phase shift circuit is built to introduce a phase difference of -72 degrees for checking the outputs in unbalanced case. Results are first analyzed at constant frequency of 50Hz with equal pahse difference case i.e. case 1 and unbalanced case i.e. case 2. The error is calculated by comparing observed output values to the theoretical output values.

5.4.1 Output at same input signal frequency (Fin=50Hz)

Case 1: Phase A=0°, Phase B=0°, Phase C=0°, Amp A=2775, Amp B=2775, Amp C=2775

In case 1 same input is applied to all three phases so in the output only zero sequence component must be present. Various output waveforms for case 1 are shown in figure 5.4a, 5.4b, 5.4c. Table 5.4a compares the observed values with theoretical/actual values for the calculation of error.

	Observed Value	Actual Value	Error (%)
Vpos	0	0	0.00%
Vneg	0	0	0.00%
Vzero	2709	2775	2.37%

Table 5.4a. Case 1 Observation Table

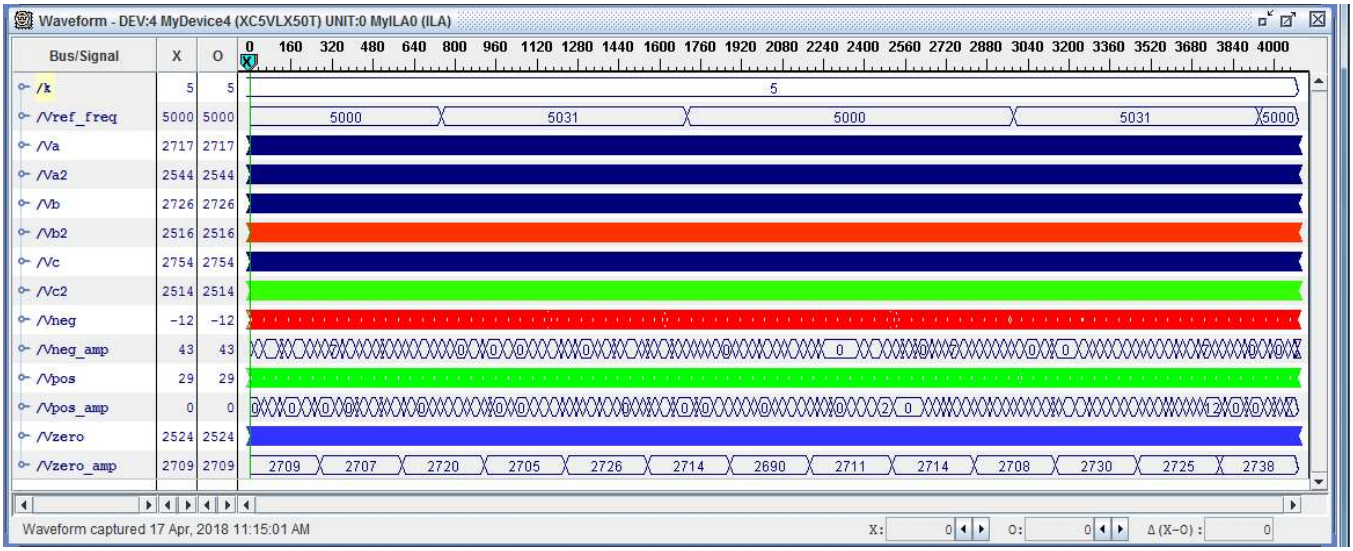

Figure 5.4a. ChipScope output waveform for case 1

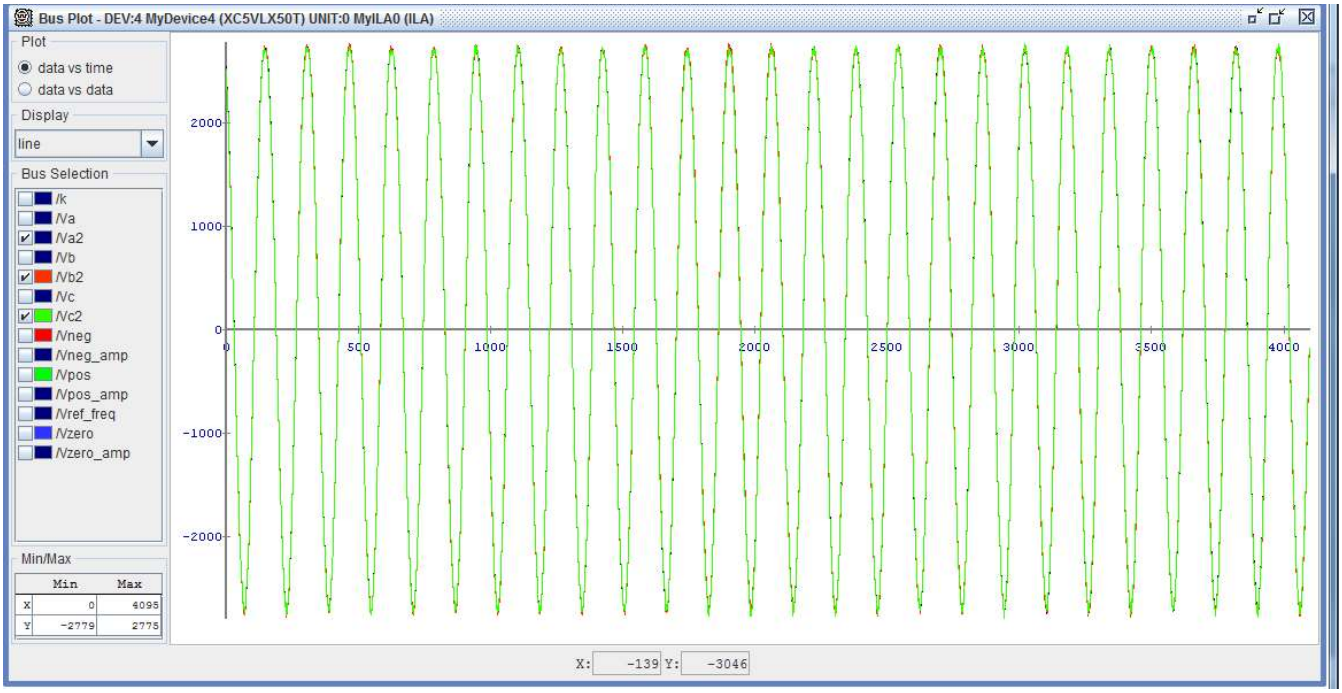

Figure 5.4b. ChipScope waveform of input voltage after offset removal for case 1

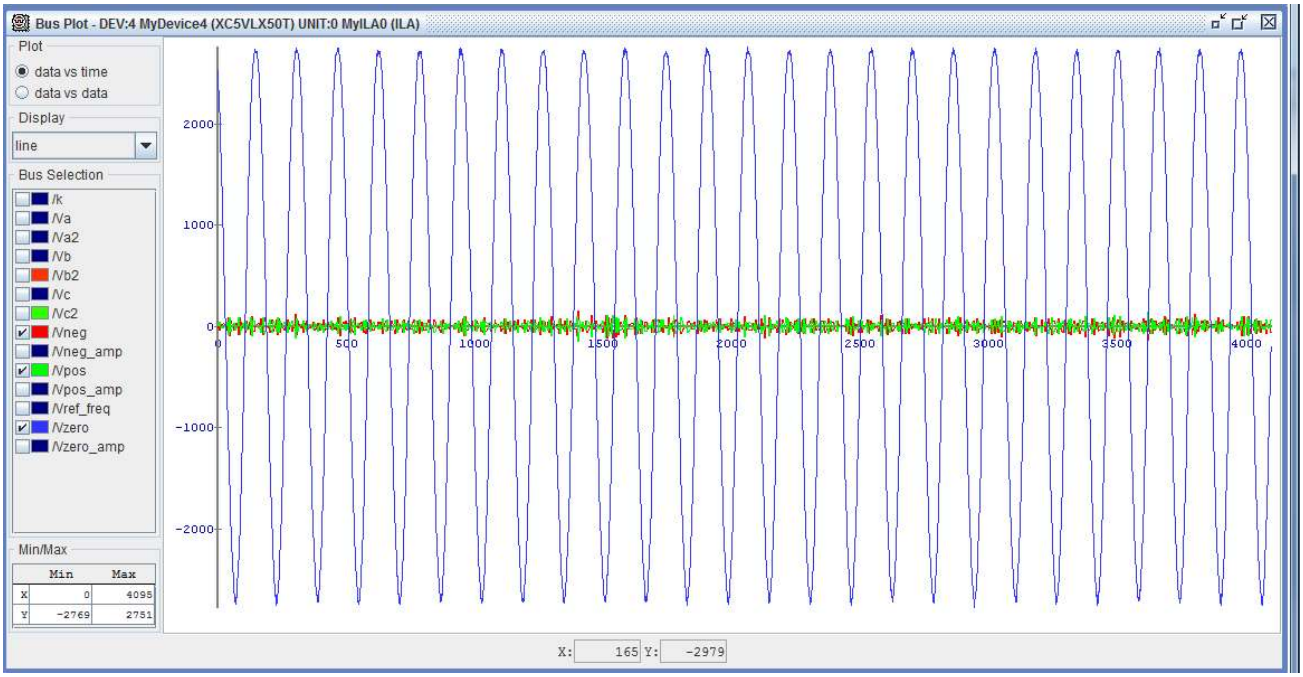

Figure 5.4c. ChipScope waveform of sequence components for case 1

Case 2: Phase A=0°, Phase B=0°, Phase C=-72°, Amp A=2727, Amp B=2727, Amp C=3223

Case 2 represent the unbalance condition in 3 phase system. In this case same input is applied to phase a and b while a RC phase shift circuit is applied to phase c so in the output only zero sequence component must be present. Various output waveforms for case 2 are shown in figure 5.4d, 5.4e, 5.4f. Table 5.4b compares the observed values with theoretical/actual values for the calculation of error.

	Observed Value	Actual Value	Error (%)
Vpos	1163	1173	0.85%
Vneg	1146	1173	2.30%
Vzero	2419	2380	1.61%

Table 5.4b. Case 2 Observation Table

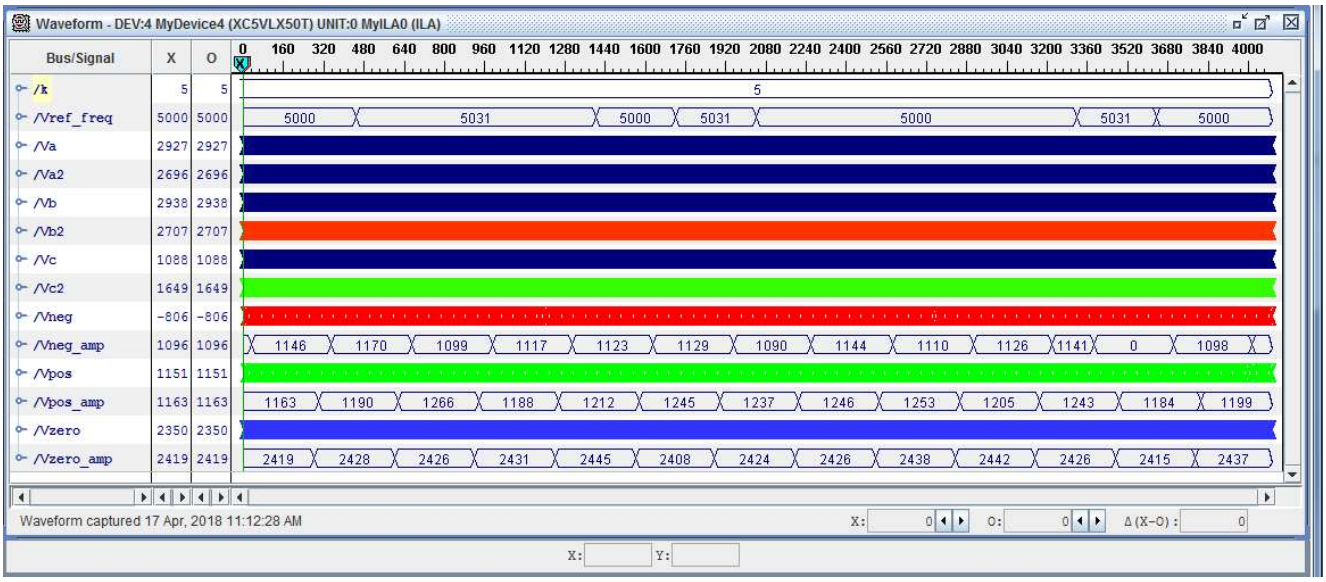

Figure 5.4d. ChipScope output waveform for case 2

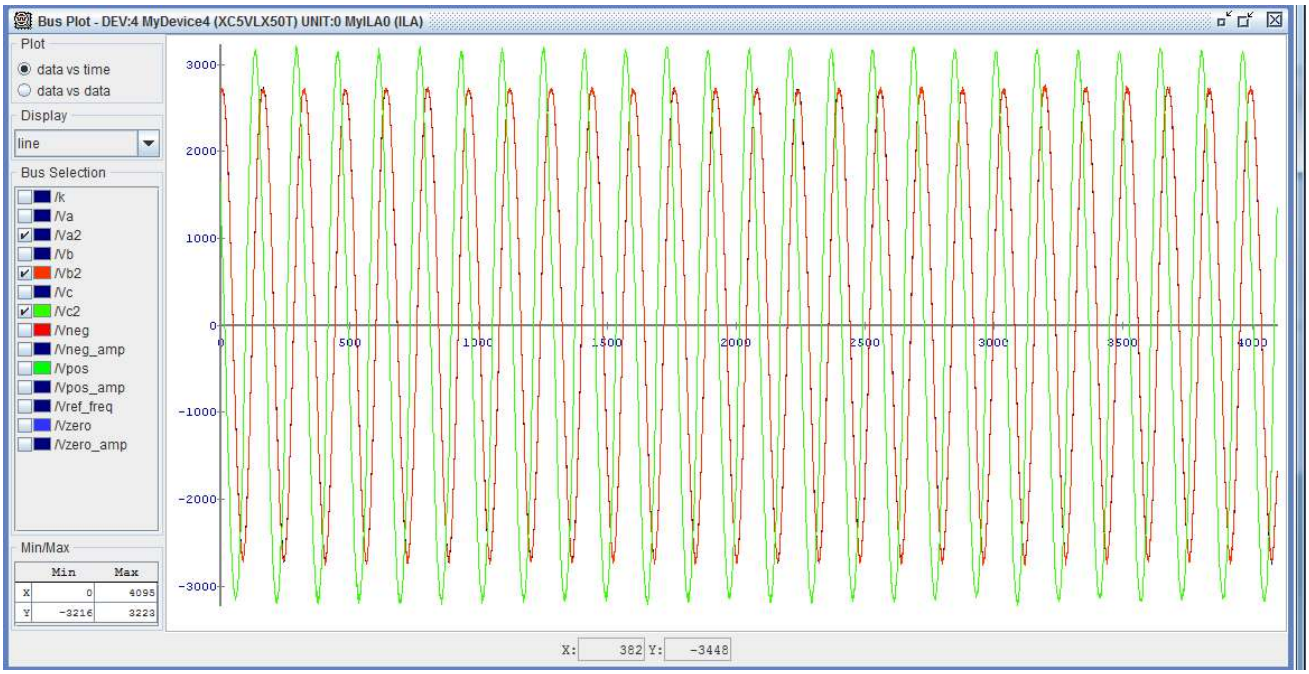

Figure 5.4e. ChipScope waveform of input voltage after offset removal for case 2

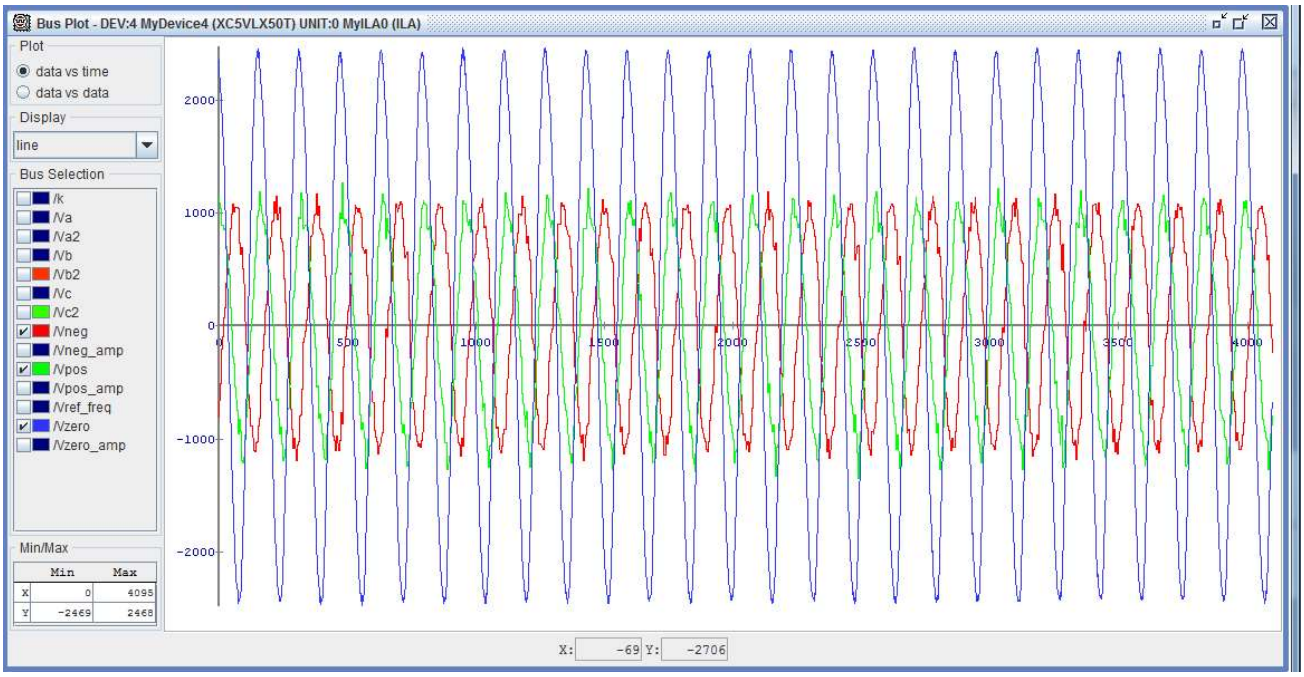

Figure 5.4f. ChipScope waveform of sequence components for case 1

5.4.2 Output at variable input signal frequency

Figure 5.4g and Figure 5.4h gives the ChipScope output waveform in same conditions as case 2 but different input frequency i.e. 40Hz and 60Hz respectively. Due to frequency adaptive nature the output is satisfactory in this case also.

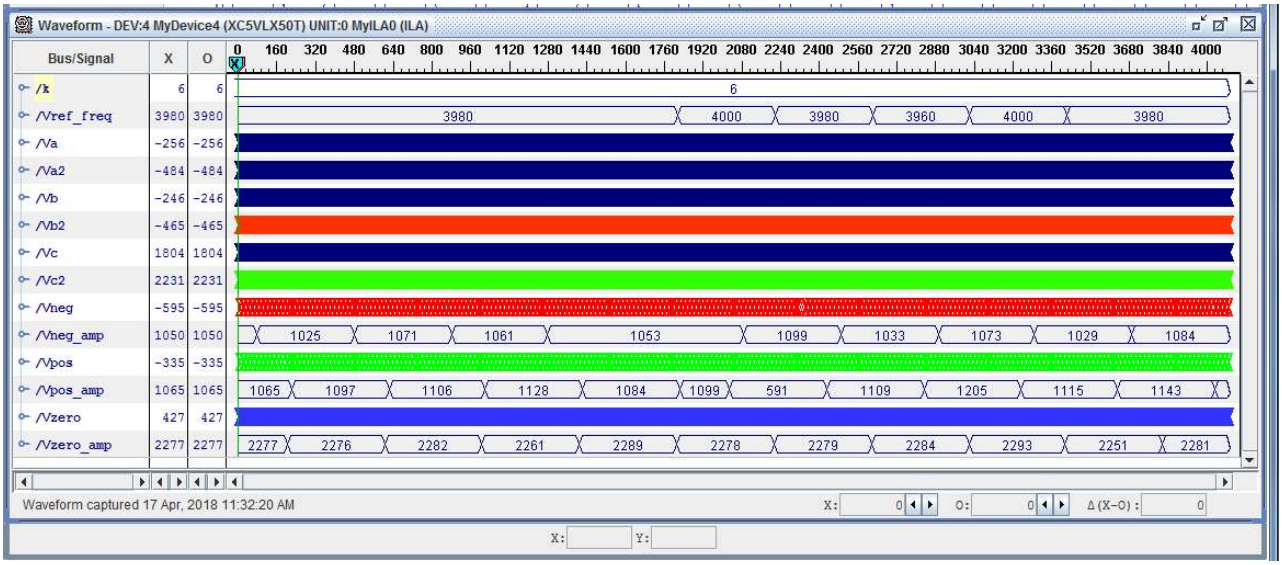

Figure 5.4g. ChipScope output waveform for unbalanced case with Fin=40Hz

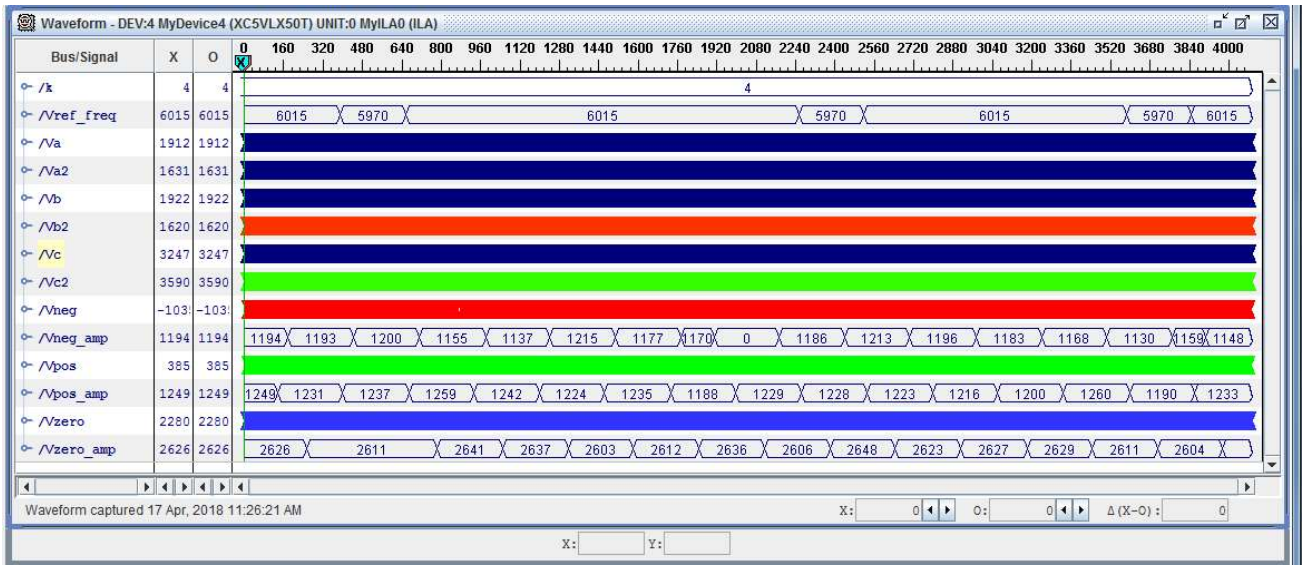

Figure 5.4h. ChipScope output waveform for unbalanced case with Fin=60Hz

Chapter 6

The Conclusion: Analyzing Results

Successfully implemented Three phase to sequence decomposer with a FIR filter in order to remove higher order harmonics.

Visualized output on different kinds of inputs such as

1) All the three inputs are in same phase

2) Two inputs in one phase and the third phase with a phase difference of 60 degrees lagging

The results were according to the manual computation and were accurate within ± 2%. It also beats the traditional tools used for computing the frequency, phase and sequence components as it gives results with a delay of 11.65 degree as opposed to the 120 degree delay. The data can be seen on a computer and can be stored for future purposes such as data analysis.

There were many constraints that we encountered while implementing the code such as LUT constraints (LUT = Look Up table) which is what Virtex-5 uses in order to implement the logic on the board. This constraints made us to optimize the code many times and large amount of time was spent in order to obtain the signals on ChipsSope for visualization.

Finally, the project along with the results under various cases are being compiled under a draft for a research paper to be published under our guide.

References

[1] J. Lewis Blackburn, "Symmetrical components for power system engineering", Marcel Dekker, Inc., 1993

[2] Li, Chunlin, and F. P. Dawson. "A new algorithm for fast retrieval of sequence components in 3-phase networks." Power Conversion Conference, 2002. PCC-Osaka 2002. Proceedings of the. Vol. 3. IEEE, 2002.

[3] El Sahwi, Essam S., Adrian Z. Amanci, and Francis P. Dawson. "A frequency adaptive three-phase sequence detector synchronization system for power systems applications." Industry Applications Society Annual Meeting (IAS), 2012 IEEE. IEEE, 2012.

[4] A.Z. Amanci, F.P. Dawson, "Synchronization System with Zero-Crossing Peak Detection Algorithm for Power System Applications", Int. Power Electron. Conf., pp. 2984-2991, Sapporo, 2010.

[5] Moore, P. J., R. D. Carranza, and A. T. Johns. "A new numeric technique for highspeed evaluation of power system frequency." IEE Proceedings-Generation, Transmission and Distribution 141.5 (1994): 529-536.

[6] Xilinx Spartan-3E FPGA Starter Kit Board User Guide UG230 (v1.2) January 20, 2011 https://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf

[7] Xilinx Virtex-5 FPGA User Guide UG190 (v5.4) March 16, 2012 https://www.xilinx.com/support/documentation/user_guides/ug190.pdf

[8] Wikipedia Normalized Frequency Theory
https://en.wikipedia.org/wiki/Normalized_frequency_(unit)

[9] Least-Squares Linear-Phase FIR Filter Design
https://ccrma.stanford.edu/~jos/sasp/Least_Squares_Linear_Phase_FIR_Filter.html

[9] MATLAB and Simulink 2017b, The MathWorks, Inc., Natick, Massachusetts, United States. http://www.mathworks.com/

[10] Xilinx ISE Design Suite 14.7 https://www.xilinx.com/products/design-tools/ise-design-suite.html