B.TECH PROJECT REPORT

Hardware Designing and Verification of Three Phase to Sequence Decomposer

Department of Electrical Engineering
Indian Institute of Technology Roorkee

PROJECT GUIDE:

Dr. Vishal Kumar

SUBMITTED BY:

Aishwarya Mittal (14115008)
K. Uday Kanth Reddy (14115055)
Shubham Chowdhary (14115115)

Contents

Abstract

Sequence components are basic tools that are widely used in power systems and industrial applications to enable convenient examination and analysis of 3-phased power networks under both balanced and unbalanced operating conditions. Typical unbalances are those caused by faults between the phases and/or ground, open phases and unbalanced loading such as static power equipment and single-phase devices.

The majority of articles published in past have focused on steady state applications of sequence components. Not much attention has been paid to methods of extracting the components quickly. This is crucial if one is to achieve a faster dynamic response. In this report we gave the brief introduction of how the sequence components can be extracted from 3-Phase using digital systems.

The method of extracting sequence components becomes problematic if base frequency is not known a priori. This problem can be resolved by continuously adapting the coefficients in discrete transfer function. As the frequency changes the coefficients are adapted and thus discrete transfer function properties becomes frequency independent.

In recent years, FPGAs are becoming more popular for hardware implementation because of their programmability and reduced development costs. This makes them ideal for rapid development and prototyping. So, we will be using FPGAs for the final model implementation which can later be designed into a PC.

1. Introduction

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

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

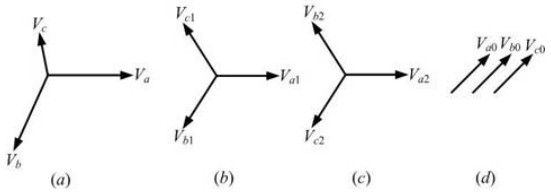


Figure 1: Representation of (a) an unbalanced network, its (b) Positive Sequence,

(c) Negative Sequence and (d) Zero Sequence

Figure 1 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 V_a , V_b and V_c , 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 V_{a1}, V_{a2} and V_{a0} respectively. Note that just like the voltage phasors given in Fig. 1 we can also resolve three unbalanced current phasors into three symmetrical components.

2. Objective

The objective of this project is to develop a hardware model of 3-Phase to sequence converter. 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.

$$3V_{a1} = V_a + V_b e^{-j240^0} + V_c e^{-j120^0} - (1)$$

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

Reference [2] outlines the derivation of the patented ***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 (V_a, V_b, V_c) based on the following discrete time formulas:

$$3V_{+}[k] = V_a[k] + P_bV_b[k] - Z_bV_b[k-1] - P_cV_c[k] + Z_cV_c[k-1] - (3)$$

$$3V_{-}[k] = V_a[k] - P_cV_b[k] + Z_cV_b[k-1] + P_bV_c[k] - Z_bV_c[k-1] - (4)$$

$$3V_0[k] = V_a[k] + V_b[k] + V_c[k] - (5)$$

Where P_b, P_c, Z_b, and Z_c 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 (3), (4) and (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:

$$ADS = round\left(\frac{f_{A/D}}{N.f_{EST}}\right) - (6)$$

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

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.

3. Work done so far

3.1 Designing Simulink Model of 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.

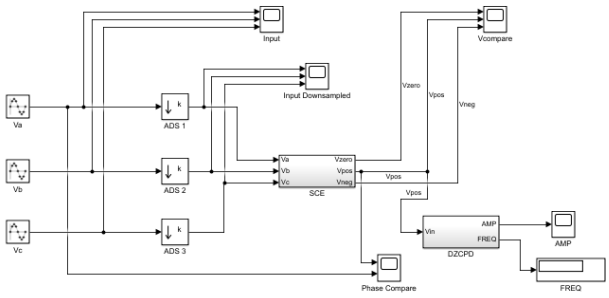


Figure 2. Sequence Decomposer Simulink Model

Figure 2. 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.

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).

2. SCE(Sequence Component Extractor) Block

Figure 3. 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 3, 4, 5 are implemented using discrete transfer function.

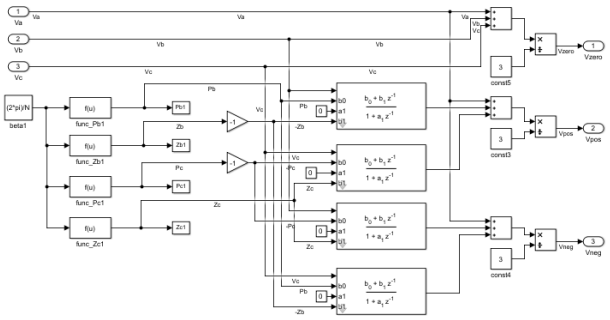


Figure 3. SCE Block

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

Figure 4. 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.

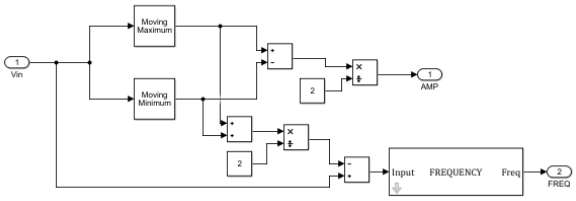


Figure 4. DZCPD Block

Figure 5. 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.

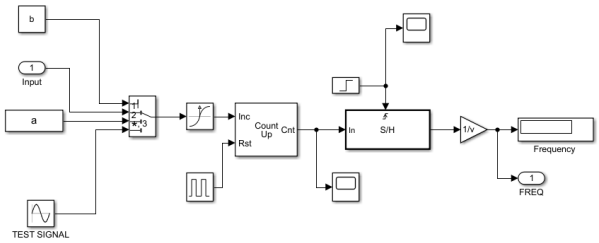


Figure 5. Frequency measurement subsystem

3.2 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) ADS Module
2) SCE Module
3) DZCPD Module for each sequence output I.e positive, negative and zero.

The parameters are initialized as:

- Sampling Frequency Fs = 80KHz i.e Ts = 12.5µs.
- No. of samples per cycle after downsampling N = 32.

ADS Rate (Adaptive Downsampling 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)

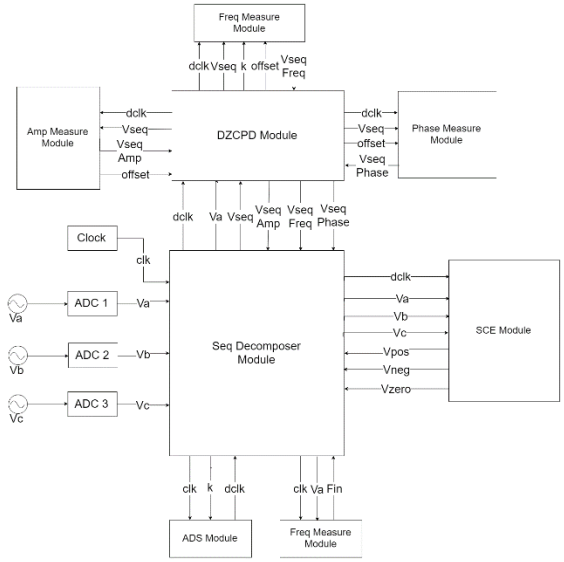


Figure 6. Schematic Diagram of Sequence Decomposer

Figure 6 gives the schematic diagram of SEQ_DECOMPOSER module. Its inputs and outputs 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:

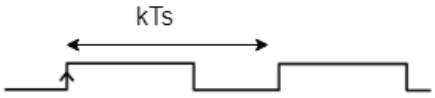
a) k: downsampling rate
b) dclk: downsampled clock
c) Vref_amp: amplitude of reference voltage i.e Va
d) Vref_freq: reference frequency in 100*(frequency in Hz)
e) Vpos: Output positive phase voltage with amplitude Vpos_amp, frequency Vpos_freq, phase Vpos_phase
f) Vneg: Output negative phase voltage with amplitude Vneg_amp, frequency Vneg_freq, phase Vneg_phase
g) Vzero: Output zero phase voltage with amplitude Vzero_amp, frequency Vzero_freq, phase Vzero_phase

The 3 major modules of SEQ_DECOMPOSER module are described below:

3.2.1 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.

Input : Clock at Sampling Frequency Fs



Output : Downsampled Clock of Frequency Fs/k

Figure 7.Working of ADS Module

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

3.2.2 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 3,4 and 5 as following:

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

3.2.3 DZCPD (DC offset compensated Zero crossing detector)

It consists of three modules:

a) AMP_measure Module
b) FREQ_measure Module
c) PHASE_measure Module

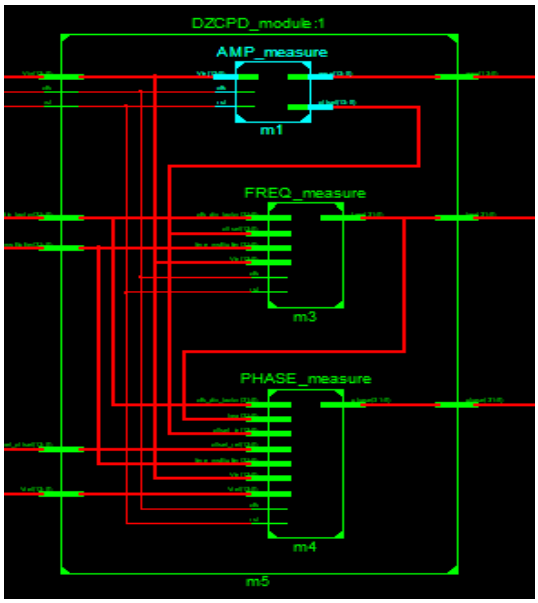


Figure 8. Top level RTL schematic of DZCPD Module

a) AMP_measure Module:

It takes three consecutive samples and finds whether a peak on either direction has been achieved or not.

Vin_1 = Sample of V1 with one time delay

Vin_2 = Sample of V1 with two time delay

The condition (Vin_1-Vin_2) or vice versa is checked so that the difference between two consecutive samples is less than 1 and the signal has not changed a lot. The inputs and outputs of ADS 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) offset = Offset in 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. We maintain a counter for which the no of pulses of Vin is greater than offset i.e Counter = No. of pulses for which Vin >= offset. Its inputs and outputs are described below:

a. Inputs:

a) clk = Input clock
b) rst = Reset Signal
c) Vin = Input sampled signal
d) offset = DC Offset of the sequence component
e) k= ADS rate

b. Outputs:

a) freq=(Fs*100)/(2*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 (Phase A input signal) as reference. As soon as the Va signal crosses its offset from positive to negative a counter is started which keeps counting pulses until the sequence component crosses its offset value. 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 = (counter*360)/N, where N=32 (No. of samples per cycle)

3.3 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)

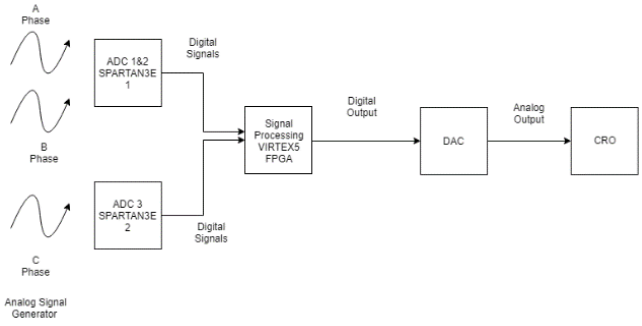


Figure 9: HIL Implementation Diagram

Figure 9 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.

3.3.1 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.

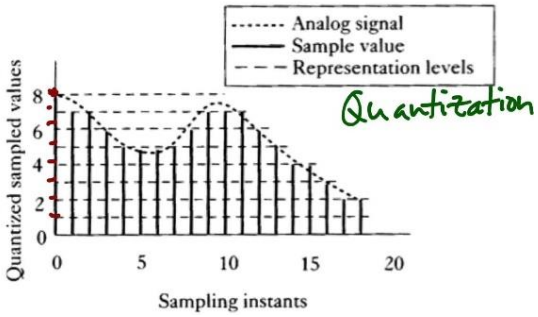


Figure 10: Quantized sampling with 8 representation levels (3 bits per sample)

In our project, we require to tap 3 real-time analog signals. So, we require 3 ADCs for the 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 1 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 1: ADC Interface Signals

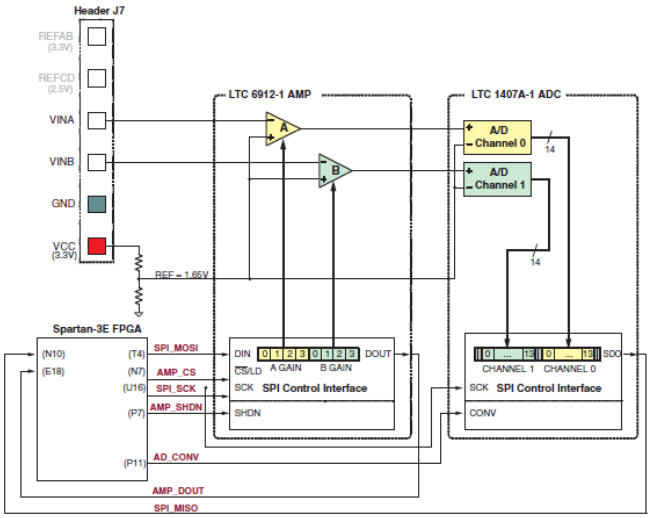


Figure 11: Detailed View of Analog Capture Circuit

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.

$$D[13:0] = GAIN \times \frac{V_{IN} - 1.65V}{1.25V} \times 8192 - (7)$$

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 10-2. Consequently, 1.65V is subtracted from the input voltage on VINA or VINB. The maximum range of the ADC is ±1.25V, centered around the reference

voltage, 1.65V. Hence, 1.25V appears in the denominator to scale the analog input accordingly.

SPI Control Interface

Figure 12 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.

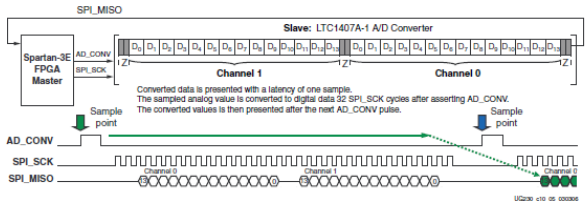


Figure 12: Analog-to-Digital Conversion Interface

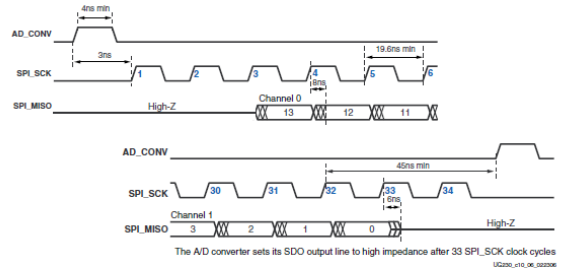


Figure 13: Detailed SPI Timing to ADC

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

3.3.2 Digital-to-Analog Converters (DACs)

Once the signals are processed in our signal splitting system, the output is the frequency, phase, and amplitude of positive sequence of the 3-phase signal. This is a digitally processed signal and hence has digitized values. To understand the significance of the work of VIRTEX5, we need to represent this signal in the form of a wave rather than a simple stream of bits. This is possible by the means of Digital to Analog Converters. These DACs work on the principle of sample and hold. The discrete values are provided to the output and is hold out for some clock period and then the next value in the sequence is produced in the output.

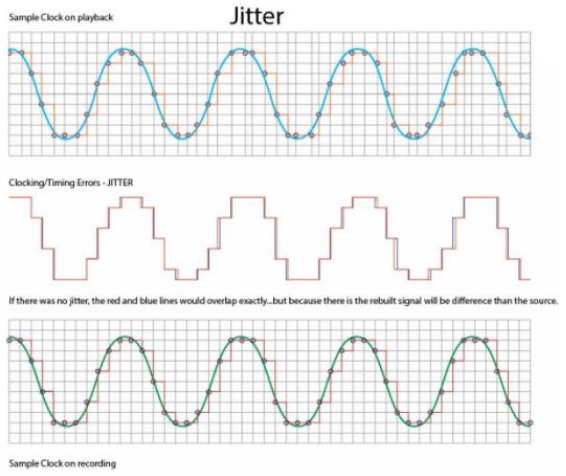


Figure 14: Jitter in DAC

At present, because of the unavailability of a DAC, we are not able to display the stream of bits graphically. So, in order to test the reliability of the system and get a visualizable output stream, both at the same time, we have tested our board system so as to output a DC signal, which can easily be seen without the use of a DAC on a CRO.

4 Results and Discussion

4.2 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 15, 16 and 17 respectively. We can see that satisfying the sequence component formula.

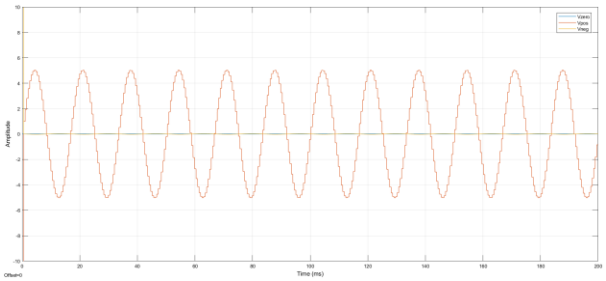


Figure 15. Balanced Three Phase output waveform (Phase A=0°, B=240°, C=120°)

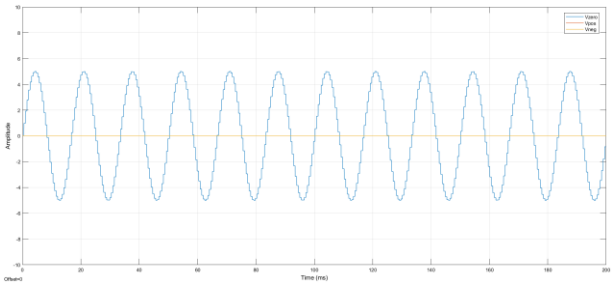


Figure 16. Equal Three Phase output waveform (Phase A=0°, B=0°, C=0°)

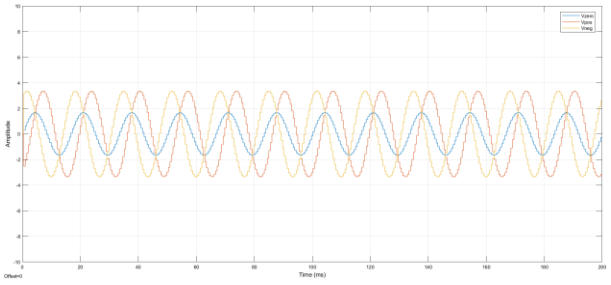


Figure 17. Phase imbalance output waveform (Phase A=0°, B=180°, C=0°)

4.3 HDL Verification of Sequence Decomposer using ModelSim

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.

1. Outputs at different phase difference

Figure 18, 19 and 20 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, Vb and Vc 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.

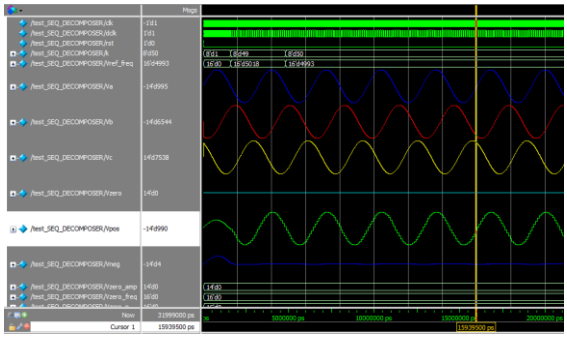


Figure 18. Balanced Three Phase output waveform (Phase A=0°, B=240°, C=120°)

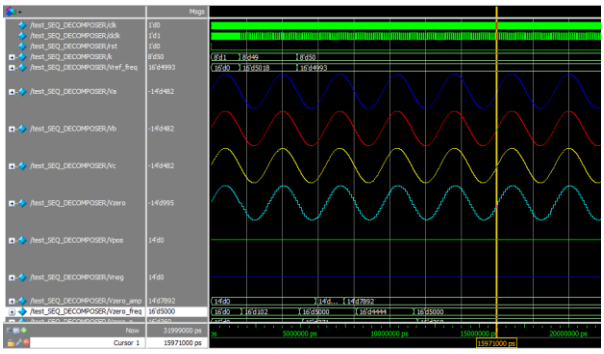


Figure 19. Equal Three Phase output waveform (Phase A=0°, B=0°, C=0°)

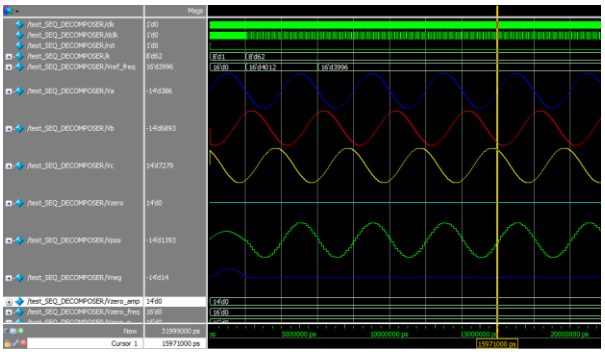


Figure 21. Balanced three phase output waveform at 40Hz

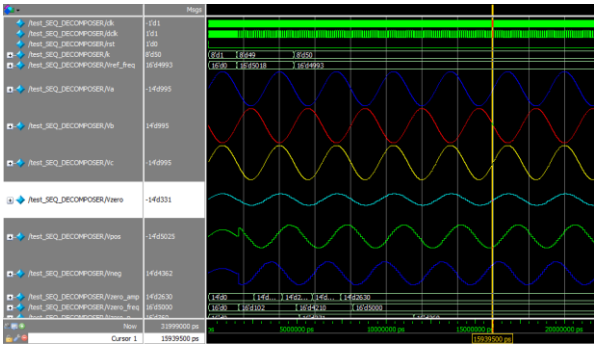


Figure 20. Phase imbalance output waveform (Phase A=0°, B=180°, C=0°)

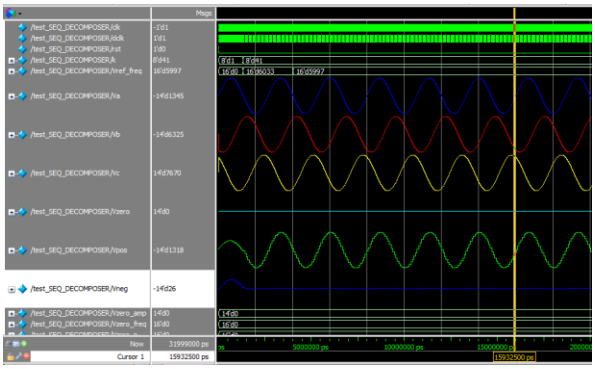


Figure 22. Balanced three phase output waveform at 60Hz

2. Outputs at different frequency and balanced three phase

Figure 21, 22 and 23 gives the balanced three phase output waveform at input frequency of 40Hz, 60Hz and 100Hz respectively. The results are satisfactory and we can say that out sequence decomposer module is frequency adaptive. For the frequency in range 20Hz - 500Hz this sequence decomposer system will work well.

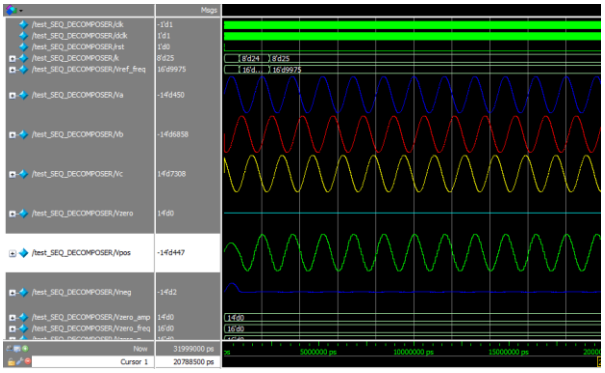


Figure 23. Balanced three phase output waveform at 100Hz

4.4 Verification of ADC

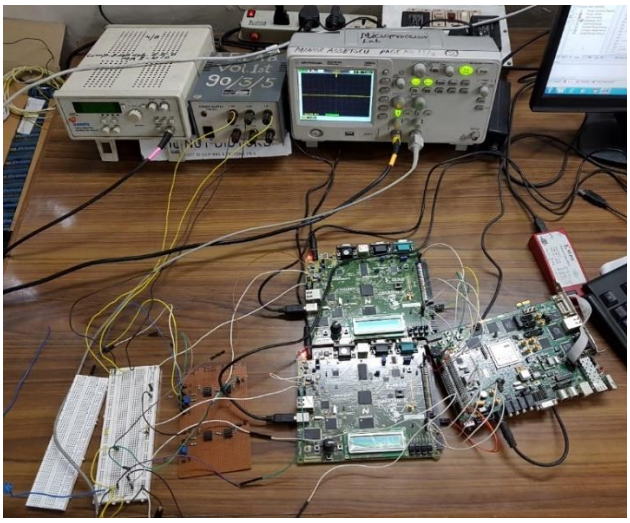


Figure 24: Hardware Setup

Figure 24 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 below:

Input	Output
+2.9V	$(01\ 1111\ 1111\ 1111)_2$ = +8191
+0.4V	$(11\ 1111\ 1111\ 1111)_2$ = -8192
+2.0V	$(00\ 1000\ 1111\ 0101)_2$ = +2293
+1.0V	$(10\ 111\ 0101\ 1101)_2$ = -4259

Table 2: ADC output result

3 Work to be done

The working of the sequence detector can be shown by adding the working ADC module at the input end and a DAC at the output end. Testing of a constant input through ADC is already done. But, the remaining work finally includes interfacing a DAC at the output end, so that when a 3 phase sine wave is applied at the input end, the digitally processed output can be converted to an analog signal, which can be easily visualized on a CRO. Alternatively Xilinx ChipScope tool can be used for debugging by viewing the FPGA internal signals directly in PC. In case any harmonic distortion is present in the input signal, a separate filtering block would be required to ensure a THD-free input to the sequence decomposer.

4 Conclusion

As discussed above, the Sequence Decomposer implementation involves 2 major challenges. The first is the sequence decomposition and the other is frequency adaptation. According to the analysis done so far, the recursive methodology to evaluate sequence components is better than the window based sampling for sequence decomposition.

The given model can also be used as an IoT device which transmits the instantaneous sequence values to a server. On server Data Analysis can be done for building fault prediction model using Machine Learning Techniques. Finally this model can be used to predict fault scenarios in 3-Phase transmission and distribution networks in advance so that required relaying action can be taken before the fault. This method increases the life of components to much large extent and reduce need of regular maintenance work.

5 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 high-speed evaluation of power system frequency." IEE Proceedings-Generation, Transmission and Distribution 141.5 (1994): 529-536.