B. TECH. PROJECT REPORT

On

Design and Implementation of Online and Real-Time Power Quality Monitoring System on FPGA.

by

Premkumar Reddy (130002027)

Sandeep Achary (130002024)

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2016

Design and Implementation of Online and Real-Time Power Quality Monitoring System on FPGA.

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of

BACHELOR OF TECHNOLOGY

in

Electrical ENGINEERING

Submitted by:

Premkumar Reddy (130002027)

Sandeep Achary (130002024)

Guided by:

Dr. Amod C. Umarikar (Assistant professor)

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2016

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"Design and Implementation of Online and Real-Time Power Quality Monitoring System on FPGA."** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Electrical' completed under the supervision of Dr. **Amod C. Umarikar, Asst. Professor, Dept. of Electrical engineering,** IIT Indore is an authentic work.

Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student(s) with date

Premkumar Reddy **Sandeep Achary**

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of BTP Guide(s) with dates and their designation

Dr. Amod C. Umarikar

Asst. Prof. Electrical Engg.

Preface

The aims of the electric power system can be summarized as " Accurate extraction of fundamental frequency component from a distorted voltage/current signal". For decades research and education have been concentrated on the first aim. Reliability and quality of supply were rarely an issue, the argument being that the reliability was sooner too high than too low. Overtime, it became clear that equipment regularly experienced spurious trips due to voltage disturbances, but also that equipment was responsible for many voltage and current disturbances. In order to improve the quality of service, electrical utilities must provide online and real-time monitoring systems that are able to identify the signatures of different disturbance events and thereby providing mitigation techniques to these power quality problems that will help practicing engineers and scientists to design better energy supply systems and mitigate existing ones.

Premkumar Reddy
(130002027)
B.Tech. IV Year
Electrical Engg.
IIT Indore

Sandeep Achary
(13002024)
B.Tech. IV Year
Electrical Engg.
IIT Indore

Acknowledgements

We would like to express our sincere gratitude to our guide, Dr. Amod C. Umarikar for giving us the opportunity to work on this project. His timely guidance and invaluable suggestions were extremely helpful throughout the course of the project.

We would like to extend our gratitude to Karthik Thirumala and Vinay Kumar Tiwari for giving us their valuable time in the useful academic discussions.

We are also extremely thankful to our families and friends for helping us with their constant support and motivation during the entire period of this project work.

Premkumar Reddy
(130002027)
B.Tech. IV Year
Electrical Engg.
IIT Indore

Sandeep Achary
(130002024)
B.Tech. IV Year
Electrical Engg.
IIT Indore

Abstract

Latest innovative ideas to make the life easier using the technology depends upon the application of power electronics in turn about power quality. With increasing quantities of non-linear loads being added to electrical systems, it has become necessary to establish criteria for limiting problems from system voltage degradation. The use of some electrical equipment attached to typical power systems creates voltage quality concerns. There is an increasing awareness that some equipment is not designed to withstand the surges, faults, distortion, and reclosing duty present on typical utility distribution systems. Traditional concerns about steady-state voltage levels and light flicker due to voltage fluctuation also remain. From this fact arises the need and interest to analyze the power quality disturbances and hence find out ways to mitigate them. In this project, these power quality disturbances are quantified as Power Quality Indices which are further used to estimate the extent of power quality disturbances. Various signal processing techniques are used to estimate these indices. Further, these estimated indices are compared with their calculated values to justify the efficiency of the processing technique used.

Our objective is "Accurate extraction of fundamental frequency component from a distorted voltage/current signal." For this Signal processing techniques employed are Fast Fourier Transform(FFT) and Adaptive Fourier Spectral Segmentation(AFSS). AFSS is a new technique developed during this project work. All the designing is carried out in system generator which provides hardware co-simulation, making it possible to incorporate a design running on FPGA into simulink environment.

Table of Contents

List of Figures

Chapter 1
Introduction

The term power quality refers to a wide variety of electromagnetic phenomena that characterize the voltage and current at a given time and at a given location on the power system. PQ is the ability of power system to operate loads without damaging or disturbing them and, also the ability of loads to operate without disturbing or reducing efficiency of the power system. Institute of Electrical and Electronic Engineers (IEEE) Standard 1100 defines power quality as, "The concept of powering and grounding sensitive electronic equipment in a manner suitable for the equipment." Ideally, the best electrical supply would be a sinusoidal voltage waveform with constant magnitude and constant frequency. The Power Quality of a system expresses to which degree a practical supply system resembles the ideal supply system. If the Power Quality of the network is good, then the loads connected to it will run satisfactorily and efficiently. If the Power Quality of the network is bad, then loads connected to it will fail or reduces the lifetime, and the efficiency of the electrical installation will reduce. Electrically-connected equipment is affected by power variations. Determining the exact problems requires sophisticated electronic test equipment.

1.1 Power quality

Various sources use the term "power quality" with different meanings. Other sources use similar but slightly different terminology like "quality of power supply" or "voltage quality." What all these terms have in common is that they treat the interaction between the utility and the customer, or in technical terms between the power system and the load. Treatment of this interaction is in itself not new.

The aim of the power system has always been to supply electrical energy to the customers. What is new is the emphasis that is placed on this interaction, and the treatment of it as a separate area of power engineering. Utilities all over the world have for decades worked on the improvement of what is now known as power quality. The recent increased interest in power quality can be explained in a number of ways. Some of them are listed below:

1.1.1 Equipment has become more sensitive to voltage disturbances:

Electronic and power electronic equipment has especially become much more sensitive than its counterparts 10 or 20 years ago. Scientists have treated this increased sensitivity to voltage disturbances. Not only has equipment become more sensitive, companies have also become more sensitive to loss of production time due to their reduced profit margins. On the domestic market, electricity is more and more considered a basic right, which should simply always be present. The consequence is that an interruption of the supply and poor quality of power.

1.1.2 Equipment causes voltage disturbances:

Tripping of equipment due to disturbances in the supply voltage is often described by customers as "bad power quality." Utilities on the other side, often view disturbances due to end-user equipment as the main power quality problem. Modern (power) electronic equipment is not only sensitive to voltage disturbances, it also causes disturbances for other customers. The increased use of converter-driven equipment (from consumer electronics and computers, up to adjustable-speed drives) has led to a large growth of voltage disturbances, although fortunately not yet to a level where equipment becomes sensitive. The main issue here is the nonsinusoidal current of rectifiers and inverters. The input current not only ontains a power frequency component (50 Hz or 60 Hz) but also so-called harmonic components with frequencies equal to a multiple of the power frequency. The harmonic distortion of the current leads to harmonic components in the supply voltage.

Equipment has already produced harmonic distortion for a number of decades. But only recently has the amount of load fed via power electronic converters increased enormously: not only large adjustable-speed drives but also small consumer electronics equipment. The latter cause a large part of the harmonic voltage distortion: each individual device does not generate much harmonic currents but all of them together cause a serious distortion of the supply voltage.

1.1.3 A growing need for standardization and performance criteria:

The consumer of electrical energy used to be viewed by most utitilies simply as a "load." Interruptions and other voltage disturbances were part of the deal, and the utility decided what was reasonable. Any customer who was not satisfied with the offered reliability and quality had to pay the utility for improving the supply.

Today the utilities have to treat the consumers as "customers." Even if the utility does not need to reduce the number of voltage disturbances, it does have to quantify them one 'way or the other. Electricity is viewed as a product with certain characteristics, which have to be measured, predicted, guaranteed, improved, etc. This is further triggered by the drive towards privatization and deregulation of the electricity industry.

1.1.4 Deregulation of market:

Open competition can make the situation even more complicated. In the past a consumer would have a contract with the local supplier who would deliver the electrical energy with a given reliability and quality. Nowadays the customer can buy electrical energy somewhere, the transport capacity somewhere else and pay the local utility, for the actual connection to the system. It is no longer clear who is responsible for reliability and power quality.

As long as the customer still has a connection agreement with the local utility, one can argue that the latter is responsible for the actual delivery and thus for reliability and quality. But what about voltage sags due to transmission system faults? In some cases the consumer only has a contract with a supplier who only generates the electricity and subcontracts transport and distribution. One could state that any responsibility should be defined by contract, so that the generation company with which the customer has a contractual agreement would be responsible for reliability and quality. The responsibility of the local distribution would only be towards the generation companies with whom they have a contract to deliver to given customers. No matter what the legal construction is, reliability and quality will need to be well defined.

1.1.5 Utilities want to deliver a good product:

Something that is often forgotten in the heat of the discussion is that many power quality developments are driven by the utilities. Most utilities simply want to deliver a good product, and have been committed to that for many decades. Designing a system with a high reliability of

supply, for a limited cost, is a technical challenge which appealed to many in the power industry, and hopefully still does in the future.

1.1.6 The power quality can be measured:

The availability of electronic devices to measure and show waveforms has certainly contributed to the interest in power quality. Harmonic currents and voltage sags were simply hard to measure on a large scale in the past. Measurements were restricted to rms voltage, frequency, and long interruptions; phenomena which are now considered part of power quality, but were simply part of power system operation in the past.

1.2 Signal Processing techniques

Various signal processing techniques can be used to break down the signal into various frequency components and hence estimate the power quality indices. To analyze the disturbances, present in the signal, data are often available as a form of sampled time function that is represented by a time series of amplitudes. When dealing with such data, the Fourier transform (FT)-based approach is most often used. FT provides the frequency information; however, it is not capable of providing time information about signal disturbances. For instance, time-frequency information related to disturbance waveforms can be obtained by using the STFT. Similarly, DWT can be used, however to make the filter more adaptive a new technique Adaptive Spectral Segmentation (AFSS) is proposed which segments the spectrum based on the boundaries calculated separately for each new signal.

1.3 System Generator

System Generator is a DSP design tool from Xilinx that enables the use of the MathWorks model-based Simulink design environment for FPGA design. Previous experience with Xilinx FPGAs or RTL design methodologies are not required when using System Generator. Designs are captured in the DSP friendly Simulink modeling environment using a Xilinx specific blockset. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file.

1.4 FPGAs

A field programmable gate array (FPGA) is a general-purpose integrated circuit that is "programmed" by the designer rather than the device manufacturer. Unlike an application-specific integrated circuit (ASIC), which can perform a similar function in an electronic system, an FPGA can be reprogrammed, even after it has been deployed into a system.

An FPGA is programmed by downloading a configuration program called a bitstream into static on-chip random-access memory. Much like the object code for a microprocessor, this bitstream is the product of compilation tools that translate the high-level abstractions produced by a designer into something equivalent but low-level and executable. Xilinx System Generator pioneered the idea of compiling an FPGA program from a high-level Simulink model.

An FPGA provides you with a two-dimensional array of configurable resources that can implement a wide range of arithmetic and logic functions. These resources include dedicated DSP blocks, multipliers, dual port memories, lookup tables (LUTs), registers, tristate buffers, multiplexers, and digital clock managers. In addition, Xilinx FPGAs contain sophisticated I/O mechanisms that can handle a wide range of bandwidth and voltage requirements. The Virtex®-4 FPGAs include embedded microcontrollers (IBM PowerPC® 405), and multi-gigabit serial transceivers. The compute and I/O resources are linked under the control of the bitstream by a programmable interconnect architecture that allows them to be wired together into systems.

Chapter 2

Hardware Co-simulation

System Generator for DSP provides the functionality for performing Hardware Co-Simulation for designs that run both in hardware and in software. Hardware Co-Simulation can verify the operation of designated parts of a design in hardware to significantly decrease simulation runtimes. This is a powerful feature of System Generator for DSP, especially when considering the parallel nature of FPGA devices. Hardware Co-Simulation can make it possible to complete even very long simulations within a much shorter period of time.

System Generator DSP simulation performance results are analyzed by reviewing simulation run times. In addition, the way that a System Generator for DSP simulation fits into a Simulink simulation also affects performance. The following sections are included to provide background about how System Generator for DSP blocks fit into the Simulink simulation environment.

It is important to understand how a Hardware Co-Simulation fits into a Simulink simulation. There are two clocking schemes available: single-stepped, and free-running.

- In a **single-stepped** simulation, the clock from the System Generator solver is sent to the logic running in hardware over the Hardware Co-Simulation interface.
- In a **free-running** simulation, the part of the design that is running in software is not synchronized with the part running in hardware. The clock driving the logic in hardware is typically an onboard oscillator, which easily outperforms the single-stepped clock provided by Simulink.

2.1 Comparison of Implementations

A comparison of the results obtained for each implementation method is provided to show which method may produce the best results for different simulation times. Results are compared at three different simulation runtimes: 1,000, 10,000 and 100,000. This is followed by a comparison of the three fastest hardware implementations over very long simulation runtimes.

Short Runtime Comparison: 1000s

For short simulation times, comparison shows that a software implementation outperforms the hardware implementations. This is expected, as the Hardware Co-Simulation implementations must initialize the hardware (configure FPGAs, etc.) as indicated by the *Sysgen: Initialize Simulation* portions of Figure. This is not necessary for any software-only implementations.

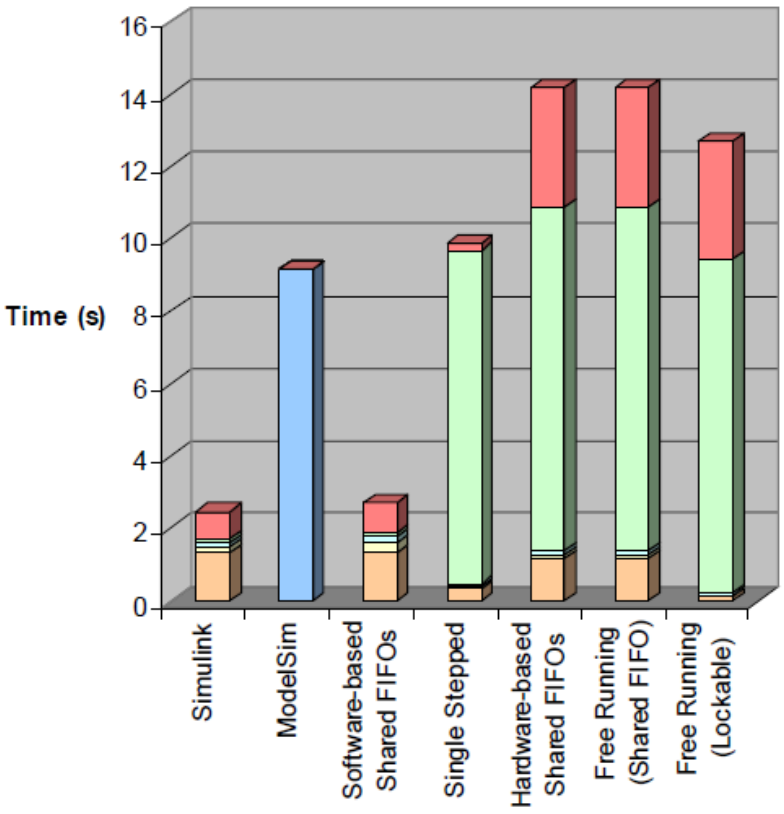

Figure 2.1

Intermediate Runtime Comparison: 10,000s

As shown, the runtimes of software and hardware implementations are fairly close to each other. This is due to the time required to initialize the hardware simulations (program the FPGA, etc.) as that is fairly constant and independent of simulation runtime. This is the major disadvantage of the hardware implementations with a short simulation. Figure illustrates the intermediate simulation times comparison.

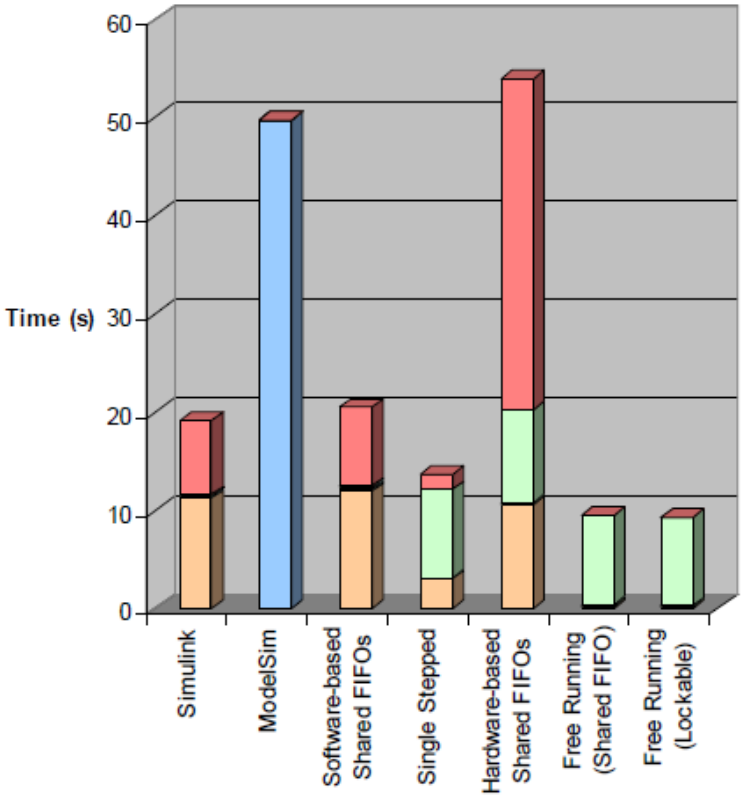

Figure 2.2

Long Runtime Comparison: 100,000s

Figure shows the longer runtime comparison. As indicated, the Hardware Co-Simulation implementations clearly outperform the software implementations, with the frame-based freerunning simulations outperforming a single stepped Hardware Co-Simulation.

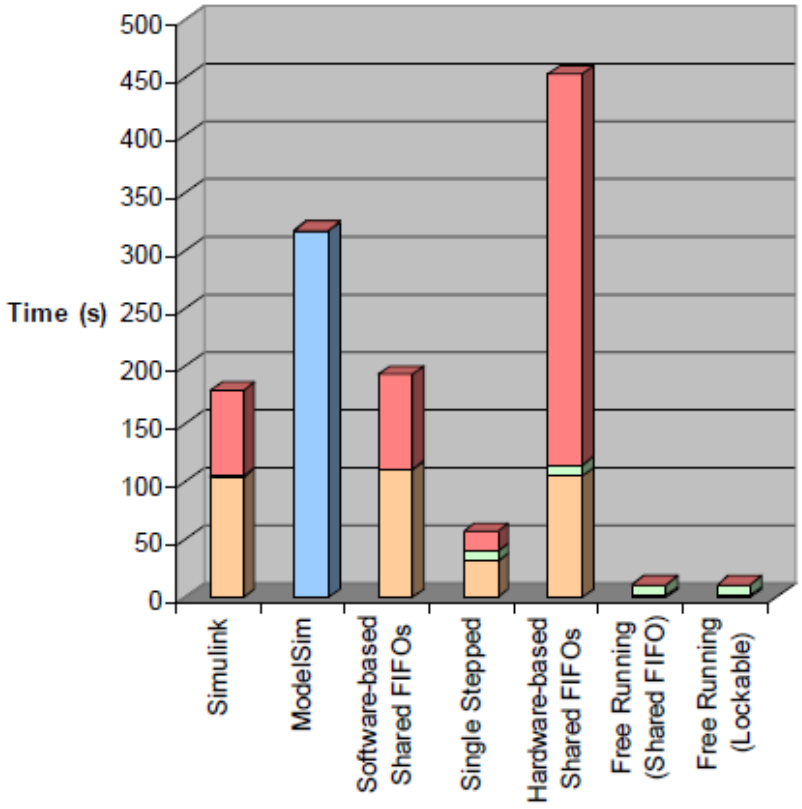

Figure 2.3

2.2 Shared Memories

The Xilinx Shared Memory block implements a random access memory (RAM) that can be shared among multiple designs or sections of a design. System Generator's hardware co-simulation interfaces allow shared memory blocks and shared memory block derivatives (e.g., Shared FIFO and Shared Registers) to be compiled and co-simulated in FPGA hardware. These interfaces make it possible for hardware-based shared memory resources to map transparently to common address spaces on the host PC. When applied to System Generator co-simulation hardware, shared memories can help facilitate high-speed data transfers between the host PC and FPGA, and further bolster the tool's real-time hardware co-simulation capabilities. This topic describes how shared memories can be used within the context of System Generator's hardware co-simulation framework.

A Shared Memory Block is uniquely identified by its name. In the blocks above, the shared memory has been named "Bar". Instances of Shared Memory "Bar", whether within the same model or in different models or even different instances of MATLAB, will share the same memory space. System Generator's hardware co-simulation interfaces allow shared memory blocks to be compiled and co-simulated in FPGA hardware. These interfaces make it possible for hardware-based shared memory resources to map transparently to common address spaces on a host PC. When used in System Generator co-simulation hardware, shared memories facilitate high-speed data transfers between the host PC and FPGA, and bolster the tool's real-time hardware co-simulation capabilities.

2.2.1 Block Parameters

- **Shared memory name:** name of the shared memory. All memories with the same name share the same physical memory. •
- **Depth:** specifies the number of words in the memory. The word size is inferred from the bit width of the data port din.
- **Ownership and initialization:** indicates whether the memory is Locally owned and initialized or Owned and initialized elsewhere. If the memory is locally owned and initialized, the Initial value vector parameter is made available. A block that is Locally owned and initialized is responsible for creating an instance of the memory. A block that

is Owned and initialized elsewhere attaches itself to a memory instance that has already been created. As a result, if two shared memory blocks are used in two different models during simulation, the model containing the Locally owned and initialized block has to be started first.

- **Initial value vector:** specifies initial memory contents. The size and precision of the elements of the initial value vector are inferred from the type of the data samples that drive din. When the vector is longer than the RAM, the vector's trailing elements are discarded. When the RAM is longer than the vector, the RAM's trailing words are set to zero. The initial value vector is saturated and rounded according to the precision specified on the data port din.
- **Access protection:** either Lockable or Unprotected. An unprotected memory has no restrictions concerning when a read or write can occur. In a locked shared memory, the block can only be written to when granted access to the memory. When the grant port outputs a 1, access is granted to the memory and the write request can proceed.
- **Access mode:** specifies the way in which the memory is used by the design. When Read and write mode is used, the block is configured with din and dout ports. When Read only mode is used, the block is configured with a dout port for memory read access. When Write only mode is used, the block is configured with a din port for memory write access.
- **Write mode:** specifies the memory behavior to be Read after write, read before write, or No read on write. There are device specific restrictions on the applicability of these modes.
- **Memory access timeout (sec):** when the memory is running in hardware, this specifies the maximum time to wait for the memory to respond to a request.

Chapter 3

Fast Fourier Transform

3.1 Introduction

The Xilinx LogiCORE™ IP Fast Fourier Transform (FFT) implements the Cooley-Tukey FFT algorithm, a computationally efficient method for calculating the Discrete Fourier Transform (DFT). For this project we are specifically using Fast Fourier Transform v7.1.

3.2 Features of Fast Fourier Transform v7.1

- Drop-in module for Kintex™-7, Virtex®-7, Virtex®-6, Virtex-5, Virtex-4, Spartan®-6, Spartan-3/XA, Spartan-3E/XA and Spartan-3A/XA/AN/3A DSP FPGAs
- Forward and inverse complex FFT, run-time configurable
- Transform sizes $N = 2m$, $m = 3 – 16$
- Data sample precision $bx = 8 – 34$
- Phase factor precision $bw = 8 – 34$
- Arithmetic types:
  - ¨ Unscaled (full-precision) fixed-point
  - ¨ Scaled fixed-point
  - ¨ Block floating-point
- Fixed-point or floating-point interface
- Rounding or truncation after the butterfly
- Block RAM or Distributed RAM for data and phase-factor storage
- Optional run-time configurable transform point size
- Run-time configurable scaling schedule for scaled fixed-point cores
- Bit/digit reversed or natural output order

- Optional cyclic prefix insertion for digital communications systems
- Four architectures offer a trade-off between core size and transform time
- Bit-accurate C model and MEX function for system modeling available for download
- For use with Xilinx CORE Generator™ software and Xilinx System Generator for DSP v13.1

3.3 Overview

The FFT core computes an N-point forward DFT or inverse DFT (IDFT) where N can be 2m, m = 3–16. For fixed-point inputs, the input data is a vector of N complex values represented as dual bx-bit two's-complement numbers, that is, bx bits for each of the real and imaginary components of the data sample, where bx is in the range 8 to 34 bits inclusive.

Similarly, the phase factors bw can be 8 to 34 bits wide. For single-precision floating-point inputs, the input data is a vector of N complex values represented as dual 32-bit floating-point numbers with the phase factors represented as 24- or 25-bit fixed-point numbers.

All memory is on-chip using either block RAM or distributed RAM. The N element output vector is represented using by bits for each of the real and imaginary components of the output data. Input data is presented in natural order and the output data can be in either natural or bit/digit reversed order. The complex nature of data input and output is intrinsic to the FFT algorithm, not the implementation.

Three arithmetic options are available for computing the FFT:

- Full-precision unscaled arithmetic
- Scaled fixed-point, where the user provides the scaling schedule
- Block floating-point (run-time adjusted scaling)

The point size N, the choice of forward or inverse transform, the scaling schedule and the cyclic prefix length are run-time configurable. Transform type (forward or inverse), scaling schedule and cyclic prefix length can be changed on a frame-by-frame basis. Changing the point size resets the core. Four architecture options are available: Pipelined, Streaming I/O, Radix-4, Burst I/O, Radix-2, Burst I/O, and Radix-2 Lite, Burst I/O. For detailed information about each architecture

3.4 Architecture

The FFT core provides four architecture options to offer a trade-off between core size and transform time.

- **Pipelined, Streaming I/O –** Allows continuous data processing.
- **Radix-4, Burst I/O –** Loads and processes data separately, using an iterative approach. It is smaller in size than the pipelined solution, but has a longer transform time.
- **Radix-2, Burst I/O –** Uses the same iterative approach as Radix-4, but the butterfly is smaller. This means it is smaller in size than the Radix-4 solution, but the transform time is longer.
- **Radix-2 Lite, Burst I/O –** Based on the Radix-2 architecture, this variant uses a time-multiplexed approach to the butterfly for an even smaller core, at the cost of longer transform time.

Figure 3.1, illustrates the trade-off of throughput versus resource use for the four architectures. As a rule of thumb, each architecture offers a factor of 2 difference in resource from the next architecture. The example is for an even power of 2 point size. This does not require the Radix-4 architecture to have an additional Radix-2 stage.

All four architectures may be configured to use a fixed-point interface with one of three fixed-point arithmetic methods (unscaled, scaled or block floating-point) or may instead use a floating-point interface.

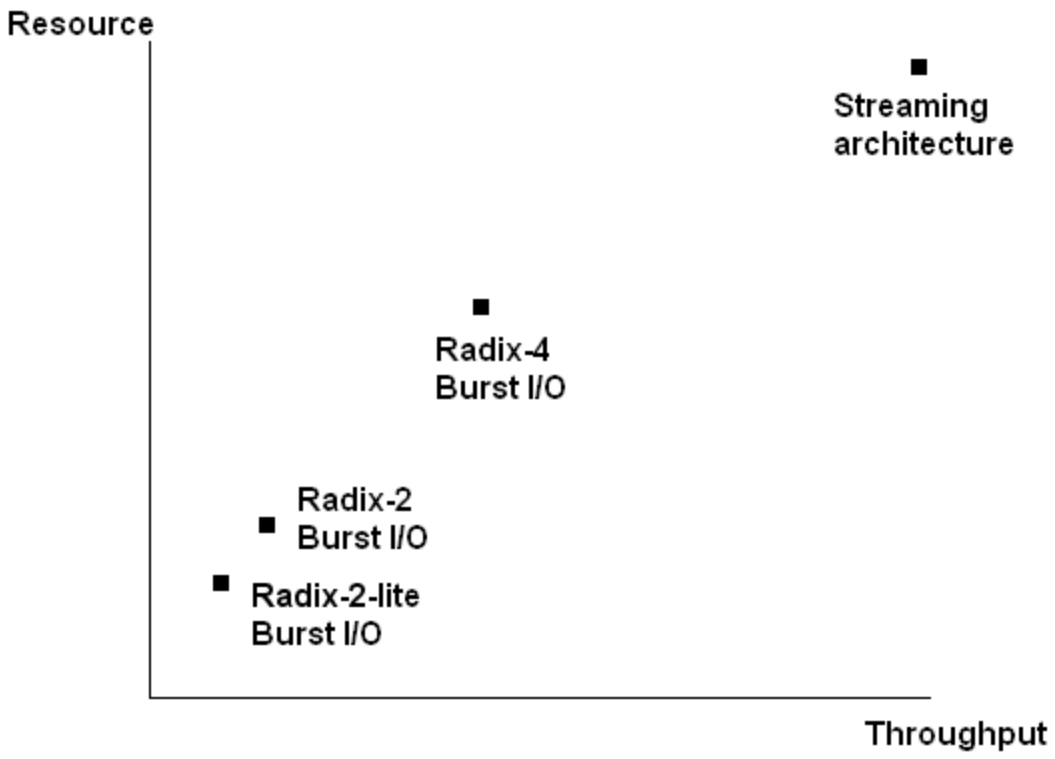

Figure 3.1

3.5 Pipelined, Streaming I/O

The Pipelined, Streaming I/O solution pipelines several Radix-2 butterfly processing engines to offer continuous data processing. Each processing engine has its own memory banks to store the input and intermediate data (Figure). The core has the ability to simultaneously perform transform calculations on the current frame of data, load input data for the next frame of data, and unload the results of the previous frame of data. The user can continuously stream in data and, after the calculation latency, can continuously unload the results. If preferred, this design can also calculate one frame by itself or frames with gaps in between.

In the scaled fixed-point mode, the data is scaled after every pair of Radix-2 stages. The block floating-point mode may use significantly more resources than the scaled mode, as it must maintain extra bits of precision to allow dynamic scaling without impacting performance. Therefore, if the

input data is well understood and is unlikely to exhibit large amplitude fluctuation, using scaled arithmetic (with a suitable scaling schedule to avoid overflow in the known worst case) is sufficient, and resources may be saved.

The input data is presented in natural order. The unloaded output data can either be in bit reversed order or in natural order. When natural order output data is selected, additional memory resource is utilized.

This architecture covers point sizes from 8 to 65536. The user has flexibility to select the number of stages to use block RAM for data and phase factor storage. The remaining stages use distributed memory.

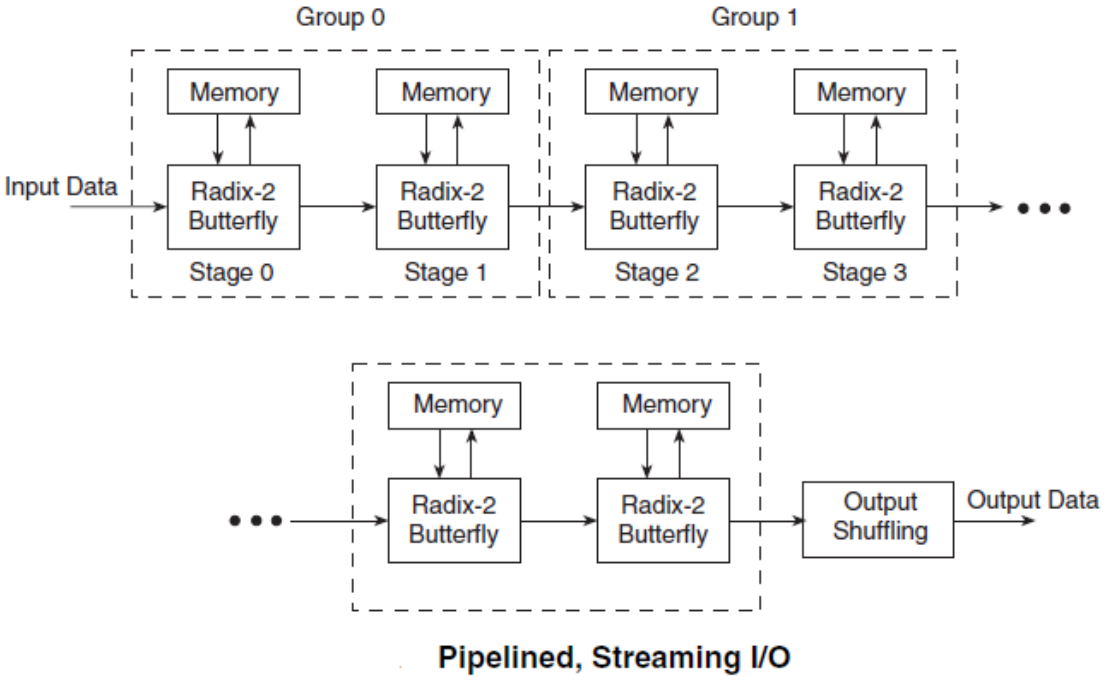

Figure 3.2

Chapter 4

Adaptive Fourier Spectral Segmentation (AFSS)

Owing to the limitations of FFT and DWT, a new method of AFSS is proposed.

In FFT, the time information of the signal is lost. We can find out what frequencies are present in a signal but we cannot locate them, we cannot find out what frequencies are present at what time. Though this information is quite represented in DWT owing to the localized and short duration of mother wavelet which is able to represent localised features but the DWT is not adaptive. In DWT, once we fix the sampling frequency, the filter design is fixed that is the segmentation becomes predefined for the signal. As a result, it may suit quite well for some signals but may not be quite for another. Now, AFSS overcomes these limitations by restructuring the filter design each time for every new signal. The boundaries or the segmentations in the spectrum are calculated separately each time for new signal which accounts for minimum spectral leakage and minimum overlapping of two frequencies on each other. Consequently, the error in the estimation in the PQIs reduces and hence the term ADAPTIVE.

The procedure followed by Adaptive Fourier Spectral Segmentation is as follows:

Step 1) Determine the frequency components of the applied signal using FFT. 20

Step 2) Calculate the number of peaks (maxima) occurring by applying constraints of magnitude threshold (3% of the fundamental frequency magnitude) and frequency distance threshold.

Step 3) Now, find out the minima between these peaks and the location of these minima.

Step 4) Segment the spectrum on the lines of these frequencies at which minima occurs.

Step 5) Now, calculate the PQIs from this segmented spectrum. FFT-based indices can be used as calculations are done in frequency domain itself.

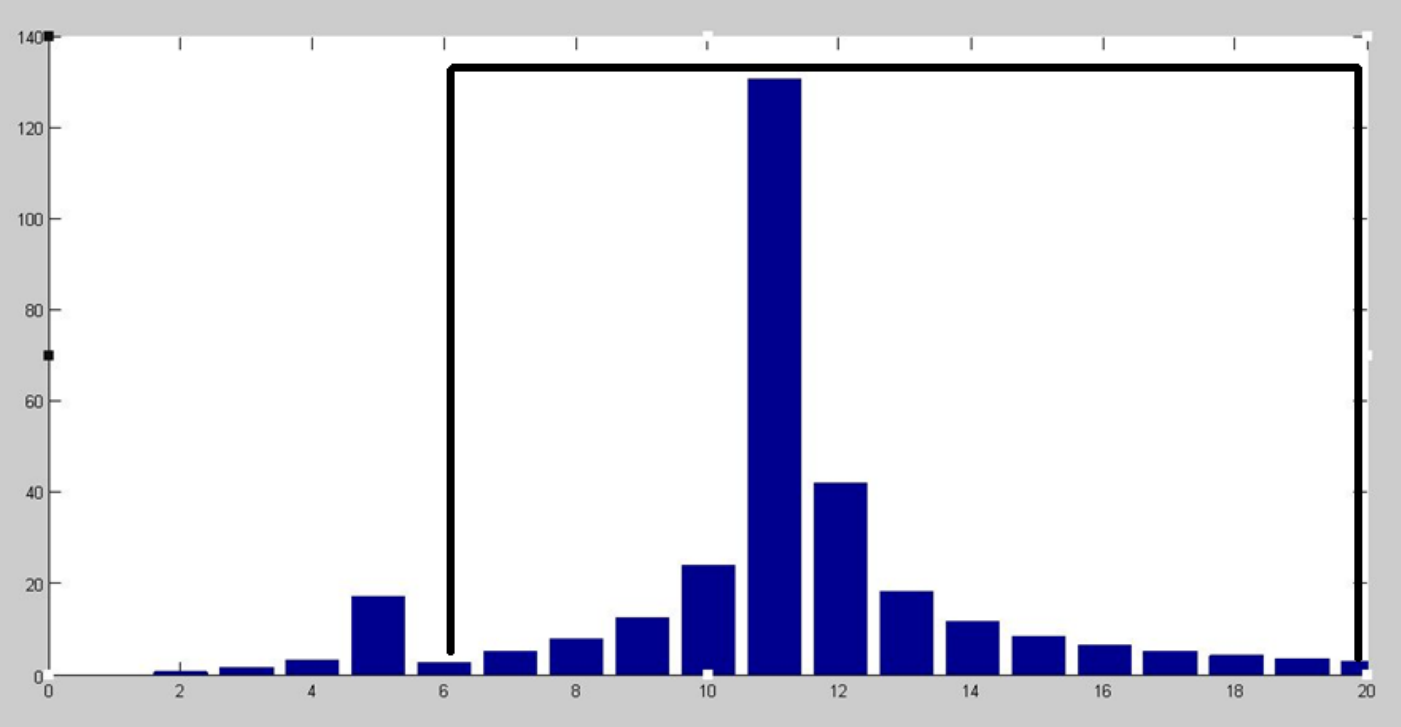

Figure 4.1

In our case for finding harmonic frequency (50Hz, the 11^{th} sample) we only consider the spectral components between the local minima on either sides of 50 Hz and group them together to get the 50 Hz component.

In system generator to calculate AFSS, we used three memories in which all stored same values in same address locations. First we started with peak value (50Hz) then comparing it with the next value. If the next value is greater than the peak value, then the process will be stopped.

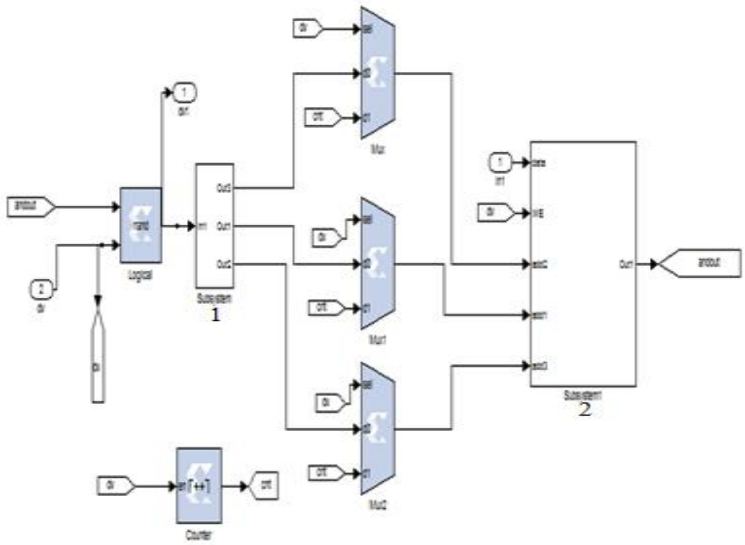

Figure 4.2

Spectral components between the local minima on either sides of 50 Hz are grouped together to get the 50 Hz component and stored.

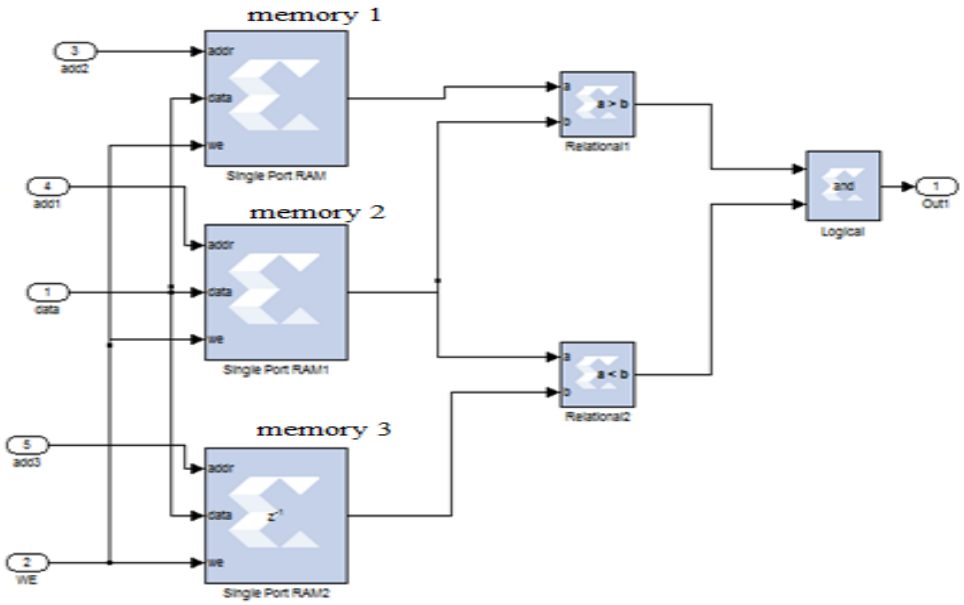

Figure 4.3

Chapter 5

Design

This section of the document gives the detailed description of the model(fig-), which is both Online and Real Time, designed to implement IEC-61000-4-7 standards using FFT. The design presented here extensively uses **Lockable Shared memories** to design the online buffer. Shared memories are copies of same memory in different parts of design. Lockable shared memories have an added advantage of mutual exclusion over shared memories. We use two different pairs shared memories; one for input buffer(mem_in) and other for output buffer(mem_out). The design uses FFT v 7.1 block configured for our requirements. The design is explained in three stages:

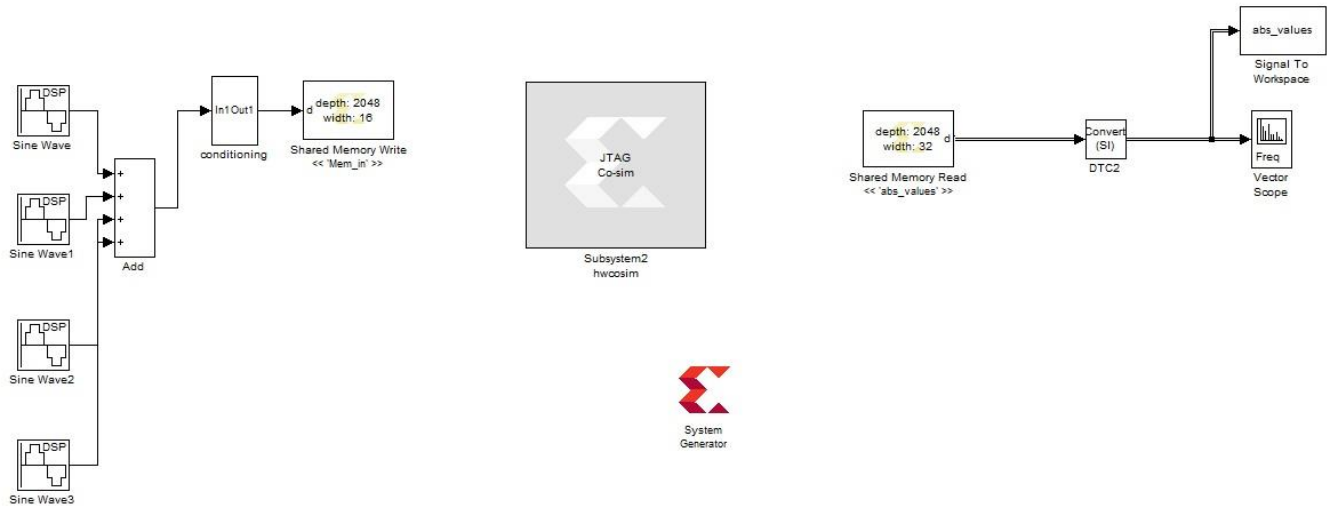

Figure 5.1

To accelerate using vector transfers, we included input buffer storage in the FPGA that can store data input samples that are written by the PC. An output buffer is also included so that the processed data values can be stored while the FPGA waits for the PC to retrieve them.

5.1 Pre -conditioning

This stage of the design contains source, conditioning subsystem (a I/p buffer and a data type converter) showed in fig-4 and a copy of shared memory "mem_in" (shared memory write) that communicates with FPGA. The DSP sinewave block is used as source and it is configured to 50Hz, with amplitude of 0.5 and is sampled at 10KHZ. The data converter converts the input data to fixdt (1,32,31) and then reinterprets to RWV (Real World Values) of type fixdt (1,32,0) this conversion is must because the lockable shared memories can only take int32, int16, int12, and int8 formats. The next stage involves buffer of 2000 samples using buffer block and then followed by padding with 48 zeros. The shared memory write block and its copy on FPGA are to set to a depth 2048 and a data width of 32 bits. The shared memory read block is set to priority 1, i.e. at the start of simulation the shared memory read block will have the lock and its copy on FPGA lay inactive. The shared memory read holds the lock till the requisite number of samples are not available i.e. 2048 in our case. Once the required number of samples are available i.e. after 0.2 s (padding happens in no time), it communicates with the input buffer and stores the data present in the buffer and releases the lock to its copy (mem_in) on FPGA, till this point the system operates at 10KHZ clock and FPGA is not active. The next step is vector transfer of 2048 data points in one cycle to mem_in on FPGA (high speed transfer). This process described above happens for every 0.2 seconds. The input buffer is designed with some additional input logic for manipulating the data flow

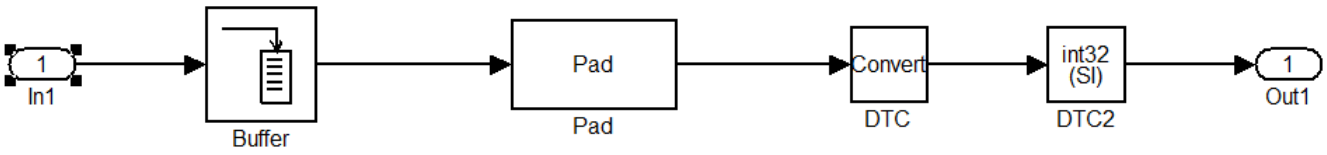

Figure 5.2

5.2 Input buffer

The system on FPGA initially lock is with the input shared memory(Mem_in) it can now access input data. Starts storing the 2048 data vales are stored in 2048 address after storing all data then releases dout_valid. For successive operations pulse(obuf_grant_pulse) is given, so that after algorithm calculation it start loading the next set of data.

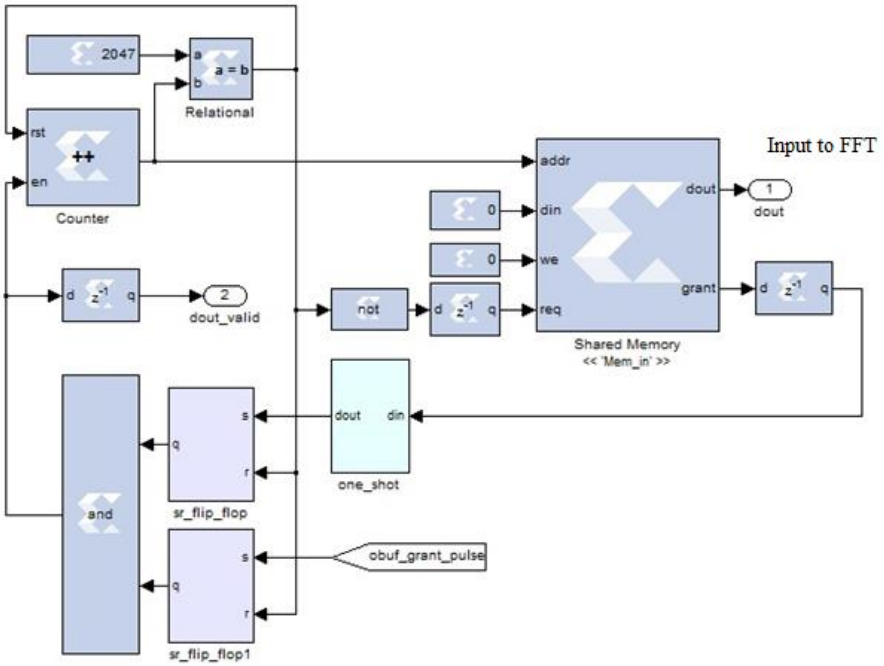

Figure 5.3

5.3 On-chip system

The system that will on the FPGA is shown below in figure. The FPGA is set to priority 2, i.e. all the shared memories in FPGA only work, when the shared memories with priority 1 releases lock, in our case it is shared memory write (mem_in), once the lock is released the memory copy of mem_1 in FPGA is active and can perform read/write operation as instructed. There are two output port: 1) d-out 2) d_valid (high when useful data is available). The data from mem_1 is fed to FFT for calculation of DFT using pipelined streaming algorithm. The mem_1 on FPGA outputs a single data point on every positive clock edge, i.e. it takes 2048 clock periods to feed the data to FFT block.

The dv signal is used as data_valid for the shared memory mem_2, which forms an integral part of output buffer systems. The FFT outputs data once the calculation is done, this data is fed to mem_2 as it has a priority 2 it possesses the lock over its copy i.e. shared memory read which is set to priority 3. Once it has requisite amount of data it passes the lock to shared memory read (mem_2) and stops being active. In above stage the system works with FPGA clock i.e. 50MHz.

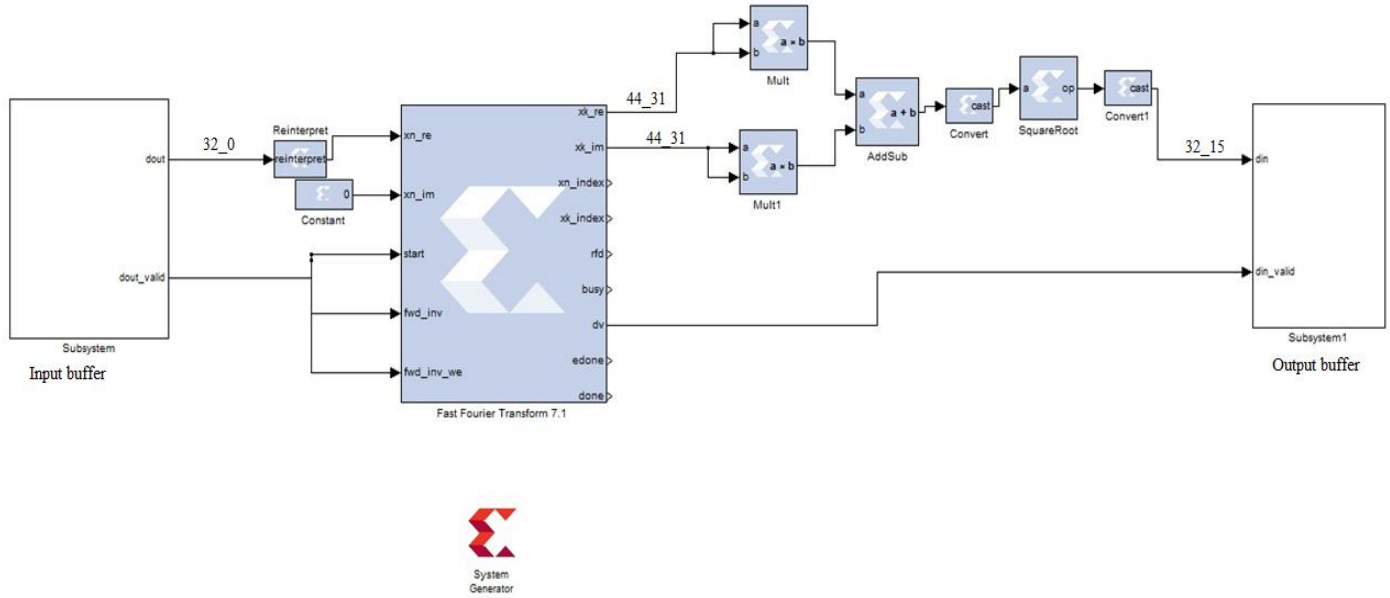

Figure 5.4

5.4 Output buffer

Once the lock is passed to shared memory read, the FPGA communicates the data stored in the mem_2 to it and later this stage shared memory read (mem_2) outputs the data at the rate 10KHz and the process repeats, the entire process described above literally takes place in very less time is the calculations are performed at 50MHz clock. The data that the source outputs is kept on buffering, and the time taken to calculate is very less than 0.2 seconds, the system is very well ready for calculating next 2000 samples.

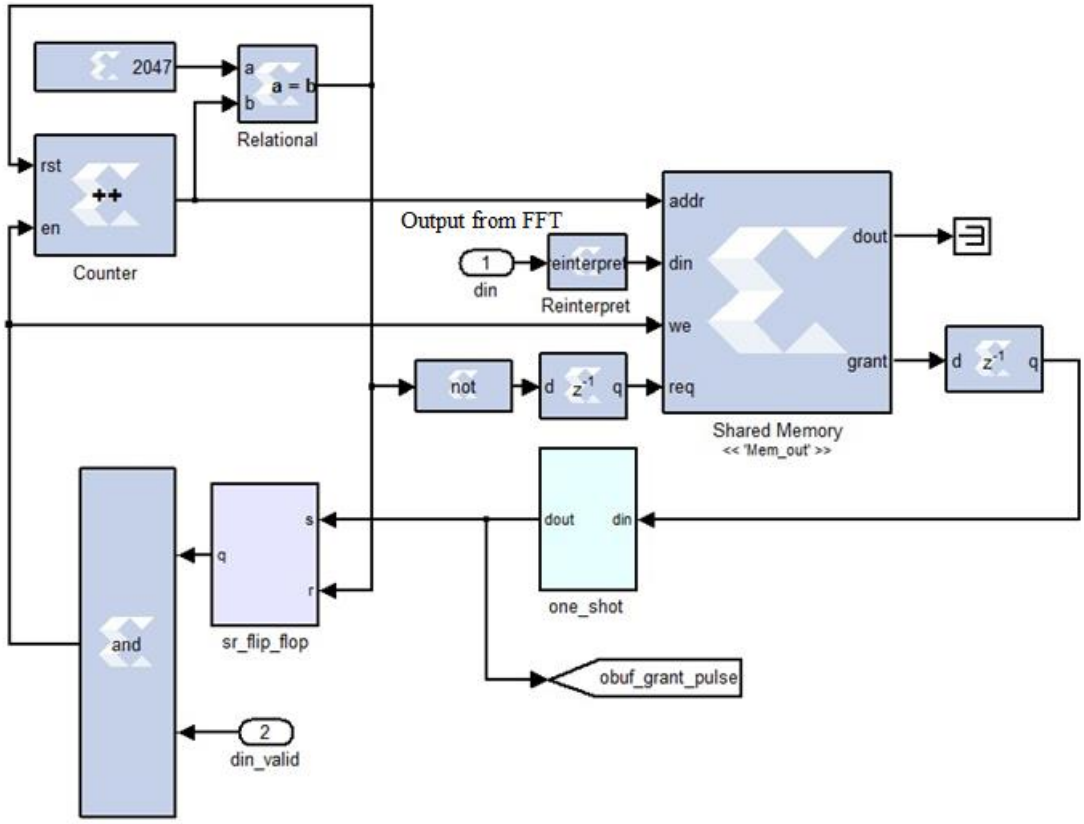

Figure 5.5

CHAPTER 6

Results and Discussion

Separating the harmonic signal (50Hz) from a set of four different signals 20Hz, 50Hz, 150Hz, 350Hz with magnitude 0.01414, 0.1414, 0.01414,0.01414 respectively

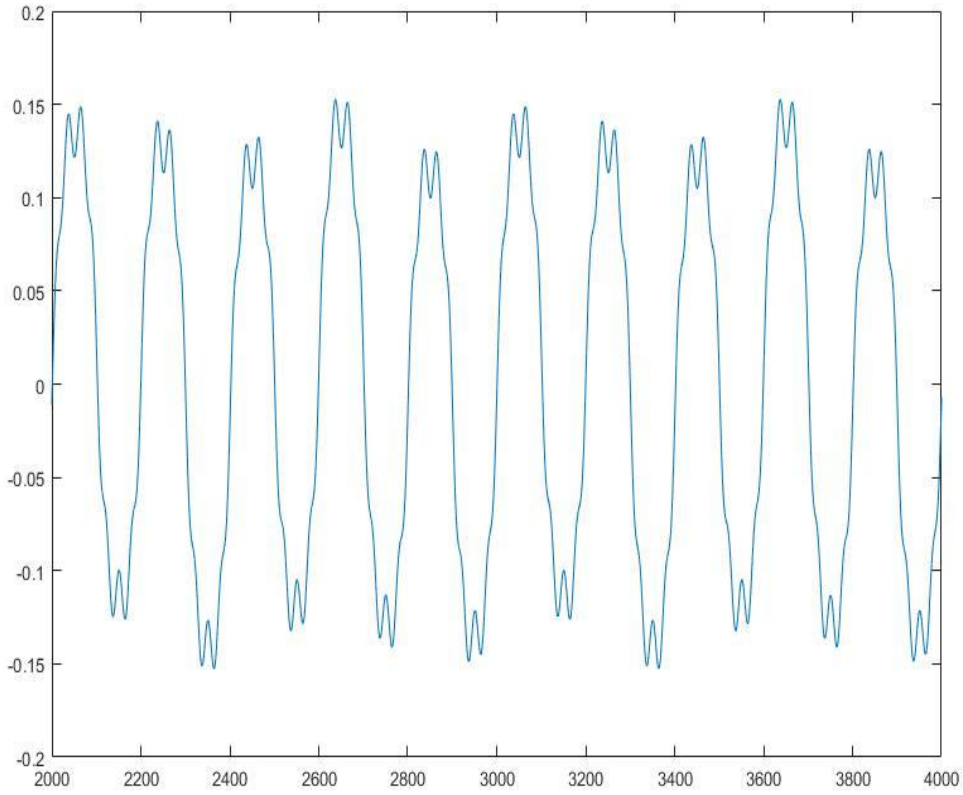

Figure 6.1

Below figure 6.2 is obtained by performing FFT (2048 samples taken) on above graph (fig 6.1).

Figure 6.2, we can observe that there are four peaks at 20Hz, 50Hz, 150Hz, 350Hz. Height of local maxima's is proportional to signal magnitude, it represents 50Hz has grater magnitude than rest of them.

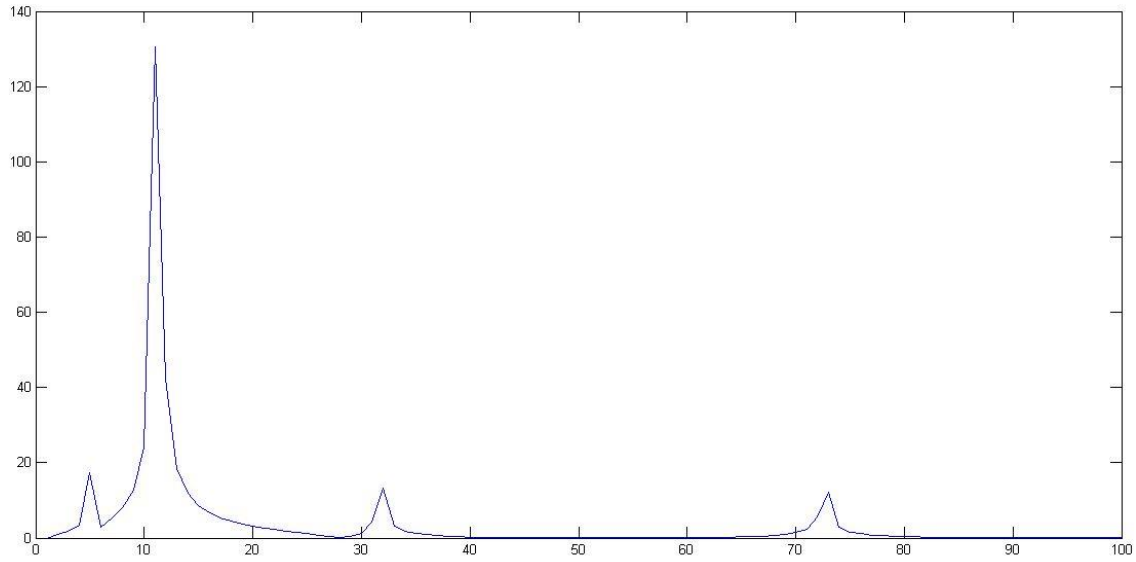

Figure 6.2

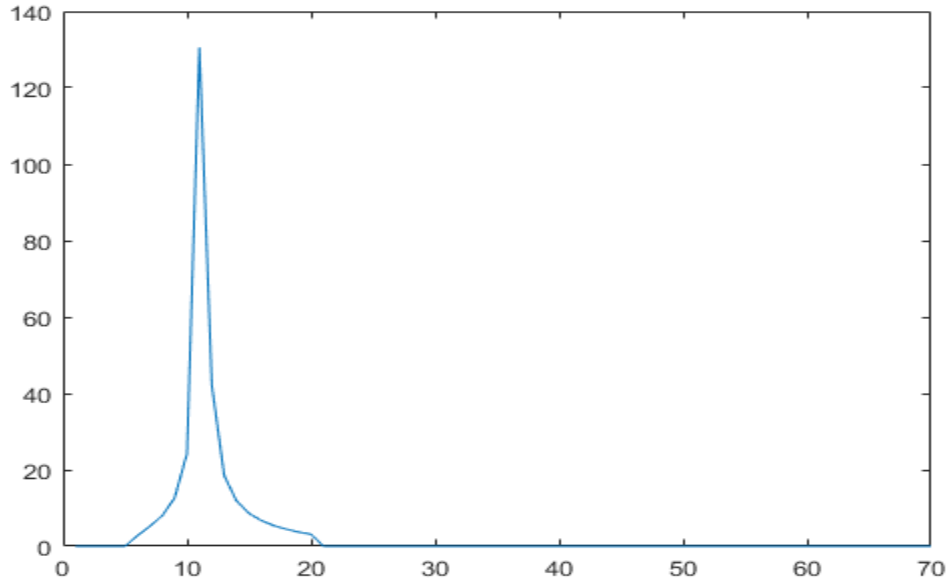

Figure 6.3

Above figure 6.3 is obtained by performing Adaptive Fourier Segmentation on graph of FFT (fig 6.2).

Figure 6.3, shows that local minima's occurs at 6^{th} and 20^{th} samples. To get 50Hz components from figure 6.2, we grouped spectral components of 25Hz(6^{th}) to 95Hz (20^{th})

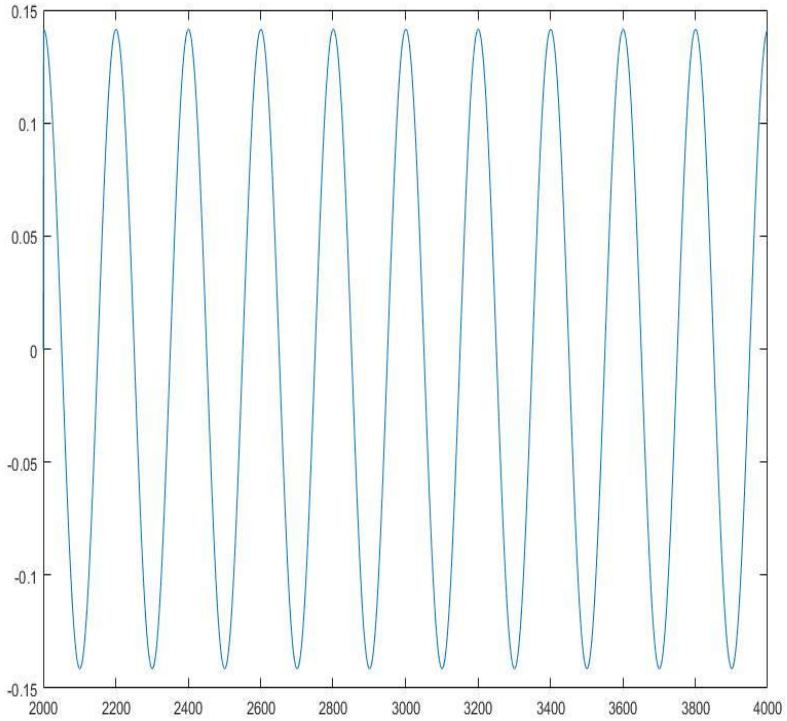

figure 6.4

We can observe that above figure is obtained by IFFT of fig 6.3

Simulink Profile report shown in Figure 6.5, it says how many times a block is called and time taken to perform operation on FPGA. Process is completed in 0.218s

untitled1 (MajorOutput) untitled1
Time: 0.21840140 s (14.0%)
Calls: 10001
Self time: 0.12480080 s (14.0%)

Function:	Time		Calls	Time/call
untitled1 (MajorOutput)	0.21840140		10001	0.0000218379562
Parent functions:				
Execute			10001	
Child functions:				
untitled1/Scope (Output)	0.04680030	21.4%	10001	0.0000046795620
untitled1/conditioning/Buffer (Output)	0.03120020	14.3%	10001	0.0000031197080
untitled1/Shared Memory Write (Output)	0.01560010	7.1%	6	0.0026000166667
untitled1/Unbuffer (Output)	0.00000000	0.0%	10001	0.0000000000000
untitled1/Shared Memory Read (Output)	0.00000000	0.0%	5	0.0000000000000
untitled1/Subsystem2 hwcosim (Output)	0.00000000	0.0%	6	0.0000000000000
untitled1/conditioning/in_data_conv/DTC (Output)	0.00000000	0.0%	6	0.0000000000000
untitled1/conditioning/Pad (Output)	0.00000000	0.0%	6	0.0000000000000
untitled1/Sine Wave (Output)	0.00000000	0.0%	10001	0.0000000000000

Figure 6.5

CHAPTER 7

Conclusion

1. Adaptive Fourier Spectral Segmentation is able to analyze all sort of disturbances resulting in accurate estimation of PQIs.
2. System is Online and Real time because values are taken from online and calculated in real time.
3. The fundamental component extracted is accurate and reflects the disturbances if any
4. Implementation of this on the FPGA reduced the computation time and made it suitable for online monitoring.

References

1. Applications of the Windowed PFT to Electric Power Quality Assessment, G. T. Heydt P. *S.* Fjeld C. C. Liu D. Pierce L. Tu *G.* Hensley
2. Refined power quality indices, M.S.Kandil, S.A.Farghal and A.Elmitwally
3. Reformulating Power Components Definitions Contained in the IEEE Standard 1459 2000 Using Discrete Wavelet Transform, Walid G. Morsi, *Student Member, IEEE*, and M. E. El- Hawary, *Fellow, IEEE*
4. IEEE Standard Definitions for the Measurement of Electric Power Quantities under Sinusoidal, Non-sinusoidal, Balanced and Un-balanced conditions.
5. IEC Recommended Practice for Monitoring Electric Power Quality
6. IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems
7. IEC 61000-4-7 Recommended to instrumentation intended to measuring spectral components in frequency range up to 9kHz.
8. Xilinx user guide
9. System Generator for DSP Getting Started Guide, UG639 (v11.4) December 2, 2009
10. © Copyright 2003 - 2011 Xilinx, Inc. XILINX, the Xilinx logo, Kintex, Virtex, Spartan, ISE and other designated brands included herein are trademarks of Xilinx in the United States and other countries. MATLAB is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.