DYNAMIC VOLTAGE RESTORER FOR SENSITIVE EQUIPMENT

by

YEA ZONG KUO

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2007

To the memory of my grandmother, who passed away in 2005.

ACKNOWLEDGEMENTS

This dissertation and its related works would not be possible without the support and encouragement of many family and friends. I want to extend my appreciation to my supervising professors (Dr. W. Lee and Dr. S. Oraintara) and committee members (Dr. J. Bredow, Dr. B. Gou, Dr. L. Peterson, and Dr. R. Shoults) for their participation. I especially want to offer my sincere appreciation to Dr. Wei-Jen Lee for his guidance throughout this journey.

Finally I want to thank my mom and dad, my brother, and my husband for their patient support. All this would not be meaningful without them.

March 7, 2007

ABSTRACT

DYNAMIC VOLTAGE RESTORER FOR SENSITIVE EQUIPMENT

Publication No. _____

Yea Zong Kuo, PhD.

The University of Texas at Arlington, 2007

Supervising Professor: Wei-Jen Lee

Co-Supervising Professor: Soontorn Oraintara

Voltage sags are considered to be the most frequent type of disturbances in the field based on recent power disturbances studies. Their impact on sensitive loads is severe. The impact ranges from load disruptions to substantial economic losses.

Despite the technical advances in electronics, there are still some pieces of equipment that are so sensitive that they are unable to withstand voltage sags. An example is the equipment in semiconductor manufacturing. As a result, there is still a search for different ways to manage voltage sags.

There are many different methods to mitigate voltage sags, but a CUstom Power Supply (CUPS) device is considered to be the most efficient method. There are many types of CUPS. This dissertation studied various methods and concluded that the Dynamic Voltage Restorer (DVR) is the most efficient and effective device to protect sensitive equipment against voltage sags. Its appeal includes lower cost, smaller size, and its dynamic response to the disturbance.

There are two general types of DVRs: Minimal Energy Control DVRs and In-Phase Control DVRs. One includes energy storage, and the other does not include energy storage, respectively. This dissertation focuses on the In-Phase Control DVRs, the one without energy storage. This dissertation describes the DVR from the initial phase of research and analysis to the final phase of testing and constructing the device.

The DVR described in this dissertation consists of many different features. One feature is that the filter used in the harmonics correction is based on the Wavelet Transform (WT). The selection of the WT based filter is different from conventional methods such as the Fourier Transform (FT) in other DVRs. Another feature of the dissertation DVR is the hardware implementation. An FPGA board was selected to implement the dynamic controls instead of the traditional DSP board. Yet another feature is the power factor correction that was added later to the design in order to further enhance the dissertation DVR's overall capabilities. These features differentiate the dissertation DVR from other types of DVRs.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

BACKGROUND

Based on recent studies of power disturbances, voltage sags are considered to be the most frequent type of disturbance in the field [1-13]. Their impact on sensitive loads is severe [14]. The impact includes disruption to the load and substantial economic losses[8].

Voltage sag is defined by IEEE std 1159-1995 as a decrease in Root Mean Square (RMS) voltage to between 0.1 and 0.9 Per Unit (PU) for a duration of between half a cycle and less than 1 minute [13, 15, 16]. In other words, if the voltage falls below 90% of nominal for less than a minute, this fluctuation is known as a voltage sag or dip [17]. It is primarily caused by frequent power system faults in the transmission and distribution level [16, 18]. An example of a voltage sag is illustrated in figure 1.1. The figure shows a voltage sag of 0.8 pu for three cycles.

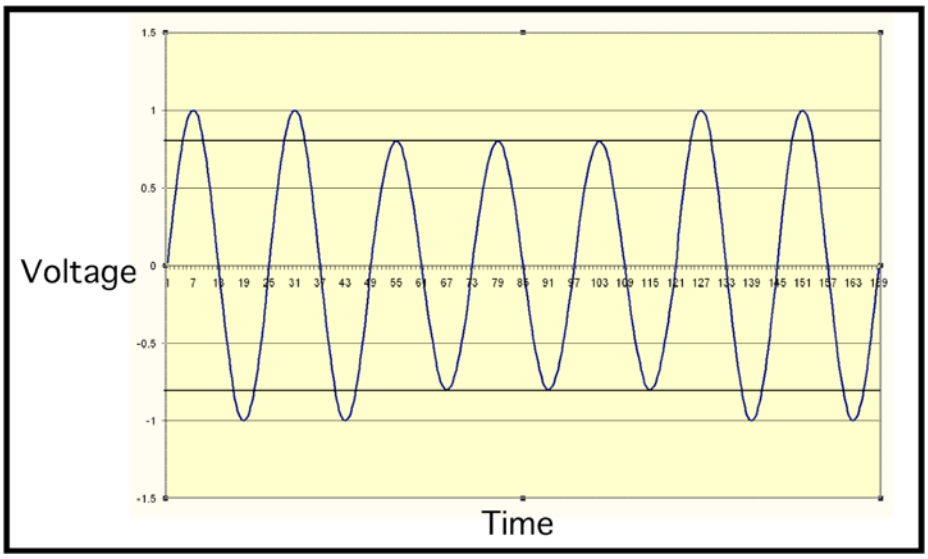


Figure 1.1 Voltage Sag

In an effort to address power disturbances such as voltage sags on electronic equipment, a general guidance for equipment tolerances to power disturbances is established by the Information Technology Industry Council (ITIC) curve. The ITIC curve, which is formerly known as the Computer Business Equipment Manufacturers Association (CBEMA) curve, illustrates the equipment tolerances in terms of percent nominal voltage versus duration. Originally, the ITIC curve defined the tolerance of main frame computer equipment to withstand disturbances in terms of voltage magnitude and duration, hence its former name Computer Business Equipment Manufacturers Association [16]. This curve has been adopted by other industries and has been evolved into a standard for measuring power quality performance for a variety of electronic equipment. The ITIC curve is shown in figure 1.2. To understand the region of interest in this dissertation, the voltage sag region is overlaid in blue on the ITIC curve figure.

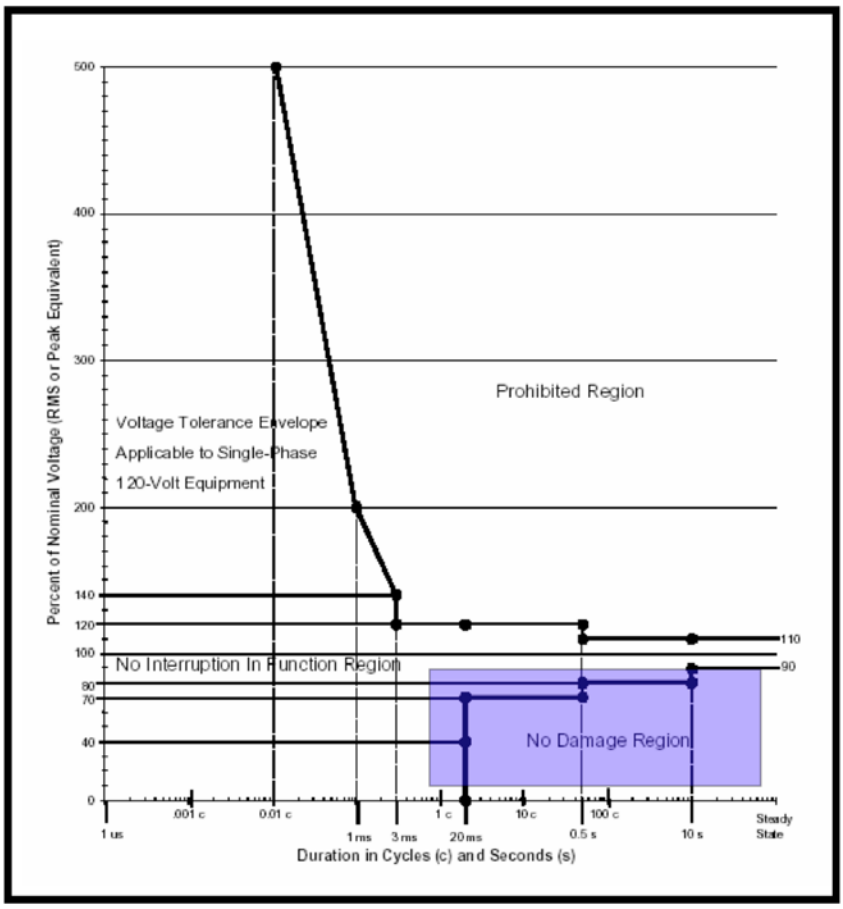


Figure 1.2 ITIC Curve [19]

Figure 1.2 shows that much of the voltage sag region (shown in blue) falls within the “No Damage Region” of the ITIC curve. Although equipment conforming to the curve may survive operation in the “No Damage Region”, sensitive pieces of equipment such as those in industrial facilities are susceptible to voltage sags, accounting for a large percentage of power quality events at these types of facilities. This is shown in figure 1.3. The red dots in the figure represent voltage events collected at one hundred large manufacturing plants in the United States over a period of one year [8]. The blue dashed lines in the figure outline the ITIC curve.

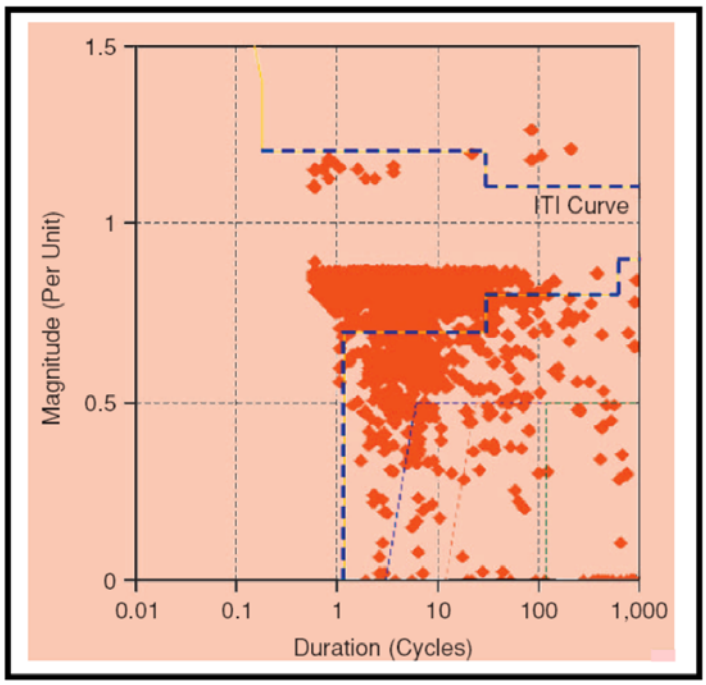


Figure 1.3 Voltage Events At Manufacturing Plants [8]

The impact of voltage sags on sensitive equipment varies. It ranges from load disruption to equipment damage. It has been found that in some industrial equipment, the current surge which occurs after a voltage sag is the actual cause of the equipment damage [8].

The impact of voltage sags on sensitive equipment is not limited to the load or the equipment alone. Usually, these pieces of equipment are an integral part of an industrial or manufacturing process, such that any disruption to the process may cost extensive down time and repairs. An example of such a sensitive process is the area of semiconductor fabrication and manufacturing [20]. The substantial economic loss as a result of this disruption is summarized in figure 1.4 [21]. According to this figure, the

semiconductor industry suffered an average loss in the range of $100K to $10M due to each voltage sag.

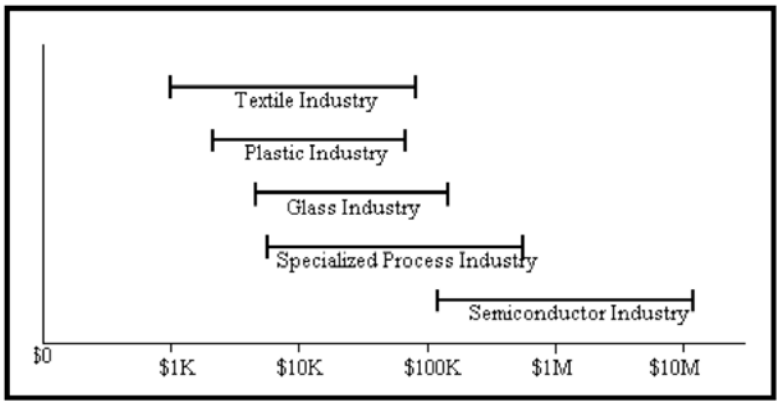


Figure 1.4 Average Loss Per Sag For Different Industries [21]

In summary, voltage sags account for a significant percentage of power disturbances, but they are not addressed in the ITIC curve in terms of tolerance levels. This is because they have minimal impact on general electronic equipment. Their greatest impact is on sensitive equipment, and the impact ranges from load disruptions to economic losses. To overcome this, it is evident that there is a need for methods to mitigate voltage sags.

There are many different methods to mitigate voltage sags, but a Custom Power Supply (CUPS) device is considered to be the most efficient method. This dissertation concludes that the CUPS device known as the Dynamic Voltage Restorer (DVR) is the most efficient and effective device to protect sensitive equipment against voltage sags. The DVR described in this dissertation consists of many different features, including harmonics correction, power factor correction, and FPGA hardware implementation. Details of these features are provided in the following chapters.

CHAPTER 2

DVR INTRODUCTION

2.1 Selection

There are two general approaches to mitigating voltage sags at the load end. The first approach is to ensure that the equipment is less sensitive to disturbances such as voltage sags. This first approach could be achieved by design or simply by acquiring Commercial Off-The-Shelf (COTS) equipment with disturbance tolerances. The second approach is to provide a CUstom Power Supply (CUPS) device that counteracts the disturbance(s).

Since most pieces of COTS equipment are designed to conform to the ITIC curve, CUPS devices are the best approach to further dealing with voltage sags for sensitive equipment. There are many types of CUPS devices. Some of these devices include: Active Power Filters (APF), Battery Energy Storage Systems (BESS), Distribution STATic synchronous COMpensators (DSTATCOM), Distribution Series Capacitors (DSC), Dynamic Voltage Restorer (DVR), Surge Arresters (SA), Super conducting Magnetic Energy Systems (SMES), Static Electronic Tap Changers (SETC), Solid-State Transfer Switches (SSTS), Solid State Fault Current Limiter (SSFCL), Static Var Compensator (SVC), Thyristor Switched Capacitors (TSC), and

Uninterruptible Power Supplies (UPS) [2, 6]. (Some of these acronyms are repeated throughout this dissertation, so an acronym list is compiled and listed in appendix A.)

Each of these CUPS devices has its own benefits and limitations. The most effective type of CUPS device is considered to be the Dynamic Voltage Restorer (DVR). There are numerous reasons why the DVR is preferred over other CUPS devices. A few of these reasons are presented as follows. The SVC pre-dates the DVR, but the DVR is still preferred because the SVC has no ability to control active power flow [22]. Another reason is that the DVR costs less compared to the UPS [23, 24]. Not only is the UPS costly, it also requires a high level of maintenance because batteries leak and have to be replaced as often as every five years [24]. Other reasons include that the DVR has a higher energy capacity and lower costs compared to the SMES device [22]. Furthermore, the DVR is smaller in size and costs less compared to the DSTATCOM [22].

Based on these reasons, it is no surprise that the DVR is widely considered as an effective custom power device in mitigating voltage sags [25]. Compared to other CUPS devices, the DVR clearly provides the best economic solution for its size and capabilities. As a result of this analysis, the DVR was selected as the device for mitigating voltage sags for sensitive equipment in this dissertation. In addition to the voltage sag mitigation capabilities, the DVR proposed in this dissertation has also provided features to further enhance the DVR's existing capabilities. (The DVR developed as part of this dissertation will be referred to as the "dissertation DVR" throughout this paper.)

2.2 Configuration

The DVR was first installed in 1996 [25]. It is normally installed in a distribution system between the supply and the critical load feeder [15]. Its primary function is to rapidly boost up the load-side voltage in the event of a disturbance in order to avoid any power disruption to that load [24, 26]. This is a daunting task, and there are many ways to accomplish this task. As a result, there are many types of DVRs.

DVRs are broken up into two general categories. One category is known as Minimal Energy Control DVRs, and the other category is known as the In-Phase Control DVRs [6]. The Minimal Energy Control DVRs are DVRs that minimizes active power consumption in the compensators and attempts to compensate with a small energy storage device [6]. The In-Phase Control DVRs are DVRs that inject active power during disturbances without any energy storage devices [6]. In short, the two categories could be summed up as with or without energy storage [17]. The dissertation DVR focuses on the In-Phase Control DVRs, the type without energy storage, and it concentrates on mitigating the power disturbance of voltage sags.

There are many ways of implementing the In-Phase Control DVRs. The general configuration consists of an isolation transformer, a Voltage Source Converter (VSC), and a control mechanism to detect and manage the disturbance. The dissertation DVR adheres to this general configuration.

The isolation transformer in the configuration is a specially designed transformer that attempts “to limit the coupling of noise and transient energy” from the

primary side to the secondary side [27]. In short, the isolation transformer serves the purpose of isolating the load from the system. This is to prevent high-frequency noise and transients from reaching the load, and any load-generated noise and transients are kept from reaching the rest of the system [16]. This is important to the DVR because the digital circuit in the DVR will be switching to compensate the voltage, and the effects from the switching will be limited to the load side by the isolation transformer.

The purpose of the VSC is to inject the compensation voltage. It is primarily a switch such as a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) or an Insulated Gate Bipolar Transistor (IGBT) that serves to convert the voltage source.

The control mechanism of the general configuration typically consists of hardware with programmable logic. In past DVR development, this would normally consist of Digital Signal Processing (DSP) boards. For example, a common DSP board used in past DVR development is the TMS series made by Texas Instruments, such as the TMS320x used in [28], [29], [30], [7], and [31].

The software on the DSP board provides the controls such as detection and correction. Filters are commonly used for these purposes. The type of filter algorithm has varied. It ranges from the Fourier Transform (FT), the Phase-Locked Loop (PLL), to the Wavelet Transform (WT), just to name a few. Although, the Fourier Transform still remains the most common type [32, 33].

CHAPTER 3

DISSERTATION DVR

The dissertation DVR is an extension of the basic DVR configuration. Its aim is to improve on the function of the basic DVR configuration. This is accomplished through modifications of the existing configuration and by adding features to further enhance its capabilities.

3.1 Configuration

The dissertation DVR configuration includes the isolation transformer, the VSC, the control mechanism, and additional features of harmonics correction and Power Factor Correction (PFC). The control mechanism was modified to improve its response capability. For the software part of the control mechanism, the dissertation DVR selected the Wavelet Transform (WT) over the Fourier Transform (FT). For the control mechanism hardware, the dissertation DVR selected a Field Programmable Gate Array (FPGA) board for processing instead of the traditional Digital Signal Processing (DSP) board. The reasons behind these modifications are further discussed in later sections.

There were no modifications made to either the isolation transformer or the VSC. Since the VSC varies depending on the switch selection, a brief background on the VSC and the switch selection for the dissertation DVR is provided in the following section.

The additional features of harmonics correction and PFC are also further discussed in later sections. The added capabilities that these features provided sets the dissertation DVR apart from other DVRs. Thus, in additional to mitigating voltage sags, the dissertation DVR provides power quality improvements in terms of harmonics correction and PFC.

3.2 Voltage Source Converter

A VSC is a power electronic device which can generate a sinusoidal voltage at any required frequency, magnitude, and phase angle [13]. In the DVR application, the VSC is used to temporarily replace the supply voltage or to generate the part of the supply voltage which is missing [13]. The heart of the VSC is a medium used for switching the voltages. These type of switches allow for DC (Direct Current) circuits to control power without the use of elaborate commutation circuitry [27]. The controller that will be used to control these switches will be from the FPGA board.

There are three main types of switching devices: Metal Oxide Semiconductor Field Effect Transistors (MOSFETs), Gate Turn-Off (GTO) thyristors, and Insulated Gate Bipolar Transistors (IGBTs). Each type has its own benefits and drawbacks. The MOSFET requires a high on-resistance and has fast switching times [27, 34, 35]. It is capable of working beyond the 20 kHz frequency [27]. The limitations are that the increasing on-resistance with increasing voltage limits the device to applications with just a few hundred volts [27].

The GTO is a latching device that can be turned off by a negative pulse of current to its gate [27, 36]. The GTO is best suited for high voltage applications [1, 37].

The disadvantages of the GTO are that GTO based devices are not able to meet the dynamic requirements of a DVR [38].

The IGBT is considered to be a newer device compared to the MOSFET and GTO. It was first introduced in the early 1980s and has become a popular device because of its superior characteristics [35]. In essence, it is a three terminal controllable switch that combines the fast switching times of the MOSFET with the high voltage capabilities of the GTO. The result of this combination is a medium speed controllable switch capable of supporting the medium power range. A table comparing the three types of devices is summarized in table 3.1 [37]. The dissertation DVR requires a fast switching device with medium voltage capacity; thus, the IGBT was selected as the device best suited for this effort. The IGBT used in this project is shown in figure 3.1. Its specification is listed in appendix C [39].

Table 3.1 Switch Comparisons [37]

Device	Power Capability	Switching Speed
MOSFET	Low	Fast
GTO	High	Slow
IGBT	Medium	Medium

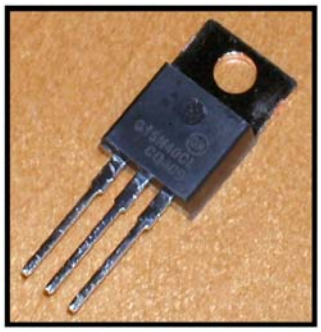


Figure 3.1 IGBT

3.3 Controls

The control mechanism part of the dissertation DVR contains modifications to improve its performance. The control mechanism is responsible for processing the control logic behind functions such as compensation and correction. This involves both hardware and software.

The control mechanism software of the dissertation DVR is in VHSIC Hardware Description Language (VHDL), and the hardware is via an FPGA board. Traditional DVR often implements the controls via software such as C on a DSP board. The implementation of VHDL on an FPGA board is not a trivial matter, so this topic is further discussed in a later chapter.

In additional to modifying the hardware and software of the control mechanism, the basis behind the filtering algorithm is also modified. The dissertation DVR uses a Wavelet Transform for filtering instead of the conventional Fourier Transform. The Wavelet Transform is further described as follows.

3.3.1 Wavelet Transform

There are many types of filtering available. Aside from the WT based filter, other algorithms range from the conventional FT based algorithm to the basic Phase-Locked Loop (PLL). The FT based algorithm focuses on the phase shift to detect the voltage disturbances [32]. The PLL focuses on the phase jumps for detection [31]. Both the Fourier Transform and the PLL based algorithms have often been criticized for having a slow response time [32]. In addition, the PLL is difficult to control since it is inherently an analog process. To address these issues, the dissertation DVR selected a WT based filtering algorithm.

Compared to the conventional Fourier Transform, the Wavelet Transform is considered a new technique. The main difference between the Wavelet Transform and the Fourier Transform is that the Wavelet Transform operates in both the time and frequency domain while the Fourier Transform operates only in the frequency domain [32]. The Wavelet Transform based algorithm also performs better with non-periodic and non-stationary signals [33]. This is ideal for detecting events such as voltage disturbances.

The Wavelet Transform based filtering algorithm used in the dissertation DVR is referred to as the Linear Phase Wavelet (LPW). The LPW consists of a wavelet based filter and a four point Root Mean Square (RMS) average. Its performance is considered robust in terms of its reaction to frequency variation and harmonics [40, 41].

3.3.2 Hardware/Software Controls

The control mechanism hardware/software components of the dissertation DVR are modified from the traditional DSP board/C to the FPGA board/VHDL, respectively. This modification was made in order to address the slow response of DSP boards. These were common complaints of past DVRs [23], [28], [42]. The main drawback in past DVRs is that the controllers were often limited in bandwidth due to the delay of the signal processing [43]. Speed is an important consideration because the FPGA board needed to accommodate the computationally intensive combination of harmonics correction, PFC, and voltage sag compensation.

FPGA boards outperform DSP boards primarily because DSP boards are based on microprocessors. These microprocessors are traditionally designed with the von Neumann architecture. This architecture is based on sequential or serial processing. The FPGA architecture is different. It is based on parallel processing. The specific FPGA architecture varies by vendor such as Xilinx or Altera, but their basic architecture is the same. It is designed with parallel processing in mind in order to allow for quick reconfigurability. As a result, parallel processing outperforms sequential processing in terms of speed. This point is best illustrated by figure 3.2 [44].

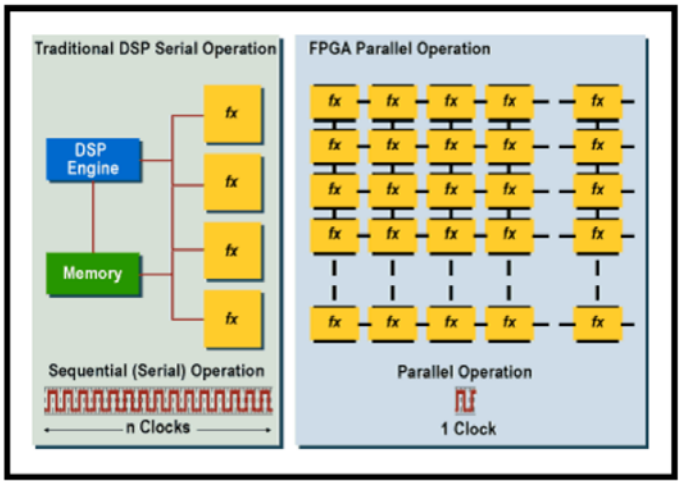


Figure 3.2 DSP vs. FPGA [44]

Figure 3.2 shows the traditional DSP serial operation on the left and compares it with the FPGA parallel operation on the right. The figure shows that it takes many clocks for the DSP to process functions because of its sequential nature. The FPGA is able to process many functions in one clock because of its parallel architecture, where multiple operations are processed at the same time.

3.4 Harmonics Correction

In addition to modifying the control mechanism, the dissertation DVR also added other features which differentiate it from other DVRs. One of these features is the harmonics correction. Power quality improvements such as harmonics correction enhances a DVR with minimal effect on its basic sag compensation performance [45].

Harmonics are defined as sinusoidal voltages with frequencies that are integer multiples of the fundamental frequency [16]. In this case, the fundamental frequency is 60 Hz because the dissertation DVR is intended for operation within the United States.

(The United States operates on 60 Hz, but other countries such as those in Europe operate on 50 Hz.) Figure 3.3 is provided for further understanding of harmonics.

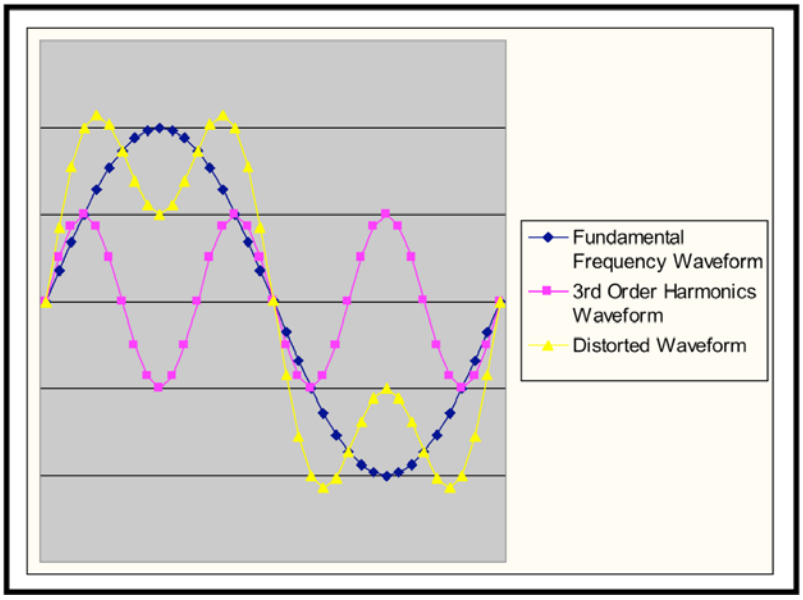


Figure 3.3 Harmonics

Based on figure 3.3, the waveform in blue is the fundamental frequency waveform, and the waveform in red is a 3rd order harmonics waveform. When a 3rd order harmonics is in the presence of the fundamental frequency waveform, it looks distorted as shown in the yellow waveform.

Harmonics are often caused by the nonlinear relationship between the voltage and the current across a nonlinear device [16]. A switching device is an example of a nonlinear device [35]. The effects of harmonics include harmonic heating and torque pulsation [46]. These effects ultimately result in damage to the equipment. Transformers and other industrial equipment are more susceptible to harmonics [16].

This is why harmonics correction is needed and is considered an important enhancement to the dissertation DVR.

Harmonics correction improves the power quality of the voltage by nulling out unwanted harmonics. In this application, since the compensated voltage will be provided by the dissertation DVR, the correction is a natural progression for the DVR by nulling out any harmonics first. The harmonics correction of the dissertation DVR is a continuous process. Initially the harmonics correction is performed as a part of the Linear Phase Wavelet algorithm. The correction is performed within the filter part of the LPW. This filter is specifically designed to null out any harmonics, if present. Since there is an inherent delay in the Linear Phase Wavelet algorithm, the power factor correction part also corrects for the harmonics in order to compensate for the delay.

The Total Harmonic Distortion (THD) calculation is used to quantify the performance of the harmonics correction. This value is a ratio of the sum of the harmonics and the fundamental frequency [13]. A high THD indicates that the sum of harmonics is greater than the fundamental frequency. After harmonics correction, a low THD is an indicator that the harmonics are suppressed such that its sum is less than the fundamental frequency. This calculation is made and shown later in the dissertation.

3.5 Power Factor Correction

Another enhancement to the dissertation DVR is the Power Factor Correction. Power factor is defined as the active power divided by the apparent power [16]. The active power is defined as the average rate of delivery of energy, and the apparent power is defined as the product of the RMS voltage and current [16]. In layman's terms,

the power factor indicates the effectiveness of a piece of equipment drawing power from the utility [37]. The power factor value ranges from 0 to 1. The ideal value for the power factor is unity or 1, and low values of power factor are not desired.

PFC attempts to keep the compensated power close to unity power factor. There are two methods of power factor correction: passive and active. Passive power factor correction corrects the power factor using passive components such as capacitors. Active power factor correction corrects the power factor actively such as using a boost converter.

The passive PFC approach is simple but it is costly and suffers from increasing weight and size as the power requirements increase. It also becomes limited in range when the voltage and power grows to a certain level. The active PFC approach is more difficult and complex, but it is considered by most power supply designers as a more effective method of PFC [27]. This is because in order for the passive PFC to work, the amount of correction needs to be known ahead of time so the capacitor could be sized correctly. The active PFC does not have this limitation and is capable of correcting for any PF dynamically. Another advantage of the active PFC is that it is often smaller in size and lighter in weight compared to the passive PFC. Thus, the active PFC is clearly the most superior type of PFC.

Due to the many advantages of the active PFC, the dissertation DVR includes active power factor correction. The PFC was added on to the original scope of the dissertation DVR in order to further enhance its capabilities. The dissertation DVR

implements its active PFC via the boost converter method. The basic boost converter topology is shown in figure 3.4.

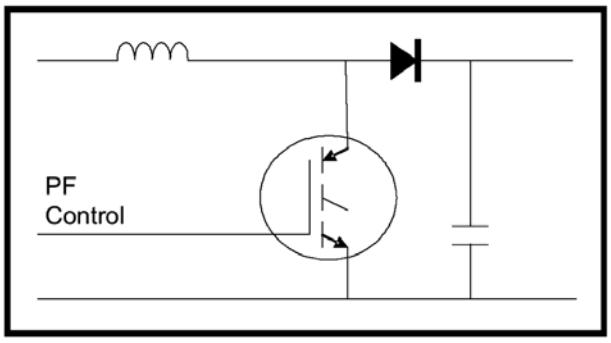


Figure 3.4 Basic Boost Converter Topology

There are three main converter topologies: buck, boost, and buck-boost. The boost method is the most popular method. This is because it is capable of operating in the continuous mode and has the ability to maintain control over the complete input voltage waveform [27].

The boost converter shown in figure 3.4 consists of a diode, an inductor, a capacitor, and a switch. The switch in the figure is used for power factor control. Many switches are available for this circuit, but the dissertation DVR selected the Insulated Gate Bipolar Transistor (IGBT) as the switch for the voltage sag compensation, so it is used for the PFC as well. The same switch is suitable for both functions. The selection process of the switch is discussed in the VSC section.

The basis of the PFC is accomplished by maintaining the load. As long as the load is maintained, the power factor is corrected to stay close to unity power factor. To understand this further, a series of equations are provided to describe this process [47]. First, consider the voltage across the boost inductor, as represented in equation 3.1.

$$v = L\frac{\partial i}{\partial t} \tag{3.1}$$

Every time the inductor is charged, the change in the inductor current is with respect to time. The relationship between the current, the inductance and the voltage is represented by equation 3.2. This equation describes the peak current as I_{pk}, and the IGBT gate time as t_1[47].

$$V_{in} = L\frac{(I_{pk} - 0)}{(t_1 - 0)} = L\frac{I_{pk}}{t_1} \tag{3.2}$$

The IGBT gate time is further described by equation 3.3. This equation describes the input current in terms of the peak current, I_{pk}, the inductor discharge time, and the IGBT gate switching characteristics. The current is one-half of I_{pk}, which is then multiplied by the sum of t_1 and t_2 and adjusted by the overall period T [47]. The overall period T represents the frequency of the IGBT pulsing. The pulse width itself is represented by t_1, and the inductor discharge ramp time is represented by t_2.

$$I_{in} = \int_0^T Idt = I_{pk} \cdot \frac{t_1 + t_2}{2} \cdot \frac{1}{T} = \frac{I_{pk}(t_1 + t_2)}{2T} \tag{3.3}$$

The next equation combines equations 3.2 and 3.3. Equation 3.4 is the overall input impedance as described by Ohm's law[47]. It is the basis behind the active PFC technique. By maintaining the impedance of the load, the power factor is maintained at unity power factor. V_{in} from equation 3.2 and I_{in} from equation 3.3 are substituted into equation 3.4. This substitution provides a correlation between maintaining the impedance and managing the components.

$$Z_{in} = \frac{V_{in}}{I_{in}} = L\frac{I_{pk}}{t_1} \cdot \frac{2T}{I_{pk}(t_1 + t_2)} = \frac{2TL}{t_1(t_1 + t_2)} \tag{3.4}$$

By controlling the pulse width of the IGBT gate to synchronize the current waveform with the voltage waveform, the input impedance is constant or slowly varying, thus, power factor is then corrected [47]. This is the basis for most boost converter power factor control schemes. It is also the scheme behind the PFC in the dissertation DVR.

In summary, the dissertation DVR consists of the basic DVR configuration with a modified control mechanism and additional features of harmonics correction and PFC. The modified control mechanism includes an FPGA board and VHDL for the control mechanism hardware/software, respectively. The filtering algorithm within the controls is also modified with a Wavelet Transform based filtering. These modifications are made to address the performance issues of past DVRs.

In addition to the modifications, two additional features are included as part of the dissertation DVR. These features further differentiate the dissertation DVR from other DVRs. These features are harmonics correction and PFC. The harmonics correction is a continuous process and is implemented as part of the LPW algorithm and as part of the power factor correction. The dissertation DVR implemented the active PFC. Implementation of active PFC is a technical challenge, but it is capable of correcting the power factor without a priori knowledge of the poor power factor.

CHAPTER 4

FPGA DESIGN

The control mechanism hardware/software of the dissertation DVR are modified and replaced with an FPGA board and VHDL instead of the traditional DSP board and C, respectively. FPGA design is not a trivial matter, so an entire chapter is devoted to this topic to help the reader understand its importance and the effort that it requires in such a task.

4.1 Background

FPGAs are nothing new, but they have evolved considerably compared to the FPGAs in the past. This evolution is due largely in part to the technological advances in the semiconductor industry. The number of transistors on one chip has significantly increased over the years. The increase is almost exponential. This effect is characterized by Moore's Law. An illustration of Moore's Law is shown in figure 4.1 [48].

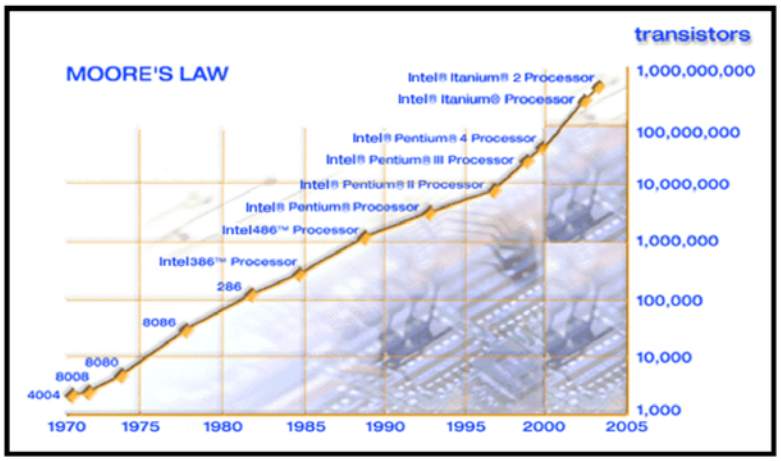


Figure 4.1 Moore's Law Graph [48]

Essentially, Moore's Law Graph shows that more and more transistors could be squeezed into one chip over the years. This benefits the FPGA. It means that one FPGA chip is capable of supporting more operations. By supporting more operations, a larger algorithm or a more complex algorithm or even multiple algorithms could fit into an FPGA. The LPW algorithm fits into modern FPGAs with room to spare. That extra room is being utilized for the dissertation DVR control functions.

In addition to more transistors in one chip, the technology also improves the speed performance of the device. A large part of the speed performance improvement is due to the parallel architecture of the FPGA. This parallel architecture is also essential to configuring and reconfiguring the device. A figure is provided to help describe the parallel architecture and how an FPGA works. Figure 4.2 is a generic representation of a snapshot within an FPGA architecture.

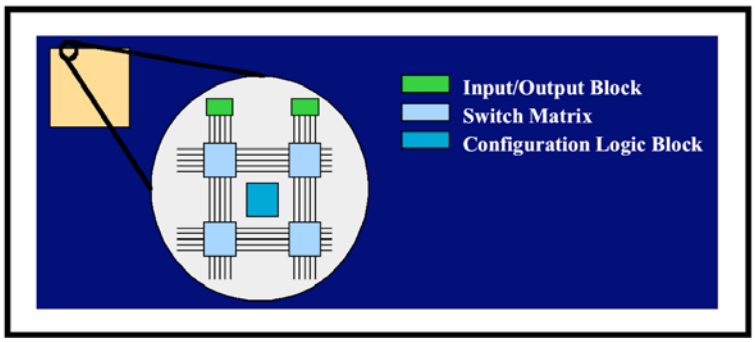


Figure 4.2 Generic FPGA Architecture

The yellow square in the figure represents an FPGA chip. The lines and squares inside the big circle show a zoomed-in view of what is in the small circle on the FPGA chip. The green squares in the zoomed-in view are often known as the Input/Output (I/O) block of an FPGA. The I/O blocks surround the outer edges of the FPGA. They are used to transport data in and out of it. The light blue squares in the zoomed-in view are commonly known as the switch matrix. The switch matrix serves as a junction for the horizontal and vertical wires. They connect a neighboring logic block with another neighboring logic block. The dark blue square in the center of the zoomed-in view is known as the Configuration Logic Block (CLB). The configuration logic block is typically where the functions reside. It generally contains registers and Look Up Tables (LUTs).

The zoomed-in view represents just a small section of the FPGA. The pattern of the switch matrix and the CLB is repeated throughout the entire FPGA. It is the repeating pattern that allows the FPGA to be reconfigurable and parallel in nature.

The figure represents a snapshot within a generic FPGA architecture, and the zoomed-in view is a block level and simplistic look at the FPGA to help the reader

understand its parallel nature and reconfigurability. This top level look at the FPGA is analogous to that of a road map. The CLB is connected to the switch matrix like locations on a road map are connected to other locations. Although a top level view of a map might only show key locations and interstates and leave out the streets. Compare this to the top level view of the FPGA, the CLB and the switch matrix are the locations and the lines represent the interstates.

The semiconductor technology advancements have significantly increased the number of transistors in a FPGA. This allowed for more functions to fit in an FPGA. This means more CLBs and more sophisticated switch matrix functions; it also means more functionality within a CLB. This combined with the improved speed of the FPGA makes the FPGA board an ideal choice for the processing within the dissertation DVR.

4.2 Programming

The advent of Hardware Description Language (HDL) has made it easier to program FPGAs. There are two hardware design languages that are commonly used to design integrated circuits (ICs). These same languages are also used for configuring FPGAs. The two languages are Verilog and VHSIC Hardware Description Language (VHDL). VHSIC stands for Very High Speed Integrated Circuits [49].

The major difference between Verilog and VHDL is that VHDL is strongly typed compared to Verilog. This means that VHDL requires more constraints in the design up front compared to Verilog. This could be both good and bad. The good thing about being strongly typed is that the initial conditions must be defined ahead of time otherwise the code would not compile. This allows for mistakes to be discovered early

in the design process rather than later where changes are harder to make. The bad thing about being strongly typed is that it takes more time and expertise to set up the initial conditions, and sometimes not all the initial conditions are known early in the design process.

Since the dissertation DVR has many different parts and controls, VHDL is selected in order to address any issues early in the design process. Although VHDL is a programming language, it is not quite the same as software such as C. It is important to keep in mind that VHDL is still hardware except it is implemented via code. The difference between Hardware Description Languages (HDL) such as VHDL and other software languages such as C is that HDL is parallel in nature and other software languages are sequential in nature. In short, this means that it is not easy to transition from coding software such as C to coding HDL such as VHDL. A different frame of mind and an understanding of hardware are essential to efficiently code in HDL. This is also part of the reason why the FPGA board is not widely used.

For example, software code is written such that each line of code is executed sequentially. In HDL, code is written to mimic hardware where multiple processes are being executed simultaneously. One process could be the clocking process, while another process could be monitoring the inputs at the same time. Examples of VHDL code from the dissertation DVR are provided in appendix B.

Despite the fact that HDL is designed to mimic hardware, it still has the appearance and benefits of software. Its benefits are the ease of use that comes with

software. Changes could be made via lines of code. The code is re-useable, so parts of one design could easily be ported into another design.

4.3 Design Strategy

A large FPGA design requires a strategy to ensure its accuracy and timely completion. The strategy involves a plan that lists the critical steps for completion. A common FPGA design plan is shown in figure 4.3.

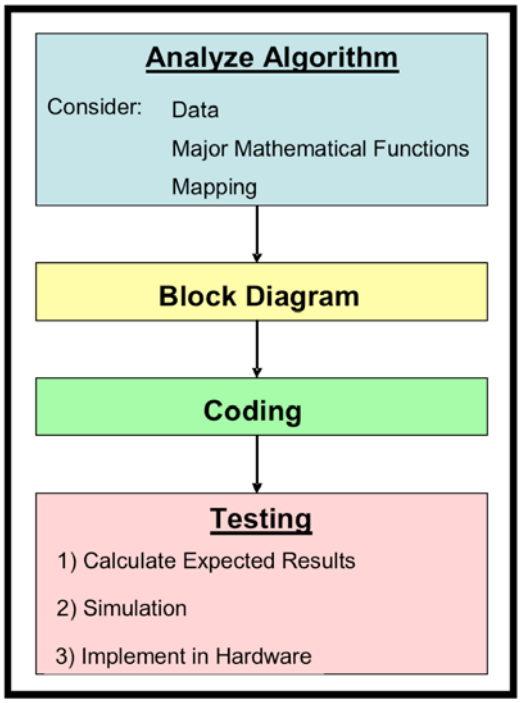


Figure 4.3 FPGA Design Plan

The FPGA design plan consists of four steps. The first step is to analyze the algorithm or algorithms to be implemented into the FPGA. The second step is to draw a block diagram of the overall algorithm or algorithms. The third step is the coding via HDL, and the last step is to test the FPGA design. Each step of the design plan is further described as follows.

4.3.1 Design Analysis

The first step in the design plan is to analyze the design; the design could be one algorithm or many algorithms. Three main topics to consider in this step are data, major mathematical functions, and mapping. When considering the data of the design, all data coming in and going out of the design must be analyzed. A major point to consider is its numerical precision, whether it is in floating-point or in fixed-point. Most algorithm development is typically done in floating-point, but fixed-point mapping is more compatible with hardware implementation, so the data is often converted to fixed-point when it is transferred over to hardware. The conversion from floating-point to fixed-point is not trivial. It involves additional analysis to ensure that the mathematical content is preserved and that the data resolution is maintained at an acceptable level.

The design analysis also includes analyzing the algorithm to determine what and how many major mathematical functions are involved. An example of a major mathematical function would be a summer. This is generally represented by an accumulator in hardware. This is important to note because an accumulator could be resource intensive and steps would need to be taken to ensure its size is under control.

Another example would be a filter. Depending on the type, a filter can be extremely resource intensive. The filter response is an important consideration, since its response is dependent on the coefficients of the filter. In algorithm development, the filter coefficients are often represented in floating-point, so they will need to be converted to fixed-point before it could be implemented in hardware.

Last but not least, a final consideration in the design analysis is the mapping. Mapping involves determining where and how to place parts of the algorithm within the FPGA. It also involves determining how to group certain parts of the algorithm together for overall efficiency.

An example of mapping consideration is by using cores. Cores are pre-designed functions that have already been optimized for both area and speed within the FPGA. Cores come from many sources. The basic functions such as an accumulator is often provided free of charge by the FPGA vendors. Other exotic functions often require a fee for them and may be available from either the FPGA vendors or third party core makers. An example core is shown in figure 4.4.

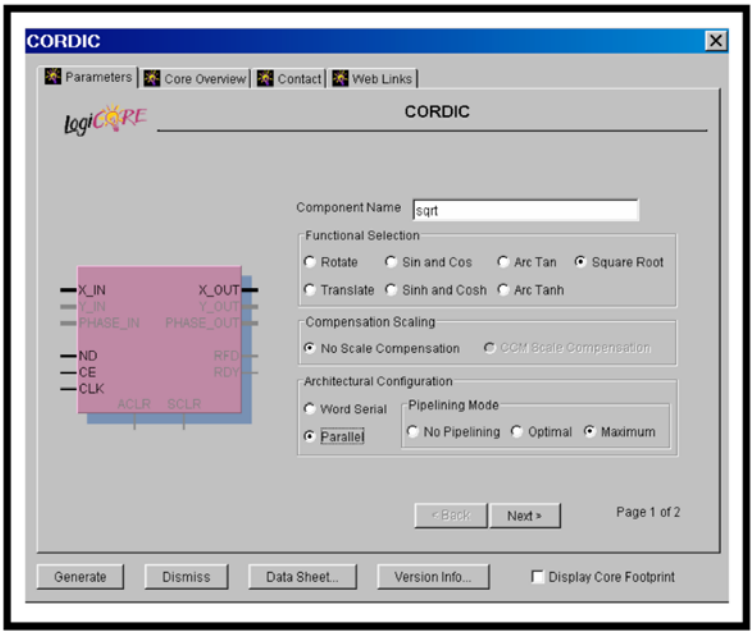


Figure 4.4 Xilinx CORDIC Core

The figure shows a Xilinx CORDIC core. CORDIC stands for COrdinate Rotation DIgital Computer. It is essentially a shift and add type algorithm that is widely

implemented in binary arithmetic. In short, CORDICs are often used to implement square root functions and Sine/Cosine functions in hardware. The core is already designed and optimized for both speed and area for this FPGA.

To use the core, the parameters need to be adjusted to meet the design requirements. This is typically done via a dialog box. Once the core is configured, it is generated, and then it is ready to use. The core is analogous to a part in a circuit design. The core needs to be configured to meet the requirements. This is like selecting a part. The core is then connected to the FPGA design, just like connecting a part onto the board.

Cores are helpful to mapping because the area is already determined and is typically available on a data sheet. Although, there are times when cores are not available or the design requirements exceed the parameters of the core. When this happens, the part needs to be designed from scratch. If this is the case, this must be factored into the design strategy in terms of extra time, speed performance, and resource allocation.

The cores and the design are managed via FPGA design tools, which are typically vendor specific. The dissertation DVR FPGA design is implemented on a Xilinx FPGA, so the Xilinx FPGA design tools will be used for this design. A snapshot of the Xilinx FPGA design tools suite, known as the ISE (Integrated System Environment) Project Navigator is shown in figure 4.5. This tool is used throughout the dissertation DVR design effort.

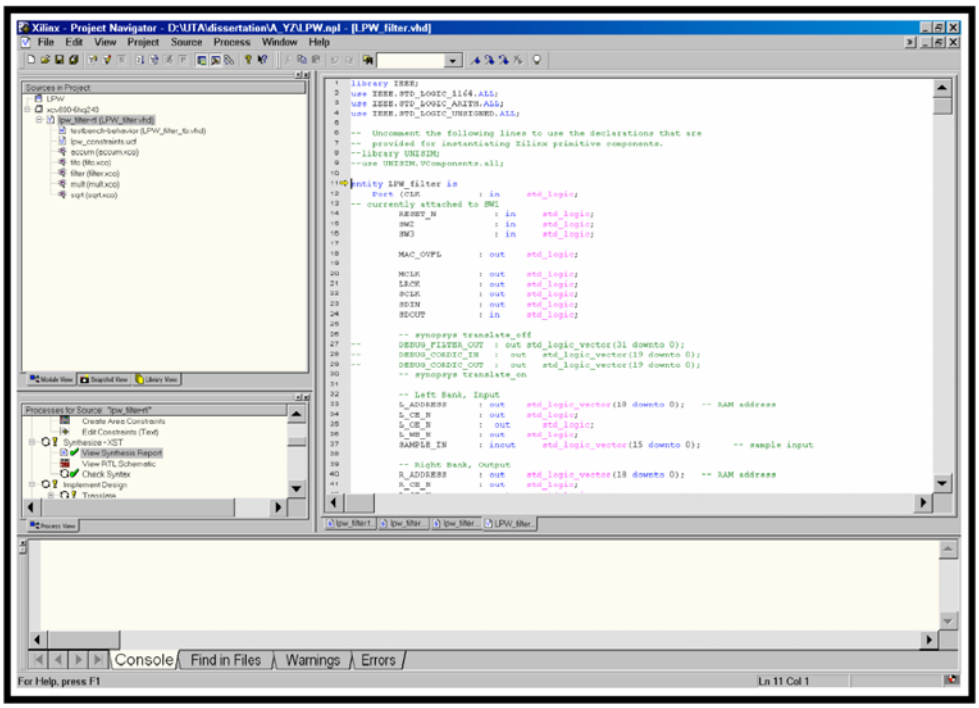


Figure 4.5 Xilinx ISE Project Navigator

Mapping is a critical step. This is because poor mapping could lead to an increase in resource utilization, slower clock speeds in the application, or failure to fit into the FPGA. Depending on the board, sometimes only one FPGA is all that is available. This is why mapping is considered as an important part of the design strategy.

After analyzing the design, the next step is to pull the analysis together into the form of a block diagram and then code it block by block. The coding is typically done within the FPGA tools; in this case it is done using the Xilinx ISE Project Navigator. After coding, the tool aids in the testing and processing of the design. When this is complete, a load for the FPGA is created. The FPGA load, also known as the bit stream, is the end product of the FPGA design process before it gets put on the board.

4.3.2 Test Methodology

There are several intermediate steps between coding and the final load creation. The steps will vary some depending on the FPGA designer and the type of tools that are used. These fundamental steps are briefly discussed in this chapter.

One of the intermediate steps is the testing of the design. Testing is considered as one of the most critical steps in the entire process, and yet it is often ignored. This is because this process is often tedious and time consuming. Testing ensures that the FPGA design is functional and that it will perform as designed in hardware.

FPGA testing often consists of three sets of tests: calculated test, simulated test, and field test. The first set of tests is the calculated test. The calculated test consists of calculating the expected results of the design. This could be done in many ways, but in this design, it will be done via Microsoft Excel Spreadsheet. The second set of tests is simulated test. The simulated test consists of simulating the coded design via a simulator. In this design, the coded design will be done in VHDL, and the VHDL simulator used in this design will be a combination of ModelSim and Xilinx ISE Project Navigator. Field test is the final step of the process where the design is implemented in hardware. In this step, the hardware is verified if it functions as it was designed.

For simulated tests, a VHDL testbench is first needed to test the design. Since the testbench is written by the designer to suit each design, it will vary from designer to designer. The typical testbench consists of an input stimulus process that mimics the input of the Design Under Test (DUT) and an output process that logs the output of the design.

The simulator runs the testbench and the DUT simultaneously, and it generates an output in the form of a waveform or multiple waveforms. The waveforms represent the signals in the design. Although the waveform view varies depending on the designer, it typically encompasses the input signals generated by the testbench, the internal signals of the design, and the output signals of the DUT. The waveforms are then examined for design functionality and timing accuracy. Verification of the design functionality is often done by comparing this with the calculated results. In this case, it will be compared with the data generated from the Excel spread sheet. An example ModelSim waveform view is shown in figure 4.6.

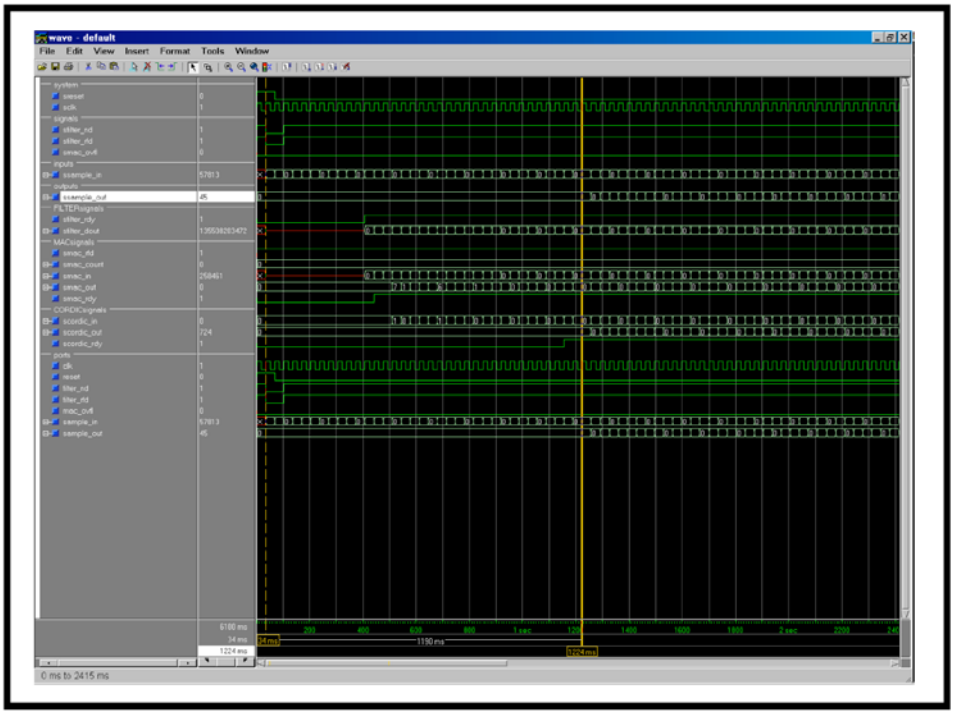


Figure 4.6 ModelSim Waveform

In addition to the waveform, an output stimulus process is also coded in the testbench to log the output of the DUT. The logging style will vary, but it is commonly

done in text format. The text format is useful for comparing results quickly. Waveforms have to be verified visually, and this takes time. The text format could be verified via a comparison routine such as “diff” and compare it with the calculated data. Once the comparison routine is set up, the verification could be done very quickly. The output text log is often saved for later use such as regression tests. Regression tests are further described in the following section. An example output text log is shown in figure 4.7.

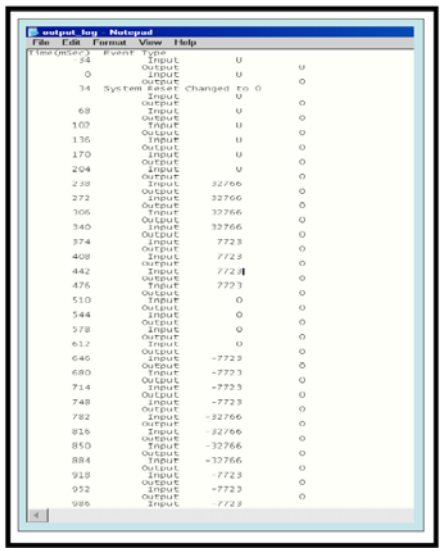


Figure 4.7 Test Output Log

The output log typically records every event at every clock. These events include both inputs and outputs. The event also includes both valid and invalid data. The log ensures that any changes to the design are recorded. This helps to verify that the design under test has not been altered unintentionally in any way. This is essential for regression tests.

Regression tests are a series of tests to verify that any additional changes to the current design have not altered the overall design in unexpected ways. These series of

tests verify different functionalities of the overall design. For example, if one part of the design has been altered after it has been verified, the regression tests need to be run to ensure that the change has not propagated to other parts of the design. If the change affected other parts of the design, that part of the design whose regression test did not pass will need to be re-verified via waveform to ensure that the overall design is still functional. In this design, an output text log is kept for regression tests. The waveform and the output log are all a part of the simulated test.

The final test of the test methodology is the field test. This is to implement the FPGA design in hardware and to verify that the hardware operates as designed. This is accomplished after design analysis, simulated tests, and design implementation. Design implementation is described in a following section.

Field test is where the FPGA design is put into the FPGA and implemented on a board. This typically involves integrating the board into a system and then verifying the system's overall functionality. If the system is not available, an alternate method of verification is to check the hardware via an analyzer or an oscilloscope. Although this would not necessarily constitute a completed test, it is useful for visually verifying the output of the design. Other means of verification such as storing the output data in Random Access Memory (RAM) or connecting the output to a display are also possibilities to verify the hardware implementation on the board.

The test methodology indicates that the FPGA design is verified as functional only if all three tests match and correspond to each other. The three tests are calculated test, simulated test, and field test. Testing methodology is important because it ensures

that the design is carried out as intended on the hardware. It also helps to reduce design errors and the overall design cycle. The dissertation DVR follows the testing methodology.

4.3.3 Design Implementation

The step between simulated test and field test is the design implementation. It is one of the intermediary steps between coding and FPGA load creation. After the design has been coded and verified via simulated tests, the design is processed through various tools to realize the code into actual components on a FPGA. This is known as the design implementation process.

There are many steps within the design implementation process. Some of these steps will vary depending on the processing tools, but only the fundamental steps are discussed in this section. In this case the processing tools are dependent on the FPGA vendor, and for the dissertation DVR, this is the Xilinx ISE Project Navigator.

There are two fundamental steps in the design implementation. These steps are synthesis and place and route. The synthesis step interprets the VHDL code and translates the code into components such as LUTs and gates. This is a very important step. Synthesis basically helps determine the preliminary resource utilization of the design. The results of synthesis are often displayed in the form of a report known as the synthesis report. An example synthesis report summary is shown in the table 4.1.

Table 4.1 Synthesis Report

Synthesis Report:

Device utilization summary:
Selected Device : v800hq240-6

Number of Slices:	2617 out of 9408	27%
Number of Slice Flip Flops:	2957 out of 18816	15%
Number of 4 input LUTs:	4310 out of 18816	22%
Number of bonded IOBs:	85 out of 170	50%
Number of BRAMs:	1 out of 28	3%
Number of GCLKs:	1 out of 4	25%

The synthesis report is typically a lengthy report that lists the synthesis results of the entire design. This includes the type and the number of components used in the design and the timing information of each component. All this information is then summarized in the report in terms of the design resource utilization. The example synthesis report summary in table 4.1 shows the resource utilization of the Linear Phase Wavelet design. The summary breaks down the utilization based on components. This is helpful information for FPGA designers to determine what resources are still available in the FPGA. It also helps FPGA designers double check on the VHDL code to make sure that no components were inferred in the code unintentionally. Many synthesis reports also show the estimated speed of the design. This information is highly estimated, since the actual speed of the design would not be known until all the components have been placed and routed. Thus, the speed information in this report is rarely used.

Place and route is the other fundamental step of design implementation. The place and route step basically places the components and routes the wires between the

components. In order to ensure that the components are optimally placed and that the wires are the shortest paths between the components, the place and route step analyzes and models various configurations of the design. This is done with guidance from various algorithms that are provided with the processing tools. These algorithms provide a balance between optimal resource configuration and reasonable run time for the design implementation.

The design implementation part of the FPGA design is just as important as the other parts of the FPGA design process. It translates the design from VHDL code into a load for the FPGA. All of the steps within design implementation are accomplished via the processing tools such as Xilinx ISE Project Navigator. These steps will vary by tools, but the two fundamental steps are synthesis and place and route.

CHAPTER 5

DISSERTATION DVR PROOF OF CONCEPT

Before implementing the dissertation DVR, its basic proof of concept is verified to ensure that the idea behind the implementation is sound. A schematic of the dissertation DVR proof of concept circuit is shown below.

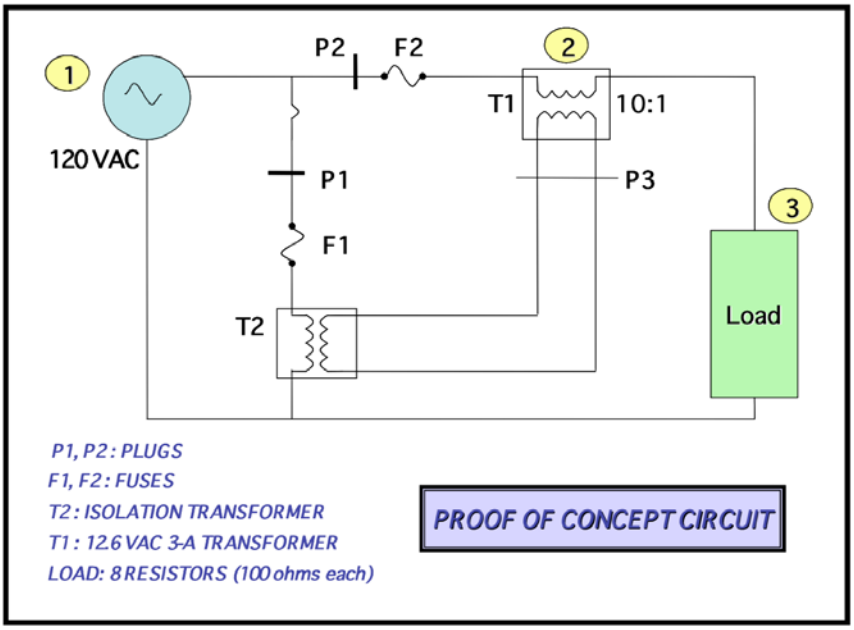


Figure 5.1 Proof of Concept Circuit

The proof of concept circuit mainly consists of two transformers and several resistors, plugs and fuses. This circuit was implemented with T2 as the isolation transformer and T1 as the 12.6 VAC (Volt Alternating Current) transformer.

The load of the proof of concept circuit is made up of eight resistors at 100 ohms each. Plugs and fuses were added to the circuit implementation for safety measures. This dissertation DVR proof of concept circuit was built and tested to demonstrate the idea behind it. The final product is shown in the figure 5.2.

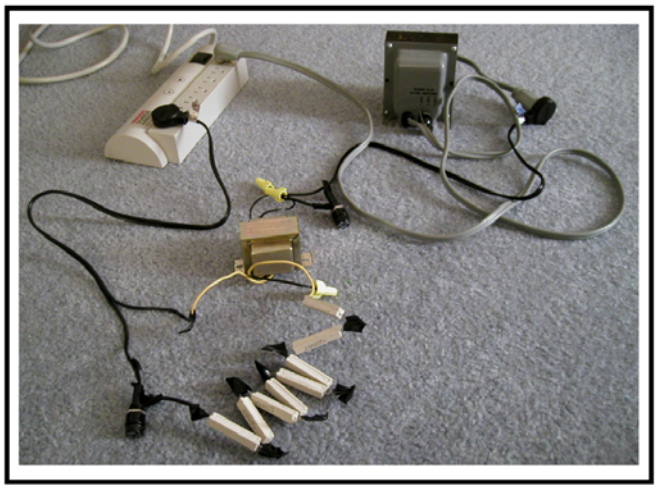


Figure 5.2 Dissertation DVR Proof Of Concept

The results from the dissertation DVR proof of concept are listed in table 5.1. The table summarizes the voltages of points 1, 2, and 3 at different locations. Point 1 is the source voltage. Point 2 is the compensated voltage from the transformer, and point 3 is the load voltage. The last column of the table verified that the voltages were compensated at the different levels. The last column shows that the voltages at points 2 and 3 always sum up close to point 1. The results showed that the voltage is compensated at point 2 of the circuit at the various stepdown settings of T2.

Table 5.1 Proof of Concept Results

Transformer Switching voltage \ Circuit		1	2	3	2 + 3
210 V	V	119.9	8.41	111.6	120.1
	ϕ	0	0	0	0
230 V	V	119.8	7.65	112.2	119.85
	ϕ	0	0	0	0
250 V	V	119.8	7.05	112.75	119.8
	ϕ	0	0	0	0

The dissertation DVR proof of concept is an important step to the development of the dissertation DVR. It demonstrated that the basic concept behind the dissertation DVR is feasible for implementation, and thus, the implementation of the dissertation DVR could proceed.

The dissertation DVR proof of concept is a static configuration. This static configuration is improved by implementing a dynamically controlled transformer configuration in the final dissertation DVR.

CHAPTER 6

DISSERTATION DVR FPGA DESIGN

Implementing the dissertation DVR primarily consists of two efforts, FPGA design and system design. This chapter focuses on the part of the FPGA design, which does not require system interaction for verification. There are many pieces to the dissertation FPGA design. This chapter uses the WT based filter and parts of the harmonics correction as an example to illustrate the FPGA design strategy. Other parts of the FPGA design such as the controls and the PFC are discussed in chapter 7 under system design. The system design chapter covers the remaining essential pieces needed to pull the entire system together into an operational device. This includes a combination of both FPGA design and hardware implementation.

The FPGA design itself is considered as one of the features of the dissertation DVR in that the FPGA board is selected over the traditional DSP board. FPGA design is the central processing mechanism of the dissertation DVR. It is utilized for most of the functionalities and features. The FPGA design effort is a large part of the overall dissertation DVR hardware implementation.

The first step in the FPGA design is to identify the hardware, such as what type of FPGA and what type of board. It is crucial to establish this in advance since it will be the basis for everything else that is to follow. There are many types of FPGA boards,

but based on availability and budget limitations, the board selected for the dissertation DVR is an XESS FPGA prototype board, model XSV, version 1.1.

The FPGA part on the board is the Xilinx Virtex FPGA, part XCV800 HQ240AFP0045. The HQ in the part reference indicates the type of packaging that the part is in. The number 240 that follows the packaging designation indicates the number of pins that are on the part. This part is also stamped with a number 6. This number indicates the speed grade of the device. This part comes in three speed grades, and the number 6 indicates that it is a medium speed grade. The speed grade means how fast the part is able to operate relative to the temperature. This information is pertinent to the implementation step later in the design process, since the tools will need to know which part it is programming. The XESS board is a prototyping board, so various peripherals are also included on the board, such as different kinds of connectors and Light Emitting Diodes (LEDs). A snapshot of the XESS FPGA XSV board is shown in figure 6.1.

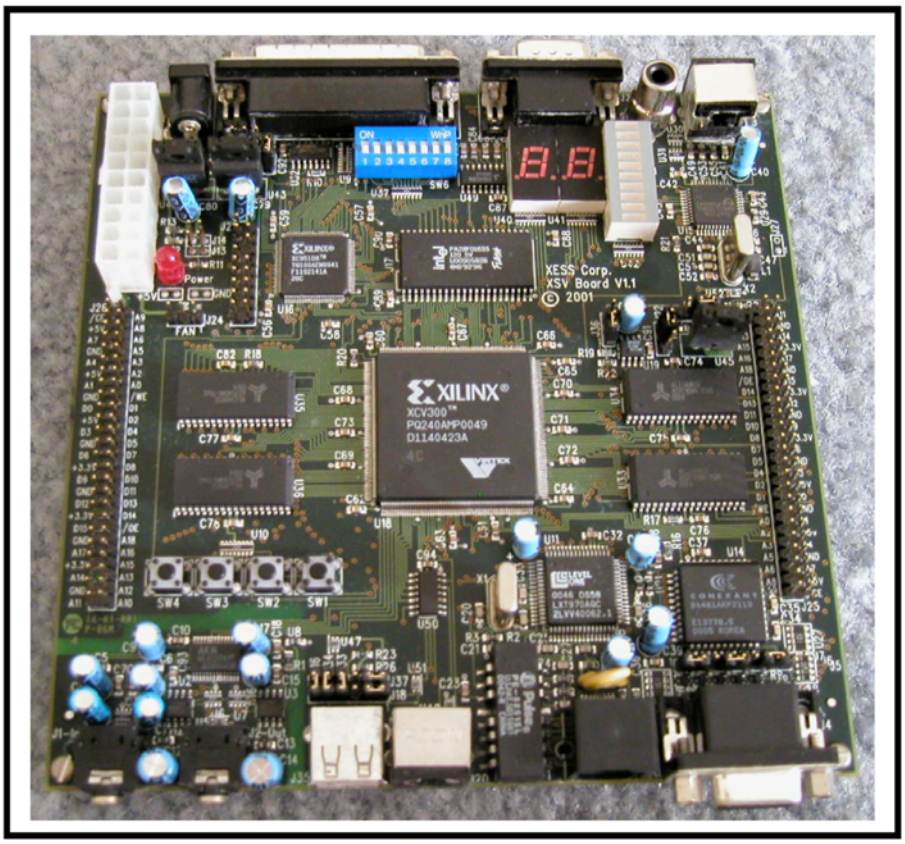


Figure 6.1 XESS FPGA Board

Since the FPGA design of the dissertation DVR is so large, the design process is broken up into different phases. The phases are distinguished between those that require system interaction for verification and those that do not. The WT based filter and parts of the harmonics correction are determined as those that are not dependent on system interaction. These functions are implemented first under the FPGA design.

6.1 Analysis

The WT based filter and parts of the harmonics correction are combined into one design. This design is referred to in this paper as the Linear Phase Wavelet (LPW) algorithm. An analysis is first performed on the LPW algorithm since it is considered to

be the most computationally and resource intensive part of the entire design. In FPGA design, filters are often one of the most resource intensive components of a design. In addition, the WT is new method compared to the traditional FT, so its development had to be coded from scratch. Thus, additional measures in terms of analysis and testing were taken to mitigate its impact on the overall effort.

The analysis included analyzing data, mathematical functions, and mapping of the LPW as described in the FPGA design strategy. The analysis examined the data flow into and out of the algorithm. It is noted that the LPW algorithm is originally based on floating-point instead of fixed-point. This meant a conversion to fixed-point is necessary in order to map the algorithm into an FPGA.

The conversion is determined by analyzing the input equation of the algorithm itself. The LPW input equation is shown in equation 6.1 [40]. By analyzing both ends of the spectrum in this input equation from positive to negative, it was determined that sixteen bit representation is adequate for the LPW algorithm.

$$v(n) = \sum_{k=1}^{5} A_{n,k} \cos(2\pi f_{n,k}(2k-1)n + \phi_{n,k}) \tag{6.1}$$

The next step in the analysis is the mapping and the major mathematical functions analysis. Analyzing how to map an algorithm into an FPGA is an involved process. If the size of the FPGA is large, not much work is needed to implement the design into the FPGA. If the FPGA is not as large, there are ways to minimize the gates in order to squeeze the design into the FPGA. One technique to minimize the gates is to

use design cores whenever possible. Since the dissertation DVR uses a Xilinx FPGA, Xilinx cores are available for this design. By applying cores wherever possible also allows for a reusable design and shortens the overall design cycle.

As part of the mapping and major mathematical functions analysis, a block diagram of the LPW algorithm is created based on Xilinx core availability. The block diagram shows that there are four major mathematical functions, and cores are available for each block. The four cores are filter, multiplier, accumulator, and the square root. The LPW algorithm block diagram is shown in figure 6.2.

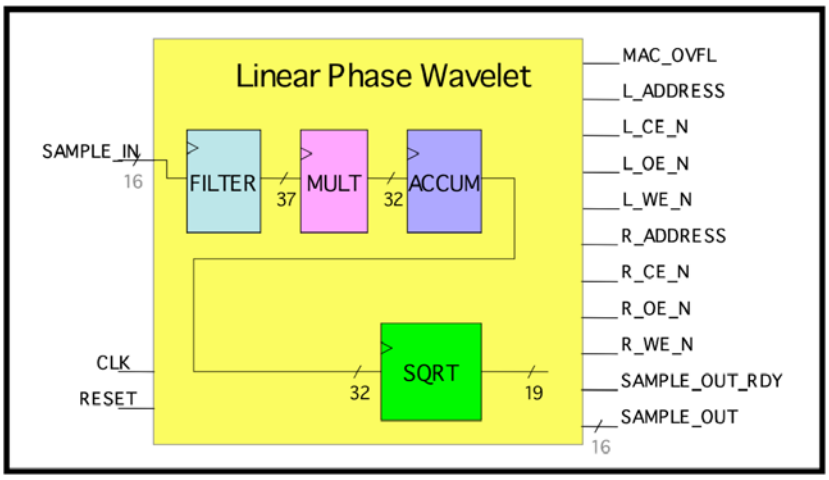


Figure 6.2 LPW Block Diagram

Although there is a core available for the filter, the filter still needed additional analysis to determine its appropriate filter impulse response. The filter impulse response showed that the sixteen-bit representation used in the data input/output (I/O) was not suitable for the filter. Instead, it was determined that eighteen bit representation provided the optimal resolution for the filter. The Xilinx filter core allowed for a different bit width filter coefficients than the filter, so the eighteen-bit representation

was implemented only on the filter coefficients. The remainder of the design remained unaffected at sixteen-bit representation. An example of the filter coefficients converted from floating-point to eighteen-bit representation is shown in table 6.1.

Table 6.1 Filter Coefficients Bit-Representation Conversion

	Floating-Point	18-Bit
0	1.1049E-02	11586
1	1.9135E-18	0
2	-3.3146E-02	-34756
3	-6.2500E-02	-65536
4	-5.5243E-02	-57926
5	-1.7222E-17	0
6	7.7340E-02	81096
7	1.2500E-01	131071
8	7.7340E-02	81096
9	2.8703E-17	0
10	-5.5243E-02	-57926
11	-6.2500E-02	-65536
12	-3.3146E-02	-34756
13	-1.3395E-17	0
14	1.1049E-02	11586

It is typically good design practice to maintain a consistent bit representation throughout the entire design, but in this case, an exception had to be made because it was not advantageous to this design. This is because increasing the bit representation also increases the resource utilization of the device. To maintain the conservative approach towards resource utilization, the eighteen-bit representation is only applied to the filter coefficients.

Based on this analysis, it was determined that the LPW algorithm is suitable in terms of resource utilization. The actual fit would not be known until it is finally coded

in VHDL, but the initial findings provided promising results. (For more information on the basics of analysis and the design strategy, the fundamentals of FPGA design are discussed in chapter 4.)

6.2 Coding and Testing

Since the analysis indicated that the resource utilization will not be an issue, the next step is to proceed to coding. The coding of the LPW is accomplished by first instantiating cores identified in the block diagram. The logic between the cores is then added as needed. For example, a multiplexer will be coded in order to use the Xilinx accumulator cores.

The cores used in the LPW design are briefly described as follows. The filter is an instantiated Xilinx Distributed Arithmetic FIR core with 15 taps. The multiplier is an instantiated Xilinx two-input multiply core. The accumulator is an instantiated Xilinx accumulator core that is instantiated four times in order to match the accumulator in the LPW algorithm. The output from each accumulator is then multiplexed into one stream of data. The square root is an instantiated Xilinx CORDIC module. The CORDIC is commonly used to translate a square root function into hardware. (A brief description on the CORDIC is provided in chapter 4.)

The output from the filter is sent as an input to the multiplier, accumulator, and square root. The timing between the filter output and these inputs is critical. If the data is off by just one clock, that entire set of data will be mis-calculated, and that mis-calculation will propagate to other parts of the design. In order to manage this series of events, a state machine is created to maintain this interaction. The multiplier,

accumulator, and the square root are combined into one module called the Four_Point_RMS to make it easier for the state machine to manage. This state machine is called the Accumulator Control state machine. A detailed Four_Point_RMS block diagram is shown in figure 6.3.

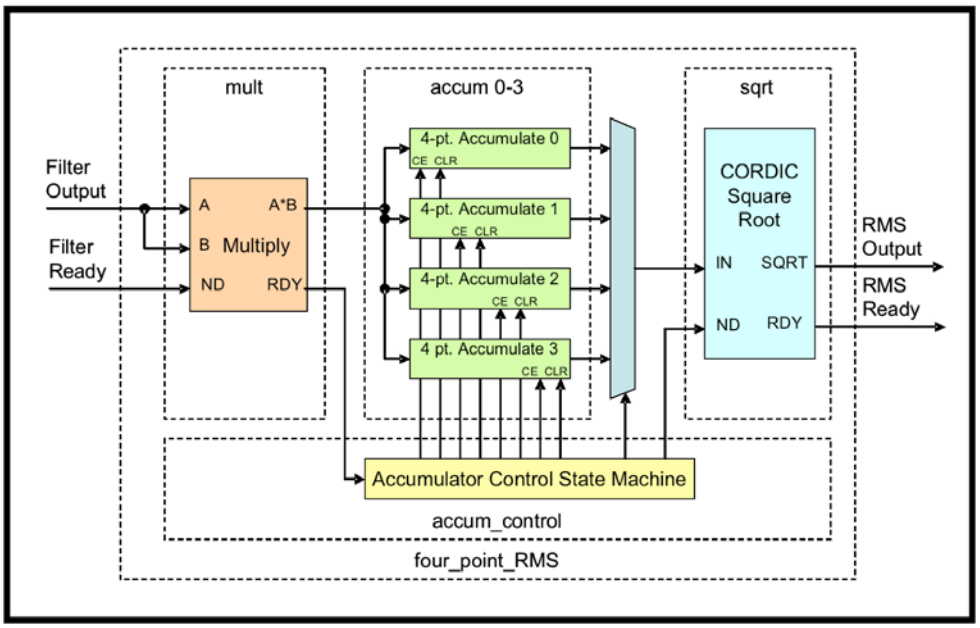


Figure 6.3 Four-Point RMS Block Diagram

Testing is performed next to verify the coding. It is also the next step in the FPGA design strategy. The expected results from the LPW algorithm were calculated via Microsoft Excel. For VHDL simulation of the LPW design, a VHDL testbench was written. Among the processes included in the testbench are the input process to mimic the input of the design and the output process to log the output of the design. Simulation of the LPW design was done via both the Mentor Graphics Modelsim tool and the built-in simulator provided in Xilinx ISE Project Navigator.

Many simulations were performed. Some simulations were performed on individual modules such as the filter, and some simulations were performed on the interaction between different modules such as that between the multiplier, accumulator, and the square root. Since the timing of the Four_Point_RMS module is critical, this module is simulated extensively.

One of the Four_Point_RMS module simulations focuses on the accumulator itself. In order to utilize the Xilinx accumulator core, the accumulator had to be instantiated four times to match the accumulator in the LPW algorithm. The output from each accumulator is then multiplexed into one data stream so that it could be sent next to the CORDIC module. The timing and multiplexing is controlled by a state machine called the Accumulation Control module. This state machine is simulated with both the multiplier and accumulator to verify timing. This simulation is shown in figure 6.4.

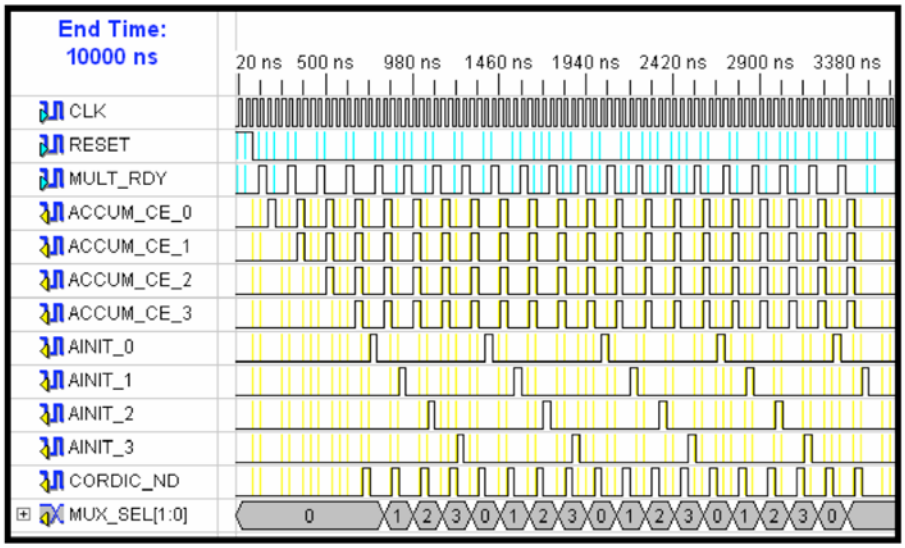


Figure 6.4 Accumulation Control Module Simulation

This simulation shows that the output of the multiplier is signaled via “MULT_RDY”. This initiates the data propagation from the multiplier to each of the accumulators. Each accumulator is indicated via the “ACCUM_CE_0-3” signals in the simulation. After the four-point sum is complete, the accumulators must be cleared to start the next summation. This is illustrated by the signal “AINIT_0-3” in the simulation. Each new sum is then multiplexed onto a single output bus called “MUX_SEL”. A qualification strobe called “CORDIC_ND” is generated simultaneously with “MUX_SEL” to notify the CORDIC module that new data is available. The “CORDIC_ND” signal is controlled by the state machine so that it is suppressed initially until all invalid initial values have left the pipeline. This information is verified in the simulation.

Another simulation verifies the overall Four_Point_RMS module. This is shown in figure 6.5. This simulation shows a stream of input data called “INPUT_DATA” and “INPUT_RDY” being presented to the module. It also shows the corresponding output data stream called “AVERAGE_DOUT” and “AVERAGE_RDY”. The timing and value of both inputs and outputs are verified. An Excel spreadsheet is used to generate the expected values for this simulation, and they matched.

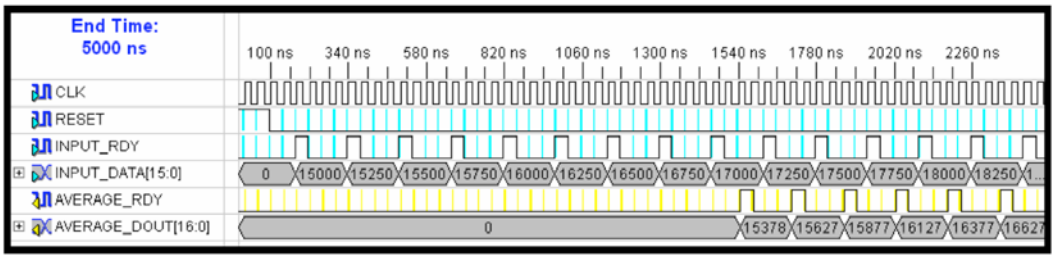


Figure 6.5 Four-Point RMS Module Simulation

All VHDL simulations are verified via both the waveform and the text output log. They are compared with the calculated results from the Excel spreadsheet, and they all matched. The match verifies the functionality of the LPW algorithm. The text output log is retained for regression testing later on as the dissertation DVR effort moves forward.

6.3 Implementation

The final step of the design strategy is the implementation of the design on the FPGA. The implementation step involves processing the code in Xilinx ISE Project Navigator and generating a load for the FPGA. The load is then put on the FPGA via the board, and hardware verification is performed.

The processing results from the Xilinx ISE Project Navigator tool indicated that the design synthesis and place and route are completed successfully. The reports from these processing steps also indicated that there is still much room left in the FPGA for additional logic. This means that the approach in optimizing resource utilization early in the analysis step proved to be effective.

Before the design is implemented on the FPGA, the board where the FPGA resides need to be checked out first. This is to ensure that both the board in general and the FPGA are functional. A board test software developed by the board manufacturer XESS is included with the board. This board test was run, and it showed that the board is in general functional. A snapshot of this process is shown in figure 6.6.

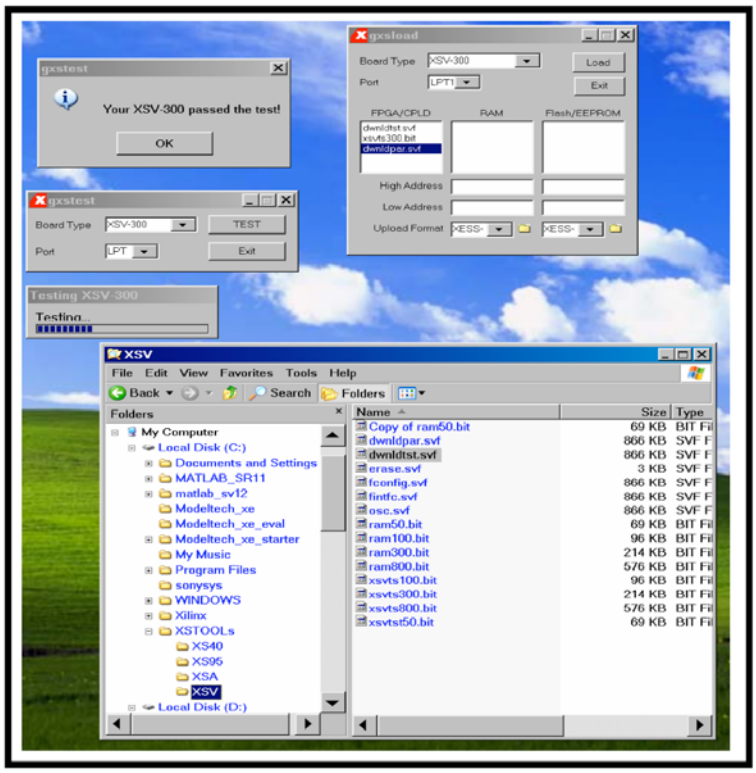


Figure 6.6 FPGA Board Test Process

The board test process involved loading the FPGA with an initial load first. Once there is a load in the FPGA, the type of board and the port location on the computer are identified on the board test software. The board test is then run. If the board test is successful as shown in figure 6.6, a message pops up to indicate that the board passed the test. If the board test is not successful, the message will indicate otherwise. This test does not cover other minor components on the board.

Once the board is deemed functional, the LPW is implemented on the FPGA. This is done via processing the code using Xilinx ISE Project Navigator. A load is then generated and loaded on to the board. (For more information on load generation, the fundamental of FPGA design is discussed in chapter 4.)

To verify board implementation, the most direct method is to obtain a visual of the FPGA output. There are many ways to do this. One method is to see the FPGA output through a scope or any measurement device. Another method is to store the output of the FPGA on to the RAM on the board. The latter is often preferred since it avoids the hassle of obtaining expensive measurement equipment.

There are four RAMs located on the XESS board, two on each side of the FPGA. For the implementation test, the input is loaded into the left banks of the RAMs, and the output is stored in the right banks of the RAMs. The board software is used to load and unload data to and from the RAMs. The configuration of the RAMs and the FPGA on the board is illustrated in a block diagram shown in figure 6.7.

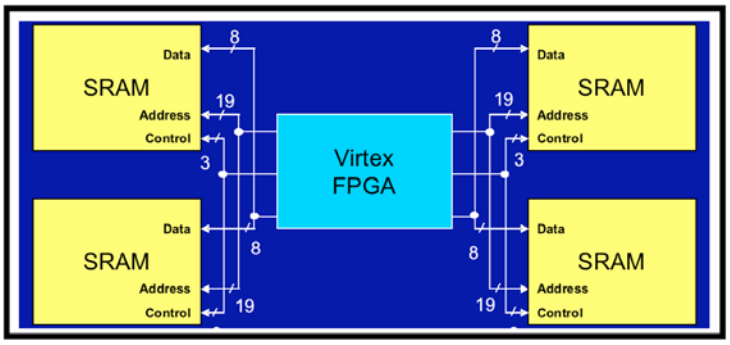


Figure 6.7 RAMs and FPGA Configuration

Additional VHDL code needed to be written in order to address and control the RAMs and then to collect the output data. The output of the RAM is collected in XESS format, and it was then converted into a readable form in Microsoft Excel. The results from the board implementation matched both the simulated results and the calculated results. The match of all three outputs verifies the functionality of the LPW design.

The verification of the LPW design is taken another step further to demonstrate that the algorithm is functional at real-time. This is an important step because ultimately the dissertation DVR needs to respond to voltage sags in real-time. In real-time operation, a display is needed to capture the output of the FPGA. Since there are no suitable displays on the board, a display outside the board is added. A simple and cost effective seven-segment display is used. To use the seven-segment display, the data needs to be converted through an Analog/Digital (A/D) converter first.

It was during this process that the A/D converter was discovered to be non-functional. It was also during this process that the oscillator clocking the A/D was discovered to be at a different frequency from the operating frequency of the design. To fix these issues, the XESS board was swapped out from an XSV 300 to an XSV 800. Both boards were essentially the same except for the FPGA. The XSV 300 had an FPGA with less amount of gates compared to the XSV 800.

In addition, a new oscillator was purchased, and it was placed externally along with the seven-segment display. The A/D converter was rewired to use the external oscillator. The A/D converter is located within a Stereo COder-DECoder (CODEC) IC, part AK4520A. The serial output of the A/D converter is accepted by an FPGA I/O pin and is converted to parallel form by the FPGA. The left channel of the CODEC is used at this stage to bring in the digitized data.

The seven-segment displays are set up on a bread-board. The bread-board is then connected to the XESS board via wires. The lay out of the XESS board with the external seven-segment displays on the bread-board is shown in figure 6.8.

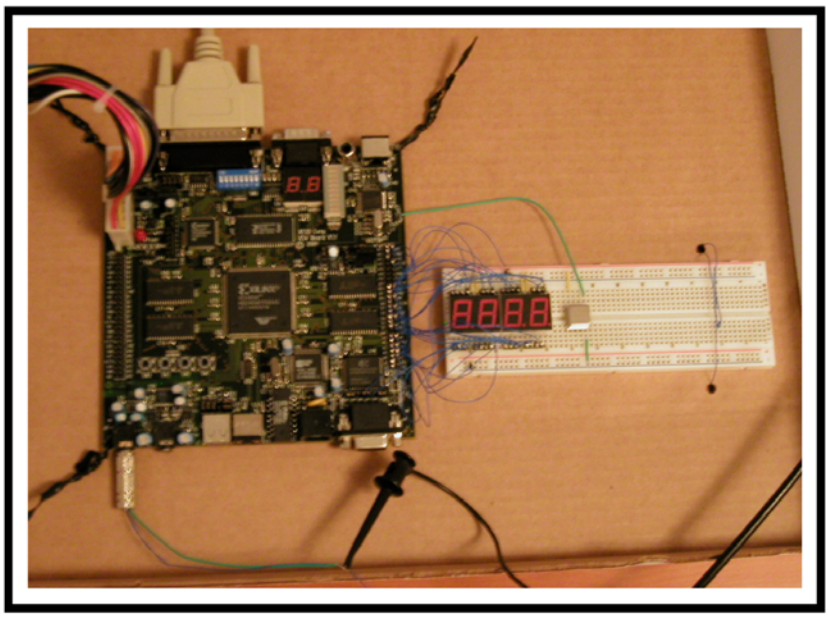


Figure 6.8 FPGA Board With Real-Time Displays

Additional VHDL code is added to the FPGA design to send the data out via the A/D converter. The output from the FPGA streamed through the seven-segment displays. The output data was displayed in real-time and passed by too quickly to accurately verify the results. A feature to pause the data and restart the data flow was added via the push-buttons SW1 and SW2 on the XESS board. Push-button SW1 resets and initializes the display. Push-button SW2 captures the output of the LPW algorithm and displays it.

To verify that the A/D process itself was operational, a function to capture input data directly from the A/D converter was added via another push-button. Push-button SW3 captures sixteen samples of input data after it has been digitized through the A/D converter. Push-button SW4 was not used at this phase of testing. A closer view of these push buttons on the XESS board is shown in figure 6.9.

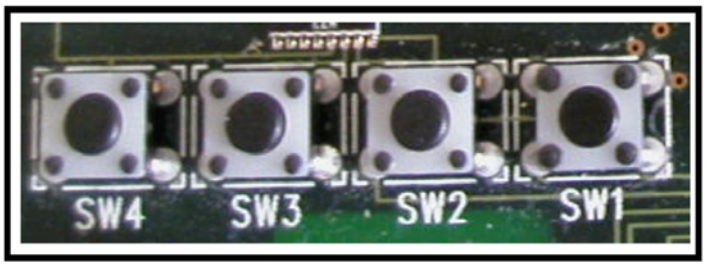


Figure 6.9 FPGA Board Push-Buttons

The data collected from the seven-segment displays matched the previous board implementation output and both the calculated and simulated results. This verified that the real-time output of the LPW algorithm is functional. A snapshot of the seven-segment displays controlled by the push buttons is shown in figure 6.10.

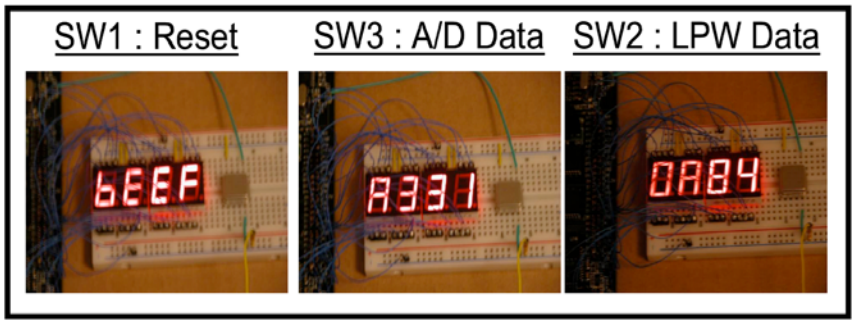


Figure 6.10 Seven-Segment Displays Via Push-Buttons

The output data from the seven-segment displays was collected and plotted via Microsoft Excel to further verify that the embedded harmonics correction within the LPW design is also functional. The input for this test is generated with a function generator. A snapshot of the function generator is shown in figure 6.11.

Figure 6.11 Test Setup Function Generator

The function generator stepped through the frequencies 0 to 240 Hz. Data is captured at each step. The results are then compiled in a spreadsheet and plotted. The output plot from this test showed a wave where the frequencies 120, 180, and 240 were rejected. It also showed that the amplitude at 60 Hz is maximized. This plot showed that the LPW algorithm is implemented as designed and is shown in figure 6.12.

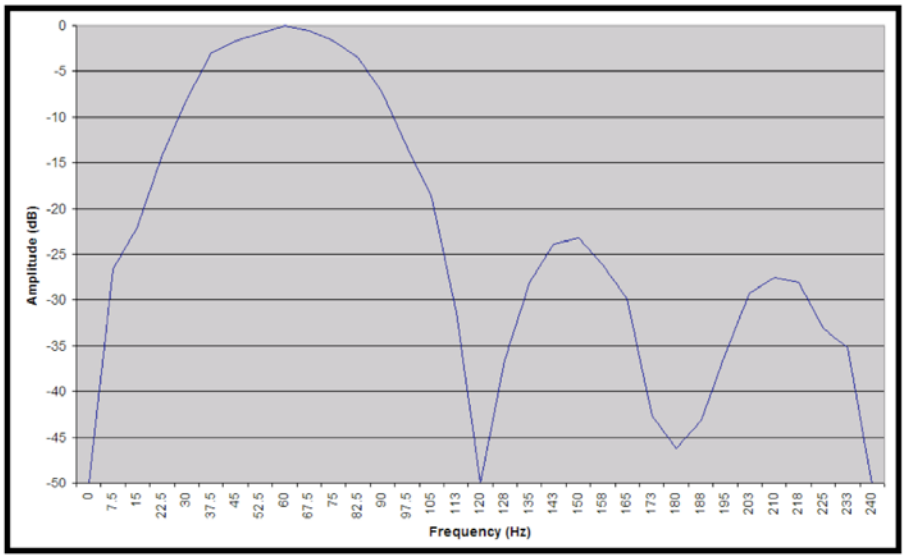


Figure 6.12 Real-Time LPW Output Plot

The LPW design includes two of the dissertation DVR features, the WT based filter and parts of the harmonics correction. Analysis of the LPW algorithm provided

important insights, such as multiple cores were utilized in order to optimize the FPGA resource utilization. The testing and implementation of this design was extremely tedious, but it helped to establish a baseline for the overall dissertation DVR effort. The testing determined the functionality of the design, and the implementation determined the hardware's viability.

CHAPTER 7

DISSERTATION DVR SYSTEM DESIGN

The dissertation DVR system design chapter covers the remaining essential parts of the design that is not covered under chapter 6. It primarily includes parts of the system that are dependent on system interaction for verification; this covers both FPGA design and hardware implementation.

7.1 Initial Hardware Configuration

In additional to the FPGA board, there are other hardware parts that are essential to the dissertation DVR. The hardware is primarily divided between the FPGA board, the VSC, and the transformers. (For more information, the fundamentals of the dissertation DVR are discussed in chapter 3.) Additional hardware components such as those of the VSC are implemented on an additional prototyping board. The system block diagram of the dissertation DVR is shown in figure 7.1.

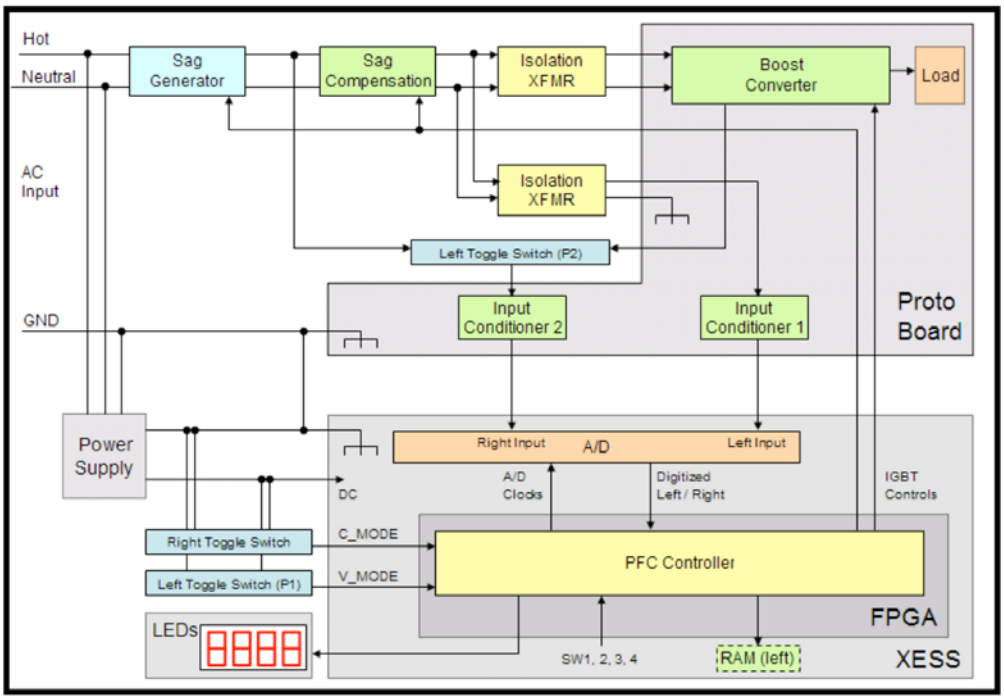


Figure 7.1 Dissertation DVR System Block Diagram

The system block diagram shows that the additional prototyping board consists of two Input Conditioners, a Boost Converter, and a test Load. The prototyping board is interconnected to the XESS board via wires. Two isolation transformers were mounted separately and connected to the prototyping board.

The XESS FPGA board includes the voltage sag detection and the power factor correction controller. The real-time functions from the LPW verification in chapter 6 is also shown in the system block diagram. This includes the seven-segment displays and both the A/D converter and the RAM, which are provided as part of the XESS FPGA board. The initial hardware configuration of this system is shown in figure 7.2.

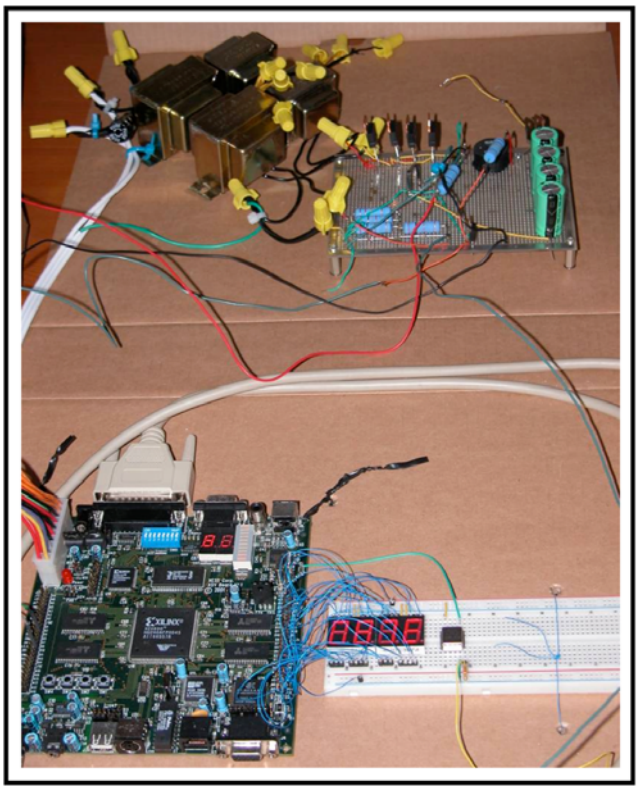


Figure 7.2 Initial Hardware Setup

The input conditioner blocks on the prototyping board are designed to protect digital circuits, primarily the XESS FPGA board. This is because the dissertation DVR consists of a mixture of both very high voltage and very low voltage components. The outcome of both types of components coming into contact with each other without first stepping the voltage down is catastrophic. The input conditioners are needed to prevent such an event.

The input conditioner consists mainly of an isolation transformer and a resistive divider. The AC input is first brought in from the outlet through the isolation transformer. This isolates the digital circuit from any transients or noise. It also ensures that the low voltage circuit is not exposed to any unreferenced AC voltages.

From the isolation transformer, the voltage is then lowered via a resistive divider to approximately 1.2 V RMS. This is within the voltage tolerance of the A/D converter on the XESS FPGA board. A series resistor and dual Zener diode are also added to provide overvoltage protection. Both input conditioners are designed similarly. The left channel of the A/D is driven by the first input conditioner via a standard stereo jack. The left channel input is the compensated line voltage waveform. The right channel of the A/D is driven by the second input conditioner via the same stereo jack. The right channel input is selected by a front panel switch between the uncompensated line voltage waveform and the PFC current waveform.

Another path does not directly input to the A/D; instead, the full-scale compensated line voltage waveform is input to the Boost Converter via an isolation transformer. The Boost Converter is designed to provide the power factor correction. It consists of a series inductor that is used to charge a series capacitor bank through a diode. The capacitor bank serves as the load of the system and nothing else. When the IGBT is in the “on” state, it is used to charge the inductor. When it is in the “off” state, it allows the capacitor bank to be charged as the inductor discharges. The capacitor bank is only large enough to withstand the 400V DC on the output. The assumption behind this design is that with 170V peak voltage from the outlet, the output voltage will not exceed twice the input, which is 340V. Therefore, the design goal is set at an approximate maximum of 400V. As the Boost Converter operates, the IGBT is switched “on” and “off” repeatedly. This allows the capacitor bank to be maintained at a high, relatively stable DC voltage, and the power factor is corrected by maintaining this load.

A schematic of the Boost Converter design used in the dissertation DVR is shown in figure 7.3.

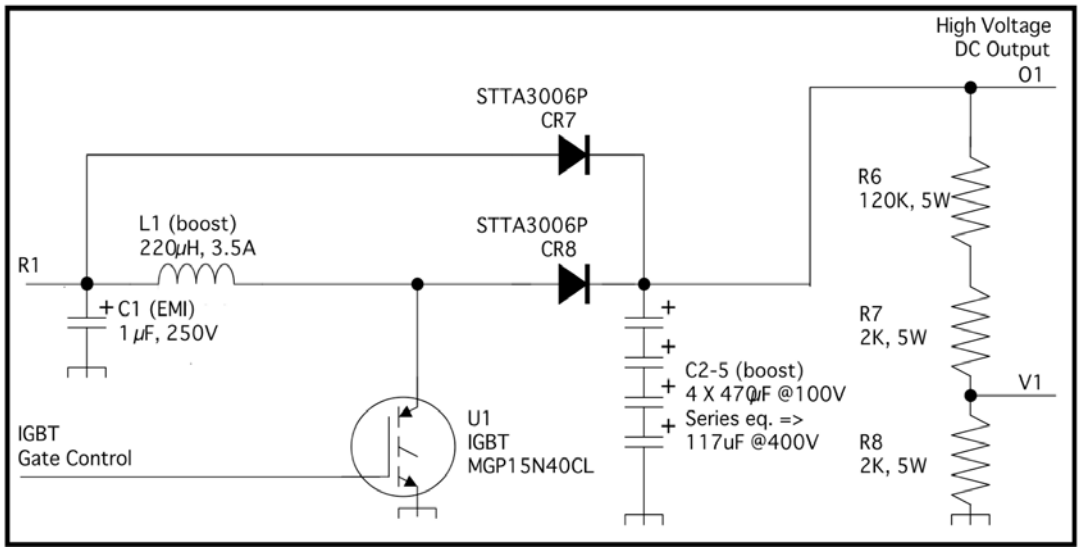


Figure 7.3 Boost Converter Schematic

For the voltage compensation part of the dissertation DVR, two transformers were set up in a similar configuration as the proof of concept configuration in chapter 5. It consists of an isolation transformer and a 10 to 1 step up transformer. The isolation transformer serves to isolate transients and noise from the system, while the 10 to 1 transformer steps up the voltage as needed in order to compensate for voltage sags. The transformers are rated to have a five percent tolerance. This five percent tolerance does not exceed the ten percent in a voltage sag definition, so its compensation is still acceptable. The hardware configuration for the transformers is shown in figure 7.4.

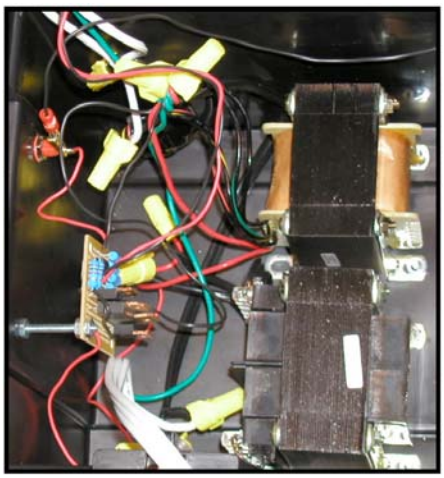


Figure 7.4 Voltage Compensation Transformers

At this point in the system design, the initial hardware configuration is essentially made up of stand alone hardware. This consists of the XESS FPGA board with the display bread-board, the prototyping board, and the transformers. These hardware components are needed for various functions within the dissertation DVR design. The next step is to control these stand alone hardware components so that they could work in conjunction with each other and as a system. The control is implemented in VHDL.

7.2 VHDL Control System

The control logic of the dissertation DVR is accomplished via VHDL. It is a control system primarily responsible for voltage compensation, harmonics correction, and PFC. All these functions must operate in unison with each other in order for the dissertation DVR to work. This means the timing is crucial.

The VHDL control system design adheres to the FPGA design strategy as discussed in chapter 4. As part of this design strategy under the analysis step, the

functions within this system are initially organized into blocks as necessary. A block diagram of the VHDL control system is shown in figure 7.5.

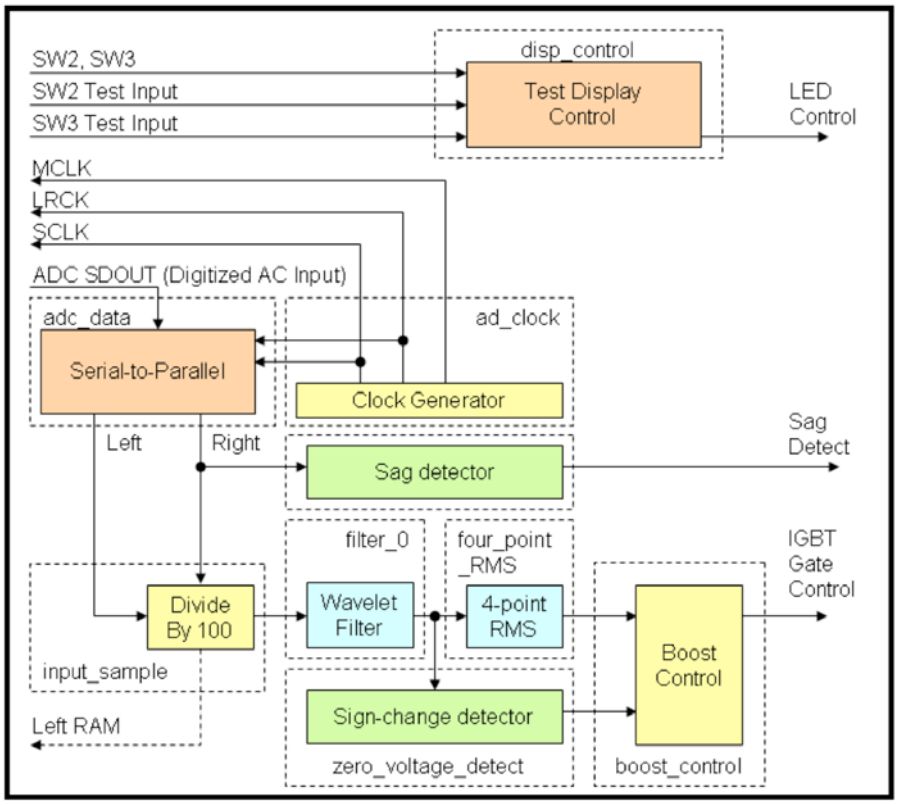


Figure 7.5 VHDL Control System

The VHDL control system is divided into blocks. Each block represents a module within the overall VHDL code. The blocks are divided by function. The block that is first designed is called the AD_Clock block. This is a module that serves as a clock generator for the system. It is used to create the required clock signals MCLK, LRCK, and SCLK. Multiple clocks are needed for different components of the design. For example, the A/D converter uses one clock, and the LPW uses another clock. Since only one external clock is provided via the external oscillator on the bread-board, all other clocks must be derived from this clock.

The AD_Clock block is the most important block in the system. Its importance stems from the fact that most of the other blocks in the system will run from the clock generated by this module. Thus, if the signal generated by this block is not clean, it will affect the performance of the overall system. In FPGA design, the clock signal is a special signal that must be treated differently from other signals. This means steps must be taken during the processing such that the tools recognize the intended signal as a clock. This was accomplished in the design using Xilinx ISE Project Navigator.

The AD_Clock block functions to accept a clock at one rate; it then derives other rates from it. It provides these other rates as outputs. For example, the left channel of the A/D converter is used to bring in the digitized line voltage. The clock at this stage is at 48 Kilo Samples per Second (KSPS). This clock is then decimated by 100 to produce the 480 Samples per Second (SPS) line voltage signal.

The 480 SPS line voltage signal is then fed to the filter module. The combination of Filter_0 and Four_Point_RMS is the LPW algorithm. This LPW algorithm is coded earlier in the design phase since its verification is independent of the overall system. (For more information on this module, it is discussed in chapter 6.)

The zero_voltage_detect module in the block diagram is designed to detect the zero crossing of the line voltage input. The zero crossing indicates the beginning of a new sinusoidal cycle. This module is used to signal the boost_control module of the zero crossing.

The sag detection module is used to detect the voltage sag. The details of this module are shown in figure 7.6. The input is from the left channel of the A/D converter.

This 48 KSPS input is used to detect zero crossings in order to synchronize the sag detection process. The delay and measure function provides precise control of the phase. The absolute value of the input is compared to a long term threshold value in order to determine the presence of a sag.

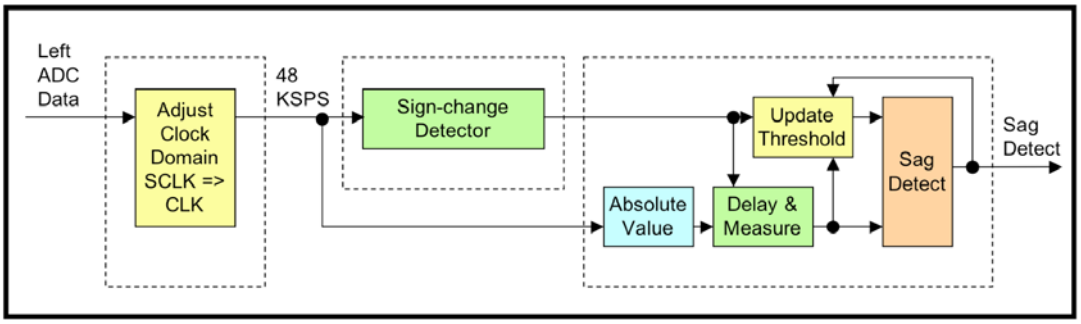


Figure 7.6 Voltage Sag Detection

The boost_control module is used for power factor correction. It controls the pulsing of the IGBT. This is done by turning the IGBT gate “on” and “off”. The signal from the zero_voltage_detect helps to time the pulses, and the signal from the LPW module serves as a handshake between the LPW module and the boost_control module.

After the VHDL control system is coded, testing is performed next. Each module is simulated by itself first to determine its timing and to ensure that it will interface to neighboring modules correctly. Many simulations were performed on the system design. The only exception is the voltage sag detection module. No simulations were performed on the voltage sag detection module. This module involved controlling the IGBT hardware along with the test voltage sag in order to determine its performance, so the testing of this module is scheduled for the field test phase of the device. Some of the more critical simulations are discussed as follows.

The first simulation is performed to verify the functionality of the external oscillator and the clock generator module. This simulation is shown in figure 7.7. The signal “MCLK” is the master clock for the CODEC part. The signal “SCLK” is the synchronization clock for the A/D converter, and “LRCK” is the CODEC left/right channel indicator. This simulation primarily verifies the timing between these three signals.

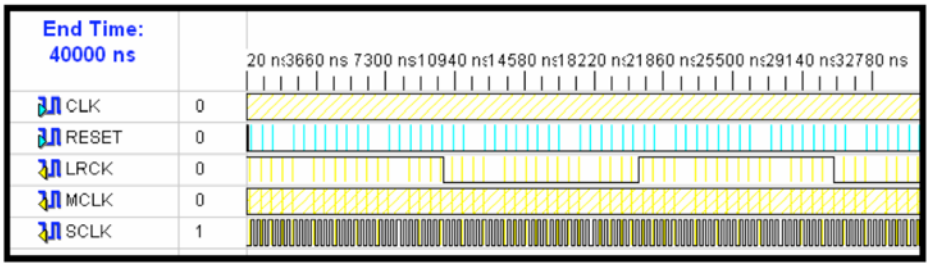


Figure 7.7 Clock Module Simulation

The next simulation is to verify the data capture from the A/D converter. This is shown in figure 7.8. The purpose of this simulation is to determine that the data is being clocked into the system correctly. The data is brought in to the XESS FPGA board via a CODEC. The CODEC consists of two channels interleaved. The system uses both the left and right channel. To differentiate between the channels, the input data stream had to be de-interleaved. This simulation verifies that the data is being de-interleaved properly.

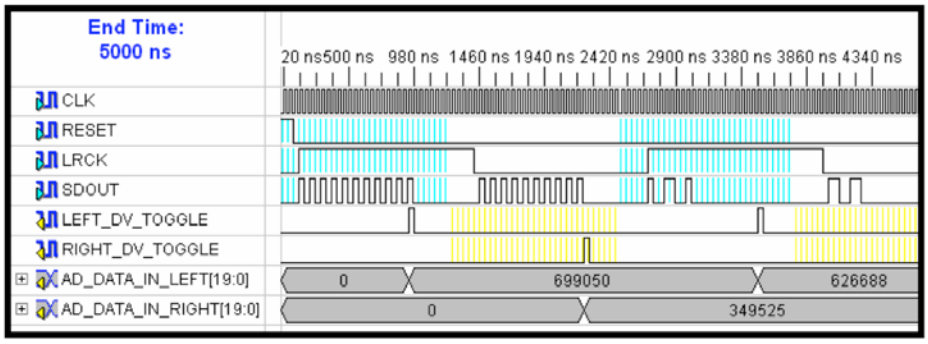


Figure 7.8 De-Interleaved Data Simulation

The next simulation is to verify the decimation function of the Input Sample Module. The primary purpose of the Input Sample Module is to decimate the output from the CODEC from a nominal sample rate of 48 KSPS down to a rate of 480 SPS. The strobe signal “FILTER_ND” is used to indicate when the output of the module is valid. The 16-bit data bus “FILTER_DIN” is used to carry the decimated output samples. In order to be compatible with the FIR filter in the next module, the output sample stream is reduced to a width of 16 bits. These bits come from the 16 most significant bits of the original 20-bit CODEC sample. The secondary purpose of this module is to write the sample stream into the left bank of the XESS card RAM for test purposes; auxiliary RAM control signals, including “L_ADDRESS”, “L_CE_N”, L_OE_N”, and “L_WE_N” are provided for this purpose. As shown in figure 7.9, the simulation provides the 20-bit parallel sample stream “AD_DATA_IN_LEFT” from the serial-to-parallel converter, along with the qualification strobe “LEFT_DV_TOGGLE”.

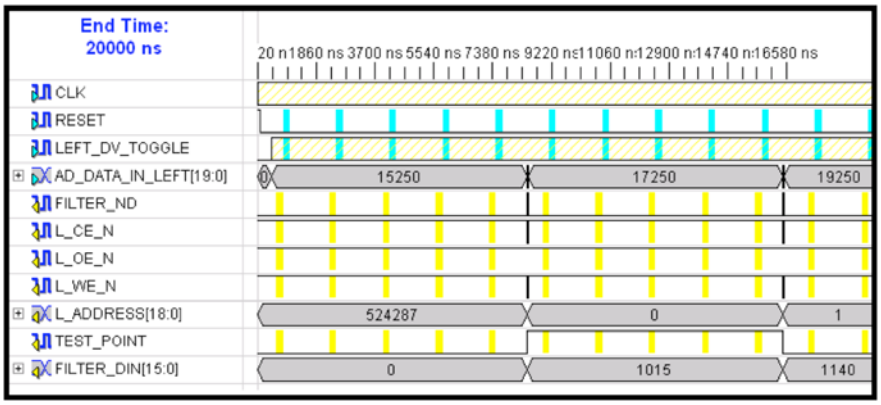


Figure 7.9 Input Sample Module Simulation

The next simulation verified the Zero Voltage Detector Module. The primary purpose of this module is to produce an output toggle pulse each time the input passes through zero. As shown in figure 7.10, a simulation was performed to verify this behavior; a continuous input sample stream, “FILTER_DOUT” and “FILTER_RDY”, was provided to the module with several embedded zero crossings. The simulation in figure 7.10 shows one of these crossings. The detection strobe, “ZV_DETECT”, was observed to toggle correctly at each zero crossing.

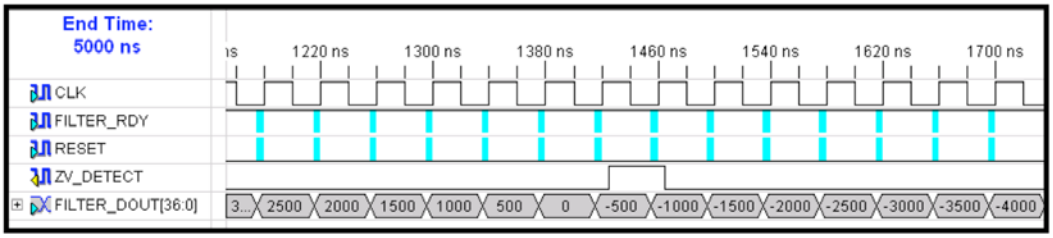
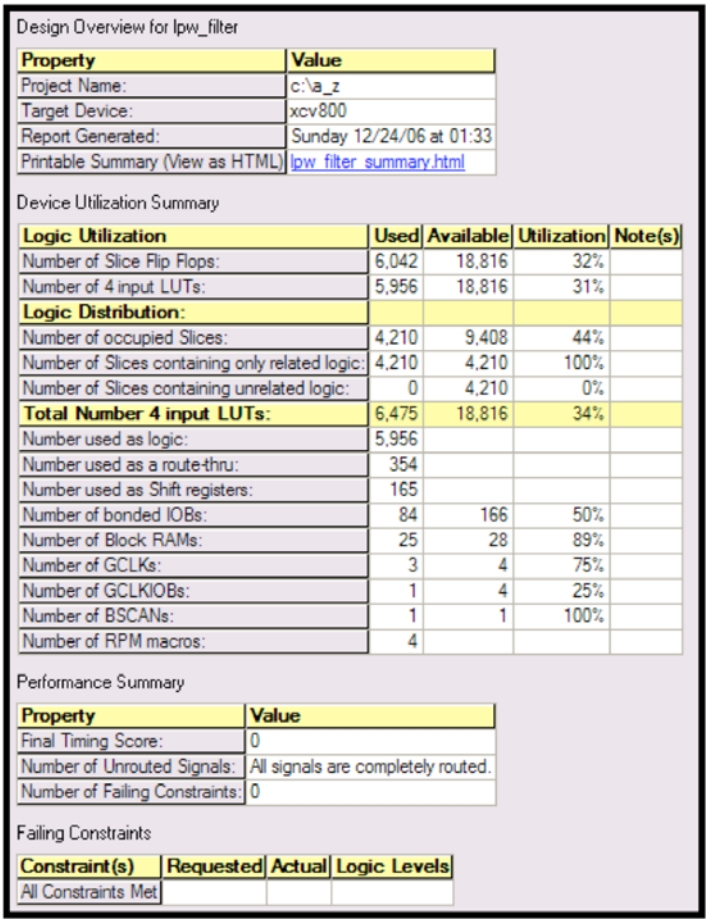


Figure 7.10 Zero Crossing Detector Simulation

The next step after testing is to implement the control system. This is an initial implementation of the control system. This step will be repeated as needed until the dissertation device is operating as expected.

The VHDL code of the system is processed by Xilinx ISE Project Navigator. The processing was successful, and the system is synthesized and placed and routed without any complications. A load is generated for the FPGA. The final resource utilization summary of the system is shown in figure 7.11. Based on this summary, the system fit into the FPGA without any complications. Furthermore, the summary indicated that there is still some room within the FPGA for additional functionalities.



Design Overview for lpw_filter

Property	Value
Project Name:	c:\a_z
Target Device:	xcv800
Report Generated:	Sunday 12/24/06 at 01:33
Printable Summary (View as HTML)	lpw_filter_summary.html

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops:	6,042	18,816	32%	
Number of 4 input LUTs:	5,956	18,816	31%	
Logic Distribution:				
Number of occupied Slices:	4,210	9,408	44%	
Number of Slices containing only related logic:	4,210	4,210	100%	
Number of Slices containing unrelated logic:	0	4,210	0%	
Total Number 4 input LUTs:	6,475	18,816	34%	
Number used as logic:	5,956			
Number used as a route-thru:	354			
Number used as Shift registers:	165			
Number of bonded IOBs:	84	166	50%	
Number of Block RAMs:	25	28	89%	
Number of GCLKs:	3	4	75%	
Number of GCLKIOBs:	1	4	25%	
Number of BSCANs:	1	1	100%	
Number of RPM macros:	4			

Performance Summary

Property	Value
Final Timing Score:	0
Number of Unrouted Signals:	All signals are completely routed.
Number of Failing Constraints:	0

Failing Constraints

Constraint(s)	Requested	Actual	Logic Levels
All Constraints Met			

Figure 7.11 System Resource Utilization Summary

7.3 System Checkout

Following the FPGA implementation, the auxiliary hardware on the prototyping board is further tested by checking it manually with an oscilloscope. A function generator is initially used to provide a low-level 60 Hz input to test the input conditioner and zero-crossing detector. This test input is transitioned over to the actual 120V AC waveform for the final testing. The oscilloscope trace of the output from the input conditioner is shown in figure 7.12.

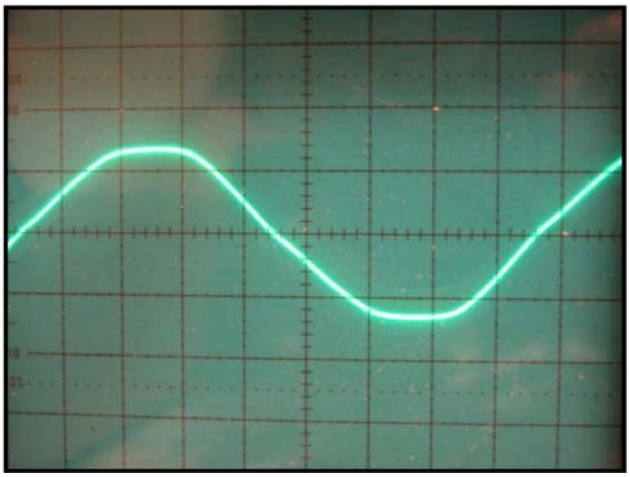


Figure 7.12 Oscilloscope Trace Of Input Conditioner Output

The oscilloscope trace showed that the input conditioner is performing as designed. The input waveform is sinusoidal and the peak voltage is within the design tolerance of the input conditioner. This checkout is performed over a duration of time to ensure that the input conditioner is consistent in its performance.

During the checkout, it was discovered that there is an issue with the capacitive load. It appears that the capacitive load develops nearly 160V across it during operation of the Boost Converter. This voltage would persist for many minutes after the device

was powered off and thus creating a safety hazard. A resistor was added in parallel with the bank to discharge the load quickly after it is powered off.

It was also discovered during checkout that the isolation transformers would get very hot over time during device operation. This posed a safety concern, so these transformers were replaced with new isolation transformers. The ratings are still the same, but the composition itself is sturdier and had more windings.

The checkout process verified the IGBT functionality next. Since it is designed to switch high voltages, control of the IGBT is challenging. If the switch is not controlled precisely, it could lead to catastrophe and serious injury. To minimize the potential for catastrophe, the IGBT control does not control the transformers directly. Instead, it drives an optocoupler which then drives the transformers. The optocoupler is an optical IC that makes the connection between the IGBT and the transformer via optics. This prevents accidental shorts between the IGBT and the transformers. Control of the IGBT is verified with an oscilloscope. The oscilloscope trace of this function showed that it is operating as designed. The oscilloscope trace of the IGBT gate control function is shown in figure 7.13. The trace shows the control switching “on” and “off”.

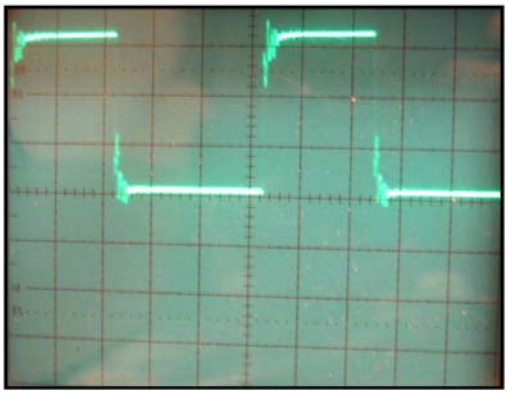


Figure 7.13 Oscilloscope Trace Of IGBT Gate Control

The system checkout is completed successfully. The next step is to demonstrate that the dissertation DVR functions as a device. This was more tedious than expected and is discussed in the following chapter.

CHAPTER 8

DISSERTATION DVR RESULTS

The final step of the dissertation DVR effort is to demonstrate that it functions as a device. In addition to requiring visibility other than an oscilloscope, the dissertation DVR also needed to be portable so that it could be demonstrated outside the lab.

Many methods to demonstrate the dissertation DVR were considered. In addition to functionality, the time, effort, and resources had to be taken into account as well. It is not practical or desirable to spend more time, effort, and resources to implement the demonstration function than it took to implement the actual device under test. With these considerations in mind, the Xilinx Chipscope tool was selected for this task.

Chipscope is a tool that functions similar to a logic analyzer. It is commonly used for checkout and demonstrations of FPGA based devices. To implement it, a VHDL module has to be instantiated into the project and connected to selected signals without disturbing the original design. This is only possible if there are enough remaining resources, such as gates and RAMS, within the FPGA. Fortunately, additional room is still available within the FPGA to implement chipscope.

The IEEE-1149 Joint Test Action Group (JTAG) interface on the FPGA is used to access this analyzer from a PC parallel port for control and display purposes. In order

to make Chipscope work with the XESS board, it was necessary to re-program the XESS Complex Programmable Logic Device (CPLD) with a special program in order to make the card compatible with Chipscope. This step is typically not necessary for newer generations of FPGAs.

It is very important to note that Chipscope captures the signals of every bit. This means that the analog looking waveforms shown in the Chipscope views need to be translated from a digital form first. The translation involved combining and converting the proper bits.

The next step in the demonstration is to provide test cases for the functions of the dissertation DVR. This meant a voltage sag needs to be re-created so that the voltage compensation function could be demonstrated. This was tedious, and after several attempts, a voltage sag is implemented via a step-down transformer. There are two test cases. The first test case voltage sag is approximately a ninety percent sag with a duration of three cycles, and the second test case voltage sag is approximately a ninety percent sag with a duration of half a cycle. These test cases are illustrated using chipscope with their compensations in figures 8.1 and 8.2, respectively. As for both the PFC and the harmonics correction, the load on the prototyping board provided for both their test cases.

The initial chipscope views of these functions showed erratic spikes in the compensated and corrected waveforms. After extensive investigation, it was determined that the A/D converter section on the XESS FPGA board suffered some damage. The cause of the damage was believed to be from the high voltage. This is prior to the

addition of the input conditioner to protect the system. Fortunately the damage is limited to the resistive network supporting the A/D converter section. Some resistors in this network were shorted, and new resistors had to be soldered onto the board and externally in order to remedy this problem. This fix salvaged the effort, and the erratic spikes were eliminated from the compensated and corrected waveforms as shown in figures 8.1, 8.2, 8.3, and 8.5. The new resistance values did not exactly match the original resistance values, so the scale between the left and right channel of the A/D converter will be slightly different from each other. Its effect may or may not be visible from the chipscope views.

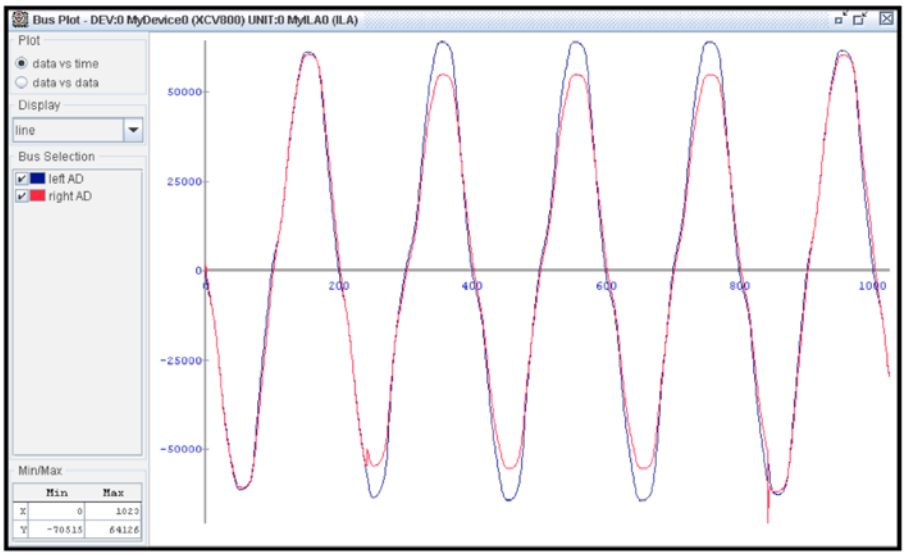


Figure 8.1 Voltage Sag Compensation 1

Figure 8.1 shows the voltage compensation of the dissertation DVR with the first test case. The first test case has a ninety percent voltage sag that lasts for a duration of three cycles. The voltage sag condition is shown in red, and the voltage compensation is shown in blue. Note the spike in the lower right corner of the voltage

sag, and note how the compensated waveform also corrects for it. Every sinusoid shown represents one 60 Hz cycle.

Figure 8.2 shows the voltage compensation of the dissertation DVR with the second test case. The second test case has a ninety percent voltage sag that lasts for a duration of half a cycle. The voltage sag condition is shown in red, and the voltage compensation is shown in blue. The compensation waveforms in both test cases show a slight increase in the magnitude. The transformer configuration is rated to have a five percent tolerance, so this variance is due to that tolerance.

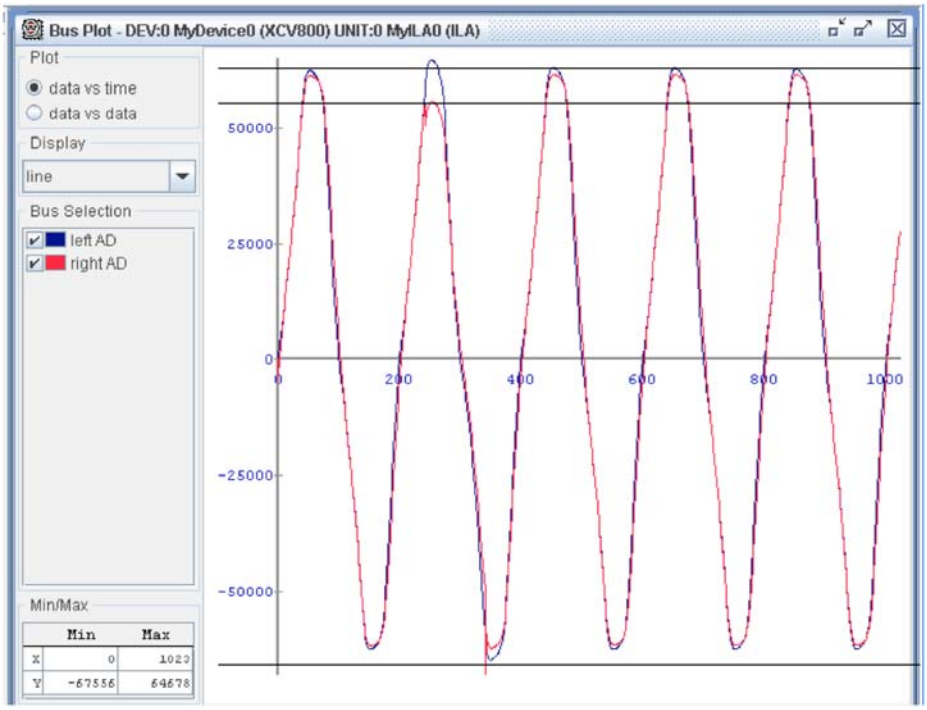


Figure 8.2 Voltage Sag Compensation 2

The voltage sag detection was implemented in VHDL. Its performance could not be simulated since it involved controlling the IGBTs hardware. Its response is shown in figure 8.3 in chipscope. The test voltage sag is ninety percent sag with a

duration of half a cycle and shown in the red waveform. The voltage compensation is shown in the blue waveform. Measurement is taken and shown in the figure. The response time in terms of how fast the VHDL could respond to the voltage sag is shown and pointed by the arrow on the figure. The response time is eleven clocks. One clock is approximately 83.333 microseconds. The voltage sag detection response is approximately 916.667 microseconds. (11* 83.333~916.667) This is a fast response time; it is faster than half a cycle and indicates that the parallel operation of the FPGA has improved the response of the DVR.

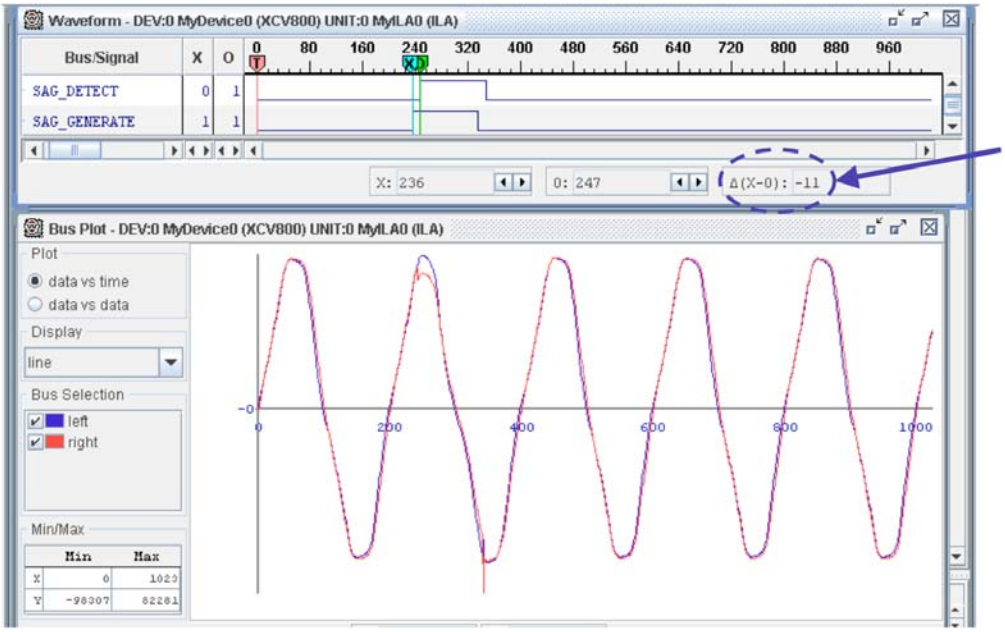


Figure 8.3 Voltage Sag Detection Response

Figure 8.4 shows the voltage and current waveform without PFC in chipscope, and figure 8.6 shows the voltage and current waveform with PFC in chipscope. The combination of figures 8.1, 8.2 and 8.6 shows a clean sinusoidal waveform after the compensation and correction. This demonstrates that any harmonics present in the input

are continuously being corrected as part of the power factor correction. This harmonics correction works in conjunction with the LPW because the LPW has an inherent delay in the algorithm. VHDL code was written to compensate for it.

The next four plots show the power factor correction and its harmonics correction to compensate for the LPW. The voltage waveform is shown in blue, and the current waveform is shown in red. Figure 8.4 shows that the power factor is not corrected, and the current is not in phase with the voltage. The shape of the current in this figure also shows that it is a nonlinear load, which is causing the power factor problems.

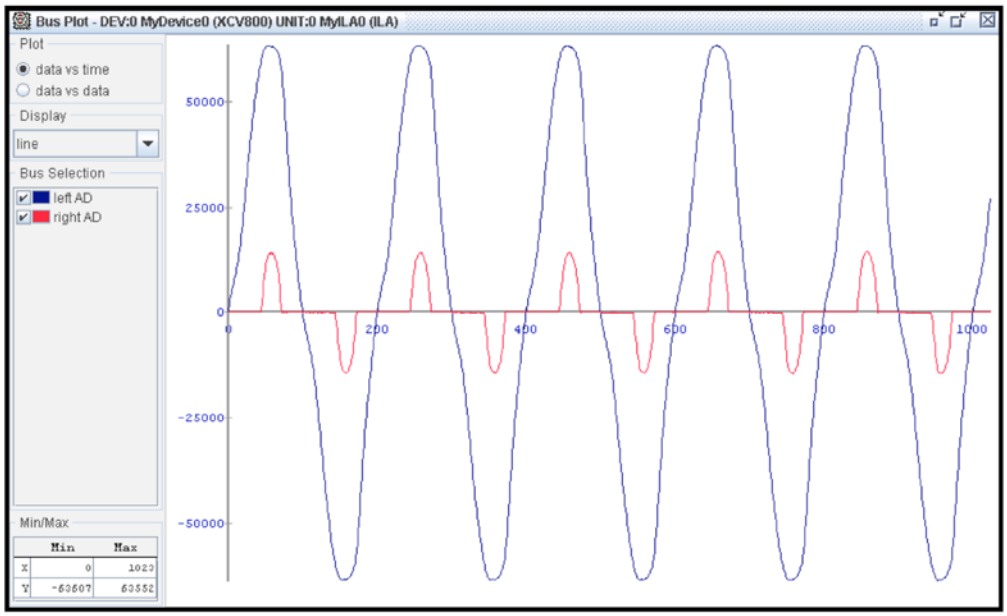


Figure 8.4 Without Power Factor And Harmonic Correction

A Fast Fourier Transform (FFT) analysis of the waveforms in figure 8.4 is performed to quantify the harmonics before the correction. The FFT analysis is using a minimum 4-term Nuttall window [52, 53], and the Total Harmonic Distortion (THD) is

also calculated from this data. The FFT analysis is shown in figure 8.5, and the THD is calculated to be 0.505 dB. This shows that there are harmonics present such that its sum is greater than the fundamental frequency. The uncorrected magnitude of the individual harmonics as a percentage of the fundamental frequency is quantified in table 8.1.

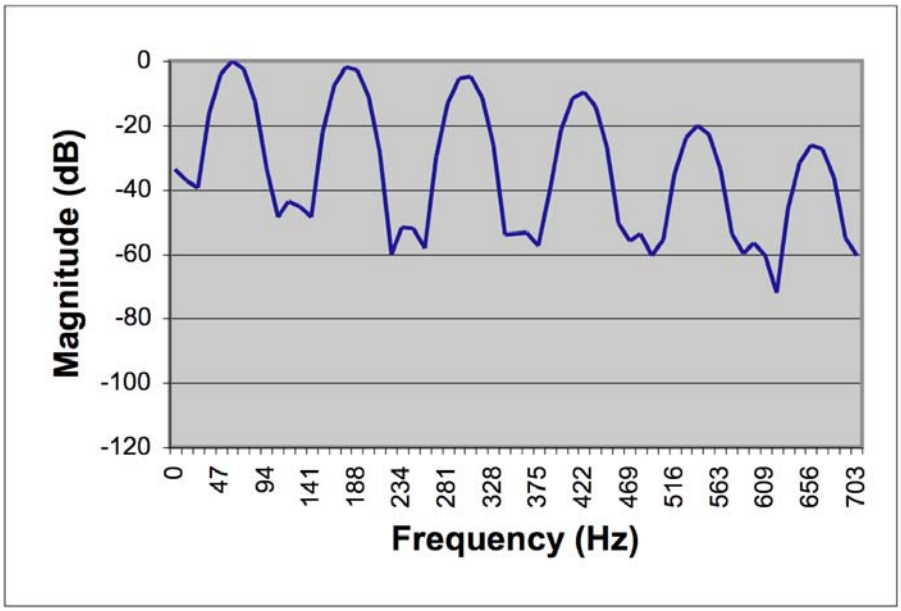


Figure 8.5 FFT Analysis Before Correction

Table 8.1 Uncorrected Harmonic Magnitudes

Uncorrected Harmonic Magnitudes

Hth	Freq(Hz)	% Abs
1	60	100.000
3	180	80.916
5	300	57.018
7	420	33.109
9	540	9.955
11	660	4.987
13	780	9.921
15	900	8.749
17	1020	3.862
19	1140	1.197
21	1260	3.420
23	1380	3.283

Figure 8.6 shows the voltage and current waveform with power factor correction in chipscope. The current is scaled up so that it could be seen more clearly. Both the voltage and the current waveform are in phase with each other. This shows the power factor is corrected.

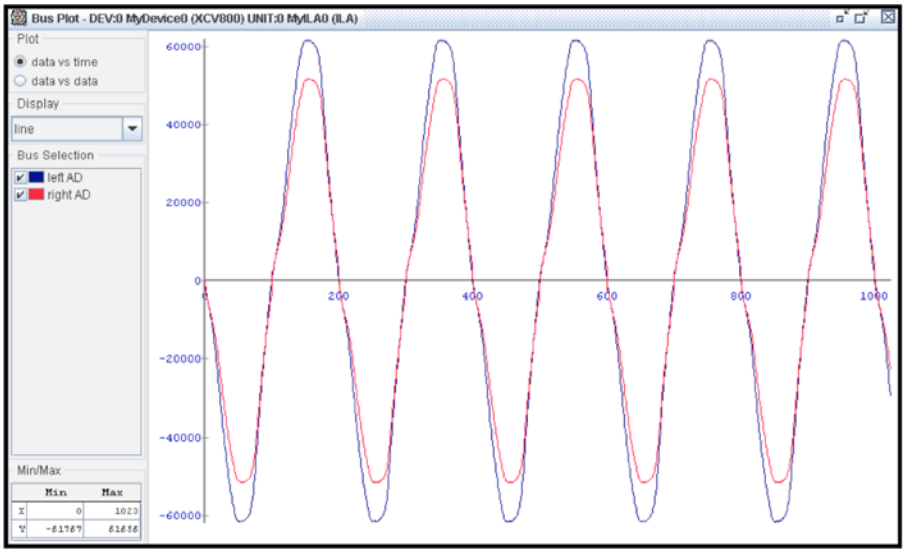


Figure 8.6 With Power Factor And Harmonic Correction

An FFT analysis of the waveforms in figure 8.6 is performed to quantify the harmonics after the correction. The FFT analysis is using a minimum 4-term Nuttall window [52, 53], and the THD is also calculated from this data. The FFT analysis is shown in figure 8.7, and the THD is calculated to be -19.987 dB. This shows that the harmonics are suppressed such that their sum is less than the fundamental frequency. The low THD indicates that the performance of the harmonics correction is functional. The corrected magnitude of the individual harmonics as a percentage of the fundamental frequency is quantified in table 8.2.

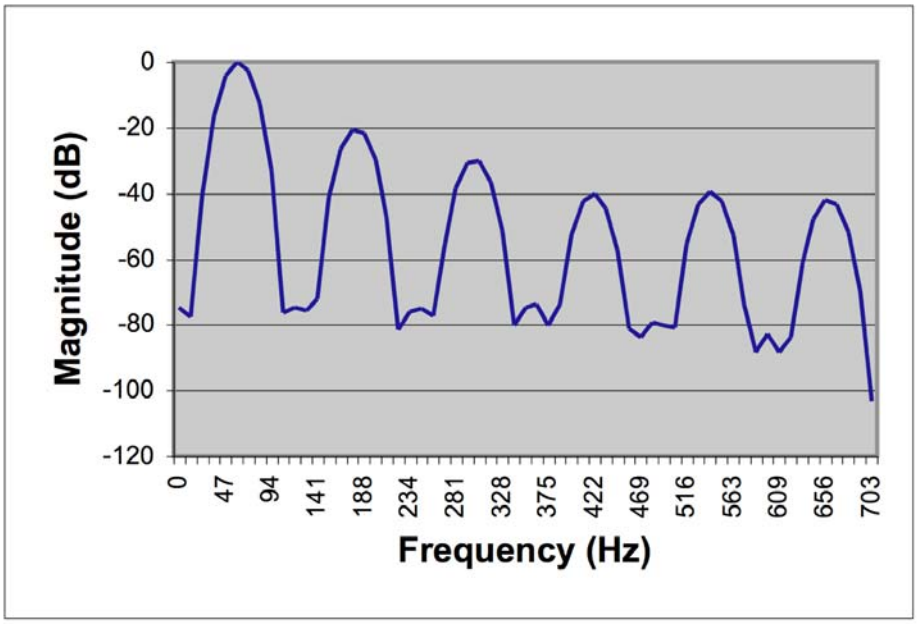


Figure 8.7 FFT Analysis After Correction

Table 8.2 Corrected Harmonic Magnitudes

Corrected Harmonic Magnitudes

Hth	Freq(Hz)	% Abs
1	60	100.000
3	180	9.342
5	300	3.088
7	420	0.982
9	540	1.062
11	660	0.784
13	780	0.306
15	900	0.642
17	1020	0.299
19	1140	0.255
21	1260	0.280
23	1380	0.196

The final step after the demonstration via chipscope is to ensure that the dissertation DVR could be demonstrated outside of the lab. To accomplish this, the dissertation DVR had to be packaged in a way that makes it portable. Several attempts were required to get it in the current form. The final dissertation DVR consists of two cases and a laptop computer. The first case contains the digital components housed inside a pc chassis. The second case housed the bulky transformers inside a tool box. The two cases are shown in figure 8.8.

Figure 8.8 Dissertation DVR

The demonstration of the dissertation DVR's capabilities proved to be more difficult than was originally conceived. This step was accomplished, and it was worthwhile. It provided a way to show that the dissertation DVR is functional outside of the lab environment.

CHAPTER 9

CONCLUSIONS

DVRs are effective devices for mitigating power disturbances such as voltage sags. The impact of voltage sags is severe, and it ranges from disruptions to the load to economic losses. Therefore, DVRs are considered an efficient solution to protect sensitive equipment against these impacts due to its low cost, small size, and dynamic response.

There are many types of DVRs. This dissertation focused on the In-Phase Control DVRs; these are the types without energy storage. The dissertation DVR mainly consists of an isolation transformer, an IGBT based VSC to inject the compensation voltage, and an FPGA based controls to detect and manage the disturbance.

The FPGA based controls are a large part of what makes this dissertation DVR novel. Included in the FPGA is the Wavelet Transform based filter algorithm. This is different compared to the conventional FT algorithm and PLLs used in other DVRs. Furthermore, by implementing the controls inside the FPGA, the speed and performance of the device are optimized. This process also allows for both reusability and reconfigurability of the dissertation DVR. This means that much of this dissertation DVR design could be ported over into another FPGA board, and the design could be

reconfigured easily. In addition, features such as harmonics correction and power factor correction were added to the dissertation DVR to further enhance its capabilities.

The final implementation of the dissertation DVR, with its combination of analog and digital circuitry, proved to be challenging. There are many issues to consider. Combining very high voltage from power systems with very small voltage within digital circuits requires many precautions. Despite this, the dissertation DVR is realized, and it demonstrated its capabilities as described.

CHAPTER 10

RECOMMENDATIONS FOR FUTURE WORK

Based on this experience, several ideas came to mind on ways to improve the dissertation DVR. One of these ideas is in regards to the FPGA. FPGAs have come a long way, and they are still continuing to improve. Future work could include migrating the current FPGA design into the latest type of FPGAs. One of the latest FPGAs in work touts its footprint compatibility, more gates, increase speed, embedded processor, and more embedded cores. With these improvements, the current FPGA design could further improve in speed and double in functionalities, or it could even include two dissertation DVR controls in one chip.

Another idea is in regards to the VSC. The VSC in the dissertation DVR is based on the IGBT. VHDL code had to be written in the FPGA to control the IGBT. This could be improved and simplified if the IGBT came with these controls built into the device already. Sophisticated IGBTs are currently in the works and could become available in the future. They include intelligent IGBT modules that have diagnostic and control logic integrated with the device [27].

Another idea is that instead of the IGBT, the VSC could be based on another type of VSC. The IGBT was selected over other types because both its switching speed and power capability suited the dissertation DVR. The MOSFET was not selected

because even though it has a very fast switching speed, its power capability was not suitable for the DVR. This device could become a contender if its power capability increases.

One type of MOSFET currently in the works is based on cryo-MOSFETs [50]. Cryo-MOSFETs are MOSFETs that operate at 77K, and its advantage over room temperature MOSFETs is that the on-resistance is much lower [34]. This would improve its power capability. This could be a possible contender depending on its ease of use and if it becomes available in the future.

Another type of switch that could be considered in the future is the Integrated Gate Commutated Thyristors (IGCT). IGCT is the newest member of the power semiconductor family and was introduced by ABB in 1997 [46]. It is based on a combination of the GTO thyristors and IGBT. It is capable of handling a very high voltage and is claimed to be superior to the GTO [46]. The IGCT merges the attractive features of GTO thyristors with the strengths of IGBT [1, 15, 20, 38, 51]. By merging the two, an IGCT based VSC is capable of providing a wide voltage range. Furthermore, several studies have shown that IGCT based DVRs have a high reliability, a Mean Time Between Failures (MTBF) of greater than twenty years [15]. The IGCT based VSC is very promising. It could be another possible contender if its physical size gets smaller in the future.

The final future work idea is rather grand, but it could revolutionize power electronics if the costs of semiconductor development ever come down in the future. Both IGBT type VSCs and FPGAs are silicon based devices. They have the same

substrate, so it is not so far fetched to integrate both on the same substrate. In the past, before FPGAs became so versatile, Application Specific Integrated Circuits (ASICs) dominated the landscape for processing on a chip. A trend from this time was to develop Power Integrated Circuits (PICs). PICs are chips that have both the control logic and the switch integrated into one chip [46]. The problem with this idea and ASICs is that the design is locked in. Unlike an FPGA, the design could not be reconfigured, ever. The final future work idea is to combine the VSC with the FPGA on one chip to form a Power FPGA. Perhaps it could be known as PFPGA.

APPENDIX A

ACRONYM LIST

1) AC - Alternating Current
2) A/D - Analog/Digital
3) APF - Active Power Filters
4) ASICs - Application Specific Integrated Circuits
5) BESS - Battery Energy Storage Systems
6) CBEMA - Computer Business Equipment Manufacturers Association
7) CLB - Configuration Logic Block
8) CODEC - COder DECoder
9) CORDIC - COrdinate Rotation DIgital Computer
10) COTS - Commercial Off The Shelf
11) CPLD - Complex Programmable Logic Device
12) CUPS - CUstom Power Supply
13) DC - Direct Current
14) DSC - Distribution Series Capacitors
15) DSP - Digital Signal Processing
16) DSTATCOM - Distribution STATic synchronous COMpensators
17) DUT - Design Under Test
18) DVR - Dynamic Voltage Restorer
19) FFT Fast Fourier Transform
20) FIR - Finite Impulse Response
21) FPGA - Field Programmable Gate Array
22) FT - Fourier Transform
23) FFT - Fast Fourier Transform
24) GTO - Gate Turn-Off
25) HDL - Hardware Description Language
26) ICs - Integrated Circuits
27) IGBT - Insulated Gate Bipolar Transistors
28) IGCT - Integrated Gate Commutated Thyristor
29) I/O - Input/Output
30) ISE - Integrated System Environment
31) ITIC - Information Technology Industry Council
32) JTAG - Joint Test Action Group
33) KSPS - Kilo Samples Per Second
34) LEDs - Light Emitting Diodes
35) LPW - Linear Phase Wavelet
36) LUTs - Look-Up Tables
37) MOSFET - Metal Oxide Semiconductor Field Effect Transistor
38) MSB - Most Significant Bit
39) MTBF - Mean Time Between Failures
40) PC - Personal Computer
41) PFC - Power Factor Correction
42) PFPGA - Power FPGA
43) PICs - Power Integrated Circuits

44) PLL	-	Phase-Locked Loop
45) PU	-	Per Unit
46) RAM	-	Random Access Memory
47) RMS	-	Root Mean Square
48) SA	-	Surge Arresters
49) SETC	-	Static Electronic Tap Changers
50) SMES	-	Super conducting Magnetic Energy Systems
51) SPS	-	Samples Per Second
52) SSFCL	-	Solid State Fault Current Limiter
53) SSTS	-	Solid-State Transfer Switches
54) SVC	-	Static Var Compensator
55) THD	-	Total Harmonic Distortion
56) TSC	-	Thyristor Switched Capacitors
57) UPS	-	Uninterruptible Power Supplies
58) V	-	Volt
59) VAC	-	Volt Alternating Current
60) VHDL	-	VHSIC Hardware Description Language
61) VHSIC	-	Very High Speed Integrate Circuits
62) VSC	-	Voltage Source Converter
63) WT	-	Wavelet Transform

APPENDIX B

VHDL CODE

```
----------------------------------------------------------------------------|
----------------------------------------------------------------------------|
--
-- Owner:        Y. Z. Kuo
-- Filename:     four_point_RMS.vhd
-- File Release:  Revision: NC
-- Last Modified: Date: 06/22/2005
--
----------------------------------------------------------------------------|
--
--   ***************************************************************
--   *        Classification: Unclassified/Restricted Rights       *
--   *        Copyright Y. Z. Kuo (C) 2003                         *
--   *        ALL RIGHTS RESERVED/UNPUBLISHED WORK                 *
--   *                                                             *
--   * This software shall not be reproduced without the expressed  *
--   * written permission of Y. Z. Kuo.                            *
--   ***************************************************************
--
--  PROJECT:          Dynamic Voltage Restorer (DVR)
--
----------------------------------------------------------------------------|
----------------------------------------------------------------------------|
--
-- CREATED    :   06/22/2005
-- LANGUAGE   :   VHDL
--
-- CONTENTS   :   VHDL Entity and Architecture
-- AUTHOR     :   Y.Z. Kuo (7453)
--
----------------------------------------------------------------------------|
-- THEORY OF OPERATION:
--
-- Four-Point RMS Averager
--
-- Computes a four-point RMS average from the FIR filters prior to subsequent
-- processing. After the input is squared by a multipler core, an FSM creates
-- the control signals necessary to coordinate the operation of four
-- accumulator cores.  The multiplier output is accepted and fed in overlap
-- fashion to the four accumulators.  Each accumulation sum requires four
-- samples and four accumulators are overlapped to accomplish this.  The
-- "MUX_SEL" signal is used to route accumulator outputs to the CORDIC square
-- root core as they become ready.  Bit truncation is used to divide the
-- sums by four prior to the square root.
--
----------------------------------------------------------------------------|
-- LIBRARY REFERENCES:
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

--  Uncomment the following lines to use the declarations that are
--  provided for instantiating Xilinx primitive components.

--library UNISIM;
--use UNISIM.VComponents.all;

----------------------------------------------------------------------------|
-- ENTITY

entity four_point_RMS is
        Port (
        CLK                                     : in        std_logic;
        RESET                                   : in                std_logic;
        INPUT_RDY           : in                std_logic;
```

```
        INPUT_DATA      : in    std_logic_vector(15 downto 0);
        AVERAGE_RDY     : out   std_logic;
        AVERAGE_DOUT    : out   std_logic_vector(16 downto 0)
        );

end four_point_RMS;
--------------------------------------------------------------------------------|
-- ARCHITECTURE

architecture rtl of four_point_RMS is
--------------------------------------------------------------------------------|

-- COMPONENT DECLARATIONS
--------------------------------------------------------------------------------|

        -- Xilinx CORE : Multiplier
        component mult
                port (
                clk                     : in    std_logic;
                a                               : in    std_logic_vector(15 downto 0);
                b                               : in    std_logic_vector(15 downto 0);
                q                               : out std_logic_vector(32 downto 0);
                aclr                    : in    std_logic;
                nd                              : in    std_logic;
                rdy                     : out std_logic
                );
        end component;

--------------------------------------------------------------------------------|

        -- Xilinx CORE : Accumulator
        component accum
                port (
                CLK                     : in    std_logic;
                CE                              : in    std_logic;
                B                               : in    std_logic_vector(32 downto 0);
                Q                               : out std_logic_vector(34 downto 0);
                AINIT                   : in    std_logic
                );
        end component;

--------------------------------------------------------------------------------|

        -- Xilinx CORE : CORDIC Square Root
        component sqrt
                port (
                x_in                    : in    std_logic_vector(32 downto 0);
                nd                              : in    std_logic;
                x_out                   : out std_logic_vector(16 downto 0);
                rdy                     : out std_logic;
                clk                     : in    std_logic;
                aclr                    : in    std_logic
                );
        end component;

--------------------------------------------------------------------------------|

        -- Accumulator Control Finite State Machine
        component accum_control
                port (
                CLK                             : in    std_logic;
                RESET                           : in            std_logic;
                MULT_RDY                : in            std_logic;

                CORDIC_ND               : out           std_logic;
                ACCUM_CE_0              : out           std_logic;
```

```
                ACCUM_CE_1              : out           std_logic;
                ACCUM_CE_2              : out           std_logic;
                ACCUM_CE_3              : out           std_logic;
                AINIT_0                 : out           std_logic;
                AINIT_1                 : out           std_logic;
                AINIT_2                 : out           std_logic;
                AINIT_3                 : out           std_logic;
                MUX_SEL                         : out           std_logic_vector(1 downto 0)
                );
        end component;

-------------------------------------------------------------------------------|

-- ATTIBUTE DECLARATIONS

        -- XST black box declaration
        attribute box_type : string;
        attribute box_type of mult: component is "black_box";
        attribute box_type of accum: component is "black_box";
        attribute box_type of sqrt: component is "black_box";

--      -- FPGA Express Black Box declaration

--      attribute fpga_dont_touch: string;
--      attribute fpga_dont_touch of mult: component is "true";
--      attribute fpga_dont_touch of accum: component is "true";
--      attribute fpga_dont_touch of sqrt: component is "true";
--
--      -- Synplicity black box declaration
--      attribute syn_black_box : boolean;
--      attribute syn_black_box of mult: component is true;
--      attribute syn_black_box of accum: component is true;
--      attribute syn_black_box of sqrt: component is true;
--
-------------------------------------------------------------------------------|

-- SIGNAL DECLARATIONS

        signal sMULT_RDY                        : std_logic;
        signal sMULT_OUT                        : std_logic_vector(32 downto 0);
        signal sACCUM_CE_0                      :       std_logic;
        signal sACCUM_CE_1                      :       std_logic;
        signal sACCUM_CE_2                      :       std_logic;
        signal sACCUM_CE_3                      :       std_logic;
        signal sACCUM_IN_0                      : std_logic_vector(32 downto 0);
        signal sACCUM_IN_1                      : std_logic_vector(32 downto 0);
        signal sACCUM_IN_2                      : std_logic_vector(32 downto 0);
        signal sACCUM_IN_3                      : std_logic_vector(32 downto 0);
        signal sACCUM_OUT_0                     : std_logic_vector(34 downto 0);
        signal sACCUM_OUT_1                     : std_logic_vector(34 downto 0);
        signal sACCUM_OUT_2                     : std_logic_vector(34 downto 0);
        signal sACCUM_OUT_3                     : std_logic_vector(34 downto 0);
        signal sAINIT_0                         : std_logic;
        signal sAINIT_1                         : std_logic;
        signal sAINIT_2                         : std_logic;
        signal sAINIT_3                         : std_logic;
        signal sMUX_SEL                         : std_logic_vector(1 downto 0);
        signal sCORDIC_IN                       : std_logic_vector(32 downto 0);
        signal sCORDIC_OUT                      : std_logic_vector(16 downto 0);
        signal sCORDIC_RDY                      : std_logic;
        signal sCORDIC_ND                       : std_logic;

  begin
-------------------------------------------------------------------------------|

-- CONCURRENT ASSIGNMENTS
```

```
            AVERAGE_DOUT(16 downto 0) <= sCORDIC_OUT;
AVERAGE_RDY <= sCORDIC_RDY;
sACCUM_IN_0 <= sMULT_OUT when (sACCUM_CE_0 = '1')
else (others => '0');
sACCUM_IN_1 <= sMULT_OUT when (sACCUM_CE_1 = '1')
else (others => '0');
sACCUM_IN_2 <= sMULT_OUT when (sACCUM_CE_2 = '1')
else (others => '0');

sACCUM_IN_3 <= sMULT_OUT when (sACCUM_CE_3 = '1')
else (others => '0');
sCORDIC_IN <=
            sACCUM_OUT_0(34 downto 2) when (sMUX_SEL = "00") else
sACCUM_OUT_1(34 downto 2) when (sMUX_SEL = "01") else
sACCUM_OUT_2(34 downto 2) when (sMUX_SEL = "10") else
sACCUM_OUT_3(34 downto 2) when (sMUX_SEL = "11") else
(others => '0');
-------------------------------------------------------------------------------|

-- COMPONENT INSTANTIATIONS
-------------------------------------------------------------------------------|

-- CORE mult instantiation
MULT_1 : mult
port map (
clk                             => CLK,
aclr                            => RESET,
a                                       => INPUT_DATA,
b                                       => INPUT_DATA,
q                                       => sMULT_OUT,
nd                              => INPUT_RDY,
rdy                             => sMULT_RDY);
-------------------------------------------------------------------------------|

-- CORE accum instantiation
ACCUM_0 : accum
port map (
      CLK                               => CLK,
CE                                      => sACCUM_CE_0,
B                                       => sACCUM_IN_0,
Q                                       => sACCUM_OUT_0,
AINIT                   => sAINIT_0);
-------------------------------------------------------------------------------|

-- CORE accum instantiation
ACCUM_1 : accum
port map (
CLK                             => CLK,
CE                                      => sACCUM_CE_0,
B                                       => sACCUM_IN_1,
Q                                       => sACCUM_OUT_1,
AINIT                   => sAINIT_1);
-------------------------------------------------------------------------------|

-- CORE accum instantiation
ACCUM_2 : accum
port map (
CLK                             => CLK,
CE                                      => sACCUM_CE_2,
B                                       => sACCUM_IN_2,
Q                                       => sACCUM_OUT_2,
AINIT                   => sAINIT_2);
-------------------------------------------------------------------------------|

-- CORE accum instantiation
ACCUM_3 : accum
```

```
port map (
CLK                     => CLK,
CE                              => sACCUM_CE_3,
B                               => sACCUM_IN_3,
Q                               => sACCUM_OUT_3,
AINIT           => sAINIT_3);
-------------------------------------------------------------------------------|

-- CORE sqrt instantiation
SQRT_1 : sqrt
port map (
clk                     => CLK,
aclr                    => RESET,
x_in                    => sCORDIC_IN,
nd                      => sCORDIC_ND,
x_out           => sCORDIC_OUT,
rdy                     => sCORDIC_RDY);
-------------------------------------------------------------------------------|

        accum_fsm_1 : accum_control
                port map (
                CLK                                     => CLK,
                RESET                                   => RESET,
                MULT_RDY                                => sMULT_RDY,
                CORDIC_ND               => sCORDIC_ND,
                ACCUM_CE_0              => sACCUM_CE_0,
                ACCUM_CE_1              => sACCUM_CE_1,
                ACCUM_CE_2              => sACCUM_CE_2,
                ACCUM_CE_3              => sACCUM_CE_3,
                AINIT_0                 => sAINIT_0,
                AINIT_1                 => sAINIT_1,
                AINIT_2                 => sAINIT_2,
                AINIT_3                 => sAINIT_3,
                MUX_SEL                                 => sMUX_SEL
                );

end rtl;
-------------------------------------------------------------------------------|
-- Revision History Log Footer:
--
-- 06/22/2005 - yzk - Original version.
--
--
```

APPENDIX C

IGBT SPECIFICATION

MGP15N40CL, MGB15N40CL

Preferred Device

Ignition IGBT
15 Amps, 410 Volts

N−Channel TO−220 and D^2PAK

This Logic Level Insulated Gate Bipolar Transistor (IGBT) features monolithic circuitry integrating ESD and Over−Voltage clamped protection for use in inductive coil drivers applications. Primary uses include Ignition, Direct Fuel Injection, or wherever high voltage and high current switching is required.

Features

- Ideal for Coil−On−Plug, IGBT−On−Coil, or Distributorless Ignition System Applications
- High Pulsed Current Capability up to 50 A
- Gate−Emitter ESD Protection
- Temperature Compensated Gate−Collector Voltage Clamp Limits Stress Applied to Load
- Integrated ESD Diode Protection
- Low Threshold Voltage to Interface Power Loads to Logic or Microprocessor Devices
- Low Saturation Voltage
- Optional Gate Resistor (R_G)
- Pb−Free Package is Available

MAXIMUM RATINGS ($-55°C \le T_J \le 175°C$ unless otherwise noted)

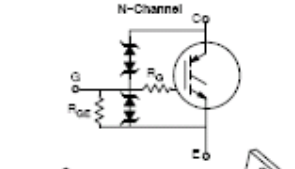
Rating	Symbol	Value	Unit
Collector−Emitter Voltage	V_{CES}	440	V_{DC}
Collector−Gate Voltage	V_{CER}	440	V_{DC}
Gate−Emitter Voltage	V_{GE}	22	V_{DC}
Collector Current−Continuous @ T_C = 25°C − Pulsed	I_C	15 50	A_{DC} A_{AC}
ESD (Human Body Model) R = 1500 Ω, C = 100 pF	ESD	8.0	kV
ESD (Machine Model) R = 0 Ω, C = 200 pF	ESD	500	V
Total Power Dissipation @ T_C = 25°C Derate above 25°C	P_D	150 1.0	W W/°C
Operating and Storage Temperature Range	T_J, T_{stg}	−55 to 175	°C

Maximum ratings are those values beyond which device damage can occur. Maximum ratings applied to the device are individual stress limit values (not normal operating conditions) and are not valid simultaneously. If these limits are exceeded, device functional operation is not implied, damage may occur and reliability may be affected.

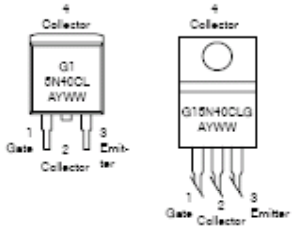
ON Semiconductor®

http://onsemi.com

15 AMPERES
410 VOLTS (Clamped)
$V_{CE(on)}$ @ 10 A = 1.8 V Max



MARKING DIAGRAMS & PIN ASSIGNMENTS



G15N40CL	= Device Code
A	= Assembly Location
Y	= Year
WW	= Work Week
G	= Pb−Free Package

ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 5 of this data sheet.

Preferred devices are recommended choices for future use and best overall value.

December, 2006 − Rev. 8

Publication Order Number:
MGP15N40CL/D

UNCLAMPED COLLECTOR−TO−EMITTER AVALANCHE CHARACTERISTICS (−55°C ≤ T_J ≤ 175°C)

Characteristic	Symbol	Value	Unit
Single Pulse Collector−to−Emitter Avalanche Energy V_{CC} = 50 V, V_{GE} = 5.0 V, Pk I_L = 17.4 A, L = 2.0 mH, Starting T_J = 25°C V_{CC} = 50 V, V_{GE} = 5.0 V, Pk I_L = 14.2 A, L = 2.0 mH, Starting T_J = 150°C	E_{AS}	300 200	mJ
Reverse Avalanche Energy V_{CC} = 100 V, V_{GE} = 20 V, L = 3.0 mH, Pk I_L = 25.8 A, Starting T_J = 25°C	$E_{AS(R)}$	1000	mJ

THERMAL CHARACTERISTICS

Characteristic		Symbol	Value	Unit
Thermal Resistance, Junction−to−Case		$R_{\theta JC}$	1.0	°C/W
Thermal Resistance, Junction−to−Ambient	TO−220 D²PAK (Note 1)	$R_{\theta JA}$ $R_{\theta JA}$	62.5 50	
Maximum Lead Temperature for Soldering Purposes, 1/8" from case for 5 seconds		T_L	275	°C

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	Test Conditions	Temperature	Min	Typ	Max	Unit
OFF CHARACTERISTICS							
Collector−Emitter Clamp Voltage	BV_{CES}	I_C = 2.0 mA	T_J = −40°C to 150°C	360	410	440	V_{DC}
		I_C = 10 mA	T_J = −40°C to 150°C	390	420	450	
Zero Gate Voltage Collector Current	I_{CES}	V_{CE} = 350 V, V_{GE} = 0 V	T_J = 25°C	–	1.5	20	μA_{DC}
			T_J = 150°C	–	10	40*	
			T_J = −40°C	–	0.7	1.5	
Reverse Collector−Emitter Leakage Current	I_{CES}	V_{CE} = −24 V	T_J = 25°C	–	0.85	1.0	mA
			T_J = 150°C	–	5.0	15*	
			T_J = −40°C	–	0.05	0.5	
Reverse Collector−Emitter Clamp Voltage	$BV_{CES(R)}$	I_C = −75 mA	T_J = 25°C	25	33	50	V_{DC}
			T_J = 150°C	25	36	50	
			T_J = −40°C	25	30	50	
Gate−Emitter Clamp Voltage	BV_{GES}	I_G = 5.0 mA	T_J = −40°C to 150°C	17	20	22	V_{DC}
Gate−Emitter Leakage Current	I_{GES}	V_{GE} = 10 V	T_J = −40°C to 150°C	364	600	1000	μA_{DC}
Gate Resistor (Optional)	R_G	–	T_J = −40°C to 150°C	–	70	–	Ω
Gate Emitter Resistor	R_{GE}	–	T_J = −40°C to 150°C	10	16	26	kΩ
ON CHARACTERISTICS (Note 2)							
Gate Threshold Voltage	$V_{GE(th)}$	I_C = 1.0 mA, V_{GE} = V_{CE}	T_J = 25°C	1.4	1.7	2.0	V_{DC}
			T_J = 150°C	0.75	1.1	1.4	
			T_J = −40°C	1.6	1.9	2.1*	
Threshold Temperature Coefficient (Neg)	–	–	–	–	4.4	–	mV/°C
Collector−to−Emitter On−Voltage	$V_{CE(on)}$	I_C = 6.0 A, V_{GE} = 4.0 V	T_J = 25°C	1.0	1.3	1.6	V_{DC}
			T_J = 150°C	0.9	1.2	1.5	
			T_J = −40°C	1.1	1.4	1.7*	
		I_C = 10 A, V_{GE} 4.0 V	T_J = 25°C	1.3	1.6	1.9	
			T_J = 150°C	1.2	1.5	1.8	
			T_J = −40°C	1.3	1.6	1.9*	
		I_C = 15 A, V_{GE} = 4.0 V	T_J = 25°C	1.6	1.95	2.25	
			T_J = 150°C	1.7	2.0	2.3*	
			T_J = −40°C	1.6	1.9	2.2	

1. When surface mounted to an FR4 board using the minimum recommended pad size.
2. Pulse Test: Pulse Width ≤ 300 μS, Duty Cycle ≤ 2%.
*Maximum Value of Characteristic across Temperature Range.

ELECTRICAL CHARACTERISTICS (continued)

Characteristic	Symbol	Test Conditions	Temperature	Min	Typ	Max	Unit
ON CHARACTERISTICS (continued) (Note 3)							
Collector−to−Emitter On−Voltage	$V_{CE(on)}$	I_C = 10 A, V_{GE} = 4.0 V	T_J = 25°C	1.9	2.2	2.5	V_{DC}
			T_J = 150°C	2.1	2.4	2.7*	
			T_J = −40°C	1.65	2.15	2.45	
		I_C = 25 A, V_{GE} = 4.0 V	T_J = 25°C	2.1	2.5	2.9	
			T_J = 150°C	2.5	2.9	3.3*	
			T_J = −40°C	2.0	2.4	2.8	
Collector−to−Emitter On−Voltage	$V_{CE(on)}$	I_C = 10 A, V_{GE} = 4.5 V	T_J = 150°C	–	1.5	1.8	V_{DC}
Forward Transconductance	g_{fe}	V_{CE} = 5.0 V, I_C = 5.0 A	T_J = −40°C to 150°C	5.0	15	25	Mhos
DYNAMIC CHARACTERISTICS							
Input Capacitance	C_{ies}	V_{CC} = 25 V, V_{GE} = 0 V, f = 1.0 MHz	T_J = −40°C to 150°C	–	1000	1300	pF
Output Capacitance	C_{oes}			–	100	180	
Transfer Capacitance	C_{res}			–	5.0	8.0	
SWITCHING CHARACTERISTICS (Note 3)							
Turn−Off Delay Time (Inductive)	$t_{d(off)}$	V_{CC} = 300 V, I_C = 5.0 A, R_G = 1.0 kΩ, L = 300 μH	T_J = 25°C	–	4.0	10	μSec
			T_J = 150°C	–	4.5	10	
Fall Time (Inductive)	t_f	V_{CC} = 300 V, I_C = 5.0 A, R_G = 1.0 kΩ, L = 300 μH	T_J = 25°C	–	7.0	10	
			T_J = 150°C	–	10	15*	
Turn−Off Delay Time (Resistive)	$t_{d(off)}$	V_{CC} = 300 V, I_C = 5.0 A, R_G = 1.0 kΩ, R_L = 46Ω,	T_J = 25°C	–	4.0	10	μSec
			T_J = 150°C	–	4.5	10	
Fall Time (Resistive)	t_f	V_{CC} = 300 V, I_C = 5.0 A, R_G = 1.0 kΩ, R_L = 46Ω,	T_J = 25°C	–	18	20	
			T_J = 150°C	–	16	20	
Turn−On Delay Time	$t_{d(on)}$	V_{CC} = 10 V, I_C = 5.0 A, R_G = 1.0 kΩ, R_L = 1.0 Ω	T_J = 25°C	–	1.0	1.5	μSec
			T_J = 150°C	–	1.0	1.5	
Rise Time	t_r	V_{CC} = 10 V, I_C = 5.0 A, R_G = 1.0 kΩ, R_L = 1.0 Ω	T_J = 25°C	–	4.5	6.0	
			T_J = 150°C	–	5.0	6.0	

3. Pulse Test: Pulse Width ≤ 300 μS, Duty Cycle ≤ 2%.
*Maximum Value of Characteristic across Temperature Range.

TYPICAL ELECTRICAL CHARACTERISTICS (unless otherwise noted)

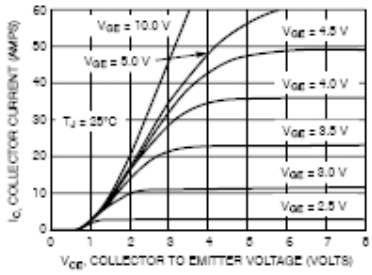


Figure 1. Output Characteristics

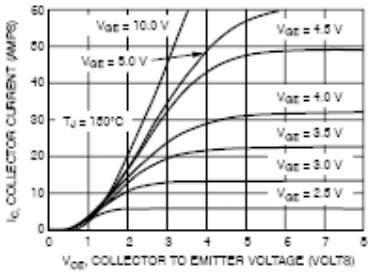


Figure 2. Output Characteristics

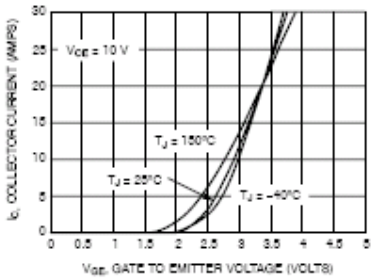


Figure 3. Transfer Characteristics

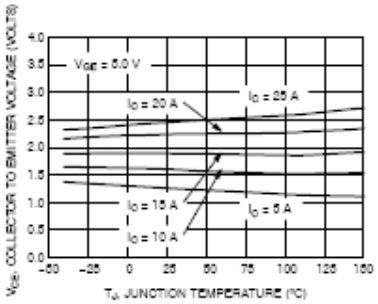


Figure 4. Collector−to−Emitter Saturation Voltage vs. Junction Temperature

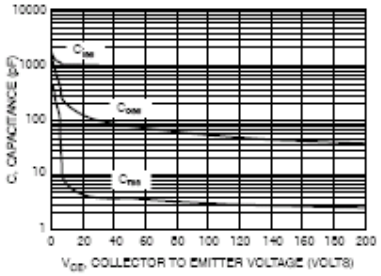


Figure 5. Capacitance Variation

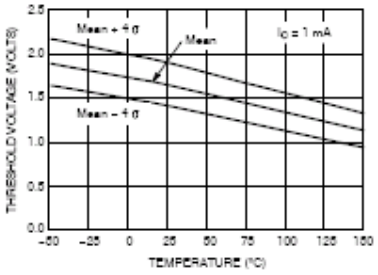


Figure 6. Threshold Voltage vs. Temperature

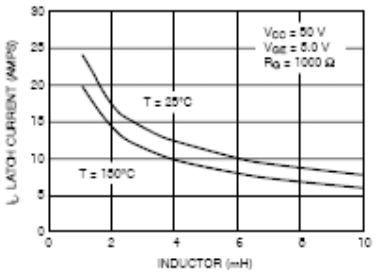


Figure 7. Minimum Open Secondary Latch Current vs. Inductor

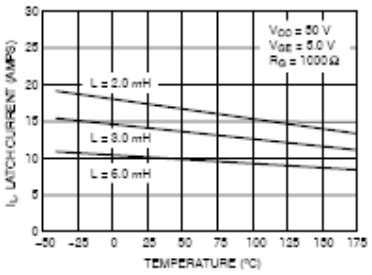


Figure 8. Minimum Open Secondary Latch Current vs. Temperature

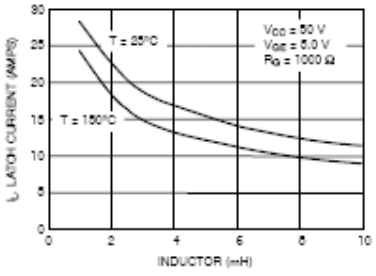


Figure 9. Typical Open Secondary Latch Current vs. Inductor

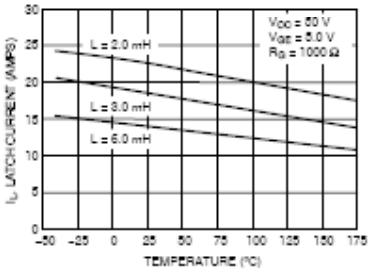


Figure 10. Typical Open Secondary Latch Current vs. Temperature

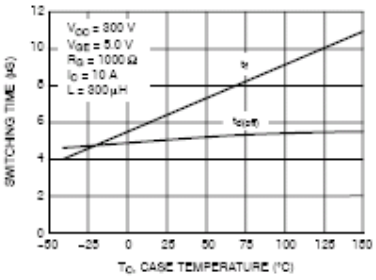


Figure 11. Switching Speed vs. Case Temperature

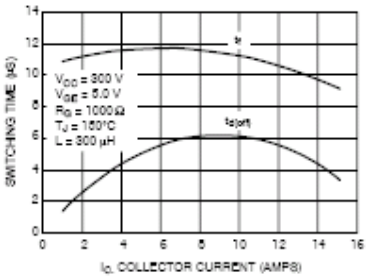


Figure 12. Switching Speed vs. Collector Current

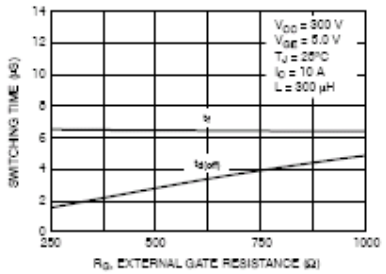


Figure 13. Switching Speed vs. External Gate Resistance

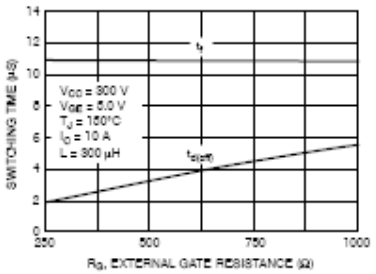


Figure 14. Switching Speed vs. External Gate Resistance

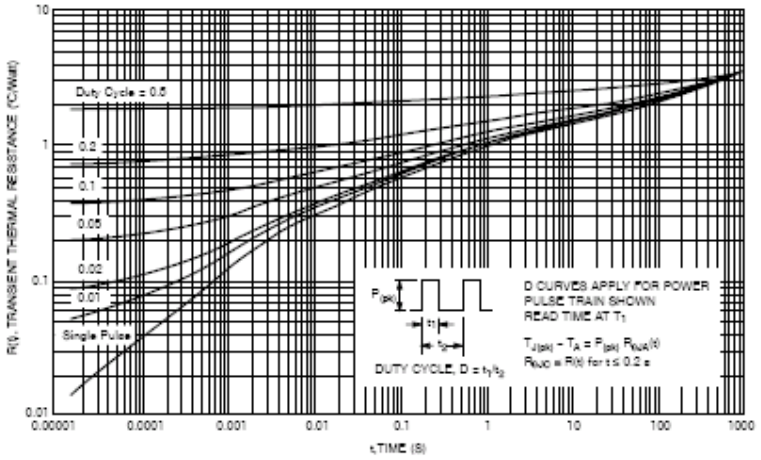


Figure 15. Transient Thermal Resistance (Non−normalized Junction−to−Ambient mounted on fixture in Figure 16)

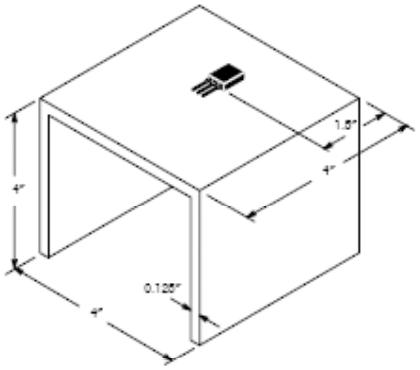


Figure 16. Test Fixture for Transient Thermal Curve
(48 square inches of 1/8" thick aluminum)

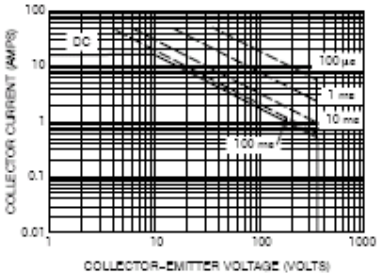


Figure 17. Single Pulse Safe Operating Area
(Mounted on an Infinite Heatsink at T_C = 25°C)

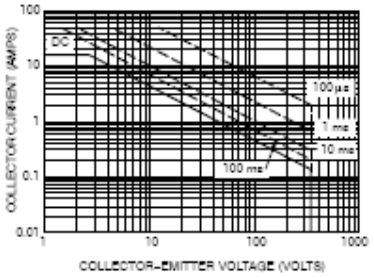


Figure 18. Single Pulse Safe Operating Area
(Mounted on an Infinite Heatsink at T_C = 125°C)

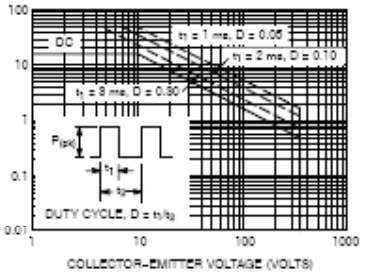


Figure 19. Pulse Train Safe Operating Area (Mounted on an Infinite Heatsink at T_C = 25°C)

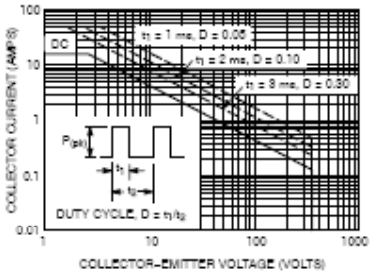


Figure 20. Pulse Train Safe Operating Area (Mounted on an Infinite Heatsink at T_C = 125°C)

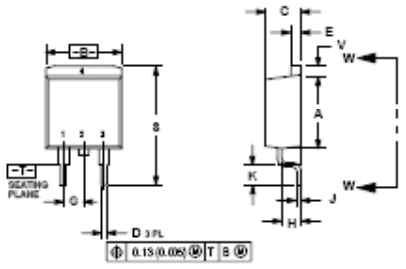
ORDERING INFORMATION

Device	Package	Shipping†
MGP15N40CL	TO−220AB	50 Units / Rail
MGP15N40CLG	TO−220AB (Pb−Free)	
MGB15N40CLT4	D2PAK	800 / Tape & Reel

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

PACKAGE DIMENSIONS

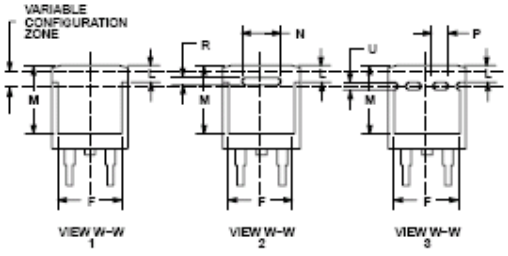
D²PAK 3
CASE 418B−04
ISSUE J



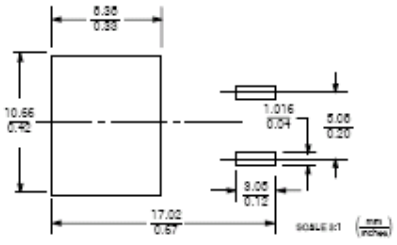
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. 418B−01 THRU 418B−03 OBSOLETE, NEW STANDARD 418B−04.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.340	0.380	8.64	9.65
B	0.380	0.405	9.65	10.29
C	0.160	0.190	4.06	4.83
D	0.020	0.035	0.51	0.89
E	0.045	0.055	1.14	1.40
F	0.310	0.350	7.87	8.89
G	0.100 BSC		2.54 BSC	
H	0.080	0.110	2.03	2.79
J	0.018	0.025	0.46	0.64
K	0.090	0.110	2.29	2.79
L	0.052	0.072	1.32	1.83
M	0.280	0.320	7.11	8.13
N	0.197 REF		5.00 REF	
P	0.079 REF		2.00 REF	
R	0.039 REF		1.00 REF	
S	0.575	0.625	14.60	15.88
V	0.045	0.055	1.14	1.40

STYLE 4:
PIN 1. GATE
2. COLLECTOR
3. EMITTER
4. COLLECTOR



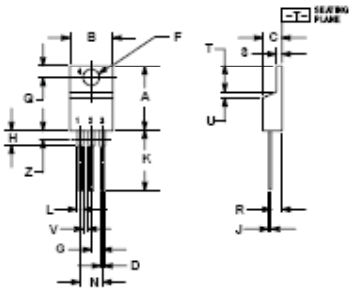
SOLDERING FOOTPRINT*



*For additional information on our Pb−Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

PACKAGE DIMENSIONS

TO−220 THREE−LEAD
TO−220AB
CASE 221A−09
ISSUE AA



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION Z DEFINES A ZONE WHERE ALL BODY AND LEAD IRREGULARITIES ARE ALLOWED.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.570	0.620	14.48	15.75
B	0.380	0.405	9.66	10.28
C	0.160	0.190	4.07	4.82
D	0.025	0.035	0.64	0.88
F	0.142	0.147	3.61	3.73
G	0.095	0.105	2.42	2.66
H	0.110	0.155	2.80	3.93
J	0.018	0.025	0.46	0.64
K	0.500	0.562	12.70	14.27
L	0.045	0.060	1.15	1.52
N	0.190	0.210	4.83	5.33
Q	0.100	0.120	2.54	3.04
R	0.080	0.110	2.04	2.79
S	0.045	0.055	1.15	1.39
T	0.235	0.255	5.97	6.47
U	0.000	0.050	0.00	1.27
V	0.045	---	1.15	---
Z	---	0.080	---	2.04

STYLE 5:
PIN 1. GATE
2. COLLECTOR
3. EMITTER
4. COLLECTOR

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 61312, Phoenix, Arizona 85082−1312 USA
Phone: 480−829−7710 or 800−344−3860 Toll Free USA/Canada
Fax: 480−829−7709 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada

Japan: ON Semiconductor, Japan Customer Focus Center
2−9−1 Kamimeguro, Meguro−ku, Tokyo, Japan 153−0051
Phone: 81−3−5773−3850

ON Semiconductor Website: http://onsemi.com

Order Literature: http://www.onsemi.com/forder

For additional information, please contact your local Sales Representative.

MGP16N40CL/D

REFERENCES

[1] R. Arnold, "Solutions to the power quality problem," *Power Engineering Journal*, vol. 15, pp. 65-73, April 2001.

[2] C. S. Chang, Y. S. Ho, and P. C. Loh, "Voltage quality enhancement with power electronics based devices," in *IEEE Power Engineering Society Winter Meeting*, 2000, pp. 2937-2942.

[3] Y. H. Chung, G. H. Kwon, T. B. Park, and K. Y. Lim, "Voltage sag and swell generator for the evaluation of custom power devices," in *IEEE Power Engineering Society General Meeting*, 2003, p. 2507.

[4] J. H. R. Enslin, "Unified approach to power quality mitigation," in *IEEE International Symposium on Industrial Electronics*, 1998, pp. 8-20.

[5] I. Etxeberria-Otadui, U. Viscarret, S. Bacha, M. Caballero, and R. Reyero, "Evaluation of different strategies for series voltage sag compensation," in *2002 IEEE 33rd Annual Power Electronics Specialists Conference*, 2002, pp. 1797-1802.

[6] H. Kim, "Minimal Energy Control for a Dynamic Voltage Restorer," in *Proceedings of the Power Conversion Conference*, Osaka, Japan, 2002, pp. 428-433.

[7] H. Kim, J. Kim, and S. Sul, "A design consideration of output filters for dynamic voltage restorers," in *2004 IEEE 35th Annual Power Electronics Specialists Conference*, 2004, pp. 4268-4272.

[8] A. Bendre, D. Divan, W. Kranz, and W. E. Brumsickle, "Are Voltage Sags Destroying Equipment?," *IEEE Industry Applications Magazine*, vol. 12, pp. 12-21, July-August 2006 2006.

[9] A. Bendre, D. Divan, W. Kranz, and W. E. Brumsickle, "Equipment Failures Caused By Power Quality Disturbances," in *2004 IEEE Industry Applications Conference*, 2004.

[10] ElectroIndustries, "PDA 1252 Portable Power Analyzer." vol. 2006, 2006.

[11] T. L. Grant and D. Divan, "Power Quality Solutions To Mitigate The Impact Of Voltage Sags In Manufacturing Facilities," 2006.

[12] NationalEnergyTechnologyLaboratory, "A Systems View of the Modern Grid," U.S. Department of Energy Office of Electricity Delivery and Energy Reliability Appendix A4.

[13] M. H. J. Bollen, *Understanding Power Quality Problems*. New York: IEEE Press, 2000.

[14] D. M. Vilathgamuwa, H. M. Wijekoon, and S. S. Choi, "Interline Dynamic Voltage Restorer: a Novel and Economical Approach for Multiline Power Quality Compensation," *IEEE Transactions on Industry Applications*, vol. 40, pp. 1678-1685, Nov.-Dec. 2004.

[15] K. Chan, "Technical and Performance Aspects of a Dynamic Voltage Restorer," in *IEE Half Day Colloquium on Dynamic Voltage Restorers - Replacing Those Missing Cycles*, 1998, pp. 5/1 -525.

[16] R. C. Dugan, M. F. McGranaghan, and H. W. Beaty, *Electrical Power Systems Quality*. New York: McGraw-Hill, 1996.

[17] C. Zhan, M. Barnes, V. K. Ramachandaramurthy, and N. Jenkins, "Dynamic voltage restorer with battery energy storage for voltage dip mitigation," in *Eighth International Conference on Power Electronics and Variable Speed Drives*, 2000, pp. 360-365.

[18] H. Sree and N. Mohan, "Voltage sag mitigation using a high-frequency-link cycloconverter-based DVR," in *26th Annual Conference of the IEEE Industrial Electronics Society*, 2000, pp. 344-349.

[19] I. T. I. C. (ITIC), "ITI (CBEMA) Curve Application Note," Washington DC, 2001.

[20] K. Chan, A. Kara, and D. Westermann, "Integrated gate commutated thyristor based dynamic voltage restorer," in *1998 International Conference on Power System Technology*, Beijing, China, 1998, pp. 635-638.

[21] N. G. Hingorani, "Power Electronics Applications In Energy Systems," in *NSF Faculty Workshop on Teaching of Courses in Power Electronics and Electric Drives*, Tempe, Arizona, 2002.

[22] R. McHattie, "Dynamic Voltage Restorer the Customers's Perspective," in *IEE Half Day Colloquium on Dynamic Voltage Restorer - Replacing Those Missing Cycles*, 1998, pp. 1/1-1/5.

[23] S. Chen, G. Joos, L. Lopes, and W. Guo, "A nonlinear control method of dynamic voltage restorers," in *2002 IEEE 33rd Annual Power Electronics Specialists Conference*, 2002, pp. 88-93.

[24] R. Buxton, "Protection from voltage dips with the dynamic voltage restorer," in *IEE Half Day Colloquium on Dynamic Voltage Restorers - Replacing Those Missing Cycles*, 1998, pp. 3/1-3/6.

[25] B. H. Li, S. S. Choi, and D. M. Vilathgamuwa, "Design considerations on the line-side filter used in the dynamic voltage restorer," *IEE Proceedings - Generation, Transmission, and Distribution*, vol. 148, pp. 1-7, Jan. 2001 2001.

[26] E. K. K. Sng, S. S. Choi, and D. M. Vilathgamuwa, "Analysis of Series Compensation and DC-Link Voltage Controls of a Transformerless Self-Charging Dynamic Voltage Restorer," *IEEE Transactions on Power Delivery*, vol. 19, pp. 1511-1518, July 2004.

[27] N. Kularatna, *Power Electronics Design Handbook: Low-Power Components and Applications*. Boston: Newnes, 1998.

[28] S. Lee, H. Kim, and S. Sul, "A novel control method for the compensation voltages in dynamic voltage restorers," in *Nineteenth Annual IEEE Applied Power Electronics Conference and Exposition*, 2004, pp. 614-620.
[29] L. Xu, O. Anaya-Lara, V. G. Agelidis, and E. Acha, "Development of prototype custom power devices for power quality enhancement," in *Ninth International Conference on Harmonics and Quality of Power*, 2000, pp. 775-783.
[30] C. P. Fitzer, Y. S. Lim, C. Zhan, V. K. Ramachandaramurthy, A. Arulampalam, M. Barnes, G. Strbac, and N. Jenkins, "Mitigation of voltage sags embracing a prediction technique and the use of a dynamic voltage restorer," in *16th International Conference and Exhibition on Electricity Distribution*, 2001, p. 5.
[31] V. K. Ramachandaramurthy, C. Fitzer, A. Arulampalam, C. Zhan, M. Barnes, and N. Jenkins, "Control of a battery supported dynamic voltage restorer," *IEE Proceedings - Generation, Transmission, and Distribution*, vol. 149, pp. 533-542, Sept. 2002.
[32] C. Fitzer, M. Barnes, and P. Green, "Voltage Sag Detection Technique for a Dynamic Voltage Restorer," *IEEE Transactions on Industry Applications*, vol. 40, pp. 203-212, Jan. - Feb. 2004.
[33] S. A. Saleh and M. A. Rahman, "Wavelet-based dynamic voltage restorer for power quality improvement," in *2004 IEEE 35th Annual Power Electronics Specialists Conference*, 2004, pp. 3152-3156.
[34] M. Fang, A. I. Gardiner, A. MacDougall, and G. A. Mathieson, "A novel series dynamic voltage restorer for distribution systems," in *1998 International Conference on Power System Technology*, 1998, pp. 38-42.
[35] M. H. Rashid, *Power Electronics Handbook*. San Diego: Academic Press, 2001.
[36] B. K. Bose, *Power Electronics and AC Drives*. Englewood Cliffs: Prentice-Hall, 1986.
[37] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*. New York: John Wiley & Sons, 1989.
[38] T. Wunderlin, O. Amhof, P. Dahler, and H. Guning, "Power supply quality improvement with a dynamic voltage restorer (DVR)," in *1998 International Conference on Energy Management and Power Delivery*, 1998, pp. 518-525.
[39] On Semiconductor, "Ignition IGBT 15 Amps, 410 Volts," in *MGP15N40CL, MGB15N40CL*, 2005.
[40] C. Yingvivatanapong, "Robust Algorithms For High Speed Voltage Sags And Swelling Detection," in *Electrical Engineering* Arlington: University of Texas at Arlington, 2001.
[41] C. Yingvivatanapong, W. J. Lee, S. Oraintara, and D. Johnson, "Robust algorithms for high speed voltage sags and swells detection," in *Industrial and Commercial Power Systems Technical Conference, 2002 IEEE*, 2002, pp. 131-137.
[42] W. Zhang, G. Feng, Y. F. Liu, and B. Wu, "DSP Implementation of Predictive Control Strategy for Power Factor Correction (PFC)," in *Applied Power Electronics Conference and Exposition, 2004*, 2004, pp. 67-73.

[43] Y. Cho and S. Sul, "Controller design for dynamic voltage restorer with harmonics compensation function," in *39th IAS Annual Meeting Industry Applications Conference*, 2004, pp. 1452-1457.
[44] Altera, "dsp vs fpga operation," in *gif*, dsp-fpga_fig2.gif, Ed., 2005, p. DSP serial operation versus FPGA parallel operation.
[45] M. J. Newman, D. G. Holmes, J. G. Nielsen, and F. Blaabjerg, "A dynamic voltage restorer (DVR) with selective harmonic compensation at medium voltage level," in *38th IAS Annual Meeting Industry Applications Conference*, 2003, pp. 1228-1235.
[46] B. K. Bose, *Modern Power Electronics and AC Drives*. Upper Saddle River: Prentice Hall PTR, 2002.
[47] On Semiconductor, "NCP1603." vol. 2005: Semiconductor Components Industries, LLC, 2005, p. PFC/PWM Combo Controller with Integrated High Voltage Startup and Standby Capability.
[48] Intel, "Moore's Law," in *gif*, MooresLawgraph3.gif, Ed., 2005, p. Moore's Law graph.
[49] P. J. Ashenden, *The Designer's Guide to VHDL*. San Francisco: Morgan Kaufmann Publishers, Inc., 1996.
[50] F. Jurado, M. Valverde, and J. Carpio, "Voltage Sag Correction by Dynamic Voltage Restorer Based on Fuzzy Logic Control," in *Canadian Conference on Electrical and Computer Engineering*, 2003, pp. 421-424.
[51] K. Chan and A. Kara, "Voltage sags mitigation with an integrated gate commutated thyristor based dynamic voltage restorer," in *8th International Conference on Harmonics And Quality of Power*, 1998, pp. 561-565.
[52] A. H. Nuttall, *Some Windows with Very Good Sidelobe Behavior*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.ASSP-29, No.1, February 1981, pp 84–91.
[53] F. J. Harris, *On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform*, Proceedings of the IEEE, Vol.66, No.1, January 1978, pp 51–83.

BIOGRAPHICAL INFORMATION

The author received her Bachelor of Science in Electrical Engineering in 1996 and her Master of Science in Electrical Engineering in 1999, both from the University of Texas at Arlington. She received her Doctorate in Electrical Engineering from the University of Texas at Arlington in May 2007. The author has been employed in engineering positions including ASIC design, digital design, FPGA design, systems engineering by Raytheon E-Systems, currently L-3 Communications, for over 10 years and has several patents pending.