12-2014

Design, Simulation and Implementation of Three-Phase Bidirectional DC-DC Dual Active Bridge Converter Using SiC MOSFETs

Tariq Aldawsari
University of Arkansas, Fayetteville

Design, Simulation and Implementation of Three-Phase Bidirectional DC-DC Dual Active Bridge Converter Using SiC MOSFETs

Design, Simulation and Implementation of Three-Phase Bidirectional DC-DC Dual Active Bridge Converter Using SiC MOSFETs

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

By

Tariq Aldawsari
University of Arkansas
Bachelor of Science in Electrical Engineering 2011

December 2014
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

____

Dr. Roy A. McCann
Thesis Director

____________________________ ____________________________

Dr. Simon Ang
Committee Member

Dr. Hameed Naseem
Committee Member

ABSTRACT

The use of SiC-based martials in fabricating power semiconductor devices has shown more interest than conventional silicon-based. Its promising abilities to improve the performance of power electronic systems made it a valuable choice in building high power DC-DC converters. This thesis presents the design and implementation of a three-phase bidirectional DC-DC Dual Active Bridge using SiC MOSFETs. The proposed circuit is first built in Matlab for simulation analysis. Then a phase shift modulation controller is designed in Simulink to test the simulation circuit. The controls are then integrated through an FPGA to test the prototype. Simulations and experimental results are evaluated to demonstrate the functionality and performance of the proposed circuit.

ACKNOWLEDGEMENTS

First I would like to thank my advisor Dr. Roy McCann for his support and guidance throughout the master program. Also, I would like to thank the thesis committee for being a part of this achievement. Second, I would also like to thank my family and friends for their continued support and encouragement throughout my graduate study. Special thanks to my father and mother for their help, support and guidance in becoming who I am today. Another special thanks for my brother Khaled for being there whenever in need.

Tariq Khalaf Aldawsari

TABLE OF CONTENT

CHAPTER 1

Introduction

1. Background

1.1.1. History of Converters

Switching converters may have been introduced to the market in the 1950s, but their applications were limited due to the high costs of power switching transistors at the time. Starting in the 1970s, semiconductor devices such as MOSFETs (Metal-Oxide-Semiconductor-Field-Effect Transistor) and IGBTs (Insulated Gate Bipolar Transistor) have become more available and reliable. This led the switching converters to become more prevalent in power applications [1]. The basic circuit of a typical bidirectional dc-dc converter will include a capacitor, inductor, diode and a switching transistor which allow the power to flow in both directions. The order these parts are placed in the circuit makes a topology. However, most of dc-dc converters can be derived from buck or boost converter which are the simplest topologies of a bidirectional converters [1].

1.1.2. State-of- the-Art Bidirectional DC-DC Converters

The terminology bidirectional emphasizes that there are two methods of operation that these converters go through considering the difference of voltage amplitude on each side of the converter. To clarify, Fig. 1 shows the basic mode operation of all bidirectional dc-dc converters.

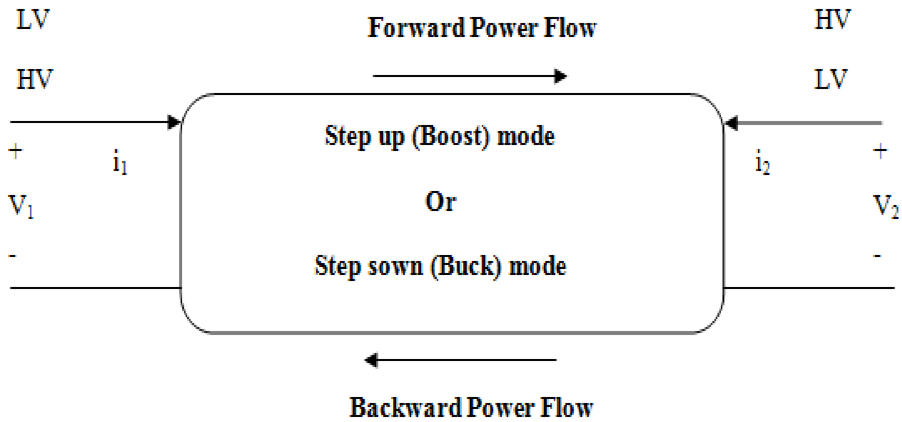

Fig. 1.1. Basic structure of bidirectional dc-dc converter.

The first mode of operation is called boost mode or step up mode where a low voltage is fed on the low voltage side (LV) and then stepped up based on the ratio of the conversion to a higher voltage on the high voltage side (HV). The second mode of operation is called a buck mode or step down mode where a high voltage amplitude is stepped down to match an amplitude of low voltage application. The converter has a forward power flow or backward power flow based on the current conditions as follows:

- Forward power flow $i_1<0$, $i_2>0$
- Backward power flow $i_1>0$, $i_2<0$

1.1.3. Non-isolated Bidirectional DC-DC Converters

The dc-dc converters have shown how they can be advantageous for a variety of reasons in a variety of applications compare to other converters. With all the different topologies discovered, dc-dc converters are categorized into two types, non-isolated and isolated converter [2-23].

In the Non-isolated bidirectional dc-dc converters, the input and the output usually have a common ground unlike the isolated converters in which these two are electrically separated. Buck converter, boost converter, buck-boost converter, Cúk converter, and full-bridge converter are the five topologies that are common non-isolated converter. But only the buck and the boost converter are considered to be the basic topologies. The full-bridge is derived from the buck converter whereas both the Cúk and buck-boost converters are a combination of the buck and boost converters [24]. These converters are sometime used as unidirectional converters either to step up or step down the voltage. This is done by replacing the controllable switches on the configuration to diodes [25].

1.1.4. Isolated Bidirectional DC-DC Converters

Isolation is usually provided by using a high frequency transformer where the input and output of the converter are electrically separated. Having isolation will assist in noise reduction, help in personnel safety, and provides protection to the system due to galvanic isolation [25]. Topologies of the isolated dc-dc converter are being investigated and new ones are proposed based on old topologies structure. These topologies are paired into groups based on the operational aspect. However, there are two basic topologies that most of the isolated families fall into, voltage source converter and a current source converter which are tied together by a high frequency transformer. As shown in Fig. 2 the voltage source is paired with a current source to form a bidirectional flow to allow smooth power transfer. For instance, when having a voltage source on the LV, a current source converter should be placed on the HV and vice versa. The HV side or the LV side can use either an inverter or a rectifier depends on the mode of operation [26].

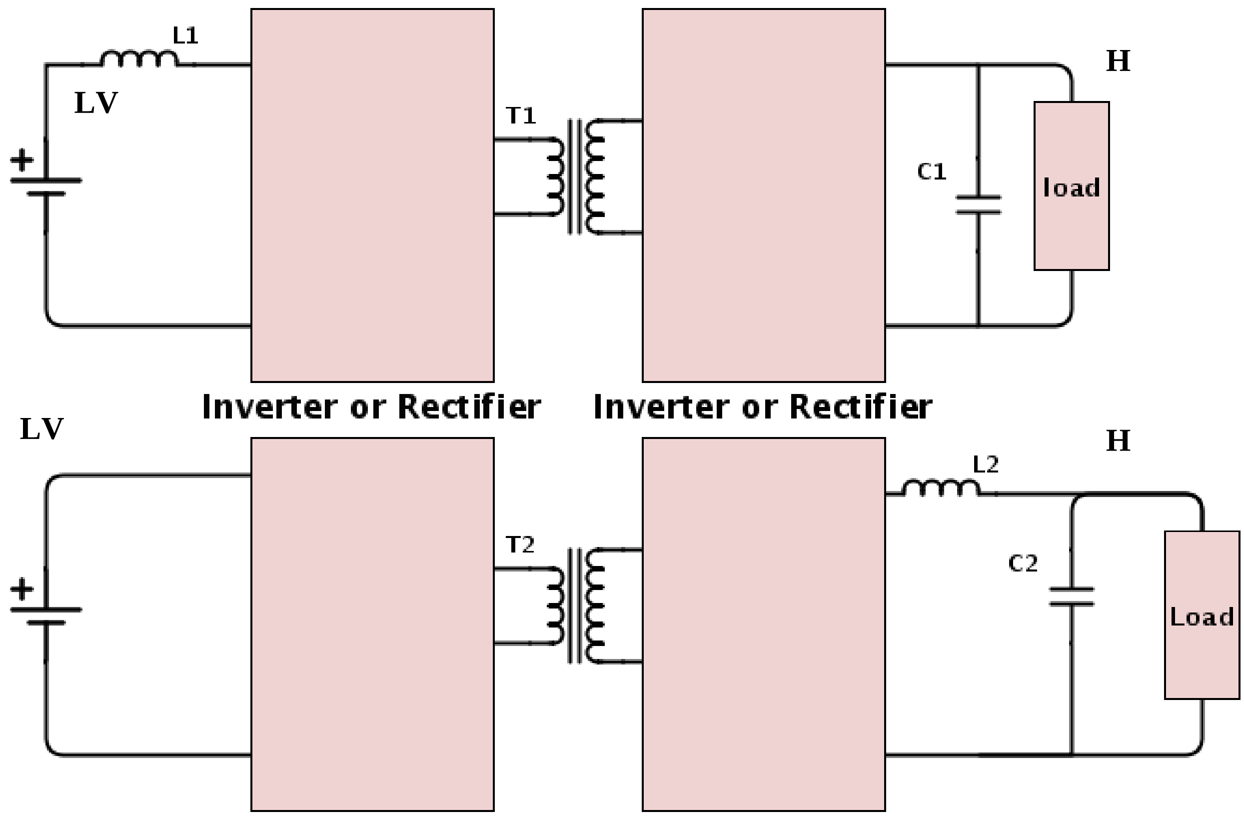

Fig. 1.2. The two basic configurations of isolated bidirectional dc-dc converter.

Each inverter or rectifier block can be in a form of voltage source or a current source converter. There are three basic structures that make a voltage source or a current source converter. These are the full-bridge, half-bridge and push-pull structure. The basic three topologies of the current source can be achieved by replacing the parallel capacitor to the dc bus in a voltage source structure with an inductor that placed in series with the dc bus. In Fig. 3 is shown the three basic voltage source converters where (a)full bridge, (b) half-bridge and (c) push-pull whereas Fig. 4 shows the same structure but in a current source mode [26].

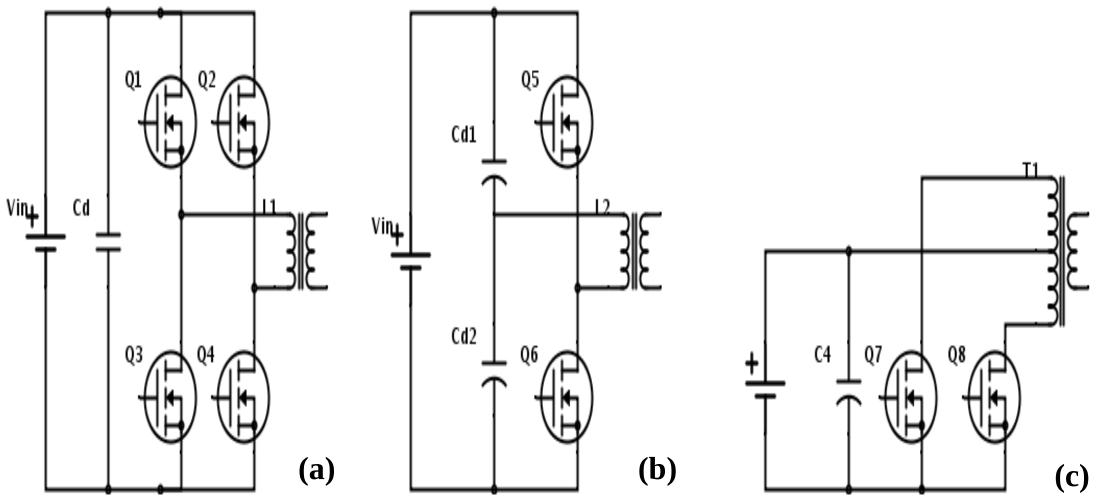

Fig. 1.3. Voltage source converters (a) Full-bridge, (b) Half-bridge, (c) Push-pull.

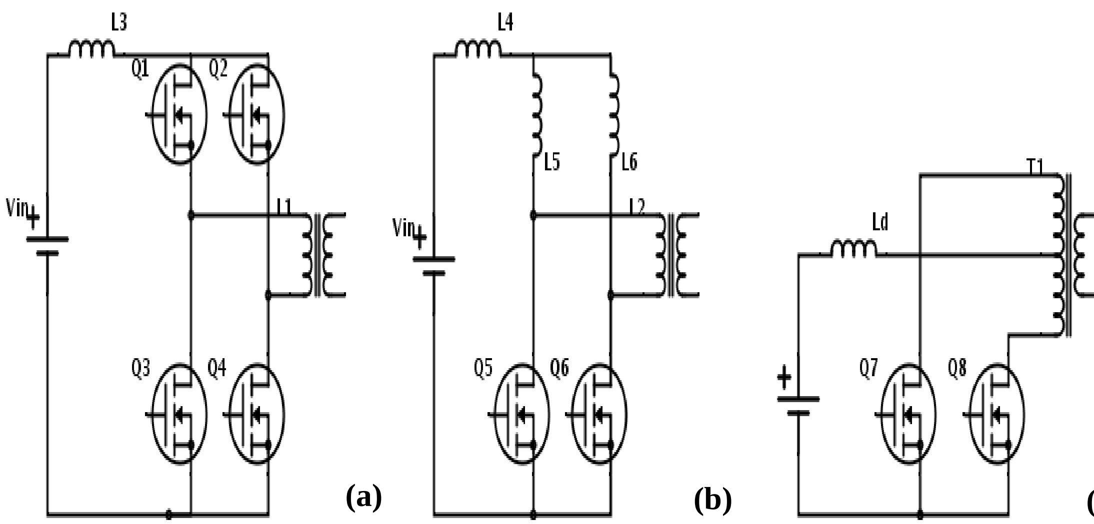

Fig. 1.4. Current source converters (a) Full-bridge, (b) Half-bridge, (c) Push-pull.

1.2. Dual Active Bridge Converters

The Dual Active Bridge (DAB) converter family is an isolated bidirectional dc-dc converter that consists of two inverters, single or three-phase, which are tied together by a high frequency transformer. Their structure could consist of either half-bridge or full-bridge topology and usually is a symmetrical configuration. Having a symmetrical structure enables the DAB transfer power smoother than other isolated dc-dc converters. The DAB family has attractive features which make them highly suitable for high power applications. Bidirectional power flow, high power density, isolation, and low component stress when zero-voltage switching are some of these features [27], [28], [29]. These structures also perform at high frequencies which decrease the harmonic content; leading to less power quality issues. Using one power converter to support a bidirectional power flow would be more preferable for many applications than two converters (one for each direction). Using one power converter enables the systems to be smaller in size, lower in weight and more cost effective [30].

1.2.1. Single-phase Dual Active Bridge

The single-phase DAB was first introduced in the 1980s. The topology consists of two inverters connected together by a transformer. The inverters could be in a form of half-bridge or full-bridge topology as shown in Fig. 5. The working operation of this structure is simple. The input voltage is converted into a high frequency square wave AC voltage in the first inverter which is then converted back by the second inverter into DC voltage after it passes through a transformer. The high frequency transformer not only provides galvanic isolation to the system but also is used as an energy storage component. The power flow is controlled by using a phase shift modulation. In each inverter the bridges are switched on at 50% duty cycle with the bridges legs phase shifted by 120 degrees. The inverters on each side of the transformer are also phase shifted to determine the direction of the power flow. The power flow depends on which bridge has leading or lagging power [31].

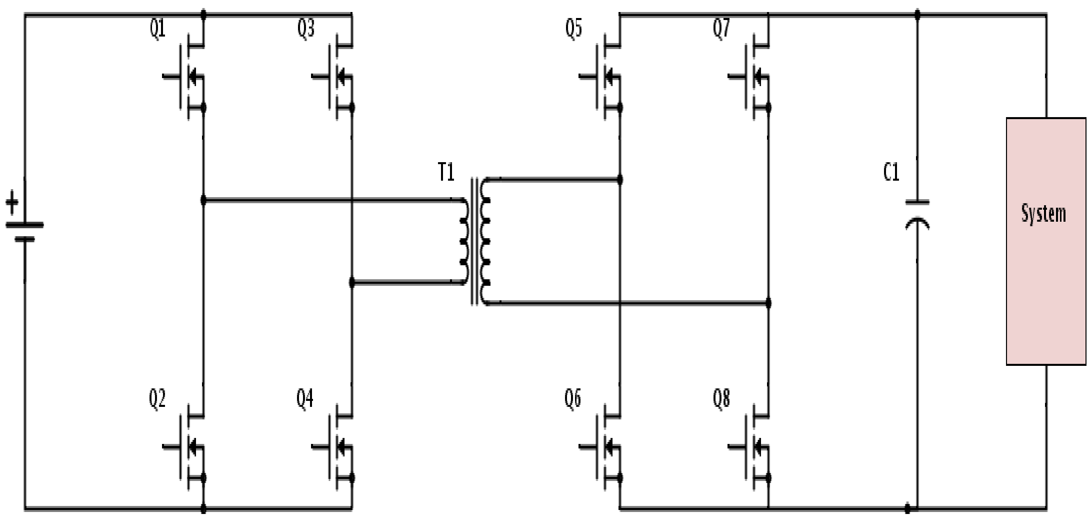

Fig. 1.5. Single-phase DAB full-bridge topology.

1.2.2. Three-phase Dual Active Bridge

Another topology of the DAB family is the three-phase structure. The three-phase DAB circuit consists of two three-phase inverters that are tied together by a three-phase transformer as shown in Fig. 6. Despite the fact that the single-phase is considered to be more dominant in research, the three-phase is poised to become more utilized. Unlike the single-phase, using three-phase transformer leads to better apparent power thus a higher power density is attainable [32]. Similar to the single phase, the upper and bottom switch in the three phase leg works at complementary 50% duty cycle. In each inverter, the legs are phase shifted by 120 degrees. Also, the inverters on each side of the transformer are phase shifted to control the direction and the amount of power flow [32].

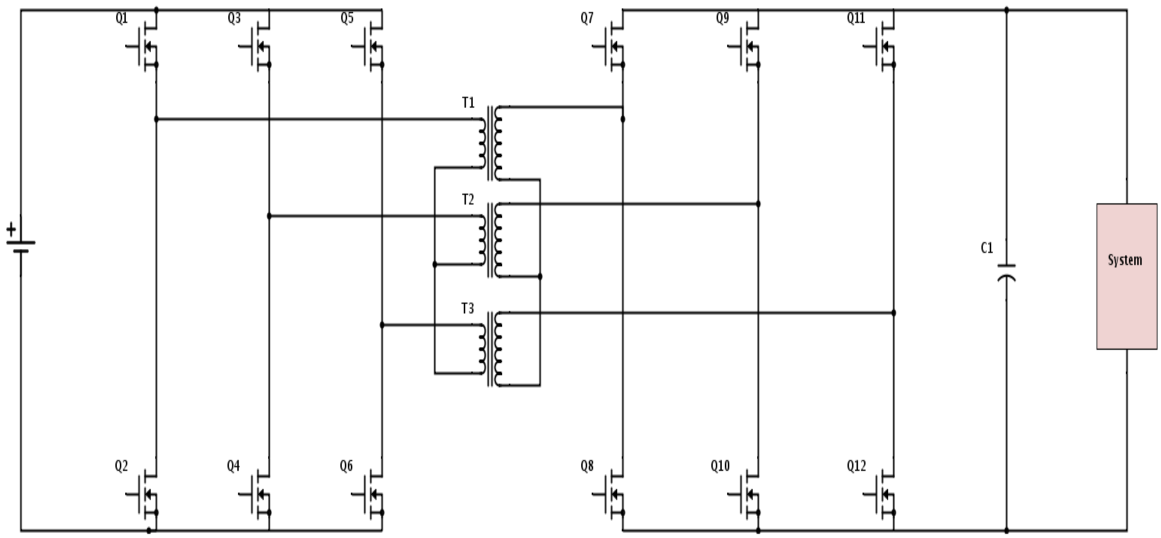

Fig. 1.6. Three-phase DAB topology.

1.2.3. Applications of bidirectional DC-DC converters

The use of DAB dc-dc converters has been increasing as the demand for bidirectional power flow with high efficiency is preferred in high voltage direct current (HVDC) transmission systems as well as battery application systems. Uninterruptible power supplies (UPS), battery management systems, renewably energy systems and auxiliary power supplies for hybrid electric vehicles and fuel cell vehicles are some of the applications that also use the DAB family to achieve high power density with high efficiency. For instance, energy management systems prefer the combination of bidirectional dc-dc convertor along with an energy storage due to its promising advantages. Having these two in one systems will not only improve the efficiency but will also have a huge impact on the size and the cost of the system [26].

In the hybrid electric vehicle (HEV), there are two suggested systems. One that works by only using an energy storage device and the second system that uses energy storage along with a bidirectional dc-dc converter as shown in Fig. 1.7. In both systems an electric generator is used to supply power to the motor drive and to charge the batteries. In the system where there is not a dc-dc converter used, a high voltage battery is needed to match the output of the generator and the rated voltage of the inverter that supplies the motor drive. In the other system a low voltage battery will do the job and it will only be used during startup and acceleration. The second system may require more parts but it is considered to be more efficient due to its capabilities. The same concept applies for fuel cell vehicles (FCV), an ultra-capacitor bank that matches the fuel cell stack voltage is used when the system lacks a bidirectional dc-dc converter whereas a low voltage battery is used when the dc-dc converter is present[26].

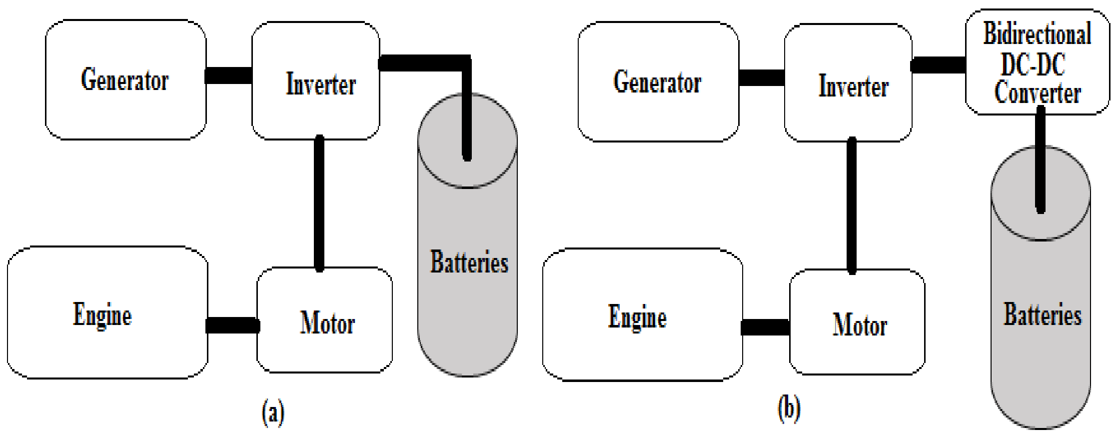

Fig. 1.7. HEV system (a) without dc-dc converter (b) with dc-dc converter.

UPS's are used when in need for backup to a system and to prevent loss of data. Many studies have shown that typical UPS uses an isolated ac-dc converter for battery charging and a dc-ac inverter to supply the grid. This process would require double conversion and thus lower efficiency. However, using a bidirectional dc-dc converter will enable the UPS to charge the batteries during normal mode and reverse power flow when the systems needs backup [33].

Renewable energy sources have become more popular even though they contribute a small share of energy production. Their usage is depended on their cost and availability. Since oil and natural gas prices have increased tremendously the usage of different energy sources became an attractive option to look into [34]. In any system, achieving high power density with high efficiency is the target of today's industry. However with renewable energy there is always the concern of power fluctuation due to nature's call. Thus, energy storage devices are used in these systems to allow a smoother power flow to the load and to reduce the fluctuation in the system [25]. In the presence of energy storage in a system, a bidirectional power flow and flexible controls are required and a good choice to accomplish that is by using a bidirectional dc-dc converters. In connecting AC systems to a renewable power source, the DAB family was considered the next generation's choice in having high efficiency as high as %99 [35]. Fig. 7 illustrate the structure of typical photovoltaic (PV) system. A bidirectional dc-dc converter is present in the system to ensure a stable bus voltage and to charge the battery when needed. The battery size may vary depends on the required power level. Having an energy storage connected to the grid would not only provide voltage support but also help in grid stabilization, load shifting, reliability enhancement [25].

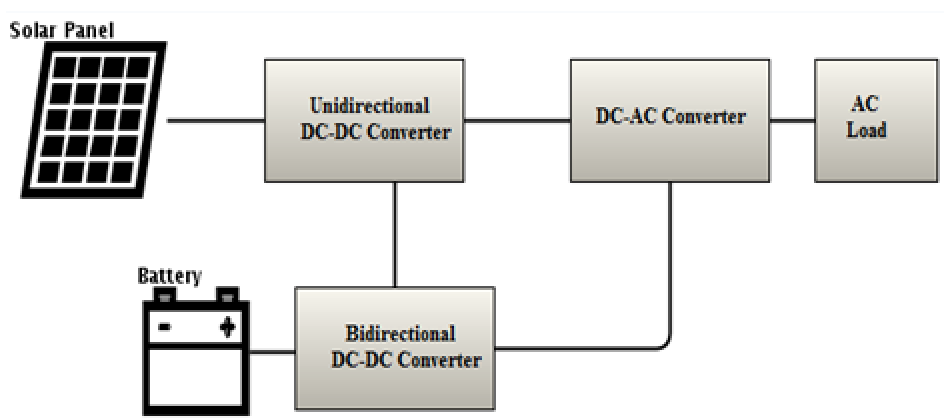

Fig. 1.8. Structure of PV power system connected to ac grid.

1.3. Objective

The objective of this project is to investigate the benefits of three-phase bidirectional dc-dc dual active bridge converter while using a silicon carbide MOSFET. After reading this, the readers should have a clear understanding of the work done in this thesis from modeling and simulations to designing and implementation of the prototype. The development of the entire process can be seen through the flow chart in Fig. 1.9.

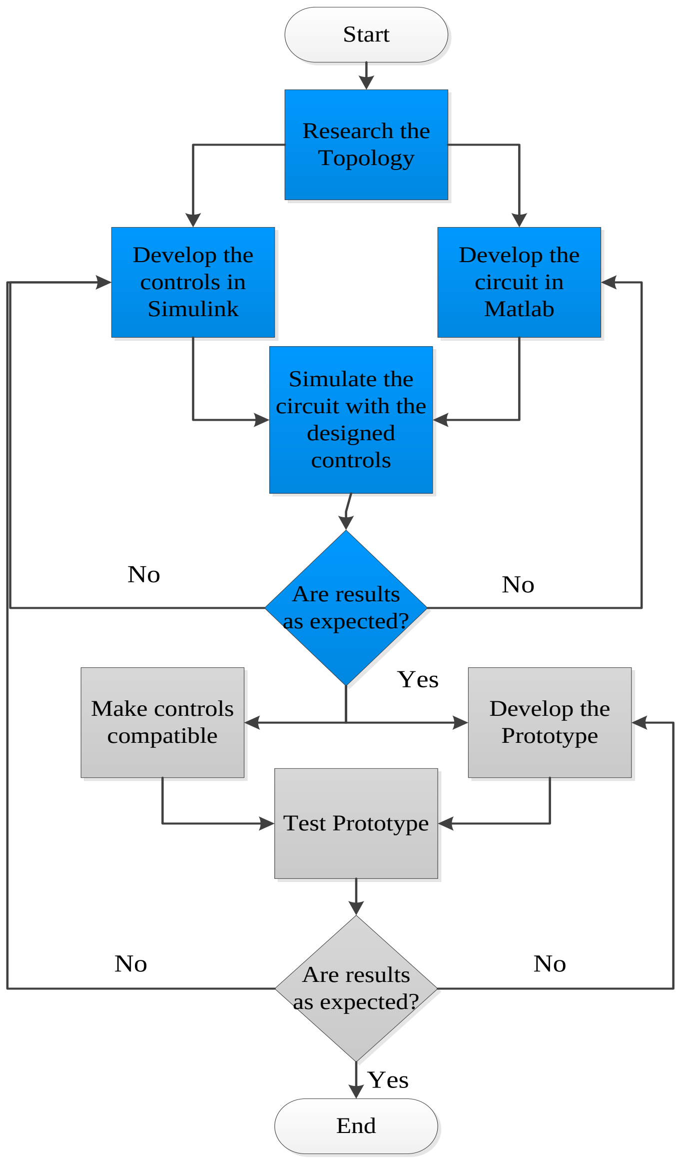

Fig. 1.9. Flow chart diagram of the project.

Each color in the flow chart represents a stage that was taken to accomplish this thesis. The first stage colored in blue focuses on the research and development of the circuit topology on a simulation program. The developments of the controls were also done in the same stage. The last step of stage one is to test the developed circuitry with the controls and see whether the results are as expected. The next stage colored in gray covers the design and implementation of the prototype. First thing in this stage is making the controls compatible to be used to test the prototype. After that, the prototype is tested and the results are checked and compared to simulation to confirm functionally of the prototype and demonstrates the benefits of the proposed three-phase DAB circuit with SiC devices. The chapters of this thesis are arranged to follow the flow chart. Chapter 1 will target the background of DC-DC converters and the motivation for this work. Chapter 2 outlines the design of the circuitry, controls, simulation testing and results. Chapter 3 gives the real world implementation of the design. Chapter 4 targets the experimental results and discussion. Chapter 5 provides conclusion found during this process and will also cover some insight for future work to improve the design.

CHAPTER 2

Modeling and Simulation

2.1. Introduction

Simulation is a powerful tool especially for power electronic designers. It is the first step that designers use before constructing a physical power electronic application. Not only does it save the designer time and effort but it is cost effective. Using simulation software gives the opportunity for a fast response and feedback. Thereby allowing users to intervene when the system has any error and fix it or investigate different options all before building the real one. Once the designer is satisfied with the simulation results, prototyping of the system can begin. Simulation is a good way to verify the concept and demonstrate the expected behavior of the design even though the prototype results may not match the simulation exactly due to some real losses.

2.2. Designing the circuit model

The circuit of the three-phase DAB was constructed in simulation software called Matlab/Simulink. Using the SimPowerSystems block-set, components of the circuitry were obtained. The major components are a diode, capacitor, resistor, inductor, ideal switch and linear transformer. There are also the current measurement blocks, voltage measurement blocks and the output scopes. As mentioned before, the three-phase dc-dc DAB model consists of two three-phase inverters connected together by a linear transformer. The input power supply is a constant DC voltage source and the output is also considered a DC voltage source that is smoothed by a capacitor. Fig. 2.1 shows the three-phase converter model designed in Matlab.

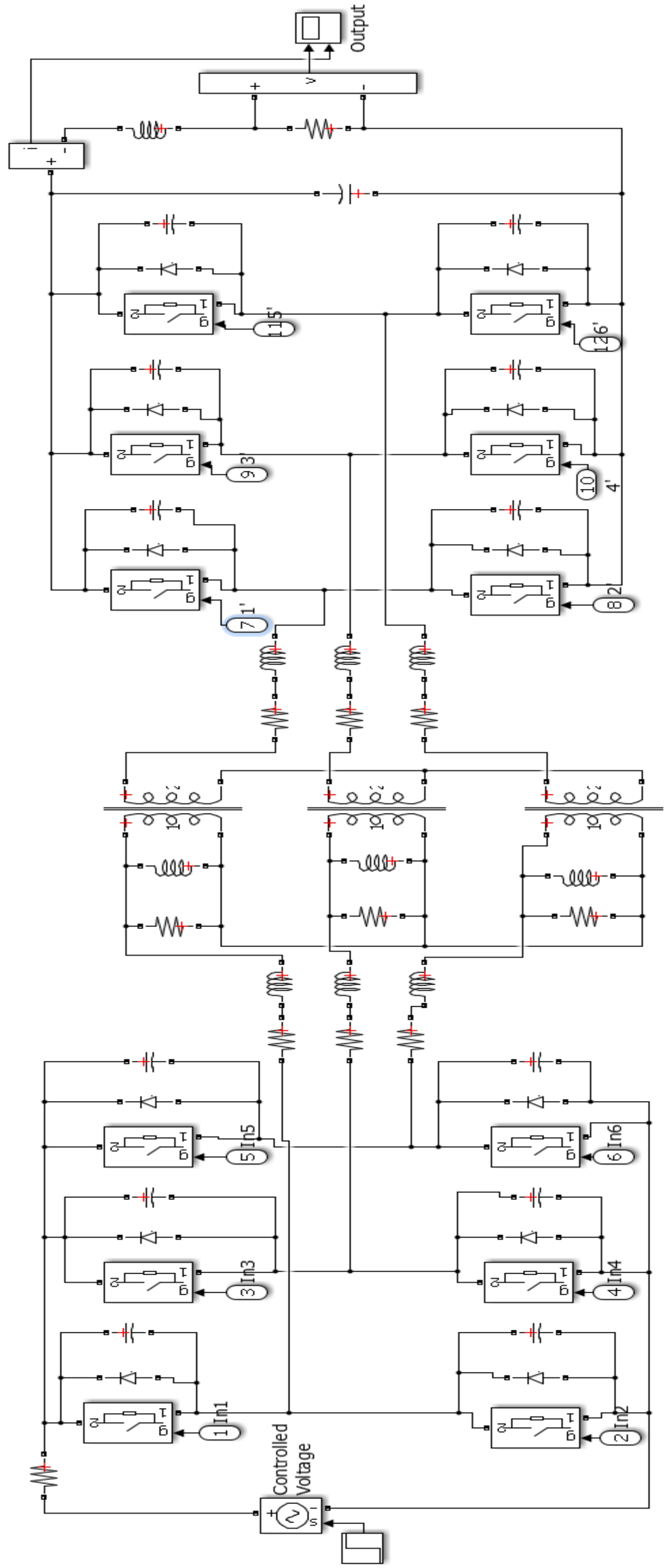

Fig. 2.1. Three-phase DAB model structure.

2.3. Controls

There are many ways to control a three-phase DAB but the working principle is always the same. In DAB topologies the switches usually activate at 50% duty cycle with a constant speed. Thus the two switches in one bridge will generate identical output. The output of one bridge will then be phase shifted from the previous bridge by 120 degrees. The energy will flow from the low voltage side to the high voltage side when the converter is in a boosting mode. The energy will reverse the direction when in buck mode. The energy flow can be controlled through the phase shift angle between the two inverters. The transformer will not only provide isolation to the topology but will also serve as energy storage using its leakage inductance. Using the phase shift modulation scheme on the DAB will enable the converter to operate under zero voltage switching conditions. However, the topology will undergo from light switching when operating at light loads.

In order for the controls that are designed for the simulation to be used to test the prototype, two frequencies were vital to know. The first one is the desired frequency. The second frequency is the clock cycle frequency of the field-programmable gate array (FPGA). Considering that the switches are made of SiC and the system also uses high frequency transformer, the controls frequency is chosen to be high. Any system that works with high frequencies will have a reduction in the harmonic content, leading to less power quality issues as well as greater power density. Three desired high frequency were chosen to test the simulation profile and the prototype. These frequencies are 100 KHz, 200 KHz, 300 KHz, and the clock cycle of the FPGA is 50MHz. In the controls, the chosen frequency is generated in the form of an integer multiple of the clock cycle. This is done by using a counter to count up one step per clock cycle. Pulses of desired length are then generated based on that timing. The signal is then shifted 120 degrees for the second bridge then another 120 degrees shift between the second and the third bridge. Also, a phase angle is introduced to the controls to control the amount and the direction of the power flow. Calculations of the respected desired frequencies are carried out next.

2.3.1. 100 KHz

First the time is calculated.

$$T = \frac{1}{f} = \frac{1}{100KHz} = 1e^{-5} \tag{2.1}$$

Then the counts per cycle is determined using the equation,

$$\frac{\mathrm{T}}{\mathrm{T_{Clock\ cycle}}} = \frac{1\mathrm{e}^{-5}}{2\mathrm{e}^{-8}} = 500 \text{ coutns} \tag{2.2}$$

Next step is to choose a dead time, were switches are OFF, to prevent shoot-through. 14 counts were chosen for the 100 KHz case. Finding when the switches are on at zero phase-shift is the next step considering the counts and dead-time. After that the results are shifted by 120 degrees for the second leg on the three-phase inverter, then shifted from that by120 degrees for the third leg bridge as follow,

At zero phase-shift, the switches 1 and 2 are on when,

$$1 \leq \text{ON} \leq 236$$

$$250 \leq \text{ON} \leq 486$$

At 120° phase shift, switches 3 and 4 are on when

$$168 \leq ON \leq 403$$

$$\left.\begin{array}{c} ON \geq 417 \\ ON \leq 153 \end{array}\right\}$$

At -120° phase shift, switches 5 and 6 are on when

$$\left.\begin{array}{c} ON \geq 334 \\ ON \leq 69 \end{array}\right\}$$

$$83 \leq ON \leq 319$$

These ranges are presented in the control as a logic gates such as AND or OR gates. Fig. 2.2 shows the full Simulink model for the 100 kHz controls. Fig. 2.3 shows a close look of the controls for the first bridge on both sides of the transformer (switches 1, 2, 1', and2'). It can be seen per the Fig. 2.3 that the control consists of a counter, a subtraction, an addition, an operational, a logical element, switches, and constant blocks.

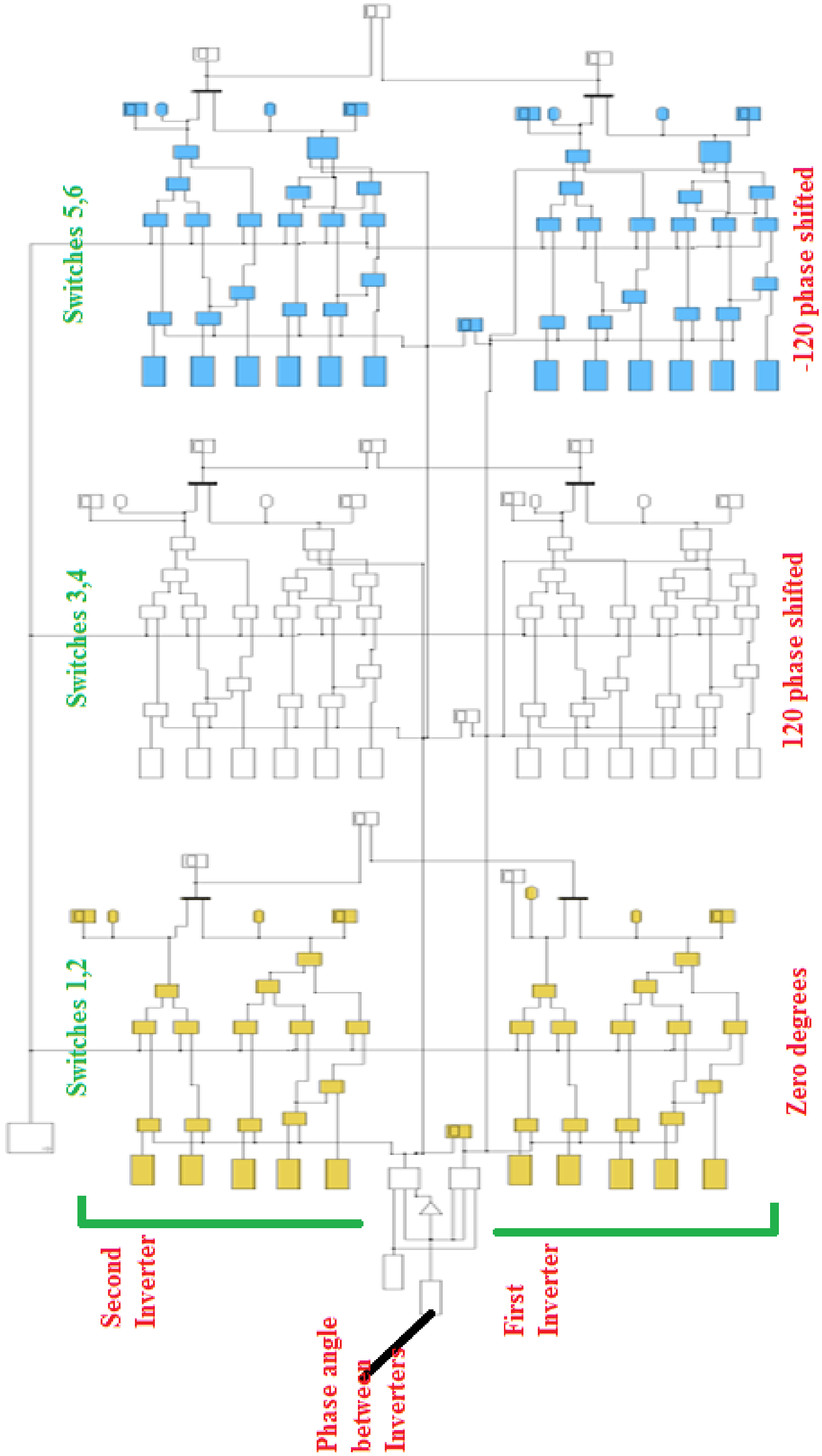

Fig. 2.2. Simulink model for the 100 kHz three-phase controls.

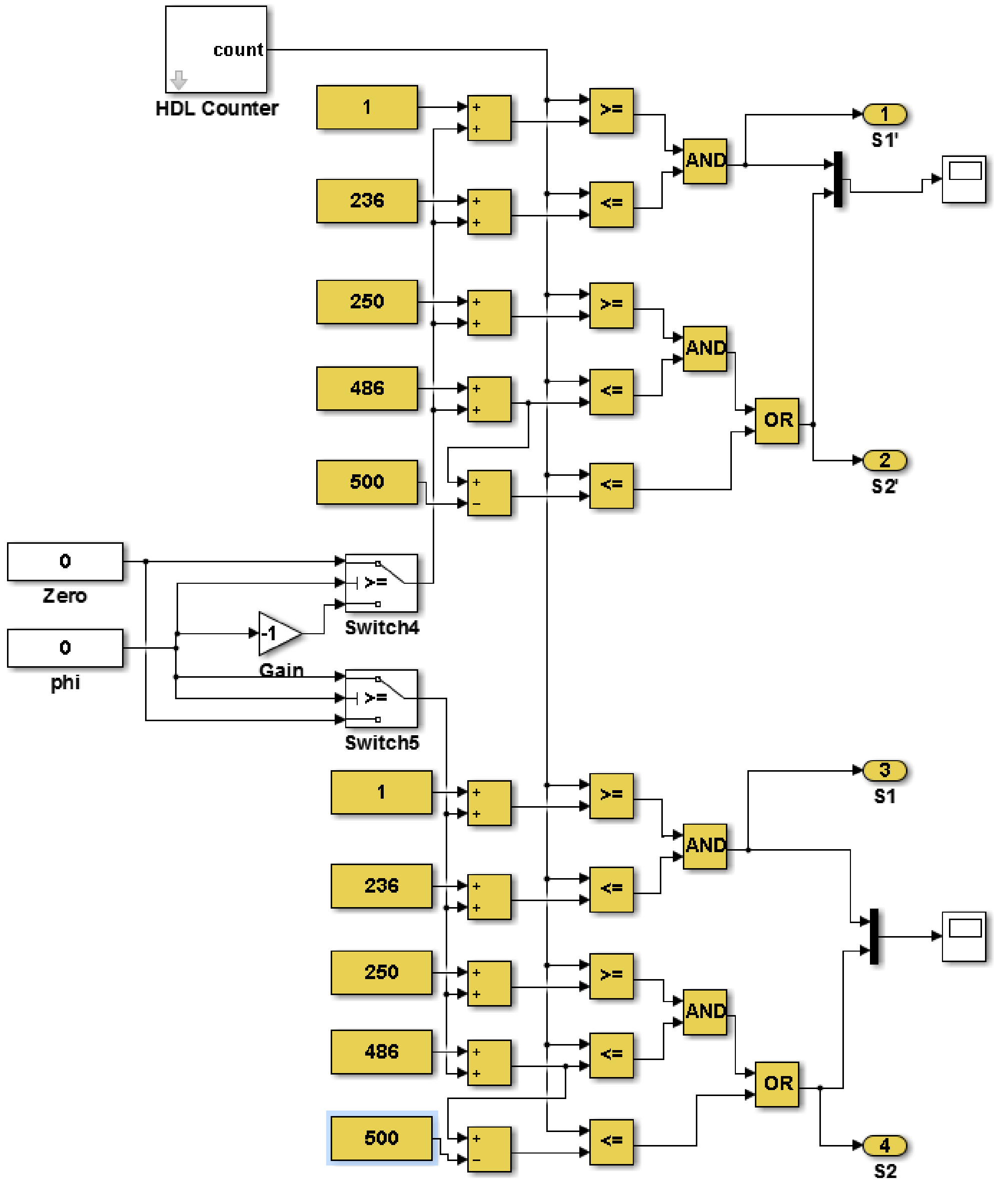

Fig. 2.3. Simulink control schematic for one leg of the three-phase inverter at 100 kHz.

2.3.2. 200 KHz

As aforementioned, the DAB family is chosen due to its ability to provide high power density with high speed. The 200 KHz and 300 KHz are built to see the system performance when increasing the frequency. Same as the 100 KHz, calculation starts by determining the time to obtain the number of counts. This time the dead time is chosen to be 5 counts since the switching device has low switching losses.

$$T = \frac{1}{f} = \frac{1}{200KHz} = 5e^{-6}s \quad (2.3)$$

Then the counts per cycle are determined using the following equation,

$$\frac{\mathrm{T}}{\mathrm{T_{Clock\ cycle}}} = \frac{5\mathrm{e}^{-6}}{2\mathrm{e}^{-8}} = 250 \text{ coutns} \quad (2.4)$$

Finding when the switches are on at zero phase-shift is the next step considering the counts and dead-time. Fig. 2.4 shows a close look of the controls for the first bridge on both sides of the transformer (switches 1, 2, 1', and2').

At zero phase-shift, the switches 1 and 2 are on when,

$$1 \le ON \le 120$$

$$125 \le ON \le 245$$

At 120° phase shift, switches 3 and 4 are on when

$$84 \le ON \le 203$$

$$\left.\begin{array}{l} ON \ge 208 \\ ON \le 78 \end{array}\right\}$$

At -120° phase shift, switches 5 and 6 are on when

$$\left.\begin{array}{l} ON \ge 168 \\ ON \le 37 \end{array}\right\}$$

$$42 \le ON \le 162$$

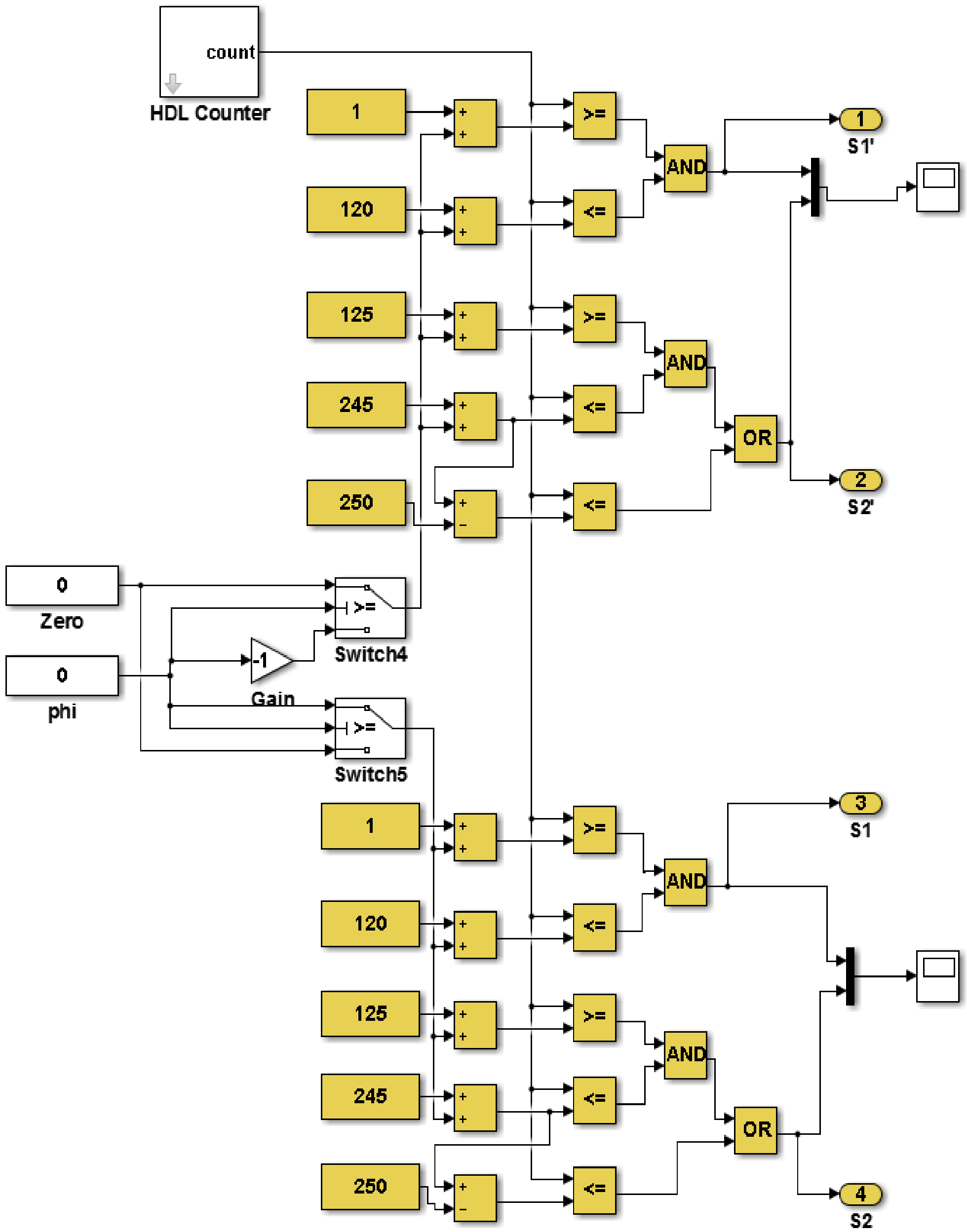

Fig. 2.4. Simulink control schematic for one leg of the three-phase inverter at 200 kHz.

2.3.3. 300 KHz Switching Function

A switching frequency of 300 KHz was chosen in case the transformer frequency range is suitable for over 200 KHz. The Simulink structure remains similar to the previous derivation.

$$T = \frac{1}{f} = \frac{1}{300KHz} = 3.333e^{-6}s \quad (2.5)$$

Then determining the counts per cycle using the equation,

$$\frac{T}{T_{\text{Clock cycle}}} = \frac{3.333e^{-6}}{2e^{-8}} = 166.667 \approx 168 \text{ coutns} \qquad (2.6)$$

At zero phase-shift, the switches 1 and 2 are on when,

$$1 \leq ON \leq 79$$

$$84 \leq ON \leq 163$$

At 120° phase shift, switches 3 and 4 are on when

$$57 \leq ON \leq 135$$

$$\left.\begin{array}{c} ON \geq 140 \\ ON \leq 51 \end{array}\right\}$$

At -120° phase shift, switches 5 and 6 are on when

$$\left.\begin{array}{c} ON \geq 113 \\ ON \leq 23 \end{array}\right\}$$

$$28 \leq ON \leq 107$$

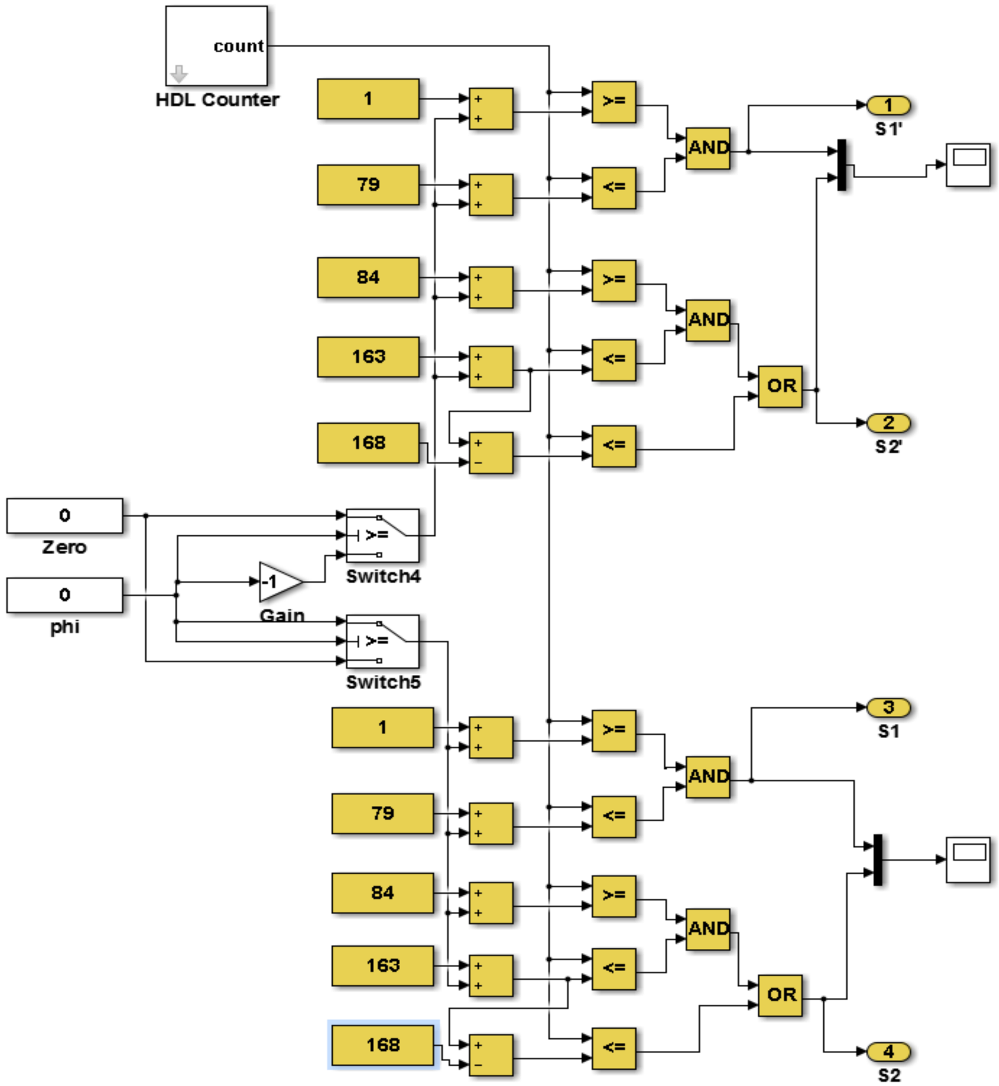

Fig. 2.5. Simulink control schematic for one leg of the three-phase inverter at 300 KHz.

Functionality of the controls were tested and approved. Fig. 2.6 shows the pulse signal going to switches 1, 2, 3, 4, 5, and 6. Notice the shifting of the signal on 3, 4, 5, and 6 from the signals applied to the switches 1 and 2. Fig. 2.7 and Fig. 2.8 focus on the first bridge in the two inverters. At zero degrees both inverters are in phase but when changing the phase angle to 45 degrees for example, the second bridge shifts from the first bridge. This could be leading or lagging depending on the desired direction of the power flow.

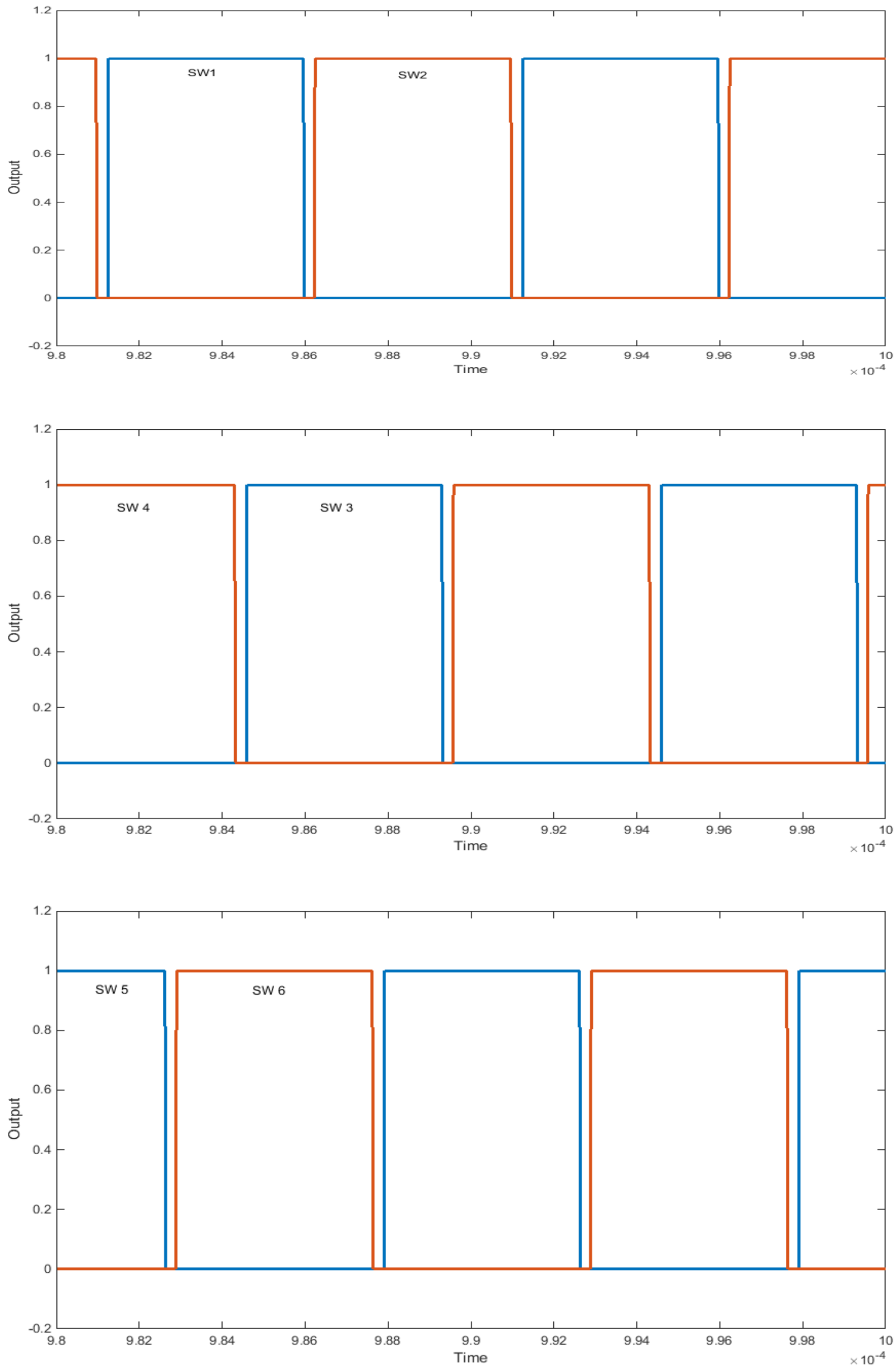

Fig. 2.6. Switches 1,2,3,4,5,6.

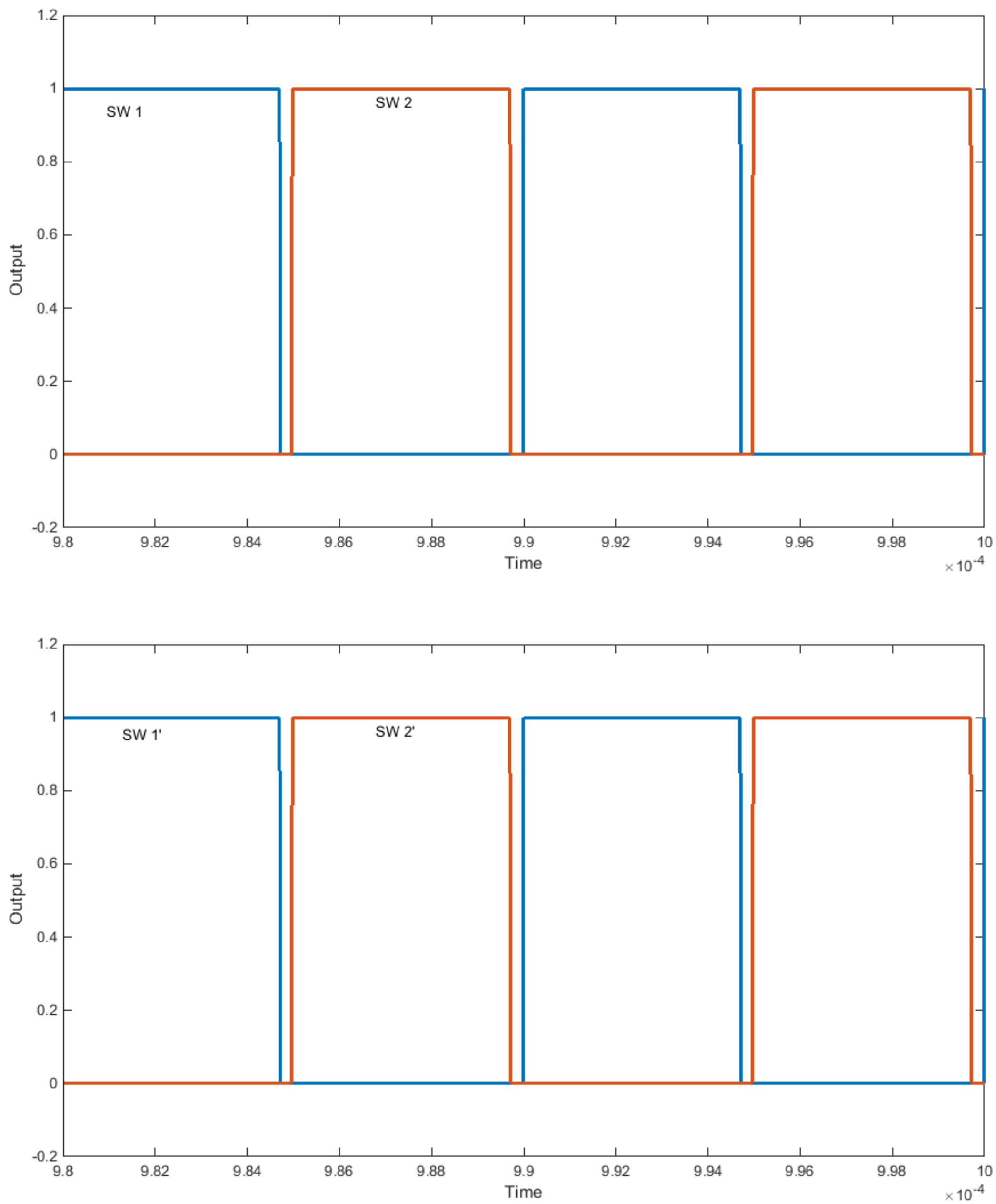

Fig. 2.7. Switches 1, 2, 1’, and 2’ in phase at 0 degrees.

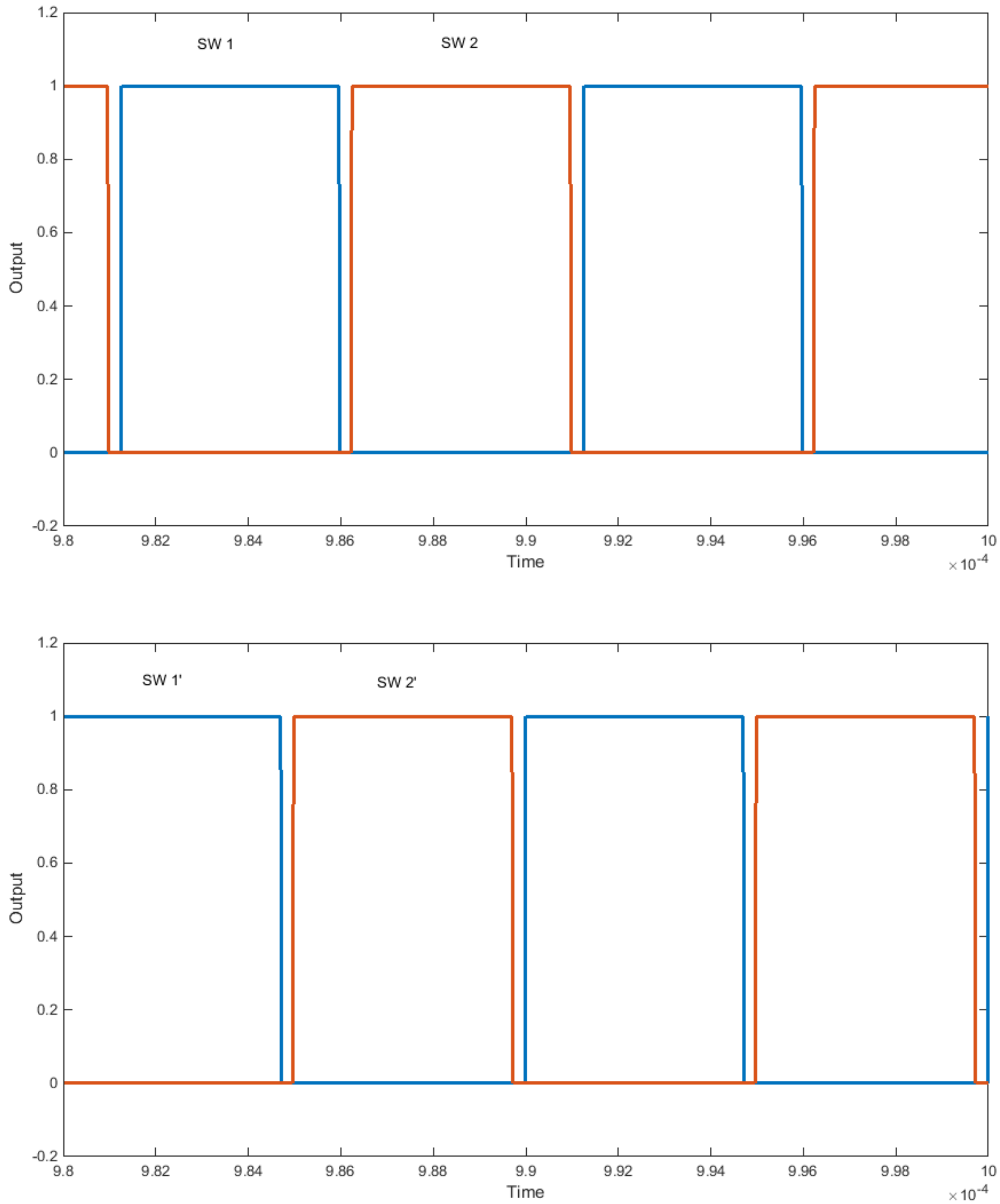

Fig. 2.8. Switches 1, 2, 1', 2' phase shifted at 45 degrees.

CHAPTER 3

Circuit Layout Design and Prototyping

3.1. Introduction

This chapter explores the process of designing and building the prototype. There are factors that need to be looked at when making a prototype. The power level that this prototype can be tested at is the first factor. Next is the components selection that drives the SiC MOSFET and withstands current limits. The last factor to consider in prototyping design and construction is the budget, how much creating the entire prototype is going to cost.

3.2. Components selection

Recently, silicon carbide (SiC) material have allowed the industry to fabricate smaller, faster, and more efficient power semiconductor devices compared to silicon (Si) [36]. Some of these devices include power diode, thyristor, power MOSFET, and IGBT. This come an advantage when building power electronic systems.

Using surface mount devise (SMD) adds another advantage when constructing printed circuit board (PCB) projects. This section will target the parts used to build the three phase dual active bridge on a PCB and discusses the reason behind selected parts.

3.2.1. SiC MOSFET

For this project a latest version of SiC MOSFET manufactured by Cree is used. As mentioned before, SiC devices have numerous advantages. Frist, the SiC MOSFET performs as a fast switching speed which leads to less switching losses. Also, it has the ability to block high voltages with low $R_{DS(on)}$. These capabilities results in higher system efficiency and increase the system switching frequency. Having the system switch at a higher frequency decreases the harmonic content resulting in less power quality issues. Lastly, SiC MOSFET can operate at high temperature which reduces cooling requirements. It is qualified to be used in building applications that use auxiliary power supplies, solar Inverters, high-frequency applications or high voltage DC/DC converters which is the target of this project.

3.2.2. Gate Driver

Finding the right gate driver depends on the specification of the chosen MOSFET. The output of the gate driver should be greater than or equal to the threshold voltage ($V_{GS(th)}$) of the desired MOSFET. This gate drive, which designed by IXYS, operates from 4.5V to 35V which is enough to drive the 1200V SiC MOSFET used in this project. It has up to 9A peak of output current with low supply current. Also, it has the ability to disable output under faults with low propagation delay time. Other features include low output supply, matched rise and fall times, and ability to withstand heat up to 125° C. overall, IXDN406SI gate drive can drive any MOSFET to minimum switching time and maximum frequency limits. Fig. 3.1 shows the gate driver connection circuitry.

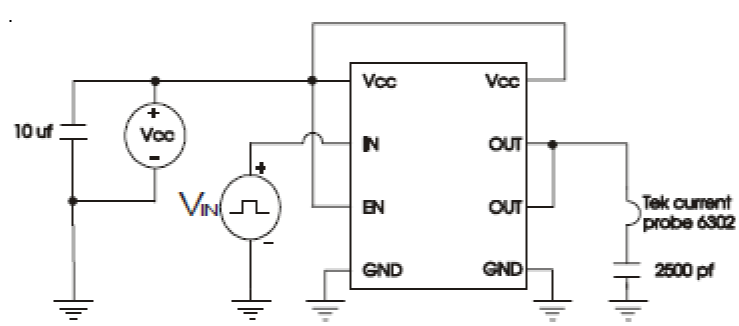

Fig. 3.1. Gate driver connection circuitry [37].

3.2.3. Optocoupler Circuit

Optocouplers, also called opto-isolators, are devices that are used to deliver electric signals between two circuits just like the operation of a switch. It could also be used to send feedback signals when used for analog devices. It provides isolation and protects circuit's components. The way this device work is quite simple. As can be seen in Fig. 3.2 It consists of a light emitting diode (LED) on the input side that produce current and a phototransistor at the output that conducts the current and transfers the signal.

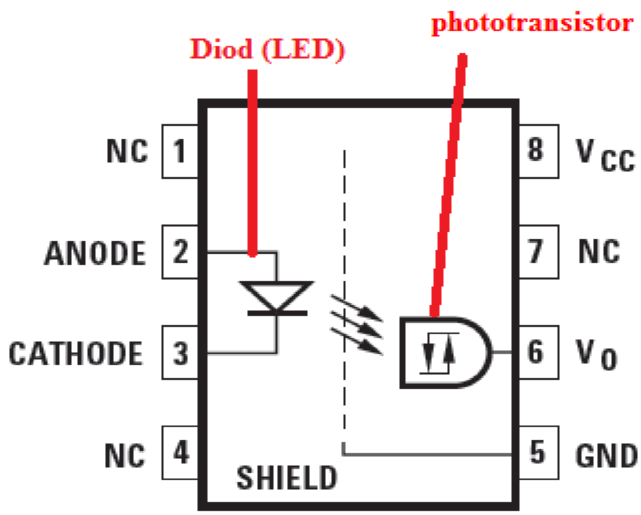

Fig. 3.2. Inside circuitry of an optocoupler [38].

The ACPL-4800-300E optocoupler designed by Avago Technologies was found suitable for this particular project due to some of the advantages that carries. It provides logic-compatible waveforms which exclude the use of extra devices to construct properly shaped waves. This device also has totem pole output therefore pull-up resistors are no longer required to drive either power modules or gate drives. This particular optocoupler activates at 4.5 volts and works up to 20 Volts. The recommended connection circuitry for this optocoupler is shown in Fig. 3.3.

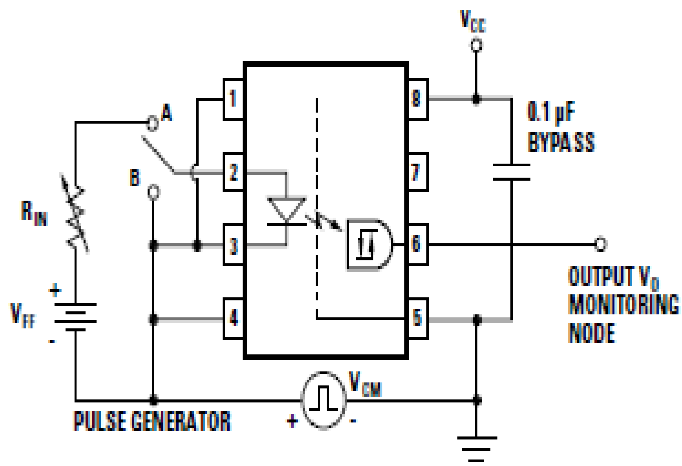

Fig. 3.3. Connection circuitry for ACPL-4800-300E opto-coupler [38].

3.2.4. DC-DC Convertors

There are two dc-dc converters used in this project. These are used to provide isolated power to the optocoupler and the gate driver. Both converters are manufactured by Recom. These RP series have up to 5.2KV isolated voltage rating with 1 W power and dual output signals. The RP-1205D provides unregulated 1W with input voltage of 12V and 5V output. The RP-1212D also provides unregulated 1W but with input voltage of 12V and +/- 12V output. Table 3.1 shows the specifications for these converters.

Table. 3.1. Specifications of the converters.

Part Number SIP7	Input Voltage (VDC)	Output Voltage (VDC)	Output current (mA)	Efficiency (%)	Max Capacitive Load
RP-1205D	12	±5	±100	74-76	±470µF
RP-1212D	12	±12	±42	79-82	±220µF

3.2.5. SMD Devices

Surface mount devices (SMDs) have shown promising outcomes in recent technology applications and products compared to through-hole devices. These devices have helped in reducing the size of components and board layouts. Also, using SMDs help to block excessive inductance and capacitance that are freeloading around a circuit. SMDs require less holes, and smaller board size when building a circuit board. Moreover, these devices can withstand mechanical conditions such as shaking and vibrations. These factors have made SMDs become a more profitable and practical choice than through-hole devices. For this project all the devices including capacitors, resistors, diode, and integrated circuit chips are surface mount. The case size usually depends on the value and rating of the part but the general shape would be something like Fig. 3.4.

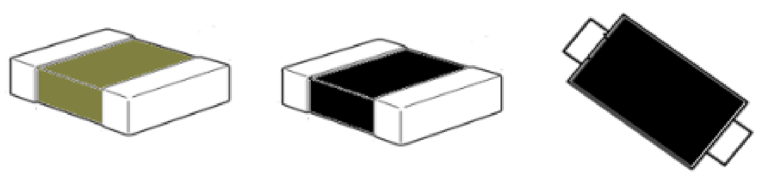

Fig. 3.4. General shape of capacitor, resistor and diode SMDs.

SMDs used in this project include the capacitors which are multilayer ceramic chip manufactured by Kemet. Its voltage can range from 4 volts up to 50 volts. Some of these capacitors were used as bypass and some were used for decoupling but the main reason for using ceramic capacitors is its ability to perform at a high frequencies. The zener diode manufactured by Diode Inc. is used to clamp the output voltage of a dc-dc convertor used in the circuit. However, the resistors are a standard thick film chip manufactured by Vishay. The other SMD parts used were an optocoupler build by Avago Technologies Inc, a gate driver manufactured by IXYS- Corporation and the PL140 planar transformer manufactured by Coilcraft.

3.2.6. Transformer

There are some factors to be considered when choosing a transformer. Operation at high frequency, skin effect and proximity effect are taken into account. These factors are achieved in different design methods, one of which is the planar transformers. Planar transformers have several types. There are thick-film based, low temperature co-fired ceramic (LTCC) based, thin-film based, and PCB based which is used in this project due to its advantages. Low cost, frequency and the power range were the lead factors in choosing this method. The typical frequency for this type of planar transformers could range from 20 KHz to 2.0 MHz and runs at wide power rating, from 1.0 W to 5.0 KW [39]. Three single-phase planar transformers were used in this project. The transformer has turn ratio of 11:1 or 11:2 depending on how it is connected. Its frequency ranges from 200 KHz to 500 KHz at 140 Watts rated power.

3.2.7. Heat Sink

Heat sinks are devices that are used in cooling power semiconductor devices. These power semiconductor devices cannot handle heat generated by it therefor an aluminum heat sink is used for that matter [24]. The junction temperature of the device must be known in order to pick the right heat sink. Also, the thermal resistance between the junction and the ambient plays a big role in choosing the size of the heat sink. The thermal resistor can be calculated using equation 3.1.

$$R_{\theta ja} = R_{\theta jc} + R_{\theta ca} + R_{\theta sa} \quad (3.1)$$

where,

$R_{\theta jc}$ is the thermal resistance between the junction and the case of the power device.

$R_{\theta ca}$ is the thermal resistance between the case of the power device and the heat sink part.

$R_{\theta sa}$ is the thermal resistance between the heat sink device and the ambient.

Using these resistors and the power dissipation of the power device, the SiC MOSFET, the junction temperature can be derived from the equivalent circuit diagram in Fig. 3.5 as follows:

$$T_j = P_d\left(R_{\theta jc} + R_{\theta ca} + R_{\theta sa}\right) + T_a \quad (3.2)$$

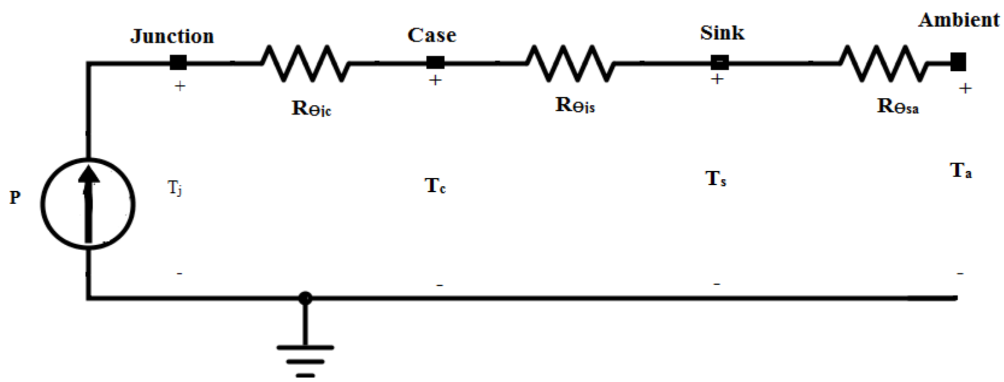

Fig. 3.5. Equivalent circuit of heat flow based on thermal resistance.

Some of these values can be obtained from the power device data sheet while some need to be calculated. Table 3.2 shows the values of the known and calculated thermal temperatures and resistance. $R_{\theta ca}$ is calculated depends on the thermal compound that will be used to seal the area between the device and the heat sink. These thermal resistances are computed using equations 3.3 and 3.4.

$$R_{\theta SA} = \frac{T_J - T_A}{P_D} - \left(R_{\theta JC} + R_{\theta CS}\right) \quad (3.3)$$

$$R_{\theta CS} = \frac{1}{R_{\theta} * A} \quad (3.4)$$

Table. 3.2. Temperatures and thermal resistance for the SiC MOSFET

Definition	Symbol	Value	Unit
Junction temperature	T_j	150°C	C
Ambient temperature	T_A	40°C	C
Power dissipation	P_D	25W	W
Junction to case thermal resistance	$R_{\theta JC}$	1°	C/W
Thermal paste thermal resistance	R_{θ}	350000	W/m^2°C
Transistor Area	A	3.276e-4	m^2
Case to sink thermal resistance	$R_{\theta CS}$	0.008°	C/W
Sink to ambient thermal resistance	$R_{\theta SA}$	3.39°	C/W

3.3. Layout Design and Prototyping

3.3.1. Introduction

The next step is to develop a prototype to demonstrate the benefits of the proposed three-phase DAB circuit. This starts by designing and manufacturing a printed circuit board (PCB). PCBs are considered to be a better method for constructing a circuit on a breadboard. It is easy to make mistakes connecting components in breadboards. Unlike breadboards, PCBs eliminate making these mistakes unless the user made the wrong connections in the schematic. Typical PCB consists of conductive and non-conductive layers. The conductive layer is made of copper and fiberglass for the non-conductive layer. The board can be a single sided layer, double sided layer, or multilayers. The copper layer forms the traces that connect the circuit together while the fiberglass provides isolation between the traces.

3.3.2. Printed Circuit Board (PCB)

The first step in designing PCBs is choosing design software. CadSoft EAGLE PCB Design Software is used in this project. The circuit is first constructed on a schematic editor sheet as shown in Fig. 3.6. The layout tool is then used to place the parts in the desired location and connect the traces based on the schematic connections. The next step after drawing the schematics and finish the layout is to check for any design rules errors. Once the design is finalized and ready to be sent out for manufacturing, the PCB software generates files that describe each layer. This would include the dimensions, drill holes locations, pads, and vias.

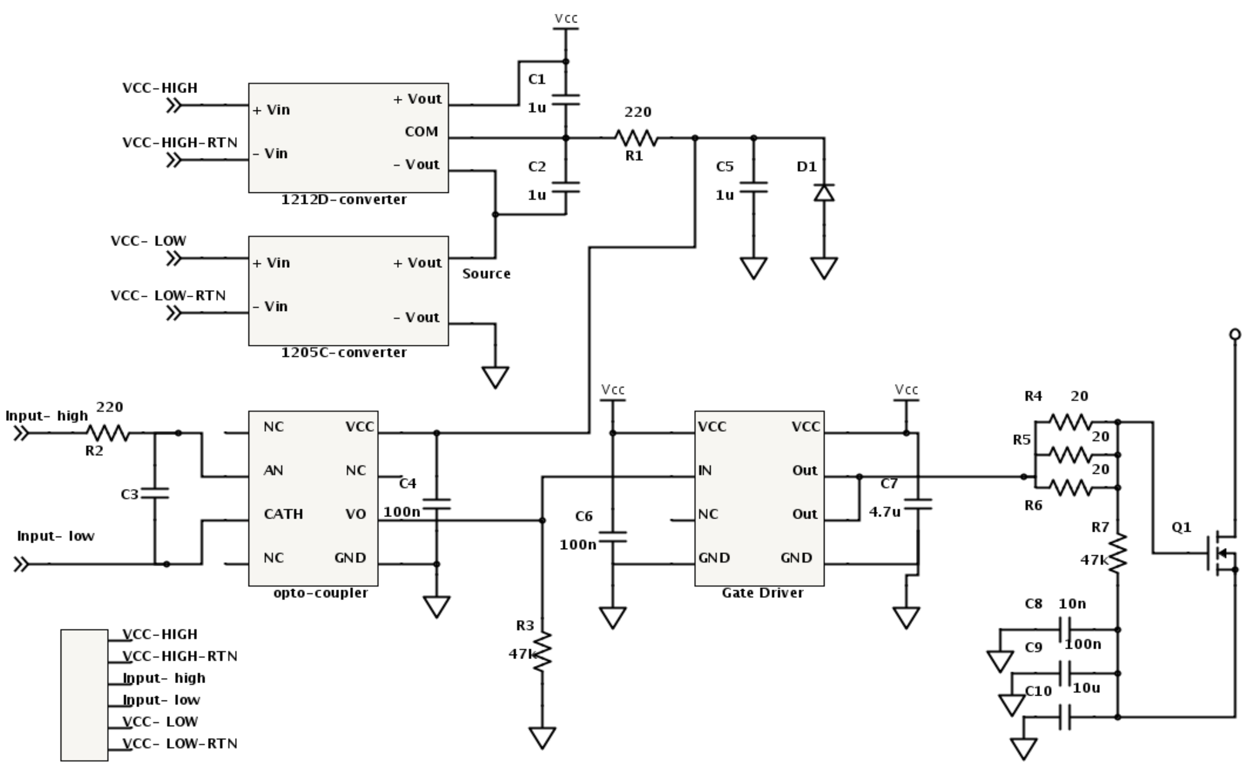

Fig. 3.6. The driver circuit for one SiC MOSFET.

As can be seen in Fig. 3.6, the driver circuit for one MOSFET consists of a gate driver, an opto-isolator, and two isolated DC-DC converters. The power of the circuitry is provided by the two dc-dc converters. One converter provides positive bias and the other provides the negative bias. The output of both converters is connected together in series and the common pin is referenced to the source of the MOSFET. Thus, they control the gate pulse positive and negative voltage. The negative voltage created from the converters is used as a reference ground for the gate driver and the optoisolator. The diode, placed at the common terminal, is used to clamp the voltage incase the voltage exceeds the maximum ratings of the optoisolator. Once the design is finalized for one switch, it is a matter of replicating the circuitry to form a half bridge board. Fig. 3.8 shows the schematic design for a half bridge circuit. Since the transformers are also SMD, PCBs are made for them. Fig. 3.7 shows the schematic design for a single phase planar transformer.

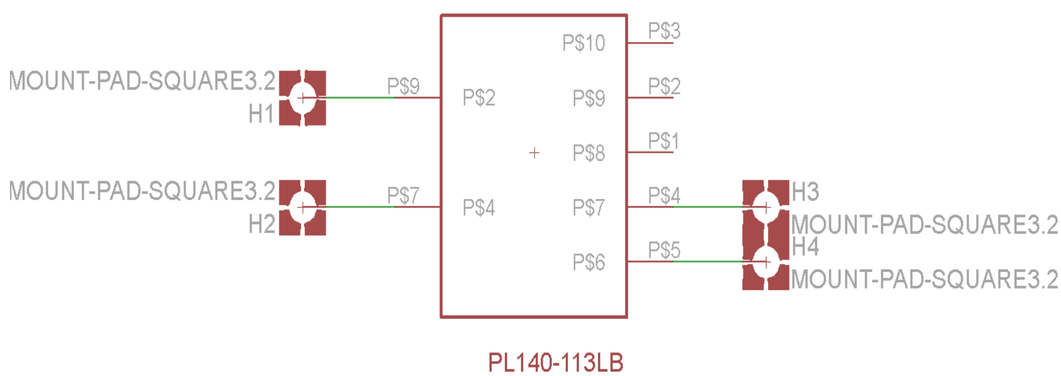

Fig. 3.7. The schematic design for a single phase planar transformer.

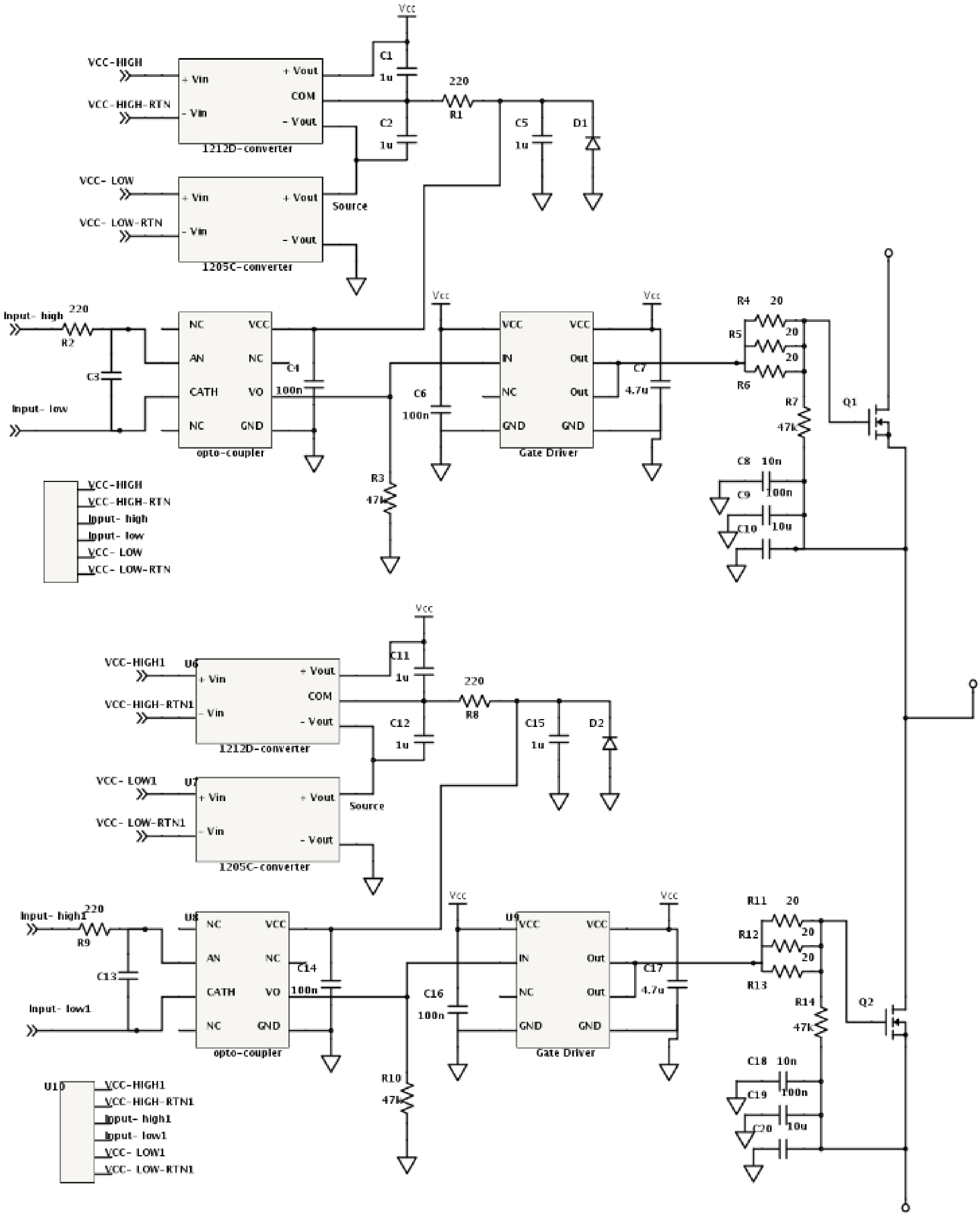

Fig. 3.8. The board schematic for a half bridge circuit.

After finishing the schematic design, finishing the board layout is next. Due to the size limitations that the CadSoft EAGLE PCB Design Software has, a half bridge is made in one. There are a couple of questions that the designer must take under consideration before the design of any PCB. Traces raise most of these questions. Some of these questions would include the traces length, trace width, number of traces, and the distance between traces. These could be answered knowing the current expected to be carried in these traces, how much heat the trace can handle and the thickness of the copper board used. Over the years IPC curves were used to determine the relationship between the temperature rise and the current depending on some factors. Some of these are PCB size and thickness, number of traces carrying the current, trace separation, or pitch, presence or absence of the ground and/or power copper plane, and System cooling conditions[40]. For this project, the trace width was calculated using formulas from IPC-2221 and the calculation was carried as follows:

First, the Area is calculated:

$$\text{Area}(\text{mils}^2) = \frac{\text{Current(Amps)}}{\text{K}*\text{Tempreture rise}(°\text{C})^{\text{b}^{1/\text{C}}}} \tag{3.8}$$

Then, the Width is calculated:

$$\text{Width}(\text{mils}) = \frac{\text{Area}(\text{mils}^2)}{\text{Thickness[oz]}*1.378[\text{mils/oz}]} \tag{3.9}$$

Where k, b, and c are constants resulting from curve fitting to the IPC-2221 curves. But since there are two type of layers found on PCB design, internal layers and external layers, the variables k, b, c are defined as follow:

For IPC-2221 internal layers: k = 0.024, b = 0.44, c = 0.725

For IPC-2221 external layers: k = 0.048, b = 0.44, c = 0.725

Using copper board thickness of 1 oz. with the assumption of using the maximum current of the MOSFET, 17 Amps, the external layer was calculated and found to be 0.59 inches. Table 3.3 shows the calculations of the required trace width.

Table. 3.3.Trace width calculations.

Current	17 Amps
Thickness	1 oz/ft^2
Required trace width	15mm=0.59 inch
Resistance	0.000857 Ohm
Voltage drop	0.0146 Volts
Power Loss	0.248 Watts
Trace Length	1 inch

After finishing the schematic design and calculating the required traces, the layout is then developed based on the desired space and location. Fig. 3.9 shows the board layout for a half bridge circuit. Since one board makes a half bridge, six boards are made to complete two three-phase inverters and three PCB board are made for the planer transformer. Once the PCBs arrive the boards are populated and the final three-phase DAB topology is put together. Fig. 3.10 shows the final prototype.

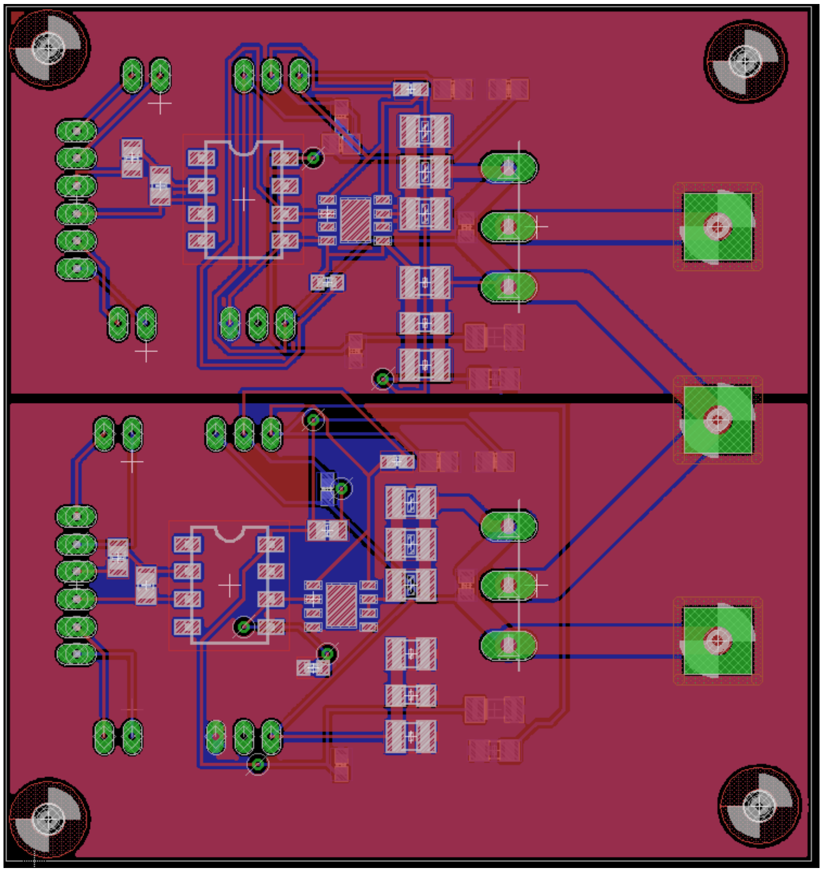

Fig. 3.9. Board layout for a half bridge circuit.

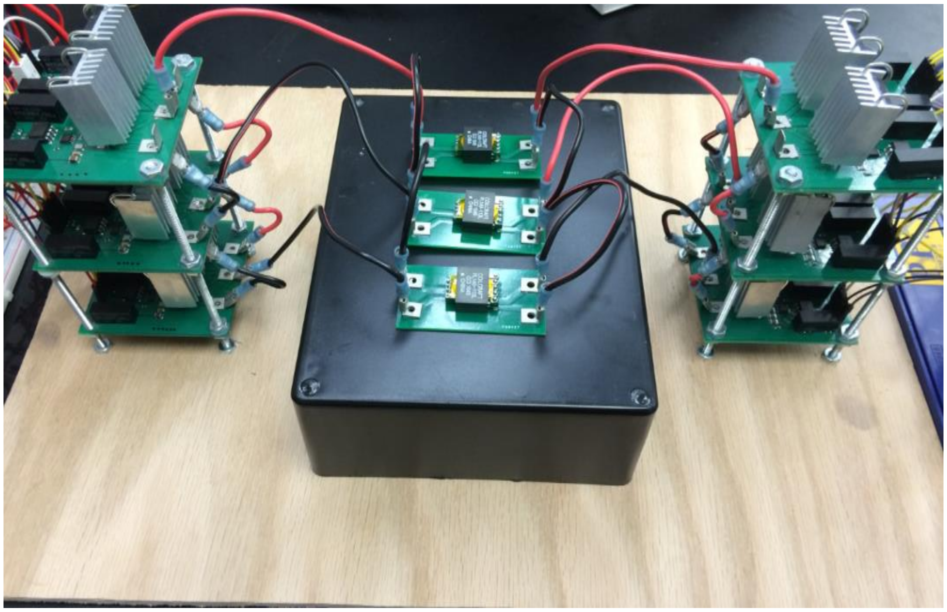

Fig. 3.10. Three-phase bidirectional dc-dc DAB prototype.

CHAPTER 4

Results and Discussion

4.1. Simulation

4.1.1. Components Values

In order to have simulation values close to the prototype results, values of the devices used are changed to datasheet values. In simulations most of the devices behavior is ideal unless the values are changed to match a real device. This enables the designer to see the expected behavior of the prototype. This process begins by the ideal switch which represents the SiC switch in this prototype, the turn on resistance $R_{DS(on)}$ is changed to 160 mΩ. The other major part that needs change is the transformer. The magnetic and the leakage inductance are calculated by running two tests, open circuit and closed circuit test. Using and LCR/ESR meter, both tests are taken across the high side (primary) and then reflected to secondary side (low) of the planer transformer used in this project. These theoretical calculations help determine the maximum limits for this transformer and provide simulation values. Equations 4.1-4.4 show these calculations at 100 KHz. The magnetizing resistance and inductance are then extracted from the total impedance. In the same way, the leakage resistance and inductance are obtained.

$$Z_{high\ side, open\ circuit\ test} = 198.5\angle 89.12 = 3.05 + j198.477\ \Omega \tag{4.1}$$

$$L_m = \frac{jx}{2\pi f} = 316\mu H \tag{4.2}$$

$$Z_{high\ side, closed\ test} = 67\angle 50.91 = 42.25 + j52\ \Omega \tag{4.3}$$

$$L_l = \frac{jx}{2\pi f} = 89.35\mu H \tag{4.4}$$

4.1.2. Simulation Results

As mentioned before the designed three-phase topology is run at constant speed with 50% duty cycle using the phase shift modulation. Each leg of three-phase inverter is phase shifted by 120 degrees from the previous leg. There is also the phase angle that controls the power flow. Two constrains are taken during the simulation process. The first case targets the behavior of the model when placed in high voltage application such as the system presented in Fig. 4.1. The second case resembles the prototype scenario since the prototype is tested at low voltage level.

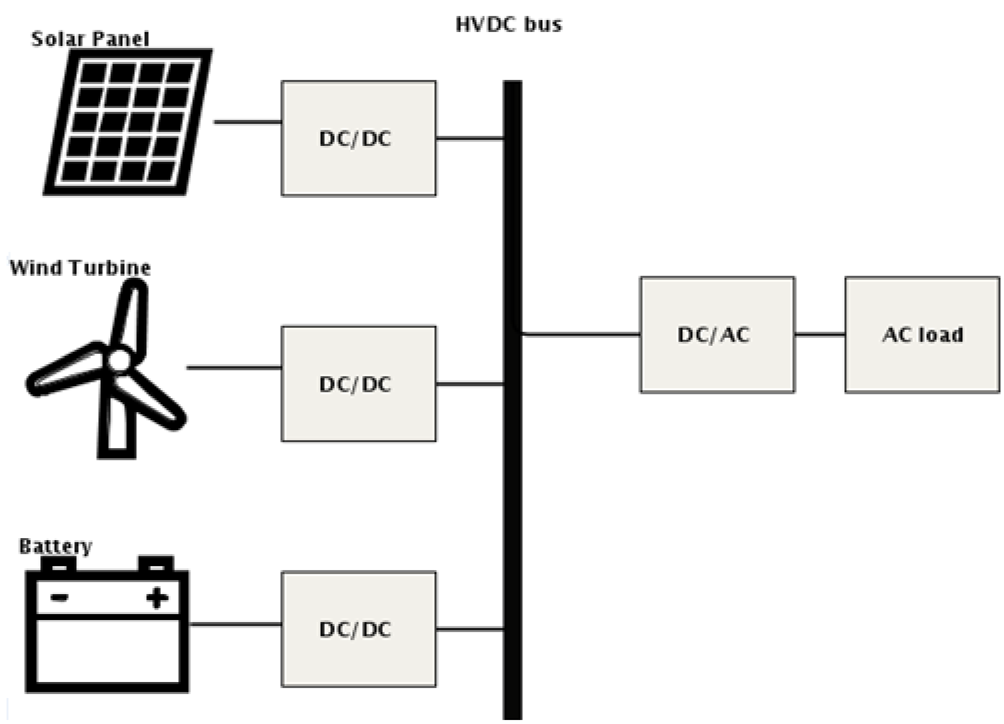

Fig. 4.1. An example of HV renewable energy system.

During the first case a large load is used to symbolize the HVDC bus. Having a large load will affect the transformer ratio. Using the controls described in chapter 2, the simulation results of the output voltage with a large load at 20 volts input can be seen in Fig. 4.2.

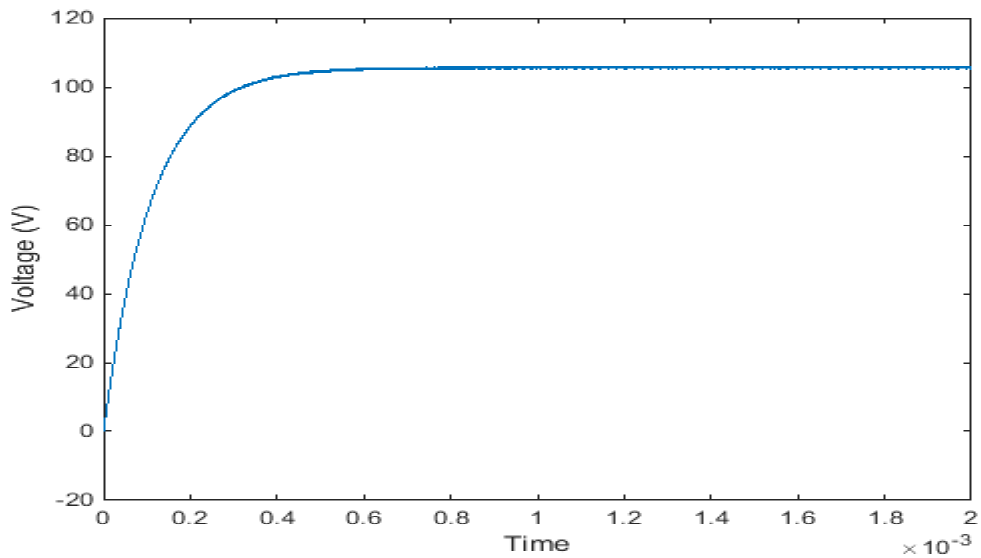

Fig. 4.2. Output dc voltage using high load.

In the second case, the target is to track the current flow through the system in which is achieved by testing with a low load at the output. Evaluating the power flow is considered the most important aspect when testing the concept of any topology. When the system uses high load the current is really low and distorted. However, testing at a lower load enable the system to draw more current and produce less power distortions. These results are saved to be compared to the prototype results later. As mentioned in Chapter 2, there are three controls with different frequencies made for this project since the SiC MOSFET as well as the planer transformer operates at very high frequency. There are some advantages and disadvantages when using switching a system at high frequency. Using high frequencies may help in reducing the size of the passive components. Moreover, the harmonic content decrease which lead to less power quality issues. The switching losses on the other hand increases at high frequency which lead to having less output power. A simulation comparison is taken between 100 KHz, 200 KHz and the 300 KHz. The results shown in Table 4.1 and Table 4.2 demonstrate that point. The test is taken at lower power with different phase angle between the two bridges. The power rating at the 100 KHz is higher than the 200 KHz and the 200 KHz power rating exceed the 300 KHz results. Also, the power increase when increasing the phase angle that controls the power flow direction.

Table. 4.1. Comparison of output voltage, current, and power between 100 KHz and 200 KHz.

Phase Angle	**Simulation at 100 KHz**			**Simulation at 200 KHz**		
	Voltage (V)	Current (A)	Power (W)	Voltage (V)	Current (A)	Power (W)
-45	2.114	0.9611	2.032	-1.109	-0.5041	0.559
-30	1.378	0.6263	0.863	-1.562	-0.7099	1.109
-15	0.6009	0.2731	0.164	-1.94	-0.822	1.711
0	2.772	1.26	3.493	1.039	0.4725	0.491
15	3.269	1.486	4.585	1.941	0.8821	1.712
30	3.24	1.47	4.763	2.114	0.9611	2.031
45	3.126	1.421	4.442	2.119	0.9632	2.041

Table. 4.2. Output voltage, current, and power at 300 KHz.

Phase Angle	**Simulation at 300 KHz**		
	Voltage (V)	Current (A)	Power (W)
-45	-1.311	-0.5959	0.781225
-30	-0.7647	-0.3476	0.26581
-15	-0.0993	-0.0451	0.004479
0	0.55	0.2491	0.137005
15	1.076	0.4891	0.526272
30	1.464	0.6656	0.974438
45	1.688	0.7625	1.2871

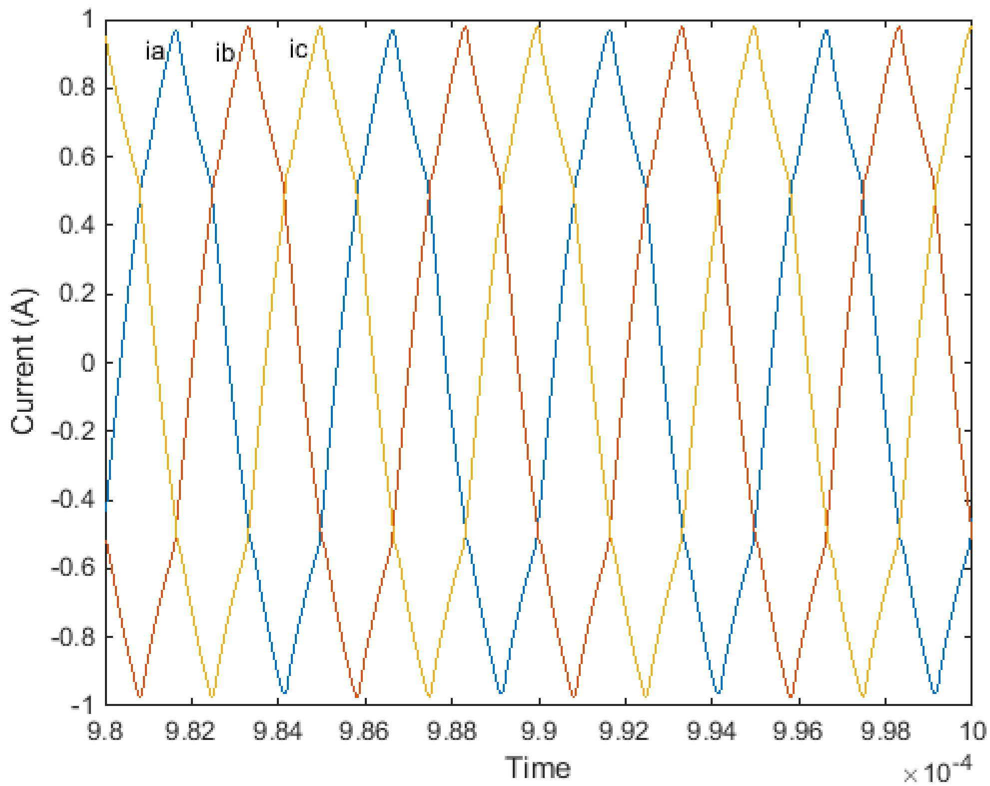

Fig. 4.3. Current at the secondary side of the transformer.

Fig. 4.3 shows the three-phase transformer output current. At the peak point, the current is 120 degrees phase shifted from the next one. Likewise, the voltage is stepped up based on the conversion ratio of the transformer and the output of V_a is phase shifted by 120 degrees to form V_b then shifted again to form V_c. This can be seen in Fig. 4.4 below:

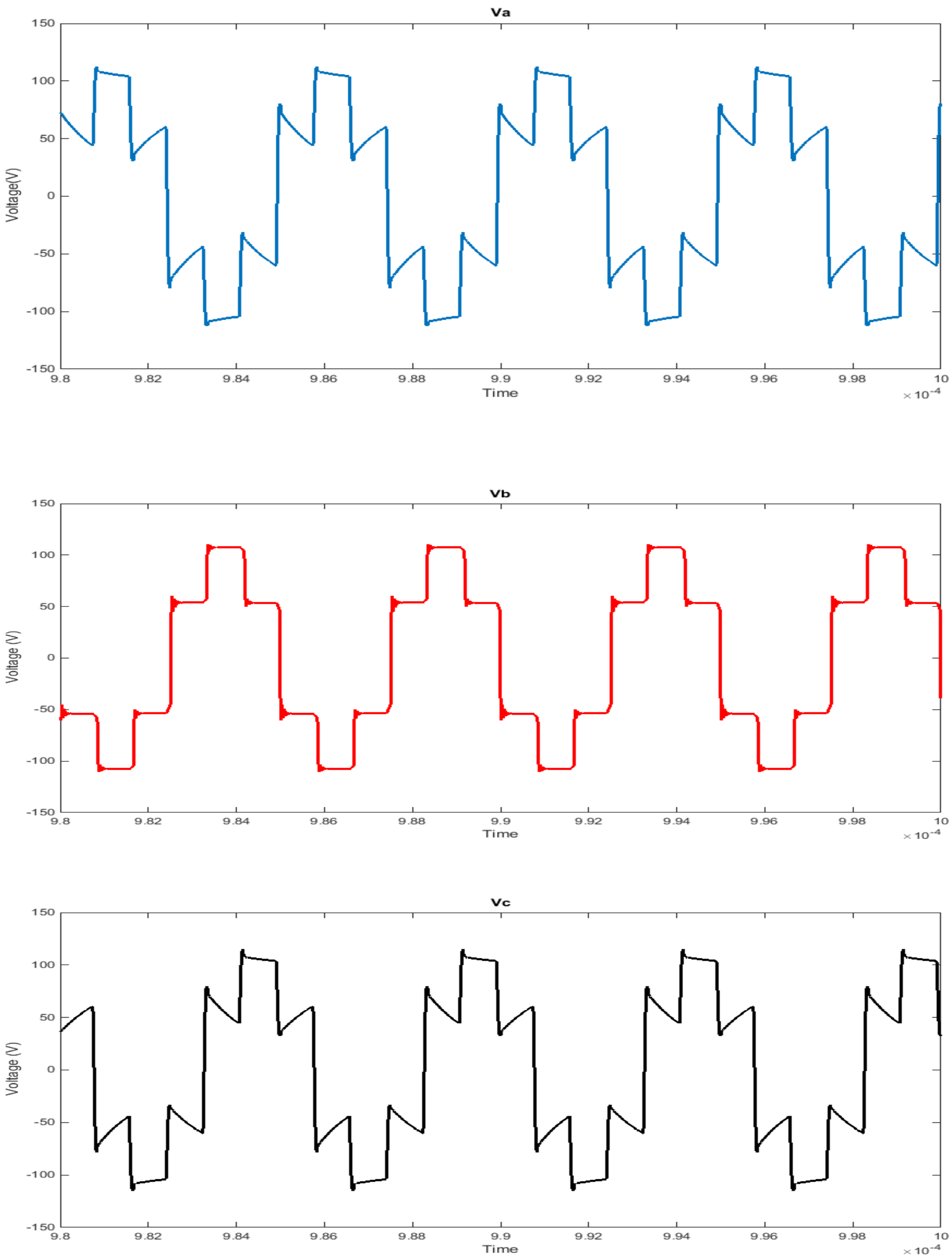

Fig. 4.4. Three-phase voltage at the secondary side of the transformer.

4.2. Bench test set up and results

4.2.1. Rapid Prototyping Environment

Testing the prototype is the next step in finishing this project. Having rapid prototyping environment (RPE) helps in accomplish this. RPE enables the designer to test the prototype using the same controls applied in simulations. Also, it eases the transition between different stages in the prototyping process which in this case changing the phase angle between the two inverters. The RPE consists of the simulation program Matlab/Simulink that is then integrated with an FPGA using HDL coder. The process starts by designing the controls on Simulink using HDL compatible blocks. Then, the FPGA integrates these codes and apply a test signal that matches the simulation’s frequency.

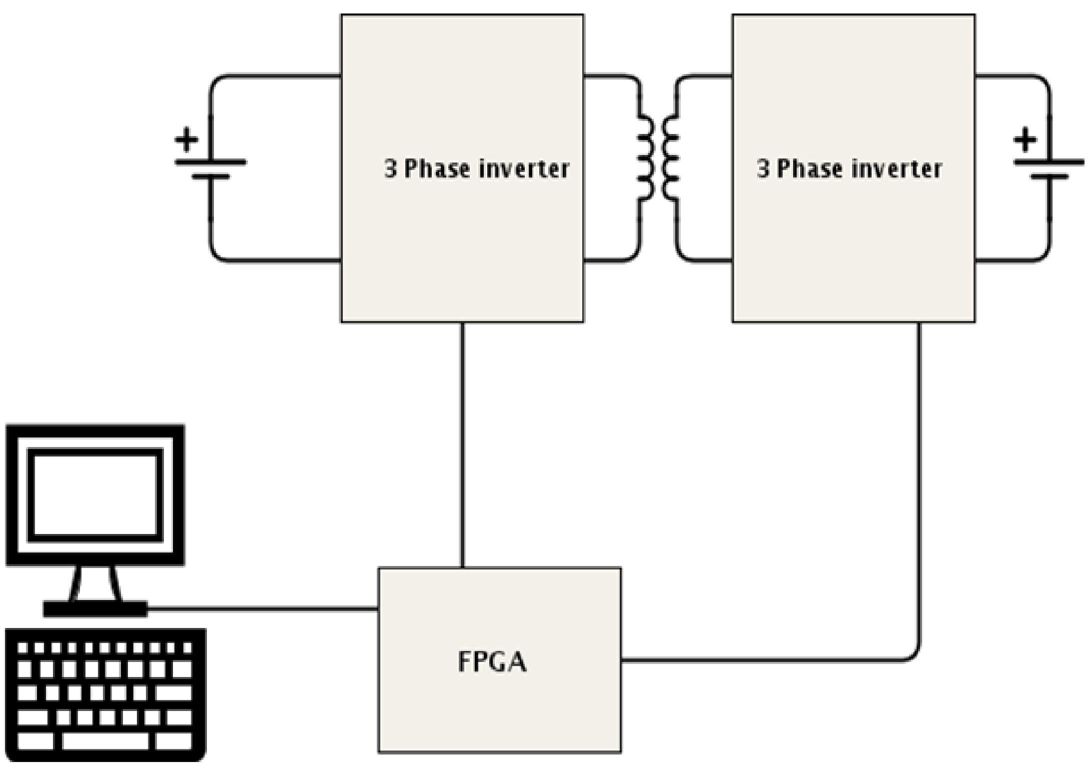

Fig. 4.5. RPE setup.

4.2.2. Experimental Results

This section covers the experimental results of the proposed DAB discussed in the last chapter. Considering that the prototype consists of two three-phase inverters and each inverter consists of three half-bridge boards tide together in parallel, testing each board for functionality before putting the whole system together is a vital step. This helps the ease of trouble shooting and avoids any delays when testing the entire system. Using a simplified version of the designed controls for the DAB, the output voltage of each half-bridge is shown in Fig. 4.4. After that, three half-bridges are tied together in parallel and tested by placing a three-phase resistance load at the output. Fig. 4.5 shows the output voltage of the two three-phase inverters. The phase shift between the bridges can be seen in those figures. The next test consists of one inverter and the high frequency planer transformer. Fig. 4.6 shows the output voltage of the transformer whereas Fig. 4.7 shows the simulation results at the same point in the circuit. In Fig. 4.6, the red line represents the voltage and the green line represents the current measured at division of 10mV/A.

Fig. 4.6. Output voltage (a) board 1 (b) board 2 (c) board 3 (d) board 4 (e) board 5 (f) board 6.

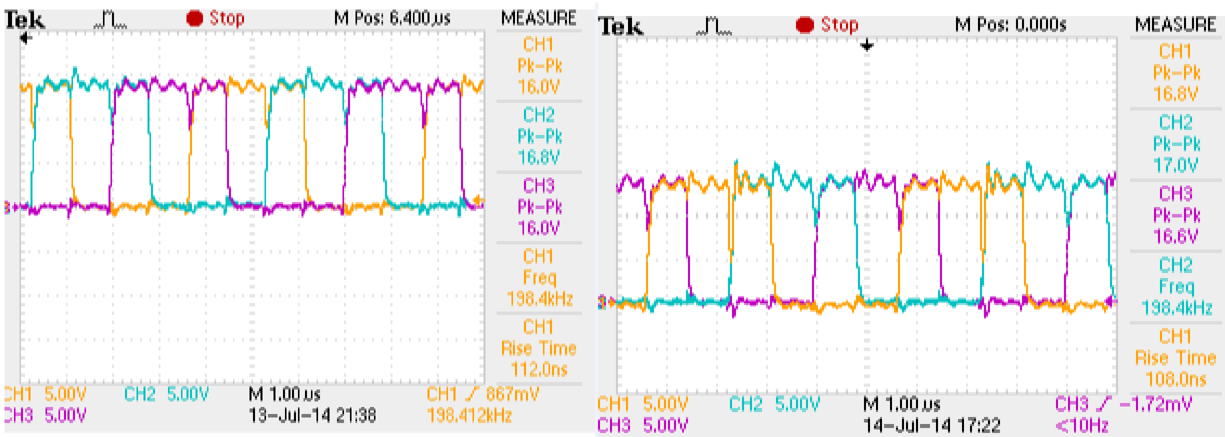

Fig. 4.7. Three-phase output voltage of both inverters.

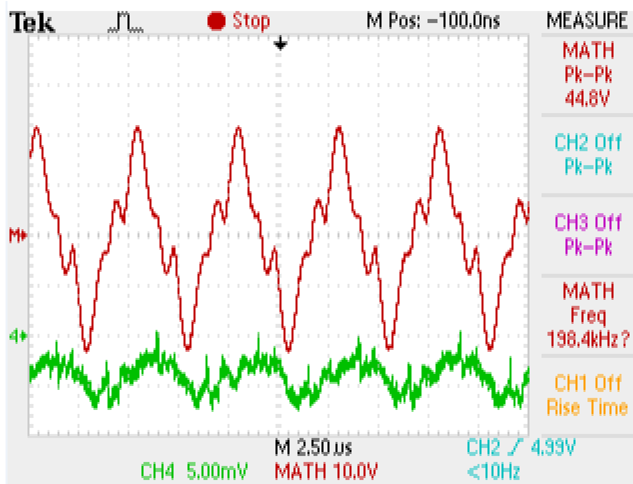

Fig. 4.8. Prototype results of the output voltage (in red) for an inverter with a transformer

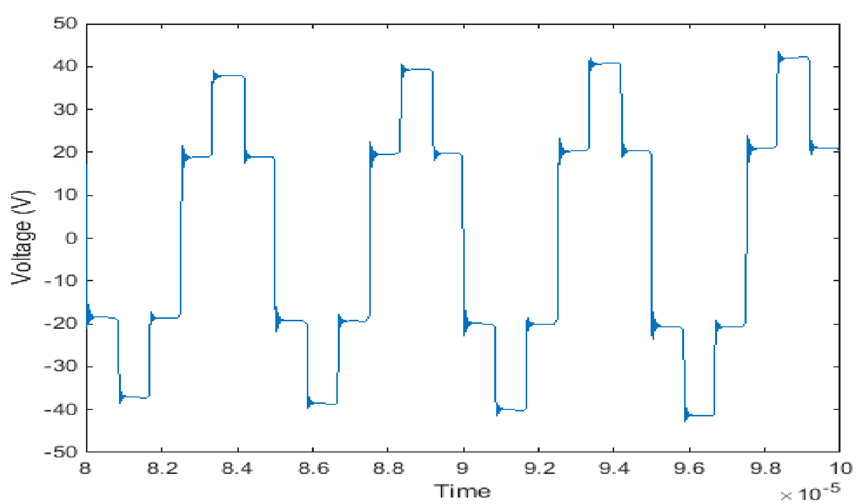

Fig. 4.9. Simulation results of the output voltage for an inverter with a transformer.

4.2.3. Simulation vs experimental results

The prototype has current limitations due to the low thickness of the copper boards. Thus the design is tested at low power level. Therefore, the full potential of the topology will not be achieved in this testing. Fig. 4.8 to Fig. 4.13 shows the output voltage, in yellow, of the DAB while the output current is shown in green. Each figure is taken at a different phase angle. The phase angle control the direction of the power flow and determine when the DAB is in zero power transfer mode or in full power transfer mode. For instance, when the phase angle is zero, both inverters are working in phase with each other. These graphs show the voltage and current are increasing as the phase angle increases until it reaches 90 degrees the values start to drop down. Table 4.3 illustrates the prototype behavior from zero degrees phase shift until 180 degrees. The simulation also shares the same behavior despite the huge difference in the values between the simulation and the prototype. Evidently Table 4.4 shows the power at the output DC bus increase until a phase angle of 75 degrees then it starts to go down as the phase angle increases.

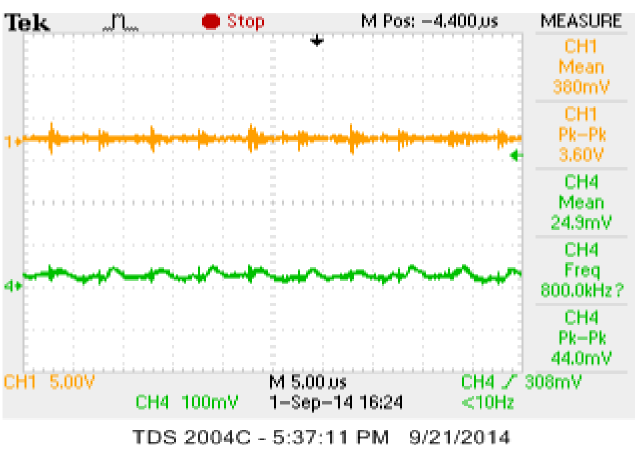

Fig. 4.10. Experiment result at Zero phase angle phase angle.

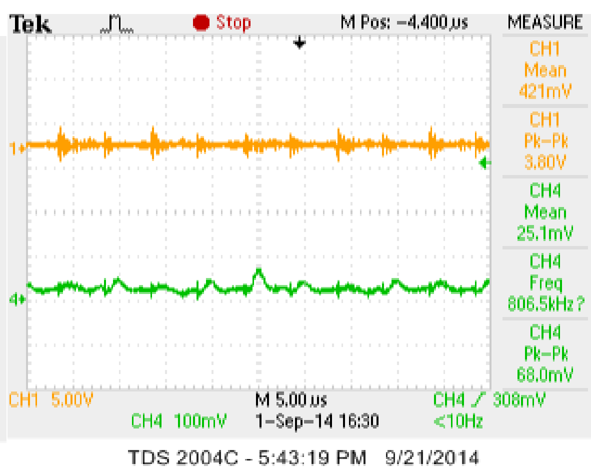

Fig. 4.11. Experiment result at 15 degrees.

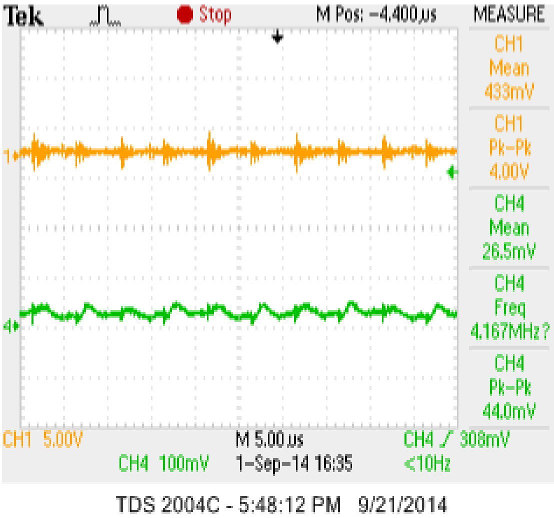

Fig. 4.12. Results at 30 degrees.

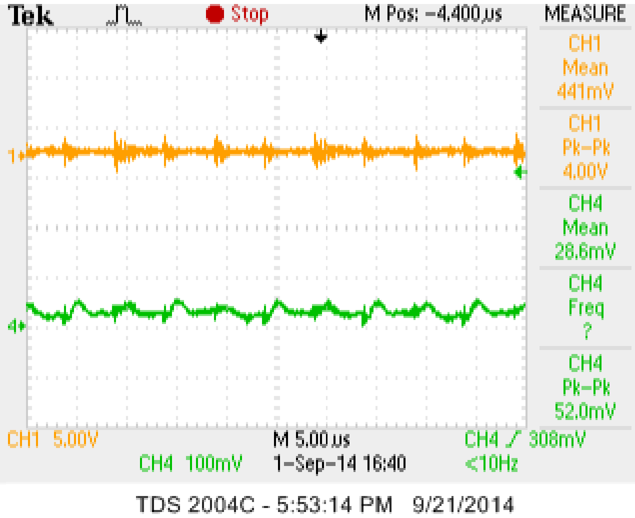

Fig. 4.13. Results at 45 degrees.

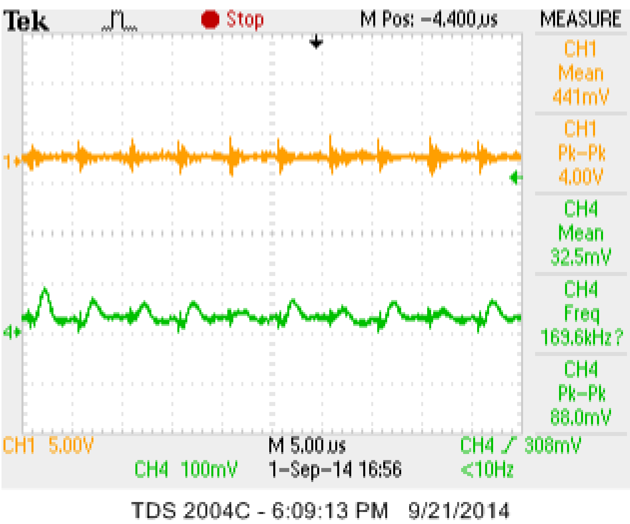

Fig. 4.14. Results at 90 degrees.

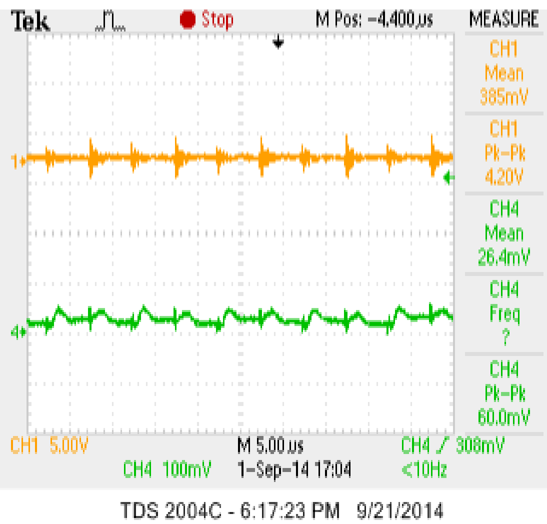

Fig 4.15. Results at 105 degrees.

Table. 4.3. Experimental results of the prototype.

Prototype results			
Phase angle	Voltage(mV)	Current(A)	Power transfer(mW)
0	380	0.249	94.62
15	421	0.251	105.671
30	433	0.265	114.745
45	441	0.286	126.126
60	475	0.321	152.475
75	461	0.335	154.435
90	441	0.325	143.325
105	385	0.264	101.64
120	381	0.307	116.967
135	361	0.278	100.358
150	338	0.267	90.246
165	300	0.242	72.6
180	271	0.241	65.311

Table. 4.4. Experimental results of the simulations.

Simulation results			
Phase Angle	Voltage(mV)	Current(A)	Power transfer(mW)
0	1.03	0.46	474
15	1.47	0.6685	983
30	1.799	0.8177	1471
45	2.002	0.9099	1822
60	2.132	0.9689	2066
75	2.129	0.9679	2061
90	1.957	0.8896	1741
105	1.663	0.7557	1257
120	1.254	0.5701	715
135	0.717	0.3259	234
150	0.21	0.0955	20
165	-0.3559	-0.1636	58
180	-0.888	-0.4039	359

4.2.4. Losses

There are different types of power losses that can be seen in DAB. The switching losses of the MOSFET, the PCB copper losses, transformer losses and there is always the conduction loss. Since SiC MOSFET and the planer transformer both switches at very high frequency, the losses are high despite the fact that a system with high frequency has less power quality issues. As mentioned in chapter 2, the output power is found to be less when the simulation topology is using the 200 KHz controls. The MOSFET also produce conduction loss since it has a forward voltage in the current conduction path. Looking at the three-phase DAB topology in Fig. 4.14 and assuming that switches Q1, Q4, and Q5 are ON, the total resistance can be graphed as shown in Fig. 4.15. Where,

- T: copper trace on half-bridge the PCB
- R_{on}: turn on resistance for the SiC MOSFET
- R_{wire}: The wires connecting the bridge to the transformer board.
- R_L: Leakage resistance of the transformer
- XT: copper trace on the transformer PCB

The PCB traces resistance is calculated using the trace length and width and some of the other resistances are found in the datasheet for each part. Table 4.5 shows the values of the each resistance. The total resistance is then calculated and the total loss, when these switches are on, is found to be 0.2V for each half-bridge board.

Table. 4.5. Resistance values.

T1	3.74 mΩ
T2	4.94 mΩ
T3	5.18 mΩ
T4	3.657 mΩ
R_{on}	290 mΩ
R_{wire1}	0.1 Ω
XT1	4.01 mΩ
R_L	48.8 mΩ
XT2	4.37 mΩ
$R_{wire1-2}$	0.1 Ω

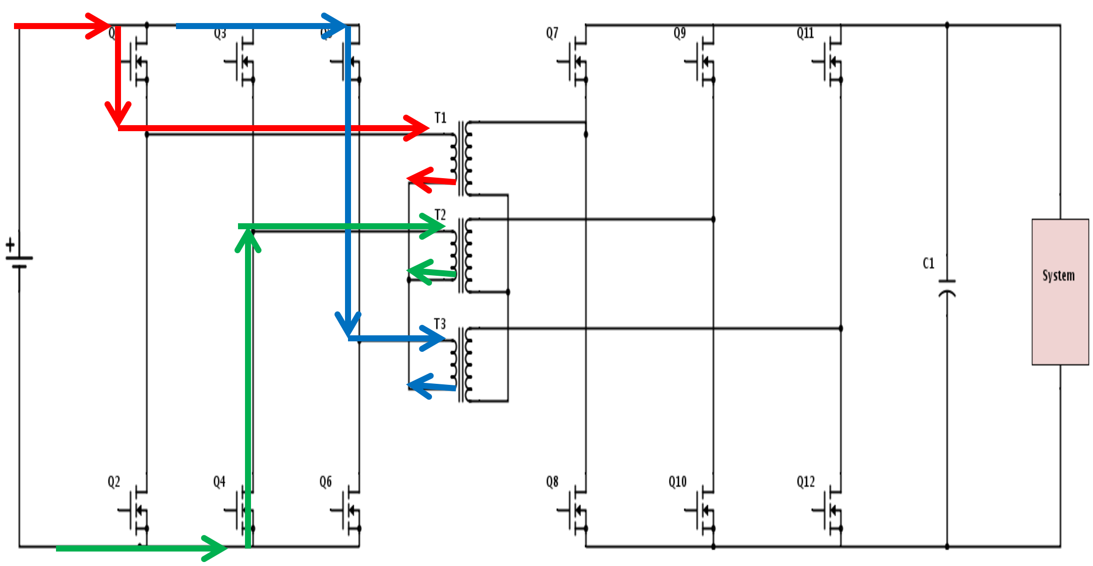

Fig. 4.16. Total resistance when switches Q1, Q4, and Q5 are ON.

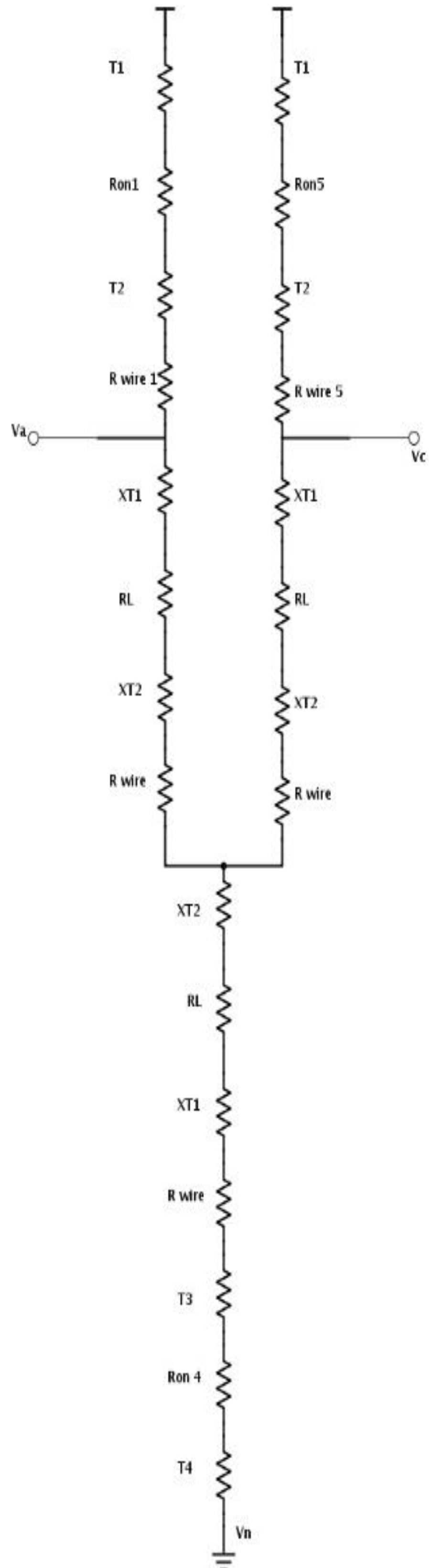

Fig. 4.17. Total resistance when switches Q1, Q4, and Q5 are ON.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

The simulation and implementation of a three-phase bidirectional dc-dc dual active bridge converter using SiC switches has been presented. Switching converters may have been around for a long time but their applications were limited due to the high costs of the switching devices. Many studies have targeted the effects of changing the materials that goes into making the semiconductor devices to improve the performance of power electronic systems. Recently, the use of SiC material to fabricate power semiconductor devices have shown more interest than conventional silicon-based devices due to its promising abilities of fast switching, operate at high voltages and has low losses. Many power applications requires bidirectional power flow which is one of the features of that the DAB converters has. High power density, isolation, and low component stress are other features that make the DAB topologies highly suitable for a variety of applications. The three-phase topology may not as popular as single-phase but can be more beneficial. Three-phase DAB show higher power destiny than single-phase due to the size of the transformer.

The process of developing the simulation and the controls of the DAB are discussed in details in Chapter 2. The design of the PCB prototype and the components selection are covered in Chapter 3. Chapter 4 reviews the experimental results. The controls designed for the simulation is integrated through the FPGA to the test the prototype. Results show that the prototype shares the same behavior as the simulation circuit. The results do not match exactly considering the different kind of losses that is present. In the circuit simulations, the switches are considered to be ideal since the control signal is going directly to the gate. On the other hand, the signal going to the MOSFET in the prototype has to go through passive devices which affect the output results. Overall, this work illustrates the benefits of the proposed three-phase DAB circuit with SiC devices.

5.2 Future work

Since the proposed topology is tested at low level, the next step is to alter the design that it could be tested at high voltage level. This modification will not only increase the output voltage level but will also reduce the losses. For the PCB, the thickness of the copper and the traces that carry power should be increased. This topology uses 1 oz/ft^2 copper board in which can handle a current of 2.3 A. in order to operate up to the maximum current of the power MOSFET, the copper thickness is at best when the copper thickness is 5 oz/ft^2. Having a high copper thickness will affect the width and length of the traces. Table 5.1 shows the calculations of the trace width and length. As can be seen the shorter the traces the less the resistance which lead to less power loss.

Table. 5.1. Calculation of trace width.

5 oz/ft		1 oz/ft	
Current	17 Amps	Current	17 Amps
Thickness	5 oz/ft^2	Thickness	1 oz/ft^2
Required trace width	2.99mm=0.0.117 inch	Required trace width	15mm=0.59 inch
Resistance	0.000857 Ohm	Resistance	0.000857 Ohm
Voltage drop	0.0146 Volts	Voltage drop	0.0146 Volts
Power Loss	0.248 Watts	Power Loss	0.248 Watts
Trace Length	1 inch	Trace Length	1 inch

Another way to reduce losses and increase voltage level is to make a power module of the design presented. Industries are already trying to perfect this. Cree made a three-phase module with SiC MOSFET that has a voltage ratting of 1.2 KV [41]. SiC capabilities of operating at high temperature and high frequency enables the industry to build smaller, lighter power modules that is more efficient than silicon-based material. Even though SiC materials have higher cost than Si materials, it makes any system more compact and less costly [42]. Also, research on high-power passive component is suggested to help in reducing the losses. Another step that could be taken would be to design a closed-loop control such that feedback is used to improve the performance of the three-phase DAB.

Bibliography

[1] S. Ang, *Power-Switching Converters*. CRC Press, 1995.

[2] Y.-S. Lee and Y.-Y. Chiu, “Zero-current-switching switched-capacitor bidirectional DC–DC converter,” *IEE Proceedings - Electric Power Applications*, vol. 152, no. 6, p. 1525, 2005.

[3] F. Caricchi, F. Crescimbini, F. Giulii Capponi, and L. Solero, “Study of bi-directional buck-boost converter topologies for application in electrical vehicle motor drives,” in *Applied Power Electronics Conference and Exposition, 1998. APEC’98. Conference Proceedings 1998., Thirteenth Annual*, 1998, vol. 1, pp. 287–293.

[4] H. Li, D. Liu, F. Peng, and G.-J. Su, “Small signal analysis of a dual half bridge isolated ZVS bi-directional DC-DC converter for electrical vehicle applications,” in *Power Electronics Specialists Conference, 2005. PESC’05. IEEE 36th*, 2005, pp. 2777–2782.

[5] A. A. Aboulnaga and A. Emadi, “Performance evaluation of the isolated bidirectional Cuk converter with integrated magnetics,” in *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, 2004, vol. 2, pp. 1557–1562.

[6] M. N. Kheraluwala, R. W. Gascoigne, D. M. Divan, and E. D. Baumann, “Performance characterization of a high-power dual active bridge dc-to-dc converter,” *Industry Applications, IEEE Transactions on*, vol. 28, no. 6, pp. 1294–1301, 1992.

[7] I.-D. Kim, S.-H. Paeng, J.-W. Ahn, E.-C. Nho, and J.-S. Ko, “New bidirectional ZVS PWM sepic/zeta DC-DC converter,” in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, 2007, pp. 555–560.

[8] D. Aggeler, J. Biela, S. Inoue, H. Akagi, and J. W. Kolar, “Bi-directional isolated dc-dc converter for next-generation power distribution-comparison of converters using Si and SiC devices,” in *Power Conversion Conference-Nagoya, 2007. PCC’07*, 2007, pp. 510–517.

[9] K. Yamamoto, E. Hiraki, T. Tanaka, M. Nakaoka, and T. Mishima, “Bidirectional DC-DC converter with full-bridge/push-pull circuit for automobile electric power systems,” in *Power Electronics Specialists Conference, 2006. PESC’06. 37th IEEE*, 2006, pp. 1–5.

[10] O. Garcia, P. Zumel, A. de Castro, and A. Cobos, “Automotive DC-DC bidirectional converter made with many interleaved buck stages,” *IEEE Transactions on Power Electronics*, vol. 21, no. 3, pp. 578–586, May 2006.

[11] R. Li, A. Pottharst, N. Frohleke, and J. Bocker, “Analysis and design of improved isolated full-bridge bidirectional DC-DC converter,” in *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, 2004, vol. 1, pp. 521–526.

[12] F. A. Himmelstoss, “Analysis and comparison of half-bridge bidirectional DC-DC converters,” in *Power Electronics Specialists Conference, PESC’94 Record., 25th Annual IEEE*, 1994, pp. 922–928.

[13] C. Qiao and K. M. Smedley, “An isolated full bridge boost converter with active soft-switching,” in *Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual*, 2001, vol. 2, pp. 896–903.

[14] Z. Salam, M. Z. Ramli, L. S. Toh, and C. L. Nge, “An isolated bidirectional inverter using high frequency center-tapped transformer,” in *Power Electronics, Machines and Drives, 2004.(PEMD 2004). Second International Conference on (Conf. Publ. No. 498)*, 2004, vol. 2, pp. 644–649.

[15] E. Hiraki, K. Yamamoto, T. Tanaka, T. Mishima, and M. Nakaoka, “An isolated bidirectional dc-dc converter based super-capacitor interface for automobile electric power systems,” in *Power Electronics and Motion Control Conference, 2006. EPE-PEMC 2006. 12th International*, 2006, pp. 722–727.

[16] M. Santhi and R. Rajaram, “A ZVCS lc-resonant push-pull power converter circuit for battery-fuel cell hybrid systems,” in *Electric and Hybrid Vehicles, 2006. ICEHV’06. IEEE Conference on*, 2006, pp. 1–6.

[17] K.-Y. Lee and Y.-S. Lai, “A novel magnetic-less bi-directional DC-DC converter,” in *Industrial Electronics Society, 2004. IECON 2004. 30th Annual Conference of IEEE*, 2004, vol. 2, pp. 1014–1017.

[18] P. Jose and N. Mohan, “A novel bidirectional DC-DC converter with ZVS and interleaving for dual voltage systems in automobiles,” in *Industry Applications Conference, 2002. 37th IAS Annual Meeting. Conference Record of the*, 2002, vol. 2, pp. 1311–1314.

[19] F. Z. Peng, H. Li, G.-J. Su, and J. S. Lawler, “A New ZVS Bidirectional DC-DC Converter for Fuel Cell and Battery Application,” *IEEE Transactions on Power Electronics*, vol. 19, no. 1, pp. 54–65, Jan. 2004.

[20] M. Cacciato, F. Caricchi, F. Giuhlii, and E. Santini, “A critical evaluation and design of bi-directional DC/DC converters for super-capacitors interfacing in fuel cell applications,” in *Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE*, 2004, vol. 2, pp. 1127–1133.

[21] F. Krismer, J. Biela, and J. W. Kolar, “A comparative evaluation of isolated bi-directional DC/DC converters with wide input and output voltage range,” in *Industry Applications*

Conference, 2005. Fourtieth IAS Annual Meeting. Conference Record of the 2005, 2005, vol. 1, pp. 599–606.

[22] S. Inoue and H. Akagi, “A Bidirectional DC-DC Converter for an Energy Storage System With Galvanic Isolation,” *IEEE Transactions on Power Electronics*, vol. 22, no. 6, pp. 2299–2306, Nov. 2007.

[23] G. K. Schoneman, “500-W zero-voltage-switched full-bridge two-quadrant power modulator,” in *Applied Power Electronics Conference and Exposition, 1993. APEC’93. Conference Proceedings 1993., Eighth Annual*, 1993, pp. 700–706.

[24] N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*. John Wiley & Sons, 2003.

[25] H. R. Karshenas, H. Daneshpajooh, A. Safaee, P. Jain, and A. Bakhshai, “Bidirectional DC-DC Converters for Energy Storage Systems,” *Energy Storage in the Emerging Era of Smart Grids*, p. 18, 2011.

[26] J.-S. Lai and D. J. Nelson, “Energy Management Power Converters in Hybrid Electric and Fuel Cell Vehicles,” *Proceedings of the IEEE*, vol. 95, no. 4, pp. 766–777, Apr. 2007.

[27] Z. Xuan, H. Shenghua, and N. Guoyun, “A Three-phase Dual Active Bridge Bidirectional ZVS DC/DC Converter,” Phys. Procedia, vol. 24, pp. 139–148, Jan. 2012.

[28] R. W. De Doncker, D. M. Divan, and M. H. Kheraluwala, “A three-phase soft-switched high-power-density DC/DC converter for high-power applications,” Ind. Appl. Ieee Trans., vol. 27, no. 1, pp. 63–73, 1991.

[29] M. N. Kheraluwala, R. W. Gascoigne, D. M. Divan, and E. D. Baumann, “Performance characterization of a high-power dual active bridge dc-to-dc converter,” Ind. Appl. Ieee Trans., vol. 28, no. 6, pp. 1294–1301, 1992.

[30] D. Segaran, D. G. Holmes, and B. P. McGrath, “Comparative analysis of single and three-phase dual active bridge bidirectional DC-DC converters,” in Power Engineering Conference, 2008. AUPEC’08. Australasian Universities, 2008, pp. 1–6.

[31] N. Soltau, H. A. B. Siddique, and R. W. De Doncker, “Comprehensive modeling and control strategies for a three-phase dual-active bridge,” in Renewable Energy Research and Applications (ICRERA), 2012 International Conference on, 2012, pp. 1–6.

[32] R. W. De Doncker, D. M. Divan, and M. H. Kheraluwala, “A three-phase soft-switched high-power-density DC/DC converter for high-power applications,” *Industry Applications, IEEE Transactions on*, vol. 27, no. 1, pp. 63–73, 1991.

[33] R. Morrison and M. G. Egan, “A new power-factor-corrected single-transformer UPS design,” *Industry Applications, IEEE Transactions on*, vol. 36, no. 1, pp. 171–179, 2000.

[34] EIA. (2012). *U.S. Energy Information Administration*, Form EIA-826, Monthly Electric Sales and Revenue Report with State Distributions Report. Retrieved Oct 07, 2012 http://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_5_6_a

[35] S. Inoue and H. Akagi, "A Bidirectional Isolated DC-DC Converter as a Core Circuit of the Next-Generation Medium-Voltage Power Conversion System," *IEEE Transactions on Power Electronics*, vol. 22, no. 2, pp. 535–542, Mar. 2007.

[36] T. R. McNutt, A. R. Hefner, H. A. Mantooth, D. Berning, and S.-H. Ryu, "Silicon Carbide Power MOSFET Model and Parameter Extraction Sequence," *IEEE Transactions on Power Electronics*, vol. 22, no. 2, pp. 353–363, Mar. 2007.

[37] IXYS," *9 Ampere Low-Side Ultrafast MOSFET Driver*" DS99054B datasheet, Aug,2004.

[38] Avago Technologies," *High CMR Intelligent Power Module and Gate Drive Interface Optocoupler*" AV01-0193EN datasheet, June,2007.

[39] W. G. Hurley and W. H. Wolfle, *Transformers And Inductors For Power Electronics Theory, Design, and Applications*. John Wiley & Sons Ltd., 2013

[40] Y. Ling, "On current carrying capacities of PCB traces," in *Electronic Components and Technology Conference, 2002. Proceedings. 52nd*, 2002, pp. 1683–1693.

[41] CREE," *1.2kV, 50A Silicon Carbide Six-Pack (Three Phase) Module"* CCS050M12CM2 datasheet, Rev. B, 2013.

[42] K. Hamada, "SiC device and power module technologies for environmentally friendly vehicles," in *Integrated Power Electronics Systems (CIPS), 2012 7th International Conference on*, 2012, pp. 1–6.

Appendix

```
-- -------------------------------------------------------------
--
-- File Name:
hdl_prj\hdlsrc\Phase_Shift_Control_200kHz_Open_Loop\Phase_Shift_Control_200kHz_Open_
Loop.vhd
-- Created: 2014-12-02 11:35:00
--
-- Generated by MATLAB 8.4 and HDL Coder 3.5
--
--
-- -------------------------------------------------------------
-- Rate and Clocking Details
-- -------------------------------------------------------------
-- Model base rate: 2e-08
-- Target subsystem base rate: 2e-08
--
--
-- Clock Enable  Sample Time
-- -------------------------------------------------------------
-- ce_out        2e-08
-- -------------------------------------------------------------
--
--
-- Output Signal                 Clock Enable  Sample Time
-- -------------------------------------------------------------
-- S1                            ce_out        2e-08
-- S2                            ce_out        2e-08
-- S1_1                          ce_out        2e-08
-- S2_1                          ce_out        2e-08
-- S3                            ce_out        2e-08
-- S4                            ce_out        2e-08
-- S3_1                          ce_out        2e-08
-- S4_1                          ce_out        2e-08
-- S6                            ce_out        2e-08
-- S5                            ce_out        2e-08
-- S6_1                          ce_out        2e-08
-- S5_1                          ce_out        2e-08
-- -------------------------------------------------------------
--
-- -------------------------------------------------------------


-- -------------------------------------------------------------
```

```
--
-- Module: Phase_Shift_Control_200kHz_Open_Loop
-- Source Path: Phase_Shift_Control_200kHz_Open_Loop
-- Hierarchy Level: 0
--
-- -------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY Phase_Shift_Control_200kHz_Open_Loop IS
  PORT( clk                             :   IN    std_logic;
        reset                           :   IN    std_logic;
        clk_enable                      :   IN    std_logic;
        ce_out                          :   OUT   std_logic;
        S1                              :   OUT   std_logic;
        S2                              :   OUT   std_logic;
        S1_1                            :   OUT   std_logic;
        S2_1                            :   OUT   std_logic;
        S3                              :   OUT   std_logic;
        S4                              :   OUT   std_logic;
        S3_1                            :   OUT   std_logic;
        S4_1                            :   OUT   std_logic;
        S6                              :   OUT   std_logic;
        S5                              :   OUT   std_logic;
        S6_1                            :   OUT   std_logic;
        S5_1                            :   OUT   std_logic
        );
END Phase_Shift_Control_200kHz_Open_Loop;


ARCHITECTURE rtl OF Phase_Shift_Control_200kHz_Open_Loop IS

  -- Signals
  SIGNAL enb                              : std_logic;
  SIGNAL HDL_Counter4_out1                : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL HDL_Counter4_step_reg            : unsigned(15 DOWNTO 0);  -- ufix16
  SIGNAL HDL_Counter4_stepreg             : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL HDL_Counter4_count               : unsigned(15 DOWNTO 0);  -- ufix16
  SIGNAL Constant17_out1                  : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL phi_out1                         : signed(15 DOWNTO 0);  -- int16
  SIGNAL switch_compare_1                 : std_logic;
  SIGNAL Gain_in0                         : signed(16 DOWNTO 0);  -- sfix17
  SIGNAL Gain_out1                        : signed(31 DOWNTO 0);  -- sfix32_En15
  SIGNAL Zero_out1                        : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL Zero_out1_dtc                    : signed(31 DOWNTO 0);  -- sfix32_En15
```

```
SIGNAL Switch4_out1                    : signed(31 DOWNTO 0);  -- sfix32_En15
SIGNAL Subtract20_add_cast             : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract20_add_cast_1           : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract20_out1                 : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator15_1_cast    : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator15_relop1    : std_logic;
SIGNAL Constant20_out1                 : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract19_add_cast             : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract19_add_cast_1           : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract19_out1                 : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator16_1_cast    : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator16_relop1    : std_logic;
SIGNAL Logical_Operator7_out1          : std_logic;
SIGNAL Constant18_out1                 : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract17_add_cast             : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract17_add_cast_1           : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract17_out1                 : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator12_1_cast    : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator12_relop1    : std_logic;
SIGNAL Constant1_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract1_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract1_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract1_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator4_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator4_relop1     : std_logic;
SIGNAL Logical_Operator6_out1          : std_logic;
SIGNAL Constant7_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract4_sub_cast              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract4_sub_cast_1            : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract4_sub_temp              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract4_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator6_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator6_relop1     : std_logic;
SIGNAL Logical_Operator3_out1          : std_logic;
SIGNAL Constant3_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL switch_compare_1_1              : std_logic;
SIGNAL phi_out1_dtc                    : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Switch5_out1                    : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract6_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator2_relop1     : std_logic;
SIGNAL Constant5_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract5_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator3_relop1     : std_logic;
SIGNAL Logical_Operator4_out1          : std_logic;
SIGNAL Constant4_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract3_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
```

```
SIGNAL Relational_Operator1_relop1      : std_logic;
SIGNAL Constant2_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract2_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator5_relop1      : std_logic;
SIGNAL Logical_Operator2_out1           : std_logic;
SIGNAL Constant6_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract7_sub_cast              : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract7_sub_cast_1            : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract7_sub_temp              : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract7_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator7_1_cast      : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator7_relop1      : std_logic;
SIGNAL Logical_Operator1_out1           : std_logic;
SIGNAL Constant9_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract12_add_cast             : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract12_add_cast_1           : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract12_out1                 : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator10_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator10_relop1     : std_logic;
SIGNAL Constant12_out1                 : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract10_add_cast             : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract10_add_cast_1           : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract10_out1                 : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator11_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator11_relop1     : std_logic;
SIGNAL Logical_Operator12_out1          : std_logic;
SIGNAL Constant21_out1                 : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract21_sub_cast             : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract21_sub_cast_1           : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract21_sub_temp             : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract21_out1                 : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator21_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator21_relop1     : std_logic;
SIGNAL Logical_Operator13_out1          : std_logic;
SIGNAL switch_compare_1_2              : std_logic;
SIGNAL Constant10_out1                 : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract9_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract9_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract9_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator9_1_cast      : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator9_relop1      : std_logic;
SIGNAL Constant8_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract8_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract8_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract8_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator17_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
```

```
SIGNAL Relational_Operator17_relop1     : std_logic;
SIGNAL Logical_Operator11_out1          : std_logic;
SIGNAL Constant19_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract14_sub_cast              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract14_sub_cast_1            : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract14_sub_temp              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract14_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator19_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator19_relop1     : std_logic;
SIGNAL Logical_Operator14_out1          : std_logic;
SIGNAL Switch1_out1                     : std_logic;
SIGNAL Constant22_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract13_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator8_relop1      : std_logic;
SIGNAL Constant13_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract11_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator13_relop1     : std_logic;
SIGNAL Logical_Operator8_out1           : std_logic;
SIGNAL Constant15_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract16_sub_cast              : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract16_sub_cast_1            : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract16_sub_temp              : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract16_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator20_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator20_relop1     : std_logic;
SIGNAL Logical_Operator9_out1           : std_logic;
SIGNAL switch_compare_1_3               : std_logic;
SIGNAL Constant11_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract22_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator22_relop1     : std_logic;
SIGNAL Constant16_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract18_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator14_relop1     : std_logic;
SIGNAL Logical_Operator5_out1           : std_logic;
SIGNAL Constant14_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract15_sub_cast              : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract15_sub_cast_1            : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract15_sub_temp              : signed(17 DOWNTO 0);  -- sfix18
SIGNAL Subtract15_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator18_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator18_relop1     : std_logic;
SIGNAL Logical_Operator10_out1          : std_logic;
SIGNAL Switch2_out1                     : std_logic;
SIGNAL Constant34_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract25_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract25_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
```

```
SIGNAL Subtract25_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator23_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator23_relop1     : std_logic;
SIGNAL Constant25_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract23_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract23_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract23_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator24_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator24_relop1     : std_logic;
SIGNAL Logical_Operator17_out1          : std_logic;
SIGNAL Constant31_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract31_sub_cast              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract31_sub_cast_1            : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract31_sub_temp              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract31_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator31_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator31_relop1     : std_logic;
SIGNAL Logical_Operator18_out1          : std_logic;
SIGNAL switch_compare_1_4               : std_logic;
SIGNAL Constant23_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract34_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract34_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract34_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator34_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator34_relop1     : std_logic;
SIGNAL Constant33_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract33_add_cast              : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract33_add_cast_1            : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Subtract33_out1                  : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator27_1_cast     : signed(32 DOWNTO 0);  -- sfix33_En15
SIGNAL Relational_Operator27_relop1     : std_logic;
SIGNAL Logical_Operator16_out1          : std_logic;
SIGNAL Constant30_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract27_sub_cast              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract27_sub_cast_1            : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract27_sub_temp              : signed(33 DOWNTO 0);  -- sfix34_En15
SIGNAL Subtract27_out1                  : signed(15 DOWNTO 0);  -- int16
SIGNAL Relational_Operator29_1_cast     : signed(16 DOWNTO 0);  -- sfix17
SIGNAL Relational_Operator29_relop1     : std_logic;
SIGNAL Logical_Operator19_out1          : std_logic;
SIGNAL Switch3_out1                     : std_logic;
SIGNAL Constant32_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract26_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
SIGNAL Relational_Operator25_relop1     : std_logic;
SIGNAL Constant26_out1                  : unsigned(15 DOWNTO 0);  -- uint16
SIGNAL Subtract24_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
```

```
  SIGNAL Relational_Operator26_relop1     : std_logic;
  SIGNAL Logical_Operator20_out1          : std_logic;
  SIGNAL Constant28_out1                  : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL Subtract29_sub_cast              : signed(17 DOWNTO 0);  -- sfix18
  SIGNAL Subtract29_sub_cast_1            : signed(17 DOWNTO 0);  -- sfix18
  SIGNAL Subtract29_sub_temp              : signed(17 DOWNTO 0);  -- sfix18
  SIGNAL Subtract29_out1                  : signed(15 DOWNTO 0);  -- int16
  SIGNAL Relational_Operator32_1_cast     : signed(16 DOWNTO 0);  -- sfix17
  SIGNAL Relational_Operator32_relop1     : std_logic;
  SIGNAL Logical_Operator21_out1          : std_logic;
  SIGNAL switch_compare_1_5               : std_logic;
  SIGNAL Constant24_out1                  : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL Subtract32_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
  SIGNAL Relational_Operator33_relop1     : std_logic;
  SIGNAL Constant29_out1                  : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL Subtract30_out1                  : unsigned(16 DOWNTO 0);  -- ufix17
  SIGNAL Relational_Operator28_relop1     : std_logic;
  SIGNAL Logical_Operator15_out1          : std_logic;
  SIGNAL Constant27_out1                  : unsigned(15 DOWNTO 0);  -- uint16
  SIGNAL Subtract28_sub_cast              : signed(17 DOWNTO 0);  -- sfix18
  SIGNAL Subtract28_sub_cast_1            : signed(17 DOWNTO 0);  -- sfix18
  SIGNAL Subtract28_sub_temp              : signed(17 DOWNTO 0);  -- sfix18
  SIGNAL Subtract28_out1                  : signed(15 DOWNTO 0);  -- int16
  SIGNAL Relational_Operator30_1_cast     : signed(16 DOWNTO 0);  -- sfix17
  SIGNAL Relational_Operator30_relop1     : std_logic;
  SIGNAL Logical_Operator22_out1          : std_logic;
  SIGNAL Switch6_out1                     : std_logic;

BEGIN
  enb <= clk_enable;

  -- Count limited, Unsigned Counter
  --  initial value   = 1
  --  step value      = 1
  --  count to value  = 250
  HDL_Counter4_step_process : PROCESS (clk, reset)
  BEGIN
    IF reset = '1' THEN
      HDL_Counter4_step_reg <= to_unsigned(16#0001#, 16);
    ELSIF clk'EVENT AND clk = '1' THEN
      IF enb = '1' THEN
        IF HDL_Counter4_out1 = to_unsigned(16#00F9#, 16) THEN
          HDL_Counter4_step_reg <= to_unsigned(16#FF07#, 16);
        ELSE
          HDL_Counter4_step_reg <= to_unsigned(16#0001#, 16);
        END IF;
```

```
    END IF;
  END IF;
END PROCESS HDL_Counter4_step_process;
HDL_Counter4_stepreg <= HDL_Counter4_step_reg;

HDL_Counter4_process : PROCESS (clk, reset)
BEGIN
  IF reset = '1' THEN
    HDL_Counter4_count <= to_unsigned(16#0001#, 16);
  ELSIF clk'EVENT AND clk = '1' THEN
    IF enb = '1' THEN
      HDL_Counter4_count <= HDL_Counter4_count + HDL_Counter4_stepreg;
    END IF;
  END IF;
END PROCESS HDL_Counter4_process;
HDL_Counter4_out1 <= HDL_Counter4_count;

Constant17_out1 <= to_unsigned(16#0001#, 16);

phi_out1 <= to_signed(16#0000#, 16);


switch_compare_1 <= '1' WHEN phi_out1 >= to_signed(16#0000#, 16) ELSE
    '0';

Gain_in0 <=  - (resize(phi_out1, 17));
Gain_out1 <= Gain_in0 & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0';

Zero_out1 <= to_unsigned(16#0000#, 16);

Zero_out1_dtc <= signed(resize(Zero_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0', 32));


Switch4_out1 <= Gain_out1 WHEN switch_compare_1 = '0' ELSE
    Zero_out1_dtc;

Subtract20_add_cast <= signed(resize(Constant17_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
Subtract20_add_cast_1 <= resize(Switch4_out1, 33);
Subtract20_out1 <= Subtract20_add_cast + Subtract20_add_cast_1;

Relational_Operator15_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
```

```
  Relational_Operator15_relop1 <= '1' WHEN Relational_Operator15_1_cast >= 
Subtract20_out1 ELSE
      '0';

  Constant20_out1 <= to_unsigned(16#0078#, 16);

  Subtract19_add_cast <= signed(resize(Constant20_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' & 
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract19_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract19_out1 <= Subtract19_add_cast + Subtract19_add_cast_1;

  Relational_Operator16_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' & 
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator16_relop1 <= '1' WHEN Relational_Operator16_1_cast <= 
Subtract19_out1 ELSE
      '0';

  Logical_Operator7_out1 <= Relational_Operator15_relop1 AND 
Relational_Operator16_relop1;

  Constant18_out1 <= to_unsigned(16#007D#, 16);

  Subtract17_add_cast <= signed(resize(Constant18_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' & 
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract17_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract17_out1 <= Subtract17_add_cast + Subtract17_add_cast_1;

  Relational_Operator12_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' & 
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator12_relop1 <= '1' WHEN Relational_Operator12_1_cast >= 
Subtract17_out1 ELSE
      '0';

  Constant1_out1 <= to_unsigned(16#00F5#, 16);

  Subtract1_add_cast <= signed(resize(Constant1_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' & 
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract1_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract1_out1 <= Subtract1_add_cast + Subtract1_add_cast_1;

  Relational_Operator4_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' & '0' 
& '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
```

```
  Relational_Operator4_relop1 <= '1' WHEN Relational_Operator4_1_cast <= Subtract1_out1
ELSE
      '0';

  Logical_Operator6_out1 <= Relational_Operator12_relop1 AND Relational_Operator4_relop1;

  Constant7_out1 <= to_unsigned(16#00FA#, 16);

  Subtract4_sub_cast <= resize(Subtract1_out1, 34);
  Subtract4_sub_cast_1 <= signed(resize(Constant7_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 34));
  Subtract4_sub_temp <= Subtract4_sub_cast - Subtract4_sub_cast_1;
  Subtract4_out1 <= Subtract4_sub_temp(30 DOWNTO 15);

  Relational_Operator6_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator6_relop1 <= '1' WHEN Relational_Operator6_1_cast <=
resize(Subtract4_out1, 17) ELSE
      '0';

  Logical_Operator3_out1 <= Logical_Operator6_out1 OR Relational_Operator6_relop1;

  Constant3_out1 <= to_unsigned(16#0001#, 16);


  switch_compare_1_1 <= '1' WHEN phi_out1 >= to_signed(16#0000#, 16) ELSE
      '0';

  phi_out1_dtc <= unsigned(phi_out1);


  Switch5_out1 <= Zero_out1 WHEN switch_compare_1_1 = '0' ELSE
      phi_out1_dtc;

  Subtract6_out1 <= resize(Constant3_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator2_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) >= Subtract6_out1
ELSE
      '0';

  Constant5_out1 <= to_unsigned(16#0078#, 16);

  Subtract5_out1 <= resize(Constant5_out1, 17) + resize(Switch5_out1, 17);
```

```
  Relational_Operator3_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) <= Subtract5_out1
ELSE
      '0';

  Logical_Operator4_out1 <= Relational_Operator2_relop1 AND Relational_Operator3_relop1;

  Constant4_out1 <= to_unsigned(16#007D#, 16);

  Subtract3_out1 <= resize(Constant4_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator1_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) >= Subtract3_out1
ELSE
      '0';

  Constant2_out1 <= to_unsigned(16#00F5#, 16);

  Subtract2_out1 <= resize(Constant2_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator5_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) <= Subtract2_out1
ELSE
      '0';

  Logical_Operator2_out1 <= Relational_Operator1_relop1 AND Relational_Operator5_relop1;

  Constant6_out1 <= to_unsigned(16#00FA#, 16);

  Subtract7_sub_cast <= signed(resize(Subtract2_out1, 18));
  Subtract7_sub_cast_1 <= signed(resize(Constant6_out1, 18));
  Subtract7_sub_temp <= Subtract7_sub_cast - Subtract7_sub_cast_1;
  Subtract7_out1 <= Subtract7_sub_temp(15 DOWNTO 0);

  Relational_Operator7_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator7_relop1 <= '1' WHEN Relational_Operator7_1_cast <=
resize(Subtract7_out1, 17) ELSE
      '0';

  Logical_Operator1_out1 <= Logical_Operator2_out1 OR Relational_Operator7_relop1;

  Constant9_out1 <= to_unsigned(16#0054#, 16);

  Subtract12_add_cast <= signed(resize(Constant9_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract12_add_cast_1 <= resize(Switch4_out1, 33);
```

```
  Subtract12_out1 <= Subtract12_add_cast + Subtract12_add_cast_1;

  Relational_Operator10_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator10_relop1 <= '1' WHEN Relational_Operator10_1_cast >=
Subtract12_out1 ELSE
      '0';

  Constant12_out1 <= to_unsigned(16#00CB#, 16);

  Subtract10_add_cast <= signed(resize(Constant12_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract10_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract10_out1 <= Subtract10_add_cast + Subtract10_add_cast_1;

  Relational_Operator11_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator11_relop1 <= '1' WHEN Relational_Operator11_1_cast <=
Subtract10_out1 ELSE
      '0';

  Logical_Operator12_out1 <= Relational_Operator10_relop1 AND
Relational_Operator11_relop1;

  Constant21_out1 <= to_unsigned(16#00FA#, 16);

  Subtract21_sub_cast <= resize(Subtract10_out1, 34);
  Subtract21_sub_cast_1 <= signed(resize(Constant21_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0'
& '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 34));
  Subtract21_sub_temp <= Subtract21_sub_cast - Subtract21_sub_cast_1;
  Subtract21_out1 <= Subtract21_sub_temp(30 DOWNTO 15);

  Relational_Operator21_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator21_relop1 <= '1' WHEN Relational_Operator21_1_cast <=
resize(Subtract21_out1, 17) ELSE
      '0';

  Logical_Operator13_out1 <= Logical_Operator12_out1 OR Relational_Operator21_relop1;


  switch_compare_1_2 <= '1' WHEN Switch4_out1 >= to_signed(1409024, 32) ELSE
      '0';
```

```
  Constant10_out1 <= to_unsigned(16#004E#, 16);

  Subtract9_add_cast <= signed(resize(Constant10_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract9_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract9_out1 <= Subtract9_add_cast + Subtract9_add_cast_1;

  Relational_Operator9_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' & '0'
& '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator9_relop1 <= '1' WHEN Relational_Operator9_1_cast <= Subtract9_out1
ELSE
      '0';

  Constant8_out1 <= to_unsigned(16#00D0#, 16);

  Subtract8_add_cast <= signed(resize(Constant8_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract8_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract8_out1 <= Subtract8_add_cast + Subtract8_add_cast_1;

  Relational_Operator17_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator17_relop1 <= '1' WHEN Relational_Operator17_1_cast >= Subtract8_out1
ELSE
      '0';

  Logical_Operator11_out1 <= Relational_Operator9_relop1 OR Relational_Operator17_relop1;

  Constant19_out1 <= to_unsigned(16#00FA#, 16);

  Subtract14_sub_cast <= resize(Subtract8_out1, 34);
  Subtract14_sub_cast_1 <= signed(resize(Constant19_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0'
& '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 34));
  Subtract14_sub_temp <= Subtract14_sub_cast - Subtract14_sub_cast_1;
  Subtract14_out1 <= Subtract14_sub_temp(30 DOWNTO 15);

  Relational_Operator19_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator19_relop1 <= '1' WHEN Relational_Operator19_1_cast >=
resize(Subtract14_out1, 17) ELSE
      '0';

  Logical_Operator14_out1 <= Logical_Operator11_out1 AND Relational_Operator19_relop1;
```

```
  Switch1_out1 <= Logical_Operator11_out1 WHEN switch_compare_1_2 = '0' ELSE
      Logical_Operator14_out1;

  Constant22_out1 <= to_unsigned(16#0054#, 16);

  Subtract13_out1 <= resize(Constant22_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator8_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) >=
Subtract13_out1 ELSE
      '0';

  Constant13_out1 <= to_unsigned(16#00CB#, 16);

  Subtract11_out1 <= resize(Constant13_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator13_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) <=
Subtract11_out1 ELSE
      '0';

  Logical_Operator8_out1 <= Relational_Operator8_relop1 AND Relational_Operator13_relop1;

  Constant15_out1 <= to_unsigned(16#00FA#, 16);

  Subtract16_sub_cast <= signed(resize(Subtract11_out1, 18));
  Subtract16_sub_cast_1 <= signed(resize(Constant15_out1, 18));
  Subtract16_sub_temp <= Subtract16_sub_cast - Subtract16_sub_cast_1;
  Subtract16_out1 <= Subtract16_sub_temp(15 DOWNTO 0);

  Relational_Operator20_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator20_relop1 <= '1' WHEN Relational_Operator20_1_cast <=
resize(Subtract16_out1, 17) ELSE
      '0';

  Logical_Operator9_out1 <= Logical_Operator8_out1 OR Relational_Operator20_relop1;


  switch_compare_1_3 <= '1' WHEN Switch5_out1 >= to_unsigned(16#002B#, 16) ELSE
      '0';

  Constant11_out1 <= to_unsigned(16#004E#, 16);

  Subtract22_out1 <= resize(Constant11_out1, 17) + resize(Switch5_out1, 17);
```

```
  Relational_Operator22_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) <=
Subtract22_out1 ELSE
      '0';

  Constant16_out1 <= to_unsigned(16#00D0#, 16);

  Subtract18_out1 <= resize(Constant16_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator14_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) >=
Subtract18_out1 ELSE
      '0';

  Logical_Operator5_out1 <= Relational_Operator22_relop1 OR Relational_Operator14_relop1;

  Constant14_out1 <= to_unsigned(16#00FA#, 16);

  Subtract15_sub_cast <= signed(resize(Subtract18_out1, 18));
  Subtract15_sub_cast_1 <= signed(resize(Constant14_out1, 18));
  Subtract15_sub_temp <= Subtract15_sub_cast - Subtract15_sub_cast_1;
  Subtract15_out1 <= Subtract15_sub_temp(15 DOWNTO 0);

  Relational_Operator18_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator18_relop1 <= '1' WHEN Relational_Operator18_1_cast >=
resize(Subtract15_out1, 17) ELSE
      '0';

  Logical_Operator10_out1 <= Logical_Operator5_out1 AND Relational_Operator18_relop1;


  Switch2_out1 <= Logical_Operator5_out1 WHEN switch_compare_1_3 = '0' ELSE
      Logical_Operator10_out1;

  Constant34_out1 <= to_unsigned(16#002A#, 16);

  Subtract25_add_cast <= signed(resize(Constant34_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract25_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract25_out1 <= Subtract25_add_cast + Subtract25_add_cast_1;

  Relational_Operator23_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
```

```
  Relational_Operator23_relop1 <= '1' WHEN Relational_Operator23_1_cast >=
Subtract25_out1 ELSE
      '0';

  Constant25_out1 <= to_unsigned(16#00A2#, 16);

  Subtract23_add_cast <= signed(resize(Constant25_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract23_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract23_out1 <= Subtract23_add_cast + Subtract23_add_cast_1;

  Relational_Operator24_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator24_relop1 <= '1' WHEN Relational_Operator24_1_cast <=
Subtract23_out1 ELSE
      '0';

  Logical_Operator17_out1 <= Relational_Operator23_relop1 AND
Relational_Operator24_relop1;

  Constant31_out1 <= to_unsigned(16#00FA#, 16);

  Subtract31_sub_cast <= resize(Subtract23_out1, 34);
  Subtract31_sub_cast_1 <= signed(resize(Constant31_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0'
& '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 34));
  Subtract31_sub_temp <= Subtract31_sub_cast - Subtract31_sub_cast_1;
  Subtract31_out1 <= Subtract31_sub_temp(30 DOWNTO 15);

  Relational_Operator31_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator31_relop1 <= '1' WHEN Relational_Operator31_1_cast <=
resize(Subtract31_out1, 17) ELSE
      '0';

  Logical_Operator18_out1 <= Logical_Operator17_out1 OR Relational_Operator31_relop1;


  switch_compare_1_4 <= '1' WHEN Switch4_out1 >= to_signed(2719744, 32) ELSE
      '0';

  Constant23_out1 <= to_unsigned(16#0025#, 16);

  Subtract34_add_cast <= signed(resize(Constant23_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract34_add_cast_1 <= resize(Switch4_out1, 33);
```

```
  Subtract34_out1 <= Subtract34_add_cast + Subtract34_add_cast_1;

  Relational_Operator34_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator34_relop1 <= '1' WHEN Relational_Operator34_1_cast <=
Subtract34_out1 ELSE
      '0';

  Constant33_out1 <= to_unsigned(16#00A8#, 16);

  Subtract33_add_cast <= signed(resize(Constant33_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));
  Subtract33_add_cast_1 <= resize(Switch4_out1, 33);
  Subtract33_out1 <= Subtract33_add_cast + Subtract33_add_cast_1;

  Relational_Operator27_1_cast <= signed(resize(HDL_Counter4_out1 & '0' & '0' & '0' & '0' &
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 33));

  Relational_Operator27_relop1 <= '1' WHEN Relational_Operator27_1_cast >=
Subtract33_out1 ELSE
      '0';

  Logical_Operator16_out1 <= Relational_Operator34_relop1 OR
Relational_Operator27_relop1;

  Constant30_out1 <= to_unsigned(16#00FA#, 16);

  Subtract27_sub_cast <= resize(Subtract33_out1, 34);
  Subtract27_sub_cast_1 <= signed(resize(Constant30_out1 & '0' & '0' & '0' & '0' & '0' & '0' & '0'
& '0' & '0' & '0' & '0' & '0' & '0' & '0' & '0', 34));
  Subtract27_sub_temp <= Subtract27_sub_cast - Subtract27_sub_cast_1;
  Subtract27_out1 <= Subtract27_sub_temp(30 DOWNTO 15);

  Relational_Operator29_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator29_relop1 <= '1' WHEN Relational_Operator29_1_cast >=
resize(Subtract27_out1, 17) ELSE
      '0';

  Logical_Operator19_out1 <= Logical_Operator16_out1 AND Relational_Operator29_relop1;


  Switch3_out1 <= Logical_Operator16_out1 WHEN switch_compare_1_4 = '0' ELSE
      Logical_Operator19_out1;
```

```
  Constant32_out1 <= to_unsigned(16#002A#, 16);

  Subtract26_out1 <= resize(Constant32_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator25_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) >=
Subtract26_out1 ELSE
      '0';

  Constant26_out1 <= to_unsigned(16#00A2#, 16);

  Subtract24_out1 <= resize(Constant26_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator26_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) <=
Subtract24_out1 ELSE
      '0';

  Logical_Operator20_out1 <= Relational_Operator25_relop1 AND
Relational_Operator26_relop1;

  Constant28_out1 <= to_unsigned(16#00FA#, 16);

  Subtract29_sub_cast <= signed(resize(Subtract24_out1, 18));
  Subtract29_sub_cast_1 <= signed(resize(Constant28_out1, 18));
  Subtract29_sub_temp <= Subtract29_sub_cast - Subtract29_sub_cast_1;
  Subtract29_out1 <= Subtract29_sub_temp(15 DOWNTO 0);

  Relational_Operator32_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator32_relop1 <= '1' WHEN Relational_Operator32_1_cast <=
resize(Subtract29_out1, 17) ELSE
      '0';

  Logical_Operator21_out1 <= Logical_Operator20_out1 OR Relational_Operator32_relop1;


  switch_compare_1_5 <= '1' WHEN Switch5_out1 >= to_unsigned(16#0053#, 16) ELSE
      '0';

  Constant24_out1 <= to_unsigned(16#0025#, 16);

  Subtract32_out1 <= resize(Constant24_out1, 17) + resize(Switch5_out1, 17);
```

```
  Relational_Operator33_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) <=
Subtract32_out1 ELSE
      '0';

  Constant29_out1 <= to_unsigned(16#00A8#, 16);

  Subtract30_out1 <= resize(Constant29_out1, 17) + resize(Switch5_out1, 17);


  Relational_Operator28_relop1 <= '1' WHEN resize(HDL_Counter4_out1, 17) >=
Subtract30_out1 ELSE
      '0';

  Logical_Operator15_out1 <= Relational_Operator33_relop1 OR
Relational_Operator28_relop1;

  Constant27_out1 <= to_unsigned(16#00FA#, 16);

  Subtract28_sub_cast <= signed(resize(Subtract30_out1, 18));
  Subtract28_sub_cast_1 <= signed(resize(Constant27_out1, 18));
  Subtract28_sub_temp <= Subtract28_sub_cast - Subtract28_sub_cast_1;
  Subtract28_out1 <= Subtract28_sub_temp(15 DOWNTO 0);

  Relational_Operator30_1_cast <= signed(resize(HDL_Counter4_out1, 17));

  Relational_Operator30_relop1 <= '1' WHEN Relational_Operator30_1_cast >=
resize(Subtract28_out1, 17) ELSE
      '0';

  Logical_Operator22_out1 <= Logical_Operator15_out1 AND Relational_Operator30_relop1;


  Switch6_out1 <= Logical_Operator15_out1 WHEN switch_compare_1_5 = '0' ELSE
      Logical_Operator22_out1;

  ce_out <= clk_enable;




  S1 <= Logical_Operator7_out1;

  S2 <= Logical_Operator3_out1;

  S1_1 <= Logical_Operator4_out1;
```

```
  S2_1 <= Logical_Operator1_out1;

  S3 <= Logical_Operator13_out1;

  S4 <= Switch1_out1;

  S3_1 <= Logical_Operator9_out1;

  S4_1 <= Switch2_out1;

  S6 <= Logical_Operator18_out1;

  S5 <= Switch3_out1;

  S6_1 <= Logical_Operator21_out1;

  S5_1 <= Switch6_out1;

END rtl;
```