**IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)**
**ISSN(P): 2347-4599; ISSN(E): 2321-8843**
**Vol. 5, Issue 6, Jun 2017, 1-18**
**© Impact Journals**

# FPGA BASED CONTROL SINGLE PHASE THREE LEVEL INVERTER

**AMEER AHAMED Z[1] & FAREEDA BANU A[2]**

[1]Assistant Professor, Department of EEE HMSIT, Tumkur, India

[2]Lecturer, Department of EEE Siddharameshwara Polytechnic, Tiptur, India

## ABSTRACT

The three level inverters are implemented using digital logic that determines all required switching sequences and the necessary switching sequence by dividing the timings using FPGA. The FPGA can be programmed using Verilog which is straight forward technique which is cost effective for single chip implementation even for very high switching frequencies.

**KEYWORDS:** FPGA, VHDL, CPLD, XLINX

## 1. INTRODUCTION

In the Power electronic converters harmonic reductions is the main issue which will affect the power factor which intern affects power quality. When there is a low power factor the availability of power to the grid will be less, the low power intern causes EMI which will affect the power quality through between different systems connected to the same grid. When we go for several applications such as UPS and inverters there should be fast response. The use of microcontrollers is one of the best methods, which is flexible straight forward in designing hardware. This project aims to present a digital controller applied to a single-phase inverter. The PWM technique is intensively used in inverters, and the switching instant is determined by the comparison between a sinusoidal signal and a triangle wave, and is chosen as the strategy to drive the switches. The switching strategy is implemented with an FPGA device.

### 1.1 Disadvantages of DSP

In recent years, Digital Signal Processors (DSPs) or microcontrollers have been widely used for implementing digital motor control algorithms in industry. But it has some disadvantages like, long development period and poor software portability/re-usability. Any change of the microprocessor, imposed by the introduction of new features or the need of better performance require a huge revision of the project, in order to fit with the new system. PWM generation and current control loop requires high sampling rate to achieve wide band width performance and for that purpose most of the computational resources of the controller must be utilized. Only a limited time is left to control other functions of the drive. A new design methodology that can satisfy these demands are FPGA based hardware implementation technology.

FPGAs have traditionally found use in high speed custom digital applications where designs tend to be more constrained by performance rather than cost. The explosion of integration and reduction in price has led to the more recent widespread use of FPGAs in common embedded applications. FPGAs, along with their non-volatile cousins CPLDs (Complex Programmable Logic Devices), are emerging as the next digital revolution that will bring about change in much the same way that microprocessors did. With current high end devices exceeding 2000 pins and topping billions of transistors, the complexity of these devices is such that it would be impossible to program them without the assistance of high level design tools. Xilinx, Altera, Actel, and Lattice all offer high end EDA tool suites designed specifically to support their own devices however they also offer free versions aimed at supporting the bulk of FPGA development. These vendors understand the importance of tool availability to increased silicon sales and they all seem committed to supporting a free version of their tools for the foreseeable future.

Through the use of EDA tools, developers can design their custom digital circuits using either schematic based technique, VHDL, Verilog or any combination of these methods. Prior to the Altium Designer system, vendor independent FPGA development tools were extremely expensive. Furthermore they were only useful for circuits that resided within the FPGA device. Once the design was extended to include a PCB and ancillary circuits, a separate EDA tool was needed. Altium Designer has changed all of this by being the first EDA tool capable of offering complete schematic to PCB tool integration along with multivendor FPGA support. Altium made the logical extrapolation of trends in the FPGA world and recognized that FPGAs are quickly becoming a staple in modern designs. By making available a range of processor cores that can be downloaded onto an FPGA device and bundling them with a complete suite of embedded software development tools, Altium Designer represents a unified PCB and embedded systems development tool.

## 2. THREE LEVEL INVERTER

Figure 2.0 shows the circuit configuration of the NPC inverter. IGBTs connected in series. The applied voltage is half that of two level inverter. The capacitor holds the voltage of each bus. Each leg is having two diodes.

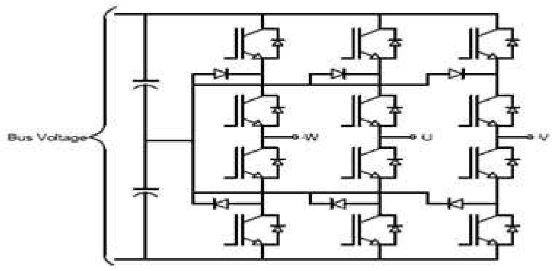


**Figure 2.0: NPC Inverter**

These inverters are used previously for industrial applications. It is having a capacity of handling higher voltage and also the NPC inverter has several favourable features of low ripple current and lone to line voltage steps.

### 2.1 Output Voltage and Switching States

The NPC inverter has 3 voltage level positive voltage, zero voltage and negative voltage. In case of two level it has only positive and negative levels only. For a one phase operation, When 1st IGBT and 2nd IGBT are turned on the output is connected to Vp and when 3rd IGBT and 4th IGBT are turned on the output is connected to $V_o$. Switching states for the four IGBTs are listed in Table 1. Diodes D4 and D5 provide the connection to the neutral point. From the switching states, it can be deduced that IGBTs 2nd IGBT and 3rd IGBT are turned on resulting in greater conduction loss, also resulting in zero voltage.

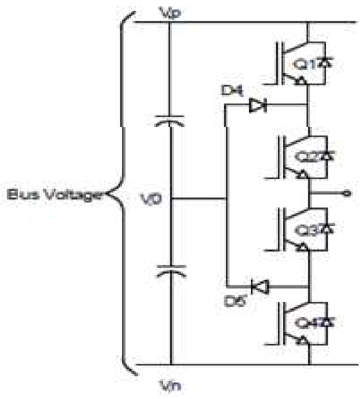


**Figure 2.1: Single Leg**

**Table 1: Switching States**

| IGBT | Vout = Vp | Vout = V0 | Vout = Vn |
|---|---|---|---|
| Q1 | On | Off | Off |
| Q2 | On | On | Off |
| Q3 | Off | On | On |
| Q4 | Off | Off | On |

The Capacitors are connected in cascade to provide $V_0$ as the middle voltage.

Figure 2.1 is a graph of the leg output voltage and Figure 2.3 is a graph of the phase-to-phase output voltage. Careful observation shows that the effective switching frequency of the phase-to-phase voltage in Figure 2.3 is twice that of the phase voltage shown in Figure 2.2. A two level inverter is required to use two times the switching frequency of an NPC inverter in order to achieve the same ripple in the output current. This simple fact coupled with the intermediate voltage steps of the NPC inverter offers two advantages over the two level inverter. First, there are far less switching losses in the NPC inverter and second, if an output filter is required, the filter components will be smaller in both value and size than the filter components for a two level inverter.

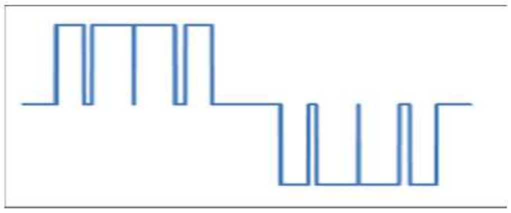


**Figure 2.2: Leg Output Voltage**

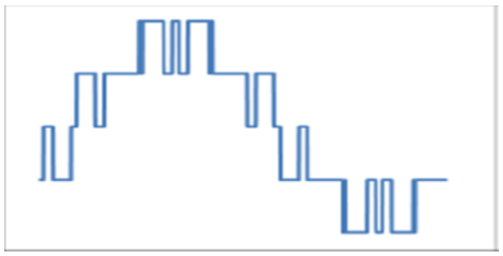


**Figure 2.3: Phase to Phase Output Voltage**

## 2.2 Diode Clamped Inverter

The diode clamped three level inverter system will be analyzed. The circuit to be investigated theoretically and experimentally is the one depicted in figure. This type of inverter corresponds to a three level topology which means that the output voltage from one inverter leg, measured using the DC midpoint, M, as a reference, can get three different values, a positive voltage, +UDC, a negative voltage, -UDC and a zero voltage or state. The single phase equivalent circuit of the diode clamped three level inverter can be found in figure, where only the one inverter leg is depicted. Below in figure a three phase diode clamped inverter. The principle of operation of the inverter can be clearly elucidated. Having two different current directions, positive and negative, and three different output voltages it is deduced that six current paths are available. In the above figure the current path depends on the state of each switch of the inverter. Below the current path is further explained for both current directions, positive and negative where, the components names are based on figure 2.4.

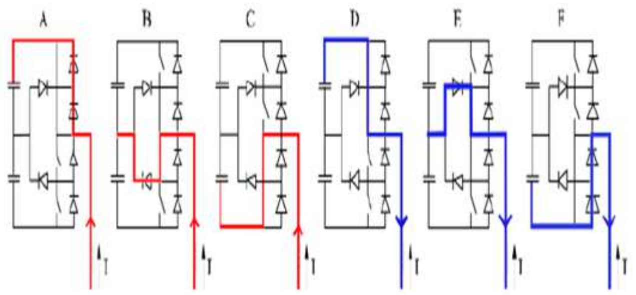


**Figure 2.4: The Current Paths of Diode Clamped Inverter**

### 2.2.0: Positive Current Direction

When the switches, First and second are turned-on the current freewheels and the voltage at 1st and 2nd reaches the capacitor. The measured output voltage is VDC.

- When 2nd and 3rd switches are turned on the $V_{ce}$ connected to the diodes and the voltage is 0.
- When the lower switches 3rd and 4th are turned on the output connects to the capacitor C2 and the output voltage is negative.

## 3. VERILOG PROGRAMMING

### 3.0:Hardware Description Language

Today most digital design of processors and related hardware system is done using a hardware description language. Such a language serves two purposes. First, it provides an abstract description of the hardware to simulate and debug the design. Second, with the use of logic synthesis and hardware compilation tools, this description can be compiled into the hardware implementation. In this section, we introduce the hardware description language Verilog and show how it can be used for combinational design. In the rest of the appendix, we expand the use of Verilog to include design of sequential logic. In optional sections of which appear on the CD, we use Verilog to describe processor implementations.

### 3.1: Basics of Verilog

Verilog is one of the two primary hardware description languages; the other is VHDL. Verilog is somewhat more heavily used in industry and is based on C, as opposed to VHDL, which is based on Ada. The reader generally familiar with C will find the basics of Verilog, which we use in this project, easy to follow.

Verilog can specify both a behavioral and a structural definition of a digital system. A behavioral specification describes how a digital system functionally operates. A structural specification describes the detailed organization of a digital system usually using a hierarchical description. A structural specification can be used to describe a hardware system in terms of a hierarchy of basic elements such as gates and switches. Thus, we could use Verilog to describes the exact contents of the truth tables and datapath of the last section. With the arrival of hardware synthesis tools, most designers now use Verilog or VHDL to structurally describe only the datapath, relying on logic synthesis to generate the control from a behavioral description. In addition, most CAD systems provide extensive libraries of standardized parts, such as ALUs, multiplexors, register files, memories, programmable logic blocks, as well as basic gates. Obtaining an acceptable result using libraries and logic synthesis requires that the specification be written with an eye toward the eventual synthesis and the desired outcome. For our simple designs, this primarily means making clear what we expect to be implemented in combinational logic and what we expect to require sequential logic. In most of the examples we use in this section, and the remainder of this appendix, we have written the Verilog with the eventual synthesis in mind.

**3.2: Data Types and Operators in Verilog**

There are two primary data types in Verilog:

- A wire specifies a combinational signal.
- A reg (register) holds a value, which can vary with time. A reg need not necessarily correspond to an actual register in an implementation, although it often will.

A register or wire, named X, that is 32 bits wide is declared as an array: reg [31:0] X or wire [31:0] X, which also sets the index of 0 to designate the least significant bit of the register. Because we often want to access a subfield of a register or wire, we can refer to contiguous set of bits of a register or wire with the notation [starting bit: ending bit], where both indices must be constant values. An array of registers is used for a structure like a register file or memory. Thus, the declaration reg [31:0] registerfile[0:31] specifies a variable register file that is equivalent to a MIPS register file, where register 0 is the first. When accessing an array, we can refer to a single element, as in C, using the notation registerfile[regnum]. The possible values for a register or wire in Verilog are

- 0 or 1, representing logical false or true
- z, representing unknown, the initial value given to all registers and to any wire not connected to something
- z, representing the high-impedance state for tristate gates, which we will not discuss in this appendix

Constant values can be specified as decimal numbers as well as binary, octal, or hexadecimal. We often want to say exactly how large a constant field is in bits. This is done by prefixing the value with a decimal number specifying its size in bits. For example:

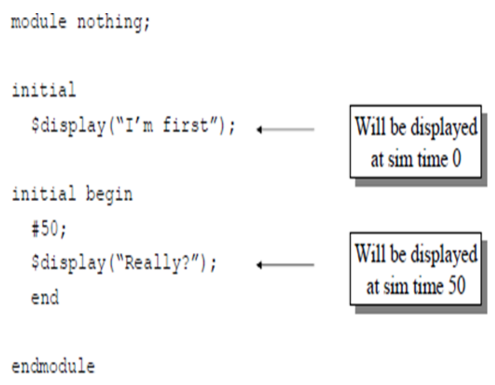
- 4‟‟b0100 specifies a 4-bit binary constant with the value 4, as does 4‟‟d4.
- 8 „h4 specifies an 8-bit constant with the value –4 (in twos complement representation)

Values can also be concatenated by placing them within { } separated by commas. The notation {x {bit field}} replicates bit field x times. For example:

- {16{2‵‵b01}} creates a 32-bit value with the pattern 0101... 01.
- {A[31:16], B[15:0]} creates a value whose upper 16 bits come from A and whose lower 16 bits come from B.

### 3.3: INITIAL Block and always Block

Start execution at time zero and finish when their last statement executes.



```
module nothing;

initial
  $display("I'm first");

initial begin
  #50;
  $display("Really?");
  end

endmodule
```

Will be displayed at sim time 0
Will be displayed at sim time 50

**Figure 3.0: INITIAL Block**

Start execution at sim time zero and continue until sim finishes

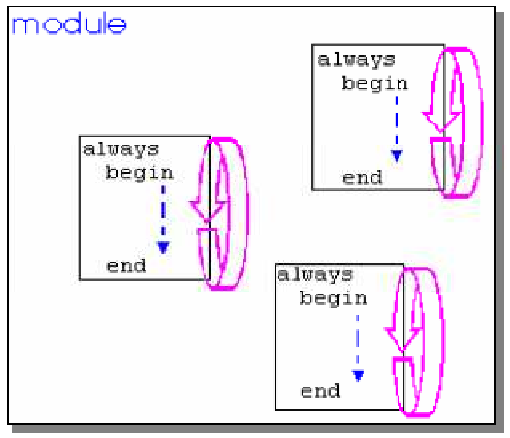


**Figure 3.1: ALWAYS Block**

## 4. FIELD PROGRAMMABLE GATE ARRAY

### 4.0: XILINX

Xilinx is disclosing this Document and Intellectual Property (hereinafter “the Design”) to you for use in the development of designs to operate on, or interface with Xilinx FPGAs. Except as stated herein, none of the Design may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Any unauthorized use of the Design may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

Xilinx does not assume any liability arising out of the application or use of the Design; nor does Xilinx convey any license under its patents, copyrights, or any rights of others. You are responsible for obtaining any rights you may require for your use or implementation of the Design. Xilinx reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of Xilinx. Xilinx assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

## 4.1: F.P.G.A

### Advantages of FPGA

Field Programmable Gate Array (FPGA) offers the most preferred way of designing PWM Generator. When design is implemented on FPGA they are flexible in design by changing the connection between the blocks, which is reprogrammed easily. It has a less delay, hence FPGA is best method for PWM generation and implementation in inverters. It is less costlier for small applications.

Similar to microprocessors, Xilinx FPGAs optionally load or boot themselves automatically from an external non volatile memory device. Alternatively, similar to microprocessor peripherals, Spartan-3 generation FPGAs can be downloaded or programmed by an external “smart agent”, such as a microprocessor, DSP processor, microcontroller, PC, or board tester. In either case, the configuration data path is either serial to minimize pin requirements or byte-wide for maximum performance or for easier interfaces to processors or to byte-wide Flash memory.

## 4.2: Software Integration

After building the project with the described components above, software must be added to make each module communicate with each other and external devices. It produces a VHDL file that specifies names associated with each module. By using those pre-defined names in software, the programmer can interface with each of the components via the OPB bus. The EDK platform is capable of quickly adding pre-made C files and header files. Each I/O pin has a specific pin value that is associated with certain module components and GPIO. These pins are declared in the UCF and lets know where to be expecting the signals, while EDK takes care of all of the routing.

## 4.3: PWM OUTPUT

The PWM output signal performed as designed, according to the pseudo-MPPT algorithm. In order to get a better visualization of the signal, the PWM period was slowed down to 25Hz so measurements could be taken of the duty cycle step size. During the start-up mode, the duty-cycle increases by 5% until 35% was reached, at which point, by adjusting the incoming channel 0 ADC voltage, the duty cycle could be increased or decreased by ~.25% steps. Figure are snapshots of the PWM right at start-up, and the increase can be seen quite clearly. The top is the initial 5% duty-cycle followed by the next ISR iteration, where the duty cycle is 10%. Furthermore, as the voltage is raised above 2.5V, not only does the PWM duty-cycle drop to zero, the LED corresponding to that battery load line shuts off, demonstrating the emergency shut-off requirement was also met. The purpose was to show that a clean PWM signal can be produced and adjusted in real-time to increase efficiency and this point was well verified.

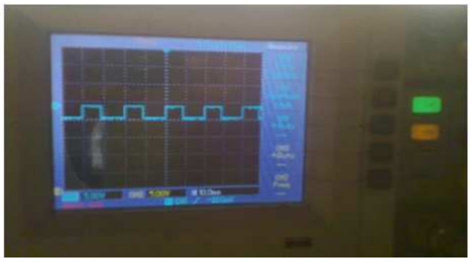


**Figure 4.0: PWM Output to ON 1 & 4 Switches**

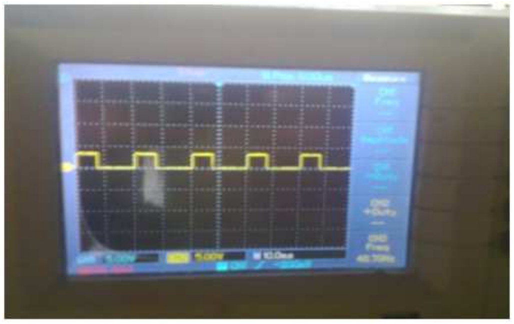


**Figure 4.1: PWM Output to ON 2 & 3 Switches**

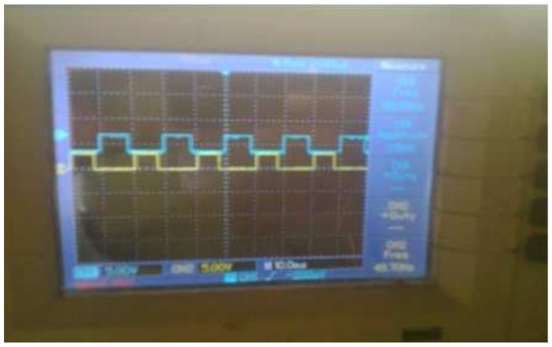


**Figure 4.2: Both PWM Outputs**

Lastly, the Spartan 3E has an output logic high voltage of 3.3V, which is not high enough to drive the MOSFETS or DC-DC converter. Additional circuitry will be required to boost the output signals to their required level by use of buffers. On the other side, step down voltage circuits must also be made to ensure all sensor voltage levels are within the required range of.7V-2.9V. New software must be implemented to make sure voltage conversions are accurately calculated in reference to the MPPT algorithm and emergency shut-off thresholds.

## 5. SPARTAN-3E

### 5.0: SPATAN 3E Basics

Below, in Figure 5.0**,** is a simplified block diagram with the Spartan 3E as the central controlling component. As you can see, several parts are cut out of the design and will be replaced with on-board devices and internal modules. Other additions have been added to the overall project, like the pyrometer and thermocouple, which need new routed signals for future operation. Two multiplexers will be added to handle all sensor signals, instead of the NiDAQ 6009"s, because the Spartan 3E has a limited A/D interface.

The entire system could efficiently be controlled and thereby limit the power consumption and wiring complexity. Thus, the requirements for the system are:

- Ensure the battery is being charged/discharged properly via switch control and sensor monitoring.
- Adjust the PWM signal to the DC-DC converter according to the MPPT algorithm. Create multiplexing and interfacing design to handle all sensor data lines.
- Allow user to select desired loads via the on-board switches.
- Allow for emergency shut-off to all loads and hardware to protect against reversed currents and current overload with a response time of 1/100 seconds.
- Log and display all sensor data.

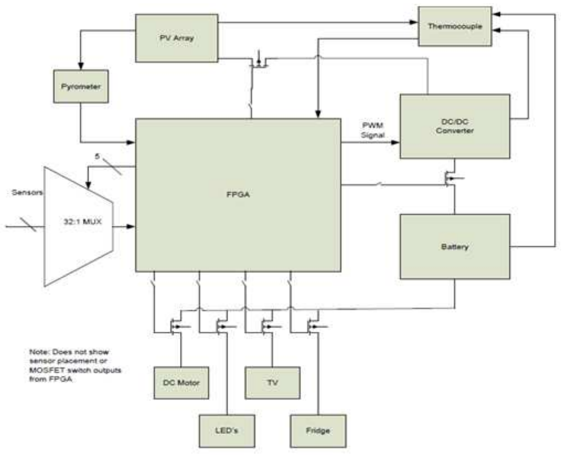


**Figure 5.0: Projected Block Diagram of Spartan 3E FPGA**

## 5.1: About SPARTAN 3 Development Kit

The Spartan-3E development board has over 100 I/O‟s that can handle all of the systems proposed 32 signals. Due to the design of the FPGA, half of those I/O‟s are shared with the FX2. In reality, there are only about 50 usable pins to interface the system. This allows us to either directly connect each system signal line to the board or to create a generic MUX design to handle the multiplexing. This latter approach would significantly reduce the number of I/O signals coming off the board to a few data lines, but would also require several MUX selector lines if the design is external to the board. The Spartan-3E has on-board ADC and DAC to handle the sensor conversions. The sensors will require a simple voltage divider to bring the voltage level within a 0-5V range and can easily be implemented with an additional PCB design.

The Spartan-3E has 16Mbytes of RAM and 4Mbytes of Flash. Estrada and Mariano suggested that Linux operating platform will require a major portion of the flash memory and further stated that an external memory device may need to be implemented to store a reasonable amount of senor data. This will not be a large expense, but a necessary one. In order to log a sufficient amount of the sensor data, including time stamps, the on-board memory devices, such as SDRAM, would not be a viable solution. Converting and logging the sensor data, as well as monitoring the MOSFET switches, can easily be performed to satisfy the timing requirements above.

The Spartan-3E"s clock can be varied to run at 25, 50, or 100MHz, so timing constraints should not be a factor and should allow ample time to cycle through the design program.

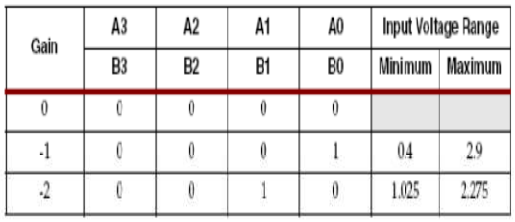
The Xilinx"s MicroBlaze (uBlaze) Embedded Development Kit (EDK) will be the softcore-based operating platform. MicroBlaze EDK has numerous IP cores that give designers pre-made modules through user friendly software. The platform allows the user to custom tailor their designs by adding modules and adjusting their parameters to fit their design needs. In particular, there is a timer/counter IP core that can be designed specifically as a PWM generator. This module will produce the required PWM scheme according to the MPPT algorithm, and can connect to the high-side driver that drives the DC-DC converter.

It requires 34 SCK cycles (SPI clock) to convert a sample analog input to digital data, including conversion and acquisition times. The SCK, therefore, can operate at maximum of 50 MHz. The Pre-Amp has a clock limitation of 10MHz, so the maximum system clock to SCK ratio is 5:1. The SPI IP core in Xilinx"s EDK allows for several parameterized configuration options regarding the two clock ratios. A ratio of 16 (system clock / SCK) was chosen to allow for extra design leeway, but, if necessary, a higher ratio of 8 can be implemented. The complete analog capture circuit below demonstrates the SPI interface with both devices and the Spartan 3E pin assignments. The SPI has a tri-stated output and is configured as the master, while each subsequent device are defined as one-hot encode slaves to the master, thereby ensuring only one device is selected at a time and won"t block communication on the bus.

**5.2: Analog Inputs Overview**

The ADC is a dual channel device which samples two separate signals during the same SPI read cycle. For system testing purposes, channel 0 is used solely while the other channel is left unconnected. It should be noted that during the SPI read cycle, both 14-bit samples are properly placed into separate channel variables within the C code of the project.

**Table 5.1: Programmable Gain Setting in Spartan 3E**



| Gain | A3 | A2 | A1 | A0 | Input Voltage Range | |
|---|---|---|---|---|---|---|
| | B3 | B2 | B1 | B0 | Minimum | Maximum |
| 0 | 0 | 0 | 0 | 0 | | |
| -1 | 0 | 0 | 0 | 1 | 0.4 | 2.9 |
| -2 | 0 | 0 | 1 | 0 | 1.025 | 2.275 |

This overview is not intended to replace the Linear Technology data sheets and you are recommended to consult the LTC1407A-1 and LTC6912-1 data sheets to review the full range of features these devices offer as well as check the operating specifications particularly in relation to analogue inputs. The LTC6912-1 provides two independent inverting amplifiers. Signals are amplified relative to 1.65v. The gain of each amplifier is programmable from -1 to -100 which enables signals as small as ±12.5mV to apply full scale inputs to the A/D converters. The LTC1407A-1 provides two analogue to digital converters. Both analogue inputs are sampled simultaneously when the AD-CONV signal is applied.Both devices have a **S**erial **P**eripheral **I**nterface (SPI) to allow the devices to be controlled and digital values to be read. The amplifier also has a shutdown control. All of these signals link directly to the Spartan-3E using the pins indicated.

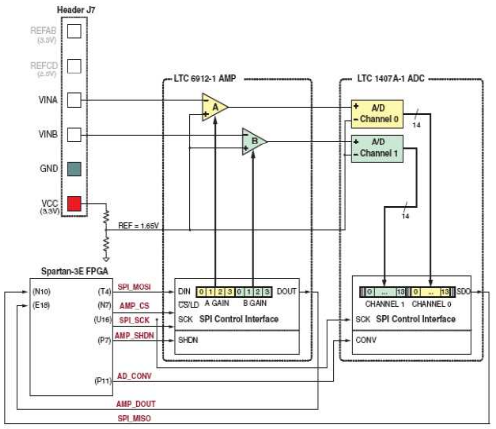


**Figure 5.1: Detailed View of Analog Capture Circuit**

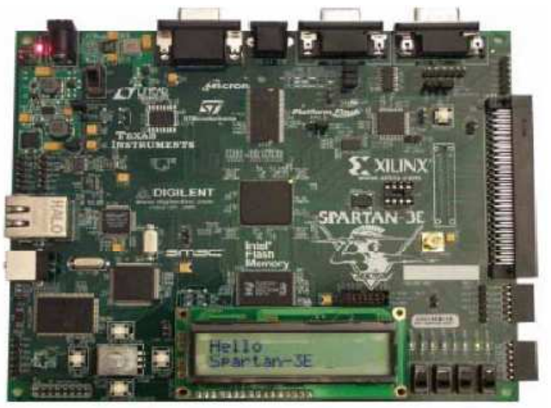


**Figure 5.2: Spartan 3E FPGA Development Kit**

## 6. SIMULATION RESULTS

### 6.0 Simulation Results

The output voltage and output current for the single-phase inverter are shown in Fig. 7.0, represented by sinusoidal waveforms. A 40% load step is applied to the inverter, so that the voltage loop can be evaluated in transient condition. This focuses on the investigation of the DCTLI using the simulation software MATLAB Simulation. All the components that are incorporated in the inverter were allocated, and thus an equivalent model of this system was build in P-Spice with the intention to implement an analysis of its overall operational behaviour, switching behaviour and estimation of power losses. Additionally, an investigation regarding the meaning of using R Clamped was implemented. The acquired results from this chapter would then be compared to the results.

This model was designed by following the main figure 7.0. Therefore, by watching the two figures can be observed. Four voltage sources have been applied across each IGBT unit of each module. These sources represented the gate driver circuit required by each IGBT unit under pragmatic conditions. The intention of these sources was to simply, turn-on and turn-off the IGBT for particular duty-ratios. In MATLAB simulation software it was found that the s circuits would act every some seconds. Thus each duty-ratio of on IGBT unit was equal to 2.3µs.

Case–i: when the system is uniformly loaded by 130% lines 1 and 10 are loaded by 126.036% and 110.97% respectively. For the BL objective function (9) is optimized using the real parameter GAs, the obtained objective function values are given in Table 2 and this overloading can be relieved by placing SVC at 25th bus with BSVC of -

0.005275p.u and TCSC in 40th line with XTCSC of -0.118683p.u. From Table 2 it can be further inferred that, when VS is considered as optimization objective, BL is reduced from its base case value and the losses have also increased from its base case value. Considering LM as optimization objective reveals that a reduction in system transmission loss is associated with reduction in BL and VS values. This clearly demonstrates the conflicting nature of three objectives considered.

## 6.1: Simulation Circuit

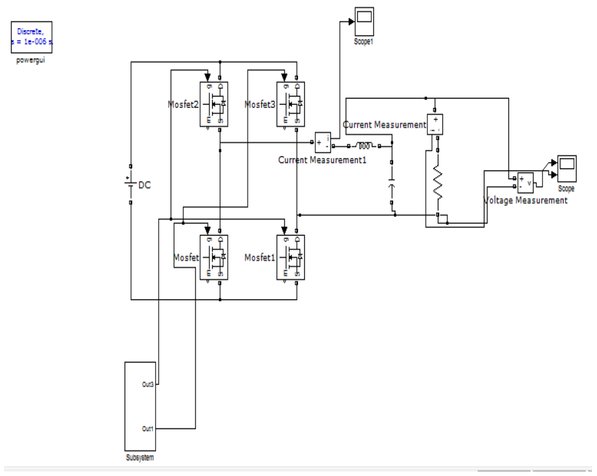


**Figure 6.0: Simulation Circuit**

In this circuit the PWM signals are got from the simulation software i.e through MATLAB simulation, the other circuit is also shown with same design as shown below

**Table 6.1: Digital Parameters**

| Description | Value |
|---|---|
| Maximum dc link voltage | Vi(max)=400V |
| Modulation index | fm=0.2 |
| Output filter inductance | Ls=0.746mH |
| Output filter capacitance | Cs=10µF |
| Load impedance | Ro=12Ω |
| Switching frequency | fs=36kHz |
| Sampling gain | β=0.013 |
| Rms output voltage | Vo=110V |
| Output frequency | f=60Hz |

### 6.2: Results

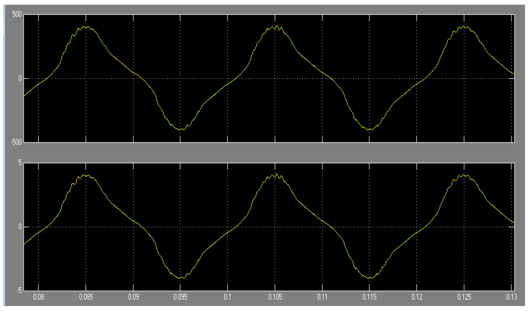


**Figure 6.1: Output Voltage and Output Current**

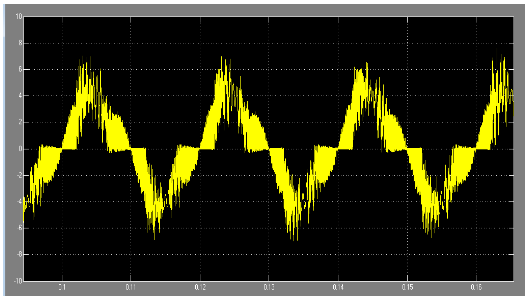
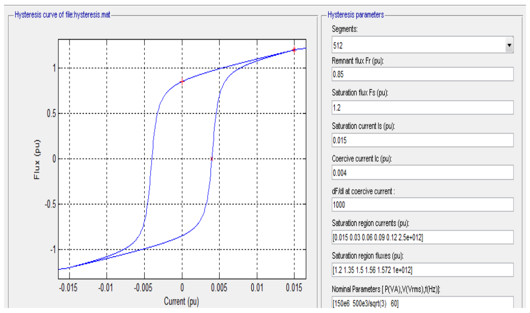


**Figure 6.2: Inductor Current**



Segments: 512

Remnant flux Fr (pu): 0.85

Saturation flux Fs (pu): 1.2

Saturation current Is (pu): 0.015

Coercive current Ic (pu): 0.004

dF/dI at coercive current : 1000

Saturation region currents (pu): [0.015 0.03 0.06 0.09 0.12 2.5e+012]

Saturation region fluxes (pu): [1.2 1.35 1.5 1.56 1.572 1e+012]

Nominal Parameters [ P(VA),V(Vrms),f(Hz)]: [150e6 500e3/sqrt(3) 60]

**Figure 6.3: Hysteresis Output**

## 7. HARDWARE IMPLEMENTATION

### 7.0 THE CIRCUIT

The NPC inverter can produce three voltage levels on the output: the DC bus plus voltage, zero voltage and DC bus negative voltage. The two level inverter can only connect the output to either the plus bus or the negative bus. For a one phase operation, when IGBTs Q1 and Q2 are turned on, the output is connected to Vp; when Q2 and Q3 are on, the output is connected to V0; and when Q3 and Q4 are on, the output is connected to Vn. Switching states for the four IGBTs. Clamp diodes D4 and D5 provide the connection to the neutral point. From the switching states, it can be deduced that IGBTs Q2 and Q3 are on for most of the cycle, resulting in greater conduction loss than Q1 and Q4 but far less switching loss. In addition, the free wheel diodes for Q2 and Q3 are for most cases, soft switched as the IGBT parallel to the diode is on, thus holding the recovery voltage across the diode to that of the IGBT Vce.

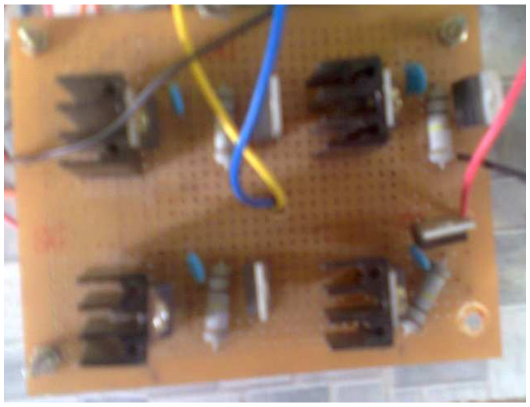


**Figure 7.0: Hardware of Three Level Inverter**

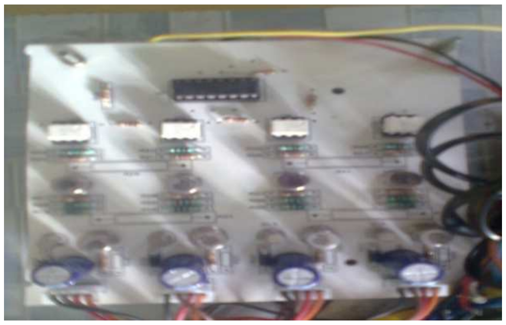


**Figure 7.1: The Hardware Driver Circuit**

## 7.1: Hardware Results

The circuit involves simple three level inverter, the pulses to the MOSFET is given by driver circuit and the pulses to driver circuit is got by FPGA kit, these driver pulses are used to drive MOSFET"s. The pulses from the driver can be shown as

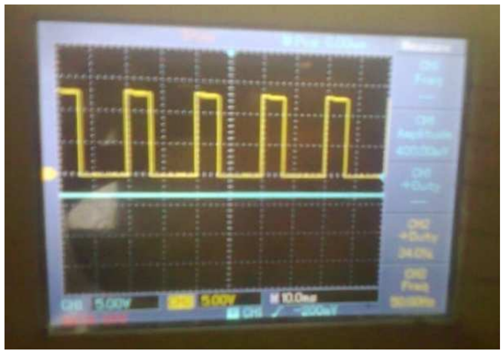


**Figure 7.2: The Driver Pulses for First Switches**

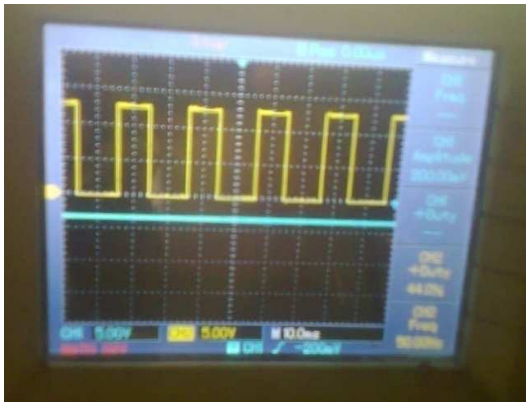


Figure 7.3: The Driver Pulses for Other Two Switches

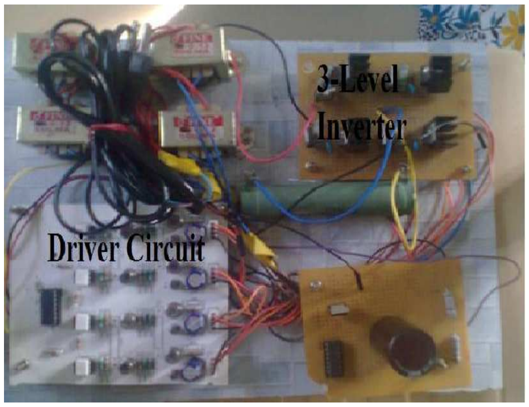


Figure 7.4: The Complete Hardware Circuit without Kit Connected

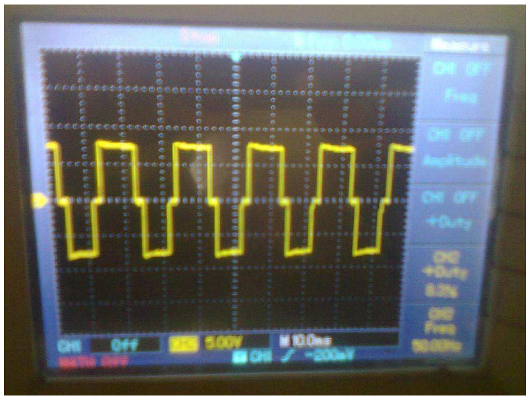


Figure 7.5: The Hardware Output

The FPGA-based controller is modular in structure. The interface for integrating the module in an inverter‟s control circuit is shown on the left, and the interface for the power electronics is shown on the right. The three-level inverter is controlled by the soft-core processor. First of all, the VHDL modules are initialized. Over-current threshold values and interlocking times can be parameterized in addition to the switching frequency. In the simplest case, an induction machine can be controlled using the voltage/frequency mode (*v*/*f*). The commands a rotating space vector to the three-level. This is done within an interrupt service routine (ISR), which is triggered cyclically by the PWM module. Alternatively, a current control algorithm can be executed by the µC within an ISR. To execute a field oriented control (FOC) as an example the needs less than 5 µs using 100 MHz CPU clock frequency and custom instructions. Should a fault occur, the processor can read out the fault type from the register of the associated module and reset it if necessary as soon as the cause of the fault is no longer active.

## 7. CONCLUSIONS

In this project has presented the theoretical analysis, the experimental implementation and hardware implementation of single phase three level inverter using FPGA. Experimental tests have demonstrated accurate results, with reduced harmonic distortion of the output voltage. Future work is supposed to include faster microcontrollers, with the development of more robust algorithms and online monitoring.

Implementing a three level space vector modulator completely in an FPGA is a lean straight forward approach. The solution is easy to implement and needs not much resources. Utilizing FPGA computing power high switching frequency does also enable high current control bandwidth. Additionally, it requires fewer components, PCB space and it allows simple connection schemes for field busses and/or other peripherals.

### Future Enhancement

In future the FPGA Programme can be written for six switches i.e project can enhanced for three phase inverter also it can be enhanced for five and seven level inverter, which will be use full to drive induction motors and also for other application.

## 8. REFERENCES

1. Jens Onno Krah, Markus Höltgen, Andreas Rath, Rolf Richter. FPGA-based Control of Three-Level Inverters PCIM Europe 2011, 17.-19. May 2011, Nuremberg, Germany
2. L. H. S. C. Barreto, P. P. Praca, C. M. T. Cruz and R. T. Bascope PID Digital Control Using Microcontroller and FPGA
3. Applied to a Single-Phase Three-Level Inverter 1-4244-0714-1/07/$20.00 C 2007 IEEE.
4. T. Brückner, D. G. Holmes,„ Optimal Pulse-Width Modulation for Three-Level Inverters‟, IEEE Transactions on Power Electronics, Vol. 20, No. 1, Jan. 2005, pp. 82 - 89.
5. J. O. Krah, J. Holtz: „High-Performance Current Regulation and Efficient PWM Implementation for Low-Inductance Servo Motors‟, IEEE Transactions on Industry Applications, Vol. 35, No. 5, Sept/Oct 1999, pp. 1039-1049.
6. Avago Datasheet ACPL-796, ACPL-C797: www.avagotech.com
7. J. O. Krah, C. Klarenbach: „FPGA based Field Oriented Current Controller for High Performance Servo Drives‟, PCIM Power Conversion Intelligent Motion, Nuremberg, May 2008.
8. C. Klarenbach, J. O. Krah: „Fast and High Precision Motor Control for High Performance Servo Drives‟, PCIM Power Conversion Intelligent Motion, Nuremberg, May 2010, pp. 326-333.

9. Ning-Yi Dai, Man-Chung Wong and Ying-Duo Han A FPGA-Based Generalized Pulse Width Modulator for Three-Leg Center-Split and Four-Leg Voltage Source Inverters IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 23, NO. 3, MAY 2008.

10. Soumya M.B., Shiny.G. FPGA Realization of Modulation Scheme for Three Level Inverters 10th National Conference on Technological Trends (NCTT09) 6-7 Nov 2009.

11. Ioannis Karampoikis DEVELOPING AND INVESTIGATION OF A DIODE CLAMPED THREE LEVEL INVERTER-LEG WITH PASSIVE UNDELAND SNUBBER Master Thesis, August 2011 Technical University of Denmark.

## AUTHORS

**Ameer Ahamed Z** working as Assistant Professor in Department of Electrical and Electronics Engineering in HMSIT Tumkur, Since from 4.5 years

**Fareeda Banu A** Working as Lecturer in Siddharameshwara Polytechnic in Department of Electrical and Electronics Tiptur