University of Alberta

FPGA-Based Real-Time Simulation of Variable Speed AC Drive

by

Aung Myaing

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Power Engineering and Power Electronics

Department of Electrical and Computer Engineering

Fall 2010
Edmonton, Alberta

Examining Committee

Dr. Venkata Dinavahi (Supervisor), Department of Electrical and Computer Engineering

Dr. Don Koval, Department of Electrical and Computer Engineering

Dr. Amit Kumar, Department of Mechanical Engineering

With love, praise and honor, this thesis is dedicated to my
family, but especially to my parents and to my beloved
wife Khin Aye Mu, for her support, understanding,
especially for her love, and bearing with me.

Abstract

Sophisticated power electronic apparatus and their digital control systems are finding increasing applications in electric power systems at generation, transmission, distribution and utilization levels. It is essential to carry out rigorous performance evaluation of such apparatus before commissioning. Field Programmable Gate Arrays (FPGAs) are becoming an attractive platform for accelerating computationally intensive applications.

This thesis presents a FPGA-based real-time digital simulator for power electronic drives based on realistic device characteristics. A 3-level 12-pulse Voltage Source Converter (VSC) fed induction machine drive is implemented on the FPGA. The system components include the 3-level VSC, the induction machine, the direct field oriented controller, and the pulse width modulator. Both system-level and device-level IGBT models are utilized to implement the VSC. The VSC model is computed at a fixed time-step of 12.5ns allowing an accurate representation of the IGBT nonlinear switching characteristics.

Altera Startix EP1S80 and EP3SL150F1152C2 FPGA boards utilized for the real-time simulation. All models were implemented in VHDL. The FPGA boards were interfaced to external DAC boards to display real-time results on the oscilloscope. The real-time results were validated using an off-line cosimulation set-up using the SABER and MATLAB/SIMULINK software.

Acknowledgements

I wish to express my deep gratitude to Dr. Venkata Dinavahi, my supervisor, for his constant and kind support, excellent advices and guidance during the whole research work, and financial support throughout the research of this work. His discerning suggestions always proved very useful and were much appreciated. I extend many thanks to the members of my M. Sc. committee chair and members for their careful reviews of this thesis and for many useful comments.

I would like to acknowledge CMC (Canadian Microelectronics Corporation) support for using the IBM computer, Stratix EP1S80 FPGA board and also thank to the NSERC (Natural Sciences and Engineering Research Council of Canada) to provide the Stratix III EP3SL150F1152C2N FPGA board and other lab equipments. I would also thank all of my friends who support me physically and mentally to finish this research project. I would really appreciate all of them.

I am wholly indebted to my parents for their support, invaluable guiding and wisdom that has bought me this far and will hopefully carry me further. Thanks for their love.

Above all, I especially thank to my beloved wife, Khin Aye Mu, for providing me constant moral support, inspiration, encouragement and assistance to complete this research project. Without her love, I could not have completed my study.

Finally, I would mention that I love you all and God bless you!

Contents

1 Introduction 1

1.1 Power electronic applications . . . 1

1.2 Real-time simulation . . . 4

1.3 FPGA technology . . . 5

1.4 Motivation for this work . . . 6

1.5 Thesis overview . . . 7

2 Overview of FPGA technology 9

2.1 Introduction . . . 9

2.2 Field Programmable Gate Arrays . . . 10

2.2.1 Architecture . . . 12

2.2.2 SRAM based FPGA . . . 12

2.2.3 Hardwired logic blocks . . . 14

2.3 Stratix and Stratix III devices . . . 14

2.3.1 Stratix architecture . . . 15

2.3.2 Stratix III architecture . . . 26

2.3.3 Flexible, efficient ALMs . . . 27

2.3.4 TriMatrix memory for the perfect fit . . . 28

2.3.5 Highest performance digital signal processing . . . 29

2.4 Digital signals processing (DSP) . . . 30

2.4.1 Basic elements of DSP system . . . 31

2.4.2 DSP hardware . . . 32

2.5 Hardware description language (HDL) 34

2.6 Summary 37

3 Modeling and Control of a Variable Speed AC Drive 38

3.1 Introduction 38

3.2 3-level voltage source converter 40

3.3 Experimental measurement of IGBT electrical characteristics . . 42

3.3.1 Generalized IGBT modeling guideline 46

3.3.2 3-level VSC model 49

3.4 Induction machine 59

3.4.1 State-space induction machine model 62

3.5 Carrier-based pulse width modulation 64

3.6 Closed-loop control system 66

3.7 Summary 69

4 FPGA-based Real-Time Implementation of a Variable Speed Drive 70

4.1 Introduction 70

4.2 Hardware platform 71

4.2.1 Stratix DSP development board 71

4.2.2 Stratix III DSP development board (DE3) 73

4.2.3 Digital-to-Analog Converters 75

4.2.4 Tektronix Digital Phosphor Oscilloscope DPO7054 . . . 76

4.3 FPGA implementation of system models 77

4.3.1 3-level VSC hardware realization 78

4.3.2 Induction machine hardware realization 82

4.3.3 Direct field oriented control hardware realization 83

4.3.4 STPWM hardware realization 85

4.3.5 FPGA hardware resources utilized for implementation . . 85

4.4 Results and Discussion 86

4.4.1 Off-line validation for real-time simulation 86
4.4.2 IGBT switching characteristics 89
4.4.3 Steady-state results 89
4.4.4 Transient results 93
4.5 Summary 95

5 Conclusions and Future Work 99
5.1 Contributions of This Work 99
5.2 Suggestions for future work 100

Bibliography 102

Appendix A Off-line cosimulation 109
A.1 SABER and Matlab/Simulink cosimulation 109

Appendix B FPGA Design Files 112
B.1 Real-Time Digital Simulator and Controller 112

Appendix C FPGA Design Files II 129
C.1 Motor and PWM 129

List of Tables

2.1 JTAG IN 10-Pin Header Pin-Outs. 26

3.1 Switching data for CM100DU-24F IGBT unit from *Powerex*®. . 46

3.2 Induction Machine Parameters 63

4.1 Stratix Device Features. 73

4.2 Stratix III Device Features. 74

4.3 Number format for the VSC model variable. 82

4.4 Number format for the controller variable. 84

4.5 EP1S80 FPGA resources used by the system components 87

4.6 EP3SL150 FPGA resources used by the system components . . 87

4.7 The line voltage magnitudes for various IGBT models 94

List of Figures

1.1 Dissertation Content. 8

2.1 Implementation Technology. 10

2.2 Typical FPGA architecture. 12

2.3 Configurable Logic Block (CLB) [23]. 13

2.4 Basic FPGA Diagram [23]. 14

2.5 Examples of Hardwired Logic Blocks. 15

2.6 Basic Logic Element. 15

2.7 FPGA Cluster-style Logic Block Contents. 15

2.8 DSP Blocks Arranged in Columns. 17

2.9 Stratix Block Diagram [23]. 18

2.10 Logic Array Blocks. 19

2.11 Logic Array Blocks control signals [23]. 20

2.12 Stratix Logic Element [23]. 21

2.13 Typical logic element (LE) (normal mode) in a SRAM based FPGA [23]. 22

2.14 Stratix DSP Block [23]. 24

2.15 Standard FPGA Logic vs. Stratix III FPGA Logic [23]. 27

2.16 ALM block diagram and example LUT configurations [23]. . . 28

2.17 TriMatrix memory [23]. 29

2.18 DSP Block [23]. 30

2.19 Program compilation, linking, and execution flow. 33

2.20 Connecting the electronics. 36

3.1 Multilevel converters topology. 39

3.2 Multilevel modulation technology. 40

3.3 12-pulse 3-level VSC drive. 41

3.4 CM50DU-24F IGBT module from *Powerex*® 43

3.5 Experimental test circuit to measure IGBT characteristics. . . 44

3.6 Generic IGBT switching characteristics with switching times and loss definitions. 44

3.7 The measured result of IGBT V_{ce} and I_c in turn-on transient. (V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.) 45

3.8 The measured result of IGBT V_{ce} and I_c in turn-off transient. (V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.) 45

3.9 One leg of the 3-level VSC. 50

3.10 Idealized timing characteristics of an IGBT during turn-on and turn-off switching. 51

3.11 One leg of the 3-level converter with ideal models for the IGBTs. 51

3.12 Switching function model 52

3.13 Switching function model representation for 3-level converter. . 53

3.14 Single pole triple throw on/off switch. 53

3.15 Input voltage, average function and output voltage (from top to bottom). 55

3.16 Averaged model representation for 3-level converter. 56

3.17 Characteristic $V_{CE} - I_C$ curve of a generic IGBT for a specific value of V_{GE}. 57

3.18 Linearized timing characteristics of IGBT hard-switching turn-on/off waveforms. 58

3.19 Schematic view of a squirrel cage induction motor. 61

3.20 3-level asymmetrical regular sampled PWM. 66

3.21 Direct field-oriented controller based on rotor flux. 68

3.22 Schematic of direct field-oriented controller for AC-motor drive. 68

4.1 Overview of hardware setup. 71

4.2 EP1S80 FPGA Stratix board. 73

4.3 EP3SL150F1152C2 FPGA Stratix III DE3 board. 74

4.4 OP5330 of Opal-RT DAC card. 75

4.5 THDB ADA (ADA) daughter board from Terasic. 76

4.6 Digital Phosphor oscilloscope DPO7054. 77

4.7 Functional block diagram of the hardware realization of IGBT system level models for one VSC leg. 78

4.8 Functional block diagram of the hardware realization of one leg of 3-level VSC using linearized device-level IGBT model. 80

4.9 Functional block diagram of the hardware realization of IGBT non-linear device-level model for one VSC leg. 81

4.10 Functional block of the implemented induction machine model. . 83

4.11 Implementation of PI controller. 84

4.12 Functional block diagram of the FPGA hardware realization for 3-level PWM. 86

4.13 IGBT Equivalent model circuit of SABER. 88

4.14 Real-time simulation results for switching function IGBT model. [V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.] 90

4.15 Real-time simulation results of linearized device-level IGBT model. [V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.] 91

4.16 Real-time simulation results of nonlinear device-level IGBT model. [V_{ce}: 100 V/div., V_{ge}: 30 V/div., I_c: 5 A/div., time: 200 ns/div.] 92

4.17 Output voltage waveform of 3-level VSC with ideal model for IGBT. [v_β: 500V/div., time: 5ms/div.] 93

4.18 Output voltage waveform of 3-level VSC with switching function model for IGBT. [v_β: 500V/div., time: 5ms/div.] 93

4.19 Output voltage waveform of 3-level VSC with averaged model for IGBT. [v_β: 500V/div., time: 5ms/div.] 94

4.20 Output voltage waveform of 3-level VSC with linearized device-level model for IGBT. [v_β: 500V/div., time: 5ms/div.] 95

4.21 Output voltage waveform of 3-level VSC with nonlinear device-level model for IGBT. [v_β: 500V/div., time: 5ms/div.] 95

4.22 Real-time simulation oscilloscope trace of steady-state line voltage v_{ab} and line current i_a applied to the machine, and harmonic spectrum of v_{ab}. [Scale: v_{ab}: 350V/div., i_a: 20A/div., Time: 5ms/div.] 96

4.23 Off-line SABER/SIMULINK simulation results of steady-state line voltage v_{ab} and line current i_a applied to the machine, and harmonic spectrum of v_{ab}. [Scale: v_{ab}: 350V/div., i_a: 20A/div., Time: 5ms/div.] 96

4.24 Real-time simulation oscilloscope trace of reference speed ω_r^*, machine speed ω_r and currents i_α, i_β for: machine running at 150rad/s and speed set-point variation to 377rad/s at t_1, speed reversal to −377rad/s at t_2, application and removal of load at t_3 and t_4, respectively. [Scale: ω_r^*, ω_r: 630rad/s/div., i_α, i_β: 150A/div., Time: 1s/div.] 97

4.25 Off-line SABER/SIMULINK simulation results of reference speed ω_r^*, machine speed ω_r and currents i_α, i_β for: machine running at 150rad/s and speed set-point variation to 377rad/s at t_1, speed reversal to −377rad/s at t_2, application and removal of load at t_3 and t_4, respectively. [Scale: ω_r^*, ω_r: 630rad/s/div., i_α, i_β: 150A/div., Time: 1s/div.] 97

A.1 SABER Block diagram of Saber/Simulink Co-simulation. . . . 110

A.2 Simulink Block diagram of Saber/Simulink Co-simulation. . . . 111

B.1 fpga.gdf. 114
B.2 VSC.gdf. 115
B.3 control.gdf. 116
B.4 control1.gdf. 117
B.5 control2.gdf. 118
B.6 control3.gdf. 119
B.7 control4.gdf. 120
B.8 cpwm.gdf. 121
B.9 igbt1nl.gdf. 122
B.10 igbt2.gdf. 123
B.11 igbt3.gdf. 124
B.12 igbt4.gdf. 125
B.13 igbt5.gdf. 126
B.14 measure.gdf. 127
B.15 motor.gdf. 128

C.1 pwm1.gdf. 130
C.2 pwm2.gdf. 131
C.3 pwm3.gdf. 132
C.4 pwm4.gdf. 133
C.5 pwm5.gdf. 134
C.6 ref.gdf. 135
C.7 VSC1.gdf. 136
C.8 Voltage Analysis.gdf. 137
C.9 Interface1.gdf. 138
C.10 Interface1a.gdf. 139
C.11 Interface2.gdf. 140

List of Acronyms

RTOS	Real-Time Operating System
FPGA	Field-Programmable Gate Array
RMS	Root-Mean-Square
STPWM	Sine wave with third harmonic modulation
PI	Proportional-Integral Controller
IGBT	Insulated Gate Bipolar Transistor
HIL	Hardware-In-the-Loop
VSC	Voltage Source Converter
IGCT	Integrated-Gate Commutated Thyristor
HEV	Hybrid Electric Vehicle
HDL	Hardware Description Language
ASIC	Application Specific Integrated Circuit
IP	Intellectual Property
SoC	System-on-Chip
PLD	Programmable Logic Device
PLBs	Programmable Logic Dlocks
DLL	Delay-Locked-Loop
LUTs	Look-up Tables

1
Introduction

1.1 Power electronic applications

Electricity runs the modern world with more beautiful, colorful and making our life more convenient and comfortable. It is also known that the power electronics makes the utilization of electrical energy more efficient, flexible and reliable as well as less expensive. Therefore, power electronic apparatus and their digital controls are finding increasing the applications in the modern power systems at the generation, transmission, distribution and the utilization levels.

Power electronic devices were broadly categorized into 3 groups: diodes, thyristors, and transistors. This classification is still valid for modern devices despite the fact that the device design and function increasingly overlap. Diodes, uncontrollable two terminal devices in nature, are normally used to rectify power and protect devices throughout all power levels. Diode manufacturers commonly use in life-time control techniques to improve diode characteristics and performance for faster, softer and more rugged devices.

Thyristors and transistors are both 3 terminal devices, controllable when switching between the forward blocking and forward conduction states. Thyristors are typically used at the highest energy levels in power conditioning circuits, up to the ratings of $3-4$kA and $4.5-6$kV in *Integrated-Gate Commutated Thyristor* (IGCT) cases. However, the low average switch speed still limits the applications of thyristors. Low to medium power circuits use transistor as the main switching elements. Specifically, IGBTs, a combination of bipolar structures and MOSFETs, are increasingly used in switching power converter applications due to their superior characteristics. New generations of IGBT devices exhibit continually improved electrical characteristics, including lower on-state voltage drops, higher blocking capability, and faster switching speeds. This refinement and achievement has made IGBTs popular placements for MOSFETs, BJTs and Thyristors. As a result, power electronics in the future will be largely IGBT-based.

It is essential to carry out the rigorous performance evaluation of the power electronic equipment and their control systems before installation. In most of the electronic apparatus, there are several power electronic converters that use power semiconductor devices as switches that allow the electronic circuits to deliver the large currents during the on-state and to withstand the large voltages during the off-state of the switching. Also the converters are widely used in industry due to their inherently bi-directional power conversion capability. Most converters are based on the bridge legs, and the basic topology is to serve as a simple switch, giving the different frequency ac as the output.

There are several areas in which power electronics plays a vital role. The research areas of power electronics include:

- Interfacing with distributed energy resources such as micro turbines, fuel cells, and solar cells.

- Multilevel converters for utility applications such as static var compensation, voltage sag support, HVDC inter-tie, large variable speed drives.

- Harmonics, power quality, and power filter design.
- Hybrid electric vehicle (HEV) applications such as motor drives or dc-dc converters.
- Soft switching inverters and dc-dc converters.
- Areas like transportation and utility applications.

There have been new trends in the AC/DC and DC/AC power conversion. The pulse width modulated three-level converters have been the dominant topology in the low power and some selected medium power applications. There has been on-going research on these power converters and as technology evolves and matures various new trends and performance of the converter can be identified. Several important issues play a key role in new trends. Factors like increasing the power density, improving performance of the converter, reducing the cost of the converter, and also increasing the VA ratings of the converter. There are several ways in which these factors can be achieved such as in case of the increasing the power density; this can be realized by reducing the switching losses due to the devices by using soft-switching techniques, efficient power devices, and improving the thermal management. Next issue is increasing the performance of the system, which is achieved by reducing the total harmonic distortions, reducing the electromagnetic interference problems, and by increasing the dynamics of the system.

Recently, multilevel converters have attracted attention in medium and high power applications so as to reduce the voltage stress of power semiconductors, voltage harmonics, and electromagnetic interferences. Multilevel power conversion began with neutral point clamped inverter topology proposed in [1]. Multilevel converters incorporate a topological structure that allows a desired output voltage to be synthesized from among set of isolated or interconnected distinct voltage sources. Numerous topologies to realize the connectivity have been proposed. The general function of the multilevel converters is to synthesize a sinusoidal voltage out of several levels of dc voltages. The multilevel converter can therefore be described as a voltage synthesizer. The voltage blocking capability of each device determines the maximum voltage level output. There are several advantages of multilevel inverters such as

- Low manufacturing costs as low rating devices are used;
- Improved waveform quality as levels in the converter are increased;
- Compact modules and no transformer needed;
- Better synthesizing of the output waveforms which reduces the output filters and the rating of the passive components;
- Many possible connections are available such as single-phase, three-phase, and multi-phase connections;
- Low switching frequency yields high efficiency.

1.2 Real-time simulation

Nowadays, real time simulators are finding a wide range of applications and play an important role in the automotive, aerospace, electrical, and mechatronic industries. Prior to the advent of digital computers, the analog simulators based on scaled-down models were relied upon to emulate and predict the behavior of actual power systems. Due to their cost, complexity, and inherent inaccuracies, such simulators are now being supplanted by digital simulators that can use as the real systems. Real-time simulators play a vital role in the design, evaluation, and development of large industrial AC drives [2,3,5].

The two main components of a modern variable-speed AC drive are the power stage and the controller stage. The power stage includes a IGBT-based

Voltage Source Converter (VSC) and the induction machine, while the controller stage includes the digital controller, and a high-frequency PWM gating pattern generator. Before prototyping, the controller stage is usually subjected to several cycles of testing and re-design in an expensive facility containing the power converter, motor-generator sets, sensors, switch-gear, and other test equipments. A real-time digital simulator that can model the power stage accurately and efficiently, provides an alternate means for testing the controller in a hardware-in-the-loop (HIL) configuration [4]- [5]. This approach has the advantage of substantially reducing cost, human resources, power consumption, and physical space, while providing immunity to damage to the actual equipment due to any malfunction of the controller. However, currently available real-time simulators still suffer from modeling inaccuracies and limited computational bandwidth. The modeling of the power converter is a particular challenge due to the demands its high frequency operation places on the accuracy and precision of accounting the gating signals coming from the digital controller. This has prompted several correction algorithms [6], both off-line and real-time, to be proposed using techniques such as interpolation/extrapolation and variable simulation step-size.

1.3 FPGA technology

The Field Programmable Gate Arrays (FPGAs) is an integrated circuit design to be configured by the customer or designer after manufacturing, so called field programmable. The FPGA configuration is generally specified using a Hardware Description Language (HDL), similar to that used for an application specific integrated circuit (ASIC). FPGAs can be used to implement any logical function that an ASIC could perform. The ability to update the functionality after shipping, and the low non-recurring engineering costs relative to an ASIC design (not withstanding the generally higher unit cost), offer advantages for many applications. FPGAs contain programmable logic components called logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be wired together. The logic blocks can be configured to perform complex combinational functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

To define the behavior of the FPGA, the user provides a HDL or a schematic design. The HDL form is easier to work with when handling large structures. By using an electronic design automation tool, a technology-mapped net-list is generated. The net-list can then be fitted to the actual FPGA architecture using a process called place-and-route. The user will validate the map, place and route results via timing analysis, simulation, and other verification methodologies. Once the design and validation process is complete, the binary file generated is used to (re)configure the FPGA.

Since their introduction in the mid-1980's FPGAs have roughly doubled their capacity every year, with the current state-of-the-art devices containing in excess of $500,000$ logic cells, thus enabling large System-on-Chip (SoC) implementation. Concurrently, development in the software CAD tools and Intellectual Property (IP) cores has also kept pace allowing rapid prototyping of complex digital designs. Still designing in VHDL ensures portability and platform independence. In the area of power electronics and drives, FPGAs have hitherto been used mainly for implementing control algorithms [7, 8] and/or PWM gating pattern generators [9, 36], either as stand-alone processors or as companion processors for DSPs.

1.4 Motivation for this work

The modeling of the power converter is a particular challenge due to the demands its high frequency operation places on the accuracy and precision of accounting the gating signals coming from the digital controller. Modeling of IGBT-based VSC remains the main challenge for real-time simulation. Specifically, the VSC model should accurately reflect the device switching times (often in the nanosecond range), diode reverse recovery, switching losses, and tailing current behavior of a realistic converter. Several off-line digital simulation software tools are available, with varying degrees of modeling and simulation capabilities, such as SABER, PSCAD/EMTDC, MATLAB/SIMULINK, EMTP-RV, PSS/E, SPICE and many others. However, one of their main drawbacks of off-line simulation is that they often lack the capability of interfacing with actual hardware, such as a digital controller or a protective relay. On the other hand, currently available real-time simulators have yet to reach such level of detail due to modeling and bandwidth limitations. A real-time digital simulator with adequate computational bandwidth can overcome this obstacle.

This thesis proposes to use the FPGA as the computational engine to simulate a variable speed AC drive in real-time. The FPGA-based real-time simulator models the complete AC drive including a 3-level 12-pulse VSC, a pulse-width modulator, the induction machine, and the field oriented controller. The Stratix and Stratix III FPGAs from ALTERA are used in this implementation.

The power electronic converter model is computed at a fixed time-step, albeit, at an extremely small step-size (of the order of a few nanoseconds). The advantages of this approach include: (1) freedom from reliance on complicated correction algorithms, (2) enables high-frequency switching (of the order of hundreds of KHz), (3) detailed representation of the device switching characteristics, and (4) ability to interface the model to a large-scale real-time simulator, such as a PC-cluster [13], modeling a larger and more complex host power system in which the power electronic converter is embedded; potential applications also include modeling of multi-pulse FACTS and HVDC systems.

1.5 Thesis overview

The contents of this thesis can be summarized as shown in the Fig. 1.1.

Chapter 2 presents an overview of FPGA technology and issues related to

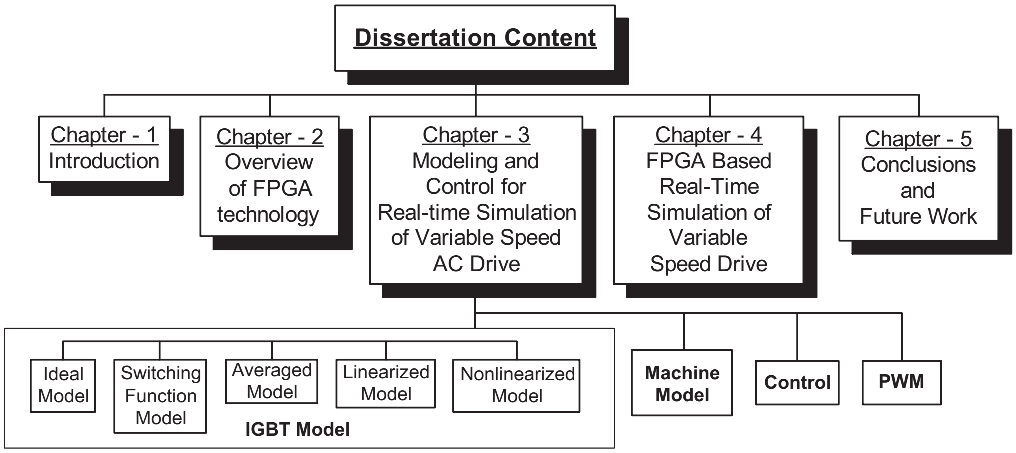

Dissertation Content
Chapter - 1 Introduction
Chapter - 2 Overview of FPGA technology
Chapter - 3 Modeling and Control for Real-time Simulation of Variable Speed AC Drive
Chapter - 4 FPGA Based Real-Time Simulation of Variable Speed Drive
Chapter - 5 Conclusions and Future Work
Ideal Model
Switching Function Model
Averaged Model
Linearized Model
Nonlinearized Model
IGBT Model
Machine Model
Control
PWM

Figure 1.1: Dissertation Content.

digital hardware implementation.

Chapter 3 presents the modeling of individual components of a variable speed AC drive: power electronic converter, machine, pulse-width modulation and control. Five different IGBT model are presented for modeling the 3-level voltage source converter.

Chapter 4 presents the implementation of the complete AC drive on two FPGA platforms. Real-time oscilloscope results are presented to confirm the accuracy of implementation. Off-line results from MATLAB/SIMULINK and SABER software are shown to validate the real-time results.

Chapter 5 presents the conclusions and suggestions for extending this work in the future.

2
Overview of FPGA technology

2.1 Introduction

In the market, a number of different types of programmable logic devices are available for use in systems as shown in Fig. 2.1. These include *Field Programmable Gate Arrays* (FPGAs) and *Application Specific Integrated Circuits* (ASICs). FPGAs offer many advantages over ASICs: small development overhead, no NRE (non-recurring engineering) cost, quick time to market, no minimum quantity order, and reprogrammability. FPGAs are basically arrays of logic gates that can be configured to perform a set of functions. Modern FPGAs have the equivalent of millions of logic gates and may also include other on-chip resources such as PowerPC.

Since the mid-1980s, FPGAs have become a dominant implementation medium for digital systems. During this time, FPGAs have grown in complexity as their ability to implement system functions has increased from a few thousands logic gates to tens of millions of logic gates. The largest FPGAs currently available exceed a billion transistors. The ability to program, or *con-*

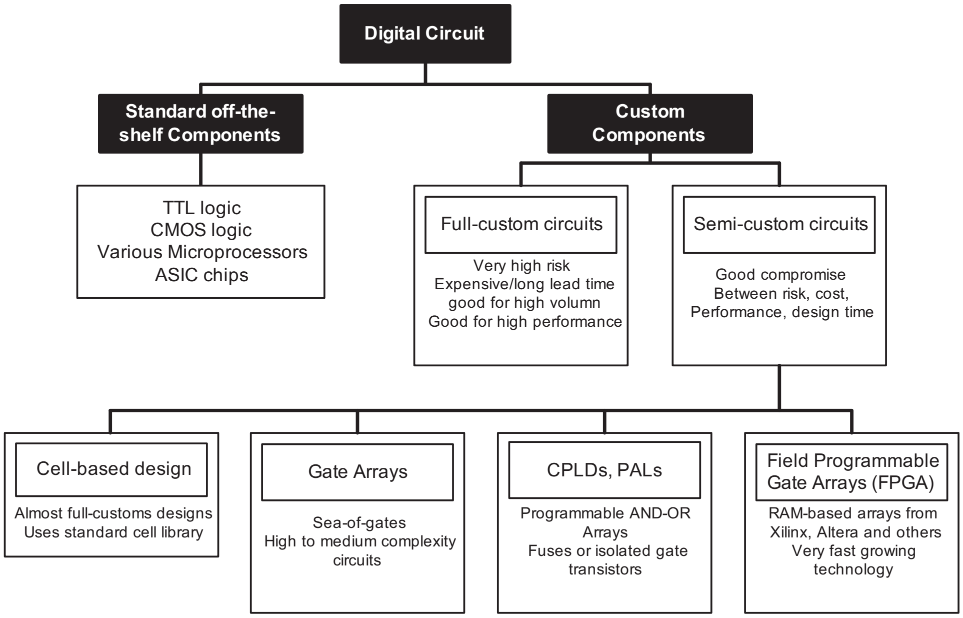

Digital Circuit
Standard off-the-shelf Components
Custom Components
TTL logic
CMOS logic
Various Microprocessors
ASIC chips
Full-custom circuits
Very high risk
Expensive/long lead time
good for high volumn
Good for high performance
Semi-custom circuits
Good compromise
Between risk, cost,
Performance, design time
Cell-based design
Almost full-customs designs
Uses standard cell library
Gate Arrays
Sea-of-gates
High to medium complexity circuits
CPLDs, PALs
Programmable AND-OR Arrays
Fuses or isolated gate transistors
Field Programmable Gate Arrays (FPGA)
RAM-based arrays from Xilinx, Altera and others
Very fast growing technology

Figure 2.1: Implementation Technology.

figure, the FPGA to perform virtually any digital logic function provides an attractive choice, not only for rapid prototyping of digital systems but also for low-to-moderate volume or fast time to market systems. The ability of *reconfigure* FPGAs to perform different digital functions without modifying the physical system facilitates the implementation of the advanced system applications such as adaptive computing and fault tolerance.

2.2 Field Programmable Gate Arrays

Field Programmable Gate Arrays (FPGAs) are of interest for use in digital signal processing systems due to their ability to implement custom hardware solutions while still maintaining flexibility through device reprogramming. Commercially available FPGAs from venders such as Altera and Xilinx, are generally used devices which were developed for applications containing varying amount of data path and control logic. FPGAs come in variety of sizes and features, which continue to change with advances in integrated circuit fabrication technology and system applications. As a result, it is difficult to write a comprehensive treatise on FPGAs because the material tends to become obsolete shortly after publication.

FPGA is similar to a programmable logic device (PLD), but whereas PLDs are generally limited to hundreds of gates, FPGAs support millions of gates. They are especially popular for prototyping integrated circuit designs. Once the design is set, hardwired chips are produced for faster performance. Field Programmable Gate Arrays (FPGAs) are divided into two major categories:

1. Reprogrammable

   - EPROM

   - EEPROM

   - Flash

   - SRAM based FPGAs (volalite)

2. One time programmable

   - Fuses (destroy internal links with current)

   - Anti-fuse based FPGAs (grow internal links)

   - PROM

SRAM FPGAs have logical connections (effectively memory cells) between the horizontal and vertical routes. SRAM FPGAs that are mostly manufactured by Altera, Lucent and Xilinx can be reconfigured many times, including in operational use. The configurations are stored in the static SRAM. FPGAs are needed to be reprogrammed every time when the chip is powered up. EPRAM FPGAs are manufactured by Altera and the configurations are stored in EPROM. The configuration remains even in the EPROM when the power is switched off. It can be also reprogrammed (via JTAG interface) into the board.

Anti-Fuse FPGAs are created by making physical connections between the horizontal and vertical routes on the FPGA; once created, anti-fuse FPGAs cannot be re-programmed. They are mostly manufacturing from Actel and QuickLogic.

Modern FPGAs that include microprocessors are used as the *System-on-Chip* (SoC) in computer architecture. They are typically described in a high level language, such as *VHDL, VeriLog, SystemC*, etc. Tool support, to turn the design and verify the layout, is provided by the device manufacturers.

2.2.1 Architecture

FPGAs generally consist of a two-dimensional array of *programmable logic blocks* (PLBs) interconnected by a programmable routing network with programmable *input/output* (I/O) cells at the periphery of the device, as shown in Fig. 2.2. Each PLB usually consists of one or more look-up tables (LUTs) and flip-flops.

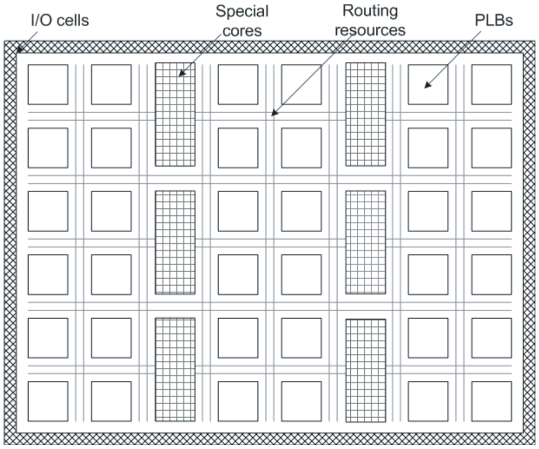


Figure 2.2: Typical FPGA architecture.

2.2.2 SRAM based FPGA

There is a wide range of FPGAs provided by many semiconductor vendors including *Xilinx*, *Altera*, *Atmel*, and *Lattice*, etc. Each manufacturer provides each own unique architecture. A typical FPGA consists of an array of logic elements and programmable routing resources used to provide the connectivity between the *logic elements* (LEs), I/O pins, and other resources such as *on-chip memory.* The structure and complexity of the logic elements, as well as the organization and functionality supported by the interconnection hierarchy, is what distinguishes the different devices from each other.

Other features such as *block memory* and *delay-locked-loop* (DLL) technology are also significant factors that influence the complexity and performance of an algorithm implemented using FPGAs. A logic element usually consists of one or more RAM based on input *look-up tables* (LUTs) where n is a number between three and six, and one or more flip-flops. LUTs are used to implement combinational logic. A typical logic element is shown in Fig. 2.3.

There may also be additional hardware support in each logic element to enable other high speed arithmetic and logic operations. A typical SRAM based FPGA architecture is shown in Fig. 2.4. The bold lines indicate how connections among two or more logic elements and I/O ports can be made.

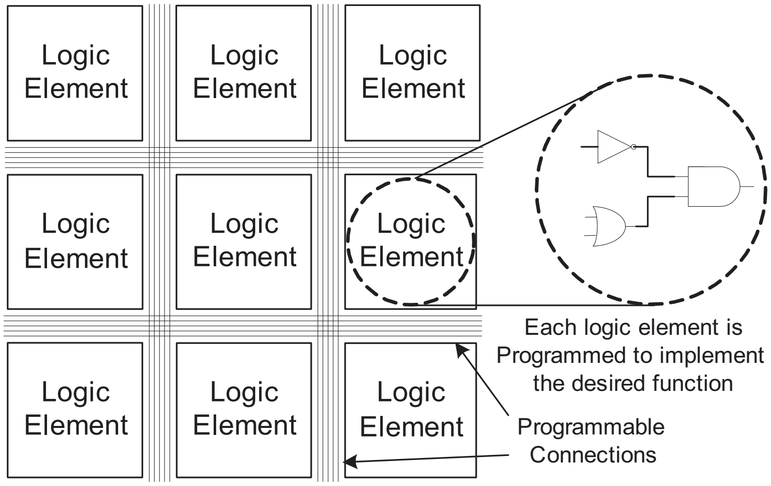


Figure 2.3: Configurable Logic Block (CLB) [23].

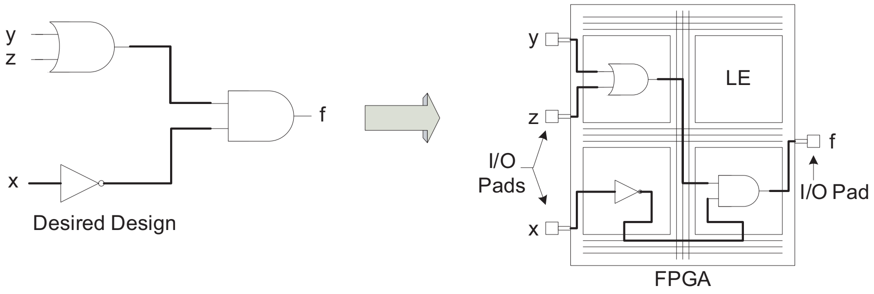


Figure 2.4: Basic FPGA Diagram [23].

2.2.3 Hardwired logic blocks

The another part will explore the use of hardwired logic blocks (HLBs) within the context of logic clusters. HLBs consist of two or more BLEs connected together by wires (shown in Fig. 2.5). These wires do not have switches and lack any form of programmability. The low resistance and capacitance of the metal wires makes the connections between BLEs fast and cheap in terms of area. In a non-HLB architecture (shown in Fig. 2.6 and Fig. 2.7) connections between BLEs must propagate through the local routing crossbar which connects all BLE outputs and inputs. The area requirements of the full-routing crossbar can be quite significant sometimes even larger than the BLE area. Also, there is a significant delay required for signals to reach from a BLE output to another BLE input. The use of HLBs alleviates the area and delay demands of local BLE connections and may improve FPGA density and performance. We explore the area and delay of one particular HLB based FPGA architecture, again in the context of clustered architectures.

2.3 Stratix and Stratix III devices

There are two types of the available boards in RTX-LAB. One is the Stratix EP1S80B956-6 DSP development board and the another is the Stratix III EP3SL150F1152C2. The new board is much faster than the previous board. Both these boards have been used in this thesis.

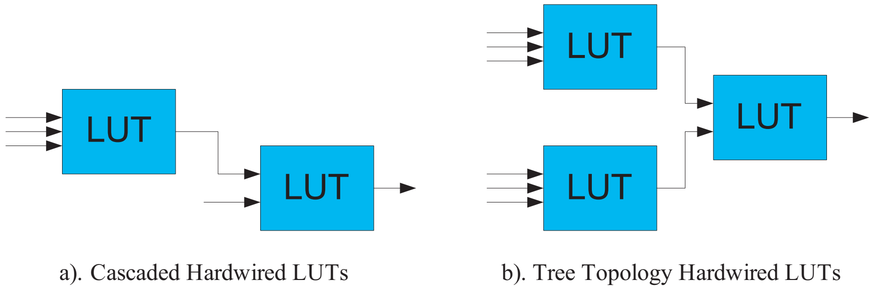


Figure 2.5: Examples of Hardwired Logic Blocks.

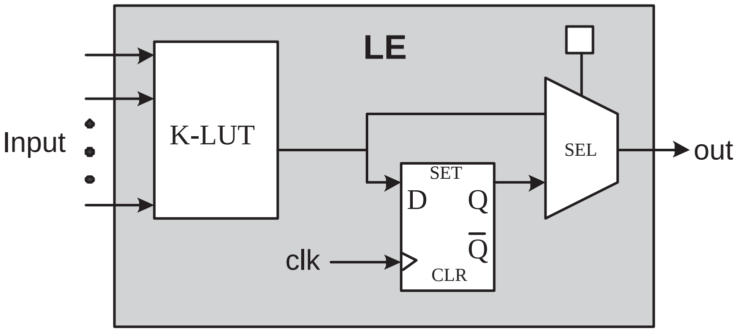


Figure 2.6: Basic Logic Element.

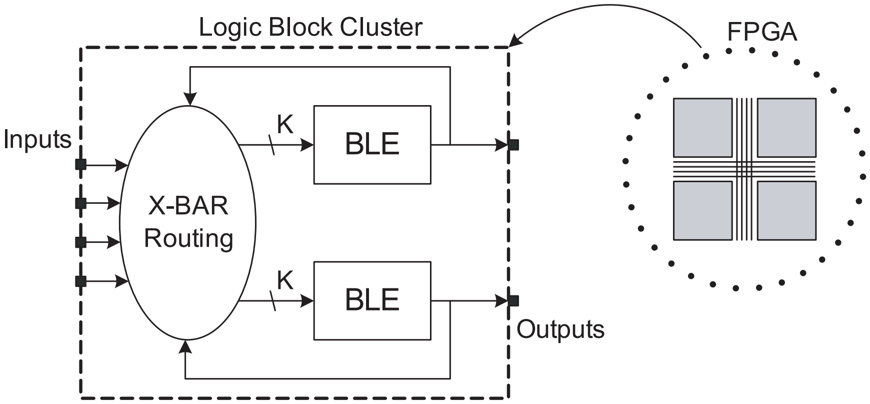


Figure 2.7: FPGA Cluster-style Logic Block Contents.

2.3.1 Stratix architecture

Stratix devices contain a two-dimensional row- and column-based architecture to implement custom logic. A series of column and row interconnects of varying length and speed provide signal interconnects between logic array blocks

(LABs), memory block structures, and DSP blocks.

Logic Array Blocks (LABs)

Each LAB consists of 10 LEs, LE carry chains, LAB control signals, local interconnect, LUT chain, and register chain connection lines. The local interconnect transfers signals between LEs in the same LAB. LUT chain connections transfer the output of one LEs LUT to the adjacent LE for fast sequential LUT connections within the same LAB. Register chain connections transfer the output of one LEs register to the adjacent LEs register within an LAB. The Quartus II Compiler places associated logic within an LAB or adjacent LABs, allowing the use of local, LUT chain, and register chain connections for performance and area efficiency. Fig. 2.10 shows the Stratix LAB. The logic array consists of LABs, with 10 logic elements (LEs) in each LAB. An LE is a small unit of logic providing efficient implementation of user logic functions. LABs are grouped into rows and columns across the device.

M512 RAM blocks are simple dual-port memory blocks with 512-bits plus parity (576-bits). These blocks provide dedicated simple dual-port or single-port memory up to 18-bits wide at up to 318 MHz. M512 blocks are grouped into columns across the device in between certain LABs.

M4K RAM blocks are true dual-port memory blocks with 4K bits plus parity (4,608 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 36-bits wide at up to 291 MHz. These blocks are grouped into columns across the device in between certain LABs.

M-RAM blocks are true dual-port memory blocks with 512K bits plus parity (589,824 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 144-bits wide at up to 269 MHz. Several M-RAM blocks are located individually or in pairs within the devices logic array.

Digital signal processing (DSP) blocks can implement up to either eight full-precision 9x9-bit multipliers, four full-precision 18x18-bit multipliers, or one full-precision 36x36-bit multiplier with add or subtract features. These blocks also contain 18-bit input shift registers for digital signal processing applications, including *finite impulse response* (FIR) and *infinite impulse response* (IIR) filters. DSP blocks are grouped into two columns in each device. Fig 2.8 shows one of the columns with surrounding LAB rows.

Each Stratix device I/O pin is fed by an *I/O element* (IOE) located at the end of LAB rows and columns around the periphery of the device. I/O pins support numerous single-ended and differential I/O standards. Each IOE contains a bidirectional I/O buffer and six registers for registering input, output, and output-enable signals. When used with dedicated clocks, these registers provide exceptional performance and interface support with external memory devices such as **DDR SDRAM**, **FCRAM**, **ZBT**, and **QDR SRAM** devices.

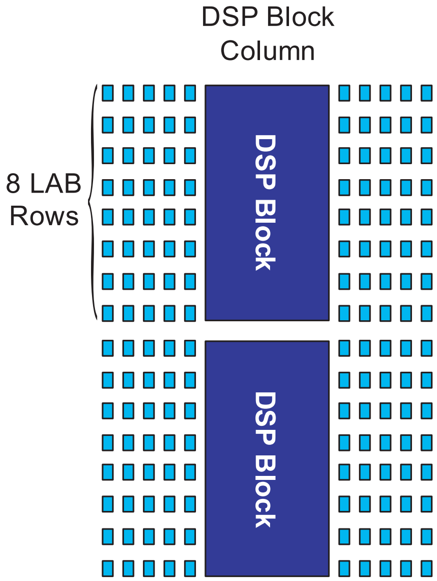


Figure 2.8: DSP Blocks Arranged in Columns.

LAB interconnects

The LAB local interconnect can drive LEs within the same LAB. The LAB local interconnect is driven by column and row interconnects and LE outputs within the same LAB. Neighboring LABs, M512 RAM blocks, M4K RAM blocks, or DSP blocks from the left and right can also drive an LABs local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each LE can drive 30 other LEs through fast local and direct link interconnects. Fig. 2.9 and Fig. 2.10 show the direct link connection and logic array blocks.

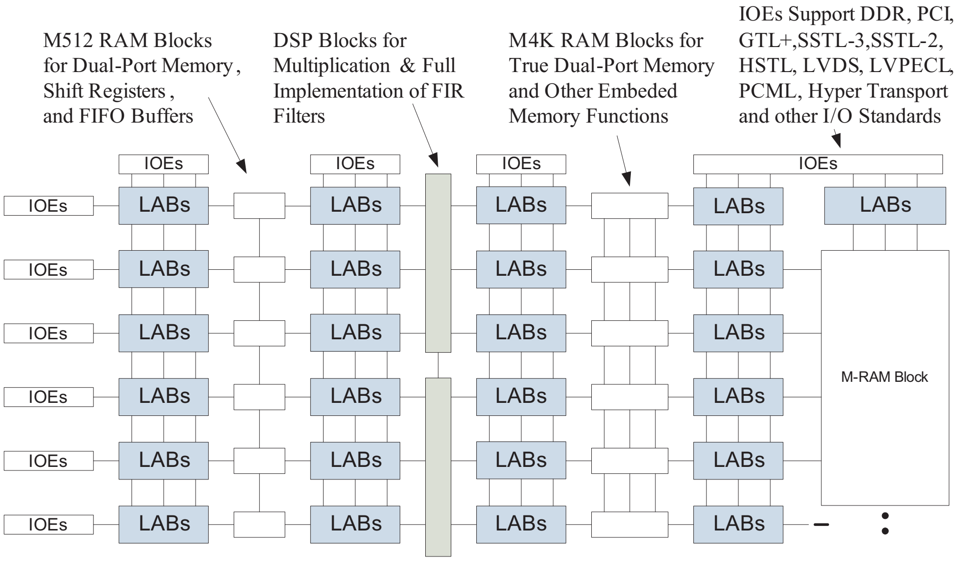

M512 RAM Blocks for Dual-Port Memory, Shift Registers, and FIFO Buffers
DSP Blocks for Multiplication & Full Implementation of FIR Filters
M4K RAM Blocks for True Dual-Port Memory and Other Embeded Memory Functions
IOEs Support DDR, PCI, GTL+,SSTL-3,SSTL-2, HSTL, LVDS, LVPECL, PCML, Hyper Transport and other I/O Standards
IOEs
LABs
M-RAM Block

Figure 2.9: Stratix Block Diagram [23].

LAB control signals

Each LAB contains dedicated logic for driving control signals to its LEs. The control signals include two clocks, two clock enables, two asynchronous clears, synchronous clear, asynchronous preset/load, synchronous load, and add/subtract control signals. This gives a maximum of 10 control signals at

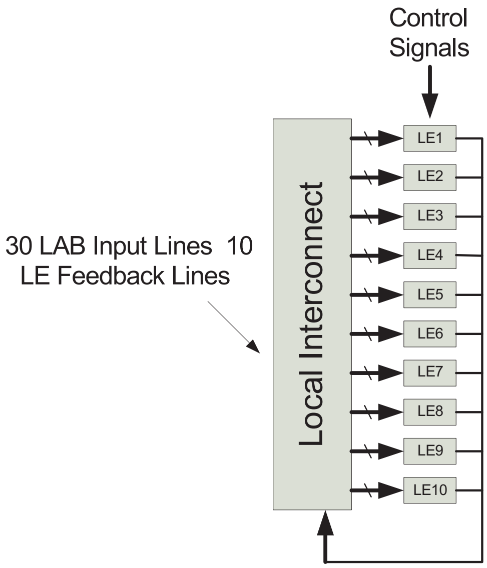


Figure 2.10: Logic Array Blocks.

a time. Although synchronous load and clear signals are generally used when implementing counters, they can also be used with other functions.

Each LAB can use two clocks and two clock enable signals. Each LABs clock and clock enable signals are linked. For example, any LEs in a particular LAB using the *labclk1* signal will also use *labclkena1*. If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock signals. De-asserting the clock enable signal will turn off the LAB-wide clock.

Each LAB can use two asynchronous clear signals and an asynchronous load/preset signal. The asynchronous load acts as a preset when the asynchronous load data input is tied high.

With the LAB-wide *addnsub* control signal, a single LE can implement a one-bit adder and subtractor. This saves LE resources and improves performance for logic functions such as DSP correlations and signed multipliers that alternate between addition and subtraction depending on data.

The LAB row clocks [7..0] and LAB local interconnect generate the LAB-wide control signals. The *MultiTrack* interconnects inherent low skew allows clock and control signal distribution in addition to data. Fig. 2.11 shows the

LAB control signal generation circuit.

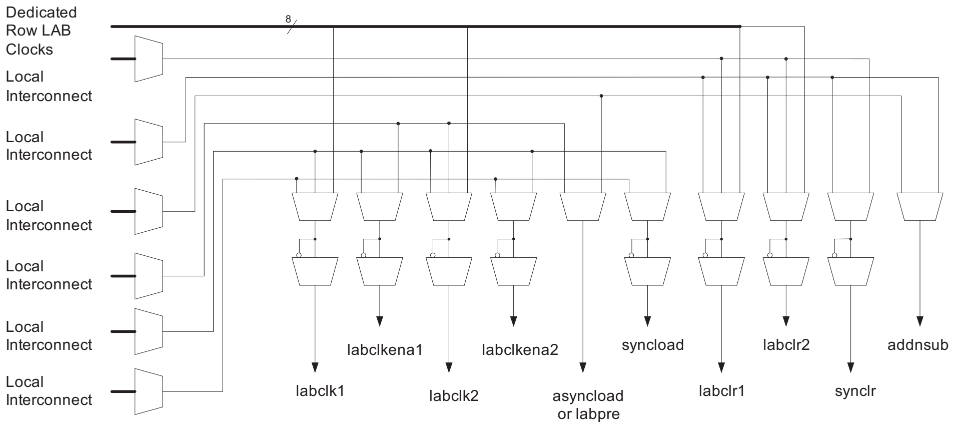


Figure 2.11: Logic Array Blocks control signals [23].

Logic Elements (LEs)

Fig. 2.12 shows the Stratix logic element (LE). The LE is the smallest unit of logic in the Stratix architecture. It is compact and provides advanced features with efficient logic utilization. Each LE contains a four-input LUT, which is a function generator that can implement any function of four variables. In addition, each LE contains a programmable register and carry chain with carry select capability. A single LE also supports dynamic single bit addition or subtraction mode selectable by an LAB-wide control signal. Each LE drives all types of interconnects: local, row, column, LUT chain, register chain, and direct link interconnects. As we can see in Fig. 2.13.

Each LEs programmable register can be configured for D, T, JK, or SR operation. Each register has data, true asynchronous load data, clock, clock enable, clear, and asynchronous load/preset inputs. Global signals, general-purpose I/O pins, or any internal logic can drive the registers clock and clear control signals. Either general-purpose I/O pins or internal logic can drive the clock enable, preset, asynchronous load, and asynchronous data. The asynchronous load data input comes from the *data3* input of the LE. For combinatorial functions, the register is bypassed and the output of the LUT drives directly to the outputs of the LE.

Each LE has three outputs that drive the local, row, and column routing resources. The LUT or register output can drive these three outputs independently. Two LE outputs drive column or row and direct link routing connections and one drives local interconnect resources. This allows the LUT to drive one output while the register drives another output. This feature, called register packing, improves device utilization because the device can use the register and the LUT for unrelated functions. Another special packing mode allows the register output to feed back into the LUT of the same LE so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The LE can also drive out registered and unregistered versions of the LUT output.

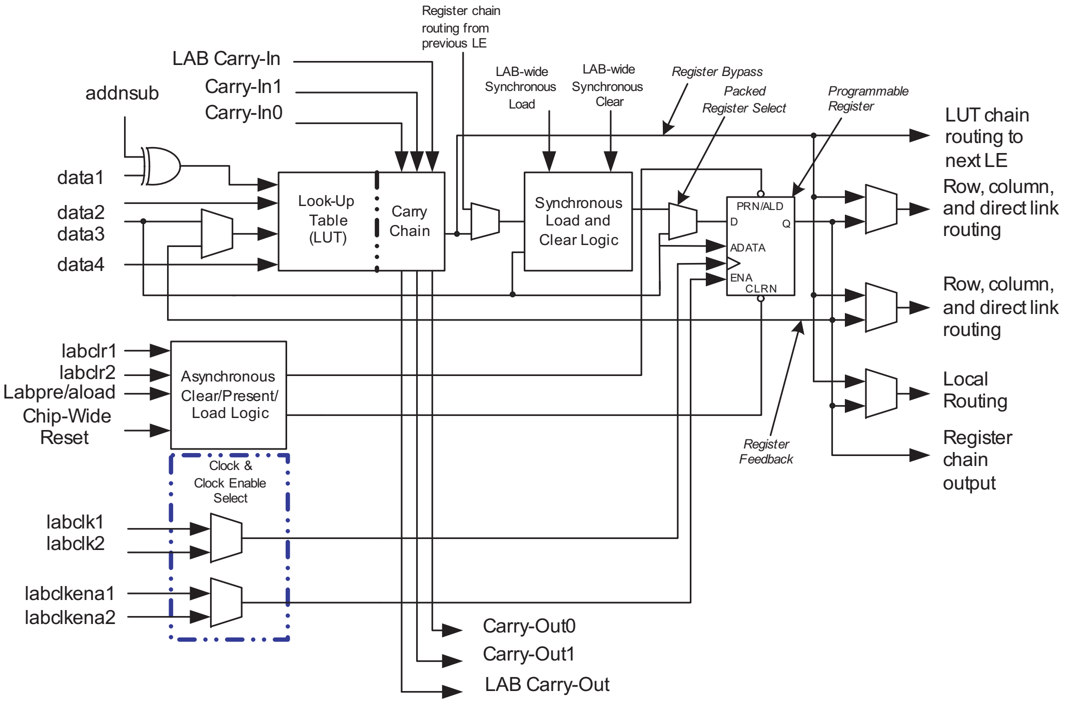


Figure 2.12: Stratix Logic Element [23].

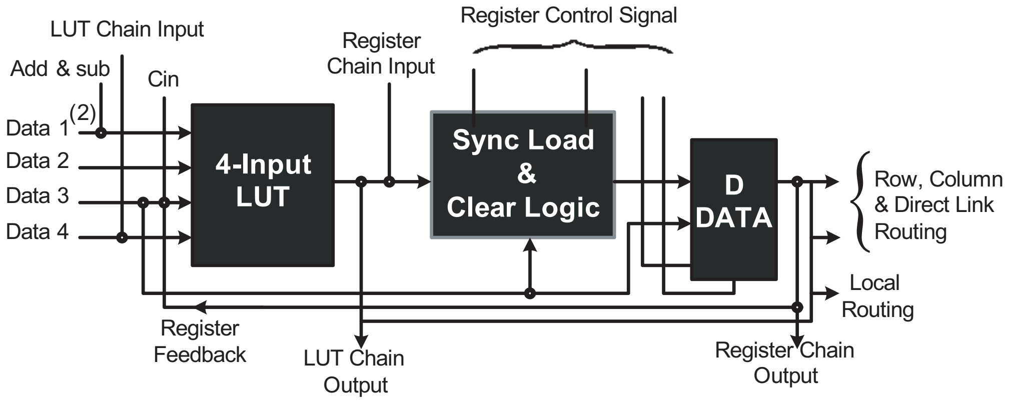


Figure 2.13: Typical logic element (*LE*) (normal mode) in a SRAM based FPGA [23].

LUT chain and register chain

In addition to the three general routing outputs, the LEs within an LAB have LUT chain and register chain outputs. LUT chain connections allow LUTs within the same LAB to cascade together for wide input functions. Register chain outputs allow registers within the same LAB to cascade together. The register chain output allows an LAB to use LUTs for a single combinatorial function and the registers to be used for an unrelated shift register implementation. These resources speed up connections between LABs while saving local interconnect resources.

addnsub signal

The LEs dynamic adder/subtractor feature saves logic resources by using one set of LEs to implement both an adder and a subtractor. This feature is controlled by the LAB-wide control signal *addnsub*. The *addnsub* signal sets the LAB to perform either A-B or A+B. The LUT computes addition, and subtraction is computed by adding the twos complement of the intended subtractor. The LAB-wide signal converts to twos complement by inverting the B bits within the LAB and setting carry-in = 1 to add one to the *least significant bit* (LSB). The LSB of an *adder/subtractor* must be placed in the first LE of the

LAB, where the LAB-wide *addnsub* signal automatically sets the *carry-in* to 1. The *Quartus II Compiler* automatically places and uses the adder/subtractor feature when using adder/subtractor parameterized functions.

LE operating modes

The Stratix LE can operate in one of the following modes:

1. Normal mode.

2. Dynamic arithmetic mode.

Each mode uses LE resources differently. In each mode, eight available inputs to the LE the four data inputs from the LAB local interconnect; *carry-in0* and *carry-in1* from the previous LE; the LAB carry-in from the previous carry-chain LAB; and the register chain connection are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, asynchronous preset load, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all LE modes. The *addnsub* control signal is allowed in arithmetic mode.

The Quartus II software, in conjunction with parameterized functions such as *library of parameterized modules* (LPM) functions, automatically chooses the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions. If required, you can also create special-purpose functions that specify which LE operating mode to use for optimal performance.

The normal mode is suitable for general logic applications and combinatorial functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT (see Fig. 2.14). The Quartus II Compiler automatically selects the carry-in or the *data3* signal as one of the inputs to the LUT. Each LE can use LUT chain connections to drive its combinatorial output directly to the next LE in the LAB. Asynchronous load data for the register comes from the *data3* input of the LE. LEs in normal mode support packed registers.

The final step in the design process is automatic place and route (APR) for an FPGA or ASIC. FPGA vendors generally supply their own APR tools. Reconfigurable devices such as Field Programmable Gate Arrays (FPGAs) can be considered an intermediate option. FPGAs provide *Configurable Logic Blocks* (CLBs) and routing resources that are programmable. Since the design can be tested and verified at the user site, it benefits from a less expensive design process than ASICs. FPGAs can provide very high flexibility and they can produce efficient devices with respect to speed, area, and power consumption relatively. The need for flexibility in case of modification or damage can be crucial in some cases; for instance in cosmic equipment or in satellites, cosmic rays might affect electronic devices, the capability to reprogram the devices remotely could be of extreme importance.

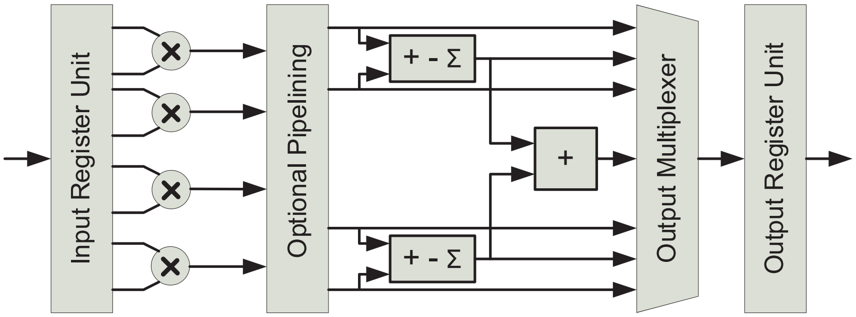


Figure 2.14: Stratix DSP Block [23].

QUARTUS II 9.0 development software

The QUARTUS II 9.0 development software contains many of the features available in the commercial version, including a completely integrated design flow and intuitive graphical user interface. This software supports schematic capture and text-based hardware description language design entry, including the Altera Hardware Description Language (AHDL) and VHDL. It also provides design programming, compilation, and verification support.

ByteBlaster parallel port download cable

Designs can be easily and quickly downloaded into the Stratix EP1S80 DSP development board using the Byte-Blaster download cable, which is hardware interface to a standard parallel port. This cable channels programming or configuration data between the Quantus II software and the EP1S80 Board. Because design changes are downloaded as the SRAM object file (.sof) directly to the devices on the board, prototyping is easy and multiple design iterations can be accomplished in quick succession.

On-board voltage regulator

The on-board voltage regulator, an LM340T, regulates the DC positive input at 5V. The DC input consists of two holes for connecting a 5-V DC regulated power source. The hole marked with a plus sign (+) is the positive input; the hole marked with a minus sign (-) is board common. A green light-emitting diode (LED) labeled POWER is illuminated when current is flowing from the 5-V DC regulated power source.

JTAG-in header

The 10-pin female plug on the ByteBlaster download cable connects with the JTAG-in 10-pin male header on the EP1S80 Education Board. The EP1S80 Education Board provides power and ground to the ByteBlaster download cable. Data is shifted into the devices via the TDI pin and shifted out of the devices via the TDO pin. Table 2.1 below identifies the JTAG IN pin names when the Byte-Blaster is operating in JTAG mode.

Table 2.1: JTAG IN 10-Pin Header Pin-Outs.

Pin	JTAG Signal
1	TCK
2	GND
3	TDO
4	VCC
5	TMS
6	No Connect
7	No Connect
8	No Connect
9	TDI
10	GND

2.3.2 Stratix III architecture

Stratix III FPGAs (Fig. 2.15) include adaptive logic modules (ALMs) and high performance, flexible memory and DSP blocks to meet your system's most demanding requirements. The I/O banks and external memory interfaces make design your boards easier, and design security features keep your designs secure.

The features include:

1. Up to 150k equivalent logic elements, 17-Mbits of memory, and 576 multipliers in the L variant for the industry's biggest FPGA.

2. 25% higher performance and 50% lower power than previous generation FPGAs.

3. Up to 896 18-bit x 18-bit multipliers operating at 550MHz in the E variant.

4. Up to 24 high-performance I/O banks.

5. Best in class signal integrity.

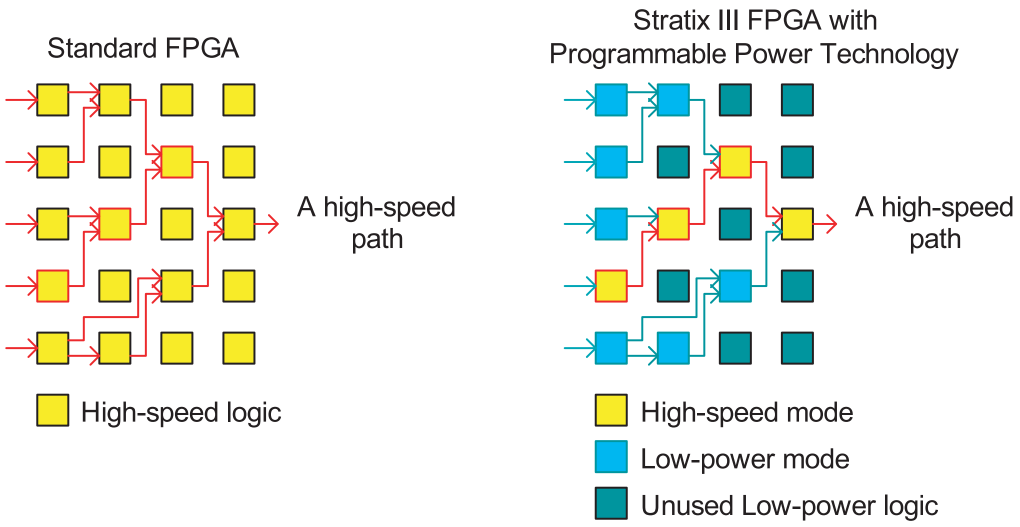


Figure 2.15: Standard FPGA Logic vs. Stratix III FPGA Logic [23].

2.3.3 Flexible, efficient ALMs

Stratix III FPGAs are based on the innovative adaptive logic modules (Fig. 2.16) that Altera introduced. ALMs are based on 8-input 'fracturable' look-up table (LUT) with two dedicated address and two registers.

The fracturable look-up table (LUT) allows ALMs to pack more functionality into the same amount of the logic. ALMs can implement six-input functions, some 7-input functions, or multiple other combinations of functions with vary numbers of inputs. ALMs are the basis of the inductor's most efficient and highest-performance logic architecture.

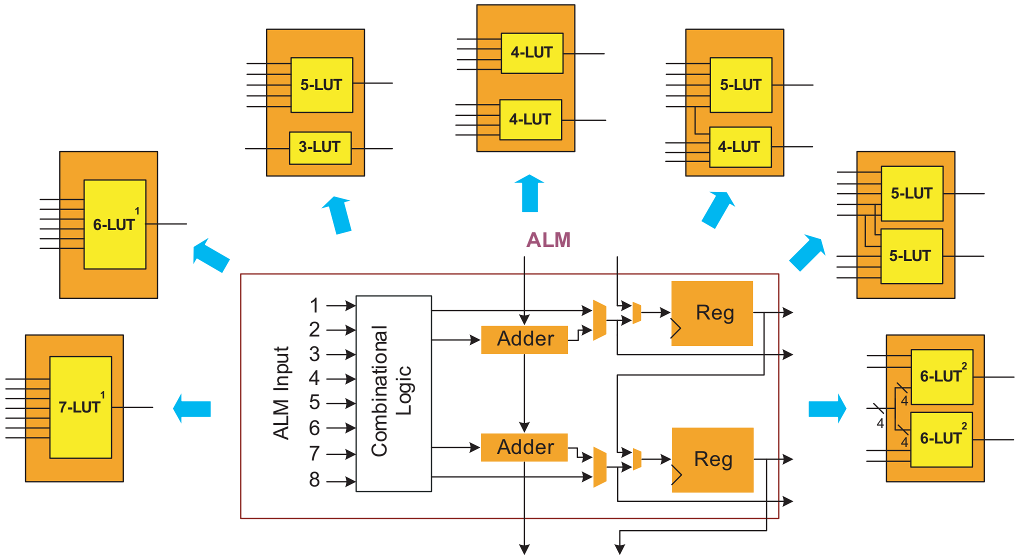
ALM details:

1. 2:1 register-to-LUT ratio in ALMs assures that the FPGA is not register-limited for register-rich design.

2. For DSP applications, ALMs include 2 dedicated adders capable of 2-bits additions, or a single ternary adder for advanced arithmatic computations.

3. Ten ALMs in a single LAB can be converted into a 640-bit MLAB mem-

ory block to get more data ports for greater memory band-width.



Notes: 1 - Stratix III ALM can implement a subset of 7-input functions
2 - Must have the same logical function

Figure 2.16: ALM block diagram and example LUT configurations [23].

2.3.4 TriMatrix memory for the perfect fit

TriMatrix memory (Fig. 2.17) is more flexible and efficient and provides higher memory bandwidth than any other FPGA memory architecture. Stratix III includes three sizes of memory blocks: MLAB blocks, M9K blocks and M144K blocks.

TriMatrix memory details:

1. Provides up to 17 Mbits of memory performing at over 600MHz.
2. Includes features such as dual-port RAM mode, error correction coding (ECC), and low power mode.
3. Stratix III E devices include the most memory and DSP blocks for memory-intensive applications.

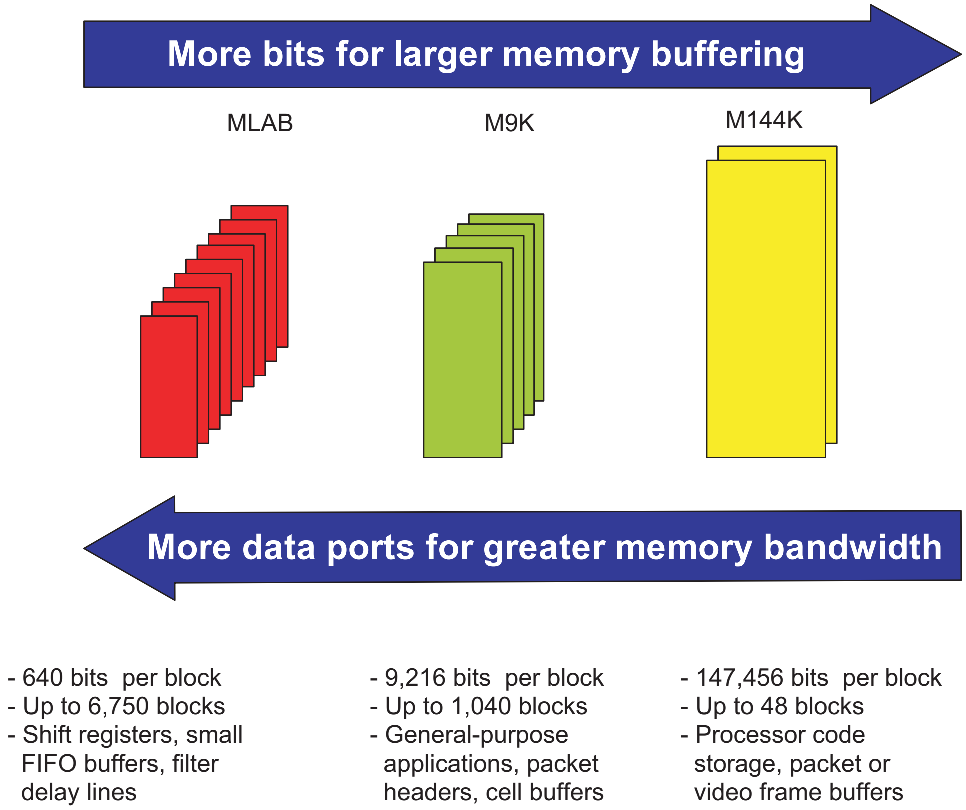

More bits for larger memory buffering
MLAB
M9K
M144K
More data ports for greater memory bandwidth
- 640 bits per block
- Up to 6,750 blocks
- Shift registers, small FIFO buffers, filter delay lines
- 9,216 bits per block
- Up to 1,040 blocks
- General-purpose applications, packet headers, cell buffers
- 147,456 bits per block
- Up to 48 blocks
- Processor code storage, packet or video frame buffers

Figure 2.17: TriMatrix memory [23].

2.3.5 Highest performance digital signal processing

For the highest performance, Stratix III FPGAs include DSP block with twice the multiplier capabilities of competing devices. These DSP Block deliver up to 60 times more multiplies, accumulator processing performance than the industry's highest-performance DSP processors with lower power, lower cost, and smaller board space for equivalent processing capability. For DSP-intensive applications, the Stratix III E devices include the most DSP and memory blocks in family.

In addition, it includes application-optimized IP (such as our Nios II embedded processors and DSP IP) and development tools (DSP Builder, SoPC Builder, and C_2H Compiler) to make your video and imaging, baseband, intermediate frequency, and encryption applicants easier to design.

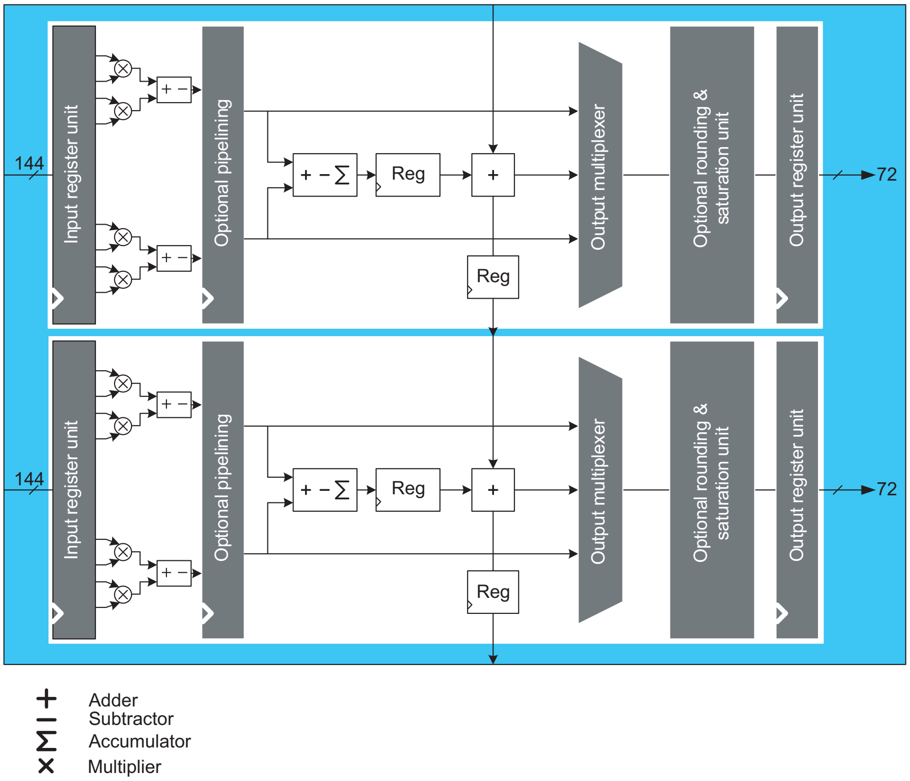


Figure 2.18: DSP Block [23].

DSP block details:

1. Variable bit-width multipliers, adders/accummulators, pipeline register, and other arithmetic operators.

2. Up to 896 18-bit x 18-bit multipliers capable of operating at up to 550MHz.

2.4 Digital signals processing (DSP)

Signals can be divided into three categories: continuous-time (*analog*) signals, discrete-time signals and digital signals. The signals daily are mostly analog signals. Discrete-time signals are only at a particular set of time instances that can be represented as a sequence of numbers that have a continuous range of values. Digital signals have discrete values in both time and amplitude. Some of the advantages of a DSP system are: flexibility, reproducibility, reliability, complexity.

With the rapid evaluation in semiconductor technologies, DSP systems have a lower overall cost compared to analog systems for most of application. DSP algorithms can be developed, analyzed and simulated using high-level language and softwares tool such as *C/C++* and MATLAB (*matrix laboratory*). The performance of the algorithms can be verified using a low-cost, general purpose computer. Therefore, a DSP system is relatively easy to design, develop, analyze, simulate, test, and maintain.

There are some limitations associated with DSP (Fig. 2.18). For instance, the bandwidth of a DSP system is limited by the sampling rate and hardware peripherals. Also, DSP algorithms are implemented using a fixed number of bits with a limited precision and dynamic range (the ratio between the largest and smallest numbers that can be represented), which results in quantization and arithmetic errors. Thus, the system performance might be different from the theoretical expectation.

2.4.1 Basic elements of DSP system

There are two types of DSP applications: nonreal-time and real-time. Nonreal-time signal processing involves manipulating signals that have already been collected in digital forms. Real-time signal processing places stringent demands on DSP hardware and software designs to complete pre-defined tasks within a certain time frame.

The basic real-time DSP system is the one in which a real-world analog signal is converted to digital signal, processed by DSP hardware, and converted back into an analog signals.

2.4.2 DSP hardware

DSP systems are required to perform intensive arithmetic operations such as multiplication and addition. These tasks may be implemented on microprocessors, micro-controllers, digital signal processors, or customer integrated circuits. The selection of appropriate hardware is determined by the applications, cost or a combination of both.

For example with the aid of Xilinx System Generation for Digital Signals Processing, a tool used to port MATLAB/Simulink model to Xilinx hardware model, a system designer can model, simulate and verify the DSP algorithms on the target FPGA hardware under the Simulink environment.

C has become the language of choice for many DSP software development engineers not only because it has powerful commands and data structures but also because it can easily be ported on different DSP processors and platforms. The processes of computation, linking/loading, and execution are outlined in Fig. 2.19.

A digital computer is a collection of logic elements that can execute arbitrary algorithms to perform data calculation and manipulation functions. A computer is composed of a microprocessor, memory, and some input/output (I/O) elements. The microprocessor, often called a *microprocessor unit* (MPU) or *central processor unit* (CPU) contains logic to step through an algorithm, called a program that has been stored in the computer's memory. The data used and manipulated by that program is help in the computer's data memory. Memory is a repository for data that is usually organized as a linear array of individually accessible locations. The microprocessor can access a particular location in memory by presenting a memory address (the index of the desired location) to the memory element. I/O elements enable the microprocessor to communicate with the outside world to acquire new data and present the results of its programmed computations. Such elements can include a keyboard or display controller. Programs are composed of many very simple

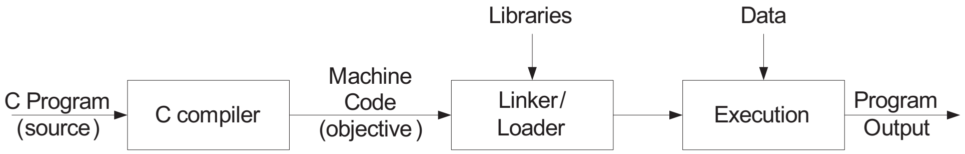


Figure 2.19: Program compilation, linking, and execution flow.

individual operations, called *instructions*, that specify in exact detail how the microprocessor should carry out an algorithm. A simple program may have dozens of instructions, whereas a complex program can have tens of millions of instructions. Collectively, the programs that run on microprocessors are called *software*, in contrast to the *hardware* on which they run. Each type of microprocessor has its own *instruction set* that defines the full set of unique, discrete operations that it is capable of executing. These instructions perform very narrow tasks that, on their own, may seem insignificant. However, when thousands or millions of these tiny instructions are strung together, they may create a word processor.

A microprocessor possesses no inherent intelligence or capability to spontaneously begin performing useful work. Each microprocessor is constructed with an instruction set that can be invoked in arbitrary sequences. Therefore, a microprocessor has the potential to perform a useful work but will do nothing of the sort on its own. To make the microprocessor perform useful work, it requires explicit guidance in the form of software programming. A task of even moderate complexity must be broken down into many tiny steps to be implemented on a microprocessor. These steps include basic arithmetic, Boolean operations, loading data from memory or an input element such as a keyboard, and storing data back to memory or an output element such as a printer.

Memory structure is one of a computer's key characteristics, because the microprocessor is almost constantly accessing it to retrieve a new instruction, load new data to operate on, or store a calculated result. While program and data memory are logically distinct classifications, they may share the same physical memory resource. *Random Access Memory* (RAM) is the term used to describe a generic memory resource whose locations can be accessed, or *addressed*, in an arbitrary order and either read or written. A *read* is the process of retrieving data from memory address and loading it into the microprocessor. A *write* is the process of storing data to a memory address from the microprocessor. Both programs and data can occupy RAM. When the computer execute a program that is located on the disk drive, that program is first loaded into the computer's RAM and then executed from a region set aside for program memory. RAM is most often *volatile* - meaning that it loses its contents when the power is turned off.

Some software cannot be stored in volatile memory, because basic initialization instructions, or *boot code*, must be present when the computer is turned on. Remember that a microprocessor can do nothing useful without software being readily available. When power is first applied to a computer, the microprocessor must be able to quickly locate boot code so that it can get itself ready to accept input from a user or load a program from an input device. This startup sequence is called *booting*, hence the term *boot code.* If the computer is turned on, the first messages that it displays on the monitor are a product of its boot code. The computer is able to access its disk drive and begins loading software into RAM as part of its normal operation. To ensure that boot code is ready at power-up, *non-volatile* memory called *read only memory* ROM exists. ROM can be used to store both programs as well as any data that must be present at power-up and immediately accessible. Software contained in ROM is also known as *firmware.*

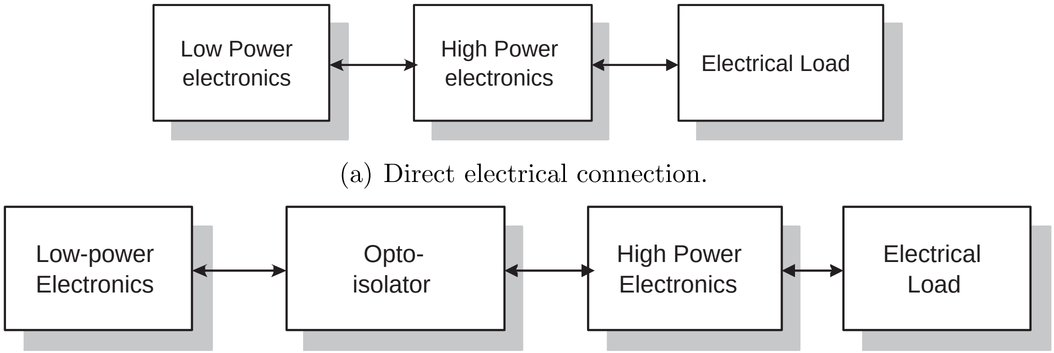
2.5 Hardware description language (HDL)

The Hardware Description Language (HDL) is a very useful language for real-time simulation on the FPGA. There are several kinds of HDL language eg., *VHDL*, *Verilog*, *SystemC*, etc. A Hardware Description Language (HDL) provides a computer like language which specifies the required digital logic. The designer uses this language to describe the system requirement. The design may be simulated and corrected as necessary. Next the designer synthesizes the circuit to either an FPGA or to a VLSI device. This uses a computer tool analogous to a software compiler which translates the HDL description into a circuit consisting of basic logic gates and flip-flops as necessary. The resulting circuit is known as a *net-list*. The synthesis process provides the designer with rapidly applied optimization techniques, techniques which will usually be more effective that manual techniques. The ease of applying these techniques also means that the designer may have time to experiment with different design approaches at a higher level.

Use of the synthesis tool also means that when we prepare a design we do not have to commit to a particular manufacturer's FPGA or silicon process. It is only when we synthesize that a particular FPGA or silicon process must be specified. This has the great advantage that the HDL designs can be easily transferred from one process to another. Indeed, this capability has opened up a new industry segment known as IP (*Intellectual Property*). A number of companies now exist solely to provide IP Cores, HDL designs of, for example, a microprocessor, or perhaps an MPEG decoder. These companies then sell this synthesizable description to designers constructing complex designs.

Field-Programmable Gate Arrays (FPGAs) have experienced tremendous growth in recent years and have become a multi-billion dollar industry. Shrinking device geometries resulting in larger gate capacity have provided for greater functionality. The instant programmability gives systems built with these devices a significant time-to-market advantage. However, this programmability comes at a price, since FPGAs are at least three times slower and demand more than ten times the silicon area when implementing the same function on a chip when compared to Standard Cells or Masked Programmable Gate

Arrays. This happens because Standard Cells use simple wires to make interconnections between logic gates but in FPGAs, gates are connected with programmable switches. These switches have much larger resistance and capacitance and hence are slower than the wires in full-fabrication chips. Ideally, to improve the performance of an FPGA, we would like to use as few switches as possible for any given circuit. In general, the three main factors affecting overall FPGA performance are the architecture of the FPGA, the quality of the CAD tools, and the electrical transistor level design of the FPGA.



(a) Direct electrical connection.

(b) Connection via an opti-isolator.

Figure 2.20: Connecting the electronics.

FPGAs provide a VLSI device consisting of a large number of logic blocks whose function is determined by bits loaded into a memory within the device. These bits determine the function of each logic block, and the interconnect between them. Loading these bit patterns is normally done when power is applied to the device by loading the bit stream from an adjacent memory chip.

Thus, the manufacturer of the FPGA uses standard VLSI techniques to produce many millions of these devices. Lower scale manufacturers then load the required bit pattern (*program the device*) to give the circuit they require within their equipment.

2.6 Summary

In this chapter, an overview of FPGA technology and its design methodology is given. There is background information related to the Altera Stratix III EP3SL150F1152C2 and Altera Stratix EP1S80B956-6 devices which are used in this research. In the next chapter, we will present the modeling of various components of a variable speed AC drive for real-time simulation.

3 Modeling and Control of a Variable Speed AC Drive

3.1 Introduction

With the advent of the power semiconductor device since *1950*'s, the power electronics area has grown extensively in industrial, transportation, residential, commercial and aerospace applications. Power electronics mainly deals with the efficient conversion, control and conditioning of electrical power by static means from its available input form into a desired electrical output form. The electrical power conversion can be realized by power converters built on power semiconductor switching devices controlled by control electronics. This kind of power converter is usually known as switch mode converter. The power semiconductor devices can be considered as controllable switches that can turn-on and turn-off according to the corresponding gate signal. Popular power devices include Gate-turn-off thyristor (GTO), Insulated-gate-bipolar transistor (IGBT) and MOS-controlled thyristor (MCT). Nowadays, with the increased

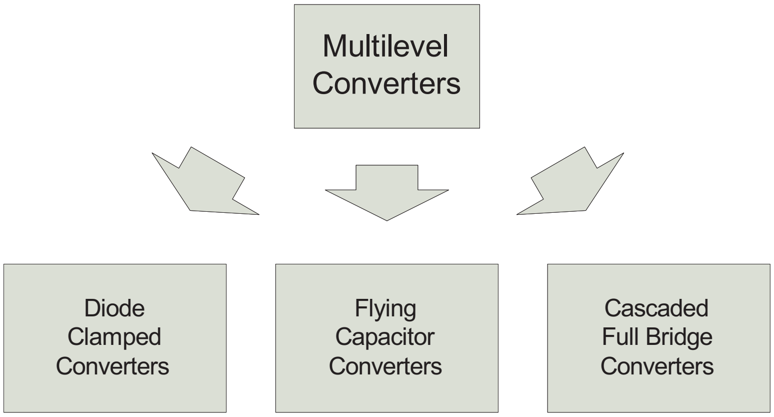


Figure 3.1: Multilevel converters topology.

power capabilities and ease of control of these modern semiconductor devices, great advances have been seen in new power converter topologies and controls. Based on the power form on the input and output side, the power converters can be divided into the following categories:

1. DC to AC (inverter).
2. AC to DC (rectifier).
3. DC to DC (chopper).
4. AC to AC (matrix converter or cycloconverter).

There are several types of multilevel (three-level and more) converters and their respective modulation strategies. Multilevel converter topologies are receiving increased attention recently, especially for use in high power applications. This increased attention is due to the fact that the output waveforms are much improved over those of the two-level converter technologies, and that the voltage rating of the converter is increased due to the series connection of the devices. Multilevel converters can also reduce the harmonics and hence improve the power quality.

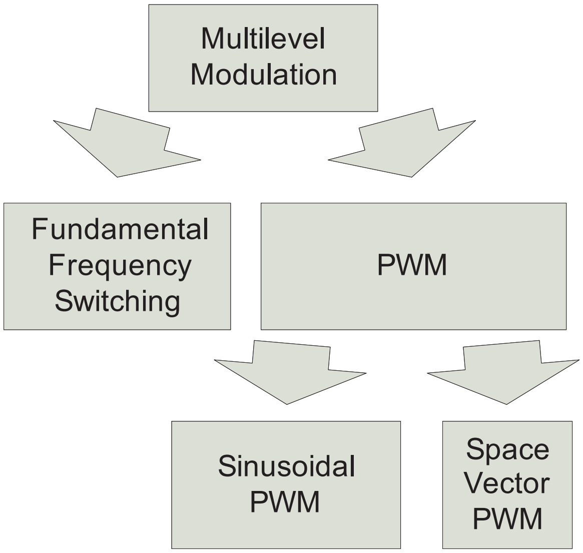


Figure 3.2: Multilevel modulation technology.

There are several kinds of multilevel converters as can be seen in Fig. 3.1. Different kinds of converters have different significance and utilization. Based on the different types of modulation strategies, we can also distinguish the types of converters as shown in Fig. 3.2. Among them, the third-harmonics sinusoidal PWM method will be used in this thesis. It is the most commonly used in the industry, today.

3.2 3-level voltage source converter

The 3-level voltage source converter (VSC) is shown in Fig 3.3. It consists of 12 IGBTs arranged in 4 IGBT's per phase-leg, 12 antiparallel diodes, and 6 clamping diodes. The DC side of the converter usually consists of 2 capacitors to provide $+V_{DC}/2$ and $-V_{DC}/2$ voltage. The AC side of the converter is connected to the load. The 12 IGBTs in the 3-level converters are switched based on specific PWM strategies. Among these strategies, sine wave with third harmonic modulation (STPWM) stands out because it offers significant flexibility to optimize switching waveforms and because it is well suited for implementa-

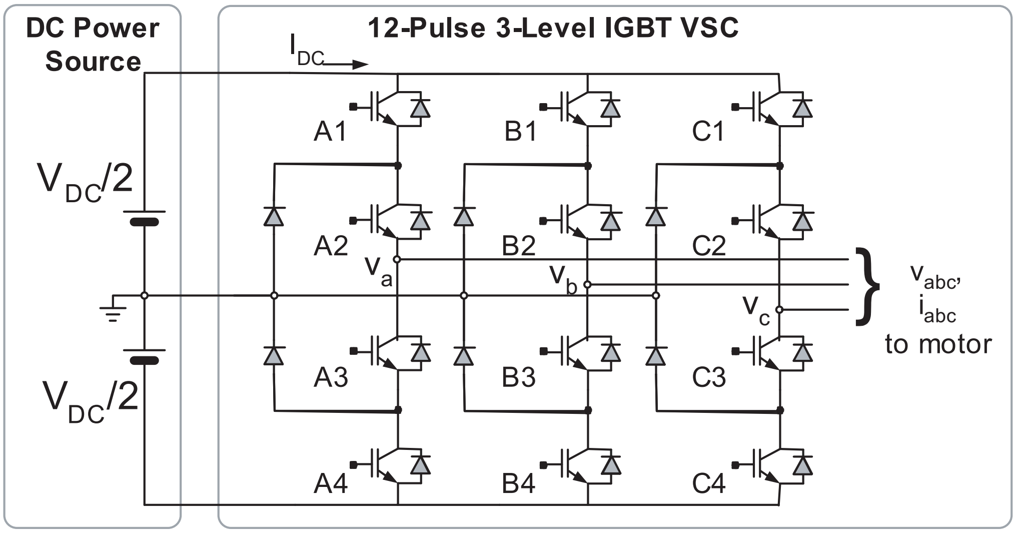


Figure 3.3: 12-pulse 3-level VSC drive.

tion on a digital computer. The modulator handles the PWM generation as well as the 3-level issues (such as the problem of load sharing between the dc sources, minimum pulse width compensation and so on).

Traditional, two-level high-frequency pulse width modulation (PWM) inverters for motor drives have several problems associated with high frequency switching, which produce common-mode voltages and high voltage change (dV/dt) rates to the motor windings and it will cause the motor aging. It will seriously effect the medium level voltage motor and the coil insulation can break down. The 3-level STPWM overcomes these disadvantages.

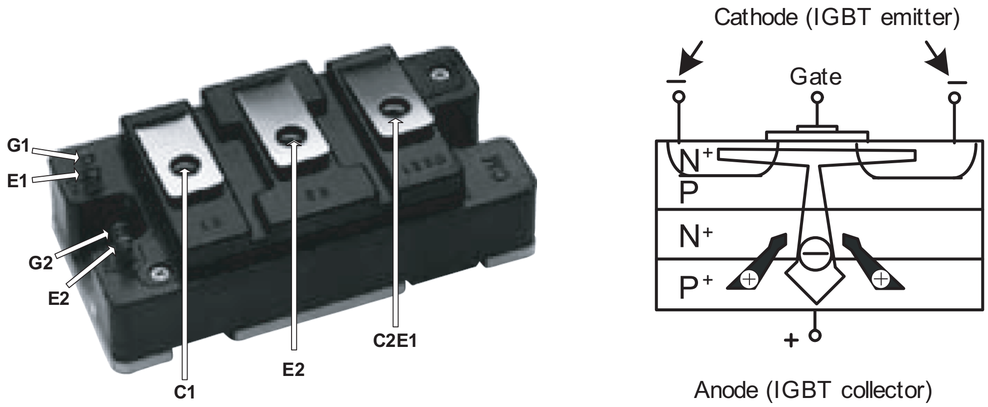
Several methods have been proposed to reduce the cost of the multilevel converter by reducing the switching losses, reducing the harmonics so that the cost of the passive components reduces, reducing the count of the devices, reducing the rating of the devices, and so on. However, the performance is limited when the reduced count of devices are used. The general idea behind the selected topology was a rearrangement of IGBTs and diodes in a three-level converter. In these topologies, the top and the bottom most devices in each leg of the inverter are replaced by clamping diodes. However, when the number of voltage levels is greater than three, for example in four-level converter, the innermost IGBTs require a rating of (2/3)Vdc, whereas the outermost IGBTs require a voltage rating of (1/3)Vdc.

3.3 Experimental measurement of IGBT electrical characteristics

To develop an accurate IGBT model for the 3-level converter, first we need to understand the switching characteristics of an IGBT. This section describes the measurement of the relevant electrical characteristics of the IGBT which include the turn-on and turn-off switching characteristics and the tailing current behavior. These measurements are based on hard switching of the IGBT. The measured IGBT characteristics are used to develop the 3-level VSC model.

Fig. 3.4 shows a snapshot of the CM50DU-24F IGBT module from *Powerex*® used for the experimental setup and the IGBT device structure. This module consists of two IGBTs and two antiparallel diodes rated at 1200V and 50A. The experimental test circuit shown in Fig. 3.5 utilized this module to measure the IGBT characteristics. The 250mH inductor in the circuit was chosen carefully to ensure the negligible current ripple, and the 1800μF capacitor across the DC source was selected to smooth the ripple for a supplied DC voltage. The IGBT2 is the Device Under Test (D.U.T) whose current was adjusted by selecting the load resistance from the variable resistors appropriately.

Fig. 3.6 shows the typical turn-on and turn-off v_{ce}-i_c characteristics of an IGBT along with the switching time definitions. It also shows the definitions of the power losses during switching. Fig. 3.7 plots the experimental data obtained for the turn-on transient. It can be seen that there is an initial delay $t_{d(on)}$ between the instant when the gate voltage v_{ge} turns on and the instant when the IGBT current i_c starts to increase (or the collector-emitter voltage of IGBT v_{ce} starts to decrease). After this delay i_c increases to its peak value before settling to its steady-state final value. The time interval for i_c to rise



(a) CM50DU-24F

(b) Device structure

Figure 3.4: CM50DU-24F IGBT module from *Powerex*®

from 10% to 90% of its final value I_c is denoted as t_r whereas the time interval between 0.1 I_c to 0.1 V_{ce} is called t_{on}.

Similarly, the experimental measurements for the turn-off transient can be seen in Fig. 3.8. The time delay between the turn-off of v_{ge} and the instant of increase in v_{ce} defines $t_{d(off)}$. After this delay, v_{ce} increases to a peak value and finally settles to a steady-state value. The fall-time t_f of i_c defines from 90% to 10% of its final value, while t_{off} defines the interval between 0.1 V_{ce} and 0.1 I_c. The tailing current time interval is defined from 10% of I_c to 1% of I_c. Fig. 3.6 to Fig. 3.8 also show the current time t_i which is the time interval between the sensed gating signal and the instant when the voltage reaches 10% of its initial value during the turn-on transient and the current reaches 10% of its initial value during the turn-off transient.

The experimental data was collected at a 0.2ns resolution. Fig. 3.7 and Fig. 3.8 show the fitted data which was used for FPGA implementation.The implemented model data was averaged over 12.5ns for the FPGA input clock frequency of 80MHz. Table 3.1 lists the measured IGBT characteristics for the CM50DU-24F against those provided by the manufacturer's datasheet. The experiment was carried out at 300V, 15A because of the limitation of

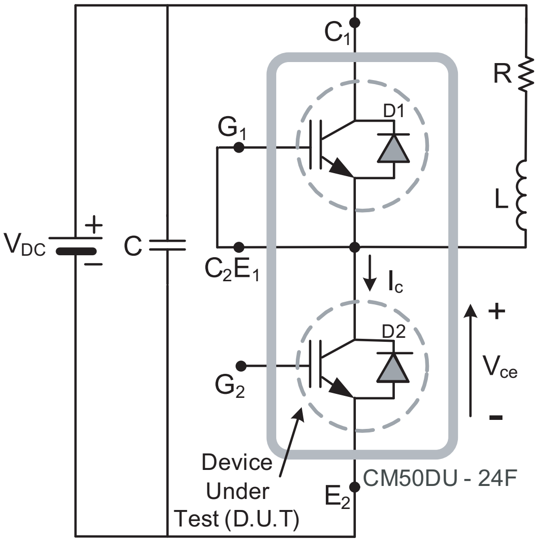


Figure 3.5: Experimental test circuit to measure IGBT characteristics.

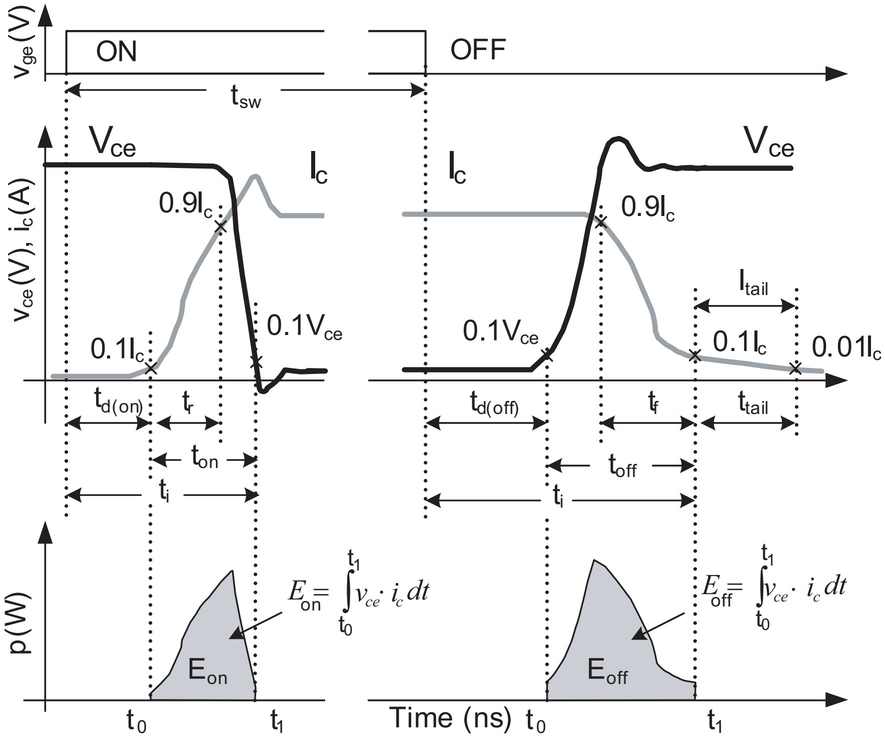


Figure 3.6: Generic IGBT switching characteristics with switching times and loss definitions.

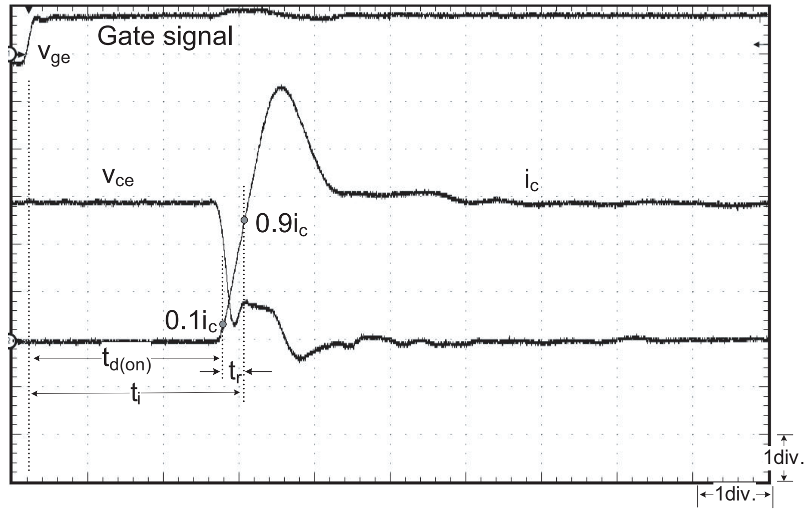


Figure 3.7: The measured result of IGBT V_{ce} and I_c in turn-on transient. (V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.)

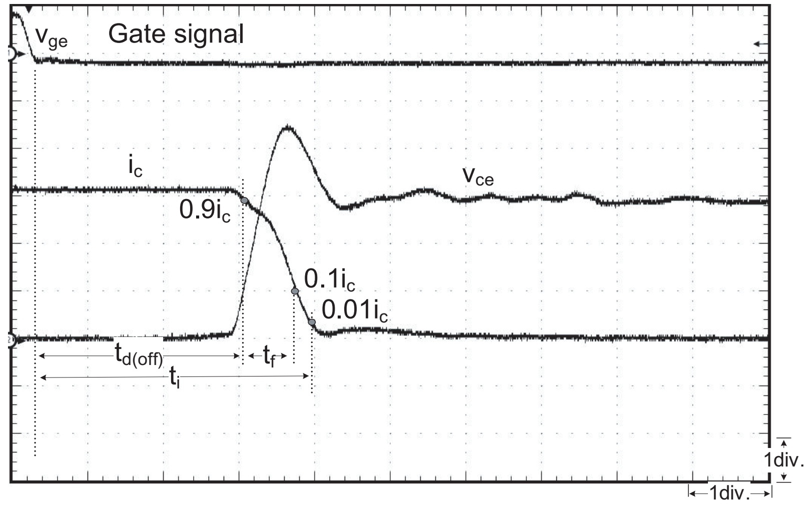


Figure 3.8: The measured result of IGBT V_{ce} and I_c in turn-off transient. (V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.)

the Lab apparatus even though the datasheet test conditions were measured at 600V, 50A.

Table 3.1: Switching data for CM100DU-24F IGBT unit from *Powerex*®.

$t_{d(on)}$	$100ns$
t_r	$50ns$
$t_{d(off)}$	$400ns$
t_f	$300ns$
$V_{CE_{sat}}$	$1.8V$
$V_{d_{sat}}$	$1.25V$

3.3.1 Generalized IGBT modeling guideline

IGBT-based power electronic apparatus can be modeled using two types of off-line [18] simulation tools: (1) system-level, and (2) device-level. In the first category, the issues of interest are the network behavior of the power electronic apparatus and its impact on the power system, such as injected voltage and current harmonics, machine dynamics, and controller performance. All EMTP-type software and general mathematical modeling packages such as MATLAB/SIMULINK belong to this category. These tools utilize nodal or state-space solution of linear ordinary differential equations discretized using numerical integration rules such as the Trapezoidal Rule. Power electronic devices are often modeled in these tools using three types of behavioral [21] models: (1) ideal model, (2) switching function model, and (3) averaged model. All of these models have also been used for off-time simulation [9, 13, 22].

In the *ideal model*, the detailed device characteristics are replaced by an ideal switch where external characteristics related to switching transients, diode reverse recovery, snubber circuits and stray components are omitted. In this modeling category, each IGBT is represented as a two-valued resistor for its *on* and *off* states. A change of switch status, however, results in a change of circuit topology which, in turn, requires the re-calculation of the system matrix. Generally, in real-time simulation, these matrices are pre-computed and stored to minimize simulation time.

In the *switching function* model, a switch converter circuit is replaced by a circuit consisting of only controlled voltage and current sources. The simulation is much faster using the switching-function models with almost the same results as the ideal-model approach gives, i.e., high-frequency effect is included. However, since individual switches no longer exist, it is not possible to monitor the voltage/current or conduction of individual switches.

The *averaged model* can be obtained directly from the switching functions by including the low-frequency components of the switching functions while ignoring all the high-frequency components.

System-level modeling is fairly fast, however, it cannot show the device characteristics accurately and not the real-time simulation. In this thesis, the linear and non-linear model including the previous three models are used to simulate in real-time simulation.

In the second category of device-level modeling, the issues of interest are the switching transients, power losses, and thermal characteristics of the device. SABER and the family of SPICE software are examples of these types of tools, which employ simultaneous nonlinear system solution using numerical methods such as the Newton-Raphson or the Katzenelson method [25]. Device-level modeling is very accurate, however, for motor drives simulation it can be very time consuming. There are three types of device-level models [27] available for IGBT modeling: (1) analytical models, (2) behavioral models, and (3) numerical models. None of these device models have yet been used for real-time simulation mainly due to their computational complexity. The analytical models are based on semiconductor physics describing the carrier dynamics in the device. In this category, the two most popular models are the Hefner model [20] and the Kraus model [28] which have been implemented in SABER as well as SPICE. An electrothermal model based on device characteristics was implemented in transient simulation [29], however, this was still done in the off-line mode. In the behavioral models, IGBT characteristics are fitted using different methods, and the resultant switching functions are then used in a simulator. This approach [30] has been used in an off-line system-level simulation tool (EMTP) to model the device accurately. However, this model still requires very small time-steps to be practically implemented in a real-time simulator based on conventional general purpose processors or DSPs. In [31], the nonlinear Hammerstein-like model was proposed and the model parameters are derived from the semiconductor physics and still very complex to implement in the real-time simulation. A device-level real-time model implemented in a FPGA, for a IGBT-based VSC drive is proposed in [4] which takes into account the precise switching times albeit based on linearized device characteristics.

The *linearized device-level model* is obtained through the linearization of the detailed model. As the characteristics of the IGBT are expressed through a linear function, the reverse recovery, tailing current and other non-linearities at turn-on and turn-off are ignored, however, switching losses, and on-state-losses are considered.

The *detailed nonlinear device-level* model can be constructed from the measured switching characteristics of the IGBT as explained later in this chapter. The measured characteristics are highly non-linear and are sampled at a very small time-step to develop the necessary curves. Using a detailed characteristic-based model, the turn-on and turn-off nonlinear characteristics, switching and conduction losses, and tailing current behavior of the device can be modeled accurately. The power electronic converter model is computed at a fixed time-step, albeit, at an extremely small step-size (of the order of a few nanoseconds). The numerical models utilize finite element methods to model the carrier diffusion in the device resulting in a very accurate although computationally expensive model.

In this thesis, 5 IGBTs models are implemented for the 3-level converter: 3 system level models (ideal model, switching function model, and averaged model) and 2 device level behavioral models (linearized model, and nonlinear model).

3.3.2 3-level VSC model

The 3-level VSC model was developed first based on a single phase leg. As shown in Fig. 3.9 one leg of the 3-level converter consists of 4 IGBTs (A1, A2, A3, A4), with 4 antiparallel diodes (D1, D2, D3, D4), and 2 clamping diodes (D5, D6) to clamp the output terminal voltage v_{out} to the neutral point. The IGBTs can be divided into two pairs A1-A3 and A2-A4. The gating signals for the IGBTs in each pair are complementary with an appropriate dead-time. There are $2^4 = 16$ possible switching combinations for the 4 IGBTs, however, 11 of those combinations are invalid due to the following two operating limits for the 3-level converter:

1. The IGBTs in each pair (A1-A3 and/or A2-A4) cannot be on simultaneously to avoid DC voltage shoot through.
2. A2 and A3 cannot be off simultaneously to avoid a floating output voltage on the phase leg.

Therefore, the following 5 valid states of the 3-level converter are considered:

State 1 : Switches A1 and A2 are closed, while switches A3 and A4 are open.

State 2 : Switches A2 and A3 are closed, while switches A1 and A4 are open.

State 3 : Switches A3 and A4 are closed, while switches A1 and A2 are open.

State 4 : Three switches A1, A2 and A4 are open during *dead-time1* between switches A1 and A3.

State 5 : Three switches A1, A3 and A4 are open during *dead-time2* between switches A2 and A4.

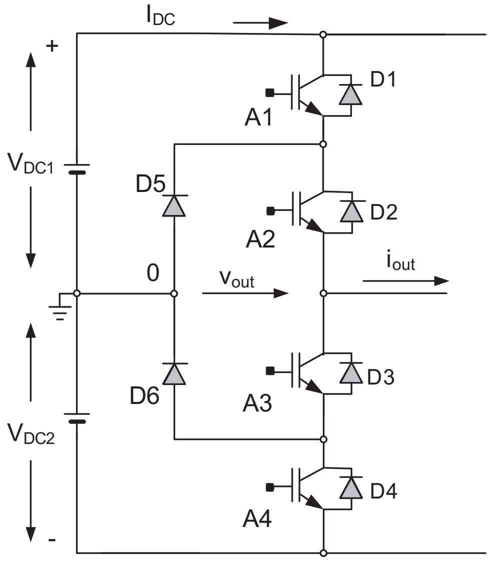


Figure 3.9: One leg of the 3-level VSC.

Ideal model of IGBT

Fig. 3.10 shows the ideal characteristics of IGBT. The detailed device characteristics can be seen in Fig. 3.7 and 3.8. The major difference between the detailed device characteristics and ideal characteristics can be compared. In the ideal model, the on/off transients of IGBT switching and reverse recovery of diode are neglected and consider as a switch for on and off only. The on-stage and off-stage resistances are replaced by 1mΩ and 1MΩ, respectively. It can be seen in Fig. 3.11. Furthermore, the stray components of IGBTs are not considered. The set of non-linear equations for the IGBT characteristics is replaced by the linear equations and it would be great to reduce the simulation time for real-time and off-line simulation. The ideal model is the most commonly used approach method in the system level simulation. There is no extra effort is required from the user point of view to prepare a circuit for simulation and suitable for a wide variety of studies in complex system model.

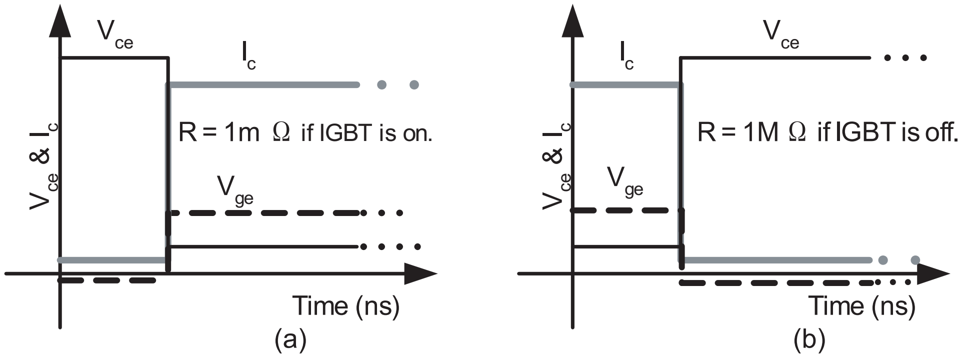


Figure 3.10: Idealized timing characteristics of an IGBT during turn-on and turn-off switching.

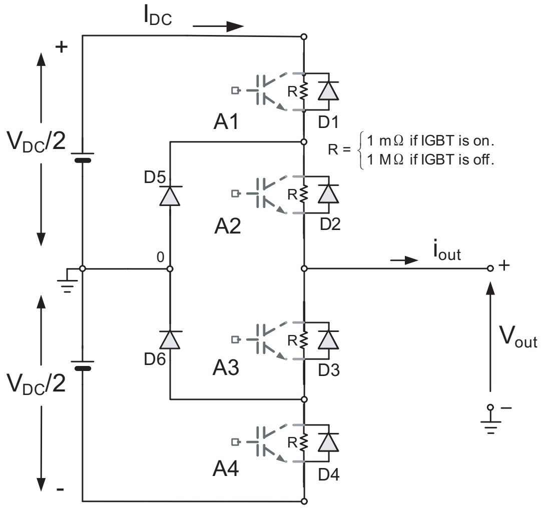
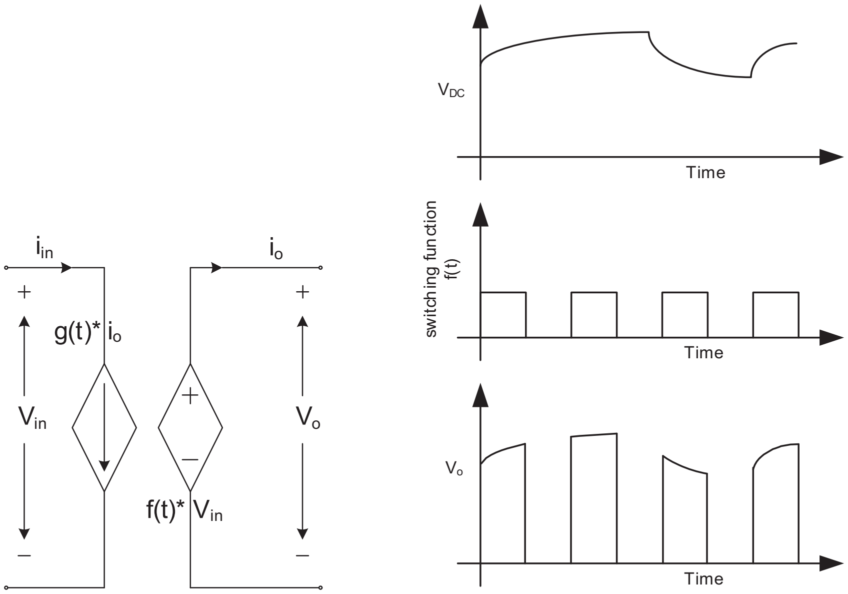


Figure 3.11: One leg of the 3-level converter with ideal models for the IGBTs.

Switching function model of IGBT

In this approach, an IGBT is replaced by a circuit consisting of only the controlled voltage and current sources, that describes the external behavior of IGBT output voltage v_{ce} and current i_c. A switched IGBT converter circuit, consisting of only switch elements, can be seen in Fig. 3.12 (a) and 3.12 (b). The relationship between the input and output quantities of each IGBT can be expressed as

$$v_o(t) = f(t) \cdot v_{in}(t) \tag{3.1}$$



(a) The equivalent circuit model of switching function for IGBT.

(b) Input voltage, switching function and output voltage.

Figure 3.12: Switching function model

$$i_{in}(t) = g(t) \cdot i_o(t) \tag{3.2}$$

In the above equations, *f(t)* and *g(t)* are the switching functions. In many cases, *f(t)* and *g(t)* are related, and one can be derived from the other. From these equations 3.1 and 3.2, the equivalent models of the 3-level converter circuits in Fig. 3.3 can be expressed as in Fig. 3.13. The switching function *f(t)* in this case is the same as the gating signal for IGBT switch. The relationship between the input voltage, switching function and output voltage can be seen in Fig. 3.12 (b)Using the switching-function representation, the converter circuit in Fig. 3.3 can be replaced by the equivalent circuit in Fig. 3.13. Usually, the switching functions can be obtained by directly inspecting the converter circuit. As an example, Fig. 3.14 shows 3-phase voltage source converter and the corresponding switching function representation.

They have three-levels (1, 0, and −1) for switching characteristics that

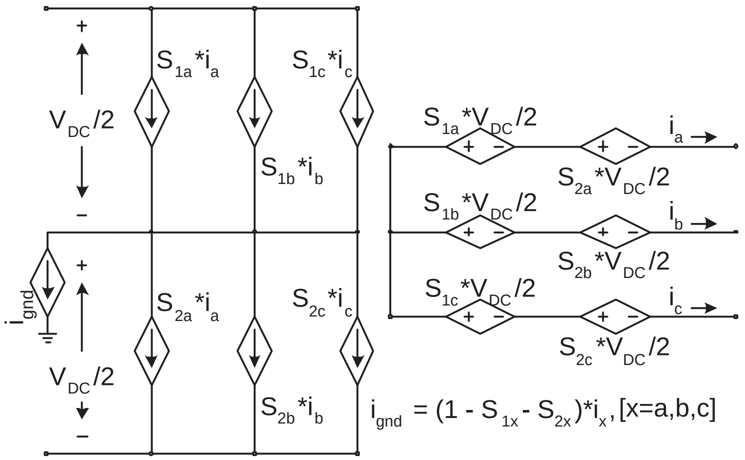


Figure 3.13: Switching function model representation for 3-level converter.

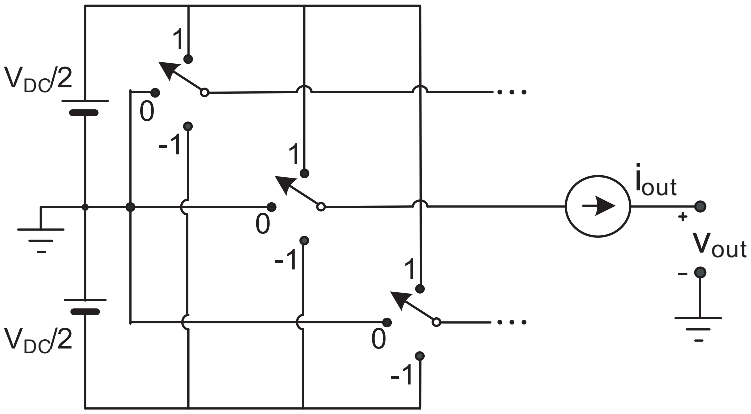


Figure 3.14: Single pole triple throw on/off switch.

depend on the control gate signals. It considers the equivalent circuit model shown in Fig. 3.14 containing three single pole triple throw switches for developing the analysis of the converter operation, and modulation strategy for 3-level converter. Here, each of the three throws for the switches is connected to a dc-bus level: top (V_1), middle (V_0), or bottom (V_{-1}). The switching action of each of the throws can be represented by their respective switching functions (h_{0a}, h_{1a}, etc.) as shown in the Fig. 3.13. On the dc-side, three dc stack currents can be defined as top (I_1), middle (I_0) or bottom (I_{-1}).

$$\begin{bmatrix} V_a(t) \\ V_b(t) \\ V_c(t) \end{bmatrix} = \begin{bmatrix} h_{1a}(t) & h_{0a}(t) & h_{-1a}(t) \\ h_{1b}(t) & h_{0b}(t) & h_{-1b}(t) \\ h_{1c}(t) & h_{0c}(t) & h_{-1c}(t) \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_0(t) \\ V_{-1}(t) \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} I_1(t) \\ I_0(t) \\ I_{-1}(t) \end{bmatrix} = \begin{bmatrix} h_{1a}(t) & h_{1b}(t) & h_{1c}(t) \\ h_{0a}(t) & h_{0b}(t) & h_{0c}(t) \\ h_{-1a}(t) & h_{-1b}(t) & h_{-1c}(t) \end{bmatrix} \begin{bmatrix} I_a(t) \\ I_b(t) \\ I_c(t) \end{bmatrix} \tag{3.4}$$

$$V_s(t) = \frac{1}{3} \begin{pmatrix} V_1 & V_0 & V_{-1} \end{pmatrix} \begin{bmatrix} h_{1a}(t) + h_{1b}(t) + h_{1c}(t) \\ h_{0a}(t) + h_{0b}(t) + h_{0c}(t) \\ h_{-1a}(t) + h_{-1b}(t) + h_{-1c}(t) \end{bmatrix} \tag{3.5}$$

The simulation is much faster using the switching-function model. The IGBTs are considered as switches and neglect the detailed characteristics. Also, the switching-function approach gives the same results as the ideal-model approach, i.e., high-frequency effect is included. However, since individual switches no longer exist, it is not possible to monitor the voltage and current or conduction of individual switches in detail and it is not possible to calculate the power losses.

Averaged model of IGBT

To develop an averaged model of the converter, the switching function is averaged over one switching period, T_s. Thus, the switching behavior and harmonic components of all parameters are not taken into account.

Basically, the averaging operator can be expressed as

$$m_{ij}(\tau) = \frac{1}{T_{PWM}} \int_{\tau}^{\tau+T_{PMW}} h_{ij}(t) \cdot dt \tag{3.6}$$

where $i = 1, 0, -1$ and $j = a, b, c$, τ = averaged period, T_{PMW} = switching period.

In both the ideal-model approach and switching function approach, the circuit still switches at high frequency. To further speed up the simulation, the model derived from the switching-function approach can be further simplified by neglecting the switching effect. That is only the low-frequency components

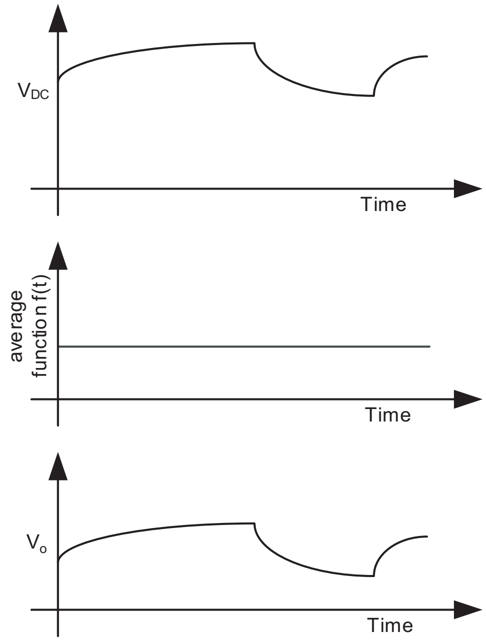


Figure 3.15: Input voltage, average function and output voltage (from top to bottom).

of the switching function are considered, and all other high frequency components due to switching are ignored. As an example, in the averaged model, the switching function $f(t)$ in Fig. 3.12 (b) becomes a dc value (assuming the duty cycle to be 0.5 in this case) and the output voltage v_o has no switching harmonics, as shown in Fig. 3.15.The averaged model can be obtained directly from the switching function. For those circuits in which switching function cannot be easily obtained, the average model can be derived using the state-space averaging technique. Fig. 3.16 shows the averaged model representation for the 3-level VSC. Basically, it is the same as Fig. 3.13 for switching function model except that the switching function S are replaced by their averaged counterparts $\bar{S}$. Since there is no switching and the circuit topology is invariant, the simulation speed can be very fast. Because of this, the average model is particularly convenient in determining the frequency response of a circuit. However, it is not possible to monitor each individual switch and evaluate the

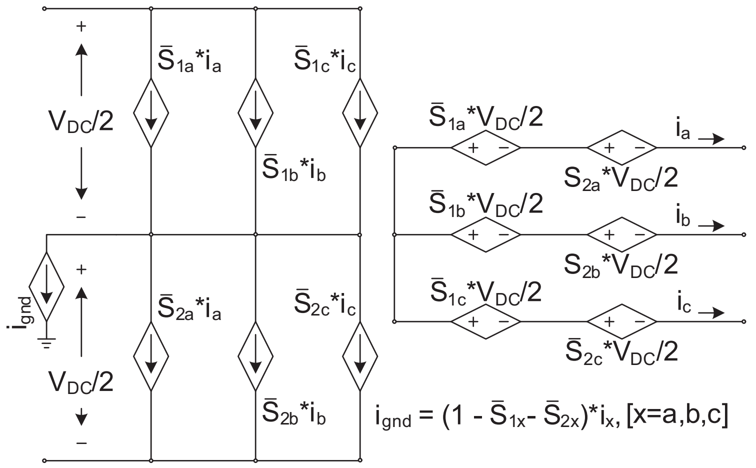


Figure 3.16: Averaged model representation for 3-level converter.

switching harmonic effect.

Linearized device-level behavioral model of IGBT

For the linearized model of IGBTs in drive applications, the IGBTs are used only in switched mode. Furthermore, the largest possible value of the gate-to-emitter voltage V_{ge} is used to minimize the conduction loss in this region. In this case, the forward characteristics of the IGBT can be represented by a linear substitute characteristic Fig. 3.17, which is given as follows:

$$v_{ce,sat}(t) = v_{ce}(t_0) + r_{ce} \cdot i_c(t) \tag{3.7}$$

where $v_{ce}(t_0)$ is the collector-emitter threshold voltage, r_{ce} is the device on-state resistance.

These data can be obtained from the IGBT manufacturer's datasheet. The another important characteristic of an IGBT is the switching transient timing chart for the on/off switching operation of the device. This chart is analyzed with the aids of the standard IGBT test circuit used by the manufacturer. The values of the circuit elements are also standard values defined by the manufacturer. Fig. 3.6 shows the typical timing diagrams of an IGBT during turn-on and turn-off transients respectively. These figures are also used to show the switching time definitions.

The characteristics of the implemented IGBT model are based on the *CM50DU-24F* IGBT from *Powerex* (Table 3.1) with the timing charts shown in Figs. 3.18(a) and (b). As can be observed in these figures, the major characteristics of time-delay, rise time and fall time are taken into account.

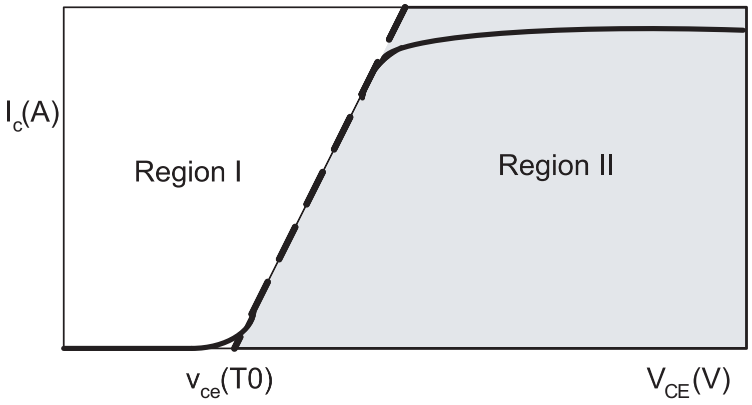
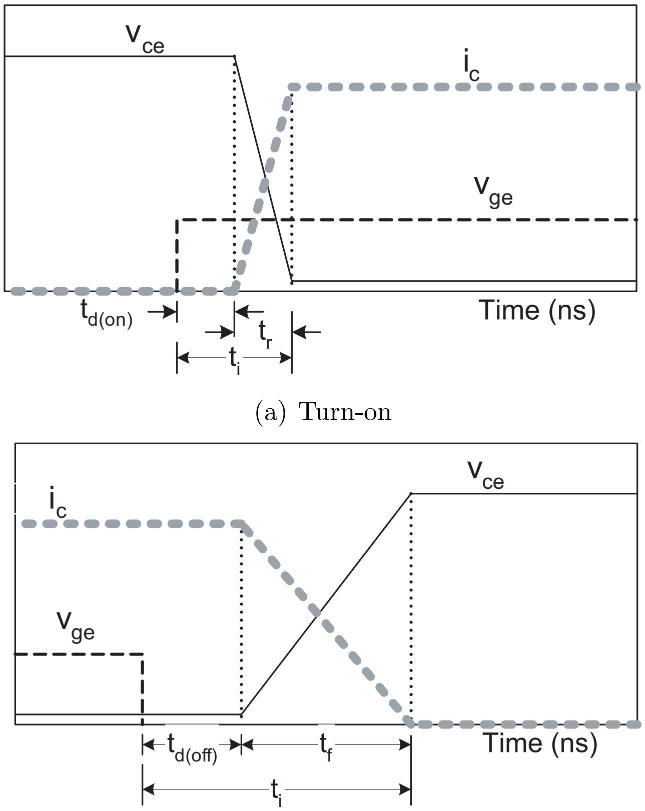


Figure 3.17: Characteristic $V_{CE} - I_C$ curve of a generic IGBT for a specific value of V_{GE}.

Nonlinear device-level behavioral model of IGBT

Using the measured IGBT characteristics (Fig. 3.7, and 3.8) nonlinear device-level model was developed using the per-unit scale and look-up table in the FPGA. From the experimental data obtained it was observed that the rise and fall times for the current and voltage are constant and are proportional to the amplitude of the final value. Therefore, the turn-on and turn-off waveforms can be obtained by scaling a per-unit device-level switching function stored in a look-up table by the final amplitudes. All of the 5 converter states described in Section 3.3.2.

For each of these combinations, at every time-step (12.5ns) of the real-time simulation, the value of the load current i_{out} is used to determine whether the



(a) Turn-on

(b) Turn-off

Figure 3.18: Linearized timing characteristics of IGBT hard-switching turn-on/off waveforms.

current will flow through the IGBTs or the clamping diodes or the antiparallel diodes.

For example, consider that at time $t = t_1$, the gating signals for A1, A2, A3, A4 are changed to 1, 1, 0, 0 respectively. If i_{out} is positive, it will flow through the diodes D3 and D4 for $t < t_1$, and the steady-state output voltage would be $v_{out} = -V_{DC2} - 2V_{d(sat)}$, where $V_{d(sat)}$ is the voltage drop across the diode. After $t = t_1 + t_{d(on)}$, A1 and A2 will start to conduct and i_{out} will flow through the IGBTs to reach its steady-state value of I_c. The output voltage will also rise until it reaches its steady-state value of $v_{out} = V_{DC1} - 2V_{ce(sat)}$.

During dead-time1 the gating signals for A1, A2, A3, A4 will change to

0, 1, 0, 0 respectively, and v_{out} would remain the same as before. After the time $t_{d(off)}$ for A1, v_{ce} will increase and collector current i_c decreases until it reaches 90% of I_c. Thereafter, i_c falls quickly to reach 10% of I_c which causes the overshoot in v_{ce}. Within the tailing region, i_c falls slowly becoming almost zero, and then gets replaced by the current of clamping diode D5 flowing through A2. For the gating pattern 0, 1, 1, 0, the positive current will continue to follow through D5 and A2, and $v_{out} = V_{d(sat)} + V_{ce(sat)}$. During dead-time2 for the gating pattern 0, 0, 1, 0, the current will flow through the antiparallel diodes D3 and D4, and $v_{out} = -V_{DC2} - 2V_{d(sat)}$. When the gating pattern changes to 0, 0, 1, 1, the positive current will continue to flow through D3 and D4, when the steady-state output voltage will be $v_{out} = -V_{DC2} - 2V_{d(sat)}$.

On the other hand, if i_{out} is negative, and assuming that the gating signals for A1, A2, A3 and A4 were equal to 1, 1, 0, 0 at time $t = t_1$, the current would flow through the diodes D1 and D2 for $t < t_1$, and at $t = t_1$, it would continue flowing through the same diodes resulting in $v_{out} = V_{DC1} + 2V_{d(sat)}$. During dead-time1 with the gating pattern 0, 1, 0, 0, v_{out} will be same as before. For 0, 1, 1, 0, the negative i_{out} will flow through A3 and D6 resulting in $v_{out} = V_{ce(sat)} + V_{d(sat)}$. During dead-time2, v_{out} will remain the same as before. Finally, when the gating pattern 0, 0, 1, 1 occurs, i_{out} will go through A3 and A4 with $v_{out} = -V_{DC2} + 2V_{ce(sat)}$.

When i_{out} is equal to zero, the steady-state value for the output voltage will be V_{DC1}, 0, or $-V_{DC2}$ depending on the gating pattern.

3.4 Induction machine

The squirrel case induction motor is basically a sample, cheap and reliable machine which can provide excellent characteristics where a constant shaft speed is required. For example, the speed of a typical motor fed from a constant frequency supply may vary less than 5% over the range of normal load and supply voltage variations. This basic characteristic of the motor, to run at a virtually constant speed close to synchronous, has for long been a challenge to designers who have sought to devise variable speed schemes.

A rotating magnetic field is set up by the voltage applied to the stator windings and the speed of rotating is uniquely determined by the number of poles and supply frequency such that

$$N_s = \frac{120 \times f}{P}; \tag{3.8}$$

where N_s is the synchronous speed in *rpm*, f is the fundamental frequency of system, P is the number of poles of synchronous machine.

The rotating field induces the currents in the squirrel cage rotor which interact with that field so as to produce a torque. This torque accelerates the rotor to revolve at some speed, N_r, less than the synchronous speed, N_s. The rotor requires a torque to satisfy the load torque and this same torque is experienced by the rotating (air gap) field. So, there are two rotating members that experience the same torque but have different speed.

It has been shown by Blaschke [41] that these problems may be overcome by field orientation, essentially reducing the control dynamics of an ac machine to those of a separately excited compensated dc motor. This is achieved by defining a time-varying vector which corresponds to a sinusoidal flux wave moving in the air gap of the machine. When referring the *MMF* wave of the stator currents described by i_s. The model of induction machine in terms of space vector is clear physically and simple mathematically [47]. The expressions of flux linkage, voltage, current, power and torque of the machine derived from the model can completely substitute for the expressions in *abc* reference frame. The theory of induction machine vector control is based on constant parameter induction machine *dq* axis model that neglects all the parameter variations, flux saturation and iron core loss. Due to inevitable variation of parameters during operation of the drive, actual performance may significantly deteriorate from ideal.

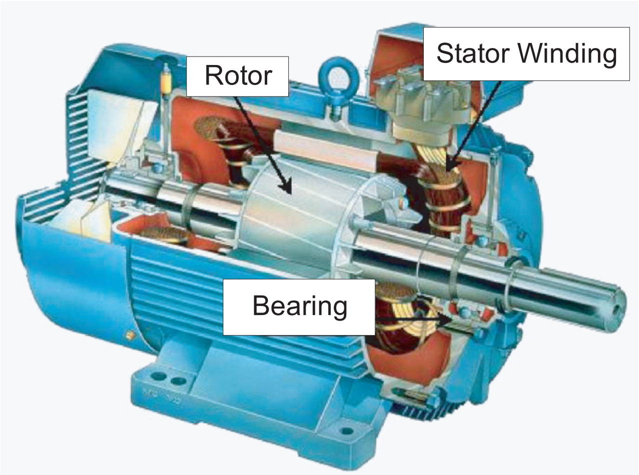


Figure 3.19: Schematic view of a squirrel cage induction motor.

In general, two approaches are identified as being available to combat the problem of parameter variations. The first approach relies on modern control techniques and attempts to increase the robustness of the drive while still relying on the same basic constant parameter *dq* axis model of the machine. The second approach attempts to improve the accuracy of orientation and thus performance of the drive by using modified *dq* axis models of induction machines. The modified models account for one or more sources of parameter variations and thus essentially provide an open-loop adaptation for the specific parameter(s).

In three-phase induction motors, the connection to the mains does not usually use the neutral. Therefore, the mains current has no homopolar component. A two-dimensional representation can then be used for describing three-phase induction motor phenomena, a suitable one being based on the current Park's vector model.

The *dq* model is well suited to drive analysis since it is readily linked to power converter, switching level (modulation) control, supervisory control, and mechanical system models. Simulations using this class of model execute very rapidly in comparison with time domain Finite Element Analysis models [48]-

[49]. However, the model has many shortcomings in terms of accuracy, particularly in the case of switching frequency effects such as current and torque ripple. Numerous efforts to address these deficiencies have appeared in the literature. For example, one approach has focused on magnetically coupled stator and rotor circuit models, whereby the winding configuration is explicitly taken into account. By appropriate modification, a bar-by-bar rotor model can be reduced to the standard model, assuming a sinusoidally distributed stator winding. A major research area has been in incorporating main-axis magnetic saturation into the model. Another area of improvement has been the incorporation of the deep-bar effect specific to squirrel cage rotors, which in effect recognizes the large rotor bars as distributed-parameter systems. Saturation of leakage inductances is frequently ignored but can have significant impact on performance.

3.4.1 State-space induction machine model

A 4-pole, 50HP, 460V squirrel cage induction motor, whose parameters are given in Table 3.2, was used in the real-time simulation. The machine representation is based on the fifth-order stationary reference frame model [10], and is described by the following state-space equations on the electrical side:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.9}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{3.10}$$

where $\mathbf{x} \in \mathbb{R}^4$ in the state vector composed of stator and rotor flux linkages; $\mathbf{u} \in \mathbb{R}^2$ is the input vector composed of the stator voltage components; and $\mathbf{y} \in \mathbb{R}^4$ is the output vector composed of stator and rotor current components, defined as:

$$\mathbf{x} = \left[\begin{array}{cccc} \lambda_{\alpha s}(t) & \lambda_{\beta s}(t) & \lambda_{\alpha r}(t) & \lambda_{\beta r}(t) \end{array} \right]^T \tag{3.11}$$

$$\mathbf{u} = \left[\begin{array}{cc} V_{\alpha s}(t) & V_{\beta s}(t) \end{array} \right]^T \tag{3.12}$$

Table 3.2: Induction Machine Parameters

R_s	0.087Ω
R_r	0.228Ω
L_m	$34.7mH$
L_s	$35.5mH$
L_r	$35.5mH$
J	$1.662Kg \cdot m^2$

and

$$\mathbf{y} = \left[\begin{array}{cccc} i_{\alpha s}(t) & i_{\beta s}(t) & i_{\alpha r}(t) & i_{\beta r}(t) \end{array} \right]^T \tag{3.13}$$

The coefficient matrices **A**, **B** and **C** are given as:

$$\mathbf{A} = \left[\begin{array}{cccc} -c_3 r_s & 0 & c_1 r_s & 0 \\ 0 & -c_3 r_s & 0 & c_1 r_s \\ c_1 r_r & 0 & -c_2 r_r & -\omega_r \\ 0 & c_1 r_r & \omega_r & -c_2 r_r \end{array} \right] \tag{3.14}$$

$$\mathbf{B} = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{array} \right], \mathbf{C} = \left[\begin{array}{cccc} c_3 & 0 & -c_1 & 0 \\ 0 & c_3 & 0 & -c_1 \\ -c_1 & 0 & c_2 & 0 \\ 0 & -c_1 & 0 & c_2 \end{array} \right] \tag{3.15}$$

with

$$\begin{aligned} c_1 &= \frac{L_m}{L_s L_r - L_m^2} \\ c_2 &= \frac{L_s}{L_s L_r - L_m^2} \\ c_3 &= \frac{L_r}{L_s L_r - L_m^2} \end{aligned}$$

L_m is the mutual inductance; L_s and L_r are the stator and rotor self-inductances; R_s and R_r are the stator and rotor resistances, respectively. The mechanical dynamics of the machine are represented by the following equations using rotor electrical speed ω_r as the state variable:

$$\dot{\omega_r}(t) = \frac{P}{2J} \left[T_e(t) - T_L(t) \right] \tag{3.16}$$

and

$$T_e(t) = \frac{3}{2}\frac{P}{2}\left[i_{\beta s}(t)i_{\alpha r}(t) - i_{\alpha s}(t)i_{\beta r}(t)\right] \tag{3.17}$$

where P is the number of poles and J is the total rotor inertia. T_e and T_L are the electrical and load torque, respectively.

3.5 Carrier-based pulse width modulation

In general a pulse width modulated switching signal is generated by comparing a constant or time varying modulating signal with a triangle signal. For the 3-level VSC in this thesis, we use two triangular signals and the carrier control modulation signals with injected 3rd harmonics as shown in Fig. 3.20.

The modulating signal $v_{con} = V_{con}sin(\omega t - \delta) + V_{con3}sin(3\omega t - \theta)$ is compared with triangular carrier signal v_{tri}, and the intersection points are used to set the switching function S high or low. When $v_{con} > v_{tri}$, $S = 1$ and when $v_{con} < v_{tri}$, $S = 0$.

In a 3-phases converter, 3 modulating control signals that are 120^0 out of phase are compared with common triangular carrier signals of fixed amplitude and frequency to generate the three switching function $S_k(k = a, b, c)$. The frequency of the triangular signal determines the converter switching frequency f_{sw} and the frequency of the modulating signal determines the fundamental frequency f_1 of the converter output. The frequency modulation ratio is expressed as $m_f = \frac{f_{sw}}{f_1}$ while the amplitude modulation index is defined as $m_a = \frac{V_{mod}}{V_{tri}}$.

With the DC bus voltage kept constant at V_{dc}, the sinusoidal PWM scheme results in switched pulses of V_{dc} across the converter AC terminals. Fourier analysis shows that the resulting output voltage waveform contains a predominant fundamental component and high frequency harmonics at the carrier and side band frequencies. The peak value of the fundamental frequency harmonics at the carrier and side band frequencies. The peak value the fundamental frequency component of voltage v_{aN} in the linear region ($m_a < 1$) is given.

Thus by varying m_a, the amplitude of the fundamental Fourier series component of the switched voltage at the AC terminal is also varied proportionally. The phase of the fundamental component of the switched voltage can be changed by varying the phase angle δ of the modulating sine wave v_{con} with respect to the reference AC sinusoidal source voltage. The harmonic spectrum of the converter output is determined by m_f.

The magnitude of the sinusoidal reference can be extended up to the magnitude of the carrier waveform. This situation corresponds to a maximum modulation index:

$$m_{max} = \frac{u_{(1)}}{u_{six-step}} = \frac{\pi}{4} = 0.785 \tag{3.18}$$

Carrier-based PWM algorithms have emerged in analog hardware support. The performance at low-modulation index is very poor because of the narrow pulse limitations and transition instabilities during the change of the modulation function.

The ratio (m_f) between the frequency of the carrier and the frequency of the control (reference) signal can be the different values. The modulating signal with a third harmonic injection with the phase selected to suppress the peak of the sinusoidal reference.

$$v_{ref,A} = V_{con} sin(\omega t - \delta) + V_{con3} sin(3\omega t - \theta) \tag{3.19}$$

$$v_{ref,B} = V_{con} sin(\omega t + 120^0 - \delta) + V_{con3} sin(3\omega t - \theta) \tag{3.20}$$

$$v_{ref,C} = V_{con} sin(\omega t + 240^0 - \delta) + V_{con3} sin(3\omega t - \theta) \tag{3.21}$$

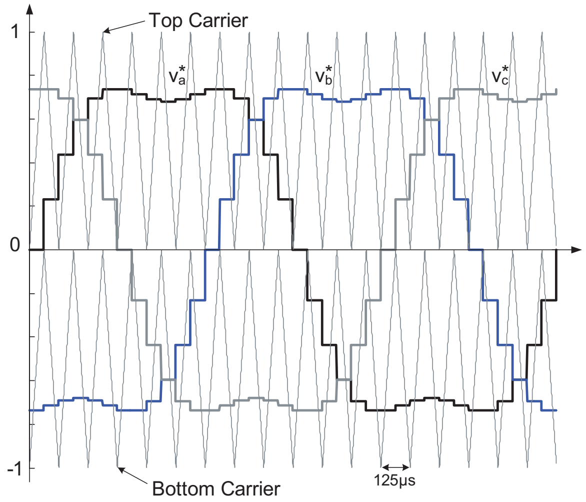


Figure 3.20: 3-level asymmetrical regular sampled PWM.

3.6 Closed-loop control system

Since its introduction by Blaschke [41], it is well known that field-oriented control allows high performance speed and torque response of induction machines. When the control is oriented by the rotor flux, the induction machine behaves like a DC machine with separate excitation.

The field orientation and reference frame transformation are the elements of vector control. It can be realized by coinciding the axes of the rotor flux reference frame with the synchronously rotating reference frame. In other words, the spatial angular frequency of the rotor flux linkage must be adjusted synchronously with the primary angular frequency, in addition, the magnitude of the rotor flux linkage must be controlled by the stator current. However, in order to implement the field-oriented control method, the induction machine is recognized as a three-input, three-output controlled system. The 3-input variables are the primary angular frequency, the magnetizing current component, and the torque current component. Furthermore, the 3-output variables are the rotor angular velocity, and the rotor flux linkage currents i_α and i_β [5]. The principle of electromechanical energy conversion is not only the basic principle of electromechanical energy converter but also an important tool for rotating electrical machine analysis.

Considering a dq machine representation, where the d-axis is aligned with the rotor flux space vector rotating at $\omega_e\ rad/s$ and the q-axis is 90^0 apart as can be seen in Fig. 3.21, the following equations [54] can be obtained:

$$\frac{\lambda_r}{\tau_r} + \frac{d\lambda_r}{dt} = \frac{L_m}{\tau_r} i_{ds} \tag{3.22}$$

$$(\omega_e - \omega_r) = \frac{L_m}{\tau_r} i_{ds} \tag{3.23}$$

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\lambda_r i_{qs} \tag{3.24}$$

As can be observed in (3.22)-(3.24), the d-axis current controls the flux and the q-axis current, considering the flux constant, controls the torque and the machine speed. In these equations, τ_r is the rotor time constant, defined as $\tau_r = \frac{L_r}{R_r}$.

When the induction machine is fed by a voltage source converter instead of a current source-converter, the following equations can also be obtained:

$$V_{ds} = \left(R_s + \frac{L_m^2}{\tau_r L_r}\right) i_{ds} + \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s \frac{di_{ds}}{dt} + V_{dx} \tag{3.25}$$

$$V_{dx} = -\omega_e \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s i_{qs} - \frac{1}{\tau_r}\frac{L_m}{L_r}\lambda_r \tag{3.26}$$

$$V_{qs} = \left(R_s + \frac{L_m^2}{\tau_r L_r}\right) i_{qs} + \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s \frac{di_{qs}}{dt} + V_{qx} \tag{3.27}$$

$$V_{qx} = \omega_e \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s i_{ds} + \omega_r \frac{L_m}{L_r}\lambda_r \tag{3.28}$$

(3.25)-(3.28) reveal that there is an axis coupling represented by the terms V_{dx} and V_{qx}. This coupling was discarded by the control scheme, being treated

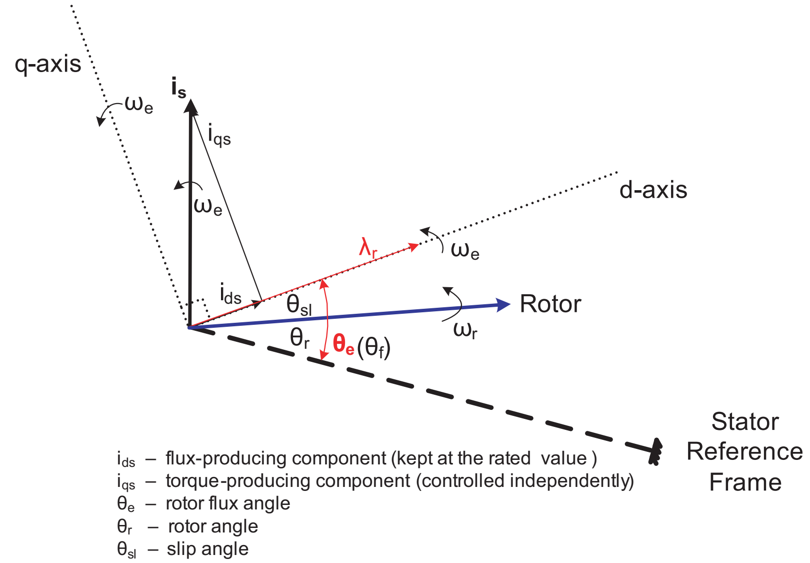


Figure 3.21: Direct field-oriented controller based on rotor flux.

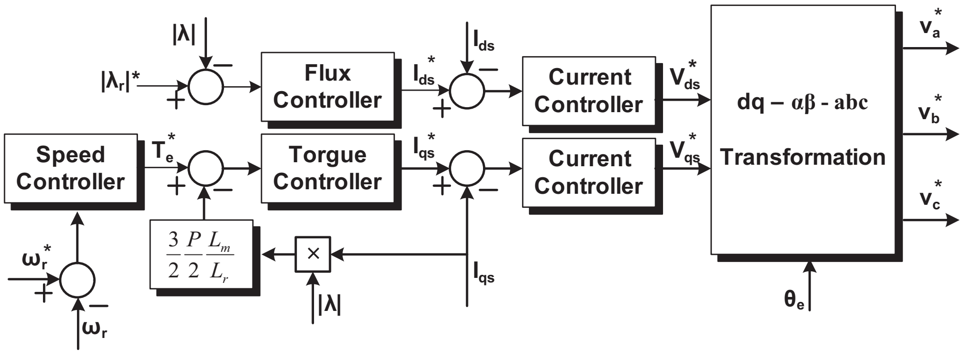


Figure 3.22: Schematic of direct field-oriented controller for AC-motor drive.

as an exogenous input due to the high gain current controller. Fig. 3.22 shows the controller block diagram.

The controller requires the measurement of rotor speed (ω_r), stator currents (i_{ds} and i_{qs}), the modulus of the rotor flux ($|\lambda_r|$) and rotor flux position ($\angle\lambda_r$). $|\lambda_r|^*$ and ω_r^* are the references for the rotor flux and electrical speed, respectively. The information about the rotor flux (magnitude and angle) is usually obtained with the aid of a flux observer [43], [34].

Field-oriented control allows for decoupled control of the rotor flux and torque of the machine by using a dq machine representation where the d-axis is aligned with the rotor flux space vector rotating at ω_e rad/s and the q-axis is 90^0 apart. The inputs to the controller are the rotor flux reference and the speed reference. The measurements are rotor speed, stator currents and the rotor flux space vector. The torque reference is derived from the speed controller, while the current references (I^*_{ds} and I^*_{qs}) come out of the flux and torque controllers. Two independent current controllers provide the respective V^*_{ds} and V^*_{qs} control commands in the field oriented rotor reference frame. The desired three phases control signals for PWM, v^*_a, v^*_b and v^*_c are generated from V^*_{ds} and V^*_{qs} by transformation to the stationary reference frame.

3.7 Summary

This chapter presented the models of individual components of a variable speed AC drive: the 3-level converter, the induction machine, the PWM, and the closed-loop controller. To model the 3-level converter, 5 different modeling methods of the IGBT are proposed. The complete FPGA implementation of the models described in the chapter will be given in Chapter 4.

4

FPGA-based Real-Time Implementation of a Variable Speed Drive

4.1 Introduction

This chapter presents the implementation of a FPGA-based real-time digital simulator for the 3-level induction machine AC drive. The objectives of this implementation is to demonstrate the practical feasibility of the component models presented in the previous Chapter. The design of the simulation is based on the Stratix EP1S80 device and Stratix III EP3SL150F1152C2 device from ALTERA. The system implementation has been properly tailored to meet the constraints of the available hardware resources and to meet the simulation objectives.

4.2 Hardware platform

The overall schematic of the hardware involved in this work is illustrated in Fig. 4.1. It consists of the stratix and stratix III FPGA boards, the host PC, the DAC boards, the oscilloscope, and the interfacing cables. The following gives a brief description of each of these components.

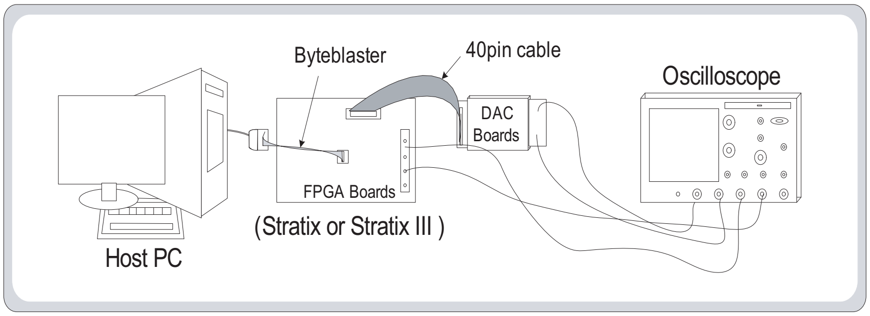


Figure 4.1: Overview of hardware setup.

4.2.1 Stratix DSP development board

Fig. 4.2 shows the stratix DSP development board. The board is based on the powerful development platform for the *Digital Signals Processing* (DSP) designs, and features with the Stratix EP1S80 device in the speed grade (−6) 956-pin package.

Table 4.1 shows the details of the stratix EP1S80B956 device.

In the Stratix board, there are several components:

1. Analog input/output

   -There are two 12-bits 125-MHz A/D converters.

   -There are two 14-bits 165-MHz D/A converters for 80-MHz clock.

   -Signal-ended or differential inputs, and single-ended outputs.

2. Memory subsystem

-2 Mbytes of 7.5-ns synchronous SRAM configured as two independent 36-bit buses.

-64 Mbits of flash memory.

3. Configuration options

   -On-board configuration via the 64-Mbits flash memory, plus the Altera EPM7064 programmable logic device (PLD).

   -Download configuration data using ByteBlasterMV download Cables.

4. Dual seven-segment display.

5. One 8-pin dual in-line package (DIP) switch.

6. Three user-definable push-button switches.

7. One 9-pin RS-232 connector.

8. Two user definable LEDs.

9. On-board 80MHz oscillator.

10. Single 5-V DC power supply.

By using the *ByteBlaster II* cable, SOF file can be downloaded to the board for real-time simulation. One 8-pin dual-in-line package (DIP) switch and three user-definable push button switches are used as the interface of human and machine. For the system reset, one of three user-definable push button switches (SW0) is used. The others are set to reverse and increase/decrease the speed of the motor. The DIP switches are used for two D/A outputs management, and the load T_L applied to the induction motor.

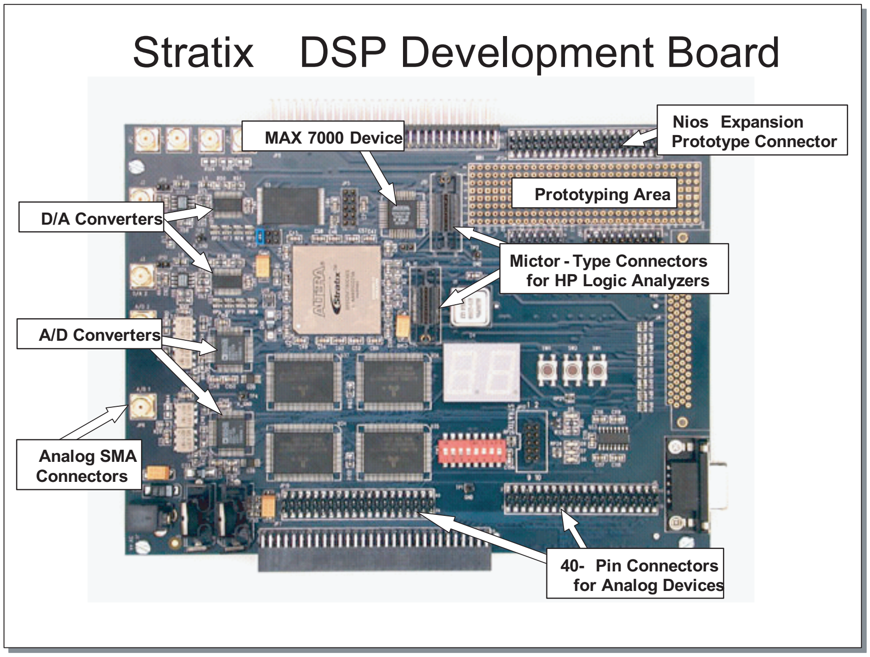

Stratix DSP Development Board
MAX 7000 Device
Nios Expansion Prototype Connector
Prototyping Area
D/A Converters
Mictor - Type Connectors for HP Logic Analyzers
A/D Converters
Analog SMA Connectors
40- Pin Connectors for Analog Devices

Figure 4.2: EP1S80 FPGA Stratix board.

Table 4.1: Stratix Device Features.

Feature	**EP1S80B956-6**
Logic elements (LEs)	$79,040$
M512 RAM Blocks (32x18bits)	767
M4K RAM Blocks (128x36bits)	364
M-RAM Blocks	9
Total RAM bits	$7,427,520$
DSP Blocks	22
Embedded multipliers (based on 9x9)	176
PLLs	12
Max user I/O pins	679
Package type	956-pin BGA
Board reference	U1
Voltage	1.5V internal, 3.3V I/O

4.2.2 Stratix III DSP development board (DE3)

Fig. 4.3 shows the DE3 board based on the Stratix III EP3SL150F1152C2 device. Table 4.2 shows the details of the Stratix III EP3SL150F1152C2 device. It depicts the layout of the board and indicate the location of the connectors and key components.

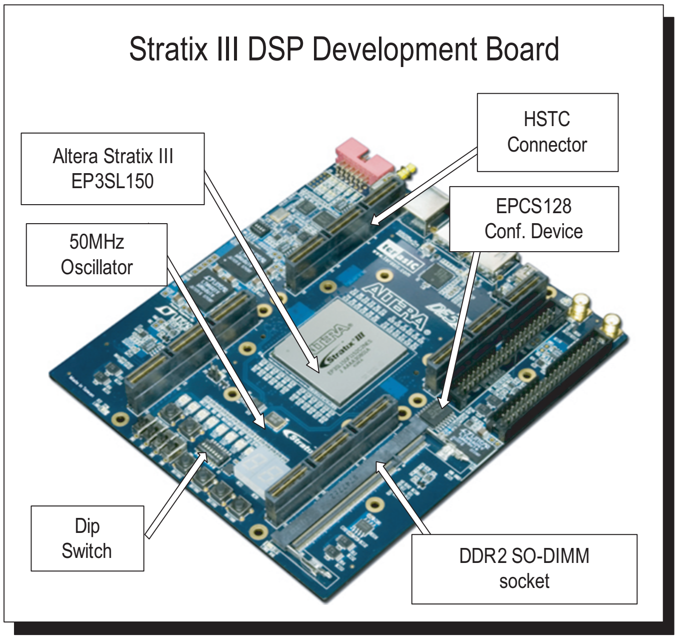


Figure 4.3: EP3SL150F1152C2 FPGA Stratix III DE3 board.

Table 4.2: Stratix III Device Features.

Feature	**EP3SL150F1152C2**
Logic elements (LEs)	142000
Memory (Kbits)	5499
DDR2 SDRAM DIMM (GByte)	1
QDRII/+SRAM (Mbit)	72
DDR2 SDRAM devices (MByte)	16
SSRAM (MByte)	4
Embedded multipliers (based on 9x9)	384
PLLs	8
Max user I/O pins	736
Package type	1152-pin BGA
Board reference	U1
Voltage	1.1V internal, 1.8V I/O

The DE3 board has plenty of features that allow users to implement a wide range of designed circuits. The Stratix III device is capable of dealing with resource consuming projects and complex algorithm verification; the HSTC

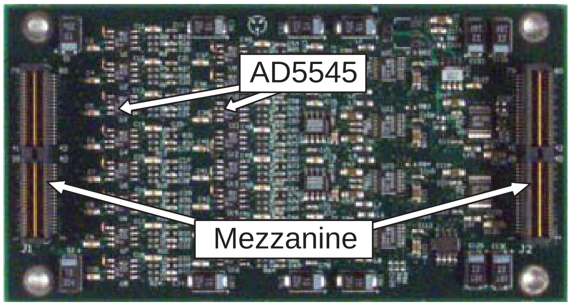


Figure 4.4: OP5330 of Opal-RT DAC card.

interface is equipped for high speed interconnection and configurable I/O standards.

For the system reset, one of four user definable push button switches (SW0) is used. The others are set to reverse and increase/decrease the speed of the motor. The DIP switches are used for two D/A outputs management, and the load T_L applied to the induction motor.

4.2.3 Digital-to-Analog Converters

There are two types of DAC boards available in RTX-LAB: OP5330 board from Opal-RT for the Stratix board, and THDB ADA card from Terasic for the Stratix III boards.

OP5330 board from Opal-RT

The OP5330 (Fig. 4.4) is one of a range of function modules for the OP5000 Signal Conditioning and I/O products from Opal-RT Technologies. It allows up to 16 analog signals to be output simultaneous from a single module.

It features 16 channels with individual 16-bit Digital-to-Analogy Converter (DAC) that ensures simultaneous signal generation from multiple channels. Each DAC can update up to 1MS/s, giving a total throughput of 16MS/s. On-board signal conditioning provides voltage ranges of $+/-16$V.

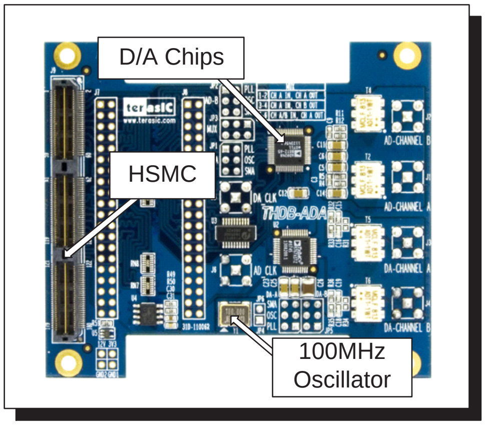


Figure 4.5: THDB ADA (ADA) daughter board from Terasic.

Terasic High Speed A/D and D/A Development board

Terasic High Speed A/D and D/A Development board is shown in Fig. 4.5. The feature set of the high speed A/D and D/A Development board (ADA) is listed below:

- Dual A/D channel with 14-bit resolution and data rate up to 65 MSPS.
- Dual D/A channel with 14-bit resolution and data rate up to 125 MSPS.
- Dual interfaces include HSMC and GPIO, which are fully compatible with Cyclon III start Kit and DE1/DE2/DE3, respectively.
- Clock sources include Oscillator 100 MHz, SMA for A/D and D/A each, and PLL from either HSMC or GPIO interface.

4.2.4 Tektronix Digital Phosphor Oscilloscope DPO7054

Fig. 4.6 shows the DPO7054 oscilloscope. The main features of this scope include:

- 500 MHz bandwidth.
- Up to 20 GS/s Real-time sample Rate on one channel and up to 5 GS/s on All 4 channel.

- Up to 400 Megasamples Record Length with MultiView Zoom feature for Quick Navigation.
- $>$ 250000wfms/s Max Waveform Capture Rate.
- User-selectable Bandwidth limit filter for Better Low-frequency Measurement Accuracy.
- Event Search and Mark to find specific events in the entire waveform.
- 12.1 inches XGA Display with ToughScreen.
- OpenChoice Software with Microsoft Windows XP OS enables Built-in Networking and Extended Analysis.

Figure 4.6: Digital Phosphor oscilloscope DPO7054.

4.3 FPGA implementation of system models

The hardware realization of the system model has been the straightforward implementation on FPGA chip. It is divided into four parts that include the 3-level VSC, the induction motor, the direct field oriented controller, and STPWM.

4.3.1 3-level VSC hardware realization

This section describes the hardware implementation of the system-level IGBT models and the device-level IGBT models for three level VSC. The implementation was done entirely in VHDL.

System-level model hardware realization

The overall hardware realization (Fig. 4.7) of the VSC using the 3 system-level IGBT models (ideal, switching function and averaged) is basically the same except for the IGBT model itself as described in Sec. 3.3.2. The DTC1 and DTC2 are counters to include dead-time for A1 and A2 switch. The 3 level VSC model was then constructed by duplicating this design for the single leg.

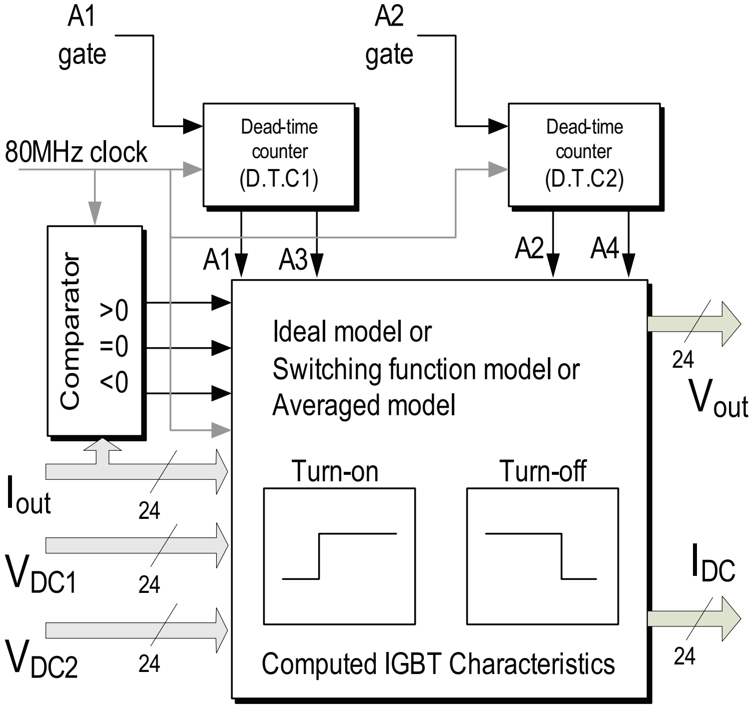


Figure 4.7: Functional block diagram of the hardware realization of IGBT system level models for one VSC leg.

Linearized device-level model hardware realization

The FPGA realization of the single leg of the 3-level VSC using a linearized device-level IGBT model is shown in Fig. 4.8. Instead of receiving the four gating signals, the model receives only the two gate signal for the switches A1 and A2. These gating signals go through a block that adds the dead-time and then generates the four gate signals for A1, A2, A3 and A4. This approach was chosen to simulate the presence of the IGBT gate drivers, which add the dead-time after the PWM generation. With the gating signals, V_{DC1}, V_{DC2} and I_{out} as inputs, the model then implements a state machine with eight states, one for each possible combination of the switches. In each state, the output current value of the leg is verified in order to determine the path of current flow. The state machine interacts with the counters responsible for modeling the time turn-on delays $t_{d(on)}$, turn-off delay $t_{d(off)}$, rise time t_r and fall time t_f, generating control signals that drive these counters. The counter outputs are used to determine the output voltage with respect to the ground terminal of dc bus. The model also outputs the current I_{DC} through A1, A2 or D1, D2 which could be used to determine the output current of the dc link.

The single leg model was duplicated to make the three-phase 3-level VSC model. The inputs to the VSC model are the three phase currents, two DC voltages, and six gating signals with their complements. The outputs of the model are the three-phase voltages V_a, V_b and V_c, and the current I_{DC}. All the model variables are chosen as signed fixed point numbers except for the gate signals which are single bits. The number format for all model voltages and currents in the FPGA implementation was 19.5, i.e., 19 bits are used to represent the integer part, and 5 bits are used to represent the fractional part.

Non-linearized device-level model hardware realization

The FPGA realization of one-leg of the 3-level VSC using a nonlinear device-level IGBT model is shown in Fig. 4.9. For the non-linear model, the measured

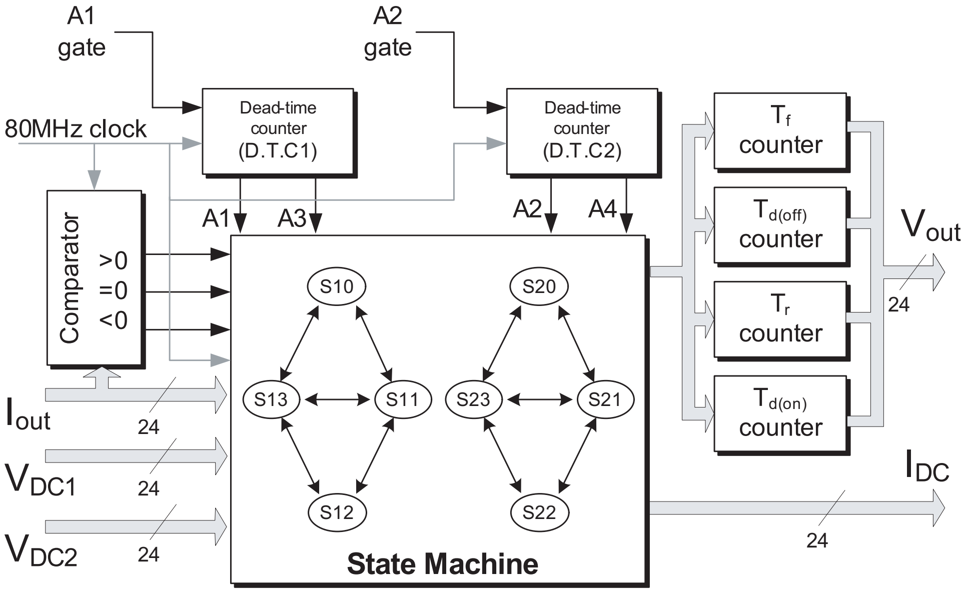


Figure 4.8: Functional block diagram of the hardware realization of one leg of 3-level VSC using linearized device-level IGBT model.

per-unit IGBT switching characteristics were implemented at a resolution of 12.5ns in a Look-Up Table (LUT) of the ROM on the FPGA board and then the behavior model was generated. *Altsyncrams* from the mega-function library of ALTERA are used with one read port as ROM mode. To get the non-linear IGBT characteristics, ROM is used to implement a combinational logic circuit which converts a 7-bit number to the corresponding non-linear characteristic signal. At each time-step, the appropriate IGBT characteristic is read from the LUT based on the gating signals. Depending on the address and clock frequency, the accurate per-unit IGBT characteristic can be read and converted to the outputs of $v_{ce}(t)$ and $i_c(t)$, and then changed to the instantaneous output voltage v_{out} and current I_{DC} exactly depending on the inputs I_{out}. Therefore, the output result values are the instantaneous current and voltage (within the nanoseconds range) that are shown accurately.

For the model of the IGBT, there are 4 LUTs to implement the IGBT characteristics of $V_{ce(on)}$, $V_{ce(off)}$, $I_{c(on)}$ and $I_{c(off)}$. From those characteristics

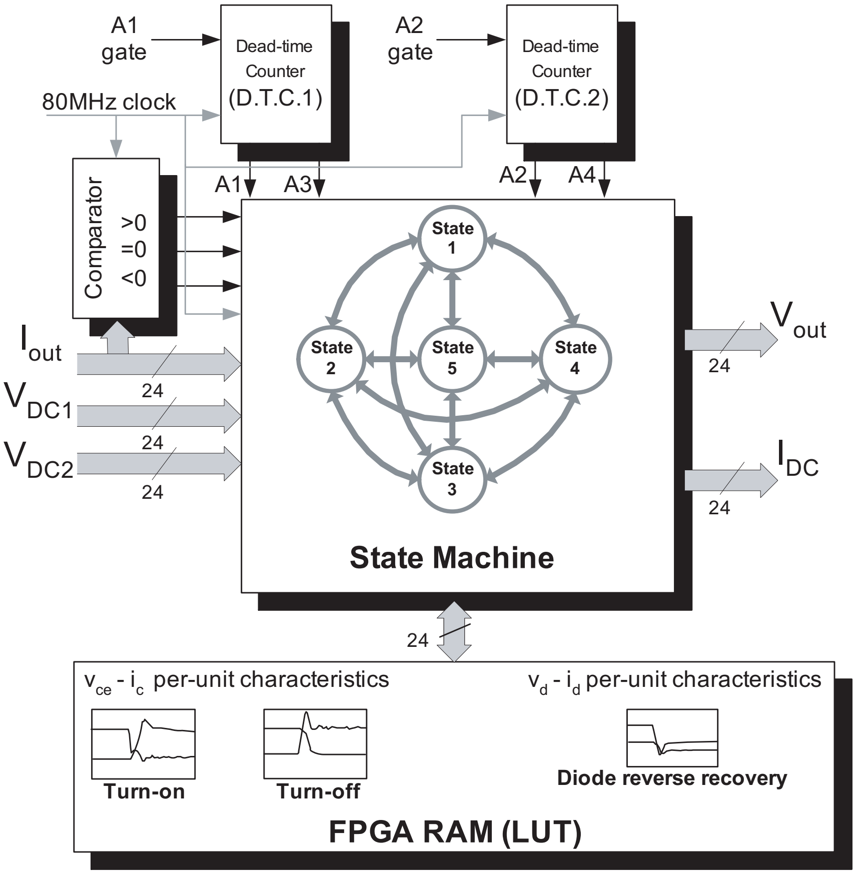


Figure 4.9: Functional block diagram of the hardware realization of IGBT non-linear device-level model for one VSC leg.

of IGBT, the instantaneous V_{ce} and I_c can be calculated. For example, the gating signal is on and the IGBT characteristics of $V_{ce(on)}$ and $I_{c(on)}$ start to read. The ROM can be initialized using a Memory Initialization File (.mif) which can be created using a simple text editor. The *address* and *q* ports are selected from the ROM.

After the model of one leg of the converter was built, it was duplicated to build the other two legs to implement the three phase VSC. The overall VSC model requires as inputs, the three phase currents, the DC voltage and the six gating signals.The outputs of model are the three voltage V_a, V_b, and V_c with respect to the ground bus and the current I_{DC}. All the variables are considered

Table 4.3: Number format for the VSC model variable.

V_{DC1}, V_{DC2}	19.5	*input*
i_a	19.5	*input*
i_b	19.5	*input*
i_c	19.5	*input*
$A1, A2, A3, A4$	1	*input*
v_a	19.5	*output*
v_b	19.5	*output*
v_c	19.5	*output*
I_{DC}	19.5	*output*

as being signed fixed point numbers except the gate signals which are single bits.

Table 4.3 summarizes the number format for the input and output variables. In this table, 19.5 means that 19 bits are used to represent the integer part of the number and 5 bits are used to represent the fraction part.

The internal variables of the model usually have more fractional bits than the input/output variables in order to preserve the resolution of the simulation calculations. The developed model considers that in one simulation time-step there is no change in the current signal. This is a valid assumption since the time-step chosen for the VSC model was 12.5ns.

4.3.2 Induction machine hardware realization

The induction motor is implemented by using the Altera's DSP block sets and generated the VHDL source code. The sample time for the motor is 10μs. The rest of the parts are directly written by the VHDL language. In the induction motor, the torque T^* and the three phase voltages are as inputs, and the currents, speeds and electrical torque are as outputs. The machine model must be discretized for the hardware realization by using the *Trapezoidal* rule. The nonlinear model of motor was implemented by using the arithmetic function DSP blocks such as subtractors, adders, and multipliers, etc.

The complete FPGA hardware implementation of this machine model is described in [4]. Fig. 4.10 shows the schematic of this implementation. Trapezoidal Rule with a time-step of 10μs is used to descretize the model given by (3.9)-(3.17). The DSP Toolbox from Altera was used to implement the model in MATLAB/SIMULINK, and the generated VHDL code was integrated with the rest of the system. In addition to the stator and rotor current components, the model also outputs the rotor flux magnitude and position which are used to implement the field oriented control in the next section.

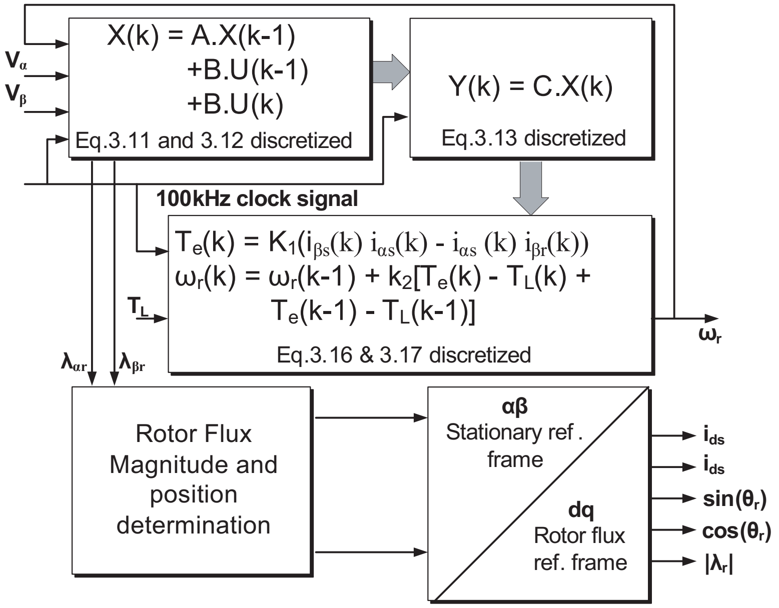


Figure 4.10: Functional block of the implemented induction machine model.

4.3.3 Direct field oriented control hardware realization

For the direct field oriented controller (FOC) shown in Fig. 3.22, the three phase currents, the rotor flux and the speeds of motor are as inputs. There are the reference speed and the reference rotor flux in the system. By using the PI controller, FOC generates the controlled signals v_a^*, v_b^* and v_c^*. The FPGA implementation of the entire control scheme and PI controller are shown in Fig. 3.22 and Fig. 4.11 respectively. These control signals go to the STPWM.

Table 4.4: Number format for the controller variable.

$Variable$	$Format$	$input/output$
i_{ds}	19.13	*input*
i_{qs}	19.13	*input*
$sin(\theta_r)$	2.11	*input*
$cos(\theta_r)$	2.11	*input*
$\|\lambda_r\|$	19.13	*input*
$\|\lambda_r\|^*$	19.13	*input*
ω_r	19.13	*input*
ω_r^*	19.13	*input*
v_α	19.5	*output*
v_β	19.5	*output*

The sample time for the controller is 62.5 μs. Table 4.4 summarizes the number format for the input and output variables.

The fraction and integer parts of results are around 5-bits and the 19-bits, and are used in order to calculate the adequate accuracy for the output voltages and currents. The internal variables are more fractional bits than inputs and outputs to get more precision results. They are implemented in the Matlab/Simulink by using the Altera's DSP toolbox. And then generated the VHDL codes and integrated with other parts of the system.

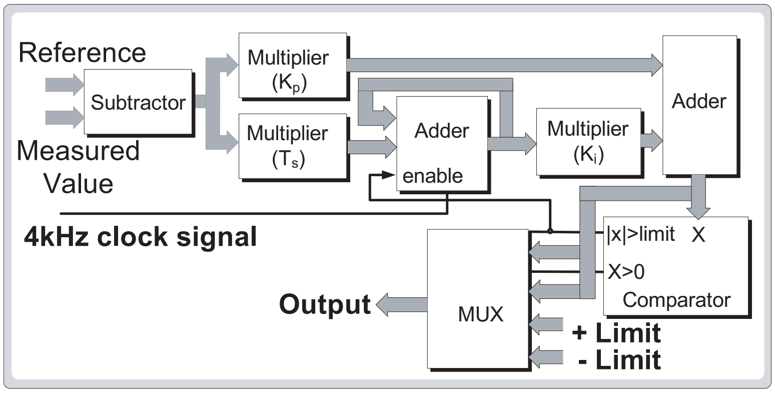


Figure 4.11: Implementation of PI controller.

4.3.4 STPWM hardware realization

The schematic of STPWM hardware implementation is shown in Fig. 4.12. The carrier frequency for the PWM is 8kHz (125 μs) while the controller sampling frequency is 16kHz (62.5 μs). As shown in Fig. 3.20, two synchronized triangle carriers are compared with the control signals (v_a^*, v_b^*, v_c^*) to generate the necessary switching signals.

Once the control signals values are sampled, they are transformed to the 3 phase system v_a^*, v_b^* and v_c^*. The comparison with control signals and the carriers waves can be seen in Fig. 4.12 and will generate the six gating signals with respect to the two channels for each phase. The carrier waveforms are generated at a resolution of 12.5ns in the FPGA with the 16-bit up-down counters. The *Sync* signal is used to synchronize the carrier waveform to the control waveform. The *reset* signal resets the counters at the top and bottom limits of the carrier signal. The gating signals coming from the STPWM need to include the dead-time. Each gating signal passes through the dead-time (2μs in this case) to avoid the DC short circuit and to reduce the IGBT stress and generate its complementary signal. Therefore, the total of 12 gating signals are obtained for three VSC legs.

4.3.5 FPGA hardware resources utilized for implementation

The FPGA logic resources utilized in Stratix and Stratix III devices for the entire system models are shown in Table 4.5 and 4.6. We can notice that there are five IGBT models with different usages of LEs and Memory bits. The switching function (SF) model consumes the least LEs because it is a direct function of the IGBT gating signals, whereas the nonlinear IGBT model is the biggest usage of memory bits for storing the IGBT switching characteristics in the ROM. The induction machine model uses more than half of the LEs of the whole system. It was developed by using the Altera DSP blocks sets and

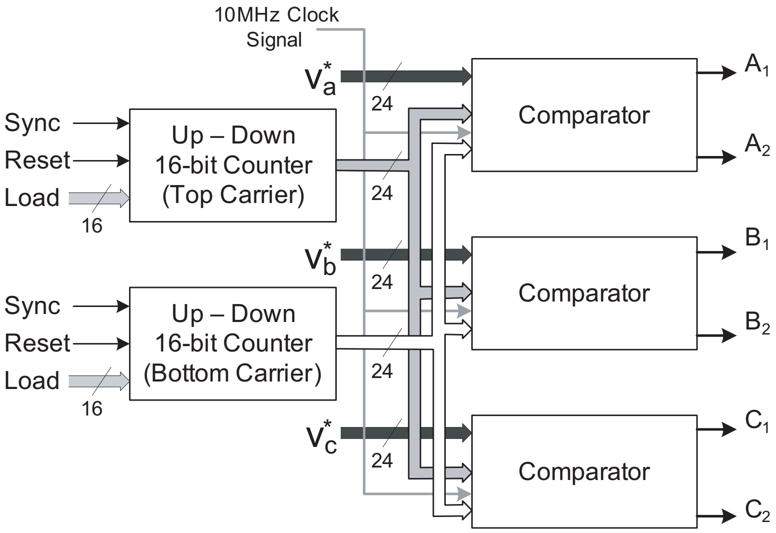


Figure 4.12: Functional block diagram of the FPGA hardware realization for 3-level PWM.

consumes more LEs.

4.4 Results and Discussion

In this section, results are presented from the real time simulation and off-line simulation of the 3-level VSC variable speed AC drive.

4.4.1 Off-line validation for real-time simulation

The off-line simulation, using these two well-known simulation programs, SABER and SIMULINK, was carried out with an objective of validating the real-time simulation of the variable speed AC drive. Fig. A.1 and Fig. A.2 show the block diagrams of the off-line system implementation in MATLAB/Simulink and SABER environment. A co-simulation environment was set up by interfacing, the 3-level converter and PWM modeled in SABER with the rest of the system (machine and controller) modeled in MATLAB/SIMULINK. A non-linear model of the IGBT1 and Diode was used in SABER to model the 3-level VSC.

Table 4.5: EP1S80 FPGA resources used by the system components

Component	**LEs**	**DSPs**	**PLLs**	**Memory bits**
VSC (Ideal)	7616	8	0	0
VSC (Switching Function)	6595	0	0	0
VSC (Average)	6745	8	0	0
VSC (Linear)	7979	0	0	0
VSC (Nonlinear)	7322	8	0	17920
Induction Machine	20529	136	0	0
Controller	5310	32	0	0
STPWM	235	0	0	0
Measurement System	72	8	0	0
D/A Interface	18	0	0	0
Total Usage (linear model)	34143	176	1	17920

Table 4.6: EP3SL150 FPGA resources used by the system components

Component	**LEs**	**DSPs**	**PLLs**	**Memory bits**
VSC (Ideal)	0	60	0	0
VSC (Switching Function)	0	0	0	0
VSC (Average)	0	60	0	0
VSC (Linear)	10790	0	0	0
VSC (Nonlinear)	0	60	0	17920
Induction Machine	12314	130	0	0
Controller	3095	69	0	0
STPWM	552	0	0	0
Measurement System	150	8	0	0
D/A Interface	255	0	0	0
Total Usage (nonlinear model only)	26627	259	1	17920

The equivalent model circuit of IGBT1 in SABER [25] is represented in the Fig. 4.13. The three interelectrode capacitances (identical to the MOSFET model) are accurately modeled to calculate switching waveforms and losses. C_{ce} and C_{cg} are voltage dependent whereas C_{ge} is assumed constant. The current source I_{mos} is controlled by the collection-emitter and gate-emitter voltages. A piece-wise linear diode is added in series with I_{mos} to represent the V_{ce} threshold voltage above which the IGBT conducts. The other elements model turn-off tail current. In addition to a comparative study of results from the two programs the co-simulation in understanding the advantages and limitations of both programs. Steady state and transient results are recorded and then compared. The comparison reveals high degree of similarity among the results obtained through both simulation environments with minor discrepancies. The SABER simulation runs at 12.5ns while the interfaced MATLAB/SIMULINK simulation runs at 10μs.

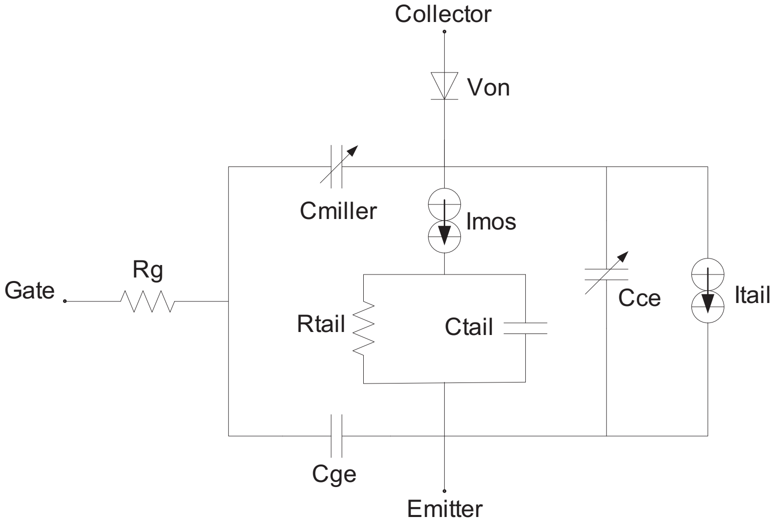


Figure 4.13: IGBT Equivalent model circuit of SABER.

4.4.2 IGBT switching characteristics

The IGBT switching characteristics for the system level and device level models are shown in Fig. 4.14 to Fig. 4.16. The characteristics for all system-level models are the same except for the averaged model. As shown in Fig. 4.14(a) and Fig. 4.14(b) for the switching function model, the IGBT transients only include simple on and off without any time delays or nonlinear transients. It can be clearly seen that there is no turn-on delay $t_{d(on)}$, or turn-off delay $t_{d(off)}$ and there is a straight forward relationship with the gating signals.

For the linearized device-level IGBT model simulation, we can notice in Fig. 4.15(a) and Fig. 4.15(b) that there are the turn-on $t_{d(on)}$ and turn-off delay $t_{d(off)}$, and the rise time t_r and fall time t_f of IGBT. In the Fig. 3.7 and Fig. 3.8, we can compare with the measured characteristics of IGBT.

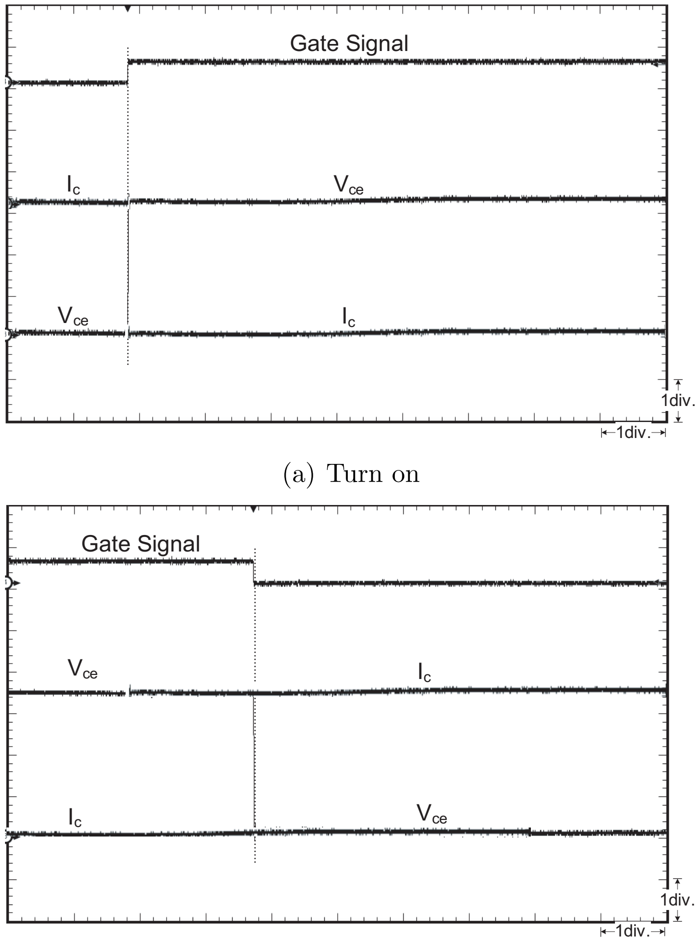
We can see the real-time simulation results of non-linear device-level models on the oscilloscope in Fig. 4.16(a) and Fig. 4.16(b). Comparing these figures with the measured characteristics in Fig. 3.7 and Fig. 3.8, we can see that they are quite satisfactory.

4.4.3 Steady-state results

For steady-state analysis, the system has been simulated for a few seconds, however, the results are shown for shorter periods due to the larger amount of data as well as the clarity of the figures. There were some initial transients which subsided very fast and the system reached steady-state thereafter.

Figs. 4.18 to 4.21 show the steady-state waveforms of the v_β voltages and the harmonics spectrum of the voltages at the inverter frequency of 60Hz. As expected, all voltages are very similar in all cases and the differences only appear in the averaged model. For the averaged model, the FFT calculation result of switching frequency components is very small compared with other models.

All figures show the output voltages of VSC after the induction machine



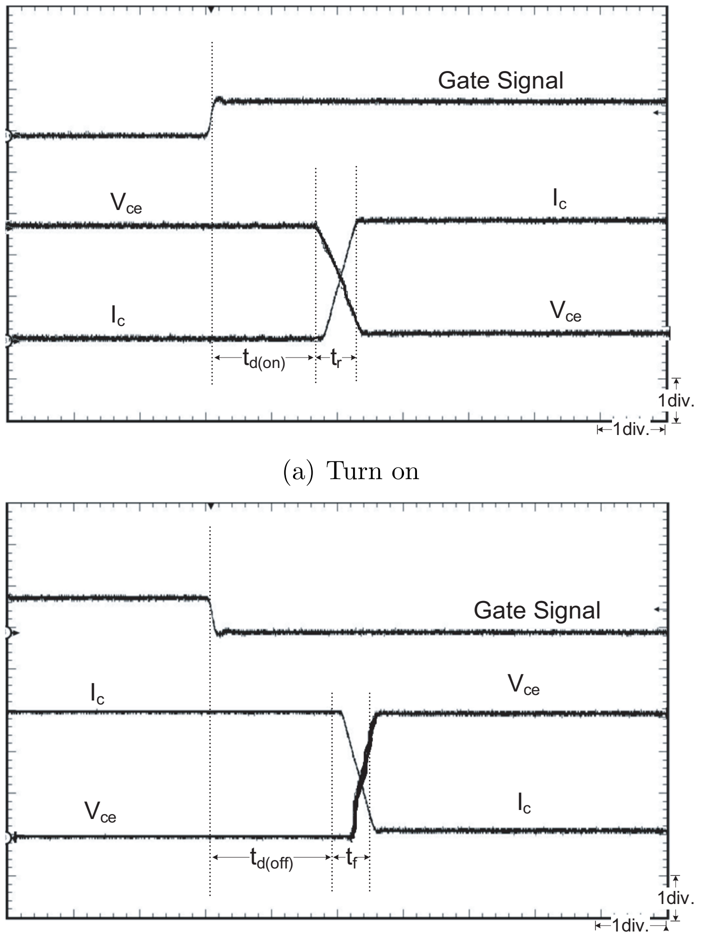
(a) Turn on

(b) Turn off

Figure 4.14: Real-time simulation results for switching function IGBT model. [V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.]

reaches its steady-state with $\omega_r^* = 377$rad/s. In these figures, it is possible to observe that the typical 3-level voltages present in the converter phase voltage. The FFT calculated by the oscilloscope shows a fundamental frequency component at $f = 60$Hz. Other significant components are the sidebands centered at $f_c = 8$kHz and $f_c = 16$kHz, since the PWM frequency is equal to 8kHz. The line voltage magnitudes for various IGBT models are given in Table 4.7.

A comparison of steady-state results from the real-time simulation and the



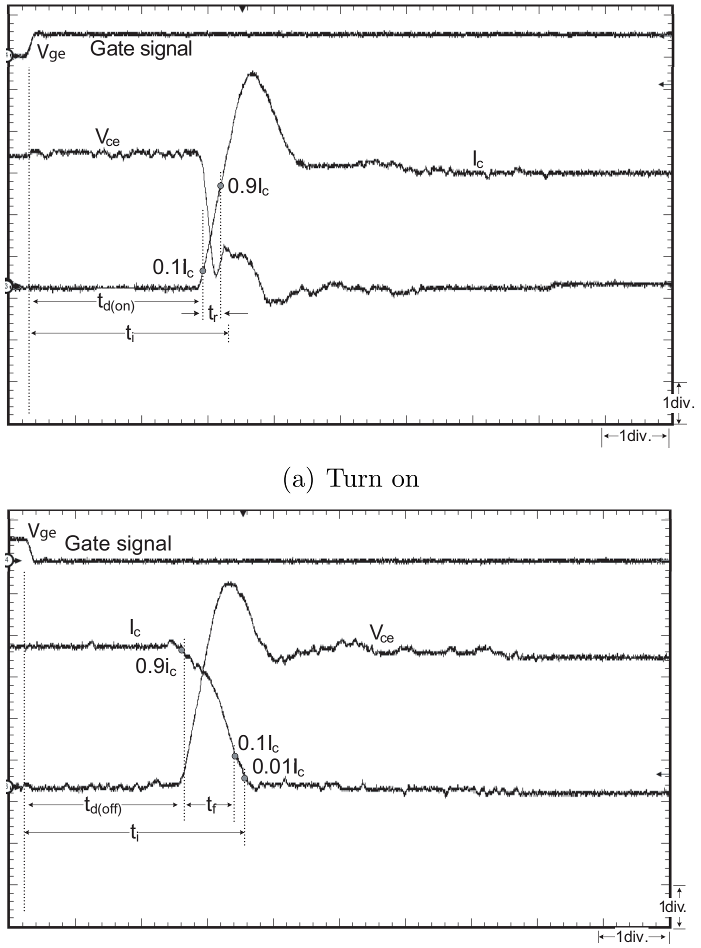
(a) Turn on

(b) Turn off

Figure 4.15: Real-time simulation results of linearized device-level IGBT model. [V_{ce}: 100 V/div., V_{ge}: 20 V/div., I_c: 5 A/div., time: 200 ns/div.]

off-line simulation is shown in Fig. 4.22 and Fig. 4.23 respectively. These two figures show the inverter line voltage, the machine line current, and the FFT of the line voltage. Both sets of results are quite comparable.

For off-line simulation, the 3-level VSC bridge which was built by the SABER nonlinear model of IGBTs and Diodes was used with a time step of 10μs. The higher sample time models for the motor and controller are simulated with MATLAB/SIMULINK with a time step of 10μs. Therefore, we can see the current distortion in off-line simulation and jitter of the waveform. In



(a) Turn on

(b) Turn off

Figure 4.16: Real-time simulation results of nonlinear device-level IGBT model. [V_{ce}: 100 V/div., V_{ge}: 30 V/div., I_c: 5 A/div., time: 200 ns/div.]

the real time simulation, the IGBTs are by non-linear characteristics and data points are spaced at 12.5ns time-step for the VSC model represented. Therefore, the line current is found to be quite smooth and sinusoidal due to this reason as well as the high inductance of the machine.

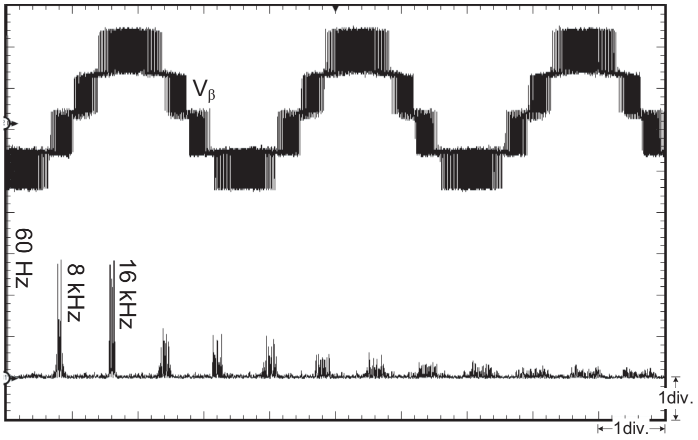


Figure 4.17: Output voltage waveform of 3-level VSC with ideal model for IGBT. [v_β: 500V/div., time: 5ms/div.]

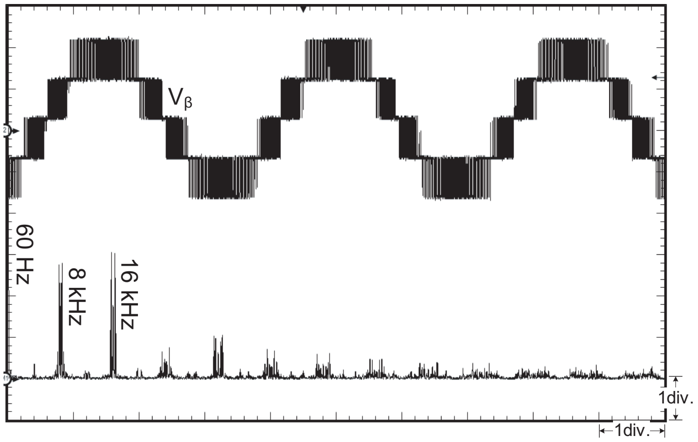


Figure 4.18: Output voltage waveform of 3-level VSC with switching function model for IGBT. [v_β: 500V/div., time: 5ms/div.]

4.4.4 Transient results

The system transient results are shown in Fig. 4.24 and Fig. 4.25 from the real time simulation and the off-line simulation respectively. The transient results are the same regardless of the IGBT models used in VSC because they are at a system-level. These figures show

the machine speed ω_r, the speed reference ω_r^*, and the currents i_α and i_β when the machine is subjected to the following transients:

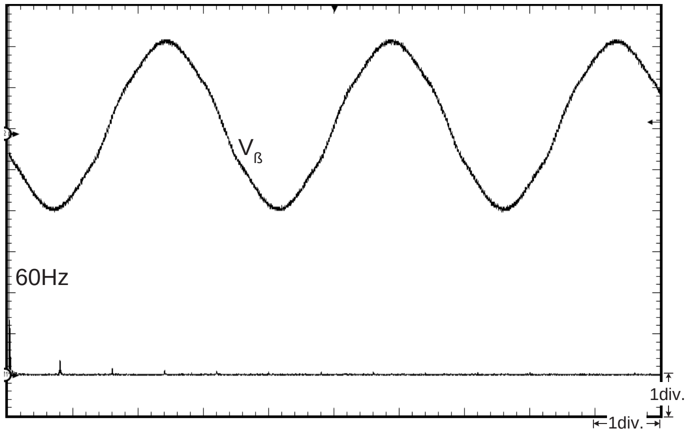


Figure 4.19: Output voltage waveform of 3-level VSC with averaged model for IGBT. [v_β: 500V/div., time: 5ms/div.]

Table 4.7: The line voltage magnitudes for various IGBT models

IGBT	$V_{ll}(peak)V$	$60Hz$	$8kHz$	$16kHz$
Ideal	750	100%	16%	16%
Switching Function	750	100%	17%	20%
Averaged	750	100%	5%	3%
Linear	750	100%	14%	16%
Nonlinear	750	100%	12%	13%

1. the startup of machine with speed set-point of 150rad/s at t_1,
2. the speed set-point variation to 377rad/s at t_2,
3. the application of 100N.m load at t_3 and removal of the load at t_4, and
4. the speed reverse to -377rad/s at t_5.

As can be observed from the machine torque and speed, the controller has a stable and fast performance during the transients and no perturbation can be seen in the speed during the application or removal of the load. As the machine was modeled by ignoring the mechanical losses, and the machine resistances are quite small, during the transients, the converter must supply or absorb the power consumed or generated by the machine as a function of the controller

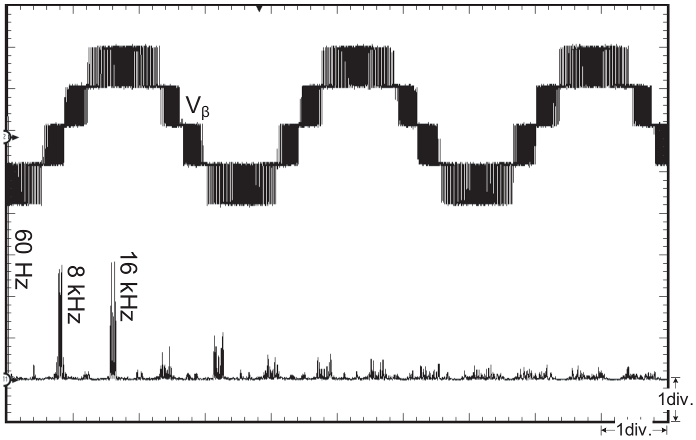


Figure 4.20: Output voltage waveform of 3-level VSC with linearized device-level model for IGBT. [v_β: 500V/div., time: 5ms/div.]

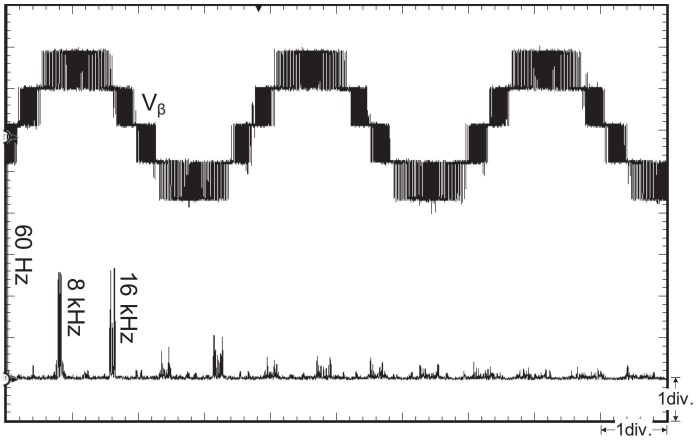


Figure 4.21: Output voltage waveform of 3-level VSC with nonlinear device-level model for IGBT. [v_β: 500V/div., time: 5ms/div.]

action. It is also possible to observe that the frequency variation of the currents and voltages during the applied load, speed-up and speed reversal.

4.5 Summary

The chapter provided the details on hardware platform used for the FPGA implementation. Subsequently, all of the individual component models of the variable speed AC drive are implemented in digital hardware. Oscilloscope results

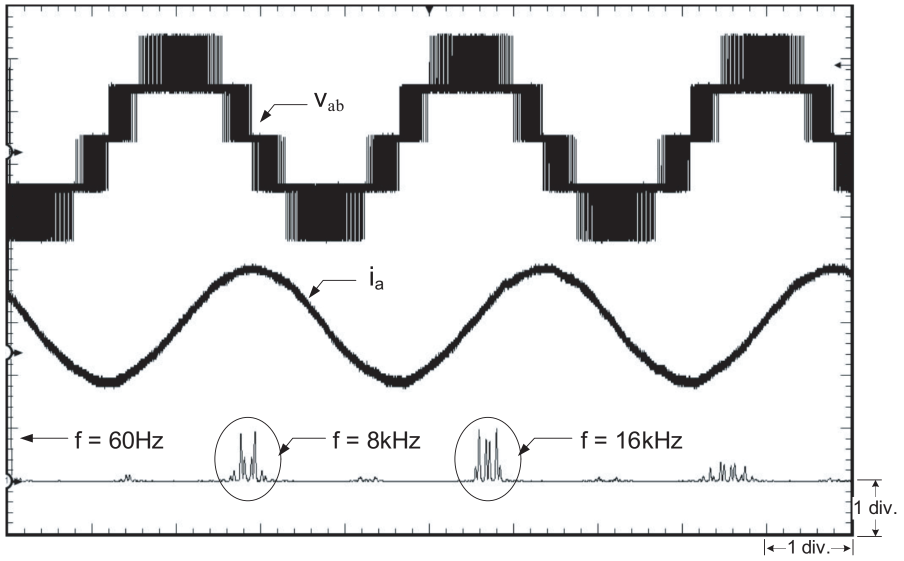


Figure 4.22: Real-time simulation oscilloscope trace of steady-state line voltage v_{ab} and line current i_a applied to the machine, and harmonic spectrum of v_{ab}. [Scale: v_{ab}: 350V/div., i_a: 20A/div., Time: 5ms/div.]

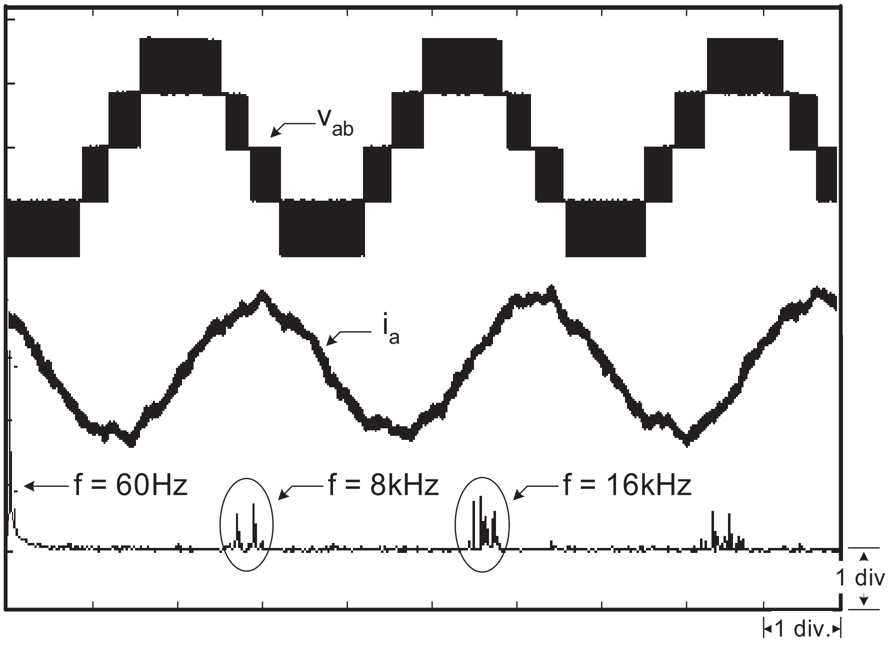


Figure 4.23: Off-line SABER/SIMULINK simulation results of steady-state line voltage v_{ab} and line current i_a applied to the machine, and harmonic spectrum of v_{ab}. [Scale: v_{ab}: 350V/div., i_a: 20A/div., Time: 5ms/div.]

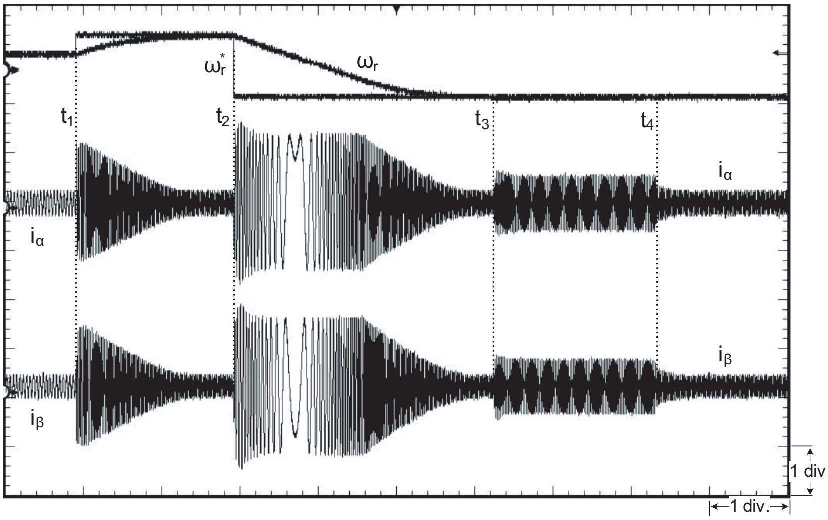


Figure 4.24: Real-time simulation oscilloscope trace of reference speed ω_r^*, machine speed ω_r and currents i_α, i_β for: machine running at 150rad/s and speed set-point variation to 377rad/s at t_1, speed reversal to -377rad/s at t_2, application and removal of load at t_3 and t_4, respectively. [Scale: ω_r^*, ω_r: 630rad/s/div., i_α, i_β: 150A/div., Time: 1s/div.]

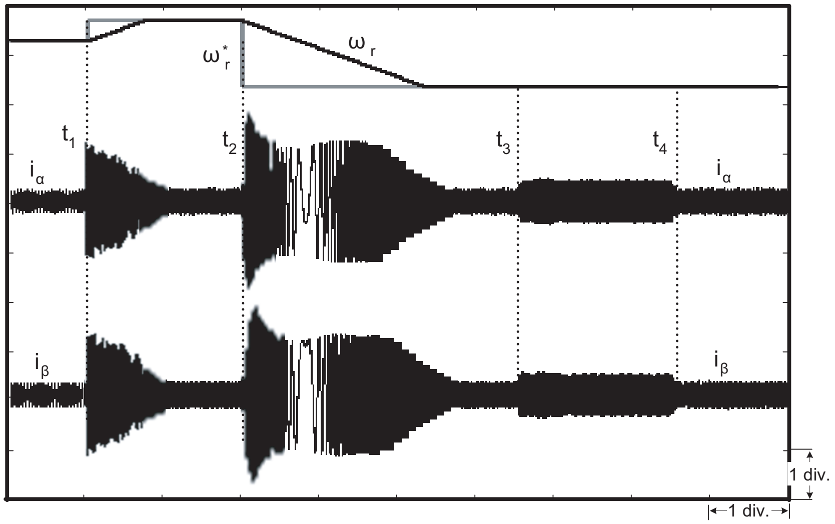


Figure 4.25: Off-line SABER/SIMULINK simulation results of reference speed ω_r^*, machine speed ω_r and currents i_α, i_β for: machine running at 150rad/s and speed set-point variation to 377rad/s at t_1, speed reversal to -377rad/s at t_2, application and removal of load at t_3 and t_4, respectively. [Scale: ω_r^*, ω_r: 630rad/s/div., i_α, i_β: 150A/div., Time: 1s/div.]

are shown to prove the performance of the real-time simulator. Furthermore, the discussion on FPGA logic resources gives an idea of the implementation complexity and resource allocation for various models. Off-line results from a SABER/SIMULINK co-simulation are provided side-by-side with the real-time results for further validation.

5 Conclusions and Future Work

This thesis has presented a real time simulation of the variable speed AC drive using the FPGA as the core computational engine. A summary of the major contributions of this work is presented in the following. In the end, suggestions for extending this research future are presented.

5.1 Contributions of This Work

The main contributions of this thesis are the following:

1. A detailed device-level nonlinear model of an IGBT based on per-unit characteristics stored in the look-up table is proposed for developing the voltage source converter model. This non-linear model can realistically predict the IGBT switches losses in the device and reproduce the IGBT switching characteristics.

2. The IGBT per-unit switching characteristics for the proposed nonlinear model are derived based on experimental measurements.

3. The implementation of the 3-level VSC model in real-time, was carried out using 5 IGBT models: 3 system level models (ideal, switching function and averaged model), and 2 device-level models (linearized, and nonlinear).

4. The complete variable speed AC drive has been implemented on the Stratix and Stratix III FPGAs using VHDL. The system components include the 3-level VSC, the induction machine, the direct field oriented controller, and the pulse width modulator. The FPGA platforms were interfaced to an external DAC board to display real-time results on the oscilloscope.

5. The real-time results are validated using an off-line co-simulation setup using the SABER and MATLAB/SIMULINK software. The SABER software models the induction machine and the field oriented controller. The off-line results match the real-time results quite well.

5.2 Suggestions for future work

1. A detailed device-level nonlinear behavioral model of the IGBT is suggested for future research. The model will require the solution of IGBT nonlinear equations using a solver such as Newton-Raphson.

2. A multi-level converter based on a behavioral nonlinear IGBT model will be of great signification for accurate real-time simulation since it would obviate the need to measure and store switching characteristics. Of course, such a model would require larger FPGA resources.

3. A detailed machine model that would provide a better insight into internal machine variables rather than simply the external performance indices is also suggested for future work. Such a model would help in the efficient design and development of AC drive.

4. A hardware platform that can exploit multiple FPGAs would enable larger and more detailed drive systems to be implemented in real-time.

Bibliography

[1] A. Nabae, I. Takahashi, and H. Akagi, "A new neutral-point-clamped PWM Inverter," in *IEEE Trans. on Industry Application*, vol. IA-17, no. 5, Page(s): 518 - 523, Sept. 1981.

[2] A. G. Jack, D. J. Atkinson, and H. J. Slater, "Real-time emulation for power equipment development. i. real-time simulation," in *IEE Proc. on Electrical Power Applications*, vol. 145, no. 2, Page(s): 92 - 97, March 1998.

[3] H. Le-Huy, G. Sybille, P. Giroux, J. Soumagne, and F. Guay, "Digital real-time simulation of a four-quadrant DC drive for static transfer switch testing," in *Proc. of IEEE PES Summer Meeting*, Page(s): 761 - 765, 1998.

[4] G. G. Parma, and V. Dinavahi, "Real-Time Digital Hardware Simulation of Power Electronics and Drives," in *IEEE Trans. on Power Delivery*, vol. 22, no.2, Page(s): 1235 - 1246, April 2007.

[5] R. Champagne, L-A. Dessaint, H. Fortin-Blanchette, and G. Sybille, "Analysis and validation of a real-time AC drive simulator," in *IEEE Trans. on Power Electronics*, vol. 19, no. 2, Page(s): 336 - 345, March 2004.

[6] M. O. Faruque, V. Dinavahi, and W. Xu, "Algorithms for the accounting of multiple switching events in digital simulation of power-electronic systems," in *IEEE Trans. on Power Delievery*, vol. 20, no. 2, Page(s): 1157 - 1167, April 2005.

[7] S. Jung, M. Chang, J. Jyang, L. Yeh, and Y. Tzou, "Design and implementation of an FPGA-based control IC for AC-voltage regulation," in *IEEE Trans. on Power Electronics*, vol. 14, no. 3, Page(s): 522 - 532, May 1999.

[8] A. de Castro, P. Zumel, O. Garcia, T. Riesgo, and J. Uceda, "Concurrent and simple digital controller of an AC/DC converter with power factor correction based on an FPGA," in *IEEE Trans. on Power Electronics*, vol. 18, no. 1, Page(s): 334 - 343, January 2003.

[9] V. Dinavahi, M. R. Iravani, and R. Bonert, "Design of a real-time digital simulator for a D-STATCOM system," in *IEEE Trans. on Industrial Electronics*, vol. 51, no. 5, Page(s): 1001 - 1008, October 2004.

[10] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery*, IEEE Press, New York, January, 1995.

[11] R. Kuffel, J. Giesbrecht, T. Maguire, R. P. Wierckx, and P. McLaren, "RTDS-a fully digital power system simulator operating in real time," in *Int'l Conf. on Energy Management and Power Delivery*, vol. 2, no. 2, Page(s): 498 - 503, Nov. 1995.

[12] Lok-Fu Pak, Dinavahi, V., G. Chang, M. Steurer, and P.F. Ribeiro, "Real-Time Digital Time-Varying Harmonic Modeling and Simulation Techniques IEEE Task Force on Harmonics Modeling and Simulation," in *IEEE Trans. on Power Deliever*, vol. 22, no. 2, Page(s): 1218 - 1227, Apr 2007.

[13] L. Pak, M. O. Faruque, X. Nie, and V. Dinavahi, "A versatile cluster-based real-time digital simulator for power engineering research," in *IEEE Trans. on Power Systems*, vol. 21, no. 2, Page(s): 1 - 11, May 2006.

[14] J. M. Conrad, and D. P. Agrawal, "Distributed, scalable, and static parallel arc consistency algorithms on private memory machines," in *IEE Proceed-*

ings of the 12th International Conference on Distributed Computing Systems, vol. 137, no.4, Page(s): 442 - 449, June 1992.

[15] G. B. Lee, J. K. Kwon, J. M. Ahn, I. J. Kang, S. Y. Shin, and S. J. Ahn, "Application-database interface design for the real time simulator system," in *International Conference on, Control, Automation and Systems, 2007. ICCAS '07*, vol. 137, no.4, Page(s): 1868 - 1872, Oct 2007.

[16] B. Johnson, H. Hess, and J. A. Martinez, "Parameter Determination for Modeling System TransientsPart VII: Semiconductors," in *IEEE Trans. on Power Delivery*, vol. 20, no. 3, Page(s): 2086 - 2095, July 2005.

[17] *RT-LAB 8.0 User Mannual*, Opal-RT Technologies Inc., Montreal, Canada, 2006.

[18] A. M. Gole, A. Keri, C. Kwankpa, E. W. Gunther, H. W. Dommel, I. Hassan, J. R. Marti, J. A. Martinez, K. G. Fehrle, L. Tang, M. F. McGranaghan, O. B. Nayak, P. F. Ribeiro, R. Iravani, and R. Lasseter, "Guidelines for modeling power electronics in electric power engineering applications," in *IEEE Trans. on Power Delivery*, vol. 12, no. 1, Page(s): 505 - 514, January 1997.

[19] K. Sheng, B. W. Williums, and S. J. Finney, "A Review of IGBT Models," in *IEEE Trans. on Power Electronics*, vol. 15, no. 6, Page(s): 1250 - 1266, November 2000.

[20] A. Hefner, and D. Blackburn, "An analytical model for the steady-state and transient characteristics of the power insulated-gate bipolar transistor," in *Solid State Electronics*, vol. 31, no. 10, Page(s): 1513 - 1532, Oct 1988.

[21] H. Jin, "Behavior-mode simulation of power electronic circuits," in *IEEE Trans. on Power Electronics*, vol. 12, no. 3, Page(s): 443 - 452, May 1997.

[22] K. L. Lian, and P. W. Lehn, "Real-Time Simulation of Voltage Source Converters Based on Time Average Method," in *IEEE Trans. on Power Systems*, vol. 20, no.1, Page(s): 110 - 118, February 2005.

[23] *Stratix Device Handbook, Volume 1, 2, 3*, Altera, July 2005.

[24] K. Strunz, and E. Carlson, "Nested fast and simultaneous solution for time-domain simultion of integrtive power-electric and electronic systems," in *IEEE Trans. on Power Delivery*, vol. 22, no. 1, Page(s): 277 - 287, January 2007.

[25] *SABER User Guide*, Synopsys, September 2005.

[26] *Stratix Device Handbook, Volume 1, 2, 3*, Altera, July 2005.

[27] K. Sheng, B. W. Williams, and S. J. Finney, "A review of IGBT models," in *IEEE Trans. on Power Electronics*, vol. 15, no. 6, Page(s): 1250 - 1266, November 2000.

[28] R. Kraus, and K. Hoffmann, "An analytical model of IGBTs with low emitter efficiency," *ISPSD93*, Monterey CA, vol. 3, Page(s): 30 - 34.

[29] A. D. Rajapakse, A. M. Gole, and P. L. Wilson, "Electromagnetic Transients Simulation Models for Accurate Representation of Switching Losses and Thermal Performance in Power Electronic Systems," in *IEEE Trans. on Power Delivery*, vol. 20, no.1, Page(s): 319 - 327, January 2005.

[30] C. Wong, "EMTP modeling of IGBT dynamic performance for power dissipation estimation," in *IEEE Trans. on Industry Applications*, vol. 33, no. 1, Page(s): 64 - 71, January/February 1997.

[31] J. T. Hsu, and Khai D. T. Ngo, "Behavioral modeling of the IGBT using the Hammerstein Configuration," in *IEEE Trans. on Power Electronics*, vol. 11, no. 6, Page(s): 746 - 754, November 1996.

[32] B. J. Baliga, *Power Semiconductor Devices*, PWS Publishing Co., Boston, 1996.

[33] D. G. Holmes, and T. A. Lipo, *Pulse Width Modulation for Power Converters*, IEEE Press, Wiley Interscience, 2003.

[34] G. C. Verghese, and S. R. Sanders, "Observers for flux estimation in induction machines," in *IEEE Trans. on Power Electronics*, vol. 35, no. 1, Page(s): 85 - 94, Feb 1988.

[35] N.Celanovic, and D.Boroyevich, "A fast Space-Vector modulation algorithm for multilevel three-phase converters," in *IEEE Trans. on Ind. Application*, vol. 37, no.2, Mar/Apr 2001.

[36] Y. Tzou, and H. Hsu, "FPGA realization of space-vector PWM control IC for three-phase pwm inverters," in *IEEE Trans. on Power Electronics*, vol. 12, no. 6, Page(s): 953 - 963, October 1997.

[37] F. Blaabjerg, F. K. Pedersen S. Sigurjonsson, and A. Elkjaer, "Extended model of power losses in hard-switched inverters," in *Proc. Conf. Rec. IAS Annu. Meeting (IEEE Ind. Applicat. Soc.)*, vol. 3, Page(s): 1454 - 1463, 1996.

[38] A. N. Githiari, B. M. Gordon, R. A. McMahon, Z. Li, and P. H. Mawby, "A comparison of IGBT models for use in circuit design," in *IEEE Trans. on Power Electronics*, vol. 14, no. 4, Page(s): 607 - 613, July 1999.

[39] V. -Q. Do, D. McCallum, P. Giroux, and B. D. Kelper, "A backward-forward interpolation technique for a precise modeling of power electronics in HYPERSIM," in *IPST01-International Conference on Power Systems Transients, Rio de Janeiro, Brazil*, Page(s): 337 - 342, June 2001.

[40] V. R. Dinavahi, M. R. Iravani, and R. Bonert, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers," in

IEEE Trans. on Power Delivery, vol. 16, no. 4, Page(s): 775 - 781, October 2001.

[41] F. Blaschke, "The principles of field orientation as applied to the new transvector closed-loop system for rotating field machines," in *Siemens Review*, vol. 34, Page(s): 217 - 220, 1972.

[42] D. W. Novotny, and T. A. Lipo, "Vector control and dynamics of AC drives," *Oxford University Press, 2000.*

[43] Y. Hori, and T. Umeno, "Implementation of robust flux observer based field orientation (fofo) controller for induction machines," in *IEEE Conference Record of the Industry Applications Society Annual Meeting*, Page(s): 523 - 528, October 1989.

[44] J. T. Boys, and P. G. Handley, "Harmonic analysis of space vector modulated PWM waveforms," in *IEE Proceedings B of Electric Power Applications*, vol. 137, no. 4, Page(s): 197 - 204, July 1990.

[45] Stratix Device Family Overview, "http://www.altera.com/products/devices /stratix/overview/stx-overview.html," *Altera Corporation, Apr 2007.*

[46] Y. H. Lee, B. S. Suh, and D. S. Hyun, "A novel PWM Scheme for a 3-level voltage source inverter with GTO Thyristors," in *IEEE Trans. on Industry Applications*, vol. 32, no.2, Page(s): 260 - 268, Mar/Apr 1996.

[47] T. Ningping, H. Wu, and Q. Peiji, "A saturation model of induction machine by space vector," in *IEEE Trans. on Industry Applications*, vol. 32, no.2, Page(s): 260 - 268, Mar/Apr 1996.

[48] J.O. Ojo, A. Consoli, and T.A. Lipo, "An improved model of saturated induction machines," in *IEEE Trans. on Industry Applications*, vol 26, Issue 2, Page(s): 212 - 221, March-April 1990.

[49] D. Scott , D. C. Aliprantis, B. T. Kuhn, and L. C. Patrick, “An Induction Machine Model for Predicting Inverter Machine Interaction,” in *IEEE Trans. on energy conversion*, vol. 17, no. 2, Page(s): 203 - 212, june 2002.

[50] A. Kaddouri, B. Khodabakhchian, L. A. Dessaint, R. Champagne, and L. Snider, “A new generation of simulation tools for electric drives and power electronics,” in *IEEE Conf. on Power Electronics and Drive Systems*, vol. 1, no. 1, Page(s): 348 - 354, July 1999.

[51] S. J. Langdon, “The Evalution of IGBT Technology,” *New Developments in Power Semiconductor Devices, IEE Colloquium on*, Page(s):2/1 - 2/7, 01 May 1991.

[52] H. W. Dommel, “Nonlinear and time-varying elements in digital simulation of electromagnetic transients,” in *IEEE Trans. on Power Apparatus and Systems*, vol. PAS-90, No. 6, Page(s): 2561 - 2567, November/December 1971.

[53] A. Abur, and H. Singh, “Time domain modeling of external systems for electromagnetic transients programs,” in *IEEE Trans. on Power Systems*, vol. 8, No. 2, Page(s): 671 - 677, May 1993.

[54] Joseph D. Law, and Thomas A. Lipo, “A Single Phase Induction Motor Voltage Controller with Improved Perfornance,” in *IEEE Transactions on power Electronic*, vol. pe-1, No. 1, Page(s): 240 - 248, Oct 1986.

[55] A. Chipperfield, P. Fleming, H. Pohlheim, and C. Fonseca, *Genetic Algorithm Toolbox for MATLAB,* Department of Automatic Control and Systems Engineering, Univerisity of Sheffield.

[56] *MATLAB User Guides* The MathWorks Inc., Natick, MA.

[57] Dorin O. Neacsu, *Power-Switching Converters, Medium and High Power*, Taylor and Francis Press, New York, 2006.

A
Off-line cosimulation

A.1 SABER and Matlab/Simulink cosimulation

- SABER Block diagram of Saber/Simulink Co-simulation.
- Simulink Block diagram of Saber/Simulink Co-simulation.

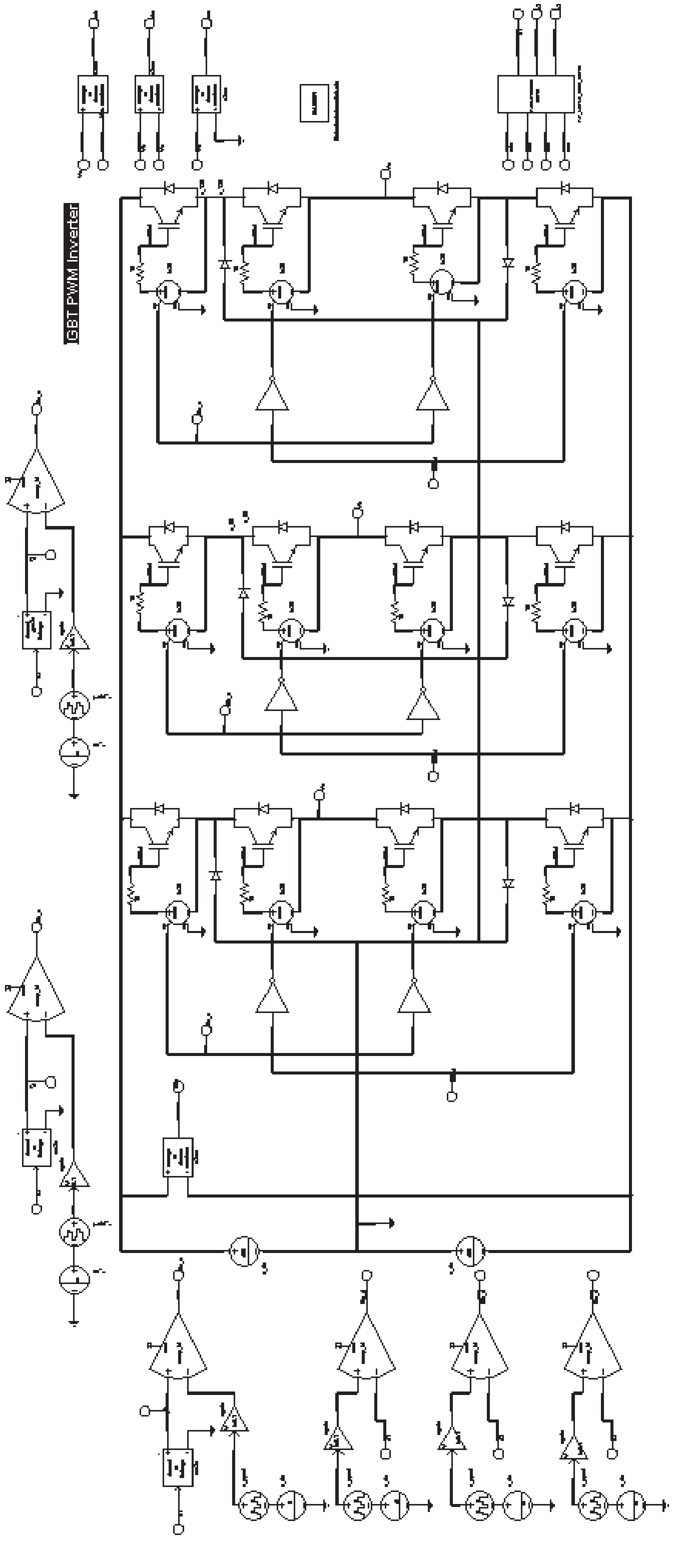


Figure A.1: SABER Block diagram of Saber/Simulink Co-simulation.

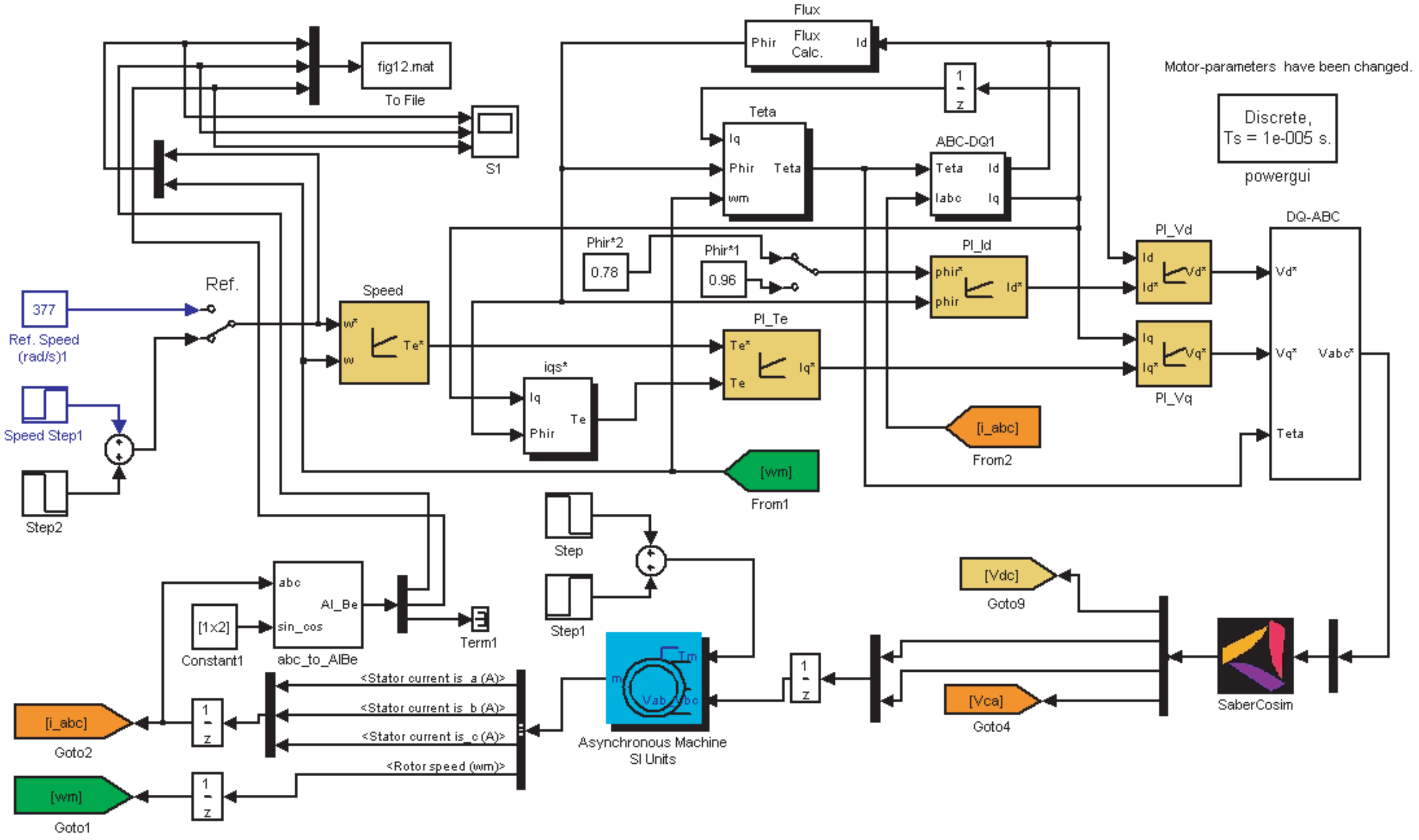


Figure A.2: Simulink Block diagram of Saber/Simulink Co-simulation.

B

FPGA Design Files

This appendix presents the FPGA design files for real-time simulation and experimental control of the three-phase PWM VSC system. The FPGA used is a EP1S80 and EP3SL150F1152C2 devices from ALTERA. The design has been carried out using the QUARTUS 9.0 graphic design software. The list of the graphic design files (.gdf) and a brief description of each file is given below.

B.1 Real-Time Digital Simulator and Controller

- *fpga.gdf* : Top Design File.
- *VSC.gdf* : Voltage Source Converter.
- *control.gdf* : FOC control file 1.
- *control1.gdf* : FOC control file 2.
- *concontrol2.gdf* : FOC control file 3.
- *concontrol3.gdf* : FOC control file 4.

- *control4.gdf* : FOC control file 5.
- *cpwm.gdf* : carrier pulse wave modulation.
- *IGBT1.gdf* : Voltage Source Converter.
- *IGBT2.gdf* : FOC control file 1.
- *IGBT3.gdf* : FOC control file 2.
- *IGBT4.gdf* : FOC control file 3.
- *IGBT5.gdf* : FOC control file 4.
- *measure.gdf* : FOC control file 5.
- *motor.gdf* : motor file.

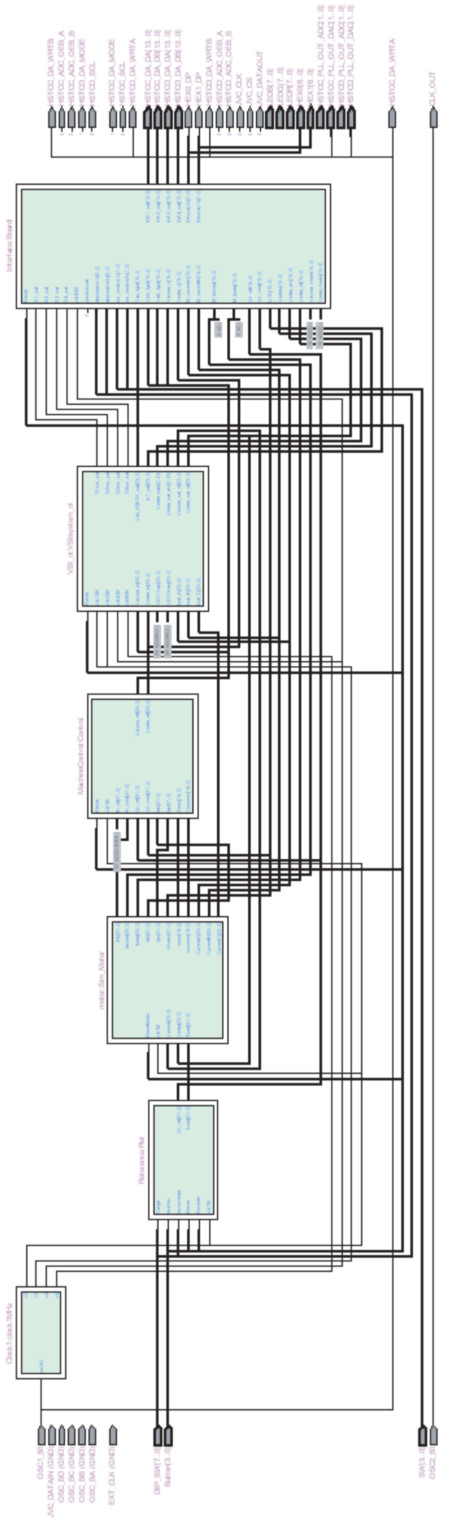


Figure B.1: fpga.gdf.

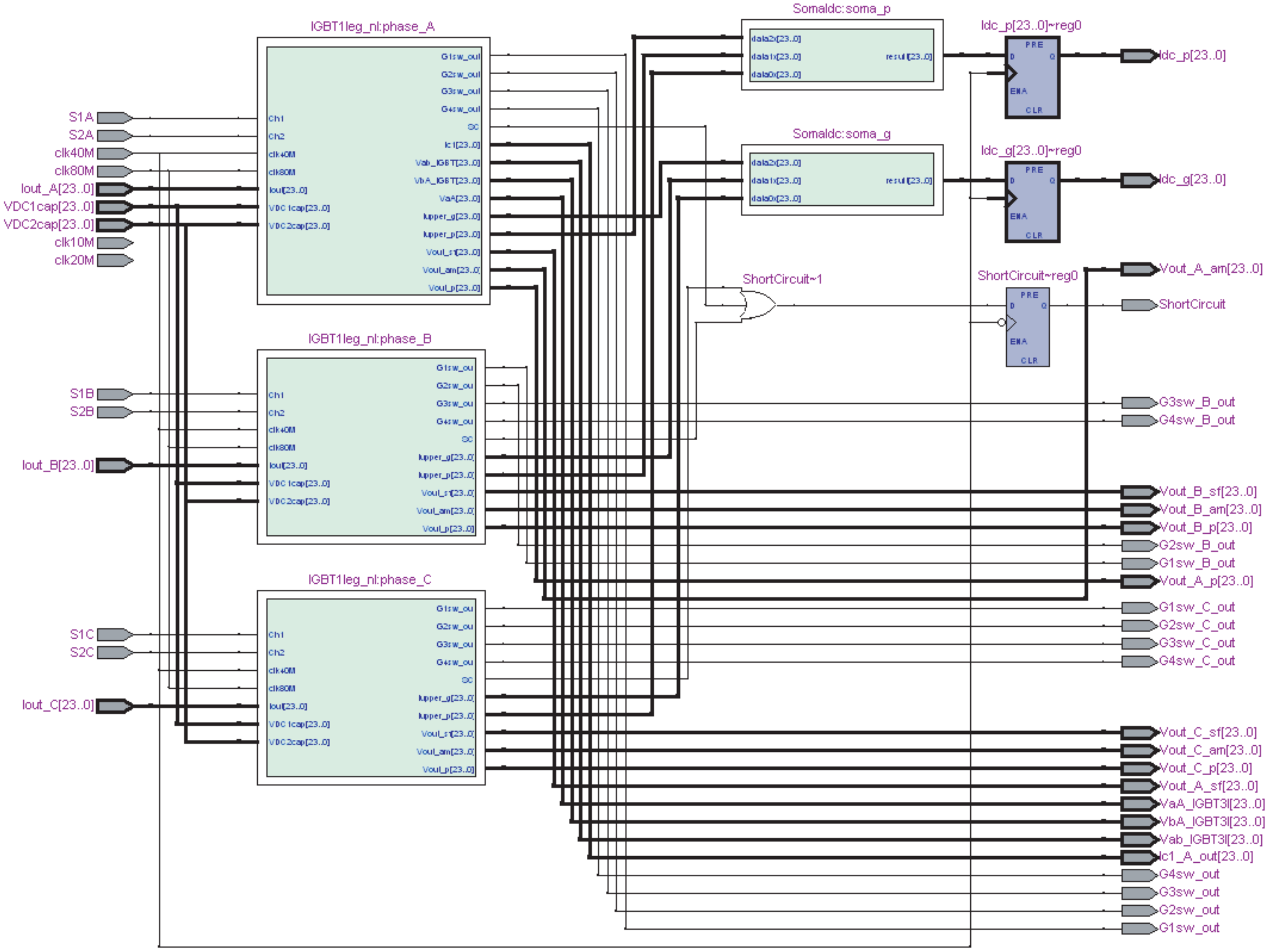


Figure B.2: VSC.gdf.

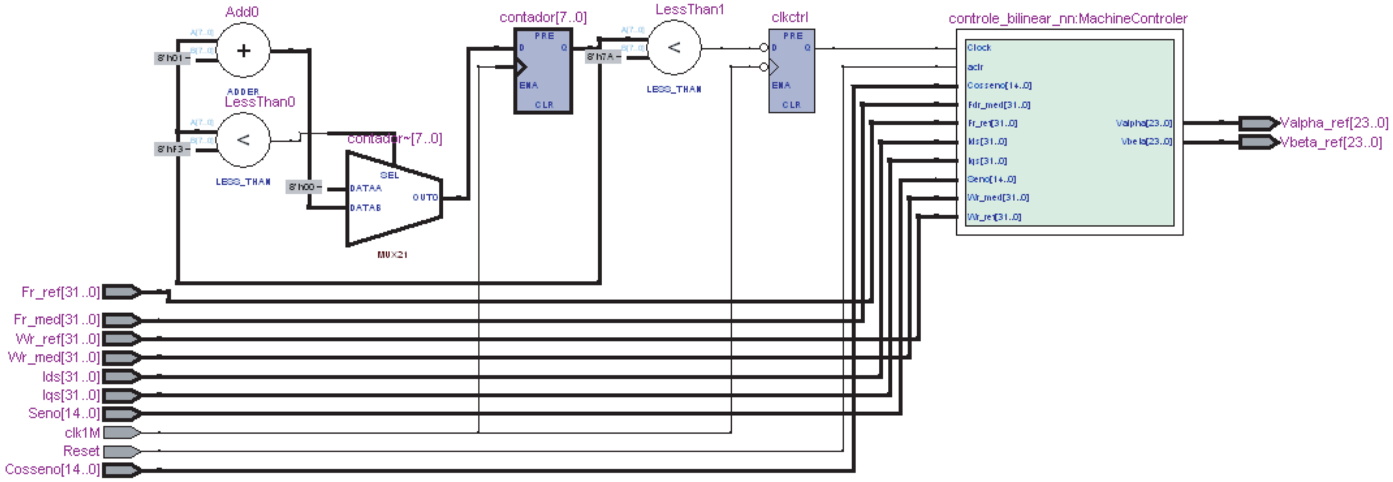


Figure B.3: control.gdf.

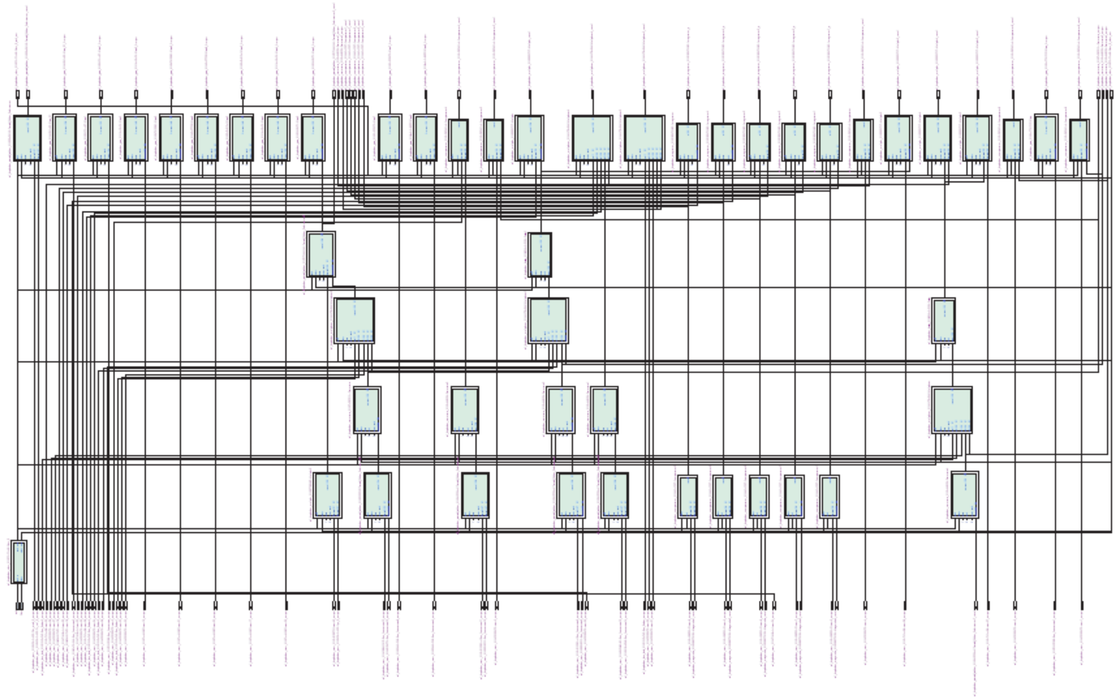


Figure B.4: control1.gdf.

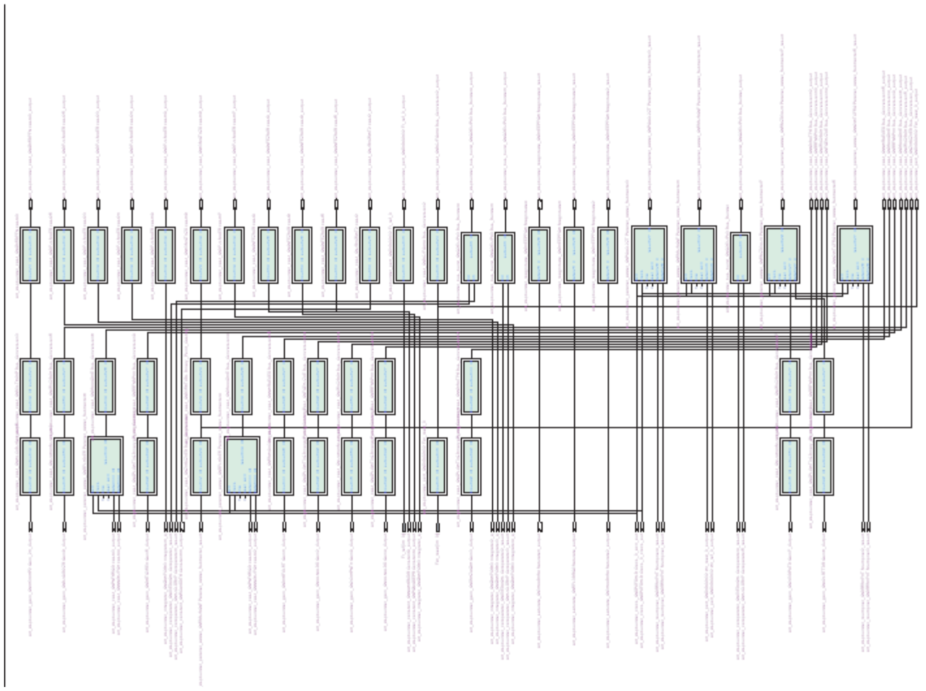


Figure B.5: control2.gdf.

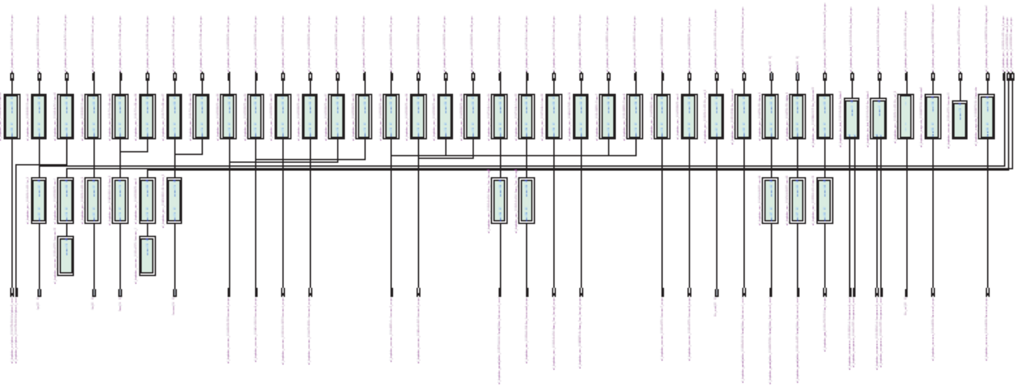


Figure B.6: control3.gdf.

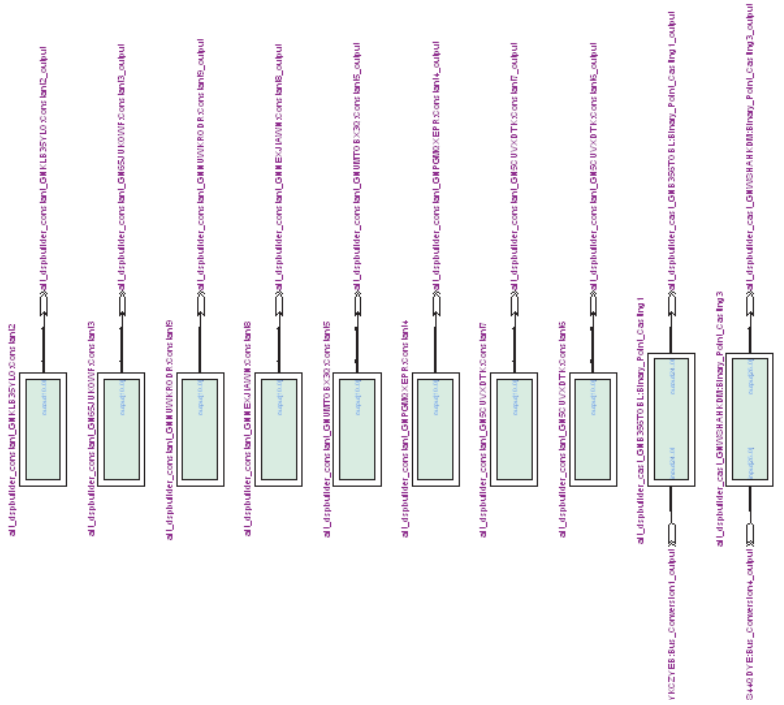

alt_dspbuilder_constant_GNKLB35YLO:Constant2
alt_dspbuilder_constant_GNKLB35YLO:Constant2_output
output[0:0]
alt_dspbuilder_constant_GN65JUKOWF:Constant3
alt_dspbuilder_constant_GN65JUKOWF:Constant3_output
output[0:0]
alt_dspbuilder_constant_GNNUWKRODR:Constant9
alt_dspbuilder_constant_GNNUWKRODR:Constant9_output
output[0:0]
alt_dspbuilder_constant_GNNEXJAWN:Constant8
alt_dspbuilder_constant_GNNEXJAWN:Constant8_output
output[0:0]
alt_dspbuilder_constant_GNUMTOBX32:Constant5
alt_dspbuilder_constant_GNUMTOBX32:Constant5_output
output[0:0]
alt_dspbuilder_constant_GNPGMM2XEPR:Constant4
alt_dspbuilder_constant_GNPGMM2XEPR:Constant4_output
output[0:0]
alt_dspbuilder_constant_GN5CUVXDTK:Constant7
alt_dspbuilder_constant_GN5CUVXDTK:Constant7_output
output[0:0]
alt_dspbuilder_constant_GN5CUVXDTK:Constant6
alt_dspbuilder_constant_GN5CUVXDTK:Constant6_output
output[0:0]
alt_dspbuilder_cast_GNB356TOBL:Binary_Point_Casting1
alt_dspbuilder_cast_GNB356TOBL:Binary_Point_Casting1_output
YKCZYEB:Bus_Conversion1_output
input[24:0]
output[24:0]
alt_dspbuilder_cast_GNWSHAHKDM:Binary_Point_Casting3
alt_dspbuilder_cast_GNWSHAHKDM:Binary_Point_Casting3_output
S4+2DYE:Bus_Conversion4_output
input[26:0]
output[26:0]

Figure B.7: control4.gdf.

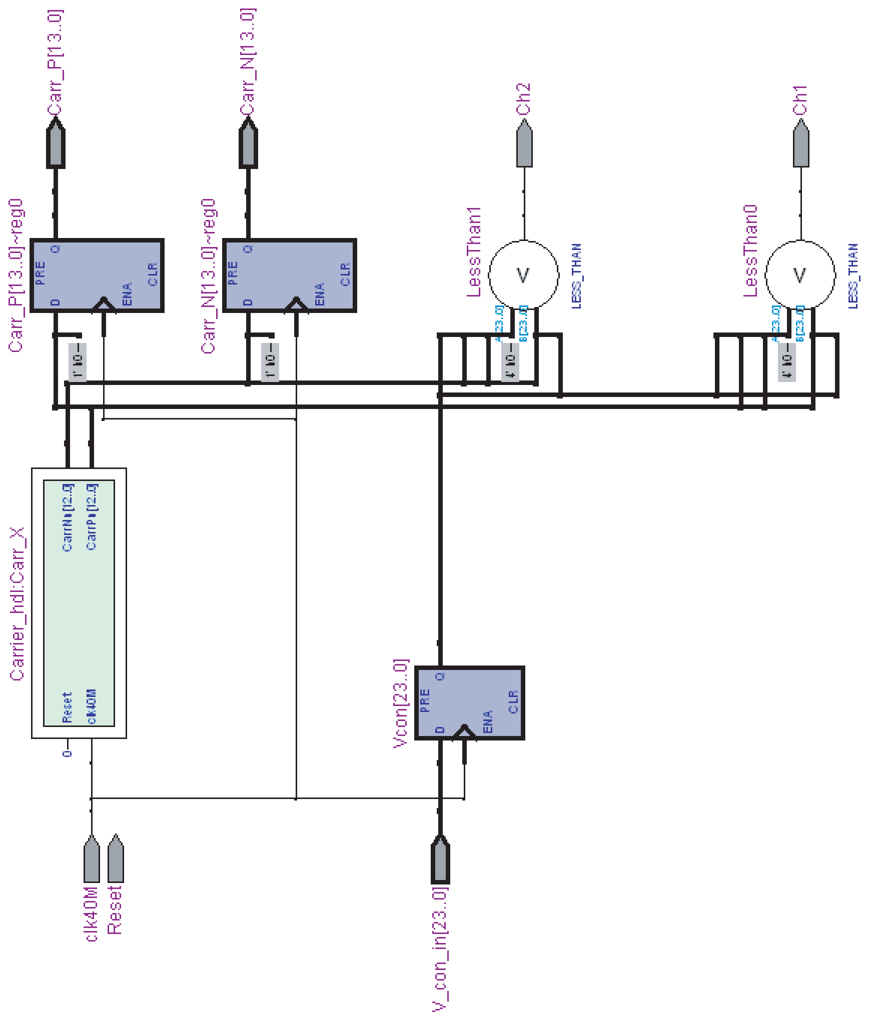


Figure B.8: cpwm.gdf.

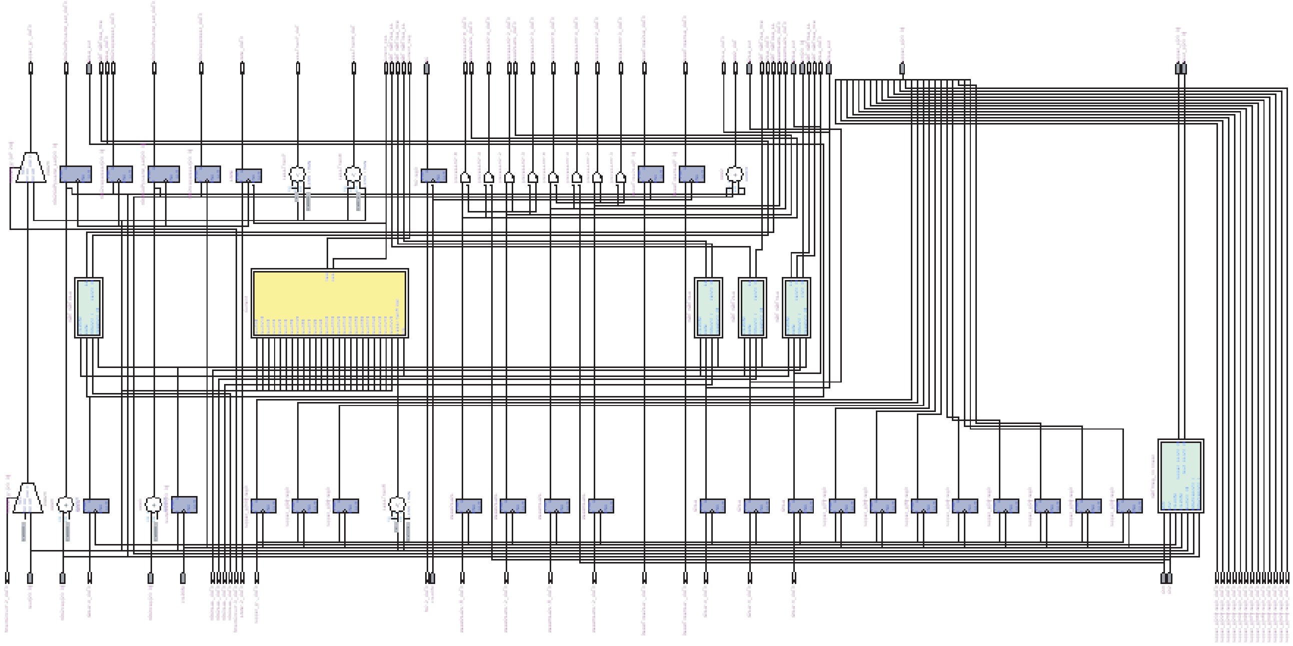


Figure B.9: igbt1nl.gdf.

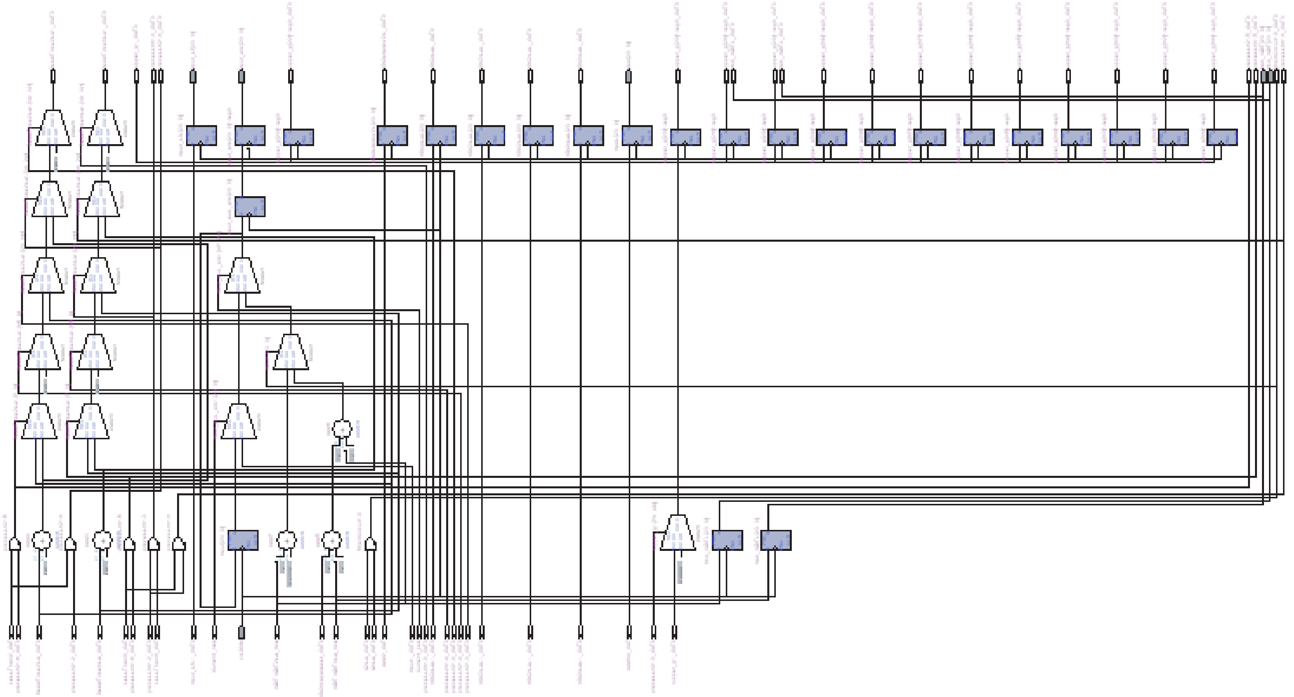


Figure B.10: igbt2.gdf.

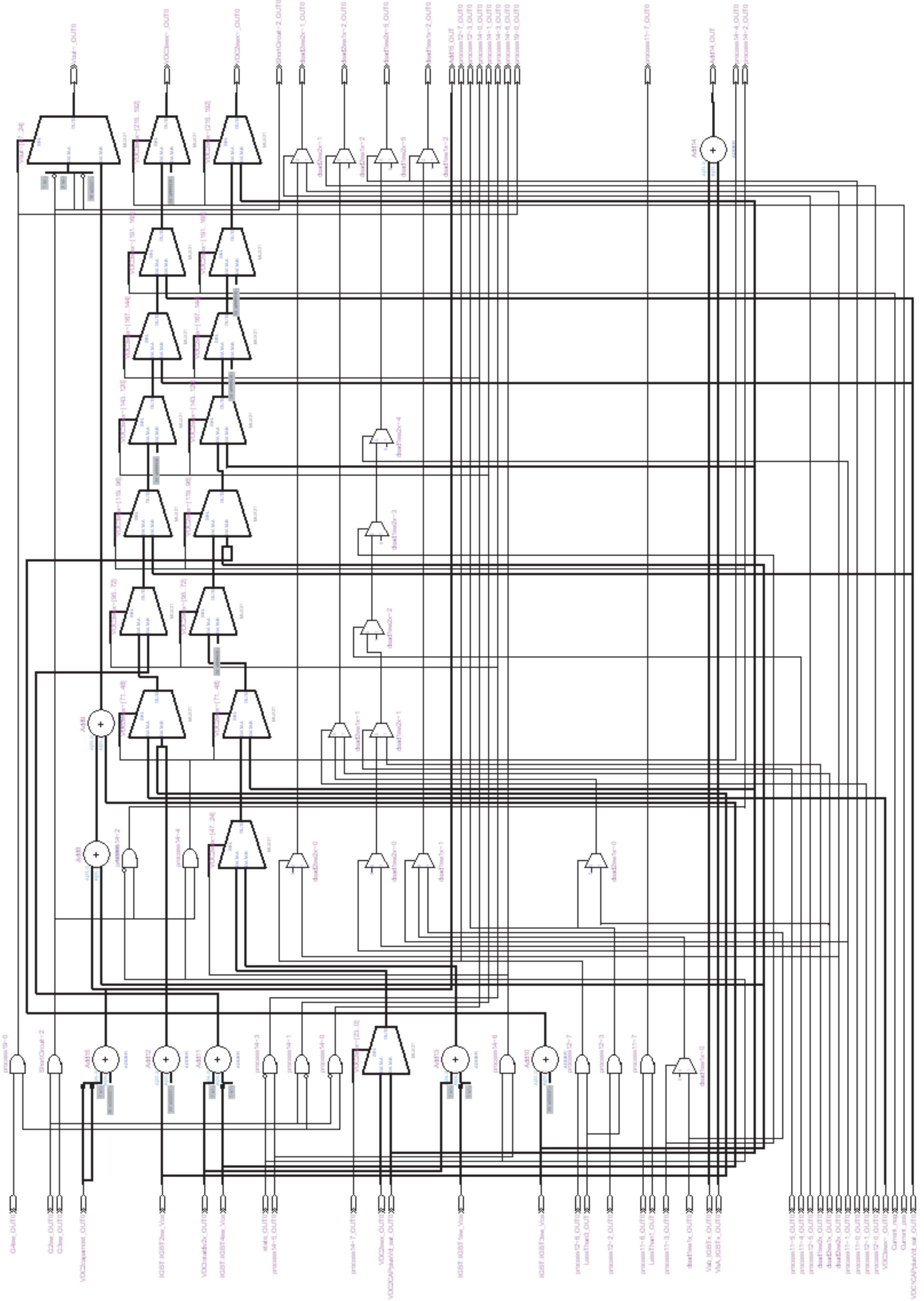


Figure B.11: igbt3.gdf.

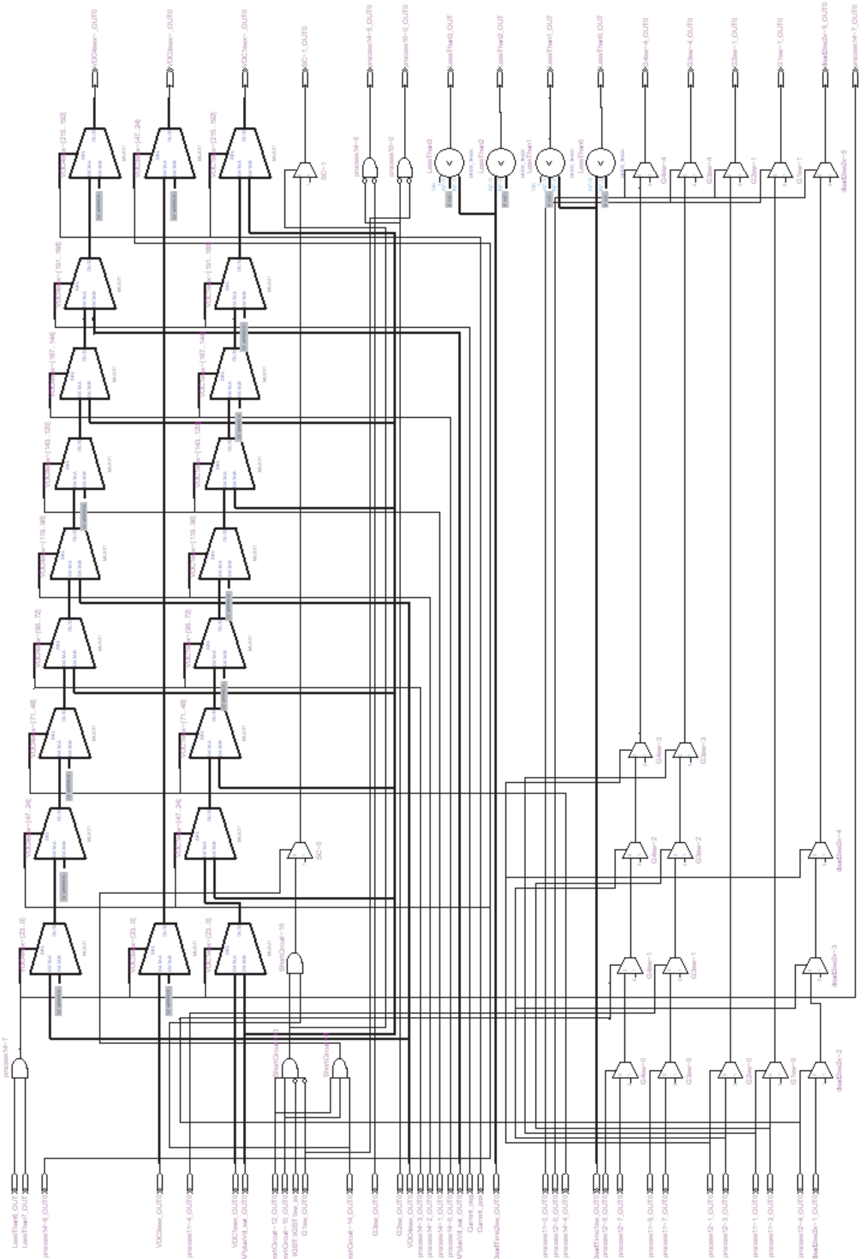


Figure B.12: igbt4.gdf.

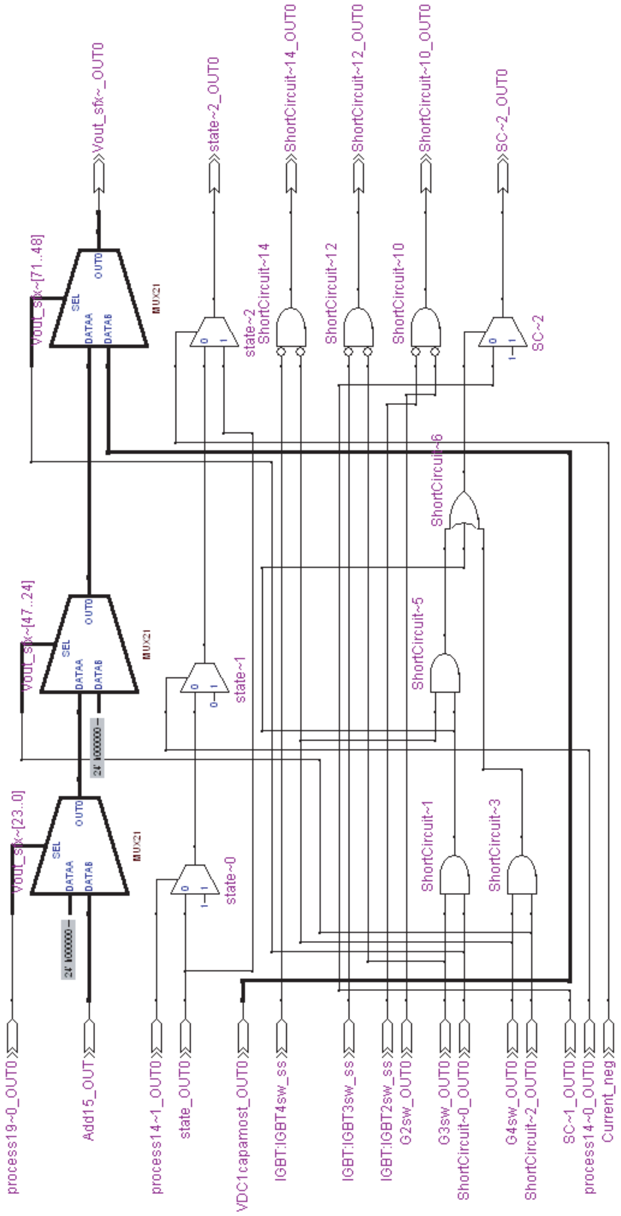


Figure B.13: igbt5.gdf.

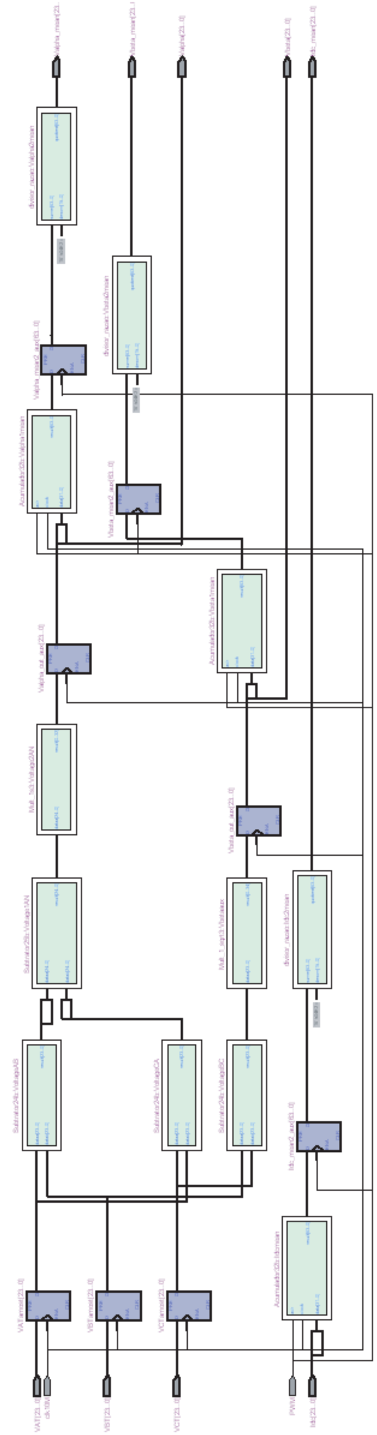


Figure B.14: measure.gdf.

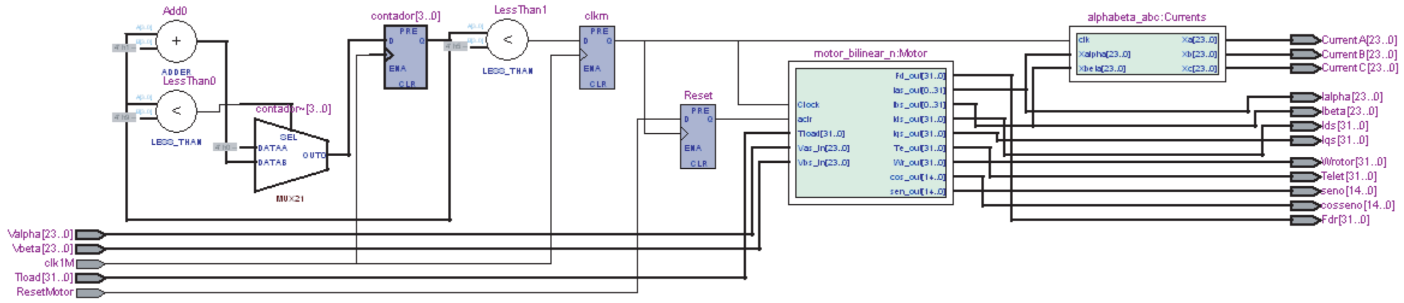


Figure B.15: motor.gdf.

C
FPGA Design Files II

C.1 Motor and PWM

- *motor.gdf* : motor file.
- *pwm1.gdf* : FOC control file 1.
- *pwm2.gdf* : FOC control file 2.
- *IGBT4.gdf* : FOC control file 3.
- *IGBT5.gdf* : FOC control file 4.
- *measure.gdf* : FOC control file 5.

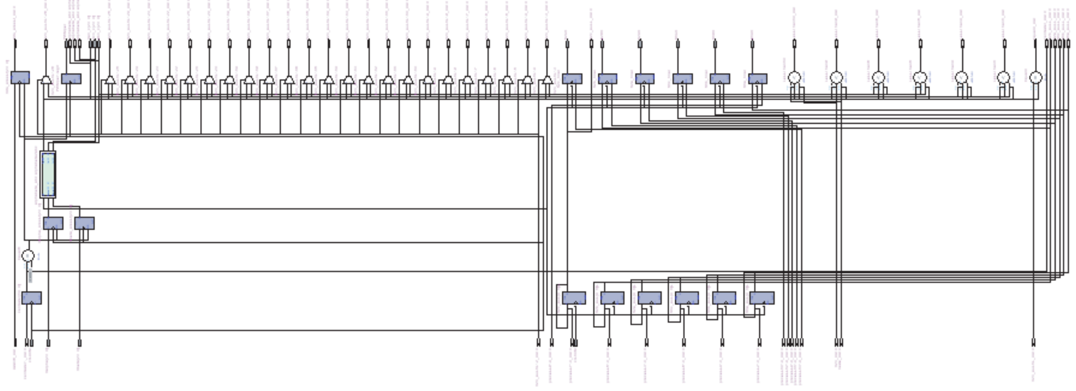


Figure C.1: pwm1.gdf.

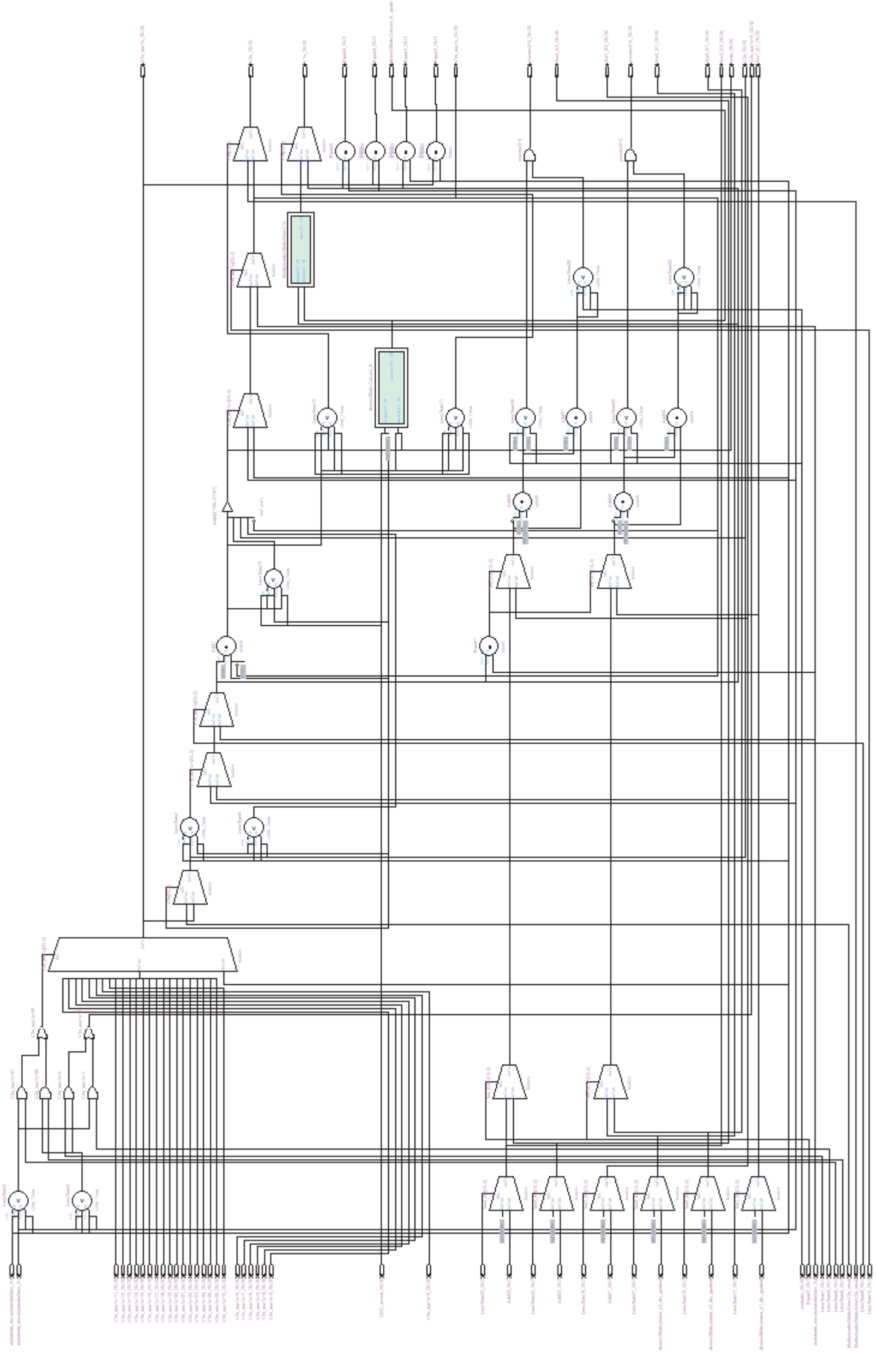


Figure C.2: pwm2.gdf.

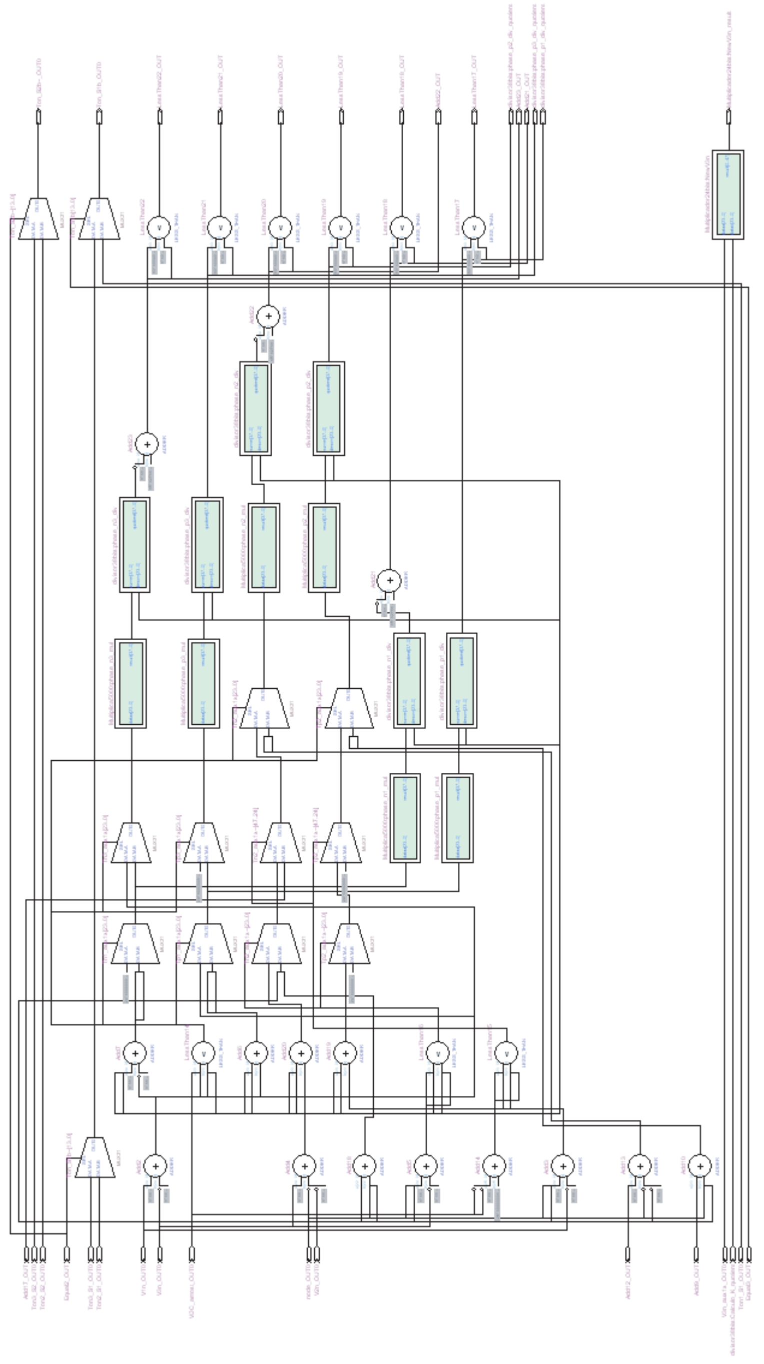


Figure C.3: pwm3.gdf.

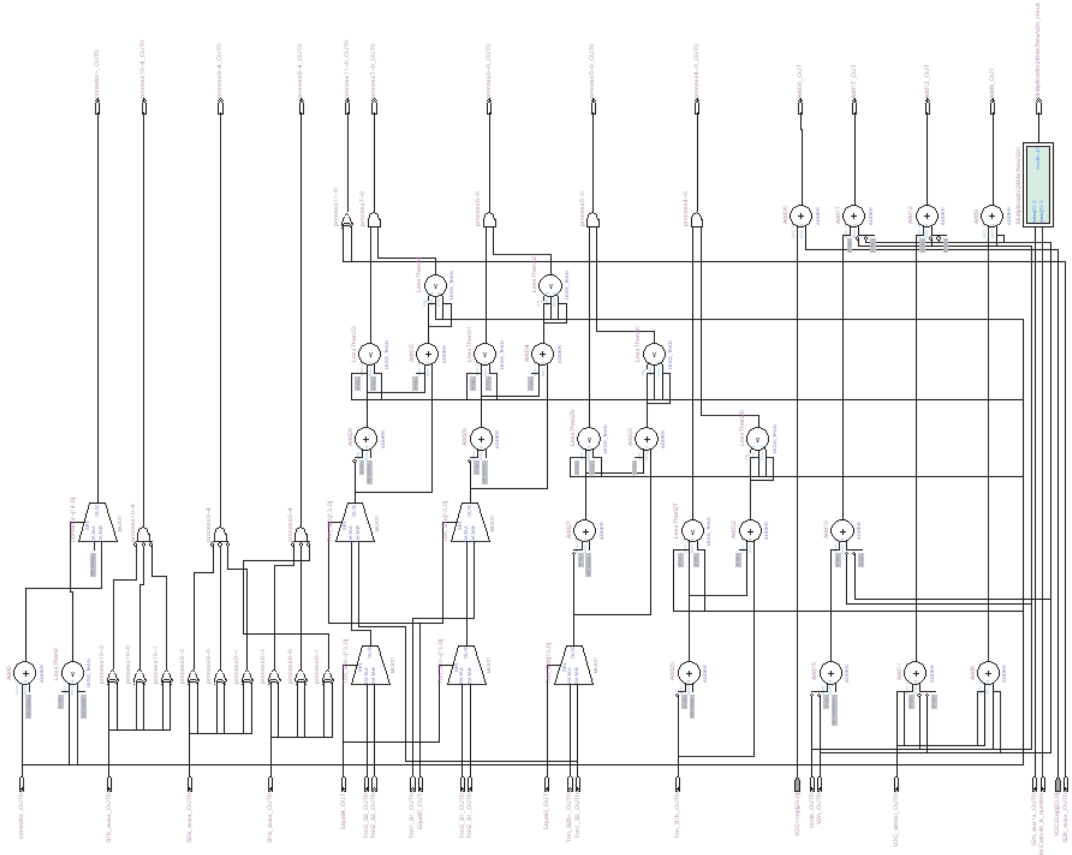


Figure C.4: pwm4.gdf.

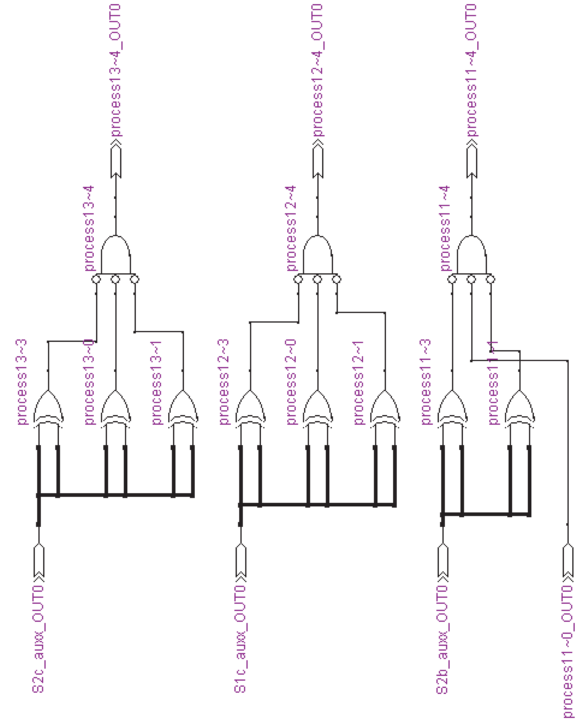


Figure C.5: pwm5.gdf.

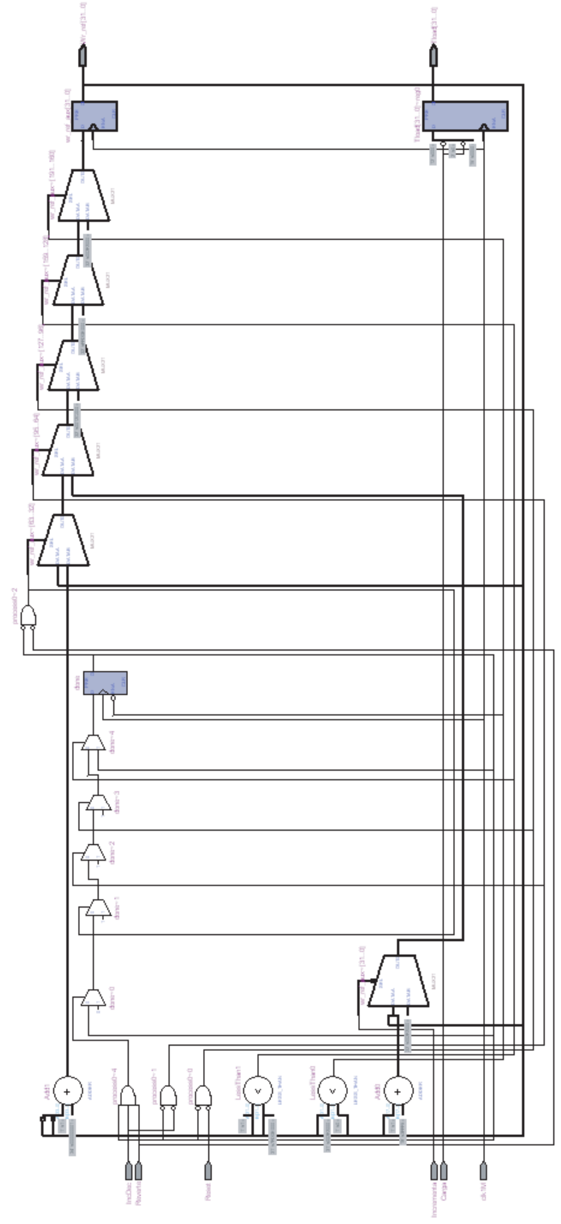


Figure C.6: ref.gdf.

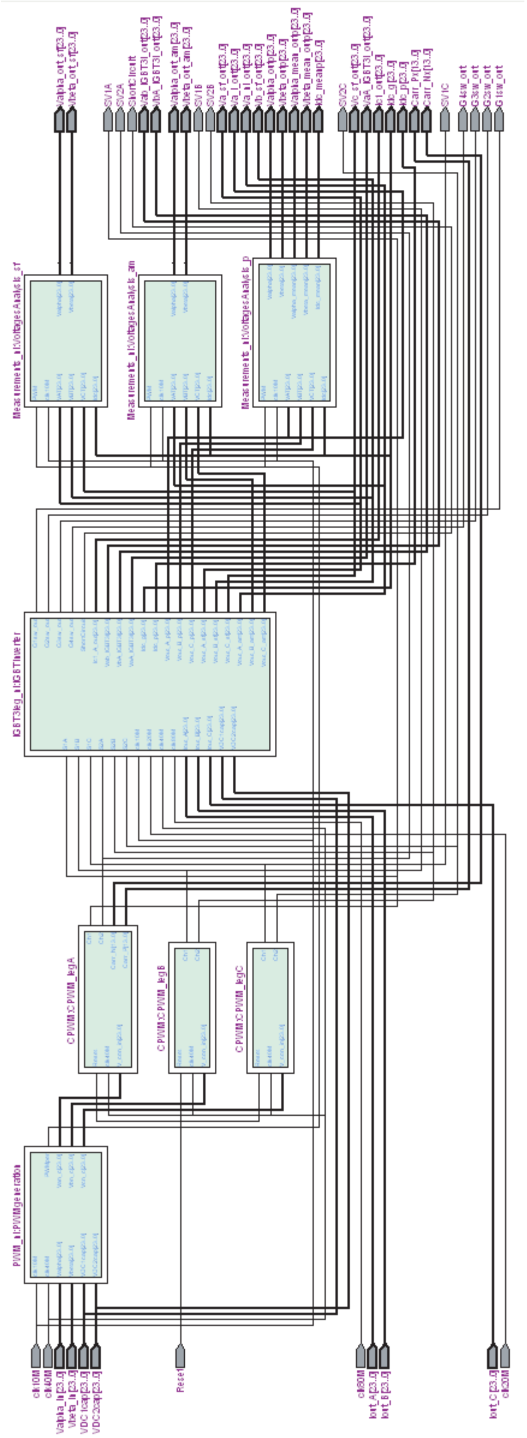


Figure C.7: VSC1.gdf.

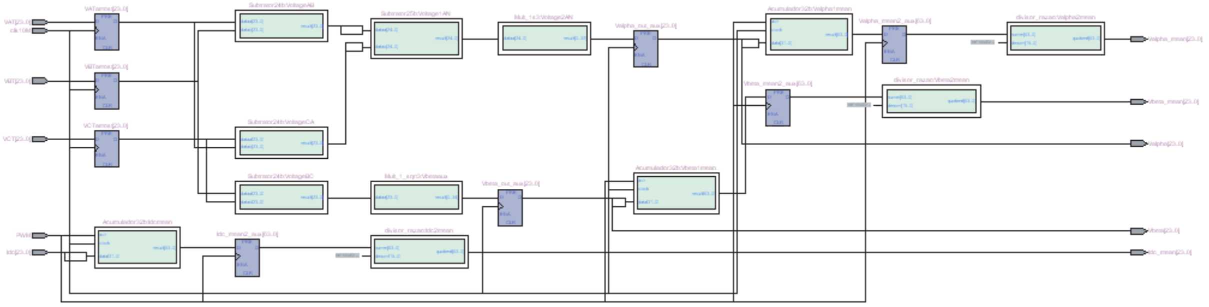


Figure C.8: Voltage Analysis.gdf.

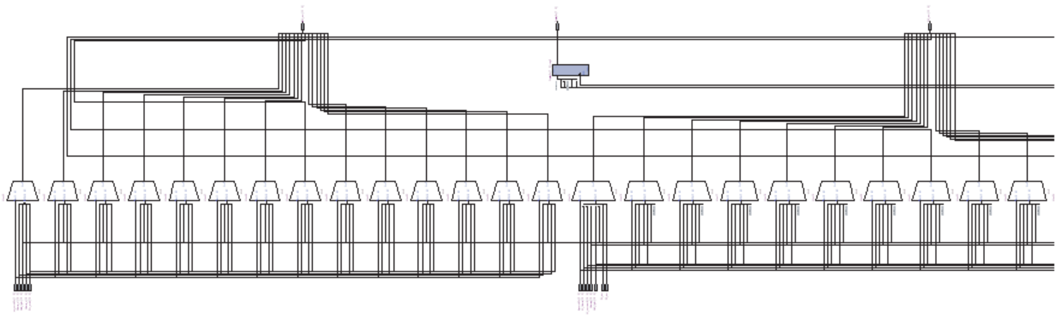


Figure C.9: Interface1.gdf.

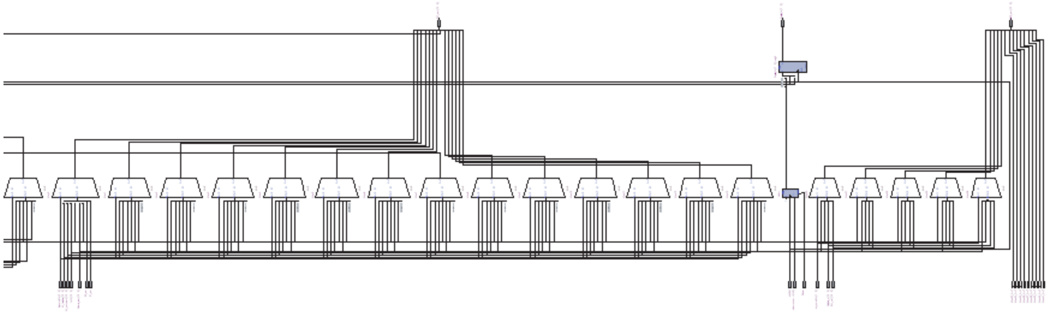


Figure C.10: Interface1a.gdf.

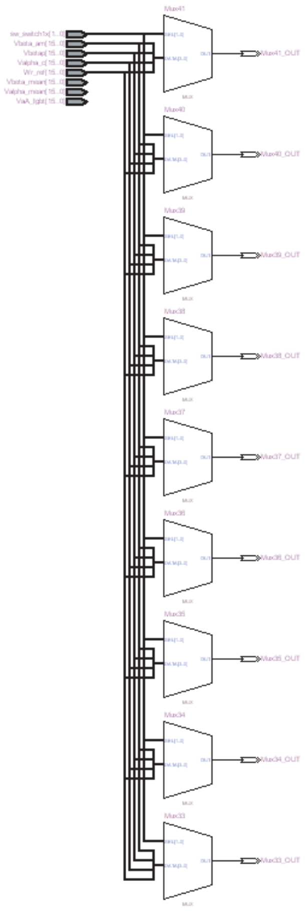


Figure C.11: Interface2.gdf.