

Real-Time Nonlinear Frequency-Dependent Electromagnetic Transient Power  
Transformer Model Hardware Emulation

by

Jia Dai Liu

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy  
in  
Energy Systems

Department of Electrical and Computer Engineering  
University of Alberta

©Jia Dai Liu, 2016

# Abstract

A transformer is a vital piece of transmission and distribution infrastructure in a power system which is used to transfer energy from one circuit to another. Real-time electromagnetic transient (EMT) simulation plays an important role in the design and testing of protection and control systems before they are commissioned in the field. An accurate power transformer model is paramount for accurately representing the host power system in real-time EMT simulation. However, due to excessive computational burden, transformer modeling has hitherto been limited to low-order, piece-wise linear, and low-frequency behavior.

On one hand the real-time EMT simulation needs to finish all the calculations in a certain small time-step, and on the another hand the components need to be modeled with adequate complexity to reproduce highly accurate simulation results. The field programmable gate arrays (FPGA) provides a hardware-rich environment to implement power system component models in real-time with its paralleled architecture and reconfigurable logic resource. This thesis develops detailed nonlinear power transformer models in real-time which can accurately emulate realistic transformer behaviour under various operating conditions, such as saturation, ferroresonance, hysteresis, and frequency-dependent eddy-currents. To investigate the detailed flux distribution in the transformer core, a high resolution magnetic equivalent circuit is proposed.

The models are fully parallelized and pipelined to achieve the lowest latency and smallest hardware resource consumption. Several power system test case studies are used to study the proposed FPGA-based real-time transformer models under various transient and power flow control conditions. The real-time results are fully validated using off-line EMT and finite-element modeling tools.

# Preface

As described in the following, several chapters in this thesis have been published or submitted as journal articles. This thesis is an original work by Jiadai Liu. My supervisor, Dr. Venkata Dinavahi has provided me with instructive comments and corrections during the research and the manuscript composition.

Chapter 3 of this thesis has been published as: J. Liu and V. Dinavahi, "A real-time nonlinear hysteretic power transformer transient model on FPGA", *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3587-3597, Jul. 2014.

Chapter 4 of this thesis has been published as: J. Liu and V. Dinavahi, "Detailed magnetic equivalent circuit based real-time nonlinear power transformer model on FPGA for electromagnetic transient studies", *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1191-1202, Feb. 2016.

Chapter 5 of this thesis has been published as: J. Liu and V. Dinavahi, "Nonlinear magnetic equivalent circuit based real-time transformer electromagnetic transient model on FPGA for HIL emulation", *IEEE Transactions on Power Delivery*, pp. 1-10, Jan. 2016.

To my parents,  
for their love, endless support,  
and encouragement.

# Acknowledgements

I would like to express my sincere thanks to my supervisor *Dr. Venkata Dinavahi* for his full support, encouragement, and guidance for my research throughout the years I have spent at the University of Alberta. His insightful guidance, passion and enthusiasm for the research has been an invaluable motivation for my life.

It is an honor for me to extend my gratitude to my Ph.D. committee members: *Dr. John Salmon, Dr. Yasser Mohamed, Dr. Greg Kish, Dr. Rama Gokaraju (University of Saskatchewan)* and *Dr. Venkata Dinavahi*, for reviewing my thesis and providing invaluable comments. Special thanks go to my colleagues and friends at the RTX-Lab: *Nariman Roshandel Tavana, Yuan Chen, Zhiyin Zhou, Wentao Wang, Bo Shi, Shenhao Yan, Yifan Wang* and *Zhuoxuan Shen*.

I wish to extend my deepest appreciation to my parents. Without their understanding, support, encouragement, and happiness, this thesis could not be finished.

Finally, financial help from NSERC and the University of Alberta for my living in Edmonton during these years is greatly appreciated.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Real-time Electromagnetic Transient Power System Simulation . . . . .	2
1.2	Survey of Transformer EMT Models . . . . .	3
1.2.1	Transformer Models . . . . .	3
1.2.2	Existing Off-line and Real-time EMT Transformer Models . . . . .	5
1.3	Motivation for This Thesis . . . . .	6
1.4	Research Objectives . . . . .	7
1.5	Thesis Outline . . . . .	8
<b>2</b>	<b>FPGA Background</b>	<b>11</b>
2.1	FPGA Architecture . . . . .	12
2.1.1	Configurable Logic Blocks (CLBs) . . . . .	12
2.1.2	Block RAMs . . . . .	14
2.1.3	Digital Signal Processing Blocks (DSPs) . . . . .	14
2.2	FPGA Design Tools and Design Flow . . . . .	15
2.3	FPGA Programming Technology . . . . .	18
2.3.1	Data Representation . . . . .	18
2.3.2	Parallel Processing . . . . .	19
2.3.3	Pipelining . . . . .	20
2.4	Summary . . . . .	21
<b>3</b>	<b>Admittance-Based Real-Time Transformer Model</b>	<b>23</b>
3.1	Admittance-based Linear EMT Power Transformer Model . . . . .	24
3.1.1	Linear Module Formulation . . . . .	24
3.1.2	Linear Module Hardware Design . . . . .	26
3.2	Nonlinear Model . . . . .	28
3.2.1	Compensation Method and Newton-Raphson Iterations . . . . .	29
3.2.2	Preisach Hysteresis Model . . . . .	32
3.2.3	Nonlinear Module Hardware Design . . . . .	33
3.3	Frequency-dependent Eddy Current Model . . . . .	34
3.4	The Overall Transformer Model . . . . .	36
3.5	Bergeron Transmission Line Model . . . . .	36

3.5.1	Bergeron Model Formulation . . . . .	36
3.5.2	Real-time FPGA Hardware Implementation of Bergeron Model . . . . .	38
3.5.2.1	Transformation Unit . . . . .	40
3.5.2.2	Update Unit . . . . .	40
3.6	Source Module . . . . .	40
3.7	Passive Elements Module . . . . .	41
3.7.1	Model Formulation . . . . .	41
3.7.1.1	Resistance Element $R$ . . . . .	42
3.7.1.2	Inductance Element $L$ . . . . .	42
3.7.1.3	Capacitance Element $C$ . . . . .	43
3.7.2	Real-time Hardware Module for Passive Elements Module . . . . .	44
3.8	Inode Unit Module . . . . .	45
3.9	Linear Network Solver Module . . . . .	46
3.10	Digital Hardware Emulation of The Power Transformer . . . . .	48
3.11	The Real-time Nonlinear Transformer Model Setup on FPGA . . . . .	50
3.12	Case Study I . . . . .	52
3.12.1	Inrush Current Transients . . . . .	52
3.12.2	Hysteresis Behaviour . . . . .	53
3.12.3	Ferroresonance Transients . . . . .	55
3.12.4	Eddy Current Transient . . . . .	56
3.13	Case Study II . . . . .	56
3.13.1	Capacitor Switching Transients . . . . .	56
3.13.2	Ground Fault Transients . . . . .	56
3.14	Summary . . . . .	60
<b>4</b>	<b>Detailed Magnetic Equivalent Circuit Based Real-Time Transformer Model</b>	<b>61</b>
4.1	Detailed Hardware Emulation of The Power Transformer . . . . .	62
4.1.1	High Resolution Magnetic Equivalent Circuit Based Transformer Model	62
4.1.1.1	The HR-MEC representation . . . . .	62
4.1.1.2	The HR-MEC formulation . . . . .	64
4.1.1.3	HR-MEC model discretization . . . . .	66
4.1.1.4	HR-MEC model hardware design . . . . .	68
4.1.2	Preisach Hysteretic Unit (PHU) . . . . .	70
4.2	Power System Network Transient Simulation . . . . .	73
4.2.1	Interfacing of the HR-MEC Based Transformer Model to the Power System Network . . . . .	73
4.2.2	LU Decomposition Based Network Solver Module . . . . .	74
4.2.2.1	LU decomposition formula . . . . .	74
4.2.2.2	LUD unit hardware implementation . . . . .	76
4.2.3	LU-based Inverse Matrix Calculator . . . . .	76

4.2.3.1	Inverse matrix calculator formula . . . . .	76
4.2.3.2	Inverse matrix calculator hardware implementation . . . . .	78
4.3	Case Study I . . . . .	79
4.3.1	Validation by Finite Element Simulation . . . . .	79
4.3.2	Energization Transients and Steady-state Waveforms . . . . .	82
4.4	Case Study II . . . . .	86
4.4.1	Fault Transients . . . . .	86
4.5	Summary . . . . .	88
<b>5</b>	<b>Magnetic Equivalent Circuit Based Real-time Sen Transformer Model for HIL Emulation</b>	<b>89</b>
5.1	Introduction . . . . .	90
5.2	Geometrical Sen Transformer EMT Model . . . . .	91
5.2.1	Sen Transformer Operating Principle . . . . .	91
5.2.2	Tap-selection Algorithm . . . . .	92
5.2.3	High-fidelity Nonlinear MEC Based ST Model . . . . .	92
5.3	High-fidelity Nonlinear MEC-based ST Hardware Emulation . . . . .	96
5.3.1	Network Transient Emulation with Embedded ST . . . . .	96
5.4	Case Studies And Real-time Emulation Results . . . . .	98
5.4.1	Finite Element Modeling and Validation . . . . .	98
5.4.2	Case Studies . . . . .	99
5.4.2.1	Energization transient and steady-state emulation . . . . .	100
5.4.2.2	Hysteresis transient emulation . . . . .	104
5.4.2.3	Power flow regulation . . . . .	104
5.4.2.4	Power flow control emulation . . . . .	107
5.4.2.5	Fault transient emulation . . . . .	109
5.5	Summary . . . . .	111
<b>6</b>	<b>Conclusions and Future Works</b>	<b>112</b>
6.1	Contributions of This Thesis . . . . .	112
6.2	Directions for Future Work . . . . .	114
	<b>Bibliography</b>	<b>116</b>
	<b>Appendix A Power System Data of Case Study in Chapter 3</b>	<b>123</b>
A.1	Case Study I . . . . .	123
A.2	Case Study II . . . . .	124
A.3	Parameters for Hysteresis . . . . .	124



<b>Appendix B Parameters for Case Study in Chapter 4</b>	<b>125</b>
B.1 Case Study for Energization Transient . . . . .	125
B.2 Case Study for Fault Transient . . . . .	126
B.3 Geometry Parameters of Transformer . . . . .	127
B.4 Node-branch Connection Matrix A . . . . .	128
<b>Appendix C Parameters for Case Study in Chapter 5</b>	<b>129</b>
C.1 Geometry Parameters of Transformer . . . . .	129
C.2 Magnetic Equivalent Circuit Parameters . . . . .	130
C.3 Power System Parameters for Case Study . . . . .	131
C.4 Tap Changer Setting . . . . .	132

## List of Tables

2.1	Module level hardware resource utilization . . . . .	13
3.1	Parameters for updating RLC elements history currents (3.72) . . . . .	45
4.1	Module level hardware resource utilization . . . . .	82
4.2	Module level operator implementation . . . . .	82
5.1	Module level hardware resource utilization . . . . .	102
A.1	Parameters of the case study I, inrush current transient . . . . .	123
A.2	Parameters of the case study I, ferroresonance and eddy current transients . . . . .	123
A.3	Parameters of the case study II . . . . .	124
A.4	Parameters for hysteresis major loop trajectory . . . . .	124
B.1	Parameters for energization transient case study . . . . .	125
B.2	Parameters for fault transient case study . . . . .	126
B.3	Parameters of transformer geometry . . . . .	127
C.1	Parameters of transformer geometry . . . . .	129
C.2	Parameters for magnetic equivalent circuit . . . . .	130
C.3	Parameters for hysteresis major loop trajectory . . . . .	130
C.4	Parameters for energization transient case study . . . . .	131
C.5	Parameters for hysteresis transient case study . . . . .	131
C.6	Parameters for power flow control and fault transient case study . . . . .	131

# List of Figures

1.1	Hardware-in-the-loop configuration. . . . .	3
1.2	STC model of single-phase N-winding transformers. . . . .	4
2.1	Xilinx Virtex-7 <sup>®</sup> FPGA architecture block diagram [50]. . . . .	13
2.2	CLB architecture and the connection to switch matrix [50]. . . . .	14
2.3	Simplified slice architecture [50]. . . . .	15
2.4	Xilinx Virtex-7 <sup>®</sup> FPGA block RAM (a) true dual-port RAM; (b) simple dual-port RAM [51]. . . . .	16
2.5	Xilinx Virtex-7 <sup>®</sup> FPGA DSP48E1 slice [52]. . . . .	16
2.6	General FPGA design flow. . . . .	18
2.7	Single precision floating-point representation. . . . .	18
2.8	Data flow for (a) sequential processing; (b) parallel processing. . . . .	19
2.9	(a) Pipeline data path design and data flow for (b) pipeline processing; (c) traditional processing. . . . .	21
3.1	An example matrix and its storage format. . . . .	26
3.2	Pipelined scheme for the linear module: (a) hardware design; (b) timing diagram. . . . .	27
3.3	MVM unit: (a) hardware design; (b) timing diagram. . . . .	28
3.4	Hysteresis characteristic with major and minor loops. . . . .	29
3.5	(a) Linear network with external nonlinear elements; (b) Compensation method demonstration for multiple nonlinearities. . . . .	30
3.6	Corresponding reversal point stack before (a) and after (b) current increasing beyond the point 6. . . . .	33
3.7	Preisach hysteretic unit (PHU) hardware design. . . . .	34
3.8	Overall nonlinear module hardware architecture. . . . .	35
3.9	Cauer equivalent circuit for frequency dependent eddy current model. . . . .	35
3.10	The overall frequency-dependent nonlinear hysteretic transformer model. . . . .	36
3.11	Discrete-time Norton equivalent circuit of the Bergeron line model. . . . .	37
3.12	<b>B-TLM</b> module hardware design. . . . .	39
3.13	Paralleled <b>Transformation Unit</b> and a <b>Update Unit</b> in the <b>B-TLM</b> . . . . .	39
3.14	Pipelined computation scheme in the <b>Transformation Unit</b> . . . . .	40

3.15	Pipelined computation scheme in the <b>Update Unit</b> . . . . .	41
3.16	Hardware implementation of <b>Source</b> module. . . . .	41
3.17	A resistance $R$ element and its discrete-time model. . . . .	42
3.18	(a) An inductance $L$ element; (b) its discrete-time Norton equivalent circuit. . . . .	43
3.19	(a) A capacitance $C$ element; (b) its discrete-time Norton equivalent circuit. . . . .	44
3.20	Hardware implementation for <b>RLC</b> module. . . . .	45
3.21	<b>Inode</b> unit: (a) an example circuit; (b) corresponding RAM assignment; (c) timing diagram; (b) $rmInode$ vector. . . . .	46
3.22	<b>Inode</b> unit: hardware design. . . . .	47
3.23	(a) Pipelined calculation scheme for $i_b$ ; (b) for $v_A$ . . . . .	48
3.24	(a) An example sparse matrix and its storage format; (b) its corresponding timing diagram. . . . .	49
3.25	Overall architecture of the real-time nonlinear hysteretic transformer model on FPGA. . . . .	50
3.26	Finite state machine of the main control module. . . . .	50
3.27	Experimental setup . . . . .	51
3.28	Model latencies for one time-step. . . . .	52
3.29	Single-line diagram for case study I. . . . .	52
3.30	Breaker current when breaker close at $t = 0s$ (a) real-time simulation results (1 div $x = 20$ ms; 1 div $y = 320$ A); (b) off-line (ATP) simulation results. . . . .	53
3.31	Real-time simulation results for the flux, magnetizing current, and the hysteresis characteristic of the nonlinear transformer. . . . .	54
3.32	Primary side three-phase voltages when breaker opened at $t = 0.2s$ (a) real-time simulation results (1 div $x = 20$ ms; 1 div $y = 100$ kV); (b) off-line (ATP) simulation results. . . . .	55
3.33	TRV real-time simulation results at $t = 0.2s$ (a) Cauer model; (b) non frequency-dependent model (constant resistor) (1 div $1x = 4$ ms; 1 div $1y = 84$ kV; 1 div $2x = 400 \mu s$ ; 1 div $2y = 84$ kV). . . . .	57
3.34	Single-line diagram for case study. . . . .	58
3.35	Capacitor switch at $t = 1 s$ (a) real-time simulation results (1 div $x = 10$ ms; 1 div $y = 1.06$ kV); (b) off-line (ATP) simulation results. . . . .	58
3.36	Ground fault at $t = 1s$ (a) real-time simulation results (1 div $x = 20$ ms; 1 div $y = 1.5$ kV); (b) off-line (ATP) simulation results. . . . .	59
4.1	Major flux paths in a three-phase three-limb transformer. . . . .	63
4.2	HR-MEC model for a three-phase three-limb transformer. . . . .	64
4.3	Norton equivalent circuit of the HR-MEC model over one simulation time-step with a time-varying admittance matrix. . . . .	68
4.4	Sparse matrix storage format: (a) example matrices; (b) corresponding storage formats. . . . .	69

4.5	Pipelined design for the sparse-dense matrix multiplication <b>SMxDM</b> unit: (a) hardware design; and (b) timing diagram. . . . .	70
4.6	Pipelined design for the multiplication accumulator calculation <b>MAC</b> unit: (a) hardware design; (b) timing diagram. . . . .	71
4.7	Hardware design of the interface between the HR-MEC transformer model and the power system network for transient simulation. . . . .	72
4.8	Finite state machine of the supervisor module in Fig. 4.7. . . . .	74
4.9	LU decomposition hardware design. . . . .	76
4.10	A 5x5 example <b>LUD</b> hardware implementation and its data path. . . . .	77
4.11	The FSM for <b>LUD</b> unit. . . . .	78
4.12	Forward substitute hardware design. . . . .	78
4.13	Backward substitute hardware design. . . . .	79
4.14	The Inverse Matrix Calculator <b>IMC</b> hardware implementation. . . . .	79
4.15	The built JMAG transformer and its corresponding studied circuit. . . . .	80
4.16	Magnetic flux density distribution of the modeled transformer in JMAG <sup>®</sup> Designer at the peak of phase-c current. . . . .	81
4.17	Single-line diagram of the modeled power system for case study with HR-MEC based transformer model. . . . .	83
4.18	Latencies for one emulation time-step for the HR-MEC based transformer model emulation. . . . .	83
4.19	Real-time and JMAG <sup>®</sup> Designer simulation results: (a)-(c) Energization transient current; (d) steady-state current. . . . .	84
4.20	Real-time and JMAG <sup>®</sup> Designer simulation results: (a) frequency analysis of the energization transient current; (b) frequency analysis of the steady-state current. . . . .	85
4.21	Real-time and JMAG <sup>®</sup> Designer simulation results: (a)-(c) flux-linkage under transient steady; (d) steady-state flux-linkage. . . . .	87
4.22	Ground fault at $t_1 = 2s$ real-time simulation results (a) transient voltage waveform (1 divx1=20 ms; 1divy1= 20 kV); (b) transient current waveform (1 divx2=20 ms; 1divy2 = 39 A ). . . . .	88
5.1	Schematic of the ST and its interconnection to the transmission network. . .	91
5.2	High-fidelity magnetic equivalent circuit representation for the ST iron core.	93
5.3	High fidelity MEC-based ST model interfaced with cascaded equivalent circuit. . . . .	96
5.4	Pipelined and paralleled hardware design architecture during one time-step of emulation for the ST and the host power system. . . . .	97
5.5	Finite state machine of the high-fidelity nonlinear MEC based ST EMT model and power system. . . . .	99
5.6	FEM simulation: (a) coupled external electrical circuit; (b,c) 3D FEM model.	100

5.7	3D-FE simulation of the ST in JMAG <sup>®</sup> . . . . .	101
5.8	Single-line diagram of the power system with the ST model for the real-time HIL case study. . . . .	101
5.9	Latencies for one emulation time-step for the high-fidelity MEC based ST model. . . . .	102
5.10	Real-time emulation and JMAG <sup>®</sup> simulation results for the energization transient. . . . .	103
5.11	Real-time emulation and JMAG <sup>®</sup> simulation results for the ST steady-state current. . . . .	104
5.12	Real-time and JMAG <sup>®</sup> Designer simulation results: (a)-(c) flux-linkage under transient steady; (d) steady-state flux-linkage. . . . .	105
5.13	Real-time emulation results of (a) the magnetizing current (1 divx1 = 0.04 s; 1 divy1 = 0.025 p.u.); (b) flux in the core (1 divx2 = 0.04 s; 1 divy2 = 0.32 p.u.); (c) the hysteresis characteristic of ST (1 divx3 = 0.022 p.u.; 1 divy3 = 0.32 p.u.). . . . .	106
5.14	Relationship between $P_r$ and $Q_r$ at different injection voltage magnitude and phase angle for the ST. . . . .	107
5.15	Real-time emulation of active power and reactive power regulation on the two transmission lines (a) Line $L_1$ (1 divx1 = 0.8 s; 1 divy1 = 30 MW/MVAr); (b) Line $L_2$ (1 divx2 = 0.8 s; 1 divy2 = 43 MW/MVAr). . . . .	108
5.16	Real-time emulation of step-by-step transition response of (a) secondary winding current (1 divx1 = 0.8 s; 1 divy1 = 0.18 p.u.); (b) injected compensation voltage of ST (1 divx2 = 0.8 s; 1divy2 = 0.084 p.u.). . . . .	109
5.17	Real-time emulation results of ground fault transient (a) secondary side current (1 divx1 = 0.01 s; 1 divy1 = 0.95 p.u.); (b) secondary side voltage (1 divx2 = 0.01 s; 1 divy2 = 0.32 p.u.); (c) injected compensation voltage (1 divx3 = 0.01 s; 1divy3 = 0.067 p.u.). . . . .	110
B.1	Transformer geometry parameter. . . . .	127

## List of Acronyms

<b>ALM</b>	Adaptive Logic Module
<b>ALU</b>	Arithmetic Logic Unit
<b>ATP</b>	Alternative Transients Program
<b>CORDIC</b>	COordinate Rotation DIgital Computer
<b>CPU</b>	Central Processing Unit
<b>DAC</b>	Digital-to-Analog Converter
<b>DSP</b>	Digital Signal Processing (Processor)
<b>EMT</b>	ElectroMagnetic Transients
<b>EMTP</b>	ElectroMagnetic Transients Program
<b>FPGA</b>	Field-Programmable Gate Array
<b>FSM</b>	Finite State Machine
<b>GJE</b>	Gauss-Jordan Elimination
<b>GPU</b>	Graphics Processing Unit
<b>HIL</b>	Hardware-in-the-Loop
<b>IP</b>	Intellectual Property
<b>JTAG</b>	Joint Test Action Group
<b>LUT</b>	Look-up Table
<b>NR</b>	Newton-Raphson
<b>PNR</b>	Piecewise Newton-Raphson
<b>SIMD</b>	Single Instruction Multiple Data
<b>(S)RAM</b>	(Static) Random Access Memory
<b>SST</b>	Solid-State Transformer
<b>VHDL</b>	Very High-Speed Integrated Circuit Hardware Description Language
<b>VLSI</b>	Very Large-Scale Integrated Circuit

# 1

## Introduction

A power transformer is an important component in power systems for transmission and distribution of electrical energy. The complicated behaviour of a power transformer, such as core saturation, hysteresis, losses, frequency-dependency, capacitive coupling and the topological aspects of core and coil structure, need to be considered when a transformer model is developed for electromagnetic transient (EMT) studies [1–3]. These behaviours can have a great impact on the transient overvoltages and overcurrents, for example, phenomena such as inrush current, ferroresonance, and harmonics, which influence the design of protection and control systems. How to properly represent a power transformer is a very important question when a power system is simulated in real-time. However, due to the complex configuration and nature of the power transformer such as a large number of windings and nonlinear parameters, the modeling aspects of the transformer can be very complicated, especially taking real-time implementations issues into consideration.

The real-time digital simulator is an important tool in the analysis of power systems for design and development of new solutions to alleviate operational challenges of complex grids. Compared to off-line simulation, the real-time simulation is required to finish all the necessary calculations with a certain amount of time, which is called simulation time-step as they happen in real world. The real-time simulators are required to reproduce whatever happens in a real power system under the digital laboratory conditions, so that it can be used to emulate a real power system in design and test procedures. It is more convenient for the user to reconstruct and test a power system under extreme conditions using a real-time digital simulator [4–11].

An accurate and efficient real-time simulator, at the system-level is desired to accommodate large power system networks, while at the electrical component level it is desired to model the detailed model behavior. With the development of complex modern power



systems, the intensity of computation has increased extremely, so that powerful computational devices have become necessary. The Field Programmable Gate Arrays (FPGAs) are making inroads into real-time simulation mainly due to its parallel hardwired architecture which allows detailed modeling of power system components with small time-step to capture transients with high-fidelity. FPGA-based digital hardware models and algorithms of several power system apparatus such as transmission lines [12–15], rotating machines [16–19], and power electronic equipment [20–24], protective relays [25], large-scale power grids [26, 27] have appeared in the literature, however, the transformer model is one of the weakest components in real-time transient simulation tools due to its relative complex nature.

## **1.1 Real-time Electromagnetic Transient Power System Simulation**

Electromagnetic transient (EMT) can occur at any portion of the power system due to various reasons such as switching actions, short-circuit and lightning strike. The EMT may produce overvoltages and overcurrents and introduce large harmonic to the power system, which can damage component insulation and cause system interruption. Thus EMT simulation plays an important role in the planning and design of power system such as determining component rating, isolation level, protection schemes and cooling techniques. The Hardware-in-the-Loop (HIL) is a popular, reliable, safe and cost-effective technology and has been widely used in various fields such as aviation, automotive systems, robotics and power systems for designing and testing purpose. The merit of employing HIL in EMT simulation is that the power system and control device can be tested under extreme high voltages and currents without damaging the equipment or endangering lives. Without connecting to the physical system, by using HIL, the designer can easily test individual device, such as a controller or protective relay, to make sure it works as planned before the device is applied to the physical process. As shown in Fig. 1.1 in HIL, the simulation results have to be synchronized with real-time clock to interface with the physical system or controller under test through an amplifier. The amplifier amplify the low-level signals to high-level signals can be accepted by physical devices. The main challenge in HIL simulation is to provide all of the electrical scenarios needed to fully exercise the under test device, so that the detailed component models can accurately reflect realistic physical phenomena. The simulation time-step is another critical factor in HIL simulation in order to reproduce the high frequency transients and power electronic phenomenons. Moreover, with the ever increasing complexity of power network, more powerful computational hardware are always in demand. In order to simulate a large power system, another option is to divide this system into several subsystems and allocate them to separate but interconnected real-time simulator hardware modules.

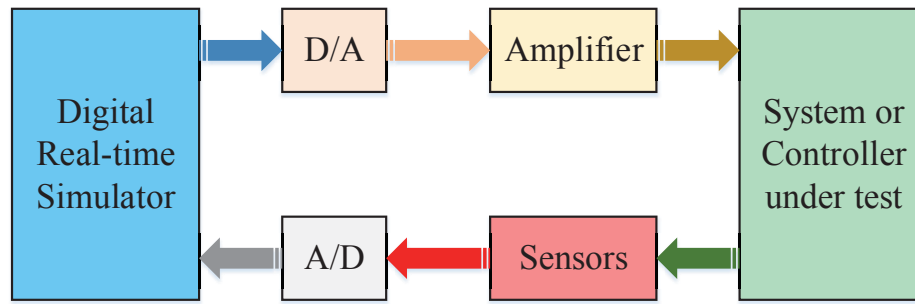


Figure 1.1: Hardware-in-the-loop configuration.

## 1.2 Survey of Transformer EMT Models

### 1.2.1 Transformer Models

The transformer models in the literature for electromagnetic transient analysis can be classified into four groups, and the concepts behind these groups are summarized in this section [29].

**Admittance matrix representation** is developed based on the equations of a multi-winding, multi-phase transformer. The transient-state model is derived by discretizing the steady-state equations using numerical method such as Trapezoid rule. The matrix entries used to represent this model can be derived from excitation tests. However, some accuracy problems may exist if the transient-state equation is expressed in terms of branch impedance matrix, as this matrix can be ill-conditioned for very small exciting currents [30]. Therefore, a more accurate and commonly used method is to express this model in terms of admittance matrix, which does always exist and its entries can be obtained from short-circuit tests. This type model includes phase-to-phase coupling, models terminal characteristics, but can not differentiate from different core topology designs or winding topologies, since all the parameters involved are derived directly from exciting and short-circuit tests. The model is linear, however, the nonlinear and frequency dependent effects such as saturation and hysteresis effects and eddy current effect can be included by connecting external circuit outside the transformer model. Since the admittance matrix of the linear part of the model is fixed, this model can be used without undue burden to emulate one or several power transformers in a large power system to predict their current and voltage waveforms.

**Saturable transformer component (STC)** is based on the star-circuit representation as shown in Fig. 1.2. In this model, the branch is treated as an uncoupled R-L branch, and the saturation effect can be modeled by adding an extra nonlinear inductor at the star point. There are some important issues regarding this model: 1) it can not be used for more than three windings, since the star circuit is not valid for N larger than three; 2) numerical instability is reported under three winding case, but can be solved by modification of the traditional model as in [31]; 3) the connection of the extra nonlinear magnetizing induc-

tance to the star point is not topological correct [32].

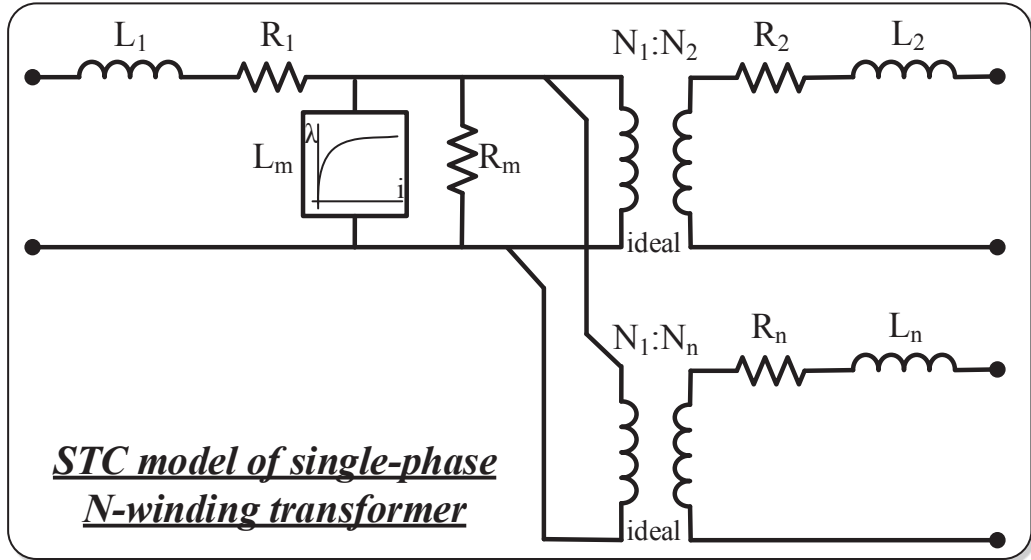


Figure 1.2: STC model of single-phase N-winding transformers.

**Topology-based model** can be further divided into two sub-groups: duality-based models and geometric models. Both sub-groups take into account the transformer physical topology aspect. The difference between these two sub-groups is whether previous mathematical description is involved. The duality-based models is built by creating a equivalent circuit model which is derived from a magnetic circuit using the principle of duality [33, 34]. In the equivalent magnetic circuit, analogous to an electrical circuit, the magnetomotive-force (MMF) sources represent the windings drives the flux flowing through the permeance elements just like a voltage source drives the current flowing through resistive elements. Linear permeance represents the leakage flux paths in the air and the nonlinear permeance represent the flux path in the core. This model can include the effects of saturation effect in each individual limb and yoke, leakage effect and interphase magnetic coupling. In the other type, geometric model is based on the magnetic equations and their coupling to the electrical equations. In both models, the winding resistance, core losses, and capacitive coupling effects can be included by adding extra circuit outside models. This model successfully fills the gap between the finite element (FE) model and the simpler transformer models. Since this model is a geometry-based model, which means the design data of the transformer core is required to develop the model, it implies that this approach can be used in transformer design optimization procedure. Also since this model is a magnetic equivalent circuit based, so the flux distribution inside the transformer core can also be obtained.

**Hybrid transformer model** combines the admittance matrix representation for both the winding modeling and the topology-based model for the representation of core and its flux in the limb and yoke. This model is usually divided into several functional parts to

describe the leakage representation, topologically core representation, winding resistances and capacitive effects [35]. Due to their detailed nature, the computational time of last two types of model can be much longer than the previous two types.

### 1.2.2 Existing Off-line and Real-time EMT Transformer Models

The currently available off-line EMT tools, such as ATP, PSCAD/EMTDC<sup>®</sup> and EMTP-RV<sup>®</sup> are very good tools to design or test a power system. However, for transformer modeling, there are still some drawbacks in these software. For example, for PSCAD/EMTDC<sup>®</sup> a topology-based model, unified magnetic equivalent circuit (UMEC) model is implemented, however in the XMF model, the inter-phase coupling is not represented, and three-phase components are connected as three, single-phase transformers. Furthermore, in the PSCAD/EMTDC<sup>®</sup>, the nonlinearity is represented by a injection current source with a time-step delay, which is not accurate and can induce numerical oscillations if the simulation time-step is not small enough [36]. For EMTP-RV<sup>®</sup>, both the admittance matrix model and the UMEC model are implemented. For ATP, a geometric model, XFORMER, was developed and implemented in [37].

For all these three off-line tools, the nonlinearity in the transformer model is represented by piecewise linear methods. For nonlinear inductance, such a method can be simulated as a switched component consisting of several inductances connected in parallel. The switches connected in series with the inductances can be closed or open to control the equivalent inductance value of this component in the system admittance matrix, so that at each simulation time-step, it is a specific linear inductor. The piecewise method is more accurate than the current injection method, but it still can cause numerical oscillations if frequent switching between linear segments is done. Another method to represent nonlinear elements is by using an analytical function and solver by the Newton-Raphson (N-R) iteration. This is a very time-consuming method and is seldomly used in off-line software due to the fact that several iterations are performed for the convergence and in each iteration the Jacobian matrix and a set of algebraic equation need to be recomputed. The hysteresis model is represented by a type-96 model in ATP and hysteresis reactor in EMTP-RV<sup>®</sup> [38]. The biggest drawback is that all of these off-line softwares do not include frequency-dependent behaviours of transformer.

All of the above statements are also true for currently available commercial real-time digital simulators such as RTDS<sup>®</sup> and RT-LAB<sup>®</sup>. The topology-based UMEC model was implemented for the RTDS<sup>®</sup> simulator [39]. However, in the paper the simulation time-step is not reported. If the simulation time-step is large, this implementation is not efficient for high-frequency or large-scale power system real-time simulation because the real-time constraint can not be satisfied. Furthermore, the UMEC model in RTDS<sup>®</sup> is a piecewise saturation low-frequency model. In the RTDS<sup>®</sup> small time-step library, the transformer model is developed based on the admittance representation and the saturation effect can

be included by connecting an external piece wise saturable function. For RT-LAB<sup>®</sup> only the admittance based piecewise linear model is available and it omits all high-frequency effects.

In the cross platform real-time simulations, the FPGAs can be connected to a traditional PC or CPU by using the optical fiber for communication purpose and the multi-rate technique can be employed to handle the requirements of different subsystem executed on different platforms.

### 1.3 Motivation for This Thesis

Following the advancement of modern power system network, the demand for more accurate and faster performing real-time simulation has greatly increased. To reproduce the transient phenomena occurring in the real world, the real-time power system component has to be represented by a detailed model such as including nonlinear and frequency dependent characteristics. Moreover, in certain applications such as the power electronic application, the simulation time-step, which is another factor effecting the accuracy of the simulation results, will decrease to several micro-second. All of these increase the computational burden in real-time simulation. In order to accommodate large size network while maintaining sufficient accurate real-time simulation, a suitable simulation platform is the first most important consideration before starting any implementation. Compared to general purpose CPUs or DSPs which are basically sequential device, the FPGA is a digital hardware device which allows massively parallel processing, supporting multiple simultaneous threads of execution. Furthermore, the FPGA is a fully custom configurable device which means it can be configured to fit any particular application with an optimized circuit in order to achieve highest performance and lowest hardware resource utilization. Due to the aforementioned reason, the modern FPGAs, in which density and speed have evolved into a mature level, is a suitable platform for the real-time power system simulator.

Nowadays, the FPGAs are being more and more used to design computationally intense applications due to its inherent parallel hardware architecture. The FPGAs have been successfully and widely employed in digital industrial control system. FPGA-based digital hardware models of several power system apparatus such a transmission lines, rotating machines, power electronic equipment, and protective relays also appear in the literature. However, when it comes to real-time transient simulation, a transformer is the most ignored power system component. In real-time simulation, to achieve a small time step, as well as to accommodate a large network size, the transformer model is either entirely omitted or represented using a simplified equivalent containing only the winding leakage impedance.

Driven by the aforementioned reasons in this thesis, the first challenge is to implement a detailed accurate real-time transformer model on FPGA including nonlinear and frequency-dependent effects. In order to represent nonlinear effect, the iterative procedure

has to perform in real-time, which increase the computational burden. On the other hand, to investigate the frequency dependent effect, the simulation time-step has to decrease to a small value, otherwise the high frequency phenomena cannot be captured in real-time. Higher computational complexity and decreased execution time are the main reasons that a detailed and accurate implementation of transformer is yet to be reported in literature. On the FPGA side, the challenge is that programming FPGA using gate-level language is quite difficult compared with software programming using high-level language such as C/C++. In every calculation, the operands are assigned to the inputs of a particular hardware algorithm module and the calculated results flow out of the output ports after certain clock cycles. Many of such modules can line up and work simultaneously in parallel fashion when the parallel algorithm are processed. Registers can be inserted for the synchronization purpose. The other challenge is that the FPGA programming is tedious and very hard to debug, which makes it restricted only for experienced FPGA user. Moreover, the exploitation of FPGA inherently hardware parallel and pipeline architecture is another challenge. The focus of this thesis is to implement different types of FPGA based real-time transformer models including nonlinear and frequency dependent effects to improve the accuracy. Other major power system components, such as transmission line and passive element models, are also developed on FPGAs in this thesis in order to build a fully FPGA-based case study environment. A power flow controller application, Sen-transformer featured by lower cost and higher reliability and efficiency, is also developed in the fifth part of this thesis.

## 1.4 Research Objectives

To achieve real-time model of power transformer implementation on the FPGA, the following main objectives need to be done in this thesis:

1. Power transformer model needs to be developed and implemented on the FPGA, and the governing equations of the model are optimized for realizing paralleled computations. More complexity of the transformer model are developed to realize more accurate emulator, which implies more hardware resource utilization and longer execution times. More dedicated paralleled and deep pipelined computation scheme need to develop to achieve real-time simulation while maintain high accuracy.
2. Other power system components real-time models are going to be implemented on FPGA to execute power system real-time simulation, such as transmission line model, ideal voltage source model, passive elements model, eddy current model and hysteretic model.
3. Paralleled and pipelined algebra solver need to be developed on the FPGA to solve linear or nonlinear equations. In the FPGA-based electrical machine real-time emula-



tion [17–19,28], several general computing units had been developed, such as Matrix-Vector Multiplication (MVM) unit and Gauss-Jordan Elimination (GJE) unit. However, to implement a detailed real-time transformer model, more hardware computing units are needed to develop such as Sparse-Dense Matrix Multiplication (SMxDM) unit, Lower-Upper Decomposition (LUD) unit, Newton-Raphson nonlinear solver (N-R) unit, LU based Inverse Matrix Calculator (IMC) unit, Forward Substitute (FWS) unit and Backward Substitute (BWS) unit. These developed units can also be used to implement other FPGA-based power system component models in real-time.

4. The EMTP algorithm is a sequential procedure, in order to take advantages of FPGA's parallel feature, the possible parallel procedures need to be extracted and allocated into parallel computational scheme.
5. The basic floating point mathematic hardware units and other digital components need to be built on FPGA, such as adders/subtractors, multipliers, dividers, RAMs, ROMs and registers. More complicated arithmetic units, such as spares matrix-vector multiplication, matrix-matrix multiplication and accumulator are made of many basic units.
6. The mathematical modeling of HR-MEC based Sen-Transformer model need to be investigated. The real-time implementation of the ST model and its corresponding control algorithm need to be developed.
7. The real-time emulation results need to be captured and evaluated by the results obtained from other methods. The real-time emulation results captured by vendor provided tools, such as Xilinx<sup>®</sup> ChipScope Pro, and oscilloscope are verified against off-line simulation results obtained from ATP or finite-element solutions.

## 1.5 Thesis Outline

The rest of this thesis consists of six chapters and is outlined as follows:

Chapter 2 presents an introduction of FPGA background and technology including its architecture, fundamental components, design tools, and design flow. This chapter also discusses and compares some FPGA design technologies and programming strategies with the traditional platform such as CPU and DSP. The IEEE 754 single-precision binary floating-point data format provides both higher precision and wider range compared to the fixed-point data format. The intrinsic space-oriented hardware architecture results in FPGA being a suitable device for the high computational burden applications such as the model development for real-time EMT simulation.

Chapter 3 presents a real-time nonlinear hysteretic power transformer model on the FPGA. The linear dynamics of the model are based on the admittance matrix representation. The hysteresis phenomena are modeled based on the Preisach theory, whereas the

eddy currents in the core are modeled using a frequency-dependent equivalent model. The nonlinear solution in real time is undertaken using full Newton iteration. This chapter elaborates on the transformer linear model formulation, the Newton-Raphson iteration method, the Preisach hysteretic formulation, and the interfacing between linear and nonlinear models. The entire power transformer nonlinear hardware model as well as other power system models, such as Bergeron transmission line model, passive element model, source model, and linear solver model, are designed and implemented. Two power system test cases were developed using the VHDL language, employing IEEE 32-bit floating-point precision and various studies are simulated and analyzed, such as inrush current transients and fault transients. All models employ full hardware parallelism and deep pipelining to optimize the hardware resource and the model latency. The real-time simulation results in each simulation time-step are sent out from the FPGA to the oscilloscope through a DAC board. The real-time results are captured and validated against the off-line simulation results.

Chapter 4 presents a real-time nonlinear high resolution magnetic equivalent circuit (HR-MEC) based transformer model on FPGA. This model is inspired by the mesh generated in finite element (FEM) tool to depict the major flux paths in the transformer core. In this model, the iron core is represented by nonlinear permeances and the leakage inductance is represented by linear permeances. The magnetomotive-force (MMF) source generated by the current drives the flux flowing through the nonlinear permeances in the iron core and the linear permeances in the air. The HR-MEC formulation is derived and the detailed hardware implementation of the proposed model is described, along with the hardware resource utilization and achieved simulation time-step. Since the nonlinear permeances in the core already take into account the nonlinear and hysteresis effects, so that no nonlinear element is required to be connected external at the linear part terminal. However, since the HR-MEC's admittance matrix is not constant, a LU decomposition based network solver module is developed to solve the nodal analysis equations and to calculate the inverse matrices. In order to observe the major flux paths in the transformer core, a 3-D finite element method (FEM) model is developed in *JMAG<sup>®</sup> Designer*. Two FPGA-alone power systems are implemented and various case studies are simulated and analyzed to show the accuracy of the real-time model.

Chapter 5 presents a MEC based Sen-transformer (ST) real-time model on FPGA. The concept of the ST is introduced by discussing the importance of a power flow controller and its advantages over other similar functional devices such as unified power flow controller (UPFC). The ST consists of a multi-winding transformer and tap-changers that can regulate the power flow through a transmission line by injecting a series connected controllable voltage. The ST operating principle is described and its MEC based Norton equivalent circuit and the corresponding discretized formulations are derived from the first principles. The FPGA hardware implementation of ST is described and the interfacing with



other power system models and computational unit are depicted by using a multiple-stages chain during one simulation time-step. A fully paralleled and pipelined hardware architecture is developed to achieve an accurate real-time emulation as well as the lowest latency and smallest hardware resource consumption. A 3-D ST FEM model as well as its coupled external circuit are developed in *JMAG<sup>®</sup> Designer* to depict the major flux paths in the ST core. Its simulation results are used to validate against the real-time simulation results captured by the oscilloscope. Another power system implemented on FPGA is also analyzed to show the power flow regulation capability of the ST. The step-by-step current and voltage transition responses during the regulating period are also captured by the oscilloscope.

Chapter 6 gives the conclusions of this thesis and the directions of the future work.

# 2

## FPGA Background

The digital system emulator can be implemented based on microprocessor (CPU), reconfigurable hardware (FPGA) or graphics processing unit (GPU). Among those platforms, FPGA successfully made inroads into real-time simulation due to its parallel hardwired architecture and pipelined computing capability. Compared to FPGA, the CPU has limited parallelism due to the fact that there are only several cores, which prevent CPU being a suitable platform for the complex real-time electromagnetic transient emulation. On the other hand, the massive threads in GPU make it possible to perform parallel computing. However, those cores are grouped and the data transfer between different groups is slow, and the local memory within each group has limited size.

The FPGA, a reconfigurable hardware device, is a two dimensional programmable logic gate array. The designer can program the device, which essentially reconfigures the gate array to implement an optimized circuit within the device to realize a specific application. The designer can generate as many computing units as long as the reconfigurable device hardware resource allows to achieve parallel computation. Even more the designer can generate as much memory and assign small local memory to computing units, which makes the memory be a part of the data path to achieve pipelined computation. In a particular application, the computing units and the memory can be interconnected by programming the FPGA by using VHDL. By doing so, the application's functionality is represented by a circuit, then the data can line-up and go through the circuit to carry out the results.

In order to accomplish the HIL simulation, the real-time simulators are required to be connected with the yet to be tested devices or controllers. The FPGA offers robust solutions to address a high bandwidth and low latency communication between the simulators and the devices with a comprehensive set of memory controllers and physical layer interfaces. The memory controllers included in the IP catalog supports DDR3, DDR4 and multi-task

dual in-line memory module with maximum achievable data rates of 1.5 Gb/s. Several standard interface protocols are also available in FPGA IP catalog such as Display Port and PCIeExpress. Xilinx<sup>®</sup> also developed a scalable link-layer protocol, Aurora<sup>®</sup>, which aims to use point-to-point high-speed serial links in an embedded system, such as chip-to-chip and board-to-board and backplane links. In summary, compared to the CPU or GPU, the FPGA is more efficient in four aspects: lower latency, smaller hardware space, lower power consumption and easier to interface with other device.

Due to the aforementioned reasons, aside from the power system applications, other fields of application such as image processing, industrial control systems, automotive and telecommunications have started to move toward FPGA due to these features [40–49].

A brief introduction to FPGA including its architecture is presented and then the design tools, design flow and certain important issues regarding the programming technology are discussed.

## 2.1 FPGA Architecture

The FPGAs is a two dimensional logical block array interconnected by matrix of wire and programmable switch. Each block can be programmed to perform certain logic function, and interconnected by programmable switch to achieve field programmability and thus implementing particular application.

Fig. 2.1 gives the Xilinx Virtex-7<sup>®</sup> FPGA architecture block diagram which is the employed platform in this thesis. All programmable logic blocks are organized by column, and each column contains one or two types of logic programmable resource. According to the function they can perform the logic resource can be classified into several categories: configurable logic block (CLB) blocks, on-chip memory blocks, digital signal processor (DSP) blocks, I/O blocks, clocking region, PCIe<sup>®</sup> interface blocks and programmable switch blocks. The detail hardware resource for Virtex<sup>®</sup>-7 VX485TFFG1761-2 FPGA is listed in Table.2.1. More discussion regarding CLBs block, memory blocks, DSPs blocks will be provided later.

### 2.1.1 Configurable Logic Blocks (CLBs)

The Configurable Logic Blocks (CLBs) are the most important and main logic resource in the FPGA and provide the reconfigurable capability when the FPGA need to be implemented for realizing particular application. As shown in Fig. 2.2, each CLB contains two slices, and each slice is composed of four 6-input Look-up Tables (LUTs), eight flip-flops, multiplexers and arithmetic carry logic. The two slices in each CLB are not connected, and each CLB, as well as each slice, is organized as a column. In each column, the slice is connected by an independent carry chain. A switch matrix connected to all slices is for accessing to the general routing matrix purpose. The 6-input LUTs in each slice has 6 inde-

Table 2.1: Module level hardware resource utilization

Resource Type	Unit	Number
Logic Resource	Cells	485,760
Configurable Logic Blocks	Slices	75,900
Memory Blocks	Kb	37,080
Digital Signal Processor Blocks	Slices	2,800
I/O Blocks	Bank	14
PCIe Block		4

pendent inputs and two independent outputs, so that it also can be implemented as two 5-input LUTs with common inputs but independent outputs. There are two types of slice: SLICEL and SCLICEM. Both two type slices can be used to provide logic, arithmetic and ROM function. In addition the SLICEM type slice can support two additional function: distributed RAMs and shift registers. Each CLB can contain two SLICEL or one SLICEL and one SLICEM. The simplified slice architecture within one CLB is depicted in Fig. 2.3

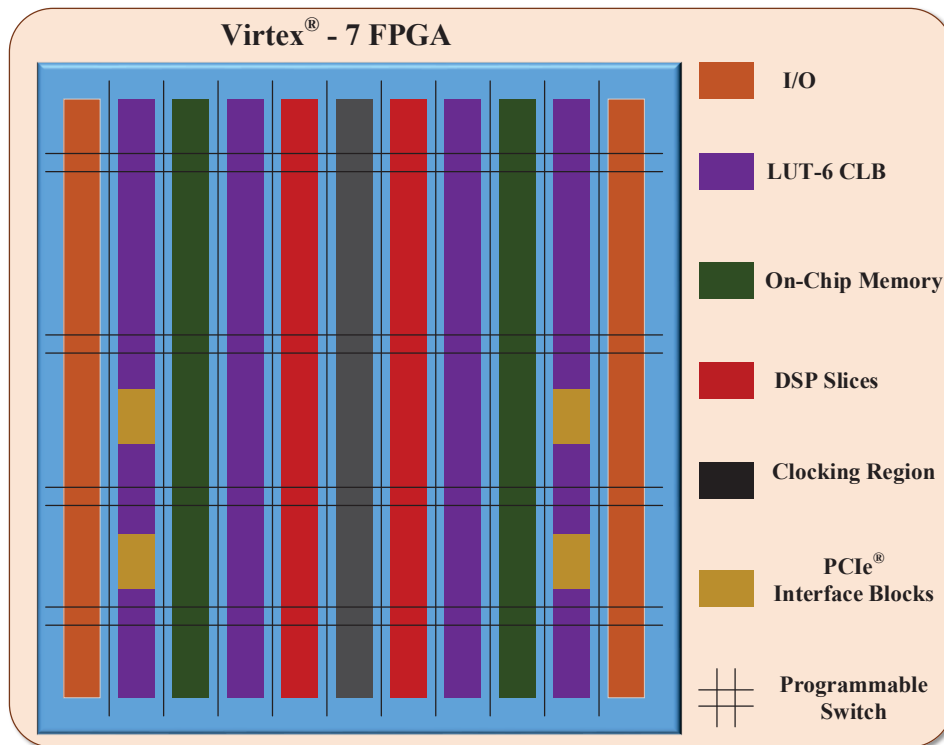


Figure 2.1: Xilinx Virtex-7<sup>®</sup> FPGA architecture block diagram [50].

[50].

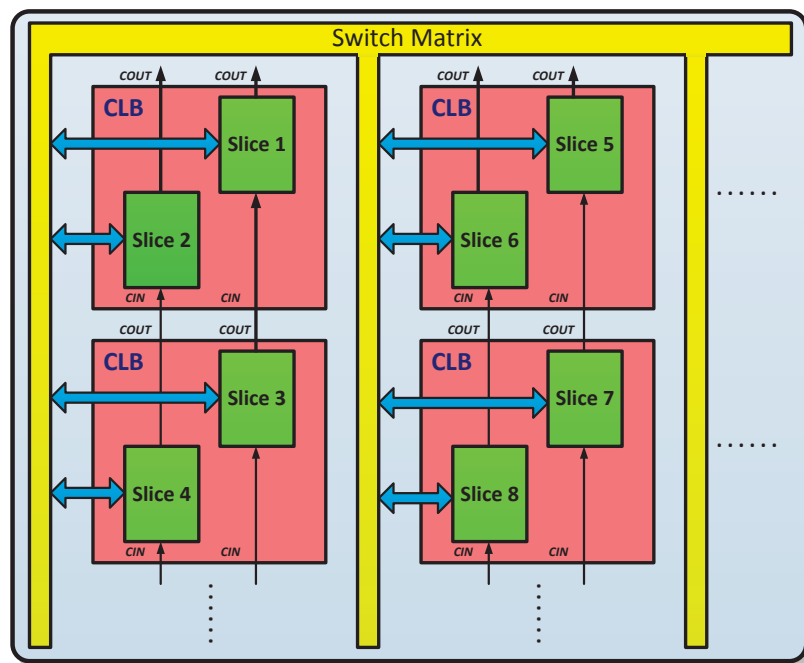


Figure 2.2: CLB architecture and the connection to switch matrix [50].

### 2.1.2 Block RAMs

Besides the distributed RAM, Xilinx Virtex-7<sup>®</sup> FPGA also feature a 36Kb block RAM to store large number of data if the implemented application require. Each 36Kb block RAM can be configured as either two independent 18Kb RAMs, or one 36Kb RAM. The 36Kb RAM can also be configured as a deeper and wider memory when cascaded with an adjacent 36Kb block memory. The true dual-port 36Kb block RAM data flow is depicted in Fig.2.4 (a). There are two completely independent access port, A and B, and each port has individual data in, data out, clock, write enable and address signal. Date can be written to and read from either access port. Both write and read operation are synchronous and require a clock edge. The 36Kb block RAM also can be configured in a single-port RAM mode or a simple dual-port RAM mode. In simple dual-port as shown in Fig.2.4 (b), the independent read and write operations can occur simultaneously, where port A and port B are used for read and write purpose, respectively [51]. All the types of RAM modules, as well as ROM module can be implemented using the Xilinx CORE Generator<sup>™</sup> block memory module.

### 2.1.3 Digital Signal Processing Blocks (DSPs)

All Xilinx Virtex-7<sup>®</sup> family have many dedicated, full-custom, low-power DSP slices block. The DSP48E1 slice in Virtex<sup>®</sup>-7 VX485TFFG1761-2 provides high efficient for digital signal

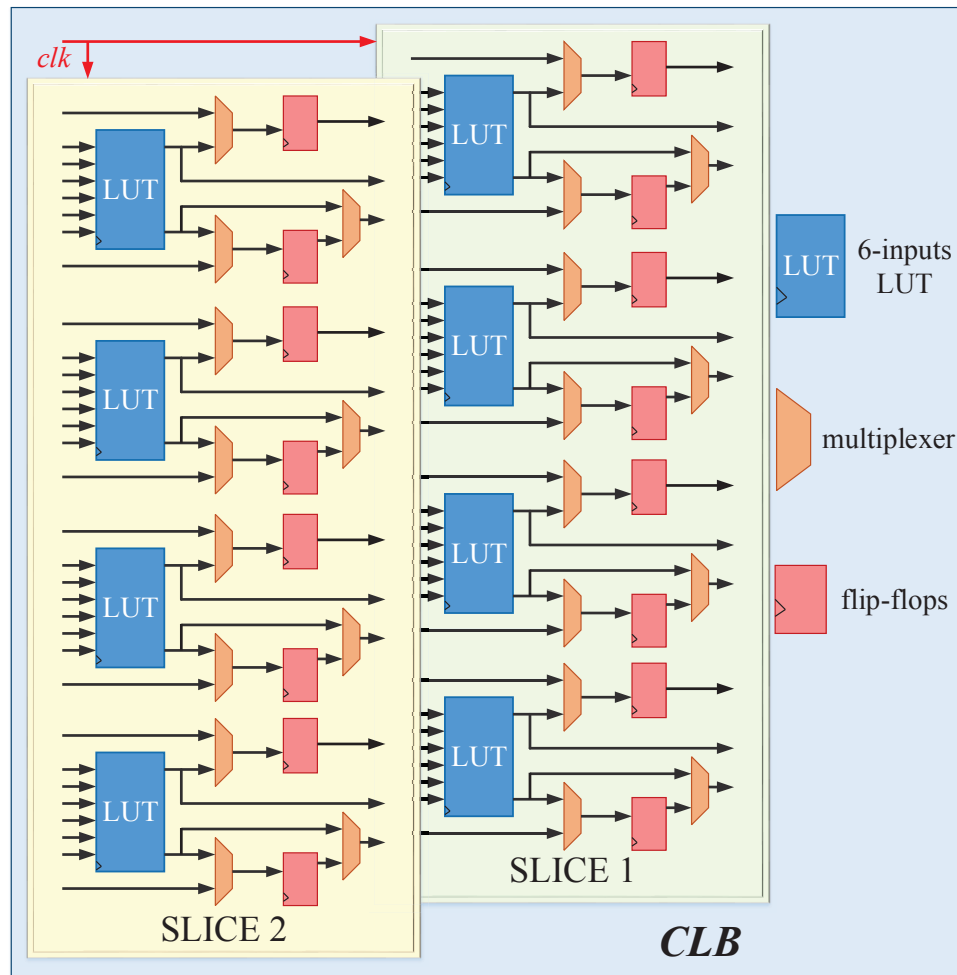


Figure 2.3: Simplified slice architecture [50].

processing application. These blocks can implement fully parallel algorithms combining high speed and system design flexibility. Fast Fourier Transforms (FFTs), floating point computation (multiply, add, subtract, divide) counters, and large bus multiplexers are some application of the DSP slice. As shown in Fig. 2.5 some features in the 7 family FPGA DSP48E1 slice are 25x18 signed multiplier, 48-bit accumulator, power saving pre-adder, 12/24-bit single-instruction-multiple-data (SIMD) arithmetic unit, cascade paths for wide functions, pipeline registers for high speed and pattern detector [52].

## 2.2 FPGA Design Tools and Design Flow

A FPGA design is built up by programming Very High-Speed Integrated Circuit Hardware Description Language (VHDL), which is gate-level programming language. Along with the development of FPGA hardware scale, the FPGA vendors also update their design tools to integrate development environment from design entry to download bitstream to

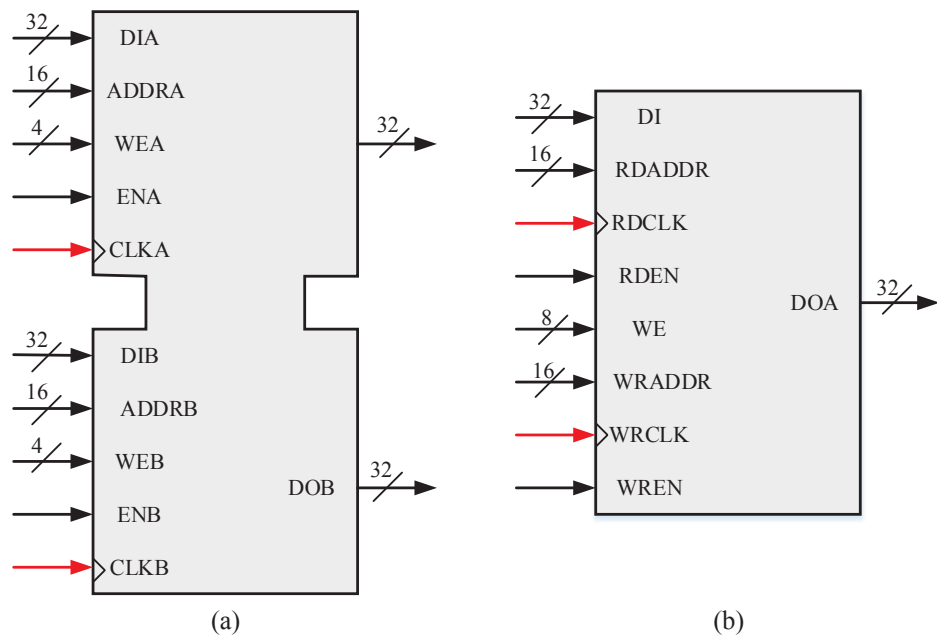


Figure 2.4: Xilinx Virtex-7<sup>®</sup> FPGA block RAM (a) true dual-port RAM; (b) simple dual-port RAM [51].

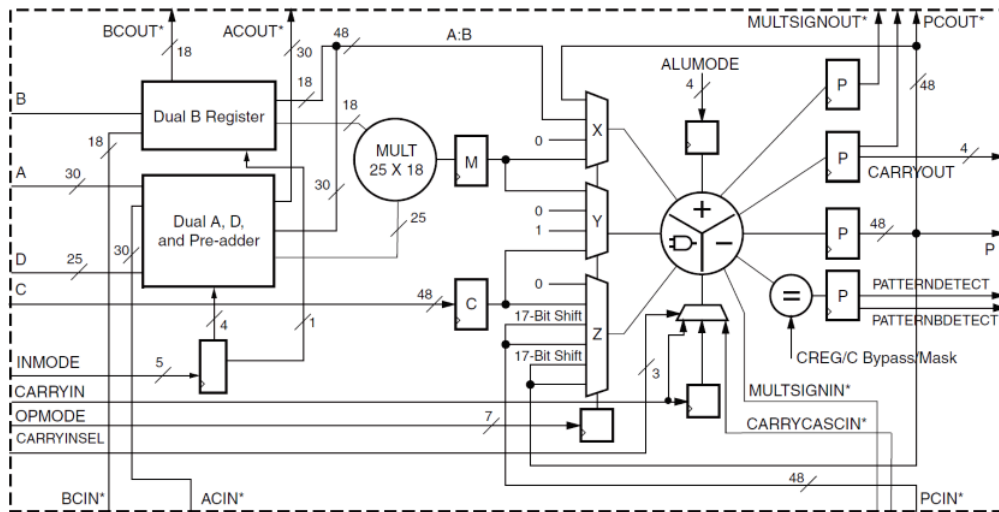


Figure 2.5: Xilinx Virtex-7<sup>®</sup> FPGA DSP48E1 slice [52].

the chip. Xilinx<sup>®</sup> have released two mature design tools, ISE<sup>®</sup> Design Suit and Vivado<sup>®</sup> Design Suit. Compared to ISE Design Suit, the Vivado Design Suit being the new generation tool, can predict the runtime during the synthesis and implementation period and faster than alternative solutions. The FPGA design flow consists of design entry, synthesis, implementation, programming and on-chip debug, as depicted in Fig. 2.6.

**Design entry-** In this stage, the user can build up a design from scratch by either writing textual programming language such as VHDL or Verilog HDL, or by using the

schematic approach. The most widely used method is to write programming language, especially when the design is algorithm oriented. When writing the VHDL or Verilog HDL, the designer essentially describe the behavior or structure of a digital system. For widely used hardware component such as memory block and arithmetic unit, the vendor provided well-designed and fully-tested Intellectual Property (IP) cores are the most convenience tools, such as Core Generator™ System included in Xilinx's design tools.

**Function simulation-** In this procedure the user can create a test bench to verify the entry design by checking the output simulation waveforms. In a large hierarchical HDL design, to make a easier debug procedure, the user can also perform separate simulation before testing the entire design. Various tools such as Isim®, Vivado®, ModelSim® and Quartus II® simulator tools can be used in this procedure.

**Synthesis-** This process will check code syntax and analyze and optimize the hierarchy of the HDL design. Synthesis procedure also generates a netlist file containing both logic design and constraints. For example, in ISE® Xilinx® Synthesis Technology (XST) creates a NGC file to describe the design in terms of logic components such as adders, subtractors.

**Translate-** This procedure merges the netlists and design constraints and outputs a file which describes the logical design in terms of logic elements, such as AND gates, OR gates, LUTs, flip-flops and RAMs. In ISE® software, the created file called Native Generic Database (NGD) file.

**Map-** This process reads the NGD file and maps the logic into FPGA elements, such as CLBs. The output design file is a Native Circuit Description (NCD) file in ISE® which physically fits the design into FPGA hardware resource.

**Place and route-** This process reads NCD file and places and routes the design to physical location and select the wire and switch for the interconnection. In ISE® software produces a new NCD file.

**Timing analysis and simulation-** This process performs after the implementation procedure to guarantee that all the timing constraints are satisfied for the FPGA design.

**Programming file generation-** This process takes the NCD file and produces a bitstream which can be accepted and downloaded into FPGA using JTAG interface form the host PC.

**On-chip debug-** After downloading bitstream to FPGA, the user can verify the configured FPGA by executing the desired function. The vendor provided tools such as Xilinx® ChipScope Pro can be used to upload data from the FPGAs to PC at this stage. The designer can also send the digital output signal to an oscilloscope through digital-to-analog converter (DAC) board. Now the FPGA design flow is finished and the FPGA is ready to use.



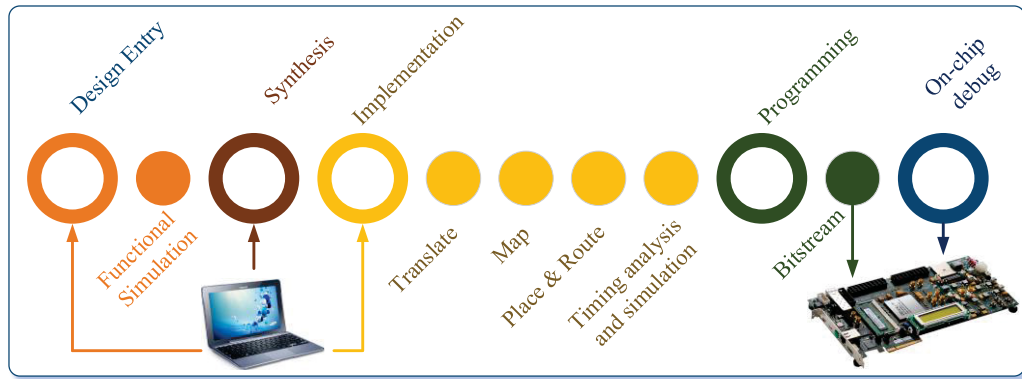


Figure 2.6: General FPGA design flow.

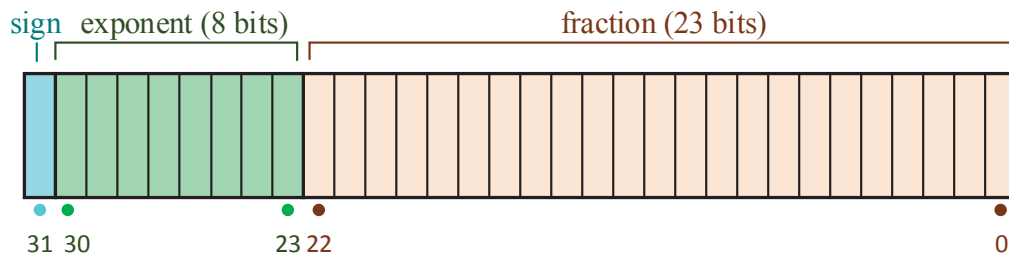


Figure 2.7: Single precision floating-point representation.

## 2.3 FPGA Programming Technology

### 2.3.1 Data Representation

Before implementing any application in a digital system, such as CPU or FPGA, the data representation is always the first consideration, since it affects the accuracy of computation and the hardware resource utilization. There are two data representation systems widely used to approximate a real number: fix-point number and floating-point number system. In fix-point system, the number of digits to represent integer part of and fractional part of a number is fixed once the decimal point is fixed. The decimal point location does not change during the entire computation and that can produce precision loss when calculated results have more bits than the operands, such as in multiplication. The floating-point number system, on the other hand, can represent a number in dynamic range and results in high accuracy computation. However, the floating-point computation requires more hardware resource and takes longer time compared to fix-point computation. In this thesis, the IEEE 754 single-precision binary floating-point format is employed in FPGA based real-time programming. The main reason is that the EMTP algorithm requires both high precision, i.e., Newton-Raphson method and wider range of the number representation, and FPGA can provide massive hardware resource to implement floating-point operators while achieving high speed. As shown in Fig. 2.7, the single-point floating-point format

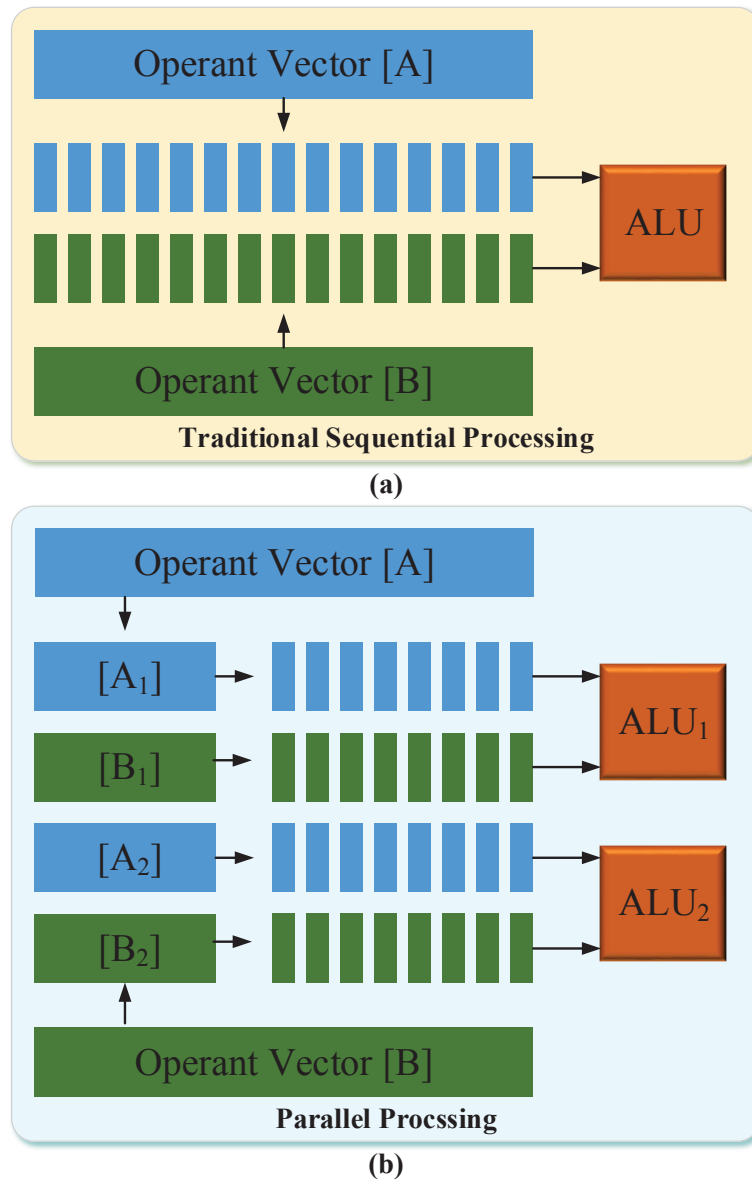


Figure 2.8: Data flow for (a) sequential processing; (b) parallel processing.

consists of one *sign* bit, 8 *exponent* bits and 23 *fraction* bits. Its value  $v$  can be calculated as

$$v = (-1)^{sign} \times 2^{(exponent - exponentbias)} \times (1.fraction), \quad (2.1)$$

where the *exponent bias* is 127. The range of  $v$  is approximately  $10^{-38}$  to  $10^{38}$ .

### 2.3.2 Parallel Processing

In the traditional processor such as general purpose CPU or DSP, the computation is executed sequentially, as shown in Fig. 2.8 (a). The operands in vector  $A$  and  $B$  are processed by the Arithmetic Logic Unit (ALU) one after the other sequentially, even if the operation

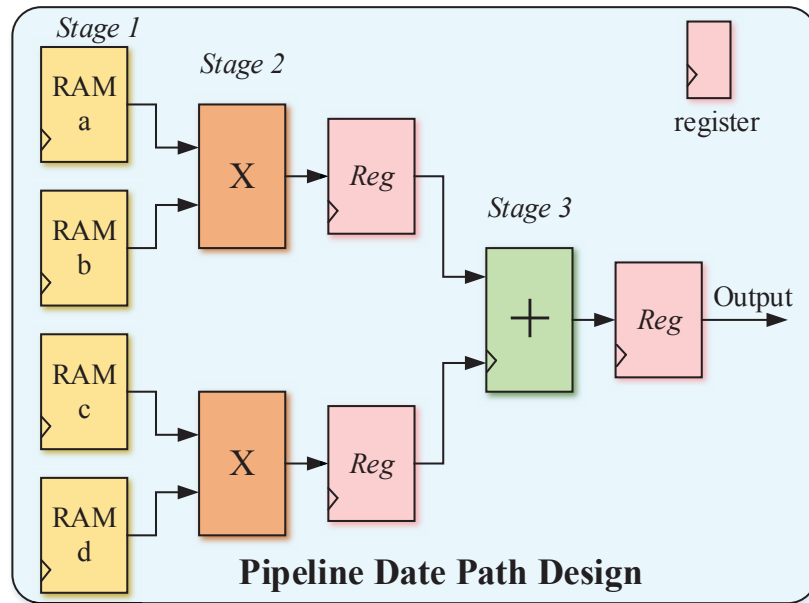
is independent to each other. On the other hand, in FPGA based implementation, the designer can configure the FPGA into a larger number of parallel processing units and all units can process their own data simultaneously. Besides as discussed earlier, both CLB slice and block memory slice can be configured into various types of memory with various depth and bus width and all the memory can be accessed concurrently. Thus by partitioning the operand vectors  $A$  and  $B$  into several small vectors, the operands in small vector can be read out and then assigned to ALU concurrently. Fig. 2.8 (b) depict the parallel processing in FPGA by partitioning memory and configuring multi-units. Even more, if the application's algorithm can be partitioned into several independent parts, the FPGA can also be partitioned into several independent circuit to perform independent calculations simultaneously.

### 2.3.3 Pipelining

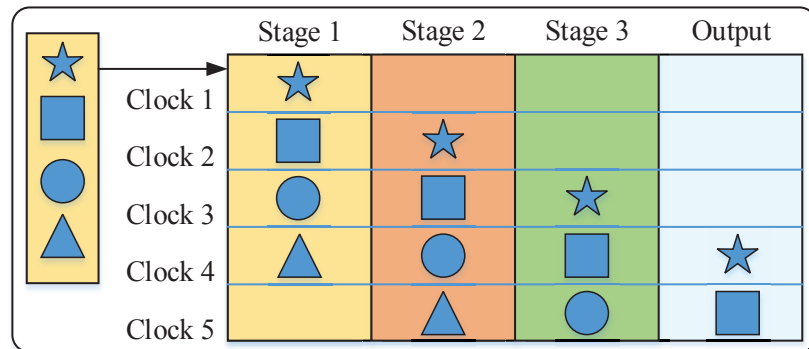
Another FPGA based implementation programming strategy to improve hardware performance is pipelining technology, in which a function is divided into several stages depending on the operations. Between each stage registers are inserted to separate the operations. When the computation is executed, data march through the operations at every clock cycle. Although pipeline technology introduces a delay in terms of clock cycles, it improves the computational throughput by increasing the number of operation in each clock cycle. Fig. 2.9 (a) gives pipeline data path design to compute,

$$output = a \times b + c \times d, \quad (2.2)$$

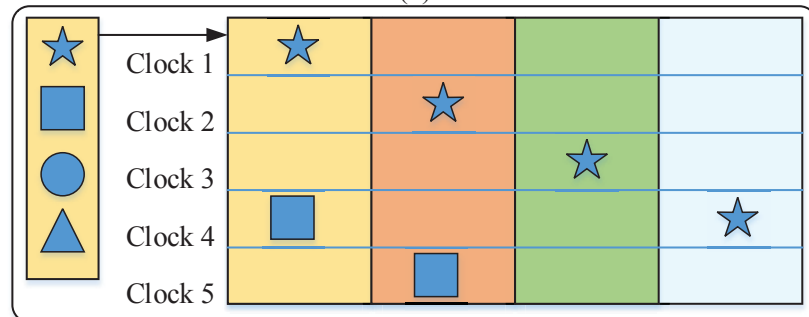
In *stage 1* all processing data are read out from RAMs block simultaneously. The *stage 2* and *stage 3* separated by registers perform multiplication and addition, respectively. Fig.2.9 (b) shows that the data flows into the pipeline every clock and is processed independently at each stage. In the fourth clock cycle the first valid output is carried out, and next valid result will flow out the pipeline in next clock cycle. However in the traditional data flow, as shown in Fig. 2.9 (c) the next valid result flow out after another 3 clock cycles. In this case, the pipeline throughput increases to one result per clock cycle from one result per three clock cycles in traditional design. The pipeline strategy can improve the computation performance especial when the algorithm is dependent on the previous results, such as accumulator.



(a)



(b)



(c)

Figure 2.9: (a) Pipeline data path design and data flow for (b) pipeline processing; (c) traditional processing.

## 2.4 Summary

This chapter describes the fundamental aspects of FPGA architecture, design tools and flow and programming technologies. Due to its intrinsic massive parallel architecture,

hardware pipelining computation and custom configuration the FPGAs outperform the general purpose CPU for digital emulation. Besides, the vendors provide all environmental support design softwares and well-designed and fully-tested IP cores allow the designer implement a FPGA-based digital hardware emulator efficiently. Therefore, the FPGAs is chosen as the platform in this thesis for the real-time EMT power transformer model emulation.

# 3

## Admittance-Based Real-Time Transformer Model

A digital hardware emulation is proposed for the power transformer on a field programmable gate array (FPGA) for real-time electromagnetic transient simulation. The linear model of the transformer is based on the admittance matrix approach; nonlinearity modeling including hysteresis phenomena is carried out based on Preisach theory; a cascade of RL lumped equivalent circuit was used to model frequency dependent behaviour. The other components in the emulated power system includes transmission lines, linear lumped RLC elements, supply sources, and circuit breakers. The power system network is solved using sparse matrix techniques to improve the efficiency. The algorithm and hardware modeling of each component are discussed before giving the hardware implementation details. The entire power transformer nonlinear hardware model and the associated numerics were developed using the VHDL language, employing IEEE 32-bit floating point precision. All the component models are fully parallelized and pipelined to achieve the lowest latency and smallest hardware resource consumption. Two case studies are analyzed to show the accuracy and efficiency of this proposed model under various transient conditions and the real-time results captured on the oscilloscope are compared with off-line results from the ATP software. The hardware resource utilized by this transformer model is 15%. Therefore, this real-time transformer model can be used to emulate transformers in a large power system network while maintaining accurate results in which other power system component models, such as machines, can be modeled by the rest 85% of the hardware resource. In this chapter, three transformers and three transmission lines are modeled on FPGA with the time-step of  $10\mu s$  with approximately 50% of hardware utilization. It is possible to simulate larger systems by either increasing the hardware utilization or simulation time-step. This real-time transformer can also be employed to model the

instrument transformers such as current transformer (CT) and potential transformer (PT) since their equivalent circuits are similar to the proposed circuit in this chapter.

### 3.1 Admittance-based Linear EMT Power Transformer Model

#### 3.1.1 Linear Module Formulation

The admittance matrix formulation results from the mutually coupled coils concept. Under transient conditions, the governing equations for a single-phase  $n$  winding transformer can be expressed as,

$$\frac{d}{dt}\mathbf{i}_T = \mathbf{L}^{-1}\mathbf{v}_T - \mathbf{L}^{-1}\mathbf{R}\mathbf{i}_T \quad (3.1)$$

where  $\mathbf{v}_T$  and  $\mathbf{i}_T$  are  $n \times 1$  vectors of instantaneous terminal voltages and currents, respectively, expressed as,  $\mathbf{v}_T = [v_{T1}(t) \ v_{T2}(t) \ \dots \ v_{Tn}(t)]^T$ , and  $\mathbf{i}_T = [i_{T1}(t) \ i_{T2}(t) \ \dots \ i_{Tn}(t)]^T$ .  $\mathbf{R}$  is an  $n \times n$  diagonal matrix of winding resistance,  $\mathbf{L}$  is an  $n \times n$  matrix of winding leakage inductances, given as

$$\mathbf{R} = \text{diag}\{R_1 \ R_2 \ \dots \ R_n\}, \mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & \dots & L_{1n} \\ L_{21} & L_{22} & \dots & L_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ L_{n1} & L_{n2} & \dots & L_{nn} \end{bmatrix} \quad (3.2)$$

The entries of  $\mathbf{R}$  and  $\mathbf{L}$  can be found directly from the short-circuit test data.

For a three-phase transformer the governing equation is the same as (3.1), except each entry of  $\mathbf{R}$  and  $\mathbf{L}$  matrices is a  $3 \times 3$  submatrix,  $\mathbf{R}_p$  and  $\mathbf{L}_{pq}$ , ( $p, q = 1, 2, \dots, n$ ) given as

$$\mathbf{R}_p = \begin{bmatrix} R_{pa} & 0 & 0 \\ 0 & R_{pb} & 0 \\ 0 & 0 & R_{pc} \end{bmatrix}, \mathbf{L}_{pq} = \begin{bmatrix} L_{paqa} & L_{paqb} & L_{paqc} \\ L_{pbqa} & L_{pbqb} & L_{pbqc} \\ L_{pcqa} & L_{pcqb} & L_{pcqc} \end{bmatrix}, \quad (3.3)$$

where a, b, c subscripts stand for the three phases.

By multiplying  $dt$  on the both sides, the (3.1) can be written as,

$$d\mathbf{i}_T = \mathbf{L}^{-1}\mathbf{v}_T dt - \mathbf{L}^{-1}\mathbf{R}\mathbf{i}_T dt. \quad (3.4)$$

Using Trapezoidal rule to discretize (3.4),

$$\mathbf{i}_T(t) - \mathbf{i}_T(t - \Delta t) = \mathbf{L}^{-1}[\mathbf{v}_T(t) + \mathbf{v}_T(t - \Delta t)] \frac{\Delta t}{2} - \mathbf{L}^{-1}\mathbf{R}[\mathbf{i}_T(t) + \mathbf{i}_T(t - \Delta t)] \frac{\Delta t}{2}, \quad (3.5)$$

where  $\Delta t$  is the simulation time-step.

By rearranging (3.5) rewritten as,

$$\left[ \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right] \mathbf{i}_T(t) = \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{v}_T(t) + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{v}_T(t - \Delta t) - \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \mathbf{i}_T(t - \Delta t) + \mathbf{i}_T(t - \Delta t), \quad (3.6)$$

with  $\mathbf{I}$  being the  $n \times n$  identity matrix.

Finally, the Norton equivalent formula can be given as,

$$\mathbf{i}_T(t) = \mathbf{G} \mathbf{v}_T(t) + \mathbf{hist}(t - \Delta t), \quad (3.7)$$

with  $\mathbf{G}$  being the  $n \times n$  admittance matrix and  $\mathbf{hist}(t - \Delta t)$  being the  $n \times 1$  vector containing history terms, which are given as,

$$\mathbf{G} = \left[ \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right]^{-1} \frac{\Delta t}{2} \mathbf{L}^{-1}, \quad (3.8)$$

$$\mathbf{hist}(t - \Delta t) = \mathbf{G} \mathbf{v}(t - \Delta t) + \left[ \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right]^{-1} \left[ \mathbf{I} - \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right] \mathbf{i}_T(t - \Delta t). \quad (3.9)$$

In order to express the vector  $\mathbf{hist}(t - \Delta t)$  in terms of  $\mathbf{v}(t - \Delta t)$  and  $\mathbf{hist}(t - 2\Delta t)$ , (3.7) is substituted into (3.9),

$$\mathbf{hist}(t - \Delta t) = \mathbf{G} \mathbf{v}(t - \Delta t) + \mathbf{AB} [\mathbf{G} \mathbf{v}(t - \Delta t) + \mathbf{hist}(t - 2\Delta t)] \quad (3.10)$$

$$= [\mathbf{G} + \mathbf{ABG}] \mathbf{v}(t - \Delta t) + \mathbf{AB} \mathbf{hist}(t - 2\Delta t) \quad (3.11)$$

$$= \mathbf{H} \mathbf{v}(t - \Delta t) + [\mathbf{I} - 2\mathbf{GR}] \mathbf{hist}(t - 2\Delta t), \quad (3.12)$$

where

$$\mathbf{A} = \left[ \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right]^{-1}, \quad (3.13)$$

$$\mathbf{B} = \mathbf{I} - \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R}, \quad (3.14)$$

gives

$$\mathbf{AB} = \left[ \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right]^{-1} \left[ \left( \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right) - 2 \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right], \quad (3.15)$$

$$= \mathbf{I} - 2 \left[ \mathbf{I} + \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R} \right]^{-1} \frac{\Delta t}{2} \mathbf{L}^{-1} \mathbf{R}, \quad (3.16)$$

$$= \mathbf{I} - 2\mathbf{GR}, \quad (3.17)$$

and finally gives the formula for  $\mathbf{hist}(t - \Delta t)$

$$\mathbf{hist}(t - \Delta t) = \mathbf{H} \mathbf{v}(t - \Delta t) + [\mathbf{I} - 2\mathbf{GR}] \mathbf{hist}(t - 2\Delta t), \quad (3.18)$$

where

$$\mathbf{H} = 2(\mathbf{G} - \mathbf{GRG}). \quad (3.19)$$



### 3.1.2 Linear Module Hardware Design

There are two main functions for this module: (1) calculate the history currents and send them to the linear solver; (2) save the updated history currents within this module so as to use them in the next simulation time-step. The main control module sends a signal to this module to indicate which operation is to be executed.

As indicated by (3.18), the history current calculation can be carried out by matrix multiplication. A matrix vector multiplication unit (MVM) which is also part of the linear solver has been developed for this purpose. Since the entries in  $\mathbf{R}$ ,  $\mathbf{L}$ ,  $\mathbf{G}$  and  $\mathbf{H}$  are the properties of transformer which are unchanged, these parameters can be saved in memory before the real-time simulation begins. Furthermore, instead of saving  $\mathbf{R}$  and  $\mathbf{L}$  individually, the result of  $2(\mathbf{G} - \mathbf{GRG})$  and  $(\mathbf{I} - 2\mathbf{GR})$  can be saved as  $\mathbf{H}$  and  $\mathbf{IGR}$  in the memory, respectively, to further reduce the computational complexity and latency. The matrix storage format consists of two RAMs defined to store the pre-calculated transformer matrices: one is a 32-bit wide RAM *val* to store the value of the matrix entities in row order; the second is a 9-bit wide RAM, consisting of the last 8-bit *cid* to store the column index of the entries, and the first bit *rlb* to label all entities in the same row with '0' or '1'. Fig. 3.1 shows an example matrix and its storage format.

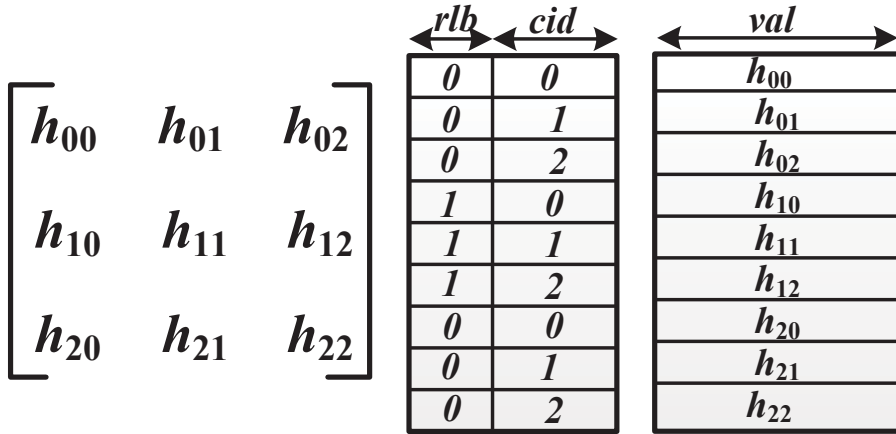


Figure 3.1: An example matrix and its storage format.

In Fig. 3.2(a), the RAMs (*rmHcol*, *rmH*, *rmIGRcol*, *IGR*) containing the transformer parameters are generated by the above rule, and *rmV* stores the transformer nodal voltages calculated by the nonlinear solver, while *rmIh* stores the history current terms calculated in previous time-step. When the linear module executes the updated function, the memories mentioned above are accessed sequentially by *adHcol*, *adH*, *adIGRcol*, and *adIGR*, and the data is sent to MVM function unit. The timing diagram of this operation is shown in Fig. 3.2(b).

The MVM unit consists of one floating point multiplier, one floating-to-fixed point converter, one fixed-floating point converter and two fixed point adders, as shown in Fig

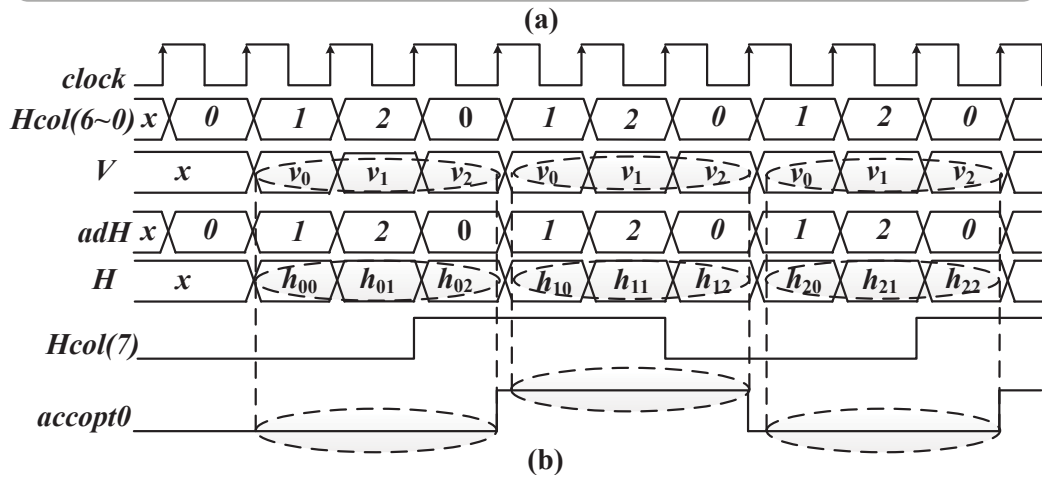
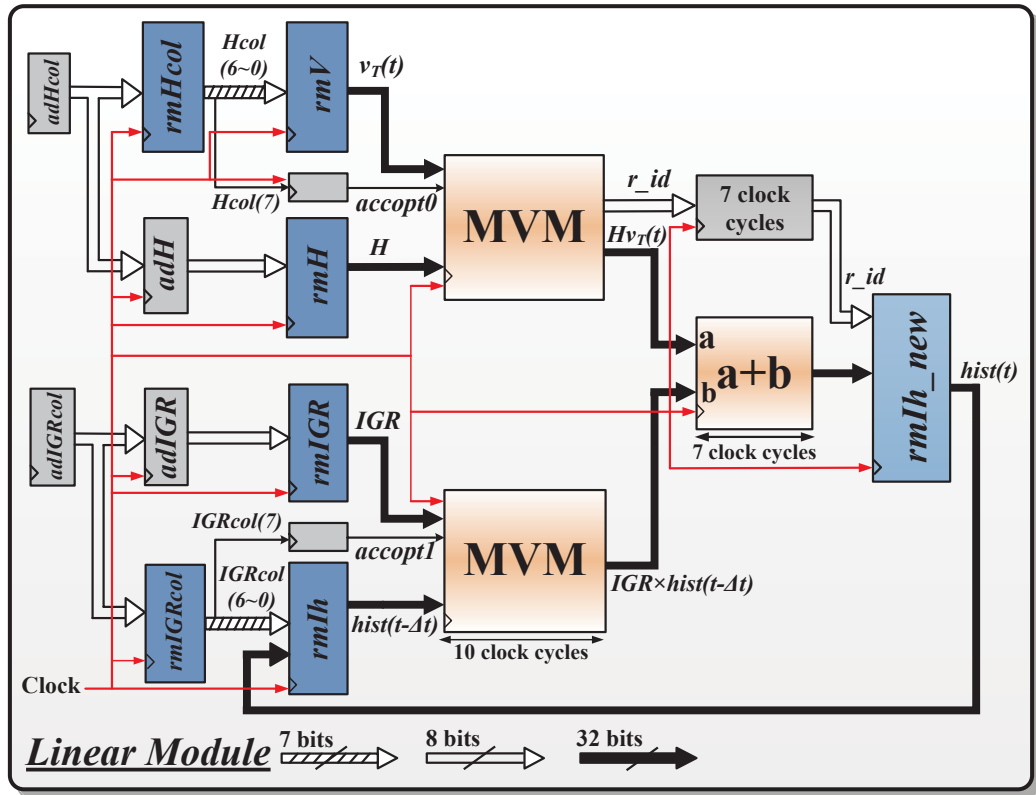


Figure 3.2: Pipelined scheme for the linear module: (a) hardware design; (b) timing diagram.

3.3(a). The reason for a fixed point adder in this module is that the floating point adder needs much more logic resources and a higher clock latency, however the fixed-point adder only needs one clock-cycle latency. The inserted registers in this unit are used for synchronization purpose. The two adders here with opposite reset signals are for pipelining purpose, so that there is no stall between two row-vector multiplications. Fig 3.3(b) show the logical timing diagram of these operations.

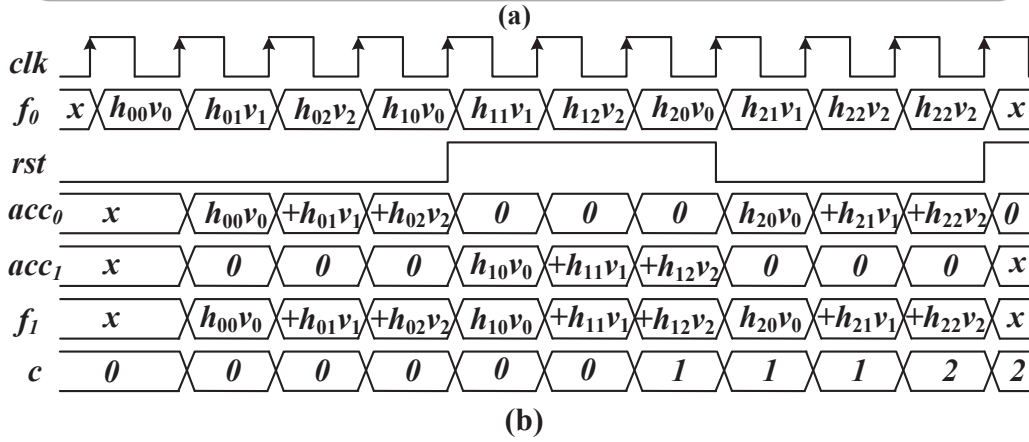
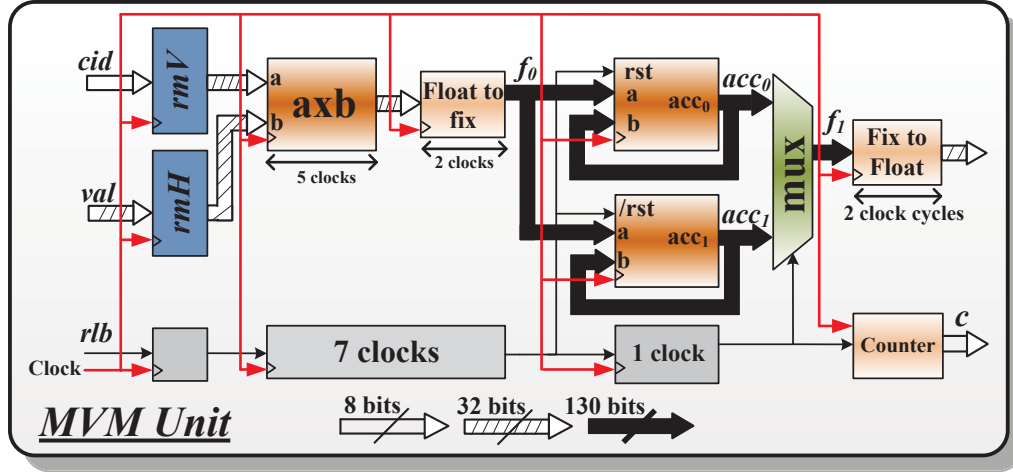


Figure 3.3: MVM unit: (a) hardware design; (b) timing diagram.

### 3.2 Nonlinear Model

Inclusion of nonlinearities is necessary for accurate modeling of transformer behaviour. To achieve that, nonlinear elements are attached externally at the linear module terminals. In most magnetic materials, the saturation phenomena can be seen in the continuous magnetization curve: as the current increases, the flux value increases to a maximum value asymptotically. While saturation is independent of the history current value, the hysteresis effect is dependent on history meaning that the flux cannot be determined by instant current value; however it can be determined given the history current value. A major hysteresis loop (Fig. 3.4) is the outermost loop which enters the saturation region at the reversal points (1-2), while other closed loops are considered minor loops (3-4 and 5-6). Furthermore, saturation is dominant when current has increased to a large value, while hysteresis plays an important role when the transformer operates with a small value of current.

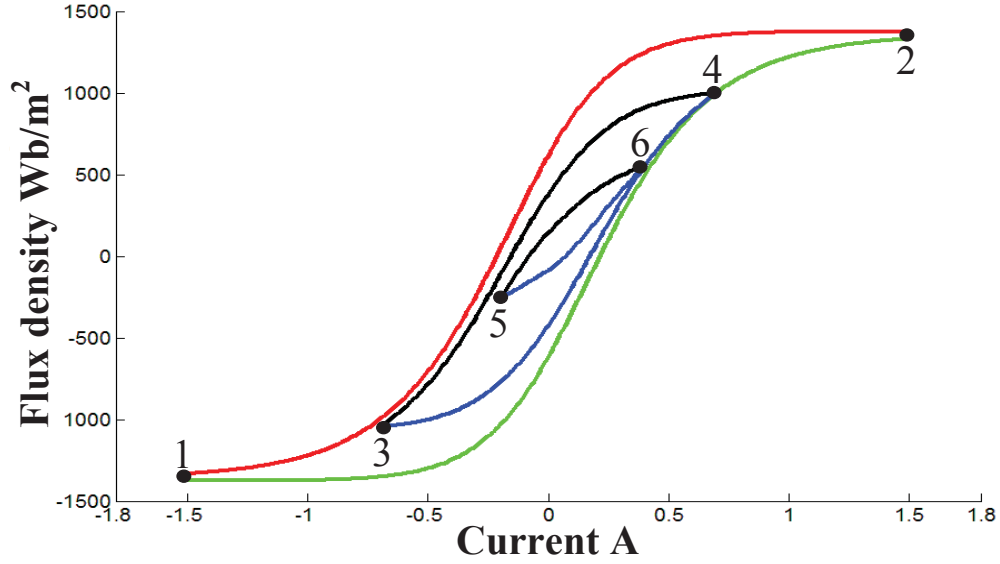


Figure 3.4: Hysteresis characteristic with major and minor loops.

### 3.2.1 Compensation Method and Newton-Raphson Iterations

By externally connecting a nonlinear element, the power system can be divided into two parts: linear part and nonlinear part, as shown in Fig. 3.5 (a). For the linear part, in an  $n$  node power system, the nodal analysis method is employed as (3.20):

$$\mathbf{Y}v_0 = \mathbf{i}, \quad (3.20)$$

where  $\mathbf{Y}$  is a  $n \times n$  admittance matrix without nonlinear elements,  $v_0$  and  $\mathbf{i}$  are  $n \times 1$  unknown nodal voltage and vector of nodal current terms respectively, expressed as,  $v_0 = [v_{01}(t) \ v_{02}(t) \ \dots \ v_{0n}(t)]^T$ , and  $\mathbf{i} = [i_1(t) \ i_2(t) \ \dots \ i_n(t)]^T$ .

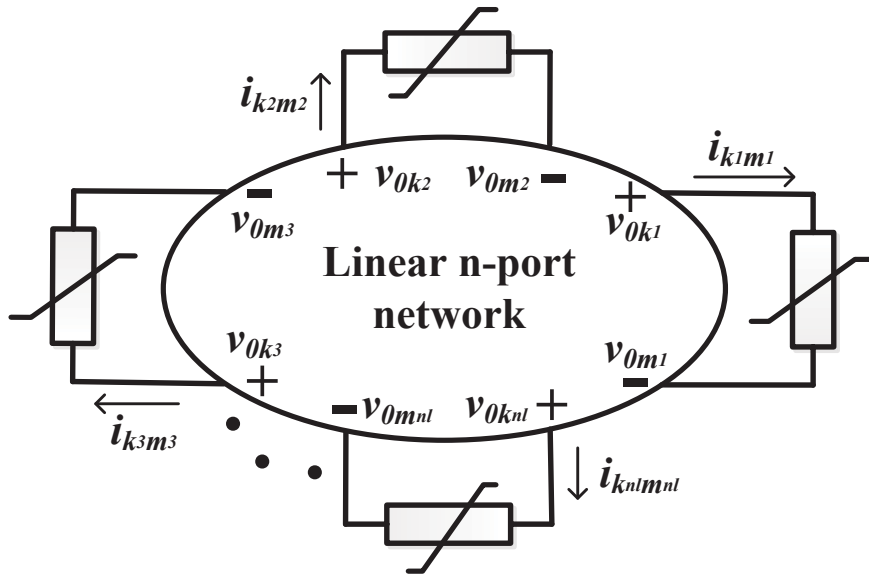
For the solution of the nonlinear part, the compensation method is employed. After the linear solution of a network is found, an  $n$ -node power system with  $n_l$  nonlinear elements is solved as illustrated in Fig. 3.5 (b) and (3.21).

$$\mathbf{f}(\mathbf{i}_{km}) = \mathbf{v}_{open} - \mathbf{r}_{thcv} \mathbf{i}_{km}, \quad (3.21)$$

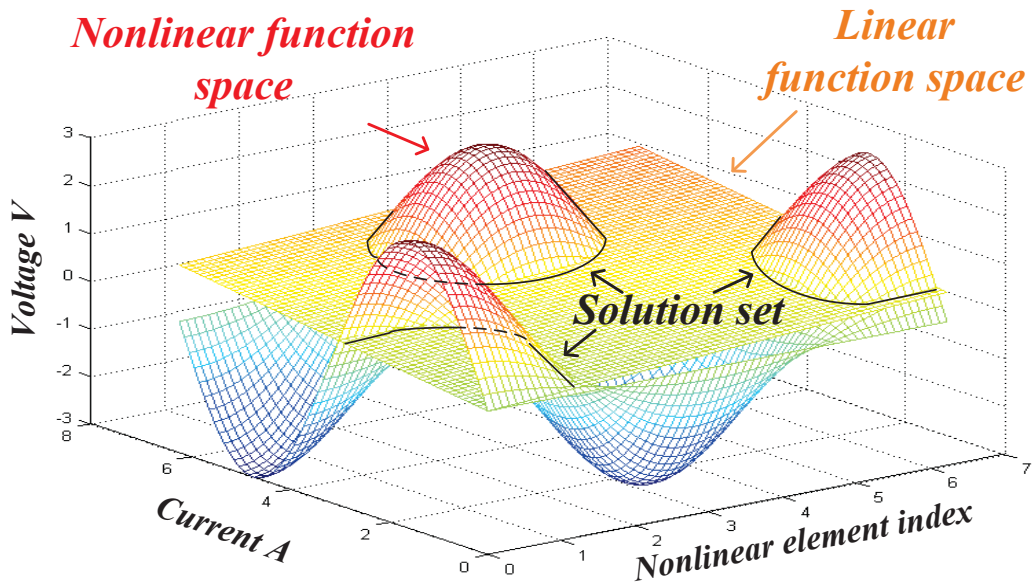
where  $\mathbf{v}_{open}$  is the  $n_l \times 1$  open circuit voltage vector (a subset of  $v_0$ ) without the nonlinear elements, and  $\mathbf{i}_{km}$  is the nonlinear elements' current injection vector with  $k_l$  and  $m_l$  ( $l = 1, 2, \dots, n_l$ ) being the node indices which the nonlinear elements are connected, expressed as:

$$\mathbf{v}_{open} = [v_{0k_1}(t) - v_{0m_1}(t) \ v_{0k_2}(t) - v_{0m_2}(t) \ \dots \ v_{0k_{n_l}}(t) - v_{0m_{n_l}}(t)]^T, \quad (3.22)$$

$$\mathbf{i}_{km} = [i_{k_1 m_1}(t) \ i_{k_2 m_2}(t) \ \dots \ i_{k_{n_l} m_{n_l}}(t)]^T. \quad (3.23)$$



(a)



(b)

Figure 3.5: (a) Linear network with external nonlinear elements; (b) Compensation method demonstration for multiple nonlinearities.

$\mathbf{r}_{thv}$  is the  $n_l \times n_l$  Thevenin equivalent resistance matrix that can be found from  $\mathbf{Y}$ , and  $f(i_{km})$  are nonlinear functions. The intersection of linear space and nonlinear space is the set of solution  $i_{km}$  as shown in Fig. 3.5 (b). In the proposed module, these nonlinear functions are generated by either a five-segments piece-wise nonlinear function to represent the saturation effect, or the Preisach model as detailed in the next section to represent both of the saturation and the hysteresis effect. (3.21) is solved simultaneously for  $i_{km}$  using the

Newton-Raphson iteration. Linearizing (3.21) using Newton-Raphson (N-R) results in the following equation:

$$\mathbf{J}(\mathbf{i}_{km}^{j+1} - \mathbf{i}_{km}^j) = -\mathbf{F}(\mathbf{i}_{km}^j), \quad (3.24)$$

where  $\mathbf{J}$  is Jacobian matrix,  $\mathbf{i}_{km}^{j+1}$  and  $\mathbf{i}_{km}^j$  are the current vectors at the  $(j+1)$ th and  $j$ th iterations, respectively. The  $\mathbf{J}$  and  $\mathbf{F}$  can be computed as:

$$\begin{cases} \mathbf{J} = \frac{\partial \mathbf{F}(\mathbf{i}_{km}^j)}{\partial \mathbf{i}_{km}^j} = \mathbf{r}_{thev} + \frac{\partial \mathbf{f}(\mathbf{i}_{km}^j)}{\partial \mathbf{i}_{km}^j} \\ -\mathbf{F} = \mathbf{v}_{open} - \mathbf{r}_{thev} \mathbf{i}_{km} - \mathbf{f}(\mathbf{i}_{km}^j) \end{cases} \quad (3.25)$$

The convergence conditions are defined as (3.26) with small enough values of  $\epsilon_1$  and  $\epsilon_2$ ,

$$\|\mathbf{i}_{km}^{j+1} - \mathbf{i}_{km}^j\| < \epsilon_1 \quad \text{and} \quad \|\mathbf{F}(\mathbf{i}_{km}^j)\| < \epsilon_2. \quad (3.26)$$

The final network solution is found by superimposing the response of  $\mathbf{i}_{km}$  on the linear network solution,

$$\mathbf{v} = \mathbf{v}_0 - \mathbf{R}_{Thev} \mathbf{i}_{km}, \quad (3.27)$$

where  $\mathbf{v}$  is a  $n \times 1$  node voltage vector with nonlinear branch, expressed as  $\mathbf{v} = [v_1 \ v_2 \ \dots \ v_n]^T$  and  $\mathbf{R}_{Thev}$  is a  $n \times n_l$  matrix obtained from  $\mathbf{Y}$ .

For example, for the nonlinear inductance, the characteristic function is provided in the form,

$$\lambda(t) = \mathbf{f}_\lambda(\mathbf{i}_{km}(t)), \quad (3.28)$$

where the flux  $\lambda(t)$  is the integral of node voltage over a time-step:

$$\lambda(t) = \lambda(t - \Delta t) + \int_{t-\Delta t}^t \mathbf{v}'(u) du. \quad (3.29)$$

Applying Trapezoidal rule results in:

$$\lambda(t) = \frac{\Delta t}{2} \mathbf{v}'(t) + \lambda_{hist}(t - \Delta t), \quad (3.30)$$

where

$$\lambda_{hist}(t - \Delta t) = \lambda(t - \Delta t) + \frac{\Delta t}{2} \mathbf{v}'(t - \Delta t). \quad (3.31)$$

From (3.30) and (3.28) the set of nonlinear equations to determine  $\mathbf{i}_{km}$  are,

$$\mathbf{F}_\lambda \equiv \frac{2}{\Delta t} [\mathbf{f}_\lambda(\mathbf{i}_{km}(t)) - \lambda_{hist}(t - \Delta t)] - \mathbf{v}_{open}(t) + \mathbf{r}_{thev} \mathbf{i}_{km}(t), \quad (3.32)$$

Then, the Jacobian matrix and the nonlinear equations for this case can be rewritten as:

$$\begin{cases} \mathbf{J}_\lambda = \frac{\partial \mathbf{F}_\lambda(\mathbf{i}_{km}^j(t))}{\partial \mathbf{i}_{km}^j(t)} = \mathbf{r}_{thev} + \frac{\partial \mathbf{f}_\lambda(\mathbf{i}_{km}^j(t))}{\partial \mathbf{i}_{km}^j(t)} \frac{2}{\Delta t} \\ -\mathbf{F}_\lambda = \mathbf{v}_{open}(t) - \mathbf{r}_{thev} \mathbf{i}_{km}^j(t) - \\ \quad [\mathbf{f}_\lambda(\mathbf{i}_{km}^j(t)) - \lambda_{hist}(t - \Delta t)] \frac{2}{\Delta t}. \end{cases} \quad (3.33)$$

### 3.2.2 Preisach Hysteresis Model

In order to include the hysteresis phenomenon, the Preisach model is employed to define the hysteretic loop. The major loop hysteresis function is geometrically represented by the following function [53]:

$$\lambda_{p.m}(\mathbf{i}_{km}) = \frac{1}{2} \sum_{u=1}^3 a_u [\tanh(b_u \mathbf{i}_{km}) + c_u \operatorname{sech}^2(b_u \mathbf{i}_{km})], \quad (3.34)$$

where  $a_u, b_u, c_u$  are the parameters to define this hyperbolic template,  $\mathbf{i}_{km}$  is current value vector flowing through the nonlinear inductances as before and  $\lambda_{p.m} = [\lambda_{p.m1} \ \lambda_{p.m2} \ \dots \ \lambda_{p.mn_l}]^T$  is the flux density corresponding to this current value vector on the major loop. The subscript 'p\_m' stand for Preisach major loop.

Once the major loop function is fixed, all the minor loop trajectories from a reversal point follow this uniform template. A reverse point is detected, when  $\lambda_{p.ml}(t + \Delta t) > \lambda_{p.ml}(t)$  on a downward trajectory, or when  $\lambda_{p.ml}(t + \Delta t) < \lambda_{p.ml}(t)$  on an upward trajectory ( $l = 1, 2, \dots, n_l$ ). In general, Preisach upward and downward trajectories from a reversal point  $(i_{rl}, \lambda_{rl})$  can be expressed as (3.35) and (3.36), respectively [54], where  $\mathbf{i}_r = [i_{r1} \ i_{r2} \ \dots \ i_{rn_l}]$  and  $\lambda_r = [\lambda_{r1} \ \lambda_{r2} \ \dots \ \lambda_{rn_l}]$ . The reversal points need to be recorded in a reversal point stack (RPS), while traveling the hysteresis loop; for example when flux density starts increasing from point 5, and reaches point 6, the trajectory beyond the reversal point 6 is defined by the template beginning at point 3, as shown in Fig 3.4. Thus, if the flux density increases or decreases out of the outer loop, the last two coordinates in the stack are removed. This example is initialized by saving the two major loop reversal points (1 and 2) in the stack; these two points cannot be removed since all the trajectories cannot be beyond the major loop as illustrated in Fig. 3.6. The trajectory is processed by going through the following reversal points: 1-4-3-6-5, with the corresponding stack shown in Fig. 3.6 (a). Once the trajectory increases beyond point 6, the reversal points 5 and 6 are removed due to the reason discussed above, shown in Fig 3.6 (b).

The flux density trajectories corresponding to the instantaneous current in a upward  $\lambda_{p.u}$  or downward  $\lambda_{p.d}$  minor loop are given as:

$$\lambda_{p.u}(\mathbf{i}_{km}) = -\lambda_{p.m}(-\mathbf{i}_{km}) - \lambda_{p.m}(\mathbf{i}_r) + \lambda_r + 2\mathbf{K}(-\mathbf{i}_r)\mathbf{K}(\mathbf{i}_{km}), \quad (3.35)$$

$$\lambda_{p.d}(\mathbf{i}_{km}) = \lambda_{p.m}(\mathbf{i}_{km}) + \lambda_{p.m}(-\mathbf{i}_r) + \lambda_r - 2\mathbf{K}(\mathbf{i}_r)\mathbf{K}(-\mathbf{i}_{km}), \quad (3.36)$$

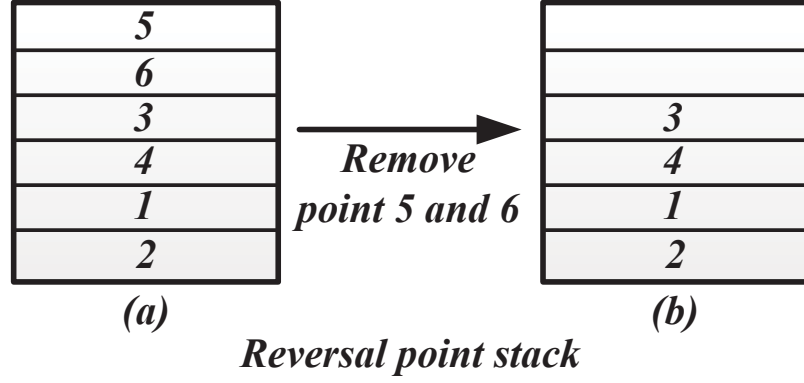


Figure 3.6: Corresponding reversal point stack before (a) and after (b) current increasing beyond the point 6.

where

$$\mathbf{K} = [K_1 \ K_2 \ \dots \ K_{n_l}],$$

$$K_l(x) = \begin{cases} \sqrt{\lambda_{p.m}(-x)}, & x < 0 \\ \frac{\lambda_{p.m}(-x) + \lambda_{p.m}(x)}{2\sqrt{\lambda_{p.m}(x)}}, & x > 0, \end{cases}$$

$$l = 1, 2, \dots, n_l.$$

The functions  $\frac{d}{d\mathbf{i}_{km}}\lambda_{p.u}(\mathbf{i}_{km})$ ,  $\frac{d}{d\mathbf{i}_{km}}\lambda_{p.d}(\mathbf{i}_{km})$  for upward and downward trajectories can be expressed as (3.37) and (3.38), respectively:

$$\frac{d}{d\mathbf{i}_{km}}\lambda_{p.u}(\mathbf{i}_{km}) = d\lambda_{p.m}(-\mathbf{i}_{km}) + 2\mathbf{K}(-\mathbf{i}_r)d\mathbf{K}(\mathbf{i}_{km}), \quad (3.37)$$

$$\frac{d}{d\mathbf{i}_{km}}\lambda_{p.d}(\mathbf{i}_{km}) = d\lambda_{p.m}(\mathbf{i}_{km}) + 2\mathbf{K}(\mathbf{i}_r)d\mathbf{K}(-\mathbf{i}_{km}). \quad (3.38)$$

Substituting (3.35) and (3.36) into (3.28), the Jacobian matrix can be computed by (3.33) using the differentials (3.37) and (3.38).

### 3.2.3 Nonlinear Module Hardware Design

Several hardware implementations are available, such as COordinate Rotation DIgital Computer (CORDIC), to evaluate the nonlinear functions required in the N-R iteration. However, since the number of iterations executed is unknown, the computational burden and latency could be too high to achieve real-time execution. In the Preisach Hysteretic Unit, look-up tables are employed instead to compute the nonlinear functions.

As shown in Fig. 3.7, taking the advantage of parallelism of logic resources, the function  $f(\mathbf{i}_{km})$  and  $\frac{d}{d\mathbf{i}_{km}}f(\mathbf{i}_{km})$  can be evaluated simultaneously. The address generator (AG) functional unit is used to convert the 32-bit floating-point  $\mathbf{i}_{km}$  into integer numbers to address the LUTs. (3.35) - (3.38) show that function  $f(\mathbf{i}_{km})$  and  $\frac{d}{d\mathbf{i}_{km}}f(\mathbf{i}_{km})$  can be evaluated



with  $i_{km}$  or  $-i_{km}$ ,  $i_r$  or  $-i_r$  depending on the trajectory direction. Therefore, before  $i_{km}$  is input into the AG unit, its trajectory direction must be known. Besides, since the function  $K(i_{km})$  is dependent on the sign of  $i_{km}$  (3.37), so that two  $K(i_{km})$  LUTs ( $K+$ ,  $K-$ ) are precalculated and the first bit of  $i_{km}$  (or  $-i_{km}$ ) determines which value is involved in the calculation by using a multiplexer  $mux$ .

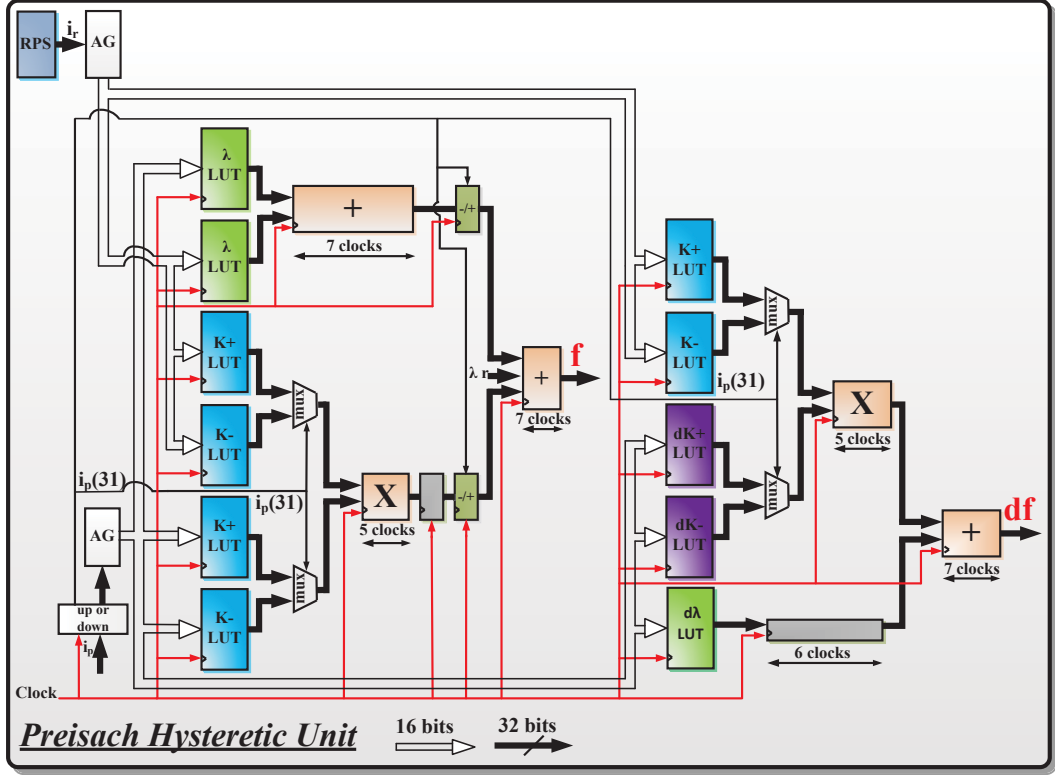


Figure 3.7: Preisach hysteretic unit (PHU) hardware design.

As shown in Fig. 3.8, after finding the values of  $f(i_{km})$  and  $\frac{d}{di_{km}} f(i_{km})$ , the Jacobian Matrix Compute Unit (JMCU) computes the values of  $-F$  and  $J$  as given in (3.33), and then the Gauss-Jordan Elimination unit (GJE) is used to solve the algebraic equations (3.24). If the calculated  $i_{km}$  is satisfy (3.26), then  $i_{km}$  is used to compensate the voltage  $v_0$  to find the final results  $v$  (3.27). If the  $i_{km}$  has not converged, then it is used as an input for the next iteration until convergence or the maximum number of iterations is reached.

### 3.3 Frequency-dependent Eddy Current Model

Eddy currents, a part of the transformer core losses, are induced by the change in the magnetic field. While they can be reduced by using a laminated core, previous works indicate that eddy current losses are frequency dependent. However, in transformer models used for transient studies, the eddy current model is normally a constant resistance in parallel with the magnetizing inductance so that this model is not valid over a wide frequency

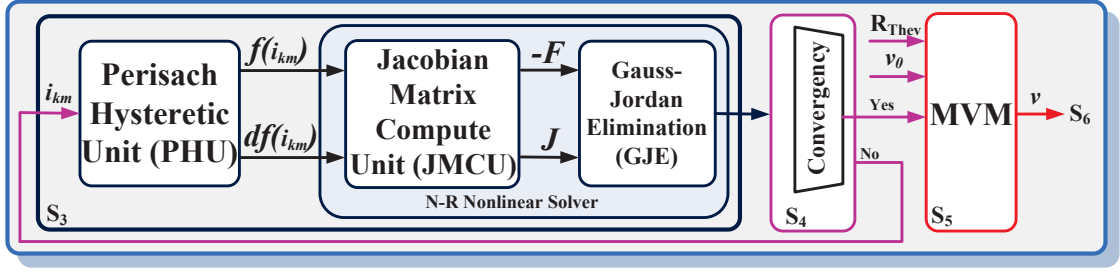


Figure 3.8: Overall nonlinear module hardware architecture.

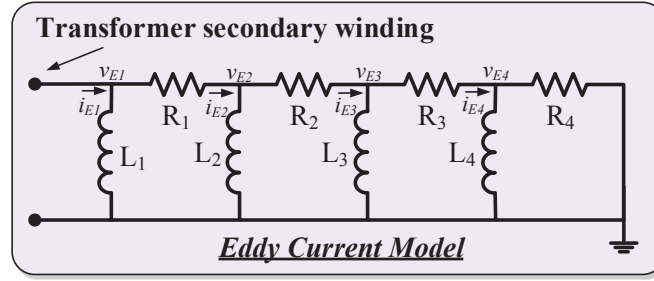


Figure 3.9: Cauer equivalent circuit for frequency dependent eddy current model.

range.

A Cauer equivalent circuit has been developed to represent the frequency dependency as a cascade of RL lumped sections (Fig. 3.9). The accuracy of this model over a wide frequency range depends on the number of RL sections. Four sections are required to represent the frequency up to 200kHz with an error of less than 5% [55]. The parameters of the Cauer model can be derived as,

$$\begin{aligned}
 L_0 &= \frac{N^2 A \mu}{l} \quad \text{and} \quad L_k = \frac{L_0}{(4k - 3)}, \\
 R_0 &= \frac{4N^2 A}{ld^2 \gamma} \quad \text{and} \quad R_k = R_0(4k - 1),
 \end{aligned} \tag{3.39}$$

where,  $\mu$  is the magnetic permeability of the steel lamination,  $\gamma$  is the electric conductivity of the steel lamination,  $A$  is the total cross-sectional area of all laminations,  $d$  is the thickness of the lamination,  $l$  is the length of core limb,  $N$  is the number of coil turns.

After discretizing the cascaded R-L equivalent circuit, the Norton equivalent formula is given as (3.40), where  $\mathbf{i}_E(t) = [i_{E1}(t) \ i_{E2}(t) \ i_{E3}(t) \ i_{E4}(t)]^T$  and  $\mathbf{v}_E(t) = [v_{E1}(t) \ v_{E2}(t) \ v_{E3}(t) \ v_{E4}(t)]^T$  are the node current and voltage vectors, respectively, and  $\mathbf{I}_{histE}(t)$  is the eddy current history term vector which is updated by (3.41):

$$\mathbf{i}_E(t) = \mathbf{Y}_E \mathbf{v}_E(t) + \mathbf{I}_{histE}(t). \tag{3.40}$$

$$\mathbf{I}_{histE}(t) = \mathbf{G}_E \mathbf{v}_E(t - \Delta t) + \mathbf{I}_{histE}(t - \Delta t), \tag{3.41}$$

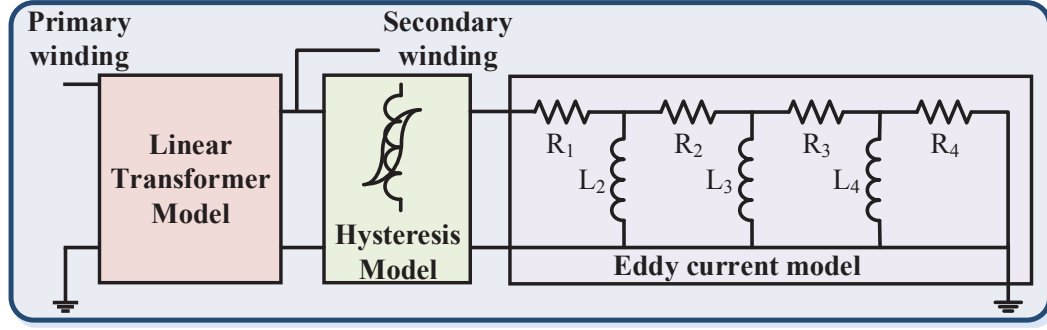


Figure 3.10: The overall frequency-dependent nonlinear hysteretic transformer model.

where,

$$\mathbf{G}_E = \text{diag} \left\{ \frac{\Delta t}{2L_1}, \frac{\Delta t}{2L_2}, \frac{\Delta t}{2L_3}, \frac{\Delta t}{2L_4} \right\}. \quad (3.42)$$

Since the eddy current model only consists of resistor and inductor components, so that the hardware emulation can be done by Passive Elements Module as described later in Section 3.7.

### 3.4 The Overall Transformer Model

The overall frequency-dependent nonlinear hysteretic transformer model is shown as Fig. 3.10. The admittance based linear transformer model is connected externally with a hysteresis model and eddy current model on the secondary side winding. As can be seen in eddy current model, the linear inductor in the first RL lumped section is replaced by the hysteresis model.

### 3.5 Bergeron Transmission Line Model

#### 3.5.1 Bergeron Model Formulation

The Bergeron line model is a discrete-time traveling wave model to represent the transmission line using distributed L and C parameters. Although, the high frequency behavior of the line is not well represented, this model accurately represents the real world propagation delay on the lines. The governing equations to describe this model are given as:

$$\begin{aligned} I_{k,m}(t) &= \frac{V_k(t)}{Z_c} + I_k(t - \tau), \\ I_{m,k}(t) &= \frac{V_m(t)}{Z_c} + I_m(t - \tau), \end{aligned} \quad (3.43)$$

where  $V_k(t)$ ,  $V_m(t)$ ,  $I_{k,m}(t)$  and  $I_{m,k}(t)$  are sending-end ('k') and receiving-end ('m') voltages and currents, respectively;  $I_k(t - \tau)$  and  $I_m(t - \tau)$  are the injection current terms at

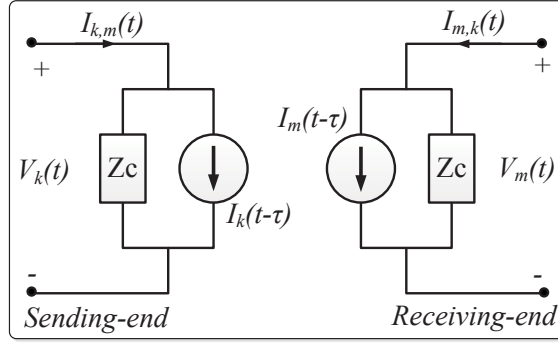


Figure 3.11: Discrete-time Norton equivalent circuit of the Bergeron line model.

sending-end and receiving-end respectively, given as

$$\begin{aligned} I_k(t - \tau) &= -\frac{2V_m(t - \tau)}{Z_c} - I_m(t - 2\tau), \\ I_m(t - \tau) &= -\frac{2V_k(t - \tau)}{Z_c} - I_k(t - 2\tau); \end{aligned} \quad (3.44)$$

$Z_c$  is the line characteristic impedance given as

$$Z_c = \sqrt{\frac{L}{C}}, \quad (3.45)$$

$\tau$  is the travel time taken from sending-end (' $k$ ') to the receiving-end (' $m$ '), given as

$$\tau = l\sqrt{LC}, \quad (3.46)$$

where  $l$ ,  $L$  and  $C$  are the length, series inductance and shunt capacitance of the transmission line, respectively.

The discrete-time Norton equivalent circuit for Bergeron line model is shown in Fig. 3.11. The inputs for this discretized model are the sending-end and receiving-end currents ( $I_{k,m}(t)$  and  $I_{m,k}(t)$ ), injection history terms ( $I_k(t-\tau)$  and  $I_m(t-\tau)$ ) and the line impedance, while the outputs are the voltages on the sending-end and receiving-end ( $V_k(t)$  and  $V_m(t)$ ).

For three-phase power system, the modal domain impedance matrix  $\mathbf{Z}_{c,modal}$  and injection history terms vectors  $\mathbf{I}_{k,modal}(t-\tau)$  and  $\mathbf{I}_{m,modal}(t-\tau)$  are built up as three separate systems as following,

$$\mathbf{Z}_{c,modal} = \begin{bmatrix} Z_{c,a} & 0 & 0 \\ 0 & Z_{c,b} & 0 \\ 0 & 0 & Z_{c,c} \end{bmatrix}, \quad (3.47)$$

$$\mathbf{I}_{k,modal}(t - \tau) = \begin{bmatrix} I_{k,a}(t - \tau) \\ I_{k,b}(t - \tau) \\ I_{k,c}(t - \tau) \end{bmatrix}, \quad (3.48)$$

$$\mathbf{I}_{m,modal}(t - \tau) = \begin{bmatrix} I_{m,a}(t - \tau) \\ I_{m,b}(t - \tau) \\ I_{m,c}(t - \tau) \end{bmatrix}, \quad (3.49)$$

$$\mathbf{V}_{k,modal}(t) = \begin{bmatrix} V_{k,a}(t) \\ V_{k,b}(t) \\ V_{k,c}(t) \end{bmatrix}, \quad (3.50)$$

$$\mathbf{V}_{m,modal}(t) = \begin{bmatrix} V_{m,a}(t) \\ V_{m,b}(t) \\ V_{m,c}(t) \end{bmatrix}, \quad (3.51)$$

where  $a, b$  and  $c$  stand for phase  $a, b$  and  $c$  respectively.

The phase-domain transmission line impedance matrix,  $\mathbf{Z}_{c,phase}$ , and injection history terms vectors,  $\mathbf{I}_{k,phase}(t - \tau)$  and  $\mathbf{I}_{m,phase}(t - \tau)$ , are transferred from the modal domain by “Karrenbauer” transformation as

$$\mathbf{Z}_{c,phase} = \mathbf{S}\mathbf{Z}_{c,modal}\mathbf{S}^{-1}, \quad (3.52)$$

$$\mathbf{I}_{k,phase}(t - \tau) = \mathbf{S}\mathbf{I}_{k,modal}(t - \tau), \quad (3.53)$$

$$\mathbf{I}_{m,phase}(t - \tau) = \mathbf{S}\mathbf{I}_{m,modal}(t - \tau). \quad (3.54)$$

The phase-domain transmission line the sending-end and receiving-end voltage vectors,  $\mathbf{V}_{k,phase}$  and  $\mathbf{V}_{m,phase}$ , are decomposed into three separate systems as,

$$\mathbf{V}_{k,phase}(t) = \mathbf{S}\mathbf{V}_{k,modal}(t), \quad (3.55)$$

$$\mathbf{V}_{m,phase}(t) = \mathbf{S}\mathbf{V}_{m,modal}(t), \quad (3.56)$$

where the transformation matrix  $\mathbf{S}$  is given by

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}. \quad (3.57)$$

### 3.5.2 Real-time FPGA Hardware Implementation of Begeron Model

A hardware real-time module **B-TLM** is built on FPGA to emulate the Begeron model as shown in Fig. 3.12 with the input and output signals. This implementation is used to

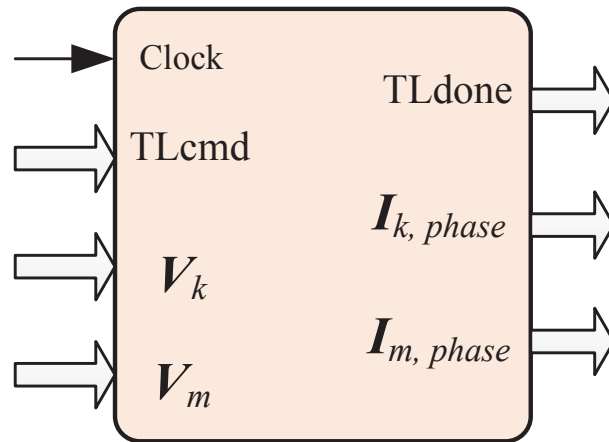


Figure 3.12: **B-TLM** module hardware design.

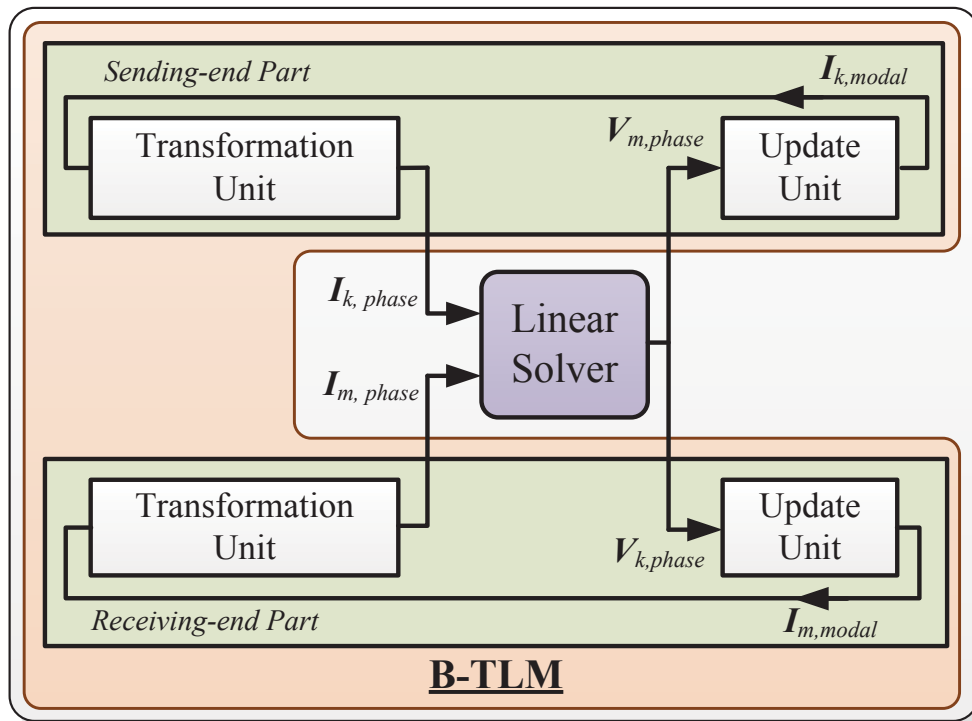


Figure 3.13: Paralleled **Transformation Unit** and a **Update Unit** in the **B-TLM**.

perform the “Karrenbauer” transformation by (3.53) - (3.56) and update the history terms by (3.44). As can be seen from (3.43) and (3.43), the calculations at sending-end and receiving-end can be computed separately, which means in hardware emulation they can be processed fully in parallel to achieve minimum latency. As shown in Fig. 3.13 within the **B-TLM** module, two identical hardware parts are implemented, and each part contains one **Transformation Unit** and one **Update Unit**.

### 3.5.2.1 Transformation Unit

The “Karrenbauer” transformation is performed in this unit to calculate the history terms under phase domain,  $I_{k,phase}$  and  $I_{m,phase}$ , by reading the modal domain history terms,  $I_{k,modal}$  and  $I_{m,modal}$ . The transformed phase domain history terms are then sent to linear solver module to perform nodal analysis method. A floating-point 6-input multiply-add unit is implemented for the transformation purpose. The RAMs are used to write in and read out the modal domain history terms and ROM are used to store the matrix of  $S$ .

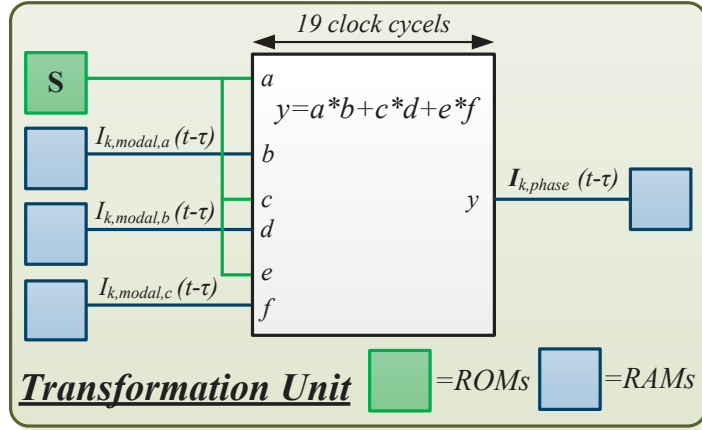


Figure 3.14: Pipelined computation scheme in the **Transformation Unit**.

### 3.5.2.2 Update Unit

After obtaining the line sending-end and receiving-end voltages from the linear solver module, the **Update Unit** is executed to update the history terms in modal domain. The first step is to transform the voltages from phase domain to modal domain which is achieved by a floating-point 6-input multiply-add unit. The second step is to update the history terms in modal domain which is completed by a floating-point 4-input multiply-add unit. The RAMs are used to write in and read out the phase domain voltages or modal domain history terms and ROM are used to store the entries of matrix  $S^{-1}$ .

## 3.6 Source Module

In this FPGA based real-time EMT hardware emulation, an ideal sinusoidal voltage source module is implemented. A voltage source can be expressed as

$$\begin{aligned}
 v(t) &= MAG * \cos(\omega t + \phi) \\
 &= MAG * \cos(\omega n \Delta t + \phi) \\
 &= MAG * \cos(\omega \Delta t + \omega(n-1)\Delta t + \phi) \\
 &= MAG * \cos(\omega \Delta t + \theta(t - \Delta t)),
 \end{aligned} \tag{3.58}$$

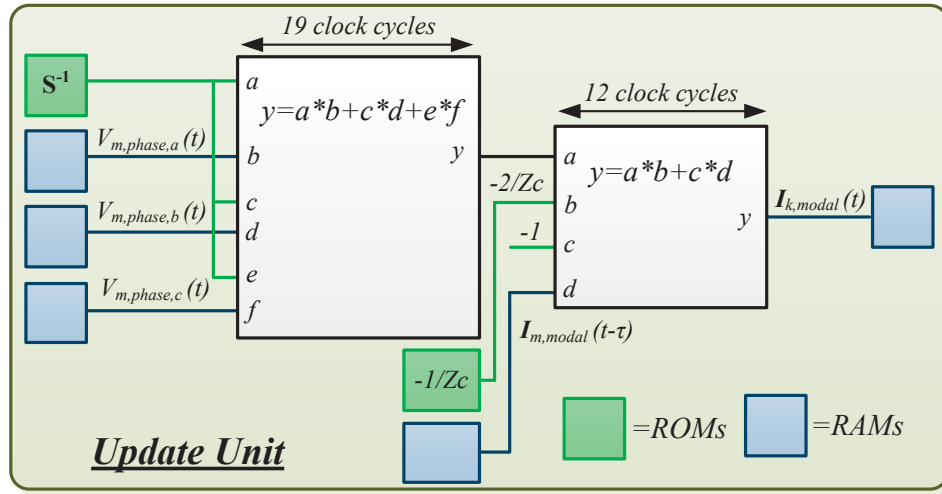


Figure 3.15: Pipelined computation scheme in the **Update Unit**.

where  $MAG$ ,  $\omega$ ,  $\phi$  and  $\theta(t - \Delta t)$  are the source magnitude, frequency, initial phase angle and previous time-step phase angle, respectively. The most challenge aspect to model an ideal voltage source in FPGA is how to properly calculate the nonlinear  $\cos$  function. The look-up table (LUT) is used to evaluate this nonlinear function and only half cycle (0 to  $\pi$ ) of values are needed and stored in the LUT since the  $\cos$  is a periodic function. The accuracy is determined by the depth of the LUT. The hardware implementation of **Source** module is given in Fig. 3.16. As can be seen in each time-step, the new phase angle is first updated by adding previous phase angle by  $\omega\Delta t$ . The new phase angle needs to be checked with  $\pi$ , if it greater than  $\pi$ , the value is subtracted by  $\pi$  and invert the sign. Then the new phase angle is accessed by LUT to evaluate the  $\cos$  function values which is finally multiplied by magnitude  $V_{mag}$  to obtain the voltage source values,  $v_s$ .

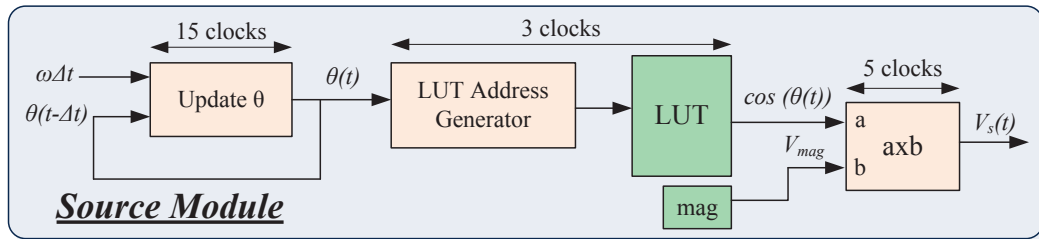


Figure 3.16: Hardware implementation of **Source** module.

## 3.7 Passive Elements Module

### 3.7.1 Model Formulation

In the EMTP, the passive elements, such as resistor, inductor and capacitor are discretized by Trapezoidal rule. After discretization in each simulation time-step, the passive elements



can be represented by a equivalent resistance paralleled with a history current source which can be carried out in previous time-step simulation.

### 3.7.1.1 Resistance Element $R$

The resistance  $R$  discretized-time model is given in Fig. 3.17, as can be seen which is the same as its continuous-time model and there is no history current term. The voltage and current relationship can be expressed as

$$v_{km}(t) = Ri_{km}(t). \quad (3.59)$$

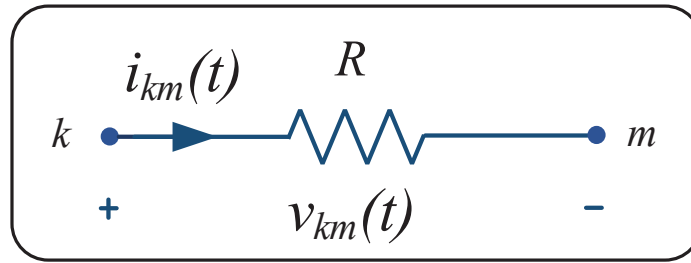


Figure 3.17: A resistance  $R$  element and its discrete-time model.

### 3.7.1.2 Inductance Element $L$

The inductance  $L$  differential equation is given as

$$v_{km}(t) = L \frac{di_{km}(t)}{dt}, \quad (3.60)$$

and its continuous-time model is shown in Fig. 3.18 (a). Integrating (3.60) from time  $(t - \Delta t)$  to  $t$  yields

$$i_{km}(t) - i_{km}(t - \Delta t) = \frac{1}{L} \int_{t-\Delta t}^t v_{km}(t) dt. \quad (3.61)$$

Applying Trapezoidal rule to (3.61) results

$$\begin{aligned} i_{km}(t) &= i_{km}(t - \Delta t) + \frac{\Delta t}{2L} [v_{km}(t) + v_{km}(t - \Delta t)] \\ &= \frac{\Delta t}{2L} v_{km}(t) + [i_{km}(t - \Delta t) + \frac{\Delta t}{2L} v_{km}(t - \Delta t)] \\ &= \frac{v_{km}(t)}{R_{eq}} + I_h(t - \Delta t). \end{aligned} \quad (3.62)$$

The discretized-time model then as shown in Fig. 3.18 (b) can be represented by a equivalent resistance  $R_{eq}$  in paralleled with a history current source  $I_h(t - \Delta t)$ . The equivalent resistance is given as

$$R_{eq} = \frac{2L}{\Delta t}, \quad (3.63)$$

and the history current source is calculated from the previous simulation time-step solution, expressed as

$$I_h(t - \Delta t) = [i_{km}(t - \Delta t) + \frac{\Delta t}{2L}v_{km}(t - \Delta t)], \quad (3.64)$$

or recursively using

$$\begin{aligned} I_h(t - \Delta t) &= I_h(t - 2\Delta t) + \frac{\Delta t}{L}v_{km}(t - \Delta t) \\ &= I_h(t - 2\Delta t) + \frac{2}{R_{eq}}v_{km}(t - \Delta t). \end{aligned} \quad (3.65)$$

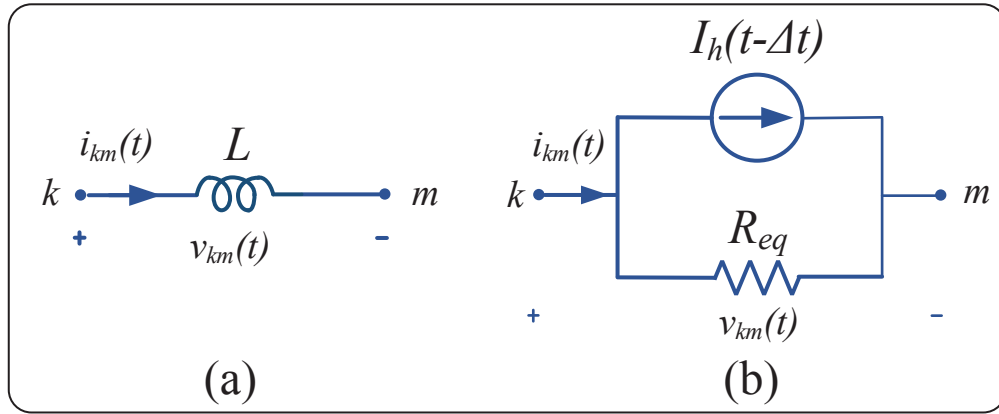


Figure 3.18: (a) An inductance  $L$  element; (b) its discrete-time Norton equivalent circuit.

### 3.7.1.3 Capacitance Element $C$

The capacitance  $C$  differential equation is given as

$$i_{km}(t) = C \frac{dv_{km}(t)}{dt}. \quad (3.66)$$

and its continuous-time model is shown in Fig. 3.19 (a). Integrating (3.66) from time  $(t - \Delta t)$  to  $t$  yields

$$v_{km}(t) - v_{km}(t - \Delta t) = \frac{1}{C} \int_{t-\Delta t}^t i_{km}(t) dt. \quad (3.67)$$

Applying Trapezoidal rule to (3.67) results

$$\begin{aligned} i_{km}(t) &= -i_{km}(t - \Delta t) + \frac{2C}{\Delta t} [v_{km}(t) + v_{km}(t - \Delta t)] \\ &= \frac{2C}{\Delta t} v_{km}(t) + [-i_{km}(t - \Delta t) - \frac{2C}{\Delta t} v_{km}(t - \Delta t)] \\ &= \frac{2C}{\Delta t} v_{km}(t) + I_h(t - \Delta t). \end{aligned} \quad (3.68)$$

The discretized-time model then as shown in Fig. 3.19 (b) can be represented by a equivalent resistance  $R_{eq}$  in paralleled with a history current source  $I_h(t - \Delta t)$ . The equivalent

resistance is given as

$$R_{eq} = \frac{\Delta t}{2C} \quad (3.69)$$

and the history current source is calculated from the previous simulation time-step solution, expressed as

$$I_h(t - \Delta t) = -i_{km}(t - \Delta t) - \frac{2C}{\Delta t} v_{km}(t - \Delta t), \quad (3.70)$$

or recursively using

$$\begin{aligned} I_h(t - \Delta t) &= -I_h(t - 2\Delta t) - \frac{4C}{\Delta t} v_{km}(t - \Delta t) \\ &= -I_h(t - 2\Delta t) - \frac{2}{R_{eq}} v_{km}(t - \Delta t). \end{aligned} \quad (3.71)$$

In order to emulate the passive elements in the EMT real-time simulation, a general for-

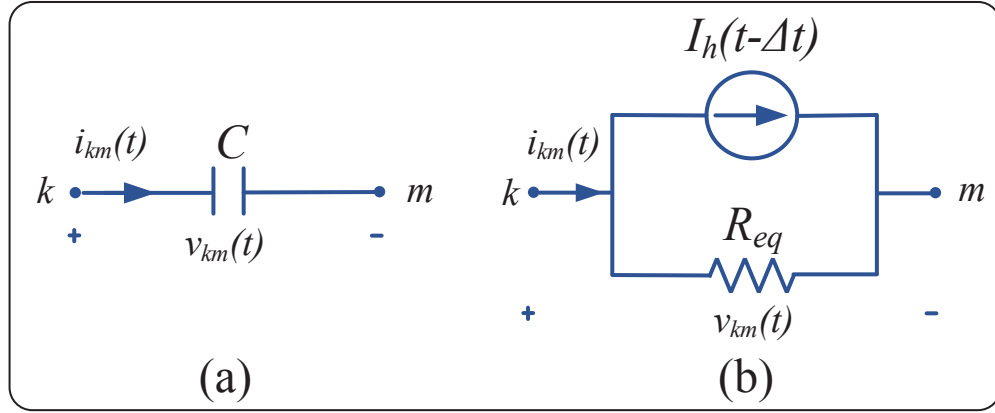


Figure 3.19: (a) A capacitance  $C$  element; (b) its discrete-time Norton equivalent circuit.

mula is needed to update all the three types of elements history current term. As can be seen from (3.65) and (3.71), the history current term updating equations follow the same format

$$I_h(t - \Delta t) = P_1 I_h(t - 2\Delta t) + P_2 v_{km}(t - \Delta t), \quad (3.72)$$

where  $P_1$  and  $P_2$  are the parameters to define the different elements which is given in Table 3.1. For resistance model  $R$ , since there is no history current term in paralleled with the equivalent resistance, both parameters  $P_1$  and  $P_2$  are zero.

### 3.7.2 Real-time Hardware Module for Passive Elements Module

A real-time hardware module is implemented to emulate the passive elements (**RLC**) on FPGA as shown in Fig. 3.20. The pre-calculated parameters  $R_{eq}$ ,  $P_1$  and  $P_2$  are saved in RAMs, and are sent to involve in the computation when this module is activated by the controller. In paralleled, besides the passive element history current terms, this module can also carry out the branch current  $i_{km}(t)$  in (3.62) and (3.68).

Table 3.1: Parameters for updating RLC elements history currents (3.72)

RLC elements	$R_{eq}(1/G_{eq})$	$P_1$	$P_2$
R	R	0	0
L	$\frac{2L}{\Delta t}$	1	$\frac{\Delta t}{L}$
C	$\frac{\Delta t}{2C}$	-1	$-\frac{4C}{\Delta t}$

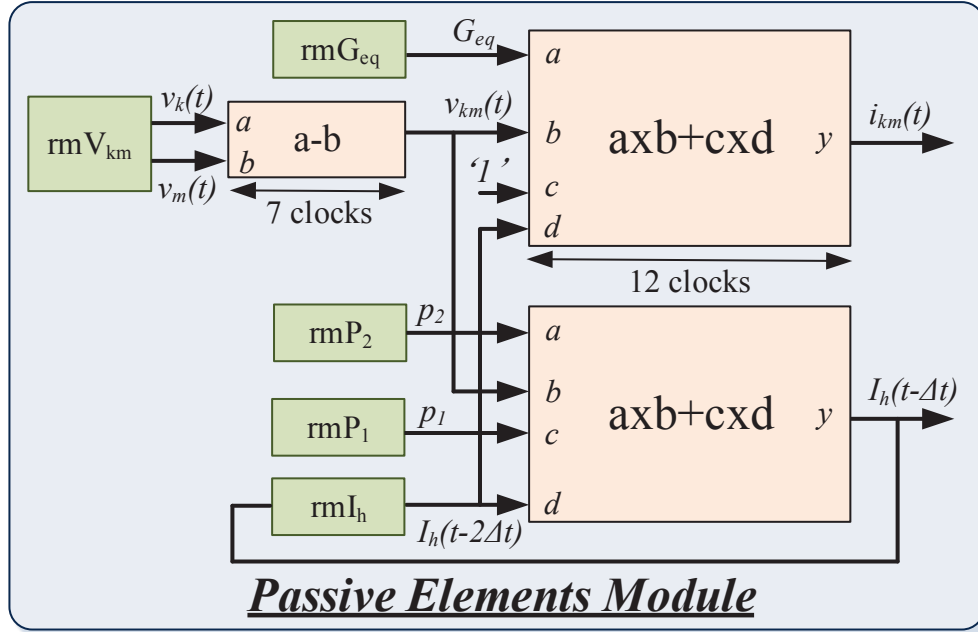


Figure 3.20: Hardware implementation for **RLC** module.

### 3.8 Inode Unit Module

After calculating the history currents, a **Inode** Unit Module is used to form the history current vector. For a specific node, more than one history currents can be connected to this node as shown in Fig. 3.21(a). Therefore, the **Inode** Unit is introduced to obtain the summation history current at each node. Three vectors are defined as shown in Fig. 3.21(b): (1) a 12-bit  $rmInodecfg$  to store configuration information of each node. The bit assignments are as follows:  $rmInodecfg(9-0)$  indicates which history current is connected to this node;  $rmInodecfg(10)$  indicates its flow direction (flow into or flow out of this node) with '1' or '0',  $rmInodecfg(11)$  indicates whether all history currents are connected to the same node with '1' or '0'; (2) a 32-bit  $rmI_{hist}$  to store the value of history current; and (3) an 8-bit  $rmINo$  to store the index of each node.

Fig. 3.22 shows hardware design of **Inode** Unit which is quite similar to the **MVM** unit. The *counter*, whose value increases by one if  $rmInodecfg(11)$  is changed, is connected to  $rmINo$ , so that the accumulated history current can be saved at the right address in

the current vector  $rmInode$ . The timing diagram and the corresponding  $rmInode$  vector is shown in Fig 3.21(c) and (d).

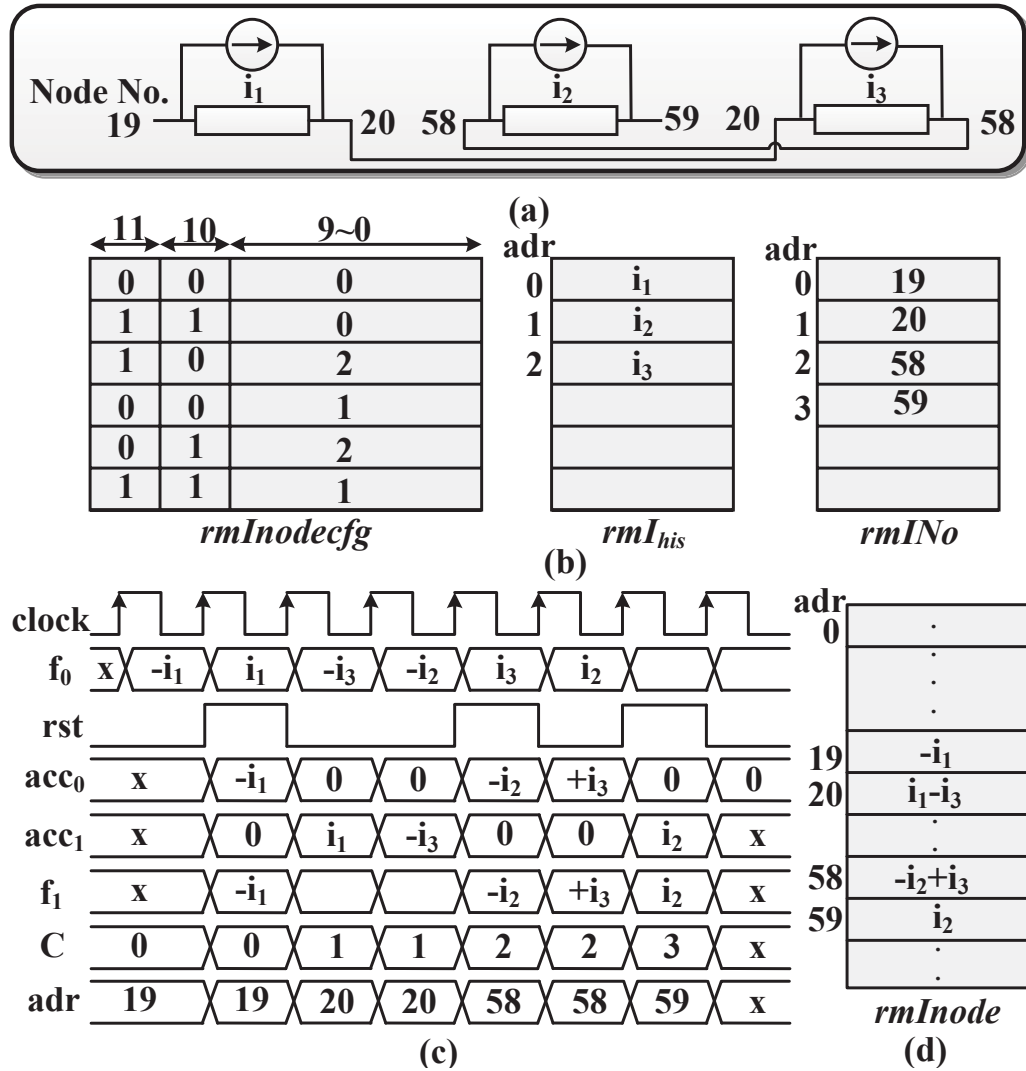


Figure 3.21: **Inode** unit: (a) an example circuit; (b) corresponding RAM assignment; (c) timing diagram; (d)  $rmInode$  vector.

### 3.9 Linear Network Solver Module

After obtaining all the history current terms of transformer model, Transmission line model and passive elements model, the power system node voltages are computed by nodal analysis method as

$$\mathbf{Y}v(t) = i(t), \quad (3.73)$$

where  $v(t)$  and  $i(t)$  are the vector of  $n$  node voltage and current sources, respectively.  $\mathbf{Y}$  is the power system admittance matrix, which is unchanged if there is no switch operation

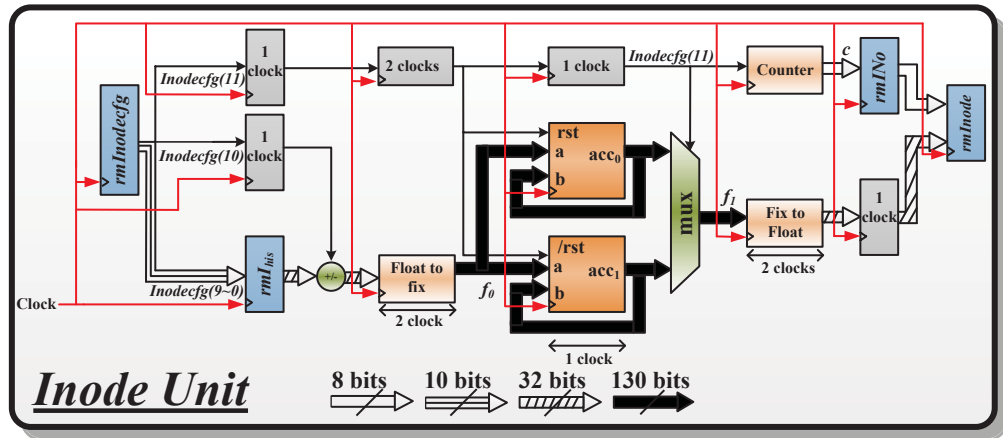


Figure 3.22: **Inode** unit: hardware design.

in the network.

To find the unknown node voltage, (3.73) are solved by partitioning vector  $v(t)$  into two vectors, vector A contains unknown node voltages and vector B contains all known node voltages. Then (3.73) can be rewritten as

$$\begin{bmatrix} \mathbf{Y}_{AA} & \mathbf{Y}_{AB} \\ \mathbf{Y}_{BA} & \mathbf{Y}_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{v}_A \\ \mathbf{v}_B \end{bmatrix} = \begin{bmatrix} \mathbf{i}_A \\ \mathbf{i}_B \end{bmatrix}, \quad (3.74)$$

so that the vector of unknown node voltage can be found by solving the following equation,

$$\mathbf{Y}_{AA}\mathbf{v}_A = \mathbf{i}_b, \quad (3.75)$$

where

$$\mathbf{i}_b = \mathbf{i}_A - \mathbf{i}_{A1}, \quad (3.76)$$

and

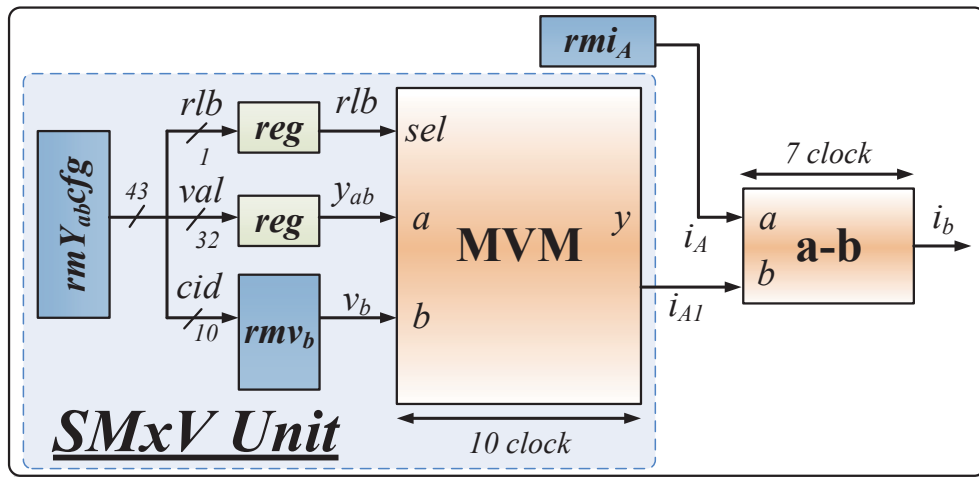
$$\mathbf{i}_{A1} = \mathbf{Y}_{AB}\mathbf{v}_B. \quad (3.77)$$

The vector of  $\mathbf{i}_A$  is consisted by history current terms from other modules, such as transformer module.

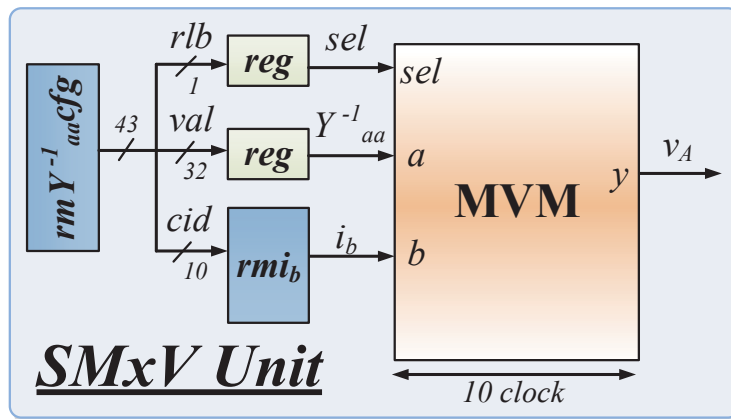
In real-time implementation, to solve the linear equation (3.75) the inverse system admittance matrix  $\mathbf{Y}$  are precalculated and then multiplied with the vector  $\mathbf{i}_b$ , given as

$$\mathbf{v}_A = \mathbf{Y}_{AA}^{-1}\mathbf{i}_b, \quad (3.78)$$

Since both matrix  $\mathbf{Y}_{AB}$  and  $\mathbf{Y}_{AA}^{-1}$  are very sparse in power system, in order to further reduce the computational burden as well as the hardware resource utilization, the sparse matrix-vector multiplication SMxV scheme is developed to carry out (3.77) and (3.78) as shown in Fig. 3.23 (a) and (b), respectively. In the SMxV, the matrix storage format is the same as matrix vector multiplication unit (MVM) discussed in Section (3.1.2). The only different is



(a)



(b)

Figure 3.23: (a) Pipelined calculation scheme for  $i_b$ ; (b) for  $v_A$

in SMxV only non-zero entities are recorded in the memory. There are also three signals for each entry stored in memory: one is a 32-bit wide signal  $val$  to store the value of the matrix entries in row order; the second is a 10-bit wide signal  $cid$  to store the column index of the entries, and the third is a 1 bit signal  $rlb$  to label all entries in the same row with '0' or '1'. An example matrix, its memory storage which is generated by the above and the corresponding timing diagram are given in Fig. 3.24 (a) and (b), respectively. In the SMxV unit, the accumulation is still done in fixed-point format, due to the reason discussed in Section (3.1.2).

### 3.10 Digital Hardware Emulation of The Power Transformer

The complete real-time transformer model hardware architecture is shown in Fig. 3.25. It can be divided into three parts: the linear module, the nonlinear module and the main control module. The linear module is built-up based on the admittance matrix approach

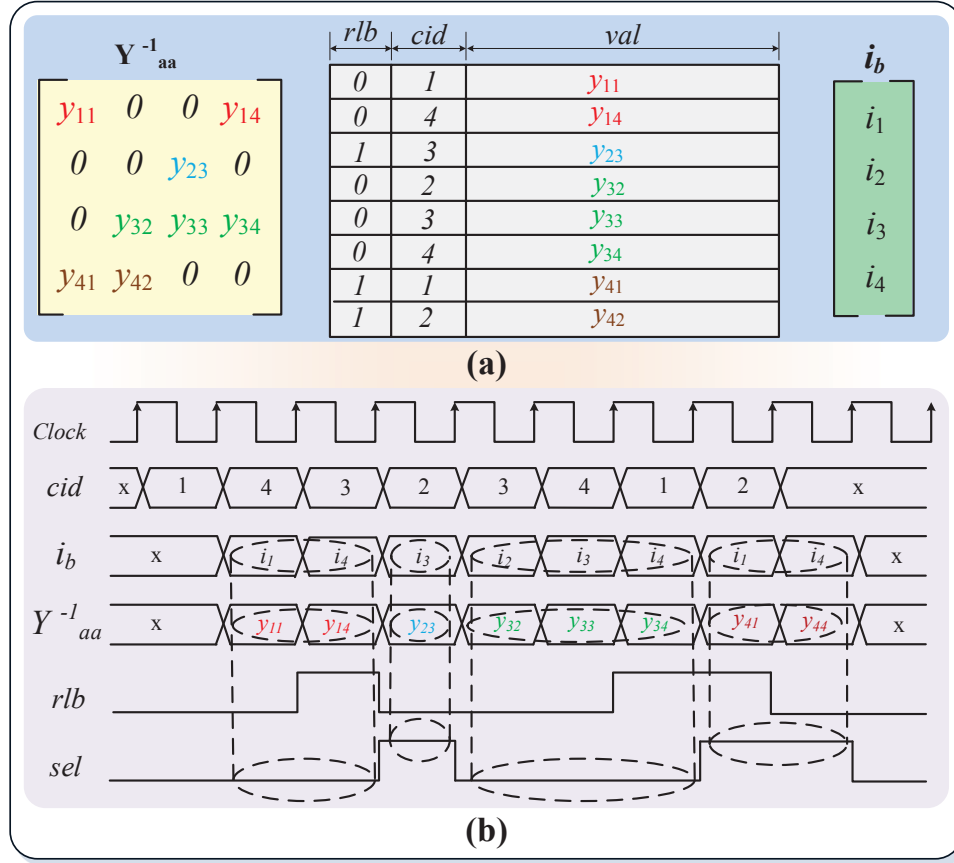


Figure 3.24: (a) An example sparse matrix and its storage format; (b) its corresponding timing diagram.

after discretizing the transformer differential equations, and is used to update the transformer history current terms, and calculates the node voltages without the nonlinear elements according to the compensation method. The nonlinear module contains two parts: the Preisach Hysteretic Unit (PHU) and the Newton-Raphson nonlinear solver unit, for finding the solution to the nonlinear equations. Further details regarding the nonlinear module are provided in Section II(C). The control module sends command signals to other modules to control which functions will be performed by each module, and also waits for the signals sent back by each module, so that all modules can be operated either in parallel or sequentially as required by the transient algorithm. Fig. 3.26 shows the detailed finite state machine (FSM) for the main control module.  $S_1$  calculates the summation of all the history current terms ( $\mathbf{I}_{hist}$ ) received from the linear transformer module ( $\mathbf{I}_{histT}$ ), eddy current module ( $\mathbf{I}_{histE}$ ) and the network components ( $\mathbf{I}_{histNET}$ ) which include the passive elements (RLC) module and transmission line (Tline) module, which together constitute the linear part of a power system.  $S_2$  computes the nodal voltages ( $v_o$ ) without the nonlinear elements.  $S_3$  evaluates the nonlinear functions of  $f(i_{com})$  and their derivative  $df(i_{km})$ .  $S_4$  performs the Newton-Raphson procedure to compute the results in each iteration; if



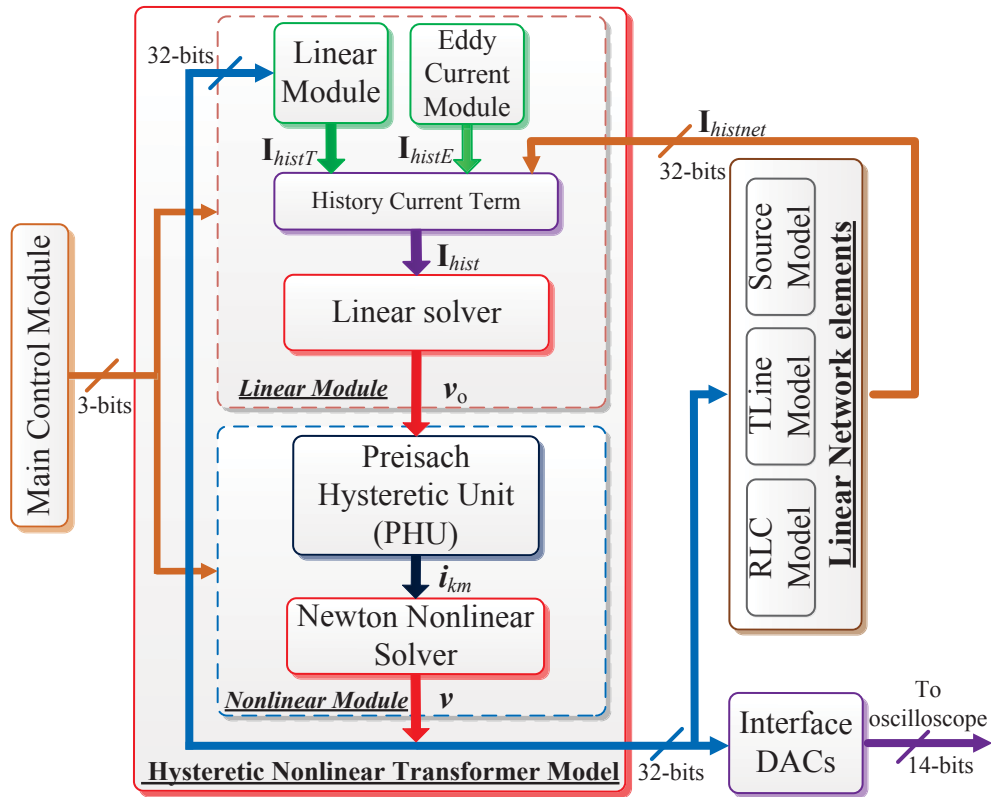


Figure 3.25: Overall architecture of the real-time nonlinear hysteretic transformer model on FPGA.

the result is converged, the FSM goes to  $S_5$  to calculate the nodal voltages ( $v$ ), otherwise it goes back to  $S_3$ . In  $S_6$  the transformer linear module, eddy current module as well as passive element and TLine modules update their history terms.

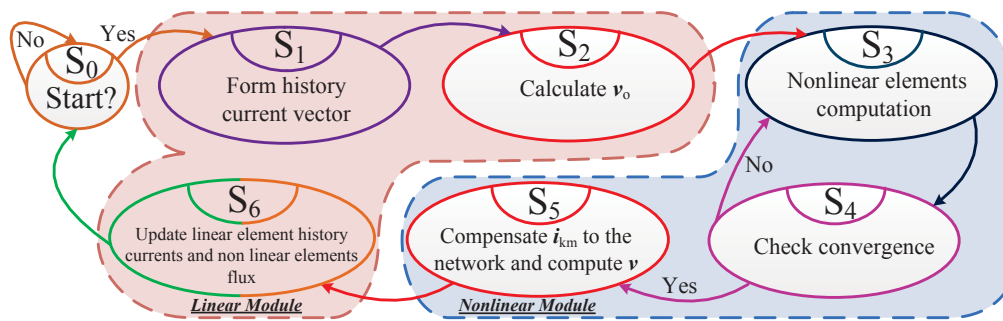


Figure 3.26: Finite state machine of the main control module.

### 3.11 The Real-time Nonlinear Transformer Model Setup on FPGA

The real-time nonlinear hysteretic transformer electromagnetic transient model is implemented on the Xilinx Virtex-7 VC707 XC7VX485T FPGA, featured as following: 485K logic

cells, 37Mbits block RAMs, 2800 DSP Slices and 700 user I/O pins. User design functionality is realized by configurable logic blocks (CLBs). The Virtex-7 FPGA outputs the real-time simulation results to a DAC board through a FMC connector integrated on the board, and a FMC-to-DAC adapter, and the real-time data are captured by an oscilloscope. The Chipscope<sup>®</sup> is a convenient tool provided by ISE software to visualize data. This tool can be used to analyze short transient periods, however, due to the limitation of the sampling number (up to 131072), this tool can not be used to trace the waveforms. The overall logic elements utilized by the nonlinear transformer model is 15%. The real-time simulation was carried out with a time-step of  $10 \mu s$  on the FPGA input clock frequency of 100MHz. The breakdown of model latencies in one simulation time-step is shown in Fig. 3.28. The execution time for one time-step is  $9.24 \mu s$  with a maximum of 3 Newton iterations per time-step. This means within a  $50 \mu s$  time-step, which is the acceptable time-step for real-time transient simulation, this FPGA-based nonlinear transformer model can simulate 5 nonlinear transformers in real-time with very small hardware resource consumption. Increasing the hardware utility or the clock frequency, it is also possible to simulate a larger power system with more transformer models including the nonlinear and frequency dependent phenomena.

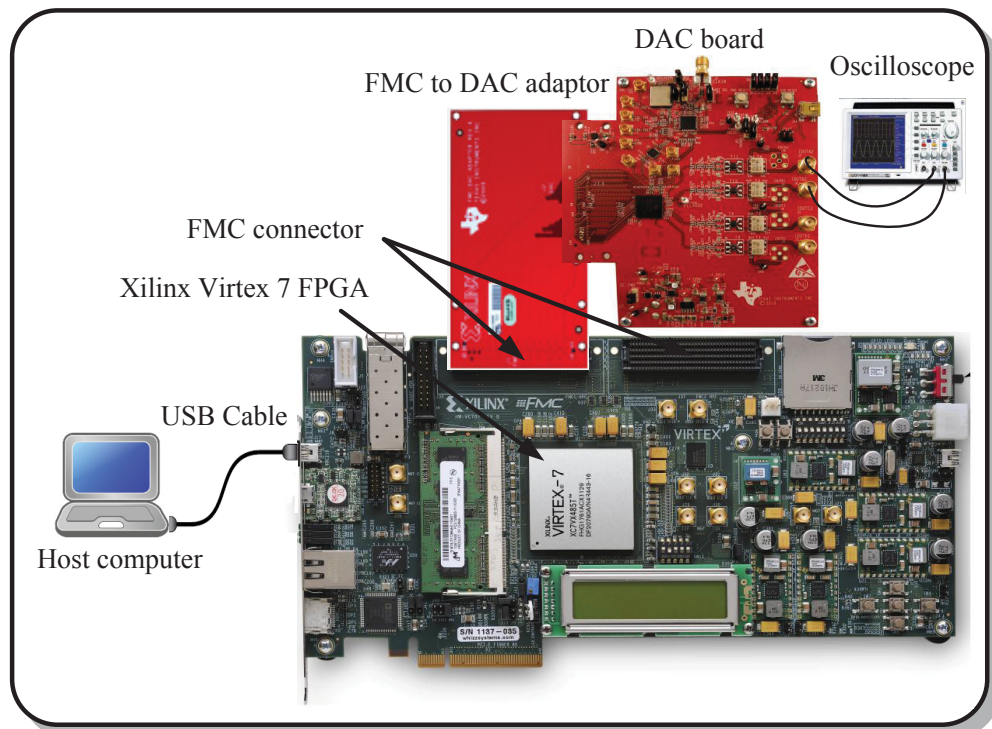


Figure 3.27: Experimental setup

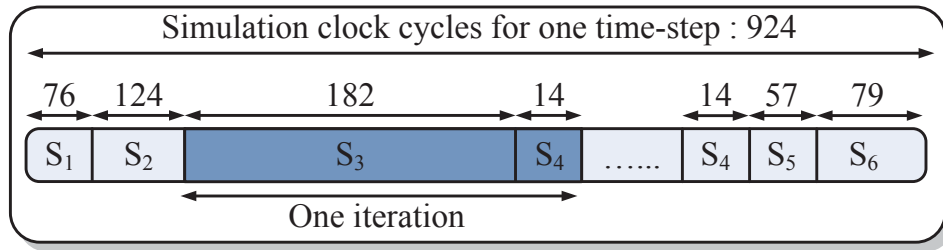


Figure 3.28: Model latencies for one time-step.

### 3.12 Case Study I

Two power systems are studied to show the effectiveness and accuracy of proposed FPGA-based real-time nonlinear transformer model. Fig. 3.29 shows the first case study: single-line diagram of a three-phase system. A transformer  $T$  is connected to a voltage source  $G$  through a circuit breaker  $S_{w1}$ .  $C_s$  is the breaker's grading capacitance, and  $C_p$  is the phase-to-earth capacitance including transformer winding capacitance. The system data is listed in the Appendix. The real-time simulation results are validated by off-line software ATP.

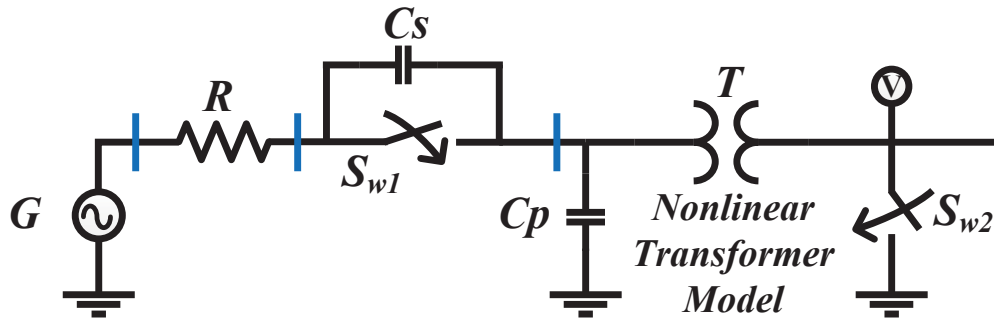


Figure 3.29: Single-line diagram for case study I.

#### 3.12.1 Inrush Current Transients

In this inrush current transients case study, capacitors  $C_s$  and  $C_p$  are removed from Fig. 3.29 and a five-segment piece-wise nonlinear function is employed to represent the saturation effect. When the breaker  $S_{w1}$  is closed at  $0.1s$ , the real-time captured three-phase breaker current waveforms are shown in Fig. 3.30 (a). As can be seen from these waveforms, phases a and c have larger current amplitude, while the amplitude of phase b current is relatively small. These unbalanced current waveforms are caused due to the instantaneous voltage supply values being different in various phases when the breaker  $S_{w1}$  is closed. The peak magnitude inrush current value of phases a and c is 210% of the transformer rated current, and it decays to steady-state after  $0.6s$ . Similar inrush current behaviour can be seen in the ATP off-line simulation results in Fig. 3.30 (b).

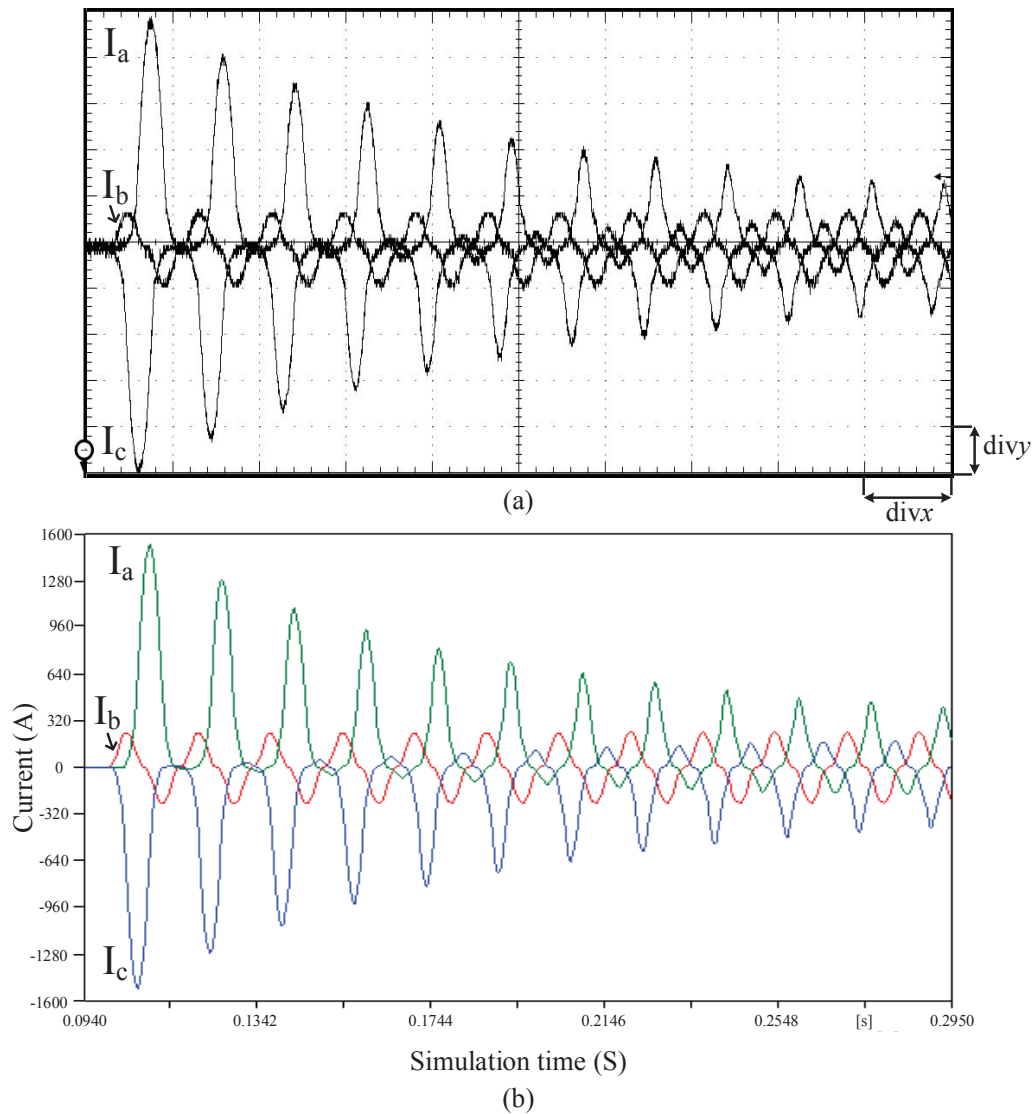


Figure 3.30: Breaker current when breaker close at  $t = 0s$  (a) real-time simulation results (1  $divx = 20$  ms; 1  $divy = 320$  A); (b) off-line (ATP) simulation results.

### 3.12.2 Hysteresis Behaviour

To investigate the hysteresis behaviour in the magnetizing core, the PHU is used to model the saturation and hysteresis during the inrush current transient. The parameters to define the hysteresis trajectory are listed in the Appendix. The initial flux in the core is set on the downward major loop trajectory. Fig. 3.31 shows the real-time simulation results for the flux and magnetizing current of the nonlinear transformer. From  $t = 0$  to 0.106 s, the power system is energized from a voltage source at a value of 0.65 pu. As shown in Fig. 3.31, during the first cycle (1-2), the flux is non-symmetrical due to the initial remanent flux and the peak flux value is 1 pu. After the first cycle, the flux is symmetrical and travels on the minor loop since the value of voltage is not high enough to drive the transformer

working in the saturation region (3-4). From  $t = 0.106$  to  $0.15$  s, the breaker is opened and the flux is locked at  $0.62$  pu. From  $t = 0.15$  s, the voltage rises to  $1.25$  pu and the breaker is reclosed, when the peak flux value reaches  $1.25$  pu, and the high voltage value drives the flux into saturation region traveling on the major loop (5-6).

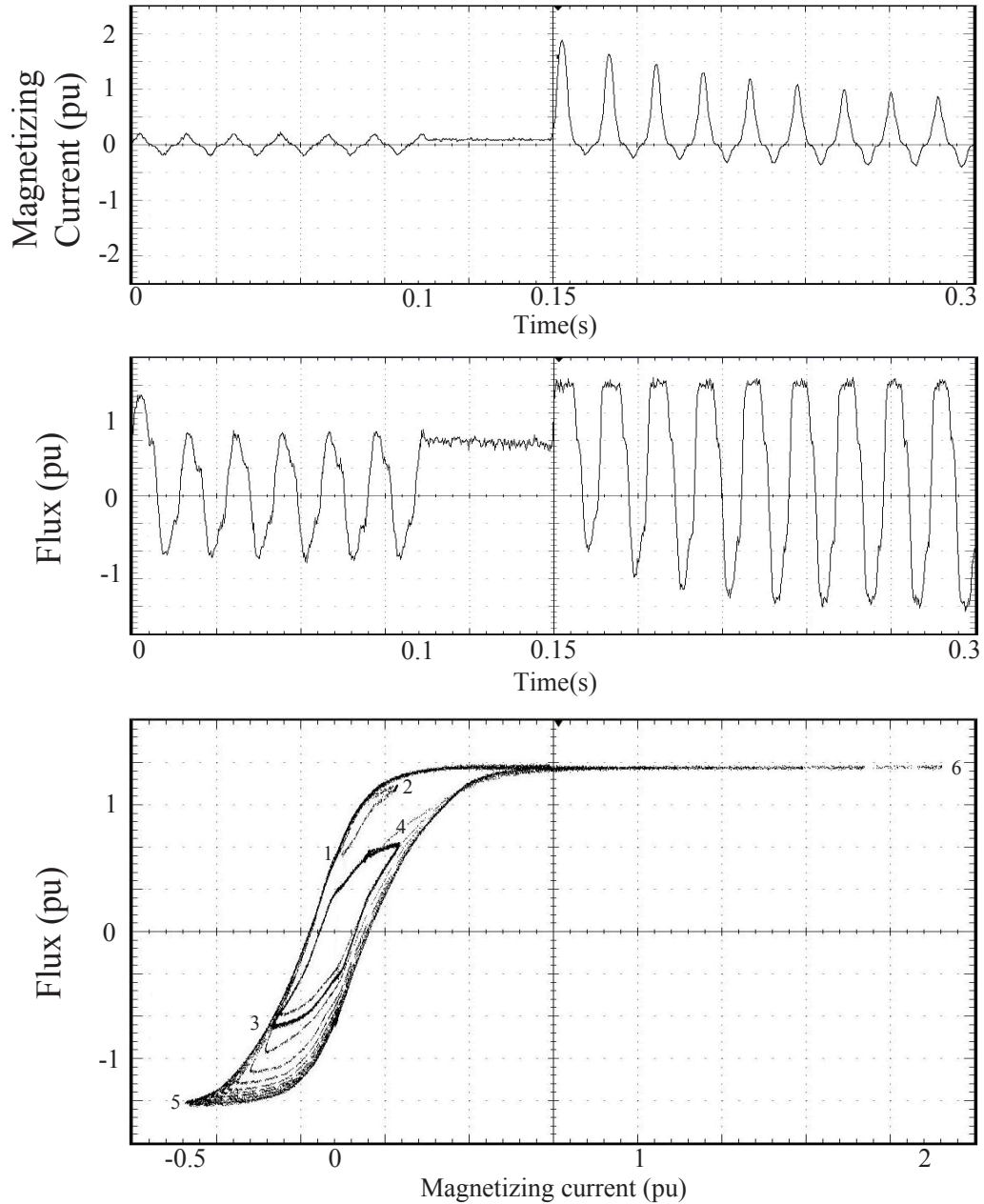


Figure 3.31: Real-time simulation results for the flux, magnetizing current, and the hysteresis characteristic of the nonlinear transformer.

### 3.12.3 Ferroresonance Transients

Ferroresonance is an electrical phenomenon which can occur when a nonlinear inductance is fed from a source. It can cause over-voltages and over-currents in the system with subsequent risk of damage to equipment. In Fig. 3.32, after opening  $S_{w1}$  while the breaker  $S_{w2}$  is closed at  $t = 0.2$  s to simulate a three-phase ground fault, the voltage waveforms are highly distorted and have a higher peak value 220% of the steady-state voltage value. Again, similar voltage behaviour can be observed in off-line simulation results in Fig. 3.32 (b).

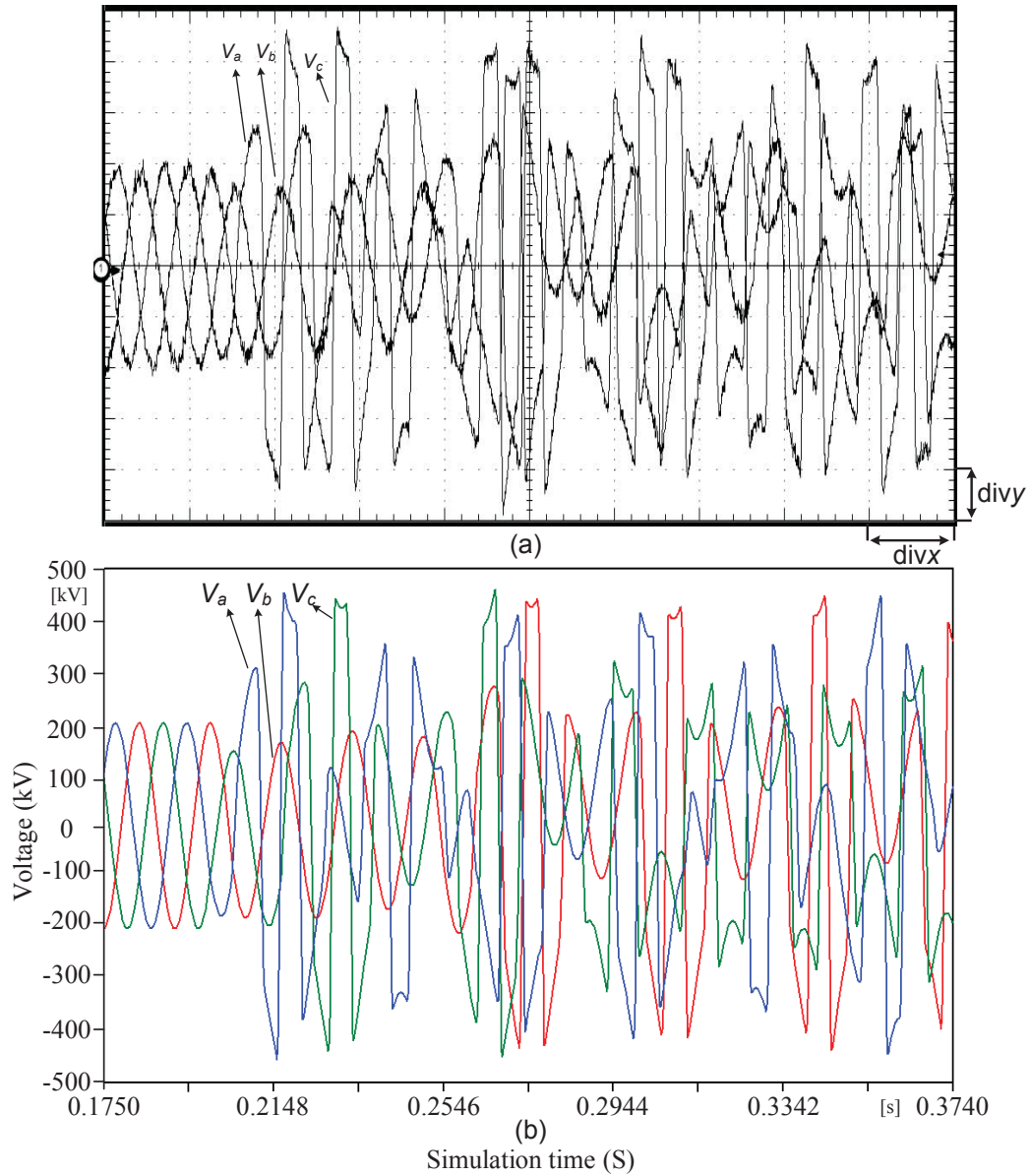


Figure 3.32: Primary side three-phase voltages when breaker opened at  $t = 0.2$  s (a) real-time simulation results (1 divx = 20 ms; 1 divy = 100 kV); (b) off-line (ATP) simulation results.



### 3.12.4 Eddy Current Transient

Transient recovery voltages (TRV) manifest across any circuit breaker in power systems when the breaker attempts to clear a nearby fault. This case study is initialized by closing  $S_{w2}$  to cause a ground fault, then reopening  $S_{w2}$  to clear this fault. The real-time simulation results of voltage across  $S_{w2}$  by using two different models, Cauer equivalent circuit and a constant resistor are captured. As shown in Fig 3.33, the Cauer model simulation result shows a higher peak value and much longer damping time than that with a constant resistor model, this implies that the eddy currents increase the power losses in the transformer during the transient recovery period.

## 3.13 Case Study II

In the second case study, the studied power system involves: (1) three transformers ( $T_1, T_2, T_3$ ), all of them employing the full nonlinear modeling; (2) three transmission lines ( $L_1, L_2, L_3$ ), modeled using distributed parameter line model (Bergeron's model); (3) one ideal voltage source ( $V_s$ ); (4) one switch ( $SW_1$ ) and a capacitor  $C$  are used to model the capacitor switching. The parameters for this case study are also listed in the Appendix.

### 3.13.1 Capacitor Switching Transients

The entire power system is energized until  $t = 1$  s when the capacitor switching is initiated by closing  $SW_1$ . Both real-time and off-line simulation results are captured to show the primary winding voltages at transformer  $T_3$ . From the waveforms, after closing the switch, the voltage dropped to zero because the voltage of capacitor at the secondary winding of this transformer cannot change immediately, and then a fast voltage recovery is observed. The distorted recovery voltage is a maximum of 150% of the steady-state voltage value, and decays to steady state after 3 s.

### 3.13.2 Ground Fault Transients

This case study is achieved by initiating a three-phase-to-ground fault at  $t = 1$  s at the secondary winding of transformer  $T_2$ , and the voltage waveforms at the primary winding of transformer  $T_3$  are captured both in real-time and off-line, as shown in Fig 3.36. Compared with capacitor switching transients, the ground fault transients behave in a highly distorted fashion, however, there is no overshoot in voltage, and steady-state is achieved after 4 s.

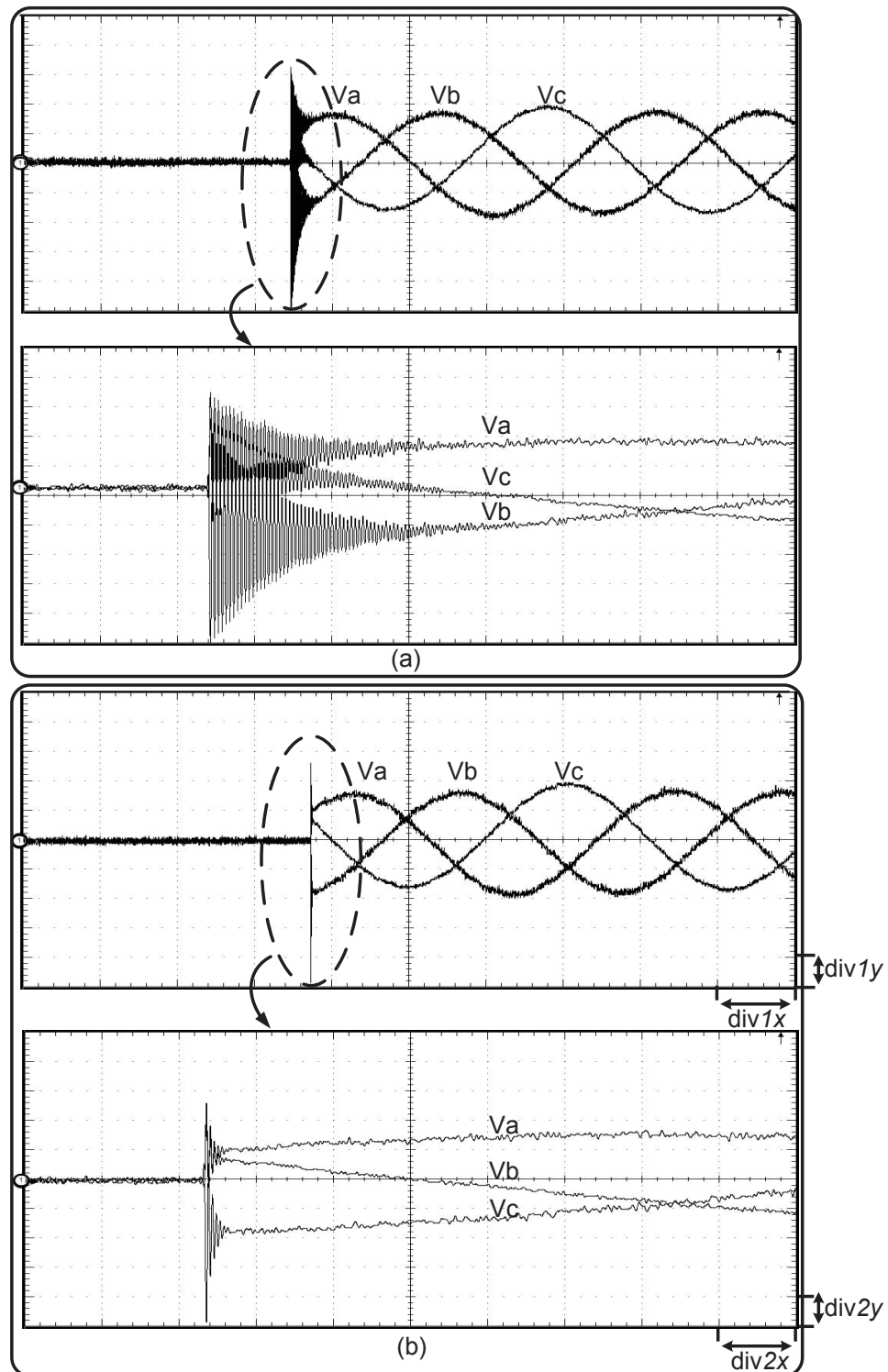


Figure 3.33: TRV real-time simulation results at  $t = 0.2s$  (a) Cauer model; (b) non frequency-dependent model (constant resistor) ( $1 \text{ div1x} = 4 \text{ ms}$ ;  $1 \text{ div1y} = 84 \text{ kV}$ ;  $1 \text{ div2x} = 400 \mu s$ ;  $1 \text{ div2y} = 84 \text{ kV}$ ).



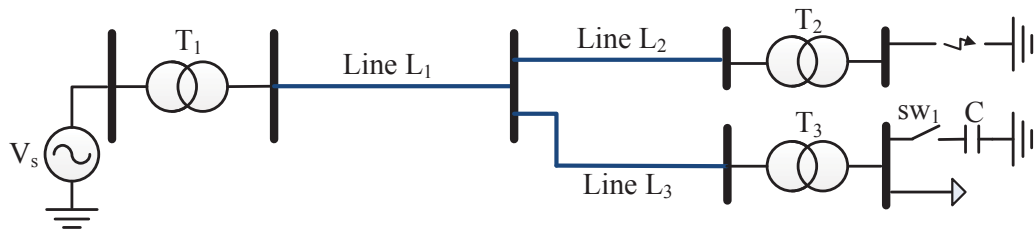


Figure 3.34: Single-line diagram for case study.

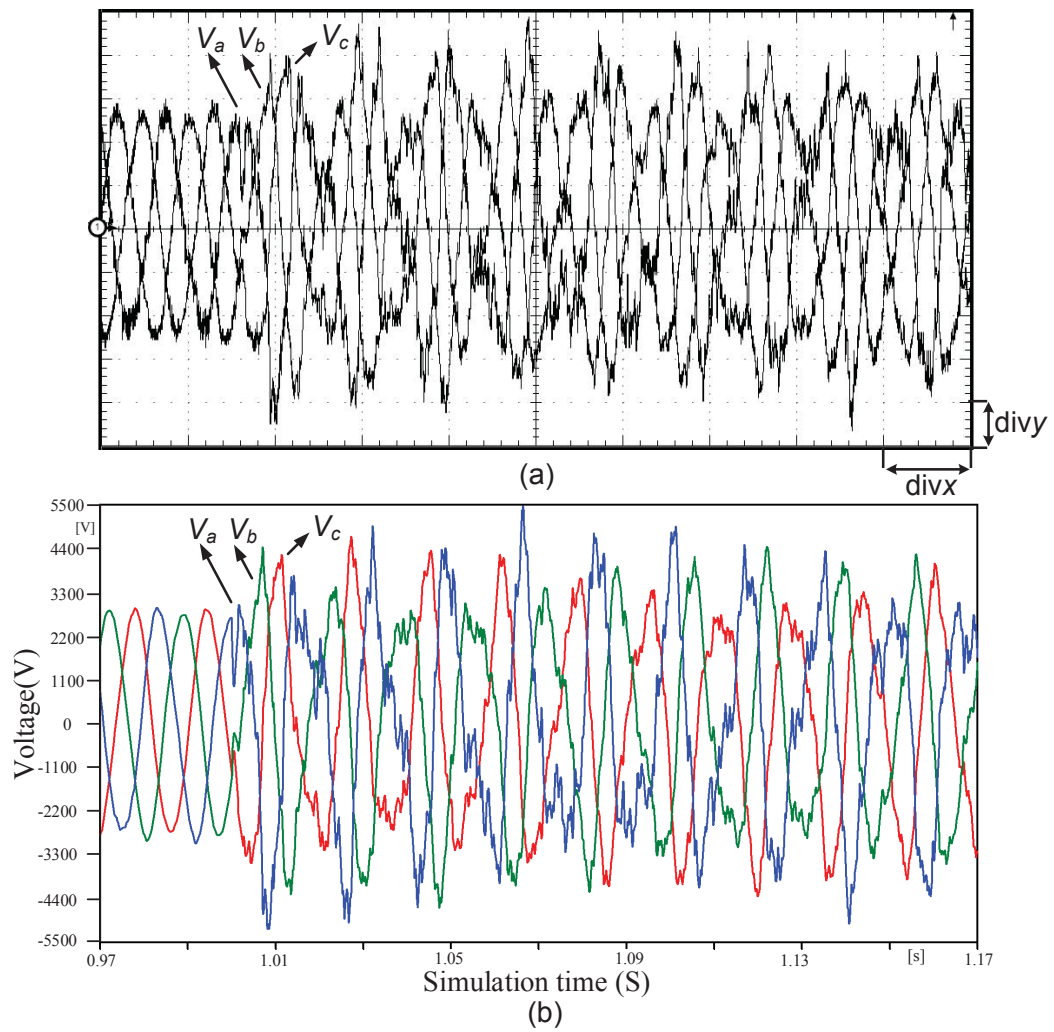


Figure 3.35: Capacitor switch at  $t = 1$  s (a) real-time simulation results (1 divx = 10 ms; 1 divy = 1.06 kV); (b) off-line (ATP) simulation results.

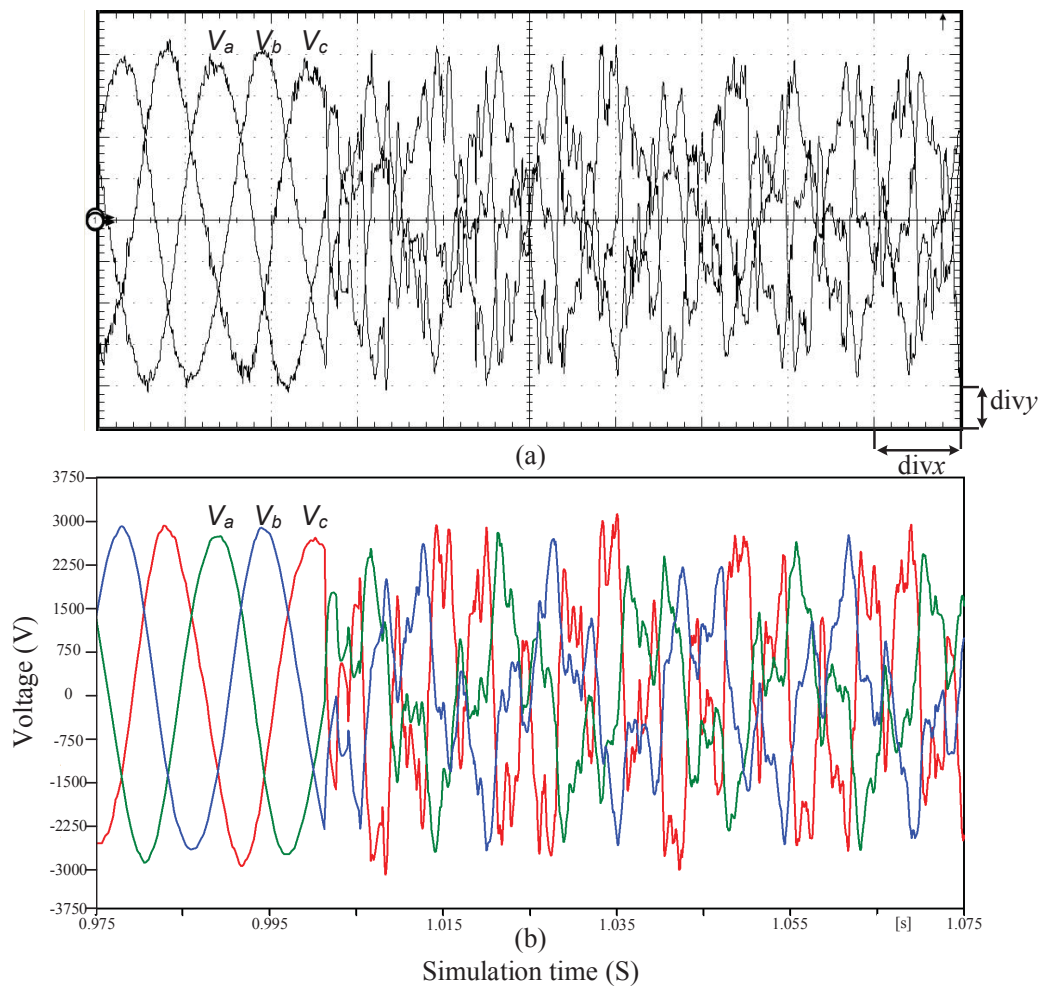


Figure 3.36: Ground fault at  $t = 1\text{ s}$  (a) real-time simulation results (1  $\text{divx} = 20\text{ ms}$ ; 1  $\text{divy} = 1.5\text{ kV}$ ); (b) off-line (ATP) simulation results.

### 3.14 Summary

A nonlinear hysteretic power transformer hardware model is developed in real-time on the FPGA. Taking the natural advantages of the hardwired architecture, parallelism, and pipelining, enabled us to achieve real-time simulation. This hardware power transformer model has the following features: linear model to update the history current terms; non-linear phenomena such as hysteresis is included with Newton iteration for solving the nonlinear equations; the Preisach model was used to define the hysteretic loop; and a frequency dependant equivalent circuit was used to model eddy current behaviour. Two case studies were analyzed to show the accuracy and efficiency of this proposed model under various transient conditions.

# 4

## Detailed Magnetic Equivalent Circuit Based Real-Time Transformer Model

A detailed power transformer electromagnetic transient (EMT) model is presented which would help in accurately predicting transient stresses and formulating adequate protection strategies in power systems. The presented FPGA-based nonlinear high resolution magnetic equivalent circuit (HR-MEC) based model for the power transformer for real-time simulation of electromagnetic transients is inspired by the mesh generated in finite element (FEM) tools to depict the major flux paths in the transformer. The advantages of this approach include:

1. The accounting of the major flux paths in the transformer core using a detailed magnetic equivalent circuit. Transformer core hysteresis modeling using the Preisach theory [53], and eddy current behavior modeling using a frequency-dependent equivalent network.
2. Reduced model latencies due to the exploitation of full hardware parallelism and pipelining on the FPGA; thus smaller time steps can be used for capturing higher frequency transients. FPGAs have proven to be quite efficacious in both industrial control [40, 56–58], as well as power system electromagnetic transient modeling [15, 27]; however, a detailed real-time modeling of the power transformer on the FPGA is yet to be reported.
3. Sparse matrix methods are exploited to further increase computational efficiency. A parallel LU decomposition method that in both time and resource efficient is proposed to solve the transformer and network nodal equations.
4. The implementation of the UMEC model on the traditional sequential processor

takes up a lot of computational resources, not only for the model execution but also to update and invert the system admittance matrix in each time-step. Therefore, the test system size becomes smaller for a given simulator hardware. With a low latency HR-MEC based transformer model on the FPGA larger system sizes can be accommodated.

5. The detailed real-time nonlinear HR-MEC of the power transformer emulation on the FPGA is validated using 3-D FEM under various operating conditions.

Based on the case study results, the FPGA based real-time HR-MEC transformer model requires longer simulation time and consumes more hardware resource to finish all the calculations compared to the admittance matrix based transformer model. This model is a good candidate to predict the major flux paths in the core, enabling it to be used in the HIL simulation for the transformer design procedure. However, the HR-MEC model utilizes more than 40% of the hardware resource; in addition, the transformer admittance matrix keeps changing in each simulation time-step, which requires an updated nodal solution. An application of this model could be a dedicated FPGA emulating the HR-MEC model interfaced with another FPGA or PC which emulates the host power system with 1 time-step delay. The formulations of the HR-MEC model are derived from the relationship between the fluxes in the core and the transformer voltages; the flux are driven by the magnetomotive force (MMF) sources induced by the transformer terminal currents in each coil. Hence, this general procedure can be applied to other types of transformer core design, such as shell type and multi-limb transformers. The potential transformer and the voltage transformer have the similar topology iron core design, so that this model can also be employed to emulate instrument transformers.

## **4.1 Detailed Hardware Emulation of The Power Transformer**

### **4.1.1 High Resolution Magnetic Equivalent Circuit Based Transformer Model**

#### **4.1.1.1 The HR-MEC representation**

Fig. 4.1 shows a three-phase three-limb transformer with the major flux paths. The MEC representation is developed based on magnetic equivalent circuit theory [59,60] and can be used to represent the transformer core construction and the flux paths. The passage of various fluxes in the magnetic circuit is governed by Ampere's and Gauss' laws analogous to Kirchhoff's voltage and current laws in an electrical circuit. In the MEC, each transformer winding is represented by a permeance element and a magneto-motive force (MMF); however, the flux in the transformer winding is different at different points, even in the same winding. To obtain the detailed representation of the flux path, a High Resolution MEC (HR-MEC) is developed in this paper, where each coil is divided into several sub-coils with smaller number of turns. In the MEC network, this sub-coil is represented

by a group consisting of different values of nonlinear permeances and MMFs in the winding branches. Consequently, the linear leakage permeances are also divided to connect in parallel with each sub-coil group. Fig. 4.2 shows the HR-MEC with two groups of sub-coils in each winding for a three-phase three-limb transformer. There are three types of components in this equivalent circuit:

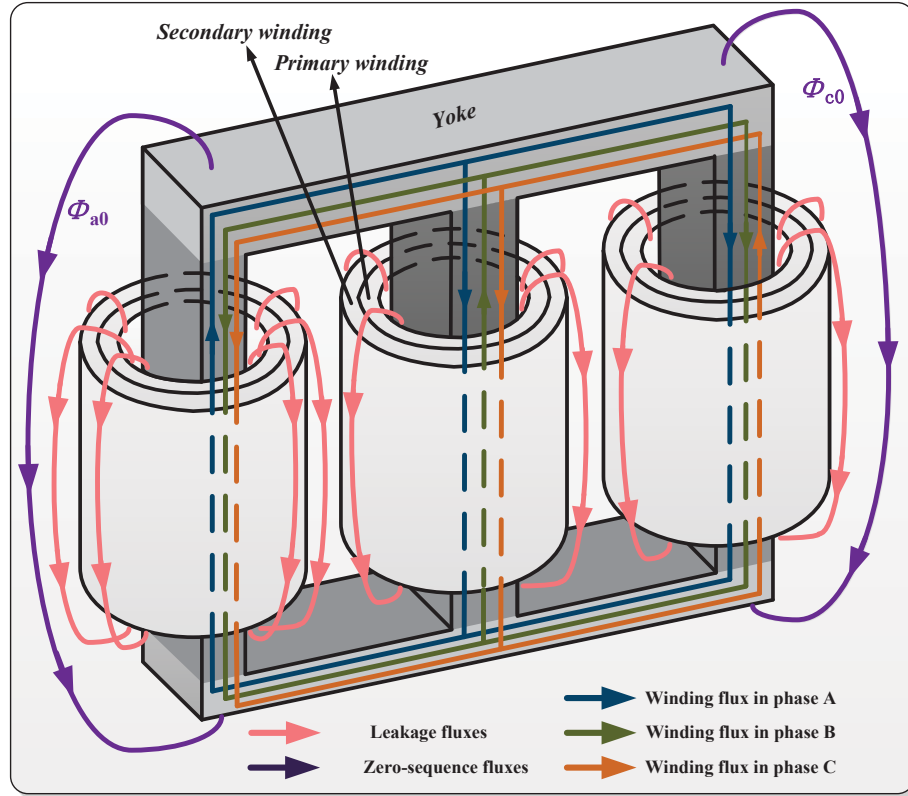


Figure 4.1: Major flux paths in a three-phase three-limb transformer.

- Nonlinear iron core permeances:
  - $P_{pa1}, P_{pa2}, P_{pb1}, P_{pb2}, P_{pc1}, P_{pc2}$  represent iron permeances for the transformer primary winding in phases  $a, b$  and  $c$ , respectively;  $P_{sa1}, P_{sa2}, P_{sb1}, P_{sb2}, P_{sc1}, P_{sc2}$  represent iron permeances for the transformer secondary winding in phases  $a, b$  and  $c$ , respectively.
  - $P_{ab}$  and  $P_{bc}$  represent iron permeances for the transformer yoke between phase- $a$  and phase- $b$ , and phase- $b$  and phase- $c$ , respectively.
- Linear leakage permeances:
  - $P_{la1}$  to  $P_{la4}, P_{lb1}$  to  $P_{lb4}$ , and  $P_{lc1}$  to  $P_{lc4}$  represent the transformer leakage air permeances in phases  $a, b$  and  $c$ , respectively.

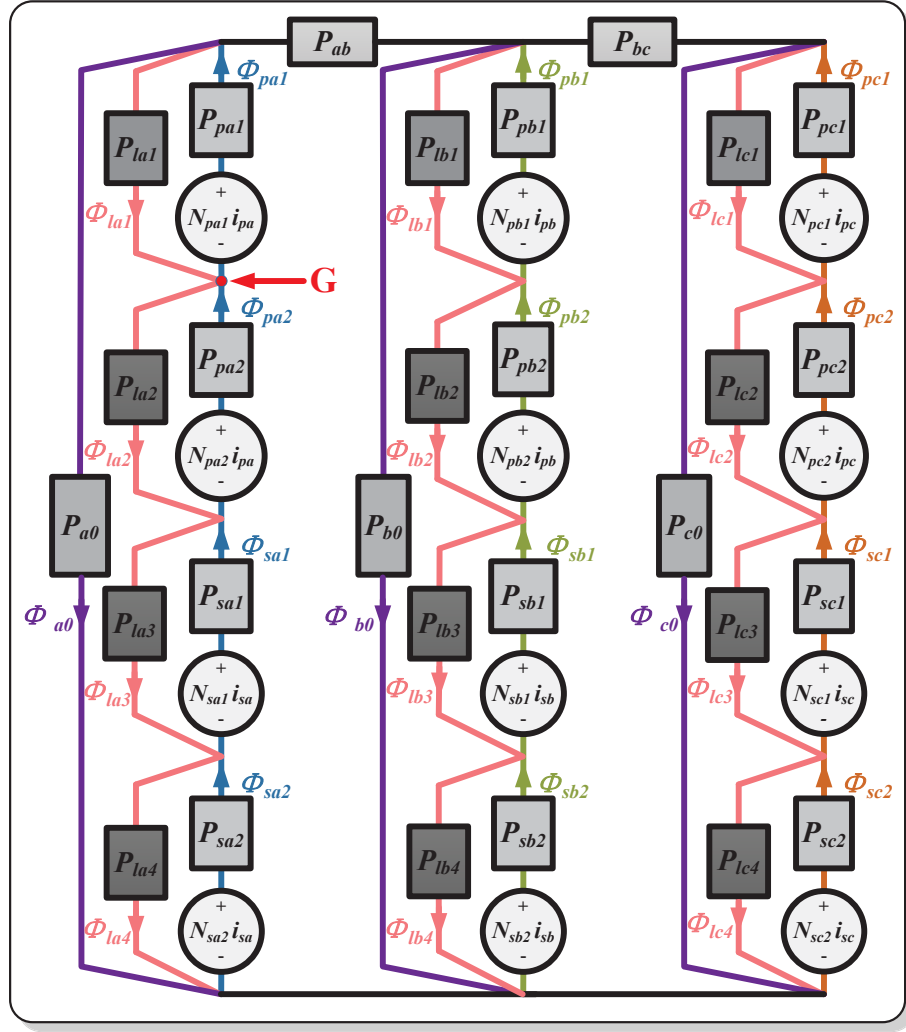


Figure 4.2: HR-MEC model for a three-phase three-limb transformer.

–  $P_{a0}$ ,  $P_{b0}$  and  $P_{c0}$  are zero-sequence permeances obtained for the three phases.

• MMF sources:

–  $N_{pa1}i_{pa}$ ,  $N_{pa2}i_{pa}$ ,  $N_{pb1}i_{pb}$ ,  $N_{pb2}i_{pb}$ ,  $N_{pc1}i_{pc}$  and  $N_{pc2}i_{pc}$  are generated by the currents flowing in the primary winding in phases  $a$ ,  $b$  and  $c$ , respectively;  $N_{sa1}i_{sa}$ ,  $N_{sa2}i_{sa}$ ,  $N_{sb1}i_{sb}$ ,  $N_{sb2}i_{sb}$ ,  $N_{sc1}i_{sc}$  and  $N_{sc2}i_{sc}$  are generated by the currents flowing in the secondary winding in phases  $a$ ,  $b$  and  $c$ , respectively.

#### 4.1.1.2 The HR-MEC formulation

The relationship between the branch fluxes and the mmf's can be written as,

$$\phi = \mathbf{P}(\mathbf{N}\mathbf{i} - \mathfrak{S}') \quad (4.1)$$

where  $\phi$ ,  $i$  and  $\mathfrak{S}'$  are  $nx1$  vectors of branch fluxes, winding currents, and branch mmf's, respectively, expressed as,  $\phi = [\phi_1(t) \ \phi_2(t) \ \dots \ \phi_n(t)]^T$ ,  $i = [i_1(t) \ i_2(t) \ \dots \ i_n(t)]^T$ , and  $\mathfrak{S}' = [\mathfrak{S}'_1(t) \ \mathfrak{S}'_2(t) \ \dots \ \mathfrak{S}'_n(t)]^T$ .  $n$  is the number of magnetic branches.  $\mathbf{P}$  and  $\mathbf{N}$  are  $nxn$  diagonal matrices of branch permeances and number of winding turns, respectively, given as,

$$\mathbf{P} = \text{diag}\{P_1(t) \ P_2(t) \ \dots \ P_n(t)\}, \mathbf{N} = \text{diag}\{N_1 \ N_2 \ \dots \ N_n\}. \quad (4.2)$$

The entries of vector  $i$  and matrix  $\mathbf{N}$  are zero for the branches with only permeance elements.

According to the Gauss' law for magnetism, the algebraic sum of fluxes entering or leaving a node must be zero. For example, at node 'G' in Fig. 4.2,  $-\phi_{pa1} + \phi_{pa2} + \phi_{la1} = \phi_{la2}$ . In general,

$$\mathbf{A}^T \phi = \mathbf{0}, \quad (4.3)$$

where  $\mathbf{A}^T$  is the node-branch connection matrix, with 1, -1, and 0 entries for branch flux entering, leaving, and no connection to the node, respectively.

Applying matrix  $\mathbf{A}^T$  to node MMF vector  $\mathfrak{S}_{node}$  results in the branch MMF vector,

$$\mathbf{A} \mathfrak{S}_{node} = \mathfrak{S}'. \quad (4.4)$$

Combining (4.1), (4.3), (4.4) gives,

$$\phi = \mathbb{P} \mathbf{N} i, \quad (4.5)$$

where,

$$\mathbb{P} = \mathbf{I} - \mathbf{P} \mathbf{A} (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T, \quad (4.6)$$

with  $\mathbf{I}$  being the  $nxn$  identity matrix.

Let the permeance network fluxes be partitioned into two sets,  $\phi^M$ , for the branches with MMF sources  $N_j i_j$  and permeance elements, and  $\phi^P$ , for branches with only permeance elements, then (4.5) can be rewritten as:

$$\begin{bmatrix} \phi^M \\ \phi^P \end{bmatrix} = \begin{bmatrix} \mathbb{P}^{MM} & \mathbb{P}^{MP} \\ \mathbb{P}^{PM} & \mathbb{P}^{PP} \end{bmatrix} \begin{bmatrix} \mathbf{P}^M & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^P \end{bmatrix} \begin{bmatrix} \mathbf{N}^M i^M \\ \mathbf{0} \end{bmatrix}, \quad (4.7)$$

For the three-phase three-limb HR-MEC as shown in Fig. 4.2, applying (4.7) results in:

$$\phi^M = \mathbb{P}^{MM} \mathbf{P}^M \mathbf{N}^M i^M, \quad (4.8)$$

where  $\mathbb{P}^{MM}$  is a 12x12 sub-matrix of  $\mathbb{P}$ .  $\phi^M$  and  $i^M$  are 12x1 vectors, and  $\mathbf{P}^M$  and  $\mathbf{N}^M$  are



diagonal 12x12 matrices, given as:

$$\begin{aligned}
\boldsymbol{\phi}^M &= [\phi_{pa1}(t) \ \phi_{pa2}(t) \ \phi_{pb1}(t) \ \phi_{pb2}(t) \ \phi_{pc1}(t) \ \phi_{pc2}(t) \\
&\quad \phi_{sa1}(t) \ \phi_{sa2}(t) \ \phi_{sb1}(t) \ \phi_{sb2}(t) \ \phi_{sc1}(t) \ \phi_{sc2}(t)]^T, \\
\mathbf{i}^M &= [i_{pa}(t) \ i_{pa}(t) \ i_{pb}(t) \ i_{pb}(t) \ i_{pc}(t) \ i_{pc}(t) \\
&\quad i_{sa}(t) \ i_{sa}(t) \ i_{sb}(t) \ i_{sb}(t) \ i_{sc}(t) \ i_{sc}(t)]^T, \\
\mathbf{P}^M &= \text{diag}\{P_{pa1}(t) \ P_{pa2}(t) \ P_{pb1}(t) \ P_{pb2}(t) \ P_{pc1}(t) \ P_{pc2}(t) \\
&\quad P_{sa1}(t) \ P_{sa2}(t) \ P_{sb1}(t) \ P_{sb2}(t) \ P_{sc1}(t) \ P_{sc2}(t)\}, \\
\mathbf{N}^M &= \text{diag}\{N_{pa1}(t) \ N_{pa2}(t) \ N_{pb1}(t) \ N_{pb2}(t) \ N_{pc1}(t) \ N_{pc2}(t) \\
&\quad N_{sa1}(t) \ N_{sa2}(t) \ N_{sb1}(t) \ N_{sb2}(t) \ N_{sc1}(t) \ N_{sc2}(t)\}.
\end{aligned}$$

For each of the transformer windings, the terminal voltages can be derived from Faraday's law, where  $k$  denotes the winding index,

$$v_k = N_{k1} \frac{d}{dt} \phi_{k1} + N_{k2} \frac{d}{dt} \phi_{k2} \quad (4.9)$$

In matrix format, (4.9) can be rewritten as,

$$\mathbf{v}_T = \mathbf{N}_Z \frac{d}{dt} \boldsymbol{\phi}^M(t), \quad (4.10)$$

where  $\mathbf{N}_Z$  is a 6x12 winding turns matrix.  $\mathbf{v}_T$  is a 6x1 terminal voltage vector, given as:

$$\begin{aligned}
\mathbf{v}_T &= [v_{pa}(t) \ v_{pb}(t) \ v_{pc}(t) \ v_{sa}(t) \ v_{sb}(t) \ v_{sc}(t)]^T, \\
\mathbf{N}_Z &= \begin{bmatrix} N_{pa1} & N_{pa2} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & N_{pb1} & N_{pb2} & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \dots & N_{sc1} & N_{sc2} \end{bmatrix}.
\end{aligned}$$

#### 4.1.1.3 HR-MEC model discretization

Applying Trapezoidal rule to discretize (4.10), the discrete-time difference equations can be obtained:

$$\mathbf{N}_Z \boldsymbol{\phi}^M(t) = \frac{\Delta t}{2} \mathbf{v}_T(t) + \boldsymbol{\phi}_{hist}^M(t), \quad (4.11)$$

where  $\Delta t$  is the simulation time-step, with  $\boldsymbol{\phi}_{hist}^M(t)$  being the 6x1 history vector term expressed as:

$$\boldsymbol{\phi}_{hist}^M(t) = \frac{\Delta t}{2} \mathbf{v}_T(t - \Delta t) + \mathbf{N}_Z \boldsymbol{\phi}^M(t - \Delta t). \quad (4.12)$$

Using (4.8), the left side of (4.11) can be expressed as:

$$\mathbf{N}_Z \phi^M(t) = \mathbf{Z} \mathbf{i}^M, \quad (4.13)$$

where  $\mathbf{Z}$  is a 6x12 matrix given as:

$$\mathbf{Z} = \mathbf{N}_Z \mathbb{P}^{MM} \mathbf{P}^M \mathbf{N}^M. \quad (4.14)$$

Further (4.13) can be rewritten as:

$$\mathbf{N}_Z \phi^M(t) = \mathbf{Z} \mathbf{i}^M = \mathbf{Z}_{(HR-MEC)} \mathbf{i}_T(t), \quad (4.15)$$

with  $\mathbf{Z}_{(HR-MEC)}$  being the 6x6 matrix and  $\mathbf{i}_T(t)$  being the 6x1 winding currents vector, which are given as:

$$\mathbf{Z}_{(HR-MEC)} = \begin{bmatrix} Z_{1,1} + Z_{1,2} & \dots & Z_{1,11} + Z_{1,12} \\ Z_{2,1} + Z_{2,2} & \dots & Z_{2,11} + Z_{2,12} \\ \cdot & \cdot & \\ \cdot & \cdot & \\ \cdot & \cdot & \\ Z_{6,1} + Z_{6,2} & \dots & Z_{6,11} + Z_{6,12} \end{bmatrix}, \quad (4.16)$$

and

$$\mathbf{i}_T(t) = [i_{pa}(t) \ i_{pb}(t) \ i_{pc}(t) \ i_{sa}(t) \ i_{sb}(t) \ i_{sc}(t)]^T. \quad (4.17)$$

Combining (4.11) and (4.15), the relationship between transformer terminal currents  $\mathbf{i}_T(t)$  and voltages  $\mathbf{v}_T(t)$  can be obtained:

$$\mathbf{Z}_{(HR-MEC)} \mathbf{i}_T(t) = \frac{\Delta t}{2} \mathbf{v}_T(t) + \phi_{hist}^M(t), \quad (4.18)$$

which finally gives the Norton equivalent formula,

$$\mathbf{i}_T(t) = \mathbf{Y}_T \mathbf{v}_T(t) + \mathbf{I}_{histT}(t), \quad (4.19)$$

where

$$\mathbf{I}_{histT}(t) = \mathbf{Z}_{(HR-MEC)}^{-1} \phi_{hist}^M(t), \quad (4.20)$$

and

$$\mathbf{Y}_T = \frac{\Delta t}{2} \mathbf{Z}_{(HR-MEC)}^{-1}. \quad (4.21)$$

The transformer terminal voltage vector in (4.19)  $\mathbf{v}_T(t)$  can be split into two sub-vectors,  $\mathbf{v}_T(t) = [\mathbf{v}_{T,p}(t) \ \mathbf{v}_{T,s}(t)]$ , where  $\mathbf{v}_{T,p}(t)$  and  $\mathbf{v}_{T,s}(t)$  are the primary side and secondary side voltage vectors, respectively. The vectors  $\mathbf{i}_T$  and  $\mathbf{I}_{histT}(t)$  can also be split into sub-vectors to present the primary side current  $\mathbf{i}_{T,p}(t)$  and history term  $\mathbf{I}_{histT,p}(t)$  and secondary side current  $\mathbf{i}_{T,s}(t)$  and history term  $\mathbf{I}_{histT,s}(t)$ . The discrete-time Norton equivalent circuit

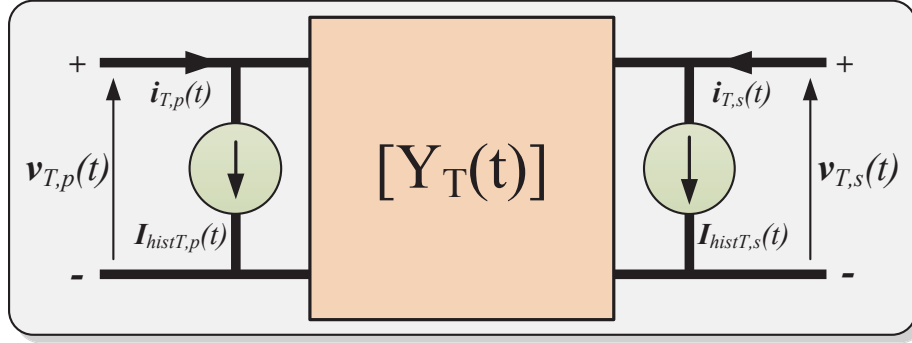


Figure 4.3: Norton equivalent circuit of the HR-MEC model over one simulation time-step with a time-varying admittance matrix.

for the HR-MEC based transformer model is shown in Fig. 4.3. The inputs for this discretized model are the primary  $i_{T,p}(t)$  and secondary  $i_{T,s}(t)$  winding currents, the primary  $I_{histT,p}(t)$  and secondary  $I_{histT,s}(t)$  history terms, and the admittance matrix  $\mathbf{Y}_T(t)$ , while the outputs are the transformer primary  $v_{T,p}(t)$  and secondary  $v_{T,s}(t)$  terminal voltages, calculated by its nodal analysis method. Since the nonlinear permeance matrices  $\mathbb{P}^{MM}$  and  $\mathbf{P}^M$  in (4.14) are time-varying, the resulting admittance matrix  $\mathbf{Y}_T$  is not constant but needs to be recalculated in every simulation time-step. Thus, an efficient parallel LU decomposition solver, described in Section III(B), is used for the network solution. This is the significant difference from a lumped parameter based transformer model such as the BCTRAN model [30], in which the admittance matrix is constant.

#### 4.1.1.4 HR-MEC model hardware design

It is obvious from the equations involved in the HR-MEC calculation process, that matrix-matrix multiplications are necessary. Some of the matrices are sparse matrices, such as  $\mathbf{A}, \mathbf{P}, \mathbf{N}$ . The entries in these sparse matrix may keep changing in each emulation time-step, but the pattern remains unchanged. In order to reduce the computational latency, a sparse-dense matrix multiplication unit (**SMxDM**) has been developed for this purpose. Two example matrices and their corresponding storage formats are shown in Fig. 4.4. As shown in Fig. 4.4(b), three RAMs are defined to store the two matrices: (1) two 32-bit wide RAMs  $rmA$  and  $rmB$  to store the value of the matrices entries of sparse matrix  $\mathbf{A}$  and dense matrix  $\mathbf{B}$  in row and column order, respectively; (2) a 20-bit wide RAM  $rmcfg$ , consisting of the first bit  $sel$  is used to label the same column entries in  $rmB$  with '0' or '1', the middle 11-bit  $adB$  to store the address in  $rmB$ , the last 8-bit  $adA$  to store the address in  $rmA$ , due to the fact that sparse matrices containing less number of entries than dense matrices. This storage scheme can be extended to save larger matrix sizes easily.

In Fig. 4.5(a), the three RAMs  $rmA$ ,  $rmB$  and  $rmcfg$  are generated by the above rules. The signals  $adA$ ,  $adB$  and  $adcfg$  are controlled to access the three RAMs sequentially, and the data are sent to Multiplication Accumulator Calculation Unit (**MAC**) in pipeline to perform

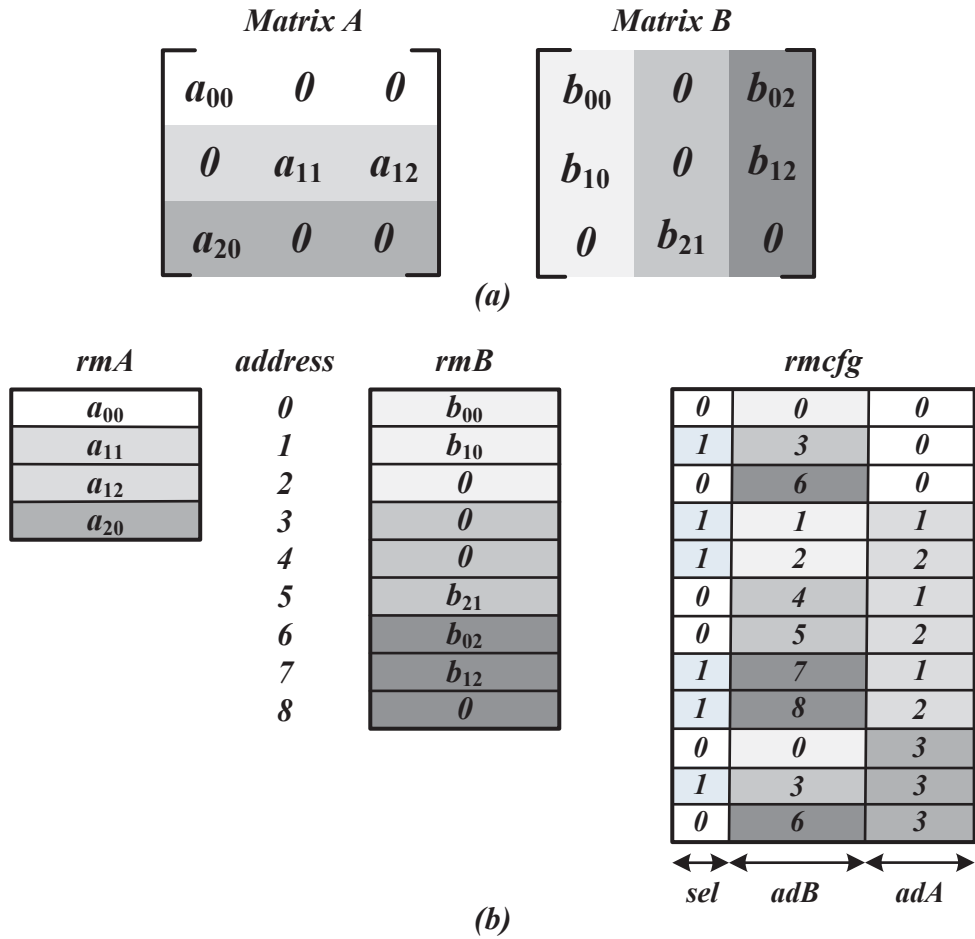
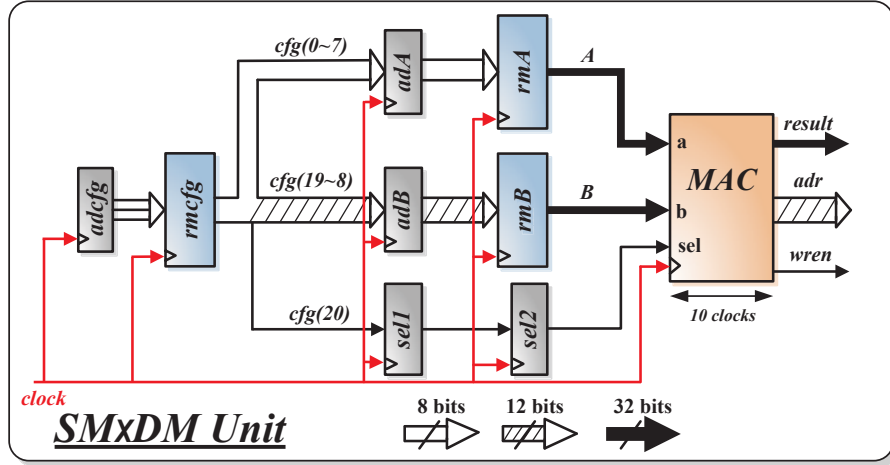


Figure 4.4: Sparse matrix storage format: (a) example matrices; (b) corresponding storage formats.

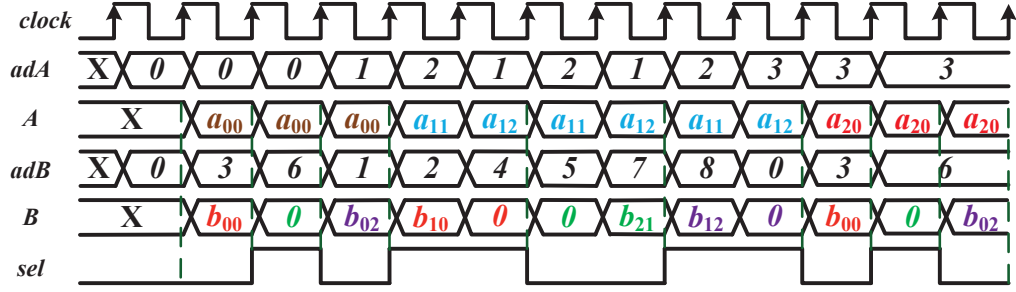
the matrix multiplication. The timing diagram of this operation is shown in Fig. 4.5(b).

The **MAC** unit, as shown in Fig. 4.6 (a), consists of one floating point multiplier, one floating-to-fixed point converter, one fixed-to-floating point converter and two fixed-point adders. The reason to employ a fixed point adder instead of floating point adder here is to increase the maximum implementation frequency when the addition operation is forced to finish in one clock cycle. The fixed point results are converted back to floating point for further computations. And the two fixed point adders with opposite reset signals are used for the pipelining so that there is no stall between two row-column multiplications, as shown in the timing diagram of **MAC** unit in Fig. 4.6 (b). The fixed point result signal  $f_1$  is converted back to floating point signal *result* in two clock cycles, and the signal *adr* increased by one if signal *sel* is changed. The inserted registers are used for synchronization purpose.

The **SMxDM** hardware implementation can also be used to perform dense-sparse, sparse-sparse, sparse-diagonal matrices multiplications. The sparse and diagonal matrices can be



(a)



(b)

Figure 4.5: Pipelined design for the sparse-dense matrix multiplication **SMxDM** unit: (a) hardware design; and (b) timing diagram.

treated as a special dense matrices, and for these matrix multiplications, the  $rmA$ ,  $rmB$  and  $rmcfg$  are generated by similar rules.

#### 4.1.2 Preisach Hysteretic Unit (PHU)

The diagonal matrix  $\mathbf{P}$  is composed of linear entries and nonlinear entries. The linear entries representing the leakage and zero-sequence permeances are constant; however, the nonlinear entries representing the winding and yoke permeances change in every simulation time step, so that a nonlinear evaluation unit is needed to model this phenomenon. In this chapter, again the Preisach model [53] was selected to represent the nonlinearity, in order to further include the hysteresis behaviour.

The major loop hysteresis function is geometrically represented by the following hyperbolic function [53]:

$$\phi_{p.m}(i_m) = \frac{1}{2} \sum_{u=1}^3 a_u [\tanh(b_u i_m) + c_u \sec h^2(b_u i_m)], \quad (4.22)$$

where  $a_u$ ,  $b_u$ ,  $c_u$  are the parameters to define this hyperbolic template,  $i_m = [i_{m1} \ i_{m2} \ \dots]$

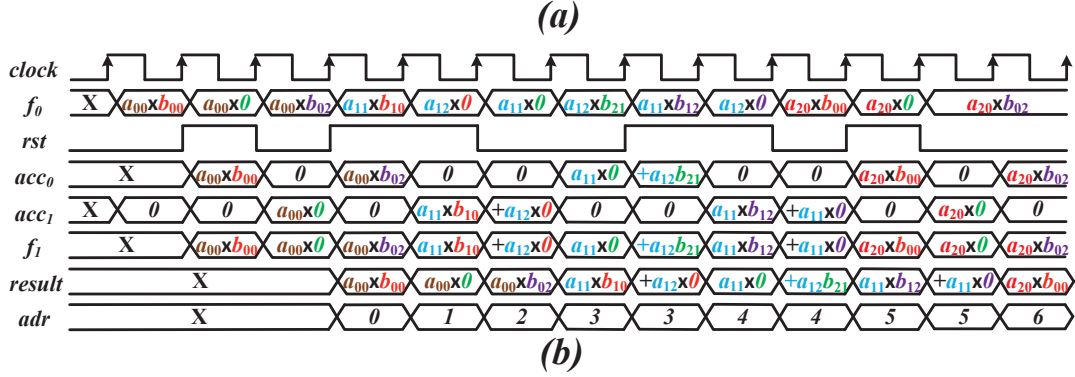
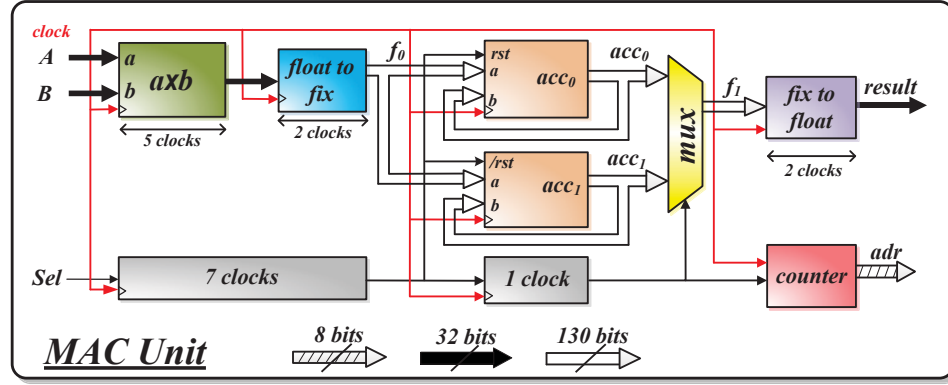


Figure 4.6: Pipelined design for the multiplication accumulator calculation **MAC** unit: (a) hardware design; (b) timing diagram.

$i_{mn_l}]^T$  is magnetization current vector flowing through the nonlinear elements and  $\phi_{p-m} = [\phi_{p-m1} \ \phi_{p-m2} \ \dots \ \phi_{p-mn_l}]^T$  are the fluxes corresponding to this current vector on the major loop. The subscript 'p\_m' stands for Preisach major loop and  $n_l$  stands for the number of nonlinear permeance.

Again, once the major loop function is fixed, all the minor loop trajectories from a reversal point follow this uniform template. In general, Preisach upward  $\phi_{p-u}$  and downward  $\phi_{p-d}$  trajectories from a reversal point  $(i_{rl}, \phi_{rl})$  can be expressed as (4.23) and (4.24), respectively [54], with  $\mathbf{i}_r = [i_{r1} \ i_{r2} \ \dots \ i_{rn_l}]^T$  and  $\phi_r = [\phi_{r1} \ \phi_{r2} \ \dots \ \phi_{rn_l}]^T$ :

$$\phi_{p,u}(\mathbf{i}_m) = -\phi_{p,m}(-\mathbf{i}_m) - \phi_{p,m}(\mathbf{i}_r) + \phi_r + 2\mathbf{K}(-\mathbf{i}_r)\mathbf{K}(\mathbf{i}_m), \quad (4.23)$$

$$\phi_{p,d}(\mathbf{i}_m) = \phi_{p,m}(\mathbf{i}_m) + \phi_{p,m}(-\mathbf{i}_r) + \phi_r - 2\mathbf{K}(\mathbf{i}_r)\mathbf{K}(-\mathbf{i}_m), \quad (4.24)$$

where

$$\mathbf{K} = [K_1 \ K_2 \ \dots \ K_{n_l}],$$

$$K_l(x) = \begin{cases} \sqrt{\phi_{p-m}(-x)}, & x < 0 \\ \frac{\phi_{p-m}(-x) + \phi_{p-m}(x)}{2\sqrt{\phi_{p-m}(x)}}, & x > 0, \end{cases}$$

$$l = 1, 2, \dots, n_l.$$

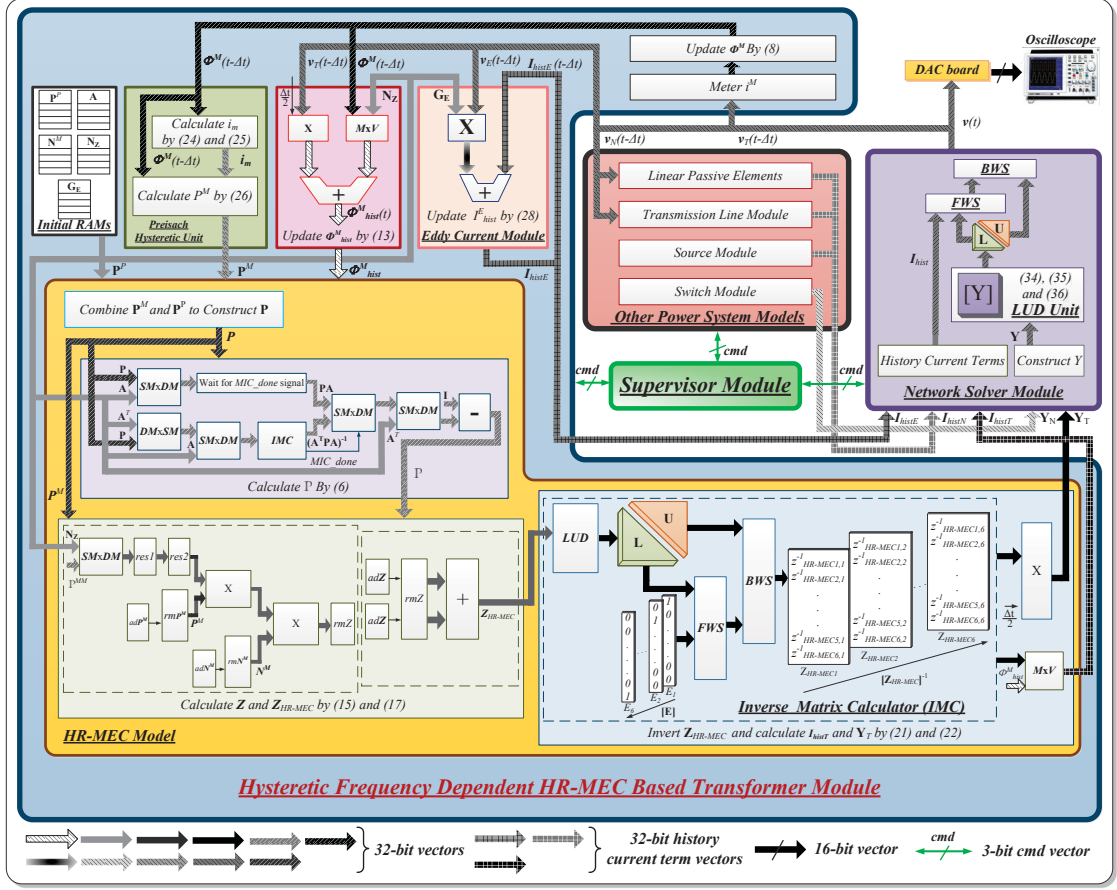


Figure 4.7: Hardware design of the interface between the HR-MEC transformer model and the power system network for transient simulation.

After calculating the vector  $\phi^M$  (composed of  $\phi_{p-u}$  and  $\phi_{p-u}$ ) by using (4.8), the magnetizing current vector  $i_m$  can be obtained from (4.23) and (4.24). The  $k$ th branch nonlinear permeance  $P_k$  can then be calculated as,

$$P_k = \frac{\phi_k^M}{i_{mk} N_k}, \quad (4.25)$$

where  $N_k$  is the number of the  $k$ th winding turns.

To calculate the linear air permeances, the value of leakage impedance  $X_{leakage}$  is necessary, which can be obtained from the short-circuit test or provided by the transformer manufacturer. By assuming the leakage inductance are equally divided amongst the primary and secondary windings, the leakage permeance can be obtained as

$$P_{leakage} = \frac{0.5 X_{leakage}}{2\pi f N_k^2}, \quad (4.26)$$

where  $N_k$  is the number of winding turns.

## 4.2 Power System Network Transient Simulation

### 4.2.1 Interfacing of the HR-MEC Based Transformer Model to the Power System Network

The interfacing of the real-time hysteretic HR-MEC based transformer model to the power system network is shown in Fig. 4.7. The supervisor module sends command signal (*cmd*) to each module to enable it to perform the designed function, and also wait to receive the signals sent back from each module, so that all modules can work either in parallel or sequentially as required by the power system transient algorithm. The transformer parameters saved in the initial RAMs are sent to the proper calculation units involved. The nonlinear permeance matrix  $\mathbf{P}^M$  computed by the Preisach Hysteresis Unit (**PHU**), combined with linear permeance matrix  $\mathbf{P}^P$  gives the overall permeance matrix  $\mathbf{P}$ . After calculating the matrix  $\mathbb{P}$  by (4.6), the matrices  $\mathbf{Z}$  and  $\mathbf{Z}_{(HR-MEC)}$  are computed by (4.14) and (4.16). As indicated by (4.20) and (4.21) the matrix  $\mathbf{Z}_{(HR-MEC)}$  needs to be inverted to obtain the transformer admittance matrix  $\mathbf{Y}_T$  and history term vector  $\mathbf{I}_{histT}$ . A LU-based Inverse Matrix Calculator (IMC) described in Section III(B) is used to invert the matrix. As shown in Fig. 4.7, the entire power system network admittance matrix  $\mathbf{Y}$  is composed of the transformer admittance matrix  $\mathbf{Y}_T$  and the remainder power system network admittance matrix  $\mathbf{Y}_N$ . The history term vector  $\mathbf{I}_{hist}$  is the summation of the transformer history term  $\mathbf{I}_{histT}$ , the eddy current history term  $\mathbf{I}_{histE}$  and the other power system network history term  $\mathbf{I}_{histN}$ . Then the Network Solver Model, consisting of a LU decomposition (**LU**), a forward substitution unit (**FWS**) and a backward substitution unit (**BWS**), is executed to find the power system network nodal voltages  $\mathbf{v}$  which are displayed on the oscilloscope through a digital-to-analog converter (DAC) board and are also sent back to the other power system models, as well as the eddy current model to update their respective history terms. The permeance network's flux term  $\phi^M$  is updated by metering the transformer terminal currents in (4.8) and the history term  $\phi_{hist}^M$  is updated by (4.12).

Fig. 4.8 shows the detailed finite state machine (FSM) for this hardware design. In state  $S_1$  the transformer winding and yoke fluxes are updated, and eddy current module as well as linear passive element and transmission line modules update their history terms  $\mathbf{I}_{histE}$  using (??) and  $\mathbf{I}_{histN}$  in parallel. State  $S_2$  performs two functions in parallel: 1) updates the transformer history term  $\phi_{hist}^M$  using (4.12); 2) calculates the nonlinear permeance values in the PHU. State  $S_3$  calculates matrices  $\mathbb{P}$ ,  $\mathbf{Z}$  and  $\mathbf{Z}_{(HR-MEC)}$  sequentially from (4.6), (4.14) and (4.16), respectively. State  $S_4$  inverts matrix  $\mathbf{Z}_{(HR-MEC)}$  and state  $S_5$  computes the vector  $\mathbf{I}_{histT}$  and the matrix  $\mathbf{Y}_T$  from (4.20) and (4.21) respectively. State  $S_6$  calculates the summation of all the history current terms  $\mathbf{I}_{hist}$  and constructs the entire power system network matrix  $\mathbf{Y}$ . State  $S_7$  performs LU decomposition and forward- and backward-substitutions to obtain the nodal network voltages  $\mathbf{v}$ .



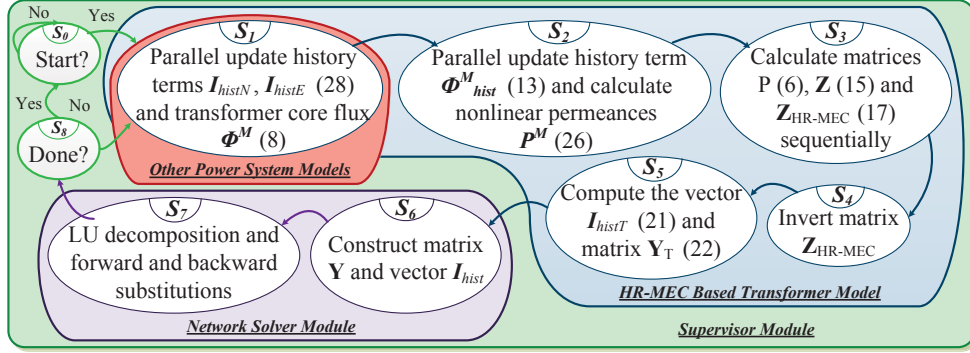


Figure 4.8: Finite state machine of the supervisor module in Fig. 4.7.

## 4.2.2 LU Decomposition Based Network Solver Module

### 4.2.2.1 LU decomposition formula

In the power system transient simulation algorithm, the nodal analysis formulation results in the following equation for each simulation time-step:

$$\mathbf{Y}\mathbf{v} = \mathbf{i} - \mathbf{I}_{hist} = \mathbf{i}_b, \quad (4.27)$$

where  $\mathbf{Y}$  is a  $n \times n$  admittance matrix, and  $\mathbf{v}$ ,  $\mathbf{i}$  and  $\mathbf{I}_{hist}$  are  $n \times 1$  nodal voltage, node injection current and history current vectors, respectively;  $n$  denotes the total number of nodes of the modeled power system. Since this admittance matrix  $\mathbf{Y}$  is not constant but keep changing during the simulation as discussed in Section (2.1.1). So that the per-calculated inverse admittance matrix  $\mathbf{Y}^{-1}$  method, as described in Chapter I, is not suitable under this circumstance, since we cannot store all possible inverse admittance matrices in the simulation program. In this work, instead of inverting the matrix  $\mathbf{Y}$ , the vector of  $\mathbf{v}$  is calculated by an LU-based network solver. Previous works on FPGA-based LU implementations have been reported; however, in some of these works, fixed-point or single-precision floating point are employed [61], which give low accuracy. In the LU solver in [62], part of the factorization is done in a host microprocessor, which increased the amount of communication between the microprocessor and FPGA. The other LU-based solvers [63,64], are developed based on block LU decomposition algorithm to partition the  $\mathbf{Y}$  into several blocks. With these algorithms, the efficiency of the solver is improved by overlapping the communication and the computation; however, the block decomposition algorithm is reported to have numerical problems and can not be used in all cases [65]. The LU-based network solver presented in this work is developed using 32-bit floating point precision for high accuracy and implemented entirely on FPGA without communication with a host processor. The implementation is fully pipelined and paralleled to achieve real-time simulation.

The LU factorization of a  $n \times n$  matrix  $\mathbf{Y}$  into a lower triangular  $n \times n$  matrix  $\mathbf{L}$  and an upper triangular  $n \times n$  matrix  $\mathbf{U}$  can be written as,

$$\mathbf{Y} = \mathbf{LU}. \quad (4.28)$$

The relationship between entries of  $\mathbf{Y}$  and  $\mathbf{L}$  and  $\mathbf{U}$  is given as,

$$l_{i,k} = \frac{y_{i,k} - \sum_{r=1}^{k-1} l_{i,r} u_{r,k}}{u_{k,k}}, \quad (4.29)$$

where  $k = 2, \dots, n-1$  and  $i = k+1, \dots, n$ , and

$$u_{k,j} = y_{k,j} - \sum_{r=1}^{k-1} l_{k,r} u_{r,j}, \quad (4.30)$$

where  $k = 2, \dots, n$  and  $j = k, \dots, n$ .

By expanding the summation components in (4.29) and (4.30), the LU decomposition of  $n \times n$  matrix can be finished in  $n$  iterations, and there are two steps in each iteration. In the  $k$ th iteration, the first step is to find the  $k$ th column of the matrix  $\mathbf{L}$ ,  $l_{i,k}$  by

$$l_{p,k} = \frac{y_{p,k}}{y_{k,k}}, \quad p = k, \dots, n, \quad (4.31)$$

and the  $k$ th row of the matrix  $\mathbf{U}$  is find by

$$u_{k,q} = y_{k,q}, \quad q = k, \dots, n, \quad (4.32)$$

In the second step of each iteration, the  $(n-k) \times (n-k)$  sub-matrix  $\mathbf{Y}'$  at the right down corner of matrix  $\mathbf{Y}$  is updated by,

$$y_{p,q} = y_{p,q} - l_{p,k} u_{k,q}, \quad p, q = k+1, \dots, n \quad (4.33)$$

After finishing LU factorization, (4.27) can rewrite as,

$$\mathbf{LU}v = i_b \quad (4.34)$$

with

$$\mathbf{U}v = x, \quad (4.35)$$

(4.34) can be further rewritten as,

$$\mathbf{L}x = i_b. \quad (4.36)$$

The entries in vector  $x$  can be obtained by applying forward substitute to (4.36) as

$$x_k = i_{b,k} - \sum_{r=1}^{k-1} l_{k,r} x_r. \quad (4.37)$$

Then the entries of vector  $v$  can be calculated by applying backward substitute to (4.35) as,

$$v_k = \frac{x_k - \sum_{r=m+1}^n u_{k,r} v_r}{u_{k,k}}. \quad (4.38)$$

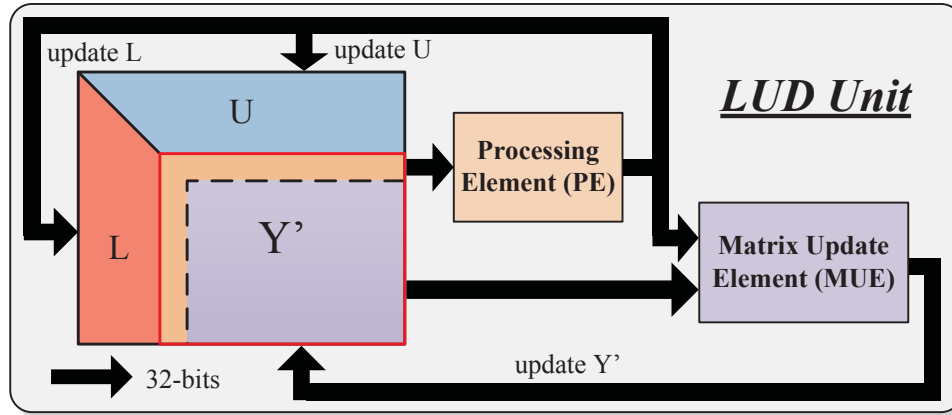


Figure 4.9: LU decomposition hardware design.

#### 4.2.2.2 LUD unit hardware implementation

In the LU Decomposition (**LUD**) Unit, as shown in Fig. 4.9, the entries in the first row and the first column of the sub-matrix  $Y'$  are sent to the Processing Element (**PE**) to calculate the elements of  $L$  and  $U$  by (4.31) and (4.32), respectively. Then the calculated entries of  $L$  and  $U$ , as well as the rest of the entries of sub-matrix  $Y'$  are sent to the Matrix Update Element (**MUE**) to update the matrix  $Y'$ . A 5x5 LU factorization hardware design example is illustrated in Fig. 4.10. Based on the (4.28) and (4.29), the **LUD** unit is consisted by three types of operator: divider, multiplier and subtracter. At the beginning of each iteration, the entries in the first row and the first column of remainder matrix are read and processed by dividers, the calculated results are sent to the multiplier. At the end of each iteration, the matrix  $Y'$  is updated by subtracting its previous entries. All the computation are fully executed in pipelined and paralleled to achieve small latency. The Finite State Machine (FSM) for the **LUD** Unit is shown in Fig. 4.11, and the signal  $LUgo$  and  $LUdone$  are command signals to communicate with control modules in the top level. After obtaining the lower and upper matrices, **FWS** and **BWS**, as shown in Fig. 4.12 and Fig. 4.13, are used to calculate the vector of  $x$  and  $v$  in (4.36) and (4.35) respectively. The **FWS** will be executed firstly by reading the entries in lower matrix  $L$  and vector  $i$ . The temporary results vector  $x$  will be stored as the inputs to the **BWS**. After receiving all the data, the **BWS** read the entries in upper matrix  $U$ , and perform the computation process described in (4.38) in fully paralleled and pipelined.

### 4.2.3 LU-based Inverse Matrix Calculator

#### 4.2.3.1 Inverse matrix calculator formula

A LU-based Inverse Matrix Calculator (**IMC**) is also developed in this paper to compute the inverse matrices in (4.6), (4.20) and (4.21). The mathematic theory behind this calculator is that the multiplication of a  $n \times n$  square matrix  $A$  and its inverse matrix  $A^{-1}$  must be an

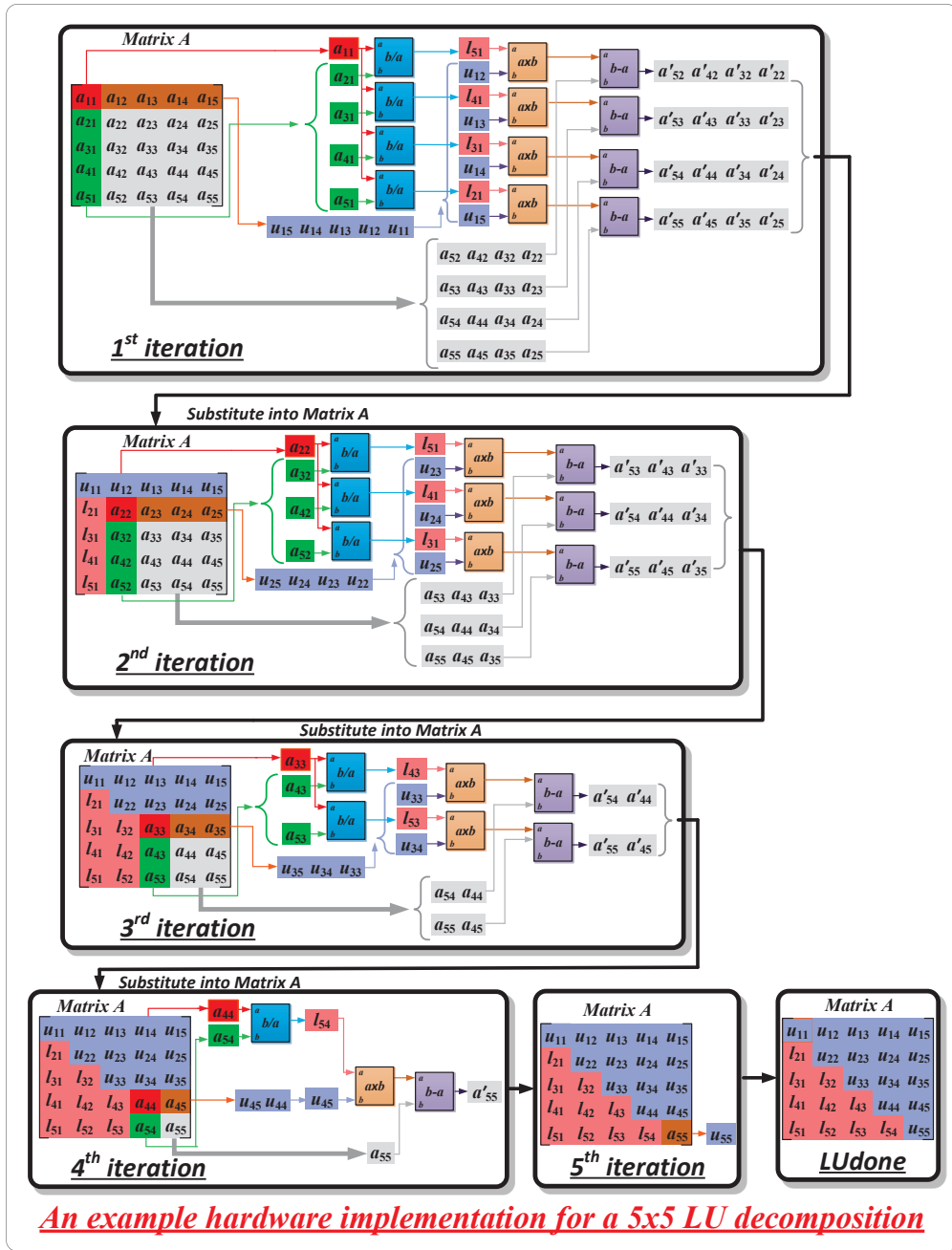


Figure 4.10: A 5x5 example LUD hardware implementation and its data path.

identity matrix E,

$$AA^{-1} = E, \quad (4.39)$$

which can be expressed as,

$$A[A_1^{-1} \ A_2^{-1} \ \dots \ A_n^{-1}] = [E_1 \ E_2 \ \dots \ E_n], \quad (4.40)$$

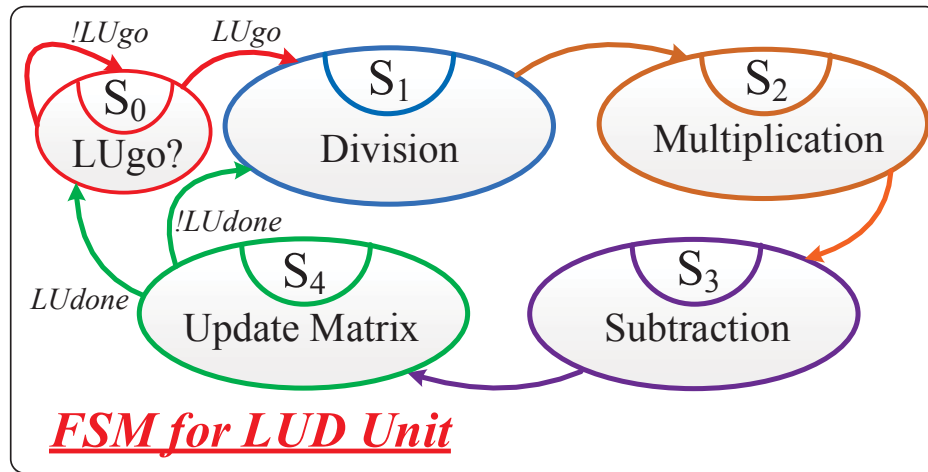


Figure 4.11: The FSM for LUD unit.

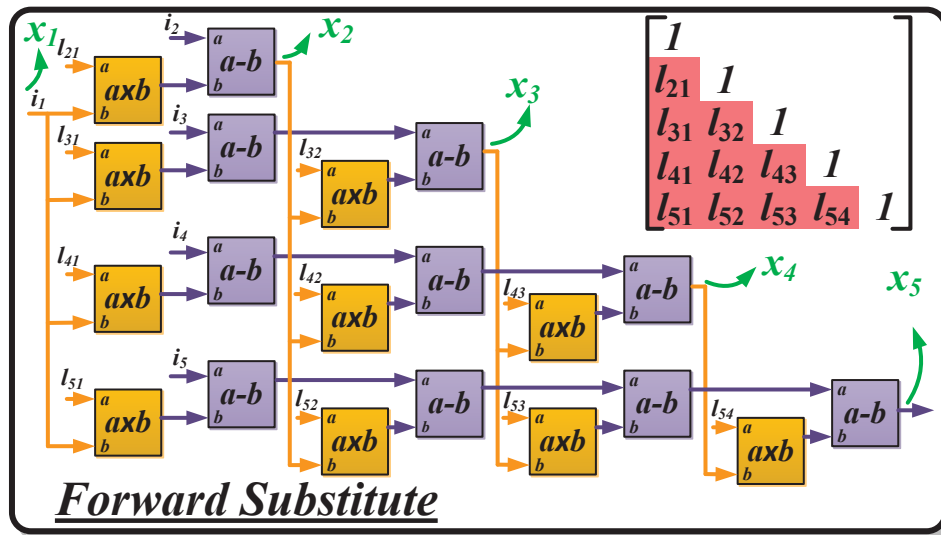


Figure 4.12: Forward substitute hardware design.

where  $A_k^{-1}$  and  $E_k$  is the  $k$ th column of matrices  $A^{-1}$  and  $E$ , respectively ( $k = 1, 2, \dots, n$ ). By applying forward- and backward- substitutions to the following equation,

$$AA_k^{-1} = E_k, \quad (4.41)$$

the vector  $A_k^{-1}$  can be obtained.

#### 4.2.3.2 Inverse matrix calculator hardware implementation

As shown in Fig. 4.14, the matrix  $A$  undergoes LU factorization only once. Then the vectors  $E_k$  ( $k = 1, 2, \dots, n$ ) are pushed into the input port of **FWS** sequentially and the vectors  $A^{-1}$  are obtained at the output port of **BWS**.

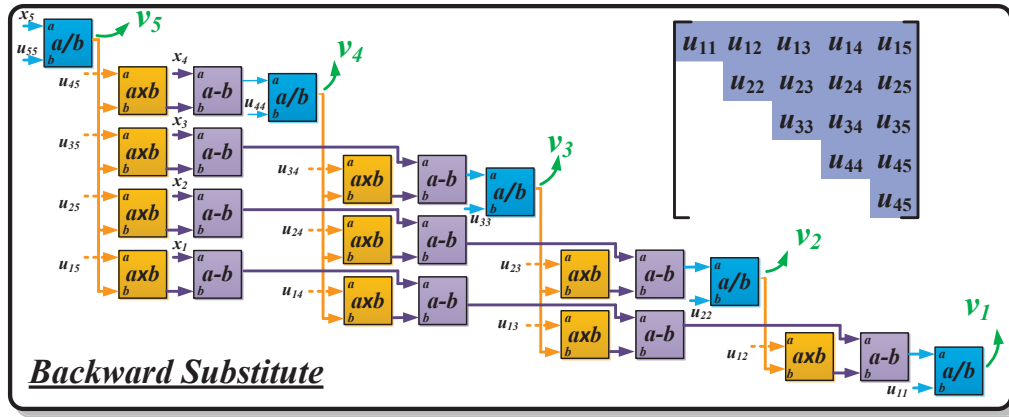


Figure 4.13: Backward substitute hardware design.

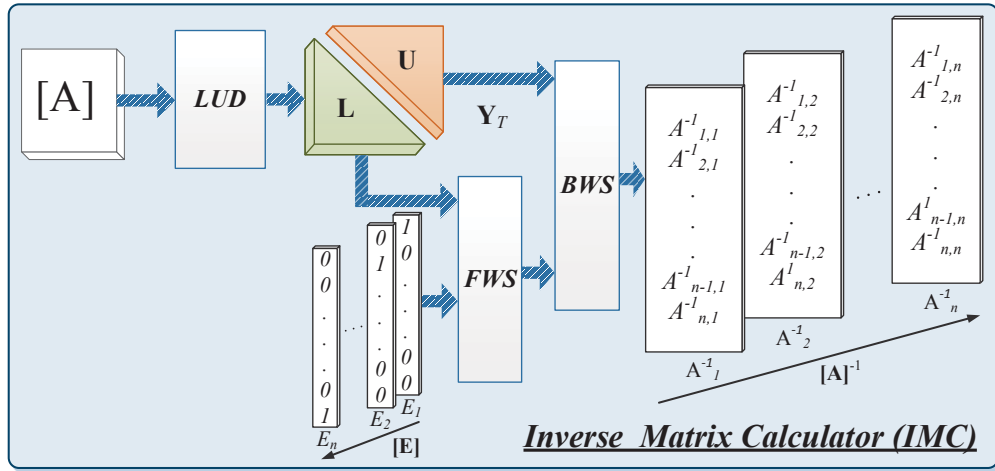


Figure 4.14: The Inverse Matrix Calculator **IMC** hardware implementation.

### 4.3 Case Study I

#### 4.3.1 Validation by Finite Element Simulation

A 3-D finite element model is built in the JMAG<sup>®</sup> Designer for a three-phase 187.5 MVA transformer to validate the hardware emulation of the developed HR-MEC model. The ratings of the transformer, and the geometric parameters are given in the Appendix A, half transformer core and coil model was developed in the FEM software, and the full transformer model was simulated by setting the symmetry boundary and using the full model and circuit conversion features of JMAG<sup>®</sup>. The size of the element is 0.15 m, and there are 37750 elements and 7753 nodes to build the mesh of the half transformer. The nonlinearity is solved employing the Newton-Raphson method and the maximum number of iteration is set to 50 to ensure convergence in every time-step. The leakage inductance value can be input in the coupled external electric circuit as  $L_{leakage}$  the in the HR-MEC

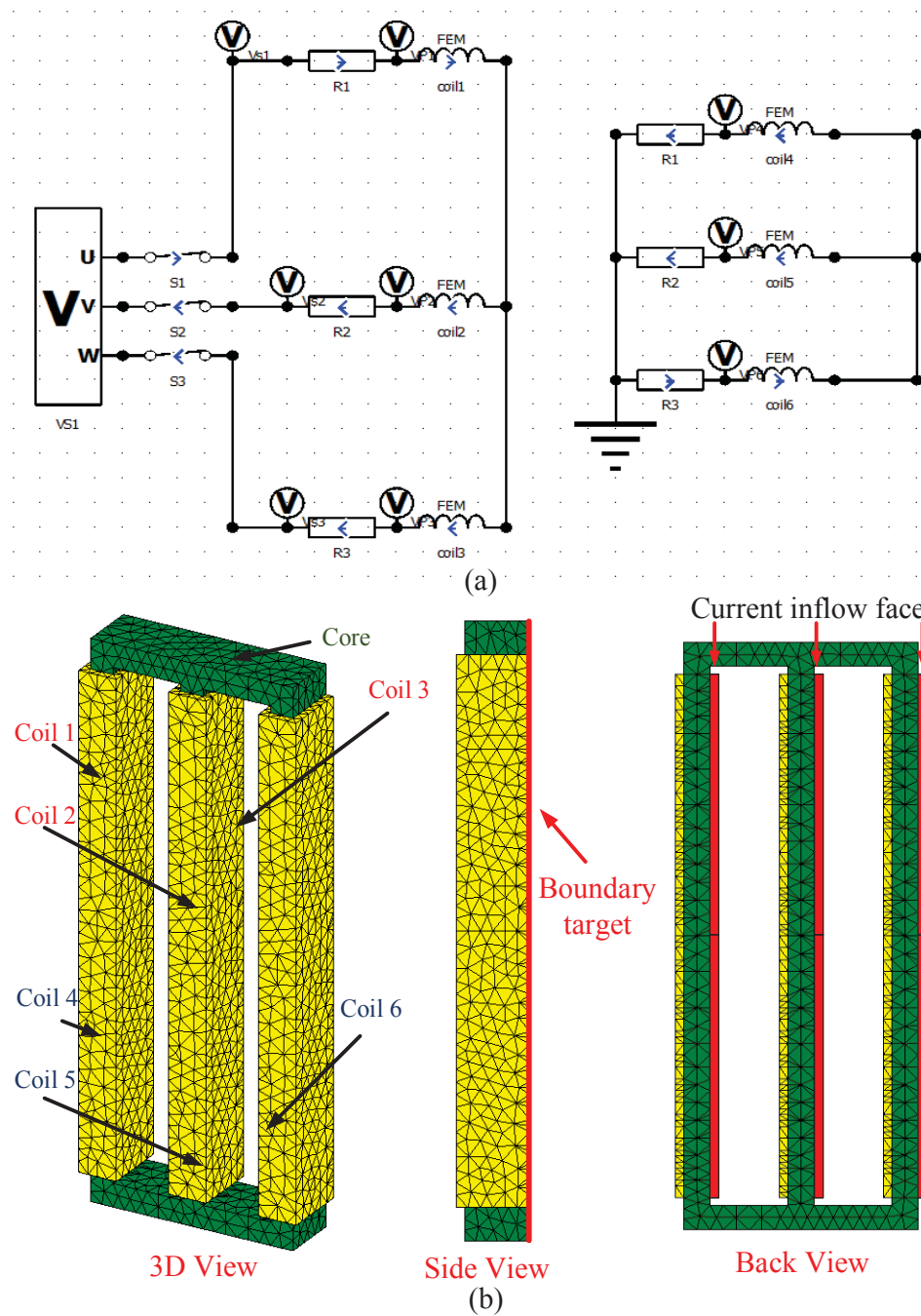


Figure 4.15: The built JMAG transformer and its corresponding studied circuit.

model the leakage permeance value can be carried out as

$$P_{leakage}/N_k^2, \quad (4.42)$$

where  $N_k$  is the number of winding turns.

The JMAG model and the corresponding studied circuit is given as Fig.4.15, where  $R_1$ ,  $R_2$  and  $R_3$  are large value resistors to represent opening at the secondary side of the

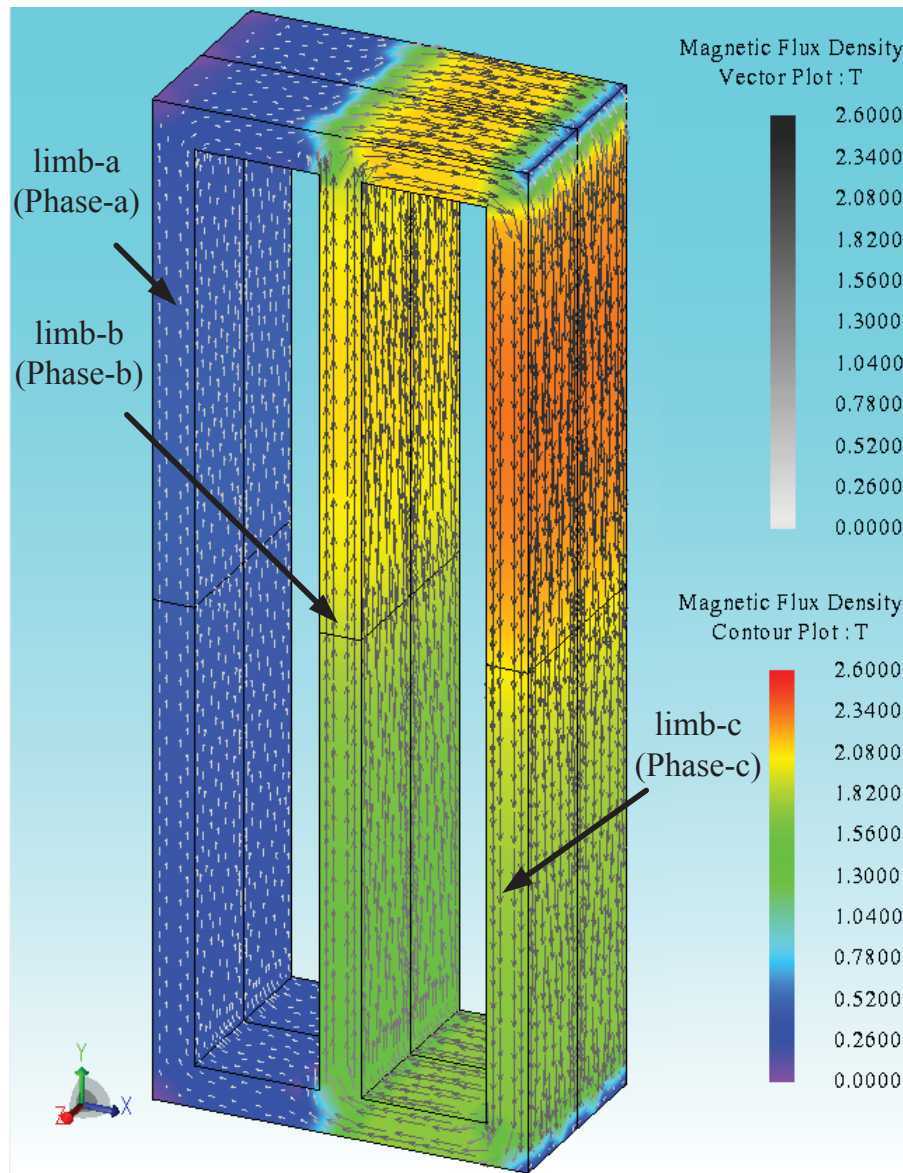


Figure 4.16: Magnetic flux density distribution of the modeled transformer in *JMAG® Designer* at the peak of phase-c current.

transformer. The primary side of the built transformer is fed by a 3-phase voltage source with the secondary side open. After energizing the transformer, an inrush current flows through the coil of the primary side, and the simulated magnetic flux density contour and vector plot is shown in Fig. 4.16 when the current in phase-c reaches its peak value. As can be seen, the direction of the flux lines in limb-c is different from the direction of limb-a and -b, because the current directions are different between phase-c and phase-a and phase-b. The flux lines in limb-b and -c flow into the limb-c, which results in the limb-c being the most saturated limb and the maximum flux density is close to 2.6 T. This highly saturated limb-c results in the phase-c inrush current being higher compared to the



inrush currents in phase-a and-b. The JMAG<sup>®</sup> Designer took almost 13 hours to run a 4 s simulation with the step time  $\Delta t = 500 \mu s$  on a PC featured by Intel<sup>®</sup> i7 CPU and 8 GB memory. The precomputed method is another candidate approach to predict the behaviours of transformer, which computes the values in advance in JMAG<sup>®</sup> Designer and stores them in LUTs to save simulation time. However, it is very difficult to build up large enough LUTs to save all the results for the transformer under all operating conditions, due to the long simulation time and massive on board storage, especially under the transient conditions.

Table 4.1: Module level hardware resource utilization

	<b>Slice Registers</b>	<b>Slice LUTs</b>	<b>BRAM/ FIFO</b>	<b>DSP48E1</b>	<b>BUFG</b>
<b>HR-MEC Module</b>	79160 (13%)	122255 (40.3%)	85 (2.8%)	746 (26.6%)	12 (37.5%)
<b>Network Solver Module</b>	15996 (2.6%)	26221 (8.6%)	21 (0.7%)	151 (5.4%)	1 (3.1%)
<b>Transmission Line Module</b>	2600 (0.4%)	6434 (2.1%)	22 (0.7%)	32 (1.1%)	3 (9.3%)
<b>Source Module</b>	851 (0.1%)	9787 (3.2%)	1 (0.0%)	6 (0.0%)	0 (0.0%)

Table 4.2: Module level operator implementation

	<b>Adder/Subtractor</b>	<b>Multiplication</b>	<b>Divider</b>
<b>HR-MEC Module</b>	147	146	34
<b>Network Solver Module</b>	37	36	11
<b>Transmission Line Module</b>	6	10	0
<b>Source Module</b>	2	1	0

### 4.3.2 Energization Transients and Steady-state Waveforms

In the case study, a power system network shown in Fig. 4.17 is emulated to show the accuracy and effectiveness of the proposed real-time nonlinear frequency-dependent HR-

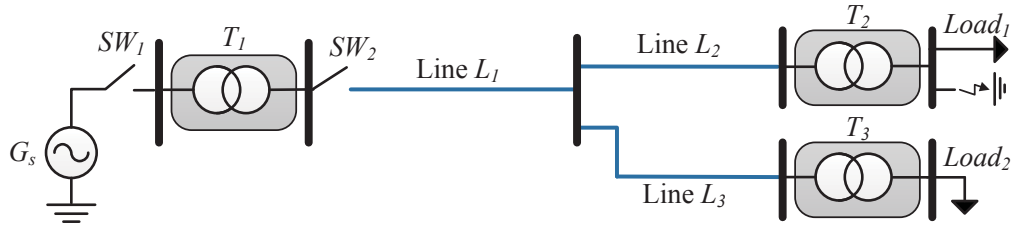


Figure 4.17: Single-line diagram of the modeled power system for case study with HR-MEC based transformer model.

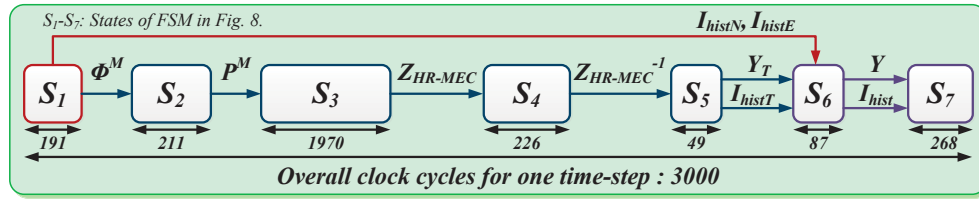


Figure 4.18: Latencies for one emulation time-step for the HR-MEC based transformer model emulation.

MEC transformer model. The network consists of three transformers ( $T_1, T_2, T_3$ ), modeled by HR-MEC transformer model, and three transmission lines ( $L_1, L_2, L_3$ ), modeled by distributed parameter line model. The primary side of the transformer  $T_1$  is fed by a 3-phase voltage source  $G_s$  through a circuit breaker  $SW_1$ . The system parameters are listed in the Appendix. The real-time detailed nonlinear transformer electromagnetic transient model and other power system network models were emulated on the Xilinx<sup>®</sup> Virtex-7 VC707 XC7VX485T device, and the module level hardware resource utilizations and number of operator implementations are summarized in Table 4.1. The maximum clock frequency for this design was 90 MHz. The detailed latencies of the model for each simulation time-step is also summarized and shown in Fig. 4.18. In this figure, the latency is labelled for each FSM in Fig. 4.8, and the overall latency is 3000 clock cycles. With the maximum frequency of 90 MHz, the execution time for one simulation time-step is  $34 \mu s$ . The Coregen<sup>®</sup> provided by ISE<sup>®</sup> is used to generate the basic 32-bit floating point operations such as adder and subtractor which are further employed to implement the other complex functional units such as **SMxDM** and **LUD**. The maximum frequency of the generated operations can reach up to 500 MHz, however for the HR-MEC based real-time transformer model the maximum frequency is 90 MHz. This is mainly due to the massive operations involved in each clock cycle results in high complexity of the hardware implementation which increase the latency between two flip-flops. Besides, the other sequential statements in the VHDL textural program such as *if* and *case* statement can also increase the complexity of the hardware implementation and result a decreased maximum frequency can be achieved in this hardware emulation model.

For the energization transient study, initially the circuit breaker  $SW_1$  and  $SW_2$  are

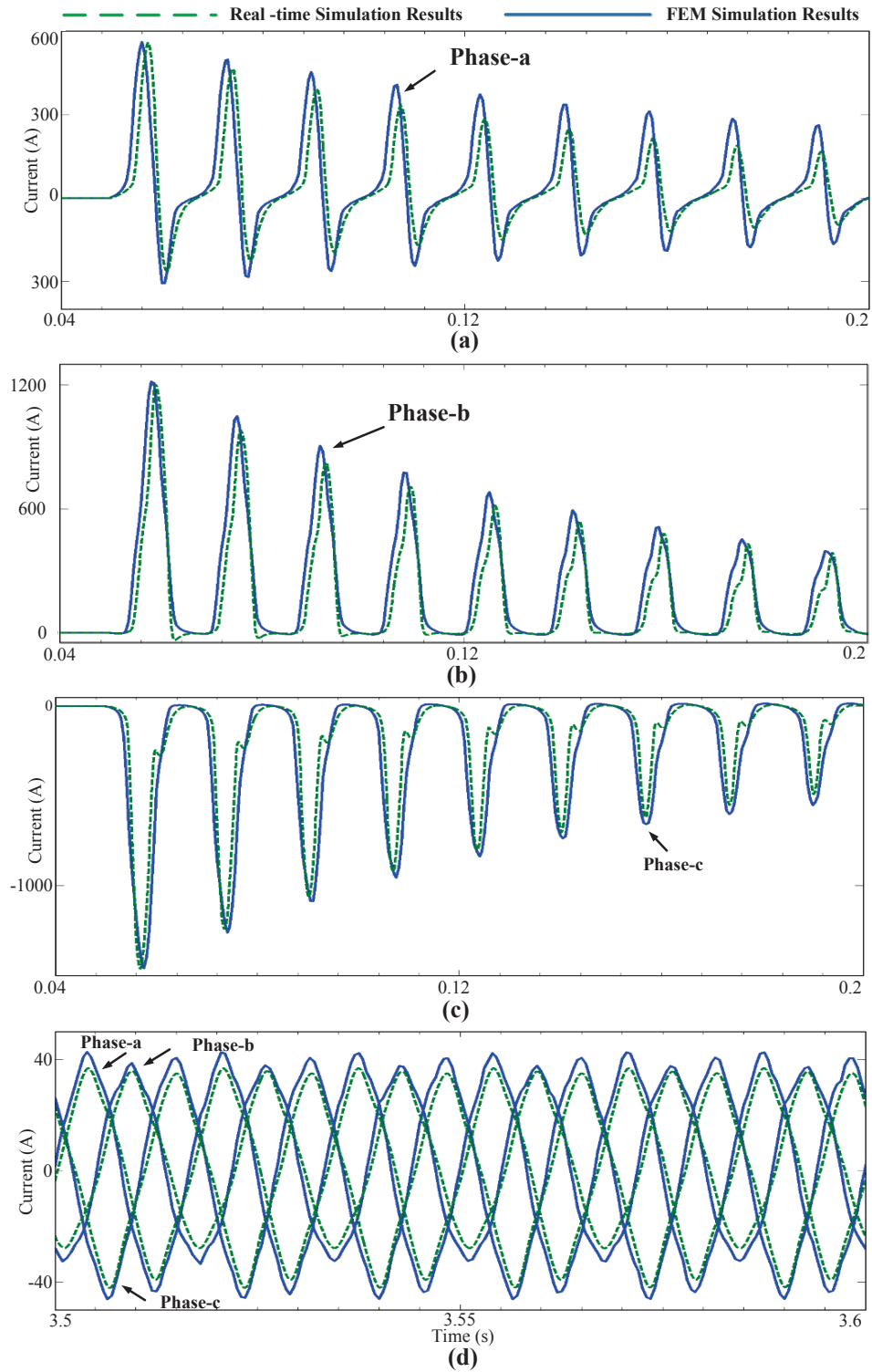
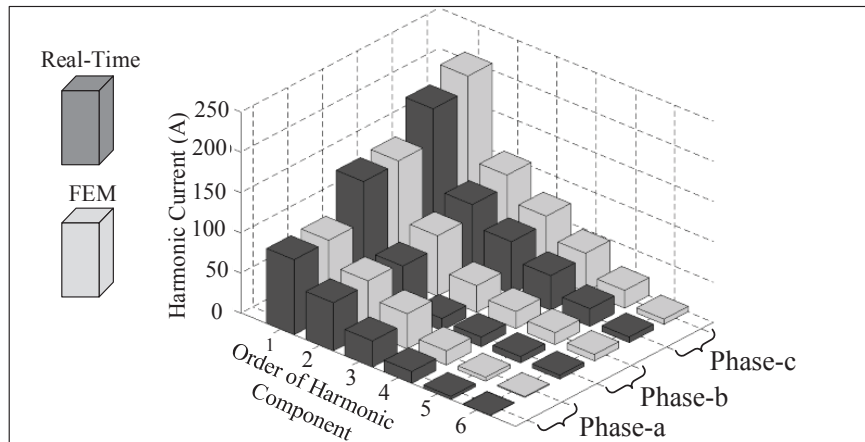
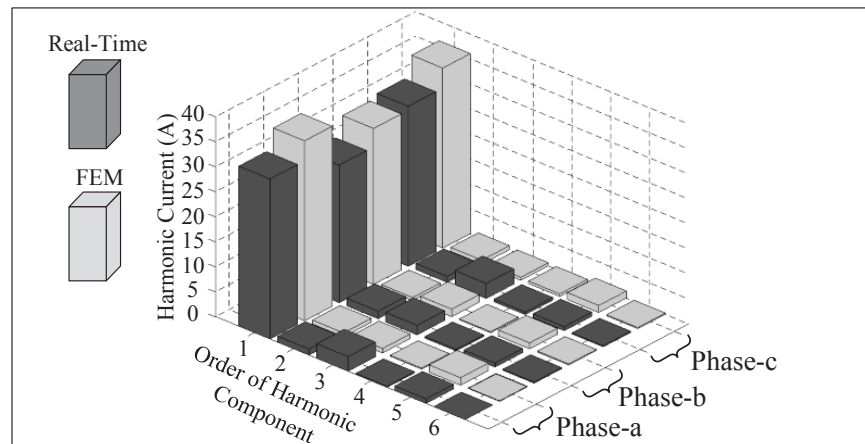


Figure 4.19: Real-time and JMAG<sup>®</sup> Designer simulation results: (a)-(c) Energization transient current; (d) steady-state current.



(a)



(b)

Figure 4.20: Real-time and JMAG<sup>®</sup> Designer simulation results: (a) frequency analysis of the energization transient current; (b) frequency analysis of the steady-state current.

opened. The energization current transient is simulated by closing  $SW_1$  at time  $t = 0.05$  s, and the real-time three-phase breaker current results are captured and plotted along with the JMAG<sup>®</sup> Designer simulation results in Fig. 4.19 (a)-(c). In these waveforms, the phase-b and -c currents are seen to be of larger magnitude compared to phase-a current, due to the different values of supplied voltage at the instant of energization. All the currents decay to steady-state after 3.5 s as shown in Fig. 4.19 (d). While there is close agreement of the wave shape of inrush currents between the real-time and FEM results, small discrepancies exist in the magnitudes. That is because for the HR-MEC model the initial fluxes in each limbs can not be all zero, otherwise the entries in matrix  $\mathbf{P}$  would be all zero, which in turn make the term  $\mathbf{A}^T \mathbf{P} \mathbf{A}$  in (4.6) can not be inverted. Thus in the HR-MEC model, the initial fluxes are assigned to very small values close to zero, however, in the FEM simulation the initial fluxes are all zero. Another reason is that even though the real-time model is based on a high resolution magnetic equivalent circuit approach, the mesh size is still

much smaller compared to that of the 3-D FEM simulation. Despite these discrepancies the proposed HR-MEC model provides acceptably accurate results and runs in real-time compared to the long execution time of the FEM simulation.

The frequency analysis of the transient state from  $t = 0.04$  s to  $t = 0.2$  s and the steady state from  $t = 3.5$  s to  $t = 4$  s are also given in Fig. 4.20 (a) and (b) as the order of harmonic components with respect to harmonic current magnitude. As can be seen, under steady-state only the fundamental is the dominant component, however for the transient-state, the harmonic components of order 2, 3 have larger values than the other order harmonic components, which is due to the high nonlinear distortion current in each phase when the transformer is energized. In both transient and steady-state, the harmonic currents values of FEM simulation results are somewhat different than those of the real-time simulation results, due to the reason given above. The maximum differences under transient and steady-state are both happened in the first order of phase-c, which are 9% and 11%, respectively.

The real-time primary winding-limb flux-linkage simulation results are also captured and plotted on top of the FEM simulation results both under transient and steady-state in Fig. 4.21 (a)-(c) and (d), respectively. As expected the flux is in phase with the three-phase breaker current waveforms and the flux in limb-c (phase-c) reach its negative peak value first and the flux in limb-a (phase-a) has the lowest peak value during the transient period. Again close agreements can be observed between the real-time and FEM results.

## 4.4 Case Study II

### 4.4.1 Fault Transients

The fault transient simulation was achieved by creating a three-phase ground fault at the secondary winding of  $T_2$  at  $t_1 = 2$  s and the voltage and current waveforms at the primary winding of  $T_3$  are captured in real-time on the oscilloscope. As shown in Fig. 4.22 (a) both waveforms are highly distorted, and the distorted voltage magnitude is up to 121% of the steady-state voltage, and the currents are unbalanced in the three phases. The fault voltage and current waveforms decay to steady-state after  $t_2 = 2.16$  s.

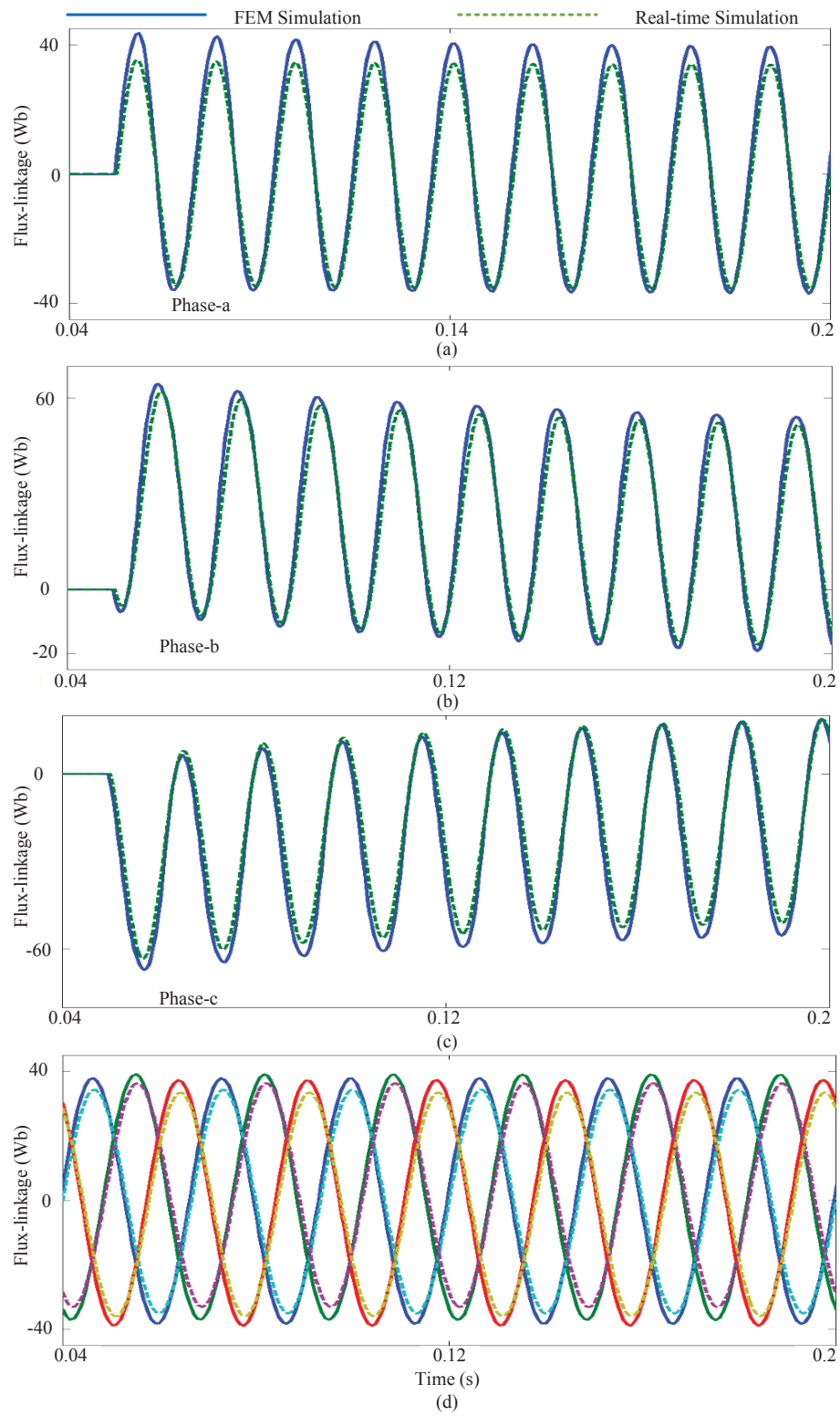


Figure 4.21: Real-time and JMAG<sup>®</sup> Designer simulation results: (a)-(c) flux-linkage under transient steady; (d) steady-state flux-linkage.

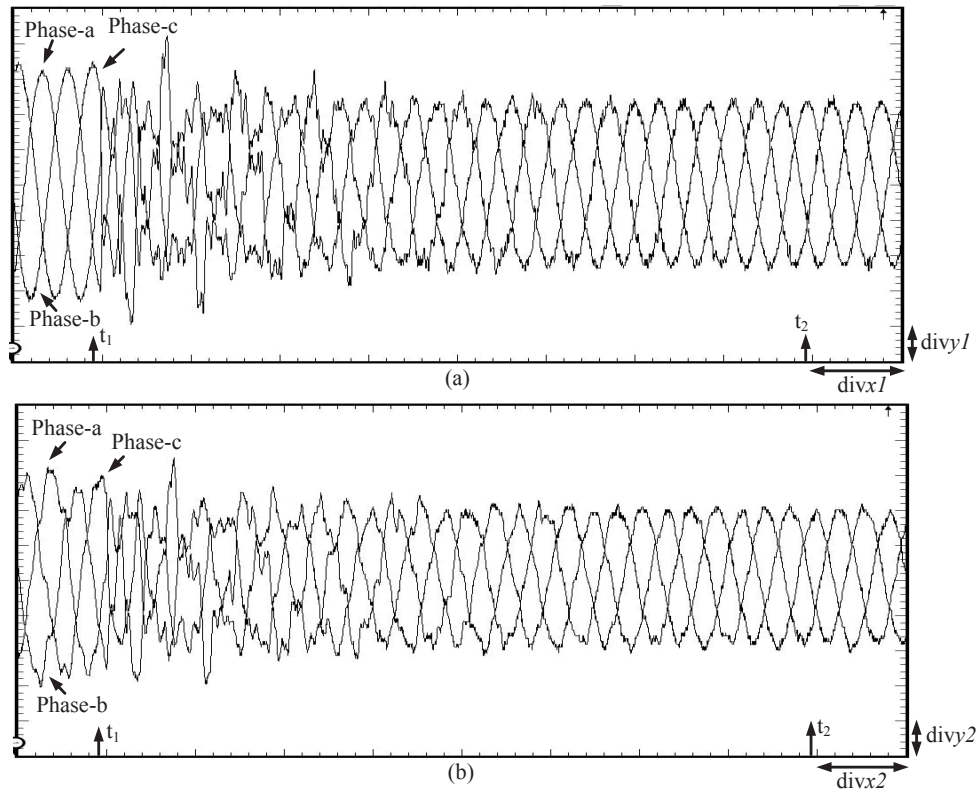


Figure 4.22: Ground fault at  $t_1 = 2s$  real-time simulation results (a) transient voltage waveform (1  $\text{divx1}=20$  ms; 1 $\text{divy1}= 20$  kV); (b) transient current waveform (1  $\text{divx2}=20$  ms; 1 $\text{divy2} = 39$  A ).

## 4.5 Summary

This chapter proposed a FPGA based digital hardware emulation of a high-resolution magnetic equivalent circuit (HR-MEC) nonlinear frequency-dependent real-time power transformer model. This model is fully paralleled and deeply pipelined in hardware to achieve real-time simulation with low hardware resource utility. The proposed real-time model is able to accurately predict the transformer behaviour under both transient and steady-state conditions. The low latency of the proposed HR-MEC model makes it applicable in closed-loop hardware-in-loop simulation setups to test the power system protection and control systems. This FPGA-based model can also be employed to design a power transformer, due the geometry-oriented features, within much shorter time compared to FEM, can also predict the local information of the transformer, such as flux distribution in the core, which are the advantages over a simpler transformer model such as matrix representation model.



# 5

## Magnetic Equivalent Circuit Based Real-time Series Transformer Model for HIL Emulation

This chapter proposes a real-time high-fidelity nonlinear magnetic equivalent circuit (MEC) based ST electromagnetic transient model on FPGA for HIL emulation. The ST can be a robust and cost-effective solution to relieve grid congestion due to increased installation of renewables. The ST consists of a 12 winding transformer and tap changer that can regulate the power flow through a transmission line by injecting a series connected controllable voltage. The geometry-based MEC approach [59, 66] is employed to model the multi-winding transformer consisting of magnetomotive force (MMF) sources and permeance elements. The nonlinear permeance elements necessitate a time-varying transformer nodal admittance matrix which in turn require the entire power system matrix to be updated in every emulation time-step. The hysteresis and eddy current behavior are modeled based on Preisach theory [54] and frequency-dependent equivalent network [55, 67], respectively. All of these details increase the computational burden of the high-fidelity electromagnetic transient model and put upward pressure on the emulation time-step. A 3-D finite element (FE) transformer model and its coupled external circuit is developed in JMAG<sup>®</sup> for the validation purpose. The power flow control emulation revealed that this proposed model has the capability to regulate the real and reactive power on both single and multiple lines. In this ST model, there are four nonlinear permeances in each core limb carrying four different currents and the secondary winding is series connected with transmission line; however, its derivation procedures came from the first principles and can also be applied to other winding configurations, making the proposed real-time model easy to modify for emulating other power transformers.



## 5.1 Introduction

Growing demand by a plethora of sensitive loads, stringent reliability constraints, and heightened environmental concerns that favour greater integration of renewables, have created perfect conditions for a highly stressed and dyspeptic transmission grid. Congestion of lines under these conditions has become commonplace with increased probability of major outages leading to blackouts [68–70]. While long-term transmission planning and expansion still remains the prudent solution, a near term economical alternative might be the strategic placement of power flow controllers to relieve transmission bottlenecks [71,72]. The unified power flow controller (UPFC) [73,74] is one such solution which consists of a series and a shunt converter that can generate a compensation voltage to insert in series with the transmission line. The compensation voltage is fully controllable both in magnitude and phase angle with respect to the line voltage, for independent regulation of the active and reactive power flowing through the line in both directions. However, high installation and operation costs have hitherto prevented the UPFC's widespread implementation. The Sen transformer (ST) proposed in [75], can address the same objectives but with a lower cost, higher reliability, and efficiency [76]. The ST consists of a single-core three-limb three-phase transformer and tap changers. The multi-winding transformer has three primary windings and nine secondary windings. By setting the tap changers into different combinations, the ST can inject a compensation voltage with controllable magnitude and phase angle much same the way like the UPFC, thereby providing an independent active power and reactive power flow control. The concept of the smart power flow controller (SPFC) was introduced in [77] which offered the ST with the option of either a low-cost electromechanical design with mechanical tap changers or a power electronics based design using thyristor based tap changers.

A detailed electromagnetic transient model of the ST in real-time is necessary to study its impact on the host power system. It would also allow rapid testing and prototyping of new control algorithms under hardware-in-the-loop (HIL) conditions. A real-time transient ST model was developed on the CPU in [78], and an advanced tap-changing algorithm was proposed in [79]; however, this real-time ST model was developed based on a lumped electrical circuit model, which ignored the detailed transformer core geometry and flux paths in the core. Furthermore, in this model nonlinear core saturation was included using a piece-wise linear approximation, which is prone to numerical oscillations; complex behavior such as hysteresis and eddy current transients were entirely omitted. The increased computational complexity of detailed electromagnetic transient ST model makes the CPU an unlikely implementation platform for real-time HIL emulation.

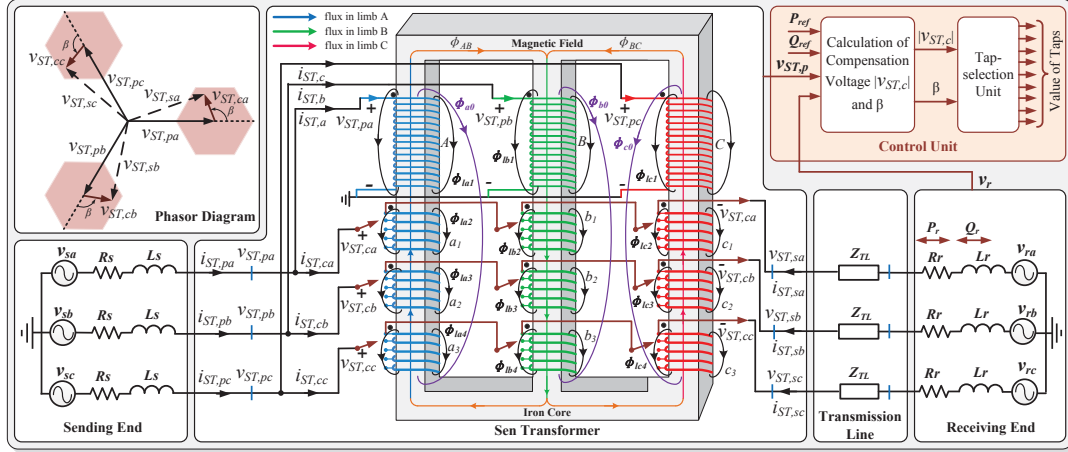


Figure 5.1: Schematic of the ST and its interconnection to the transmission network.

## 5.2 Geometrical Sen Transformer EMT Model

### 5.2.1 Sen Transformer Operating Principle

The three-phase, three-limb ST circuit schematic and its interconnection to the transmission line is shown in Fig. 5.1. At the sending-end the three-phase transmission line supplies voltages,  $V_{ST,pa}$ ,  $V_{ST,pb}$  and  $V_{ST,pc}$  to the ST primary windings, which are connected in Y and in shunt with the transmission line. The nine secondary windings:  $a_1$ ,  $a_2$ , and  $a_3$  are placed in the first limb;  $b_1$ ,  $b_2$ , and  $b_3$  are placed in the second limb; and  $c_1$ ,  $c_2$  and  $c_3$  are placed in the third limb. The three secondary windings which are placed in different limbs are connected in series with the transmission line and inject a compensation voltage, e.g., windings  $a_1$ ,  $b_1$  and  $c_1$  for the compensation voltage  $V_{ST,ca}$  in phase- $a$ ; windings  $a_2$ ,  $b_2$  and  $c_2$  for the compensation voltage  $V_{ST,cb}$  in phase- $b$ ; windings  $a_3$ ,  $b_3$  and  $c_3$  for the compensation voltage  $V_{ST,cc}$  in phase- $c$ . By using the on-load tap changers, the active number of turns of the nine secondary windings can be configured so that the induced compensation voltage can be controlled both in magnitude and phase angle independently. If the  $120^\circ$  phase-shift compensation voltage is series connected with the transmission line, as shown in Fig. 5.1, the previous sending-end voltage (the ST primary side voltage),  $v_{ST,p}$ , is modified to a new effective sending-end voltage (the ST secondary side voltage),  $v_{ST,s}$ , as can be seen from the top lefthand corner of Fig. 5.1. The number of turns of the secondary windings can be classified into three groups:  $a_1$ ,  $b_2$  and  $c_3$  ( $GP_1$ );  $b_1$ ,  $c_2$  and  $a_3$  ( $GP_2$ ); and  $c_1$ ,  $a_2$  and  $b_3$  ( $GP_3$ ). The windings in one group have the same tap setting, however, one group may have a different tap setting from others. The major flux paths consisting of core and leakage fluxes in the yoke and limb are also depicted in Fig. 5.1.

## 5.2.2 Tap-selection Algorithm

The *Control Unit* takes the measurements of the sending end voltage,  $V_{ST,p}$ , and the receiving end voltage,  $v_r$ , from the transmission network as well as the required power level,  $P_{ref}$  and  $Q_{ref}$  as inputs. The calculated compensation voltage magnitude  $v_{ST,c}$  and the phase angle  $\alpha$  are further forwarded to the *Tap-setting Unit* to generate the combinations of the tap settings for each secondary winding. Further information regarding the tap-setting algorithm and its implementation can be found in [79].

## 5.2.3 High-fidelity Nonlinear MEC Based ST Model

The three-phase, three-limb MEC based ST representation, is shown in Fig. 5.2. Each limb consist of four types of components: 1) MMF sources,  $N_A i_{ST,a}$ ,  $N_B i_{ST,b}$  and  $N_C i_{ST,c}$  generated by the current flow through the primary windings in phases  $a$ ,  $b$  and  $c$ , respectively;  $N_{a1} i_{ST,sa}$ ,  $N_{a2} i_{ST,sb}$ ,  $N_{a3} i_{ST,sc}$ ,  $N_{b1} i_{ST,sa}$ ,  $N_{b2} i_{ST,sb}$ ,  $N_{b3} i_{ST,sc}$ ,  $N_{c1} i_{ST,sa}$ ,  $N_{c2} i_{ST,sb}$  and  $N_{c3} i_{ST,sc}$  are generated by the current flow through the secondary windings in phases  $a$ ,  $b$  and  $c$ , respectively; 2) nonlinear permeances,  $P_A$ ,  $P_B$  and  $P_C$  represent nonlinear iron core permeances for ST primary windings in phases  $a$ ,  $b$  and  $c$ , respectively;  $P_{a1}$ ,  $P_{a2}$ ,  $P_{a3}$ ,  $P_{b1}$ ,  $P_{b2}$ ,  $P_{b3}$ ,  $P_{c1}$ ,  $P_{c2}$  and  $P_{c3}$  represent nonlinear iron core permeances for the ST secondary windings in phases  $a$ ,  $b$  and  $c$ , respectively; 3) linear permeances,  $P_{la1}$  to  $P_{la4}$ ,  $P_{lb1}$  to  $P_{lb4}$  and  $P_{lc1}$  to  $P_{lc4}$  represent the ST leakage air permeances in phases  $a$ ,  $b$  and  $c$ , respectively; 4) linear permeances  $P_{a0}$ ,  $P_{b0}$ ,  $P_{c0}$  represent ST zero-sequence permeances in phases  $a$ ,  $b$  and  $c$ , respectively. In addition, there are also two nonlinear iron permeances  $P_{ab}$  and  $P_{bc}$  that represent ST yoke between phase- $a$  and phase- $b$ , and phase- $b$  and phase- $c$ , respectively. Based on Ampere's and Gauss' laws [80], the relationship between the branch fluxes and the winding currents can be written as:

$$\phi = \mathbb{P}\mathbf{N}\mathbf{i}, \quad (5.1)$$

where

$$\mathbb{P} = \mathbf{I} - \mathbf{P}\mathbf{A}(\mathbf{A}^T\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^T. \quad (5.2)$$

$\phi$  and  $\mathbf{i}$  are  $n \times 1$  vectors of branch fluxes and winding currents,  $\mathbf{P}$  and  $\mathbf{N}$  are  $n \times n$  diagonal matrices of branch permeances and number of winding turns,  $\mathbf{I}$  is the  $n \times n$  identity matrix, and  $\mathbf{A}^T$  is the node-branch connection matrix, respectively. By partitioning the permeance network into two branch sets,  $\phi^M$ , for the branches with MMF sources and permeance elements, and  $\phi^P$ , for branches with only permeance elements, (5.1) can be rewritten as:

$$\phi^M = \mathbb{P}^{MM}\mathbf{P}^M\mathbf{N}^M\mathbf{i}^M, \quad (5.3)$$

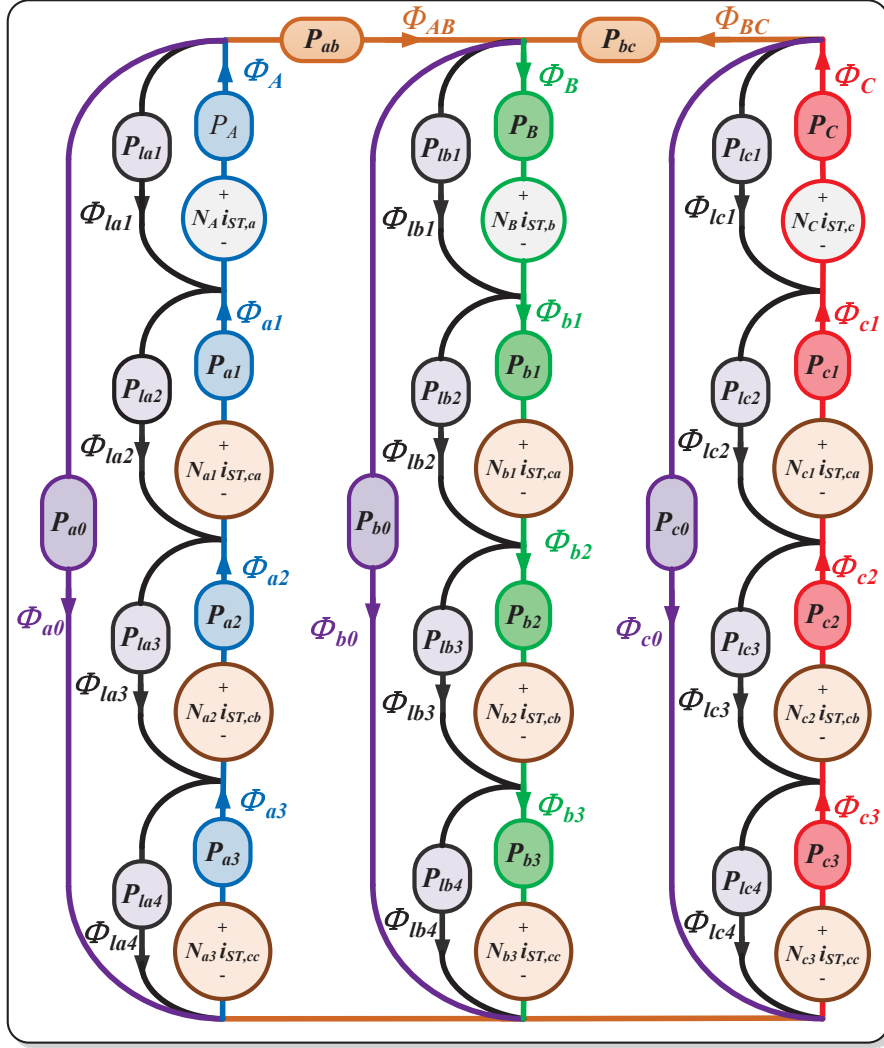


Figure 5.2: High-fidelity magnetic equivalent circuit representation for the ST iron core.

where  $\mathbb{P}^{MM}$  is a 12x12 sub-matrix of  $\mathbb{P}$ .  $\phi^M$  and  $i^M$  are 12x1 vectors, and  $\mathbf{P}^M$  and  $\mathbf{N}^M$  are diagonal 12x12 matrices, given as:

$$\begin{aligned} \phi^M &= [\phi_A(t) \ \phi_B(t) \ \phi_C(t) \ \phi_{a1}(t) \ \phi_{b1}(t) \ \phi_{c1}(t) \\ &\quad \phi_{a2}(t) \ \phi_{b2}(t) \ \phi_{c2}(t) \ \phi_{a3}(t) \ \phi_{b3}(t) \ \phi_{c3}(t)]^T, \\ i^M &= [i_{ST,a}(t) \ i_{ST,b}(t) \ i_{ST,c}(t) \ i_{ST,sa}(t) \ i_{ST,sa}(t) \ i_{ST,sa}(t) \\ &\quad i_{ST,sb}(t) \ i_{ST,sb}(t) \ i_{ST,sb}(t) \ i_{ST,sc}(t) \ i_{ST,sc}(t) \ i_{ST,sc}(t)]^T, \\ \mathbf{P}^M &= \text{diag}\{P_A(t) \ P_B(t) \ P_C(t) \ P_{a1}(t) \ P_{b1}(t) \ P_{c1}(t) \\ &\quad P_{a2}(t) \ P_{b2}(t) \ P_{c2}(t) \ P_{a3}(t) \ P_{b3}(t) \ P_{c3}(t)\}, \\ \mathbf{N}^M &= \text{diag}\{N_A(t) \ N_B(t) \ N_C(t) \ N_{a1}(t) \ N_{b1}(t) \ N_{c1}(t) \\ &\quad N_{a2}(t) \ N_{b2}(t) \ N_{c2}(t) \ N_{a3}(t) \ N_{b3}(t) \ N_{c3}(t)\}. \end{aligned}$$

From Faraday's law the winding terminal voltages can be written in matrix format as,

$$\mathbf{v}_{ST} = \mathbf{N}_Z \frac{d}{dt} \boldsymbol{\phi}^M(t), \quad (5.4)$$

where  $\mathbf{N}_Z$  is a 6x12 winding turns matrix.  $\mathbf{v}_{ST}$  is a 6x1 ST terminal voltage vector, given as:

$$\mathbf{v}_{ST} = [v_{ST,pa}(t) \ v_{ST,pb}(t) \ v_{ST,pc}(t) \ v_{ST,sa}(t) \ v_{ST,sb}(t) \ v_{ST,sc}(t)]^T,$$

and

$$\mathbf{N}_Z = \begin{bmatrix} N_A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_B & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N_C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ N_A & 0 & 0 & N_{a1} & N_{b1} & N_{c1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_B & 0 & 0 & 0 & 0 & N_{a2} & N_{b2} & N_{c2} & 0 & 0 & 0 \\ 0 & 0 & N_C & 0 & 0 & 0 & 0 & 0 & 0 & N_{a3} & N_{b3} & N_{c3} \end{bmatrix}.$$

The discrete-time difference equations of (5.4) can be obtained by applying Trapezoidal rule,

$$\mathbf{N}_Z \boldsymbol{\phi}^M(t) = \frac{\Delta t}{2} \mathbf{v}_{ST}(t) + \boldsymbol{\phi}_{hist}^M(t), \quad (5.5)$$

where  $\Delta t$  is the simulation time-step, with  $\boldsymbol{\phi}_{hist}^M(t)$  being the 6x1 history vector term expressed as:

$$\boldsymbol{\phi}_{hist}^M(t) = \frac{\Delta t}{2} \mathbf{v}_{ST}(t - \Delta t) + \mathbf{N}_Z \boldsymbol{\phi}^M(t - \Delta t). \quad (5.6)$$

Substituting (5.3), the left hand side of (5.5) can be rewritten as,

$$\mathbf{N}_Z \boldsymbol{\phi}^M(t) = \mathbf{Z} \mathbf{i}^M, \quad (5.7)$$

where  $\mathbf{Z}$  is a 6x12 matrix given as:

$$\mathbf{Z} = \mathbf{N}_Z \mathbb{P}^{MM} \mathbf{P}^M \mathbf{N}^M. \quad (5.8)$$

Further (5.7) can be rewritten as:

$$\mathbf{N}_Z \boldsymbol{\phi}^M(t) = \mathbf{Z} \mathbf{i}^M = \mathbf{Z}' \mathbf{i}^{M'}, \quad (5.9)$$

with  $\mathbf{Z}'$  being the 6x6 impedance matrix and  $\mathbf{i}^{M'}$  being the 6x1 current vector, given as:

$$\mathbf{Z}' = \begin{bmatrix} Z_{1,1} & \dots & Z_{1,3} & Z_{1,4} + Z_{1,5} + Z_{1,6} & \dots & Z_{1,10} + Z_{1,11} + Z_{1,12} \\ Z_{2,1} & \dots & Z_{2,3} & Z_{2,4} + Z_{2,5} + Z_{2,6} & \dots & Z_{2,10} + Z_{2,11} + Z_{2,12} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ Z_{6,1} & \dots & Z_{6,3} & Z_{6,4} + Z_{6,5} + Z_{6,6} & \dots & Z_{6,10} + Z_{6,11} + Z_{6,12} \end{bmatrix},$$

and

$$\mathbf{i}^{M'} = [i_{ST,a}(t) \ i_{ST,b}(t) \ i_{ST,c}(t) \ i_{ST,sa}(t) \ i_{ST,sb}(t) \ i_{ST,sc}(t)]^T.$$

By noting that the ST current vector  $\mathbf{i}_{ST}$  is expressed as,

$$\mathbf{i}_{ST} = [i_{ST,pa}(t) \ i_{ST,pb}(t) \ i_{ST,pc}(t) \ i_{ST,sa}(t) \ i_{ST,sb}(t) \ i_{ST,sc}(t)]^T,$$

and

$$i_{ST,pa}(t) = i_{ST,a}(t) - i_{ST,sa}(t),$$

$$i_{ST,pb}(t) = i_{ST,b}(t) - i_{ST,sb}(t),$$

$$i_{ST,pc}(t) = i_{ST,c}(t) - i_{ST,sc}(t).$$

Equation (5.5) can be further rewritten as

$$\mathbf{Z}_{ST}\mathbf{i}_T(t) = \frac{\Delta t}{2}\mathbf{v}_{ST}(t) + \boldsymbol{\phi}_{hist}^M(t), \quad (5.10)$$

where  $\mathbf{Z}_{ST}$  is the 6x6 matrix given as,

$$\mathbf{Z}_{ST} = \begin{bmatrix} Z'_{1,1} & Z'_{1,2} & Z'_{1,3} & Z'_{1,4} + Z'_{1,1} & Z'_{1,5} + Z'_{1,2} & Z'_{1,6} + Z'_{1,3} \\ Z'_{2,1} & Z'_{2,2} & Z'_{2,3} & Z'_{2,4} + Z'_{2,1} & Z'_{2,5} + Z'_{2,2} & Z'_{2,6} + Z'_{2,3} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ Z'_{6,1} & Z'_{6,2} & Z'_{6,3} & Z'_{6,4} + Z'_{6,1} & Z'_{6,5} + Z'_{6,2} & Z'_{6,6} + Z'_{6,3} \end{bmatrix}.$$

Then, the Norton equivalent formula for MEC based ST is given,

$$\mathbf{i}_{ST}(t) = \mathbf{Y}_{ST}(t)\mathbf{v}_{ST}(t) + \mathbf{I}_{hist,ST}(t), \quad (5.11)$$

where

$$\mathbf{I}_{hist,ST}(t) = \mathbf{Z}_{ST}^{-1}\boldsymbol{\phi}_{hist}^M(t), \quad (5.12)$$

and

$$\mathbf{Y}_{ST}(t) = \frac{\Delta t}{2}\mathbf{Z}_{ST}^{-1}. \quad (5.13)$$

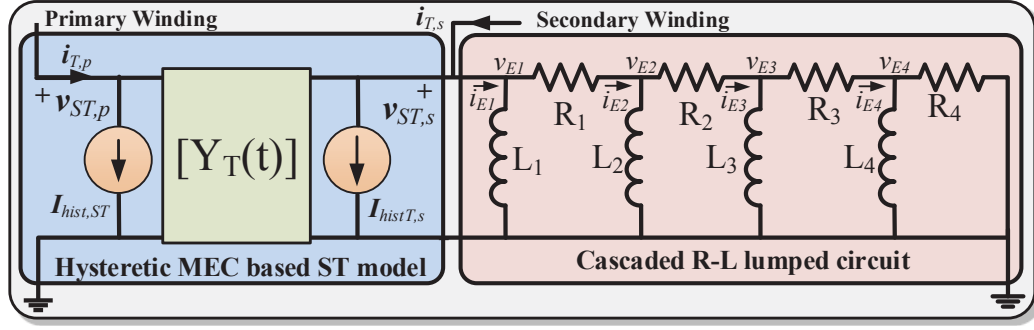


Figure 5.3: High fidelity MEC-based ST model interfaced with cascaded equivalent circuit.

The time-varying matrix  $\mathbf{Y}_{ST}(t)$ , due to the nonlinear permeance matrices  $\mathbb{P}^{MM}$  and  $\mathbf{P}^M$  in (5.8), causes the overall power system admittance matrix  $\mathbf{Y}$  to be time-varying as well. Therefore, a LU decomposition needs to be performed to invert the matrix  $\mathbf{Y}$  in every simulation time-step and then to find the final network solution by using the nodal analysis method.

The eddy current model is again represented by connecting externally cascade R-L equivalent circuit interfacing to the MEC-based ST model. Then, the overall detailed frequency dependent nonlinear ST model is as shown in Fig. 5.3

## 5.3 High-fidelity Nonlinear MEC-based ST Hardware Emulation

### 5.3.1 Network Transient Emulation with Embedded ST

The hardware emulation of the ST and other power system components, such as sources, passive elements, transmission lines and switch modules is depicted in Fig. 5.4, which fully exploits pipelining and parallelism to achieve real-time emulation. Each emulation time-step can be finished in 5 stages, and while the stages are executed sequentially, all the modules or units within any stage are executed in parallel. All the calculations inside the hardware modules and units are fully pipelined and paralleled to achieve the lowest latency and resource consumption.

In Stage 1, the Preisach hysteretic unit (PHU) is designed to evaluate the ST nonlinear iron core permeance matrix  $\mathbf{P}^M$  by reading the ST core fluxes  $\phi^M$ . Combining  $\mathbf{P}^M$  with the linear permeance matrix  $\mathbf{P}^P$  gives the overall permeance matrix  $\mathbf{P}$ , which is then involved in the computation of the matrix  $\mathbb{P}$  using (5.2). The eddy current module, as well as the transmission line and passive element modules in parallel update their history current terms by reading the node voltage vectors,  $v_E$  and  $v_O$ , respectively. The source module is designed to output the source voltage by reading a sinusoidal function look up table (LUT) implemented in it. The switch module sends the network admittance matrix  $\mathbf{Y}_N$  to the network solver module, combined with the ST admittance matrix  $\mathbf{Y}_{ST}$  to carry out the finally

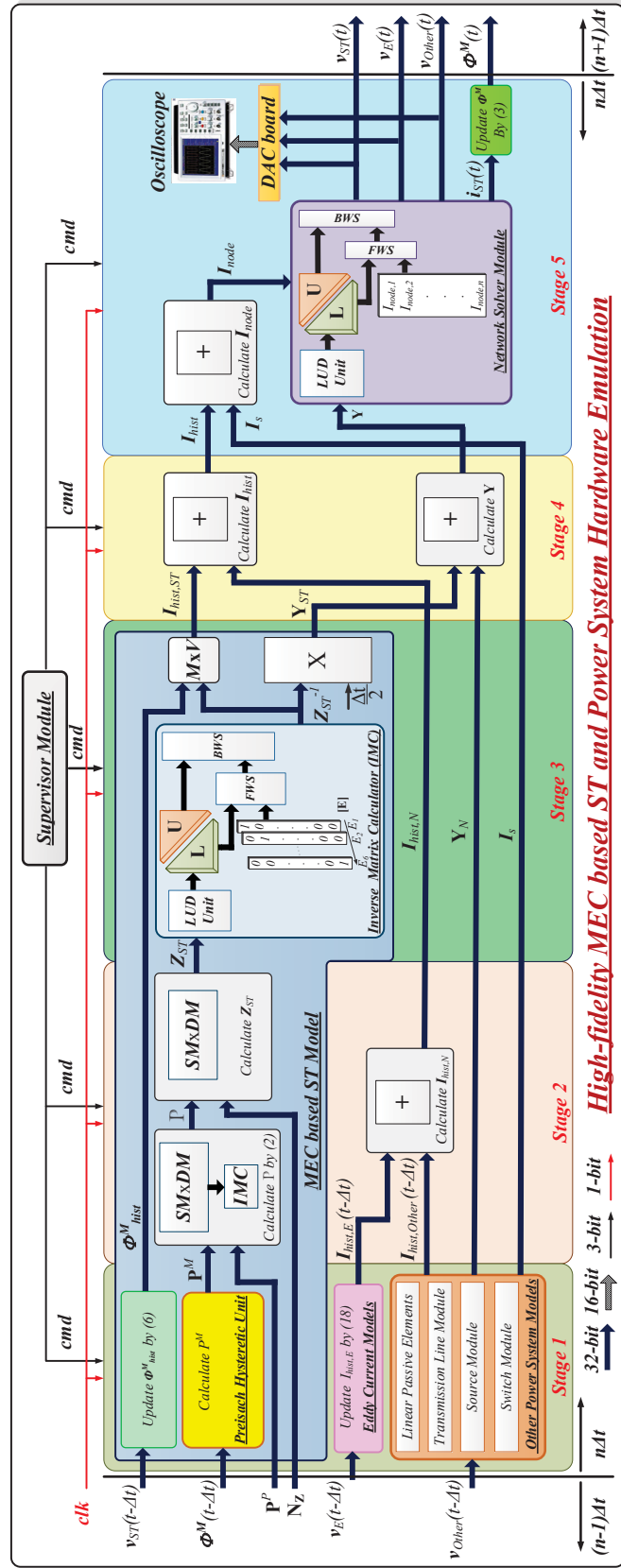


Figure 5.4: Pipelined and paralleled hardware design architecture during one time-step of emulation for the ST and the host power system.



node voltage calculation. To efficiently perform the sparse and dense matrix-matrix and matrix-vector multiplications, such as in (5.2) and (5.12), a sparse-dense matrix multiplication unit (**SMxDM**) and matrix-vector multiplication unit (**MxV**) were designed respectively, in this work. In order to perform the matrix inversion operation in (5.2), (5.12) and (5.13), an LU based Matrix Inverter Calculation (**IMC**) was implemented in the ST module, as described in Sec. III (B). The ST admittance matrix  $\mathbf{Y}_{ST}$  is combined with the network admittance matrix  $\mathbf{Y}_N$  to obtain the overall matrix  $\mathbf{Y}$ . Similarly, the ST history current vector  $I_{hist,ST}$  is combined with the network history current terms  $I_{hist,N}$  and the source current injection vector  $I_s$  resulting in the  $I_{node}$  vector. Then the overall admittance matrix  $\mathbf{Y}$  and the injection current vector  $I_{node}$  are forwarded to the Network Solver Module to finally find the network node voltages, consisting of  $v_{ST}$ ,  $v_E$  and  $v_O$ . The LU based decomposition (**LUD**), forward substitution unit (**FWS**) and backward substitution unit (**BWS**) are implemented in the network solver module, and the vector  $I_{node}$  is pushed into the **FWS** unit input ports for the nodal solution.

Fig. 5.5 shows the detailed finite state machine (FSM) for the MEC based ST and host power system hardware design. In the state  $S_1$ , the calculations in Stage 1 are executed including the transformer nonlinear iron core permeances, the eddy current module, as well as the passive element and transmission line modules update their history terms  $I_{hist,E}$  using (3.41) and  $I_{hist,O}$  in parallel. The Stage 2 is split into two states,  $S_2$  and  $S_3$ . The state  $S_2$  performs two functions in parallel: 1) calculates matrix  $\mathbb{P}$  using (5.2), and 2) calculates network history terms  $I_{hist,N}$  by combining  $I_{hist,E}$  and  $I_{hist,O}$ . State  $S_3$  calculates the ST admittance matrix  $\mathbf{Z}_{ST}$  using (5.10). Stage 3 is also split into two states:  $S_4$  and  $S_5$ . In  $S_4$  the matrix  $\mathbf{Z}_{ST}$  goes through the **IMC** unit to compute the inverse matrix  $\mathbf{Z}_{ST}^{-1}$ . In state  $S_5$  the ST history term vector  $I_{hist,ST}$  and admittance matrix  $\mathbf{Y}_{ST}$  are computed in parallel using (5.12) and (5.13), respectively. The Stage 4 is fully implemented in  $S_6$ , to parallel compute the summation of the all the history currents  $I_{hist}$  and construct the entire power system network admittance matrix  $\mathbf{Y}$ . The stage 5 is divided into two states,  $S_7$  and  $S_8$ . In state  $S_7$ , the current injection vector  $I_{node}$  is calculated and the network solver module is executed to compute the network node voltages  $v$ . Finally state  $S_8$  updates the transformer winding and yoke fluxes using (5.3).

## 5.4 Case Studies And Real-time Emulation Results

### 5.4.1 Finite Element Modeling and Validation

To validate hardware emulation of the real-time MEC based ST model, a 3-D three-phase, three-limb, twelve winding FE transformer model was developed in JMAG<sup>®</sup>. A half transformer core and coil model was built in JMAG<sup>®</sup> Designer<sup>®</sup>, which was then converted to a full transformer model by setting boundary symmetry. The symmetry boundary target is set in the middle of the model and its external coupled electrical circuit is shown in Fig. 5.6

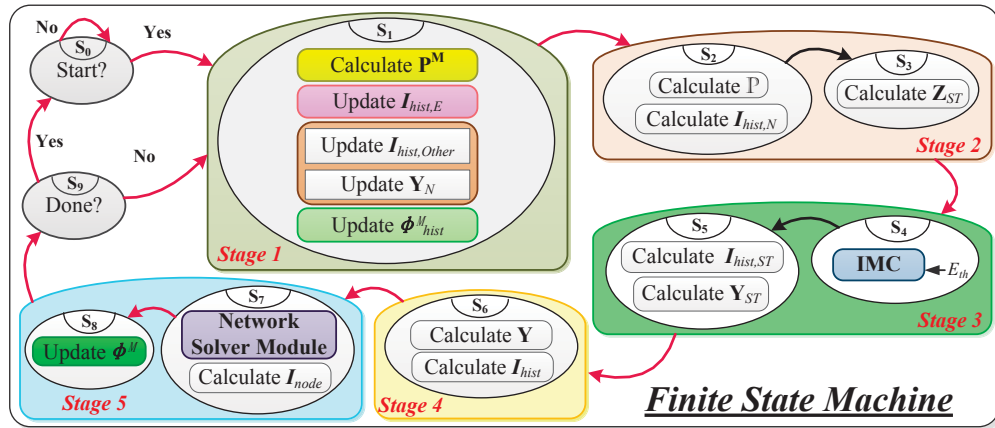


Figure 5.5: Finite state machine of the high-fidelity nonlinear MEC based ST EMT model and power system.

(a) and (b). In the coupled electrical circuit, each winding is assigned to a coil in the geometry model, and the current inflow face of each coil is as shown in Fig. 5.6 (b). This 3D-FE ST model was connected with a voltage source on the primary side and an open-circuit on the secondary side. To build the half transformer model, 47730 elements and 8647 nodes were used with the element size of 0.15 m. This 3D-FE ST model was connected with a voltage source on the primary side and a resistive load in the secondary side. By energizing the ST at  $t = 0.05$  s, an inrush current flows through the ST, and the magnetic flux density vector and contour are plotted in shown in Fig. 5.7 when the inrush current in phase- $c$  reaches its peak at  $t = 0.056$  s. The fluxes in phase- $a$  and phase- $b$  flow into phase- $c$  which make the primary side of phase- $c$  being the most saturated limb with maximum flux density close to 2.6 T. To run a  $t = 0.5$  s FE simulation with the time-step of  $\Delta t = 500 \mu s$  took more than 5 hours on a PC featured by Intel<sup>®</sup> i7 and 8 GB memory. The geometric parameters and the rating of the transformer can be found in the Appendix A.

## 5.4.2 Case Studies

The following case studies are analysed in this paper to show the accuracy and efficiency of the proposed the real-time hardware high-fidelity nonlinear MEC-based ST model. The power system shown in Fig. 5.8 consists of one ST model, two voltage sources,  $G_{s1}$  and  $G_{s2}$  and two transmission lines,  $L_1$  and  $L_2$ , modeled by the distributed parameter line model. In the case studies, the power system network including the ST is emulated on the Xilinx<sup>®</sup> Virtex-7 VC707 XC7VX485T FPGA, and the detailed latency for each FSM in one emulation time-step is summarised in Fig. 5.9. The maximum clock frequency of this design can reach to 90 MHz, with 3575 clocks latency for one time-step emulation giving the execution time of 40  $\mu s$ . The hardware resource utilization is summarised in Table I.

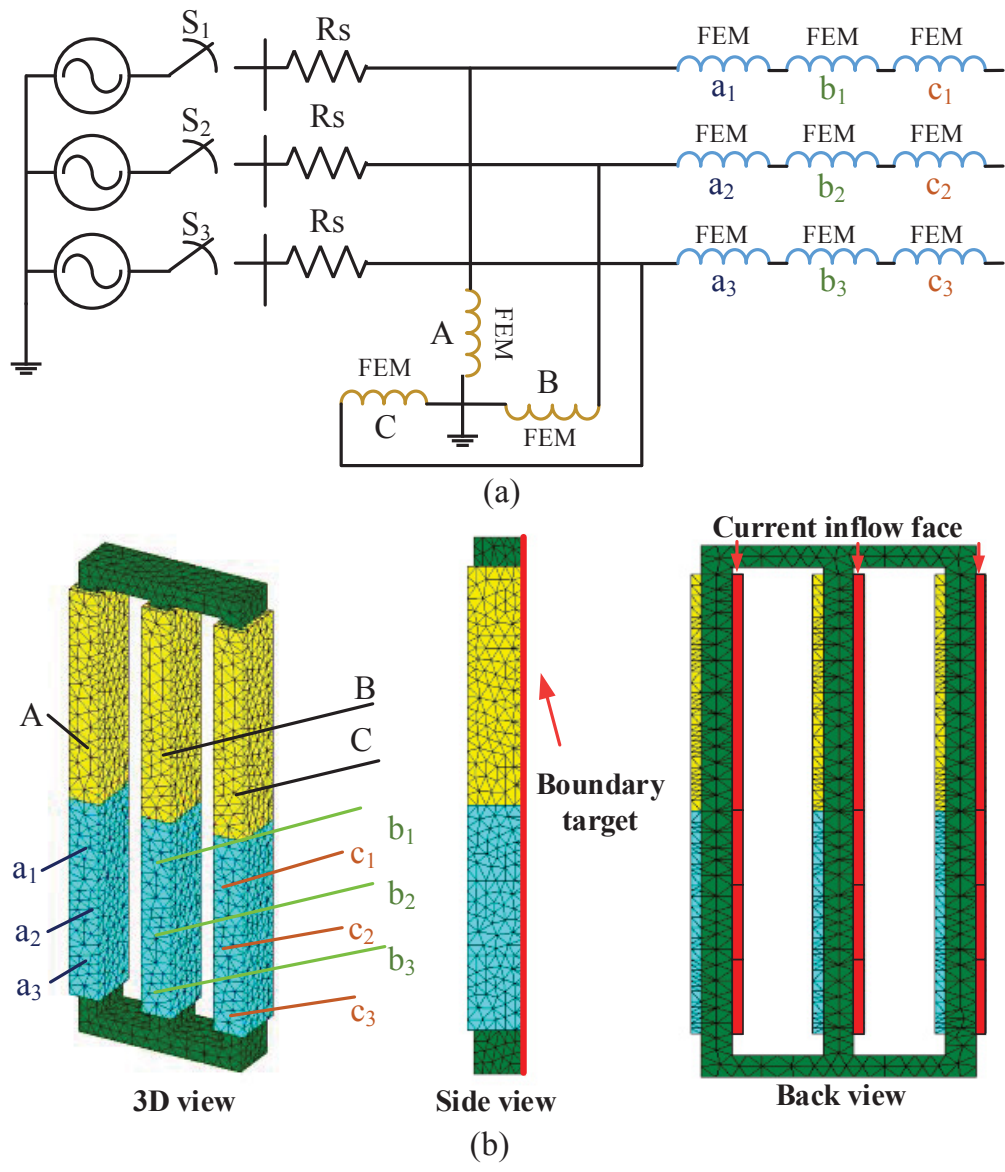


Figure 5.6: FEM simulation: (a) coupled external electrical circuit; (b,c) 3D FEM model.

**5.4.2.1 Energization transient and steady-state emulation**

To validate against the FE simulation results, the same power system is configured in Fig. 5.8 by the switching of  $SW_3$  to position 2 and leaving other breakers open. The parameters of the power system are given in Appendix B. The ST energization transient is initialized by closing the three-phase breaker  $SW_1$  at  $t = 0.05$  s, and the inrush transient current flows through the breaker. The real-time inrush current emulation results and the corresponding 3D-FE simulation results are plotted in Fig. 5.10. The magnitude of the inrush current in different phases is determined by the instantaneous magnitude of the applied voltages in different phases when the breaker  $SW_1$  is closed. As can be seen in Fig. 5.10, the magni-

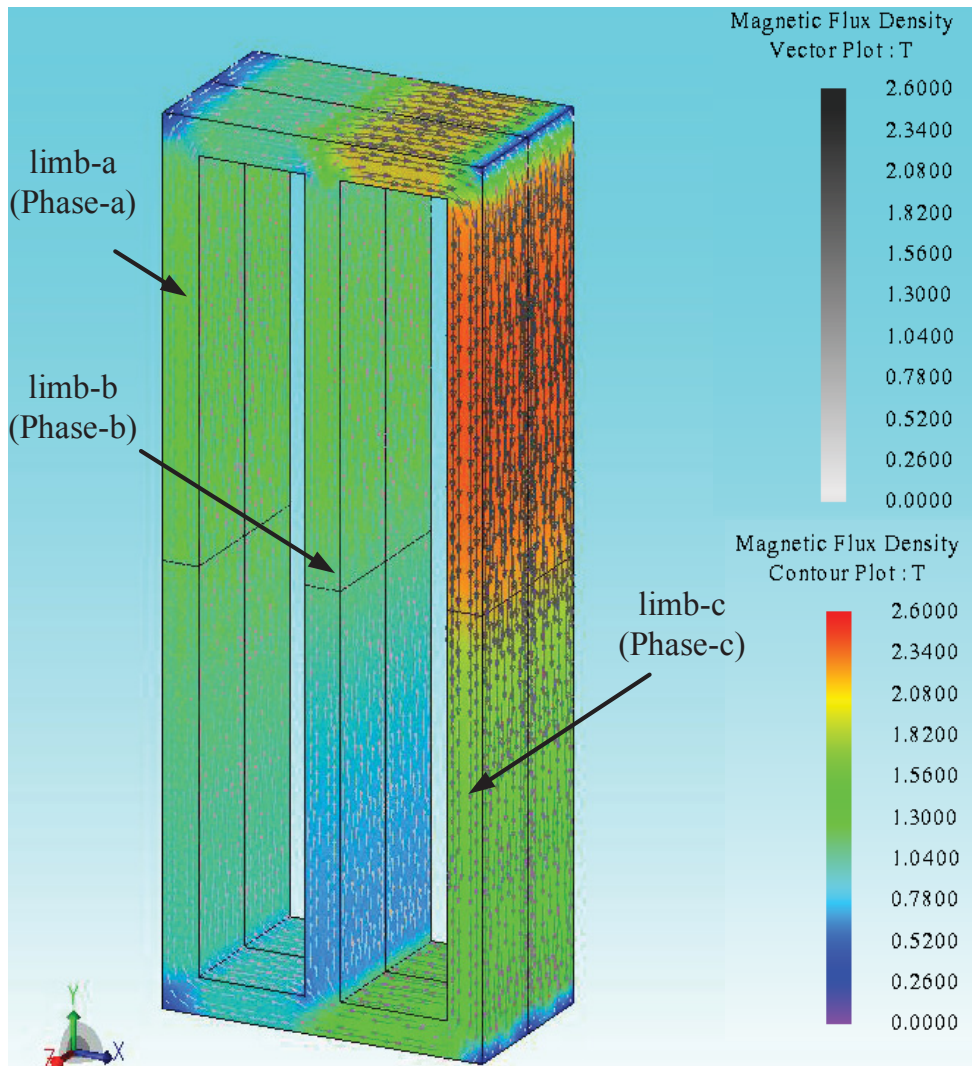


Figure 5.7: 3D-FE simulation of the ST in JMAG®.

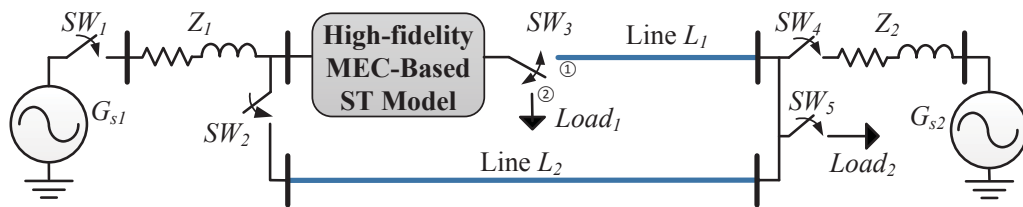


Figure 5.8: Single-line diagram of the power system with the ST model for the real-time HIL case study.

tude of transient current in phase-*b* and phase-*c* whose peak value can reach up to 150% of the ST rated current, is much larger than that in phase-*a* due to the above reason and all the currents in each phase decay to steady-state after  $t = 0.3$  s. The real-time emulation results show close agreement with the 3D-FE simulation results both in transient and steady

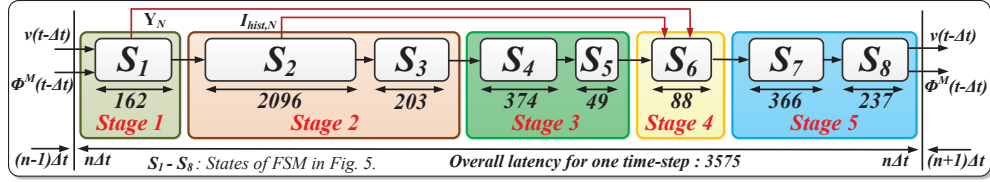


Figure 5.9: Latencies for one emulation time-step for the high-fidelity MEC based ST model.

Table 5.1: Module level hardware resource utilization

	MEC Based ST Model	Other Modules
<b>Slice Registers</b>	113746 (37.4%)	50562 (16.7%)
<b>LUTRAM</b>	3180 (24%)	9636 (7.35%)
<b>BRAM/FIFO</b>	90 (2.9%)	59 (9.6%)
<b>DSP48E1</b>	746 (26.6%)	221 (7.6%)
<b>BUFG</b>	11 (34.4%)	4 (12.4%)

state. However discrepancies exist in the magnitudes of the waveforms. The main reason for these differences is that even though the proposed real-time ST model was developed based on high-fidelity MEC approach, the mesh number and node are still much smaller than those in the FEM model. The Trapezoidal discretization can also introduce numerical differences between these two simulation results. Furthermore, in the FEM model, the nonlinear effect is modeled by a piece-wise nonlinear curve; however in the real-time MEC model the nonlinear effect in the ST core is represented by Preisach approach. In addition, even though the transformer core nonlinearity and frequency-dependent phenomena are included in the proposed model, this model cannot predict the high-frequency effects in the windings such as skin effect or proximity effect.

The real-time primary winding-limb flux-linkage simulation results are again captured and plotted on top of the FEM simulation results both under transient and steady-state in Fig. 4.21 (a)-(c) and (d), respectively. As expected the primary winding-limb three-phase flux waveform are in phase with the three-phase breaker current waveforms and at  $t = 0.056$  s the flux in limb-c (phase-c) reach its negative peak value and the flux in limb-a (phase-a) has the lowest peak value in the transient-state. Again close agreements can be observed between the real-time and FEM results.

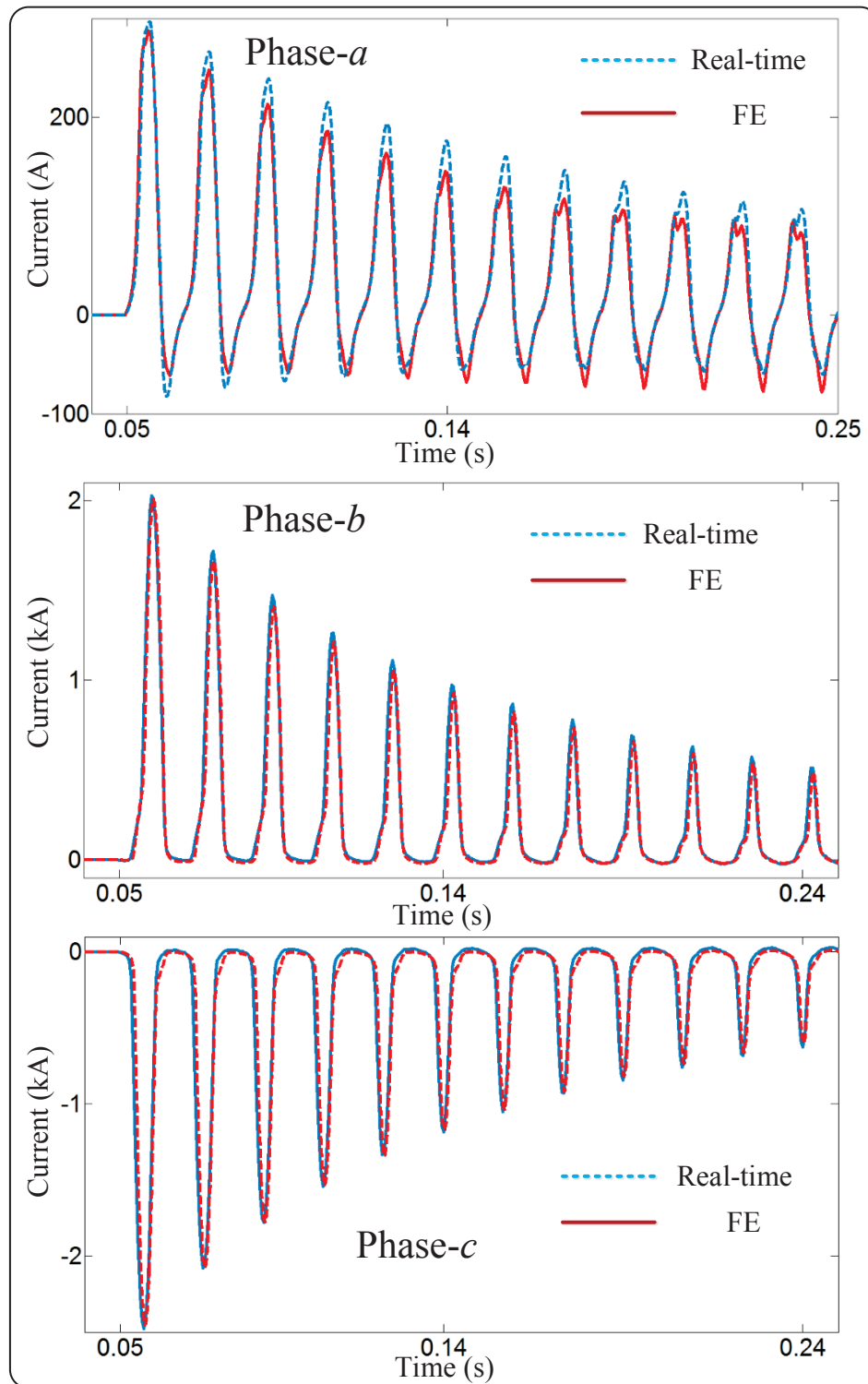


Figure 5.10: Real-time emulation and JMAG<sup>®</sup> simulation results for the energization transient.



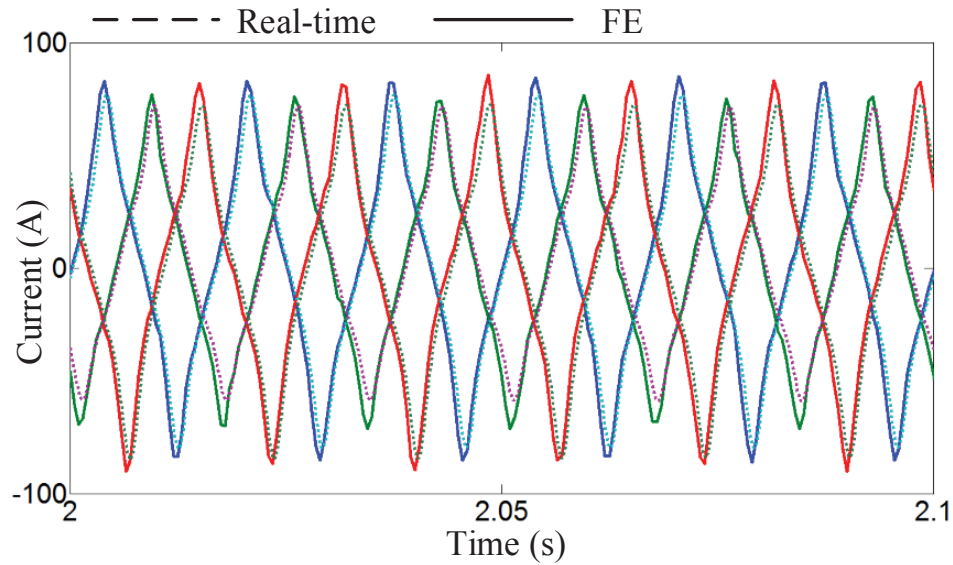


Figure 5.11: Real-time emulation and JMAG<sup>®</sup> simulation results for the ST steady-state current.

#### 5.4.2.2 Hysteresis transient emulation

The hysteresis transient is emulated initializing the transformer core flux on the downward major loop trajectory (point A). Fig. 11 shows the real-time emulation results of the flux and the magnetizing current in the core and the hysteresis characteristic of the transformer. The parameters to define the hysteresis major loop trajectory are listed in Appendix B. The ST is energized by a voltage source at a value of 0.8 p.u. starting from  $t = 0$  s and the corresponding hysteresis trajectory during the first cycle (A-B) is asymmetric due to the initial flux. After the first cycle, the flux travels on the set of asymmetric minor loops (C-D) due to the asymmetric magnetizing current through the transformer core. From  $t = 0.087$  s to  $t = 0.168$  s, the breaker  $SW_1$  is opened and the flux is locked at 0.35 p.u. After  $t = 0.168$  s the breaker  $SW_1$  is closed again and the value of voltage source increases to 1.2 p.u. The corresponding hysteresis trajectory (E-F) travels on the major loop since the voltage source is high enough to drive the transformer into the saturation region when the peak flux value reaches 1.2 p.u.

#### 5.4.2.3 Power flow regulation

To investigate the power flow control capability of the ST, the case study power system is configured by closing breaker  $SW_1$  and  $SW_4$  and switching breaker  $SW_3$  to position 1 to in order to obtain a constant voltage at the receiving end of line  $L_1$ . The real-time ST model works in compensation mode to make the voltage at the sending end of line  $L_1$  variable, so that the active power and reactive power flow through this line is controllable. The power-flow variation of the active power  $P_r$  and reactive power  $Q_r$  for  $V_{ST,c}$  of 0.1 p.u.,

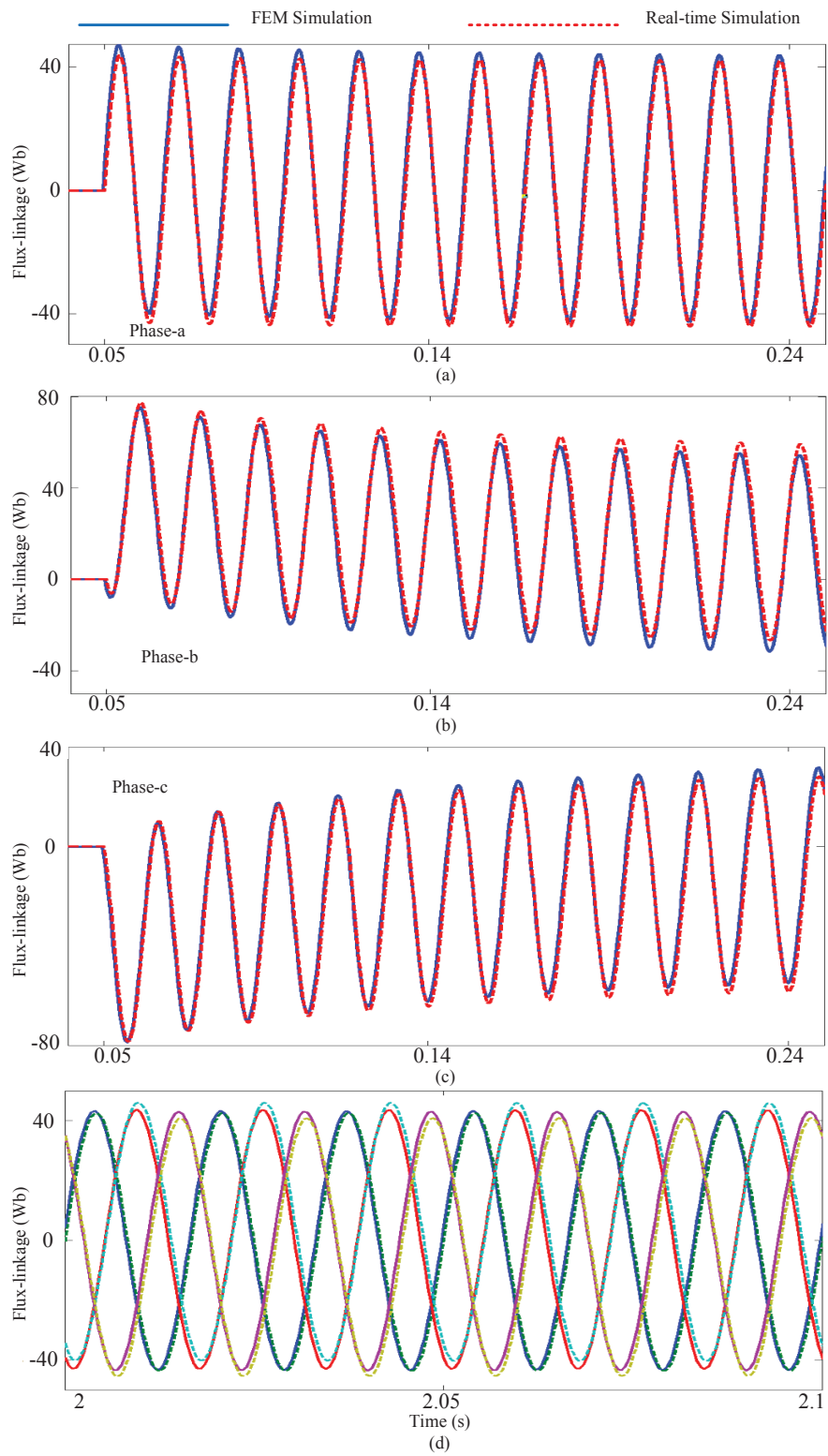


Figure 5.12: Real-time and JMAG<sup>®</sup> Designer simulation results: (a)-(c) flux-linkage under transient steady; (d) steady-state flux-linkage.



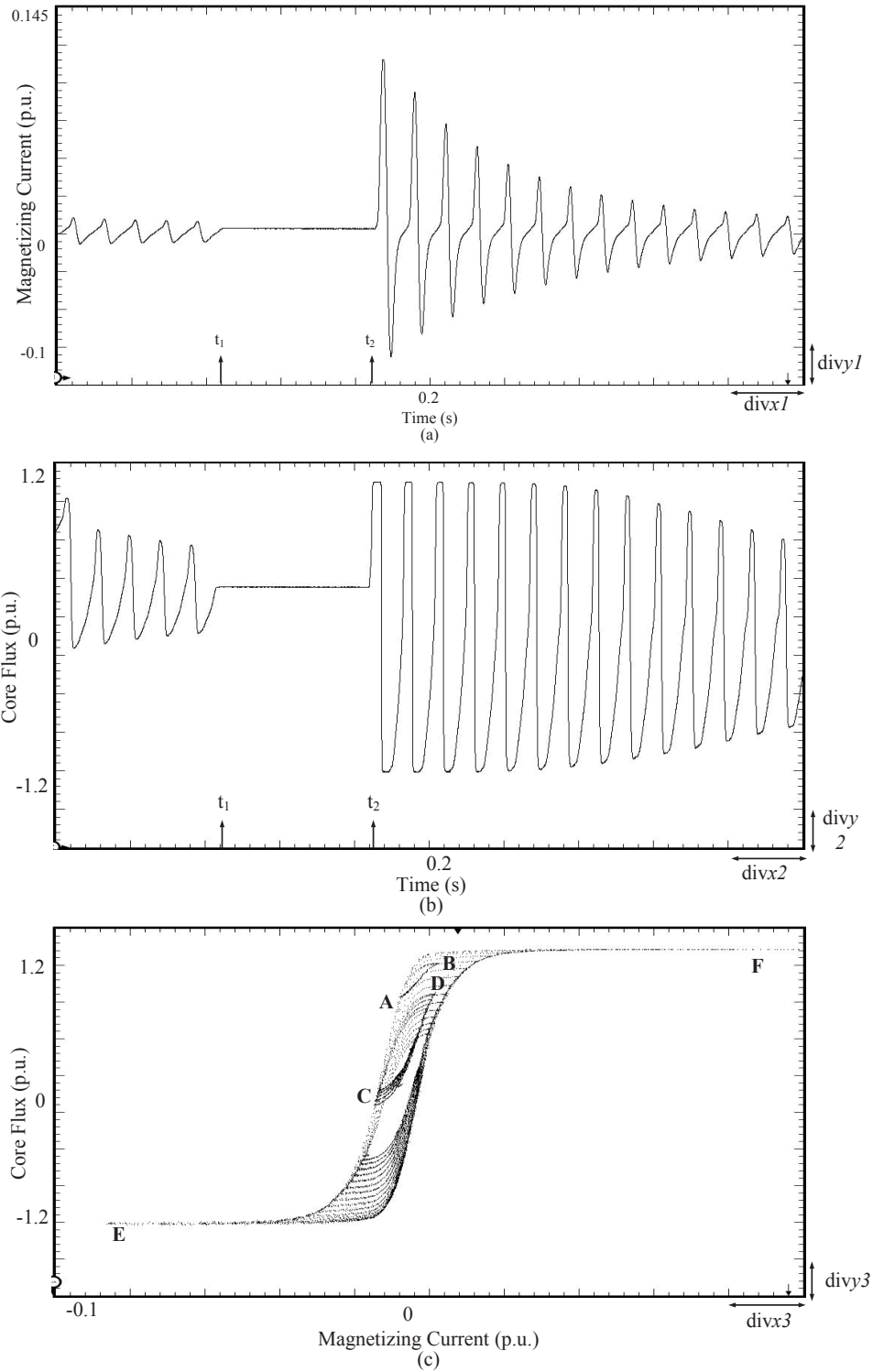


Figure 5.13: Real-time emulation results of (a) the magnetizing current (1 divx1 = 0.04 s; 1 divy1 = 0.025 p.u.); (b) flux in the core (1 divx2 = 0.04 s; 1 divy2 = 0.32 p.u.); (c) the hysteresis characteristic of ST (1 divx3 = 0.022 p.u.; 1 divy3 = 0.32 p.u.).

0.2 p.u., 0.3 p.u., and 0.4 p.u. at angle of  $0^\circ$ ,  $60^\circ$ ,  $120^\circ$ ,  $180^\circ$ ,  $240^\circ$ , and  $160^\circ$  is given in Fig. 5.14. As can be seen from this figure, the ST produce a slightly counter-clockwise hexagon profile, and for the angle of  $60^\circ$ ,  $240^\circ$ , the reactive power  $Q_r$  change much slower than the active  $P_r$  as the magnitude of injection voltage increasing.

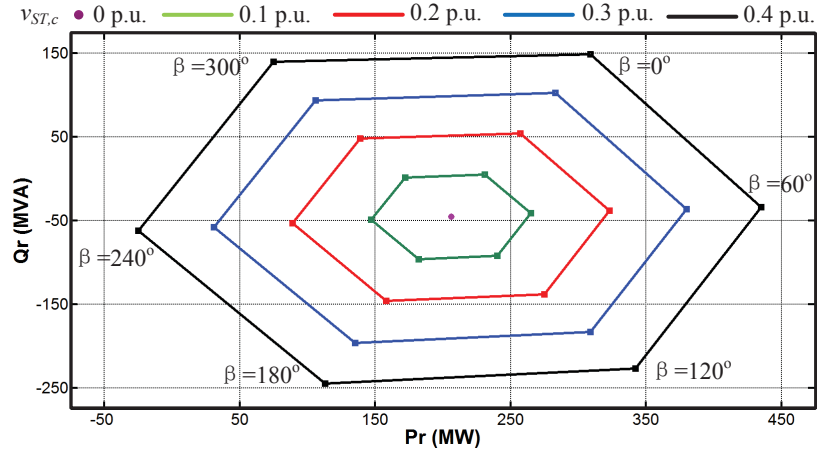


Figure 5.14: Relationship between  $P_r$  and  $Q_r$  at different injection voltage magnitude and phase angle for the ST.

#### 5.4.2.4 Power flow control emulation

To investigate of the ST transition response, during the step-by-step regulation active and reactive powers in transmission lines, the circuit shown in Fig. 5.8 was configured into a two transmission line system by opening breaker  $SW_4$  and switch  $SW_3$  to position 1 and leaving other breakers closed. The real-time emulation results of power regulation of the two lines, and the secondary winding current and injected compensation voltages of ST are shown in Fig. 5.15 and Fig. 5.16, respectively. At the beginning of the emulation, as shown in these figures, the ST works in the uncompensated mode; the line  $L_1$  active power  $P_{r1}$  and reactive power  $Q_{r1}$  is equal to 80 MW and -20 MVA, respectively and the line  $L_2$  active power  $P_{r2}$  and reactive power  $Q_{r2}$  is equal to 38 MW and 20 MVA, respectively, and the injected voltage is equal to zero. At  $t_1 = 0.5$  s a compensation voltage of 0.25 p.u. at angle of  $0^\circ$  is requested by the controller which means that the tap-setting on the secondary winding of  $GP_1$  should start to increase to 0.25 p.u. with step size of 0.05 p.u., which require 0.5 s for each step transition. Thus from  $t_1 = 0.5$  s to  $t_2 = 3$  s the tap-setting increased to 0.25 p.u. with the step size of 0.05 p.u. in 2.5 s. As can be seen in Fig. 5.15 during this period, the powers  $P_{r1}$  and  $Q_{r1}$  increase while the powers  $P_{r2}$  and  $Q_{r2}$  decrease, which proves that the ST has the capability to regulate the power flow in both transmission lines. Finally, the  $P_{r1}$  and  $Q_{r1}$  increase to 134 MW and 98 MVA, and  $P_{r2}$  and  $Q_{r2}$  decrease to 5 MW and -98 MVA respectively. At  $t_3 = 3.5$  s, the controller sends a request to ST to inject a compensation voltage of 0.4 p.u. at an angle of  $240^\circ$ , thus the ST

secondary winding tap-setting of  $GP_1$  needs to decrease to 0 p.u. again and  $GP_3$  needs to increase to 0.4 p.u. Starting from  $t_3 = 3.5$  s to  $t_5 = 6$  s, the  $GP_1$  windings' tap-setting decreases to 0 p.u. position, and  $GP_3$  increases to 0.25 p.u. position, respectively. From  $t_5 = 6$  s, the  $GP_3$  winding increases further to 0.4 p.u. As can be seen during this period the reactive power in both  $L_1$  and  $L_2$  is kept constant since the angle of the injection voltage is equal to  $240^\circ$ . The ST secondary winding current transition behavior during the entire period ( $t = 0$  s to 7.5 s) is shown in Fig. 5.16 (a). The current magnitude decreases during the power reversal period, starting from  $t_3 = 3.5$  s until  $t_4 = 5.5$  s since the absolute value of active power is decreased, however at the time  $t_4 = 5.5$  s the current magnitude starts to increase due to the fact that the absolute active power value is increased at this time, as shown in Fig. 5.15. The transition of the corresponding injected compensation voltage is shown in Fig. 5.16 (b); the initial value is zero since the ST works in uncompensated mode, and then the step-by-step changing of the voltage magnitude can be clearly seen at the request of the controller.

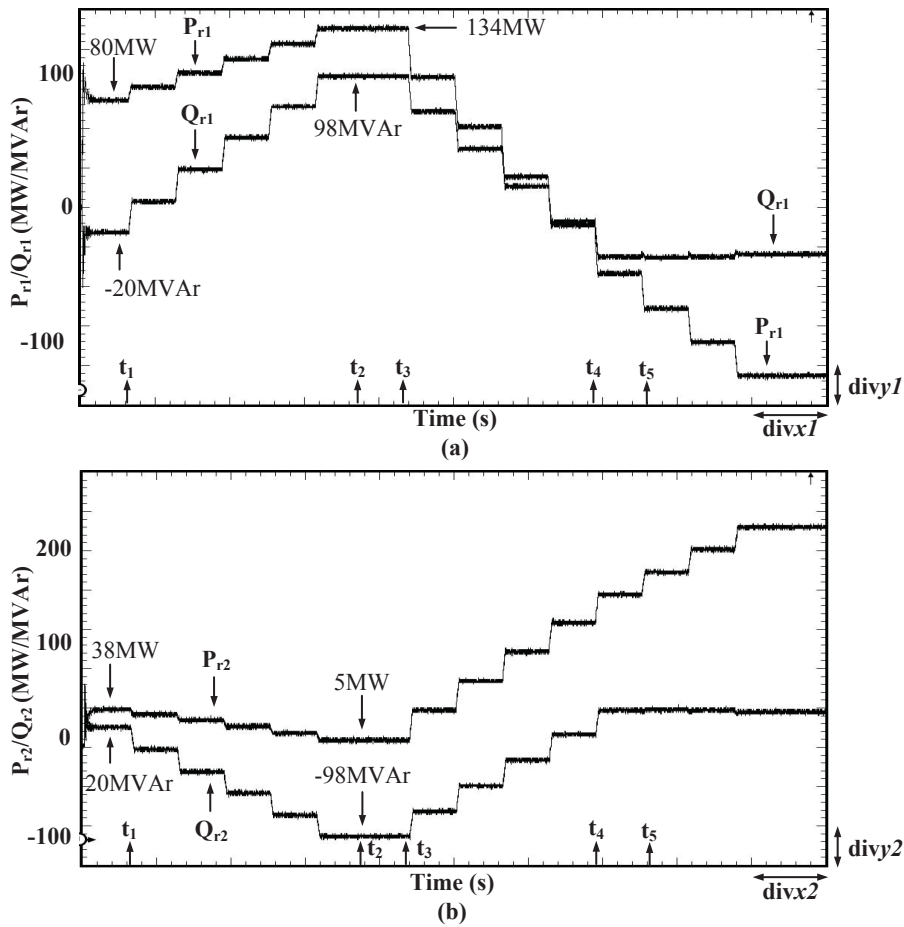


Figure 5.15: Real-time emulation of active power and reactive power regulation on the two transmission lines (a) Line  $L_1$  (1 divx1 = 0.8 s; 1 divy1 = 30 MW/MVAr); (b) Line  $L_2$  (1 divx2 = 0.8 s; 1 divy2 = 43 MW/MVAr).

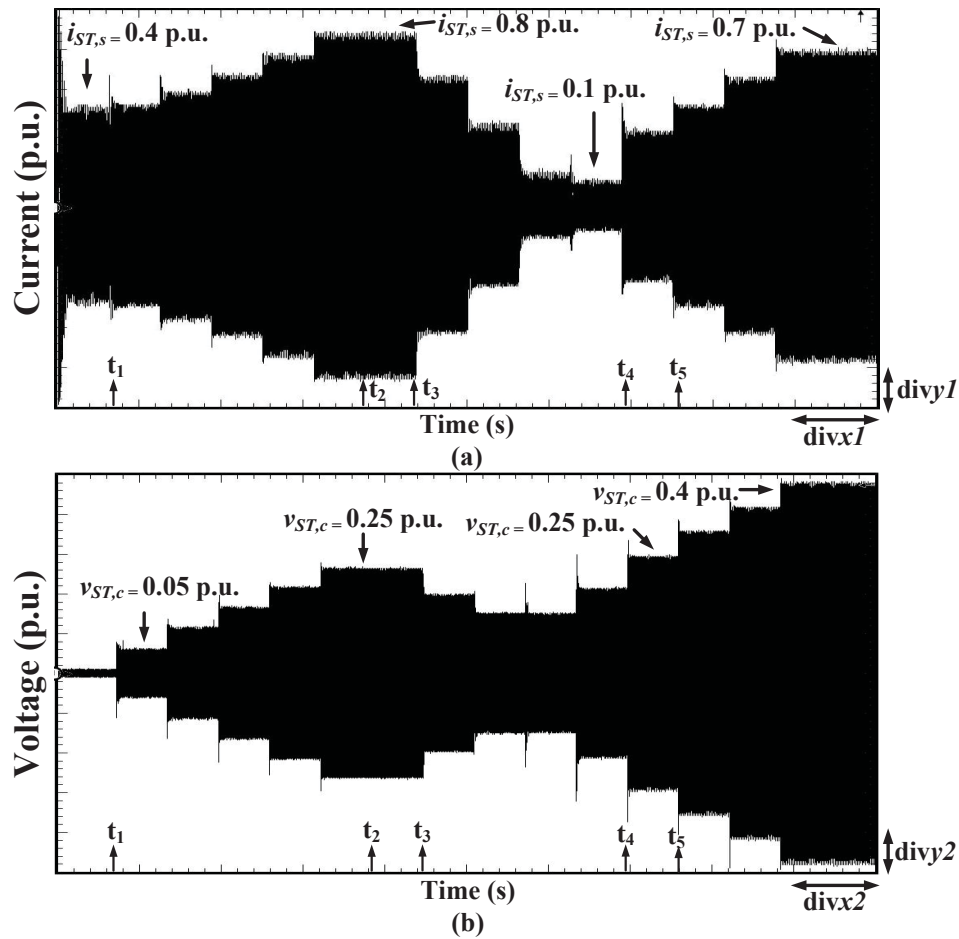


Figure 5.16: Real-time emulation of step-by-step transition response of (a) secondary winding current (1  $divx1 = 0.8$  s; 1  $divy1 = 0.18$  p.u.); (b) injected compensation voltage of ST (1  $divx2 = 0.8$  s; 1  $divy2 = 0.084$  p.u.).

#### 5.4.2.5 Fault transient emulation

A three-phase to ground fault was created at  $t_1 = 2.1$  s at the receiving end of the Line  $L_1$  and cleared at  $t_2 = 2.15$  s. Fig.5.17 shows the real-time emulation results of the ST secondary side voltage and current, as well as the injected compensation voltage. The fault current peak magnitude reached up to 3.8 p.u. compared to the steady-state magnitude of 0.7 p.u. and decays to steady-state after 2.4 s. The magnitude of ST secondary side fault transient voltage decreased to 0.64 p.u. from 1.16 p.u. in the steady-state, and the recovery voltage reach to 1.40 p.u. then decays to steady-state in two cycles. The recovery injection compensation voltage is up to 0.3 p.u. compared to 0.2 p.u. under steady-state. At the inception and clearing of the fault the high frequency transient can be clearly observed.

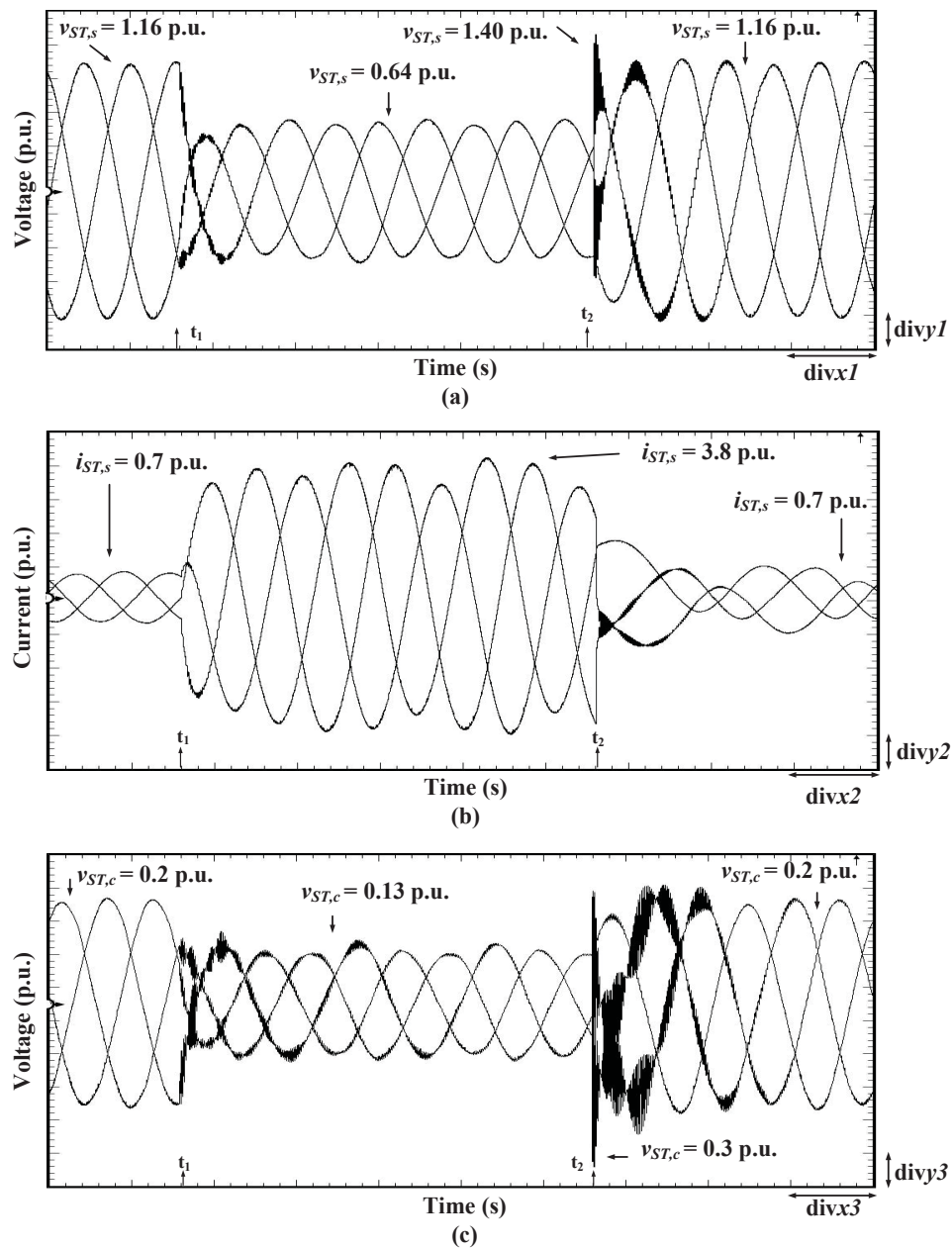


Figure 5.17: Real-time emulation results of ground fault transient (a) secondary side current (1  $divx1 = 0.01$  s; 1  $divy1 = 0.95$  p.u.); (b) secondary side voltage (1  $divx2 = 0.01$  s; 1  $divy2 = 0.32$  p.u.); (c) injected compensation voltage (1  $divx3 = 0.01$  s; 1  $divy3 = 0.067$  p.u.).

## 5.5 Summary

This chapter presents a smart power flow controller comprised of the Sen transformer which has the potential of providing a low-cost alternative for power flow regulation in stressed transmission networks. A real-time electromagnetic transient model of the ST based on a detailed magnetic equivalent circuit accounting for the major linear and non-linear flux paths in the transformer core are also presented in this chapter. Such a model can be used in a HIL scenarios to not only implement and test newer power flow control algorithms, but can also be used for the transformer design optimization, and for designing protection schemes. The exploitation of full parallelism and deep pipelining on the FPGA has resulted in an ST model that is not only highly accurate but also has a low computational latency and resource consumption to be included in HIL simulations. The proposed real-time model is able to accurately capture the nonlinear behavior due to core saturation, hysteresis and eddy currents. Comparison with finite element simulations prove satisfactory performance of the real-time model both under steady-state and transient conditions.

# 6

## Conclusions and Future Works

The power transformer is a widely used power system component to deliver and distribute electrical power energy. As the importance of HIL simulation is growing in the industry in recent years which allows an apparatus to be tested in a virtual environment under various operation conditions before being commissioned in physical world, a detailed power transformer real-time model has become necessary for electromagnetic transient studies.

Due to the rapidly increasing density and speed, FPGAs have been gaining great interest for high performance computation applications. The FPGAs by its nature can execute an application in parallel; however, the novel design of the model and its solution algorithm in a parallel and pipelined fashion is dependent on the user to extract the highest accuracy and efficiency. As the FPGAs continues to develop the power consumption, cost, and the time spent on VHDL programming will decrease further.

In this thesis, detailed FPGA-based real-time transformer hardware emulations are developed and various case studies are investigated to show the accuracy and effectiveness of the proposed models. The summary of the completed thesis work and suggested future work are given in this chapter.

### 6.1 Contributions of This Thesis

The main contributions of this thesis can be summarized as follows:

1. A admittance-based transformer model is proposed for FPGA-based real-time hardware emulation.
  - (a) The mathematical of transformer model is derived from first principles.

- (b) Other power system component real-time models are developed on FPGA, such as Bergeron transmission line model. Spares matrix vector multiplication scheme are developed and implemented on FPGA in order to achieve lower computation latency and hardware resource utilization. The overall hardware architecture design are depicted and explained and its corresponding hardware resource utilization and computational latency are also provided.
  - (c) Sequentially EMP algorithm is reformulated to extract possible parallel computation and accommodate them into FPGA parallel architecture. Moreover, the pipeline processing are realized in each computation model by inserting memory to be a part of the data path.
  - (d) To increase the accuracy of the detailed transformer model, nonlinear and frequency dependent models are developed based on Preisach model and a cascade of RL lumped sections, respectively.
  - (e) To solve the nonlinear equations, the compensation method is employed and a paralleled hardware Newton-Raphson implementation is developed.
2. A topology based transformer model, HR-MEC model, is proposed for FPGA-based transformer real-time emulation.
- (a) The mathematical of HR-MEC based model is derived from first principles.
  - (b) The involved parallel computation schemes such as spares matrix computing unit and accumulator unit are developed.
  - (c) As in HR-MEC hardware emulation, the Norton admittance matrix representing the transformer changes in each time-step, so a LU-based linear solver is developed to solve the nodal analysis method. Its implementation is fully pipelined and paralleled to achieve real-time simulation.
  - (d) A LU based matrix inverter are also developed in order to compute certain equations in the HR-MEC transformer model.
  - (e) A 3-D FEM transformer model is developed in JMAG<sup>®</sup>, and used to verify the proposed HR-MEC hardware emulation.
3. A real-time high-fidelity MEC based EMT model for the power-flow control apparatus ST is presented.
- (a) The mathematical of HR-MEC based model is derived from first principles.
  - (b) An advanced tap-changing algorithm is developed to regulate both active power and reactive power flowing through the transmission lines.
  - (c) A 3-D twelve winding FEM ST model is developed in JMAG<sup>®</sup> software to compare with the proposed model under both steady-state and transient conditions.



4. An IEEE 754 single-precision floating point number representation is used for high accuracy at most part of the hardware emulation. The only exception is in accumulator unit, the accumulation is done by fix number representation to ensure the real-time simulation. All hardware arithmetic units are deeply pipelined to achieve high computation throughput.

## 6.2 Directions for Future Work

The following topics are proposed for future work:

1. A 3-D HR-MEC model of transformer in real-time on FPGA could be developed in future. Such a model can depict the flux lines within the transformer in 3 directions. Again, the permanence can be used to represent the flux path in three directions. With this model, the transformer performance and its behavior can be duplicated with higher precision.
2. High-frequency surge real-time modeling of transformer using capacitance need more research in the future work. There are three sources of capacitance: inter-winding capacitors, winding-to-winding capacitors and winding-to-ground capacitors. The inter-winding capacitors can be represented by series capacitance. These capacitances form a network of several different capacitance values. The winding-to-winding capacitors and winding-to-ground capacitors can be represented by shunt capacitances, and their values can be calculated by geometrical formulas. To develop the high-frequency model, those capacitors should also be included to generate the admittance matrix. At high frequencies, the flux does not penetrate through the core, so the nonlinear effects such as the saturation in the iron core can be neglected.
3. More efforts are needed to model the transformer tap-changer. The transformer tap changers can be generally classified into three types: mechanical, electronically assisted and fully electronic. In the mechanical tap changers, taps can only change from one position to the previous or next position, which is a step-by-step process; however, in electronic tap-changer, there is no such limitation, the taps can jump from the highest to the lowest position instantaneously. However, the disadvantage in electronic tap changers is that the non-conducting device still dissipates power, adding up to a few kilowatts which appears as heat and results in a low transformer efficiency. Additionally, as the taps change very quickly, it is necessary to detect the voltage changes in the bus quickly.
4. This thesis is focused on the transformer core modeling, more effort would be done to the transformer winding modeling, such as skin effect, proximity effect and winding inner fault.

5. As the real-time simulator can be employed for HIL simulation, it would be useful if more research can be done regarding the transformer design procedure by using the proposed FPGA-based real-time transformer models.
6. The Solid-State Transformer (SST), as known as power electronic transformer, is a recently emerging technology in power transmission and distribution system. The basic idea behind the SST is that the AC voltage is transformed into a high frequency voltage, and then step up/down by a high-frequency transformer, and at last transformed back into the desired frequency AC voltage. Compared to traditional transformer, as the solid-state semiconductor devices exist in SST, so that the voltage and current regulation can be done and potentially reduce the volume, weight and cost of the transformer. However, due to the limitation of voltage and power rating of devices and circuit topologies, more research is need before the SST can be widely used in the power system. In the HIL side, since the electronic devices have already been implemented on FPGA, it is a potential possibility to implement a SST model on FPGA.

## Bibliography

- [1] J. Avila-Rosales and F. L. Alvarado, "Nonlinear frequency dependent transformer model for electromagnetic transient studies in power systems", *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 11, pp. 4281-4288, Nov. 1982.
- [2] A. Narang and R. H. Brierley, "Topology based magnetic model for steady-state and transient studies for three-phase core type transformers", *IEEE Trans. on Power Systems*, vol. 9, no. 3, pp. 1337-1349, Aug. 1994.
- [3] Y. Shibuya, S. Fujita and T. Shimomura, "Effects of very fast transient overvoltages on transformer", *Proc. Inst. Elect. Eng., Gen., Transm. Distrib.*, vol. 146, no. 5, pp. 459-464, Sep. 1999.
- [4] V. Dinavahi, M. R. Iravani and R. Bonert, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers", *IEEE Trans. Power Del.*, vol. 16, no. 4, pp. 775-781, Oct. 2001.
- [5] V. Dinavahi, R. Iravani and R. Bonert, "Design of a real-time digital simulator for a D-STATCOM system", *IEEE Trans. Ind. Electron.*, vol. 51, no. 5, pp. 1001-1008, Oct. 2004.
- [6] L.-F. Pak, M. O. Faruque, X. Nie and V. Dinavahi, "A versatile cluster-based real-time digital simulator for power engineering research", *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 455-465, May 2006.
- [7] X. Nie, Y. Chen and V. Dinavahi, "Real-time transient simulation based on a robust two-layer network equivalent", *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1771-1781, Nov. 2007.
- [8] L.-F. Pak and V. Dinavahi, "Real-time simulation of a wind energy system based on the doubly-fed induction generator", *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1301-1309, Aug. 2009.
- [9] V. Jalili-Marandi and V. Dinavahi, "Instantaneous relaxation-based real-time transient stability simulation", *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1327-1336, Aug. 2009.
- [10] V. Jalili-Marandi, L.-F. Pak and V. Dinavahi, "Real-time simulation of grid-connected wind farms using physical aggregation", *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3010-3021, Sept. 2010.

- [11] M. O. Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. Strunz, M. Saeedifard, X. Wang, D. Shearer and M. Paolone, "Real-time simulation technologies for power systems design, testing, and analysis", *IEEE Power Energy Technol. Syst. J.*, vol. 2, no. 2, pp. 63-73, June 2015.
- [12] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP", *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892-902, Apr. 2009.
- [13] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA", *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2547-2555, June 2011.
- [14] B. Asghari, and V. Dinavahi, "Real-time nonlinear transient simulation based on optimized transmission line modeling", *IEEE Trans. Power Syst.*, vol. 26, no. 2, pp. 699-709, May 2011.
- [15] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation", *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1300-1309, Feb. 2012.
- [16] B. Asghari, and V. Dinavahi, "Experimental validation of a geometrical nonlinear permeance network based real-time induction machine model", *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4049-4062, Nov. 2012.
- [17] N. R. Tavana, and V. Dinavahi, "A general framework for FPGA-based real-time emulation of electrical machines for HIL applications", *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2041-2053, Apr. 2015.
- [18] N. R. Tavana, and V. Dinavahi, "Real-time FPGA-based analytical space harmonic model of permanent magnet machines for hardware-in-the-loop simulation", *IEEE Trans. Magnetics*, vol. 51, no. 8, pp. 1-9, Aug. 2015.
- [19] N. R. Tavana, and V. Dinavahi, "Real-time nonlinear magnetic equivalent circuit model of induction machine on FPGA for hardware-in-the-loop simulation", *IEEE Trans. Energy Conversion*, vol. PP, no. 99, pp. 1-11, 2016.
- [20] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives", *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235-1246, Apr. 2007.
- [21] M. O. O. Faruque, and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization", *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1146-1158, Apr. 2010.
- [22] A. Myaing, and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics", *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 358-368, Jan. 2011.

- [23] W. Wang, Z. Shen and V. Dinavahi, "Physics-based device-level power electronic circuit hardware emulation on FPGA", *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2166-2179, Nov. 2014.
- [24] Z. Shen and V. Dinavahi, "Real-time device-level transient electro-thermal model of the modular multilevel converter on FPGA", *IEEE Trans. Power Electronics*, vol. PP, no. 99, pp. 1-1.
- [25] Y. Wang and V. Dinavahi, "Low-latency distance protective relay on FPGA", *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 896-905, March 2014.
- [26] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems", *IET Gen. Transmiss. Distrib.*, vol. 7, no. 5, pp. 451-463, May 2013.
- [27] Y. Chen and V. Dinavahi, "Hardware emulation building blocks for real-time simulation of large-scale power grids", *IEEE Trans. Ind. Inform.*, vol. 10, no. 1, pp. 373-381, Feb. 2014.
- [28] N. R. Tavana, "Real-time emulation of electrical machines for hardware-in-the-loop applications", Doctoral thesis, University of Alberta, 2015, 124 pages.
- [29] J. A. Martinez and B. A. Mork, "Transformer modeling for low- and mid-frequency transients - a review", *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235-1246, April 2007.
- [30] V. Brandwajn, H. W. Dommel and I. Dommel, "Matrix representation of three-phase n-winding transformers for steady-state and transient studies", *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 6, pp. 1369-1378, June 1982.
- [31] X. Chen, "Negative inductance and numerical instability of the saturable transformer component in EMTP", *IEEE Trans. on Power Del.*, vol. 15, no. 4, pp. 1199-1204, Oct. 2000.
- [32] "Alternative transients program rule book", Heverlee, Belgium, K. U. Leuven EMTP Center, 1987
- [33] E. C. Cherry, "The duality between interlinked electric and magnetic circuits and the formation of transformer equivalent circuits", *Proc. Physical Society*, vol. 62, 1949, pp. 101-111.
- [34] G. R. Slemon "Equivalent circuits for transformers and machines including non-linear effects", *Proc. Inst. Elect. Eng. IV*, vol. 100, 1953, pp. 129-143.
- [35] B. Mork, F. Gonzalez, D. Ishchenko, D. Stuehm and J. Mitra, "Hybrid transformer model for transient simulation: part I: development and parameters", *IEEE Trans. on Power Delivery*, vol. 22, no. 1, pp. 248-255, Jan. 2007.

- [36] H. W. Dommel, *EMTP theory book*, BPA, Portland, OR, 1985.
- [37] X. Chen, "A three-phase multi-legged transformer model in ATP using the directly-formed inverse inductance matrix", *IEEE Trans. on Power Del.*, vol. 11, no. 3, pp. 1554-1562, Jul. 1996.
- [38] A. Rezaei-Zare, R. Iravani, M. Sanaye-Pasand, and S. Farhangi, "An accurate hysteresis model for ferroresonance analysis of a transformer", *IEEE Trans. on Power Del.*, vol. 23, no. 3, pp. 1448-1456, Jul. 2008.
- [39] Y. Zhang, T. Maguire and P. Forsyth, "UMEC transformer model for the real time digital simulator", *Int. Conf. on Power Syst. Trans. (IPST)*, pp. 1-5, June 2005.
- [40] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems—a review", *IEEE Trans. Ind. Electron.*, vol.54, no.4, pp. 1824-1842, Aug. 2007.
- [41] J. J. Rodriguez-Andina, M. J. Moure, M. D. Valdes, "Features, design tools, and application domains of FPGAs," *IEEE Trans. on Ind. Electron.*, vol. 54, no. 10, pp. 1810-1823, Aug. 2007.
- [42] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan and M. W. Naouar, "FPGAs in industrial control applications", *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224-243, May 2011.
- [43] T. J. Vyncke, S. Thielemans and J. A. Melkebeek, "Finite-set model-based predictive control for flying-capacitor converters: cost function design and efficient FPGA implementation", *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 1113-1121, May 2013.
- [44] H. Guo, H. Chen, F. Xu, F. Wang, G. Lu, "Implementation of EKF for vehicle velocities estimation on FPGA", *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3823-3835, Sept. 2013.
- [45] Shyu Kuo-Kai, Chiu Yun-Jen, Lee Po-Lei, Lee Ming-Huan, Sie Jyun-Jie, Wu Chi-Hsun, Wu Yu-Te, Tung Pi-Cheng, "Total design of an FPGA-based brain-computer interface control hospital bed nursing system", *IEEE Trans. Ind. Electron.*, vol. 60, no. 7, pp. 2731-2739, July 2013.
- [46] M. Curkovic, K. Jezernik, R. Horvat, "FPGA-based predictive sliding mode controller of a three-phase inverter", *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 637-644, Feb. 2013.
- [47] J. Abrahamsson, M. Hedlund, T. Kamf, H. Hans Bernhoff, "High-speed kinetic energy buffer for electric vehicles in urban traffic: design and optimization", *IEEE Trans. Ind. Electron.*, vol. PP, no. 99, 2013.

- [48] Tsai Ching-Chih, Huang Hsu-Chih, Lin. Shui-Chun, "FPGA-based parallel DNA algorithm for optimal configurations of an omnidirectional mobile service robot performing fire extinguishment", *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 1016-1026, March. 2011.
- [49] A. Dinu, M. N. Cirstea, S. E. Cirstea, "Direct neural-network hardware-implementation algorithm", *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1845-1848, May 2010.
- [50] "7 series FPGA configurable logic block", Xilinx User Guide, November 2014
- [51] "7 series FPGA memory resources", Xilinx User Guide, November 2014
- [52] "7 series FPGA DSP48E1 slice", Xilinx User Guide, November 2014
- [53] I. D. Mayergoyz, "Mathematical models of hysteresis", Berlin, Germany: Springer-Verlag, 1991.
- [54] S. R. Naidu, "Simulation of the hysteresis phenomenon using Preisach's theory", *Proc. Inst. Elect. Eng. A*, vol. 137, no. 2, pp. 73-79, Mar. 1990.
- [55] J. Avila-Rosales, A. Semlyen, "Iron core modeling for electrical transients", *IEEE Trans. on Power App. Syst.*, vol. PAS-104, no. 11, pp. 3189-3194, Nov. 1985.
- [56] M.-W. Naouar, E. Monmasson, A. A. Nassani, I. Slama-Belkhodja, N. Patin, "FPGA-based current controllers for AC machine drives- a review", *IEEE Trans. Ind. Electron.*, vol.54, no. 4, pp. 1907-1925, Aug. 2007.
- [57] Z. Hajduk, B. Trybus, J. Sadolewski, "Architecture of FPGA embedded multiprocessor programmable controller", *IEEE Trans. on Ind. Electron.*, vol. PP, no. 99, 2014.
- [58] R. Horvat, K. Jezernik, M. Curkovic, "An event-driven approach to the current control of a BLDC motor using FPGA", *IEEE Trans. on Ind. Electron.*, vol. 61, no. 7, pp. 3719-3726, July 2014.
- [59] W. Enright, O. B. Nayak, G. D. Irwin, and J. Arrillaga, "An electromagnetic transients model of multi-limb transformers using normalized core concept", in *Proc. Int. Conf. Power System Transients*, Seattle, WA, Jun. 1997, pp. 93-98.
- [60] W. Enright, N. Watson and O. Nayak, "Three-phase five-limb unified magnetic equivalent circuit transformer models for PSCAD V3", *Int. Conf. on Power Syst. Trans. (IPST)*, pp. 462-467, June 1999.
- [61] T. Nechma, M. Zwolinski and J. Reeve, "Parallel sparse matrix solver for direct circuit simulation on FPGA", *Proc. Circuits and Systems*, May 30 - June 2, 2010.



- [62] T. Hauser, A. Dasu, A. Sudarsanam and S. Young, "Performance of a LU decomposition on a multi-FPGA system compared to a low power commodity microprocessor system", *Scalable Computing: Parcice and Experience*, vol. 8, no. 4, pp. 373-385, 2007.
- [63] M. K. Jaiswal and N. Chandrachoodan, "FPGA-based high-performance and scalable block LU decomposition architecture", *IEEE Trans. on Comp.*, vol. 61, no. 1, pp. 60-72, Jan. 2012.
- [64] W. Guiming, D. Yong, S. Junqing and G. D. Peterson, "A high performance and memory efficient LU decomposer on FPGAs", *IEEE Trans. on Comp.*, vol. 61, no. 3, pp. 366-378, Mar. 2012.
- [65] J. W. Demmel and N. J. Higham, "Stability of block algorithms with fast level-3 BLAS", *ACM Trans. on Math. Software*, vol. 18, no. 3, pp. 274-291, Sept. 1992.
- [66] D. L. Stuehm, "Three-phase transformer core modeling", *Final Report to Bonneville Power Administration Award*, No. DE-BI79-92BP26700, Feb. 28.
- [67] E. J. Tarasiewicz, A. S. Morched, A. Narang, E. P. Dick, "Frequency dependent eddy current models for nonlinear iron cores ", *IEEE Trans. on Power Syst.*, vol. 8, no. 2, pp. 588-597, May. 1993.
- [68] R. Podmore, M. R. Robinson, "The role of simulators for smart grid development", *IEEE Trans. on Smart Grid*, vol. 1, no. 2, pp. 193-198, Setp. 2010.
- [69] S. Bifaretti, P. Zanchetta, A. Watson, L. Tarisciotti and J. C. Clare, "Advanced power electronic conversion and control system for universal and flexible power management", *IEEE Trans. on Smart Grid*, vol. 2, no. 2, pp. 231-243, Jun. 2011.
- [70] Q. Guo, H. Sun, M. Zhang, J. Tong, B. Zhang and B. Wang, "Optimal voltage control of PJM smart transmission grid: study, implementation, and evaluation", *IEEE Trans. on Smart Grid*, vol. 4, no. 3, pp. 1665-1674, Sep. 2013.
- [71] D. Van Hertem, J. Rime and R. Belmans, "Power flow controlling devices as a smart and independent grid investment for flexible grid operations: belgian case study", *IEEE Trans. on Smart Grid*, vol. 4, no. 3, pp. 1656-1664, Apr. 2013.
- [72] P. Guha Thakurta, J. Maeght, R. Belmans and D. Van Hertem, "Increasing transmission grid flexibility by TSO coordination to integrate more wind energy sources while maintaining system security", *IEEE Trans. on Sustainable Energy*, vol. PP, no. 99, pp. 1-9, Aug. 2014.
- [73] B. A. Renz, A. Keri, A. S. Mehraban, C. Schauder, E. Stacey, L. Kovalsky, L. Gyugyi, and A. Edris, "AEP unified power flow controller performance", *IEEE Trans. on Power Del.*, vol. 14, no. 4, pp. 1374-1381, Oct. 1999.



- [74] K. K. Sen and E. J. Stacey, "UPFC-unified power flow controller: theory, modeling and applications", *IEEE Trans. on Power Del.*, vol. 13, no. 4, pp. 1453-1460, Oct. 1998.
- [75] K. K. Sen and M. L. Sen, "Introducing the family of "sen" transformer: a set of power flow controlling transformers", *IEEE Trans. on Power Del.*, vol. 18, no. 1, pp. 149-157, Jan. 2003.
- [76] K. K. Sen and M. L. Sen, "Comparison of the "sen" transformer with the unified power flow controller", *IEEE Trans. on Power Del.*, vol. 18, no. 4, pp. 1523-1533, Oct. 2003.
- [77] K. K. Sen and M. L. Sen, "Introducing the SMART power flow controller - an integral part of smart grid", *Elect. Power and Energy Conf. (EPEC), 2012 IEEE*, pp. 98-104, 10-12 Oct. 2012.
- [78] B. Asghari, M. O. Faruque and V. Dinavahi, "Detailed real-time transient model of the "sen" transformer", *IEEE Trans. on Power Del.*, vol. 23, no. 3, pp. 1513-1521, Jul. 2008.
- [79] M. O. Faruque and V. Dinavahi, "A tap-changing algorithm for the implementation of "sen" transformer", *IEEE Trans. on Power Del.*, vol. 22, no. 3, pp. 1750-1757, Jul. 2007.
- [80] W. H. Hayt, J. A. Buck, "Engineering electromagnetics", McGraw Hill, 2001



## Power System Data of Case Study in Chapter 3

### A.1 Case Study I

Table A.1: Parameters of the case study I, inrush current transient

Parameter	Values and configuration
Transformer, T	250 MVA, 200 kV / 400 kV, Y-Y connection
Voltage source, G	20 kV
Resistance, R	1 $\Omega$

Table A.2: Parameters of the case study I, ferroresonance and eddy current transients

Parameter	Values and configuration
Transformer, T	250 MVA, 125 kV / 500 kV, Y-Y connection
Voltage source, G	50 kV
Capacitance, $C_s$	0.05 $\mu F$
Capacitance, $C_s$	0.0125 $\mu F$
Resistance, R	500 $\Omega$ .

## A.2 Case Study II

Table A.3: Parameters of the case study II

Parameter	Values and configuration
Transformer, T1	20 MVA, 15 kV/35 kV, Y-Y connection
Transformer, T2	20 MVA, 32 kV/14 kV, Y-Y connection
Transformer, T3	20 MVA, 32 kV/20 kV, Y-Y connection
Transmission line inductance	mode +: 1.556 mH/m, mode 0: 5.819 mH/m
Transmission line capacitance	mode +: 0.0194 $\mu$ F/m, mode 0: 0.0120 $\mu$ F/m
Transmission line length	L1, L2, L3: 120 m, 140 m, 180 m
Voltage source, G	50 kV

## A.3 Parameters for Hysteresis

Table A.4: Parameters for hysteresis major loop trajectory

Parameters	Values
$a_1$	0.7900
$a_2$	0.2430
$a_3$	200
$b_1$	0.0186
$b_2$	1.1236
$b_3$	0.0105
$c_1$	0.0214
$c_2$	0.5736
$c_3$	0.5203

# B

## Parameters for Case Study in Chapter 4

### B.1 Case Study for Energization Transient

B. Parameters of the case study, for energization transient: Voltage source  $G$ : 13.06 kV; Transformer  $T_1$ : 187.5MVA, 65kV/450kV;

Table B.1: Parameters for energization transient case study

Parameters	Values
Voltage source $G$	13.06 kV
Transformer $T_1$	187.5 MVA, 65 kV/450 kV, Y-Y connection

## B.2 Case Study for Fault Transient

Table B.2: Parameters for fault transient case study

Parameter	Values and configuration
Transformer, T1	187.5 MVA, 65 kV/350 kV, $X_{leakage}=0.113$ p.u., Y-Y connection
Transformer, T2	187.5 MVA, 350 kV/60 kV, $X_{leakage}=0.113$ p.u., Y-Y connection
Transformer, T3	187.5 MVA, 350 kV/100 kV, $X_{leakage}=0.113$ p.u. Y-Y connection
Transmission line inductance	mode 0: $5.819 \times 10^{-7}$ H/m, mode +: $1.556 \times 10^{-7}$ H/m
Transmission line capacitance	mode 0: $0.012 \times 10^{-8}$ F/m, mode +: $0.0194 \times 10^{-8}$ F/m
Transmission line length	L1, L2, L3: 10 m, 200 m, 160 m
$Load_1$	100 $\Omega$
$Load_2$	50 $\Omega$ and 10 mH
Voltage source, G	10 kV

### B.3 Geometry Parameters of Transformer

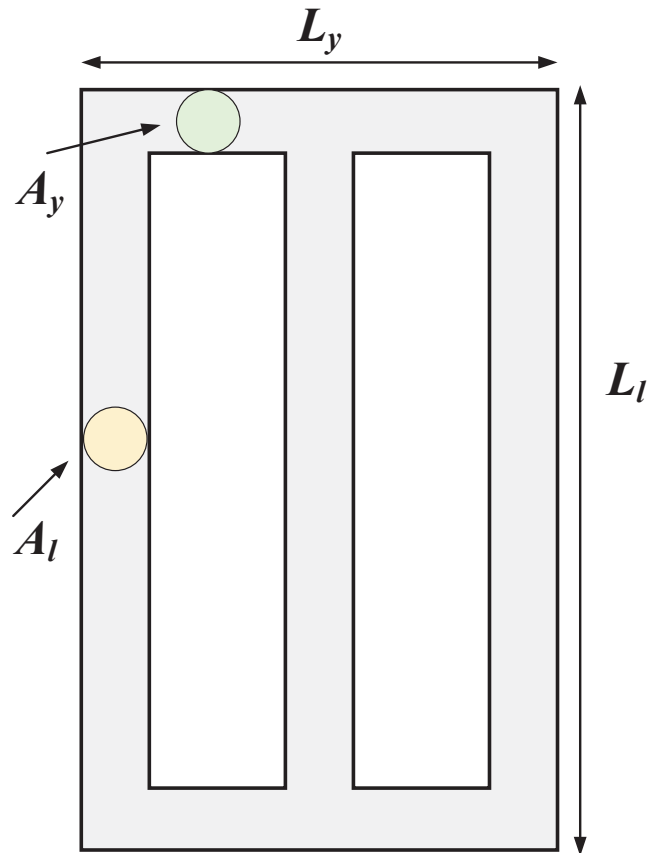


Figure B.1: Transformer geometry parameter.

Table B.3: Parameters of transformer geometry

Parameters	Values
Limb length $L_l$	7.18 m
Limb cross-section area $A_l$	0.454 m <sup>2</sup>
Yoke length $L_y$	2.66 m
Yoke cross-section area $A_y$	0.454 m <sup>2</sup>



# C

## Parameters for Case Study in Chapter 5

### C.1 Geometry Parameters of Transformer

Table C.1: Parameters of transformer geometry

Parameters	Values
Limb length $L_l$	7.18 m
Limb cross-section area $A_l$	0.454 m <sup>2</sup>
Yoke length $L_y$	2.66 m
Yoke cross-section area $A_y$	0.454 m <sup>2</sup>



## C.2 Magnetic Equivalent Circuit Parameters

Table C.2: Parameters for magnetic equivalent circuit

Parameters	Values
Material resistivity $\rho$	$5.2 \times 10^{-7} \Omega/m$
Lamination thickness $d$	0.0005 $m$
Primary winding $X_{leakage}$	0.25 $mH$
Secondary winding $X_{leakage}$	0.08 $mH$
Primary winding turn number	64
Secondary winding turn number	26

Table C.3: Parameters for hysteresis major loop trajectory

Parameters	Values
$a_1$	0.7900
$a_2$	486
$a_3$	312
$b_1$	0.0186
$b_2$	0.011236
$b_3$	0.01898
$c_1$	0.4271
$c_2$	0.3824
$c_3$	1.0406

### C.3 Power System Parameters for Case Study

Table C.4: Parameters for energization transient case study

Parameter	Values and configuration
System base values	$V_{base} = \frac{20}{\sqrt{3}}$ kV
Impedance $R_s$	0.507 $\Omega$
Voltage source $G_1$	1.0 p.u.

Table C.5: Parameters for hysteresis transient case study

Parameter	Values and configuration
System base values	$V_{base} = \frac{16}{\sqrt{3}}$ kV
Impedance $R_s$	0.507 $\Omega$
Voltage source $G_1$	0.8 p.u. and 1.2 p.u.

Table C.6: Parameters for power flow control and fault transient case study

Parameter	Values and configuration
System base values	$V_{base}=34$ kV, $S_{base}=260$ MVA
Transformer $ST$	187.5 MVA, 350 kV/100 kV, $X_{leakage}=0.113$ p.u., Y-Y connection
Transmission line inductance	mode 0: $5.819 \times 10^{-7}$ H/m, mode +: $1.556 \times 10^{-7}$ H/m
Transmission line capacitance	mode 0: $0.012 \times 10^{-8}$ F/m, mode +: $0.0194 \times 10^{-8}$ F/m
Transmission line length	$L_1, L_2$ : 10 km, 12 km
$Load_1$	10 $\Omega$
Impedance $Z_1$	0.5 $\Omega$ and $19.17 \times 10^{-6}$ H
Impedance $Z_2$	0.5 $\Omega$ and $19.17 \times 10^{-6}$ H
Voltage source $G_1$	34 kV
Voltage source $G_2$	34 kV lagging 20 <sup>0</sup>

## **C.4 Tap Changer Setting**

Tap changer: 8 tap positions, 0.05 p.u./step, 0.5 s/step.