

# **FPGA-BASED EMT SIMULATION**

by

XIN MA

A thesis submitted to

The University of Birmingham

for the degree of

**DOCTOR OF PHILOSOPHY**

Department of Electronic, Electrical and Systems Engineering

School of Engineering

The University of Birmingham

March 2023

UNIVERSITY OF  
BIRMINGHAM

**University of Birmingham Research Archive**

**e-theses repository**

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

*Dedicated to*

*My parents*

# ABSTRACT

With the massive integration of renewable energy and power electronics, the structure of modern power system becomes increasingly complex. This is a great challenge for the secure and reliable operation of large power systems. It has been recognised that Electromagnetic Transients (EMT) simulations play an important role in analysing large power system operation, control and protection. Hence it is desirable to investigate faster and more efficient simulation methods for large-scale power systems. The existing off-line software packages, such as MATLAB and PSCAD, are time-consuming to simulate EMTs of large scale power systems. There is a growing demand to develop cheaper and faster EMT simulation technologies/techniques. The rapid evolution of computing technologies makes it possible to enhance EMT simulations in real-time domain. With fast calculation speed, configurable resources and programmable structure, Field-Programmable Gate Array (FPGA) is a powerful platform for EMT simulation. Therefore, this thesis aims to propose high-performance FPGA-based EMT simulation models and algorithms with improved accuracy, speed, and efficiency.

First, Single-Precision, Double-Precision and Mixed-Precision algorithms are proposed and compared to enhance numerical accuracy for the first time. Existing research publications only consider Single-Precision as default option and ignore possible numerical phase shift in FPGA. As a basis for EMT simulation, a library of fundamental power system components is built up, including linear and nonlinear components. Single-Precision and Double-Precision algorithm are using the same precision for all components. Consider component dynamics, the key of Mixed-Precision is simulating linear and non-linear component using Single-Precision and Double-Precision respectively to eliminate phase shift. Hardware structure is also optimized to

make these different precision algorithms achievable on single FPGA board. By comparing with the same referenced models in MATLAB, three algorithms are tested via case studies, including linear components, rotating non-linear components on smaller and larger systems. The adaptability of these algorithms is verified effectively in terms of accuracy, resource utilization, and timing.

Second, four initialization methods are developed to minimize accumulated error for FPGA-based EMT simulation for the first time. Most researches from non-initialized time point, which will increase random error. To simulate from initial steady state, fast and reliable initialization are worth to be investigated. For necessary information, key issues for FPGA-based initialization are discussed first, including initialization model type, memory unit and sequence. Both VHDL file (.VHD) and Coefficient (.COE) file allows defining initial data. To maximize flexibility, four initialization methods, including Method 1 (physical interface), Method 2 (signal declaration), Method 3 (signal assignment) and Method 4 (COE) file are proposed and provided with detailed programming codes. For ahead-of-time evaluation, routing and timing performances are compared between these four methods, and Method 4 is the simplest method. The implementation structure and algorithm of Method 4 are developed to allow flexible data transfer between different platforms. For flexible scalability, device-level and system-level case studies are both provided to compare practical performance of Method 1-4.

Third, a generic MATLAB-to-FPGA toolbox is developed for users to simplify hardware design. This is motivated by FPGA programming is complex and using FPGA to model EMT is more time-consuming for beginners. Without any EMT functionalities, existing translation toolboxes are focused on direct translation from other language to VHDL/Verilog. Therefore, the development toolbox can accelerate beginners to familiarize FPGA-based EMT. In a user-

friendly environment, development framework, design features and requirements are developed for fast processing. For general processing form, data format, structure and partition are developed using intelligent MATLAB built-in functions. To support low-level calculations, translation and resource reutilization for using IP CORES are presented. To integrate and control low-level calculations, high-level main controllers using FSM (Finite State Machine) is setting up sequencers for pipelined and non-pipelined stage. A 39-bus network case study is provided to verify the effectiveness of proposed MATLAB-to-FPGA toolbox.

To support high-frequency switching, power electronic devices and control systems are also developed for FPGA-based EMT simulation. Existing research focuses on using multi-FPGA to simulate HVDC-MMC system, this research aims at implementing whole HVDC-MMC system operation and control on single FPGA platform. This can help reduce FPGA area cost and improve resource utilization efficiency. As a supplement to existing power system components, power electronic devices, such as IGBT and MMC (Modular Multi-level Converter), are built up in discrete-time mathematical models. Based on trapezoidal rule, the aggregate model of MMC is derived to use only one equivalent module to represent all modules, regardless of arbitrary modules. Control system, such as PWM control and current control loop, is also modelled and simplified to be more suitable for hardware implementation. Optimized strategies, such as shift memory and interpolation, are also proposed to get lower resource utilization and faster calculation speed in FPGA. With these strategies, PWM control block and HVDC-MMC case studies can both be successfully implemented on single FPGA board with high-performance accuracy and resource utilization.

# Journal Publications Related to the PhD Thesis

- [1] X. Ma, C. Yang, X. -P. Zhang, Y. Xue and J. Li, "Real-Time Simulation of Power System Electromagnetic Transients on FPGA Using Adaptive Mixed-Precision Calculations," IEEE Transactions on Power Systems, 2022, doi: 10.1109/TPWRS.2022.3199181. <https://ieeexplore.ieee.org/document/9858646>
- [2] X. Ma and X. -P. Zhang, "Initialization Methods for FPGA-based EMT simulation," IEEE Transactions on Power Delivery, under review.
- [3] X. Ma and X. -P. Zhang, "A Generic MATLAB-to-FPGA toolbox for EMT simulation," IEEE Transactions on Power Systems, under review.
- [4] X. Ma and X. -P. Zhang, "FPGA-based HVDC-MMC EMT simulation," IEEE Transactions on Power Systems, in preparation.

# ACKNOWLEDGEMENT

Time flies. I have already spent 3 years and 6 months at the University of Birmingham. I would like to take the opportunity to thank Prof. Xiao-Ping Zhang first. His rich knowledge and encouragement helped me to overcome so many hard barriers. I will never forget how many times we discussed the modelling details and his continuous support motivated me to focus all my energies on research. I always keep in mind that I am so lucky to find such a brilliant supervisor.

For my detailed research, I would also like to thank Prof. Xue Ying, Dr. Conghuan Yang and Dr. Jianing Li. Thanks for their patience when I just started my research. I will never forget how they explain synchronous machines and FPGA to me when I know nothing. Their encouragement and patience helped me to build up my models step by step. I still remembered how we discussed the results difference on software and hardware. This is the valuable experience for my whole life.

When balancing my academic career and life, I would like to thank all of my colleagues. Thanks for them being always kind to me. Especially when I came to the UK, I lacked confidence to communicate with others. It is their patience and lovely help that keep me happy and relaxed. When I feel homesick, it is their patience to cheer me up.

From the deepest bottom of my heart, I would like to appreciate my parents. I am most grateful to their support for my focusing on career I like. I thank for all the pleasure when I communicate with them. Thanks for their patience and understanding of my choice. My career freedom would not be built up without their strong encouragement and continuous support.



Life is short, career is valuable. I will never forget how many times I have been standing on the platform at Birmingham New Street Station. Now I have reached my final destination for the PhD thesis. If I were able to go back to four years ago, I would want to be to cheer up innocent myself at the beginning and not afraid of anything. Trains come and trains go, there are always old passengers and new passengers. My PhD career will not be faded by time but will be shining all the time. Leaving always arrives without notice, you will never know when to get on the train for future career. When we look back to our career, you can always see when you are young and creative. Although future is full of uncertainties, there is no excuse to be fear and afraid.

# CONTENT

CHAPTER 1 INTRODUCTION .....	1
1.1 Background.....	1
1.1.1 Modern Power System Evolution .....	1
1.1.2 The need for fast EMT modelling.....	7
1.1.3 The Importance of Simulation Accuracy .....	12
1.1.4 Challenges.....	13
1.2 Literature Review.....	14
1.2.1 EMT Solution.....	14
1.2.2 Simulation Tools.....	16
1.2.3 FPGA Technologies.....	18
1.2.4 Research Gap .....	21
1.3 Aim, Objectives, Contributions and Outline of The Thesis.....	23
1.3.1 Aim .....	23
1.3.2 Objectives .....	23
1.3.3 Contributions.....	23
1.3.4 Thesis Outline .....	25
CHAPTER 2 SINGLE-PRECISION, DOUBLE-PRECISION AND MIXED-PRECISION ALGORITHMS FOR FPGA-BASED EMT SOLUTION.....	28
2.1 Integration Methods .....	28

2.2 Power System Components Modelling.....	30
2.2.1 RLC branch.....	30
2.2.2 Transmission line.....	33
2.2.3 Transformer.....	39
2.2.4 Synchronous Machines.....	41
2.2.5 Excitation system.....	47
2.2.6 Governor and prime mover system.....	50
2.2.7 Power system stabilizer.....	52
2.2.8 Circuit breaker.....	54
2.3 Single-Precision, Double-Precision and Mixed-Precision Algorithm.....	56
2.3.1 Single-Precision and Double-Precision Comparison.....	56
2.3.2 Mixed-Precision Algorithm.....	59
2.4 Case Study.....	63
2.4.1 Hardware validation.....	64
2.4.2 4-bus linear network.....	67
2.4.3 Synchronous Machine.....	70
2.4.4 11-bus 4-machines network.....	75
2.5 Conclusion.....	82
<b>CHAPTER 3 FOUR INITIALIZATION MEHODS FOR FPGA-BASED EMT SIMULATION.....</b>	<b>83</b>

3.1 Motivation.....	83
3.2 Key Issues of Initialization Models .....	86
3.2.1 Initialization model type .....	86
3.2.2 Memory unit.....	87
3.2.3 EMT model initialization sequence .....	88
3.3 Four Initialization Methods.....	91
3.3.1 Method 1-Physical Interface .....	91
3.3.2 Method 2-Signal Declaration .....	93
3.3.3 Method 3-Signal Assignment .....	94
3.3.4 Method 4-COE File.....	95
3.4 Theoretical Comparisons four methods .....	97
3.5 Implementation Structure and Algorithm .....	100
3.6 Case Study .....	104
3.6.1 Device-level Initialization.....	105
3.6.2 System-level Initialization .....	108
3.7 Conclusion .....	112
<b>CHAPTER 4 MATLAB-TO-FPGA TOOLBOX .....</b>	<b>113</b>
4.1 Motivation.....	114
4.2 Development Framework.....	116
4.2.1 Overall framework .....	116

4.2.2 Design features.....	121
4.2.3 Software-to-hardware Platform requirements.....	122
4.3 Memory unit.....	123
4.3.1 Data Format .....	123
4.3.2 Data Structure .....	124
4.3.3 Data Partition .....	127
4.4 Sub-module .....	130
4.4.1 Translation .....	130
4.4.2 Reutilization and simplification.....	132
4.5 Main controller.....	136
4.5.1 FSM controller structure .....	136
4.5.2 Pipelined stage .....	137
4.5.3 Non-pipelined stage .....	138
4.5 Case study .....	140
4.6 Conclusion .....	145
<b>CHAPTER 5 FPGA-Based Power Electronic Devices and Control System Simulation .....</b>	<b>147</b>
5.1 Motivation.....	147
5.2 Power Electronic Device Modelling.....	148
5.2.1 Diode.....	148
5.2.2 IGBT .....	150

5.2.3 Half-Bridge MMC .....	150
5.3 Control System Modelling .....	152
5.3.1 Discrete mean value .....	152
5.3.2 Variable mean value .....	153
5.3.3 Current control loop .....	156
5.3.4 PWM control .....	158
5.4 Hardware Implementation Strategies .....	159
5.4.1 Shift memory unit .....	159
5.4.2 Partitioned pipeline design .....	160
5.4.3 Switch block .....	161
5.4.4 Interpolation .....	162
5.4.5 Different time step interface .....	163
5.5 Case study .....	164
5.5.1 PWM control block .....	164
5.5.2 HVDC-MMC control system .....	166
5.6 Conclusion .....	171
CHAPTER 6 CONCLUSIONS AND FUTURE WORK .....	172
6.1 Conclusions .....	172
6.2 Future works .....	175
REFERENCES .....	178

APPENDIX TEST SYSTEMS .....	194
A.1 FPGA IP Cores Latency.....	194
A.2 4-bus network.....	194
A.3 Synchronous machine and control system .....	195
A.4 11-bus 4-machines network .....	198
A.5 39-bus 10-machines network .....	199
A.6 HVDC-MMC interconnection .....	201

# LIST OF FIGURES

Fig. 1.1 Maps showing the current ownership landscape for energy network companies in Great Britain	4
Fig. 1.2 Different time scales of power system transients	11
Fig. 1.3 Comparison of real-time and off-line simulation	16
Fig. 1.4 A typical digital design	20
Fig. 2.1 IEEE AC4A exciter system [1]	48
Fig. 2.2 Steam governor and non-reheat turbine [1]	51
Fig. 2.3 Power system stabilizer	53
Fig. 2.4 Double-precision and Single-precision format comparison	57
Fig. 2.5 Relative error accumulation in sine wave function between double-precision and single-precision format	57
Fig. 2.6 Mixed-Precision EMTP system framework composed of linear and non-linear components	60
Fig. 2.7 Simulation algorithm for hybrid double-precision and single-precision floating point	61
Fig. 2.8 Virtex6 ML605 FPGA board	63



Fig. 2.9 The Place and Route Report	65
Fig. 2.10 Timing errors	66
Fig. 2.11 Simulation topology of 4-bus network	68
Fig. 2.12 Simulation results of currents of a 4-bus system. (a) Long-duration comparison (b) Short-duration comparison (c) Absolute error comparison	70
Fig. 2.13 Synchronous machine with AC4A excitation system, PSS and governor system	71
Fig. 2.14 4 types of synchronous machine simulation results	74
Fig. 2.15 4-machine 11-bus test system	76
Fig. 2.16 Time schedule of Mixed-Precision structure	76
Fig. 2.17 Results of synchronous machine G1 in Double-Precision and Mixed-Precision when applying single-phase grounding fault	78
Fig. 2.18 Results of synchronous machine G1 in Double-Precision and Mixed-Precision when applying three-phase grounding fault	80
Fig. 3.1 Random error of non-initialized calculation	84
Fig. 3.2 Initialization memory data structure	88
Fig. 3.3 Initialization procedure comparison	98
Fig. 3.4 Initialization global time schedule comparison	99

Fig. 3.5 Initialization schematic comparison	99
Fig. 3.6 Processing algorithm for novel initialization method	101
Fig. 3.7 Hardware structure design for off-line data flow initializing real-time module.	102
Fig. 3.8 Synchronous machine simulation model	105
Fig. 3.9 Results of synchronous machine G1 with initialization and without initialization.	107
Fig. 3.10 4-machine 11-bus test system.	109
Fig. 3.11 Results of synchronous machine G2 with initialization and without	111
Fig. 4.1 Difference between intelligent and independent programming	116
Fig. 4.2 Data formatter for repetitive calculations	118
Fig. 4.3 Global workflow for the proposed toolbox	120
Fig. 4.4 MATLAB-to-FPGA hardware platform.	122
Fig. 4.5 Data structure for synchronous machine	125
Fig. 4.6 Data structure for ideal voltage source	126
Fig. 4.7 Data structure for RLC branch	126
Fig. 4.8 Data structure for transmission line	127

Fig. 4.9 Data standardization	129
Fig. 4.10 Simple pipelined translation	131
Fig. 4.11 Complex non-pipelined translation	132
Fig. 4.12 Matrix decomposition structure.	133
Fig. 4.13 Hardware design for 8-input vector multiplication.	134
Fig. 4.14 Hardware connection of RLC branch current.	134
Fig. 4.15 Essential translation structure for main controller and sub-modules.	136
Fig. 4.16 Programming logic for pipelined stage	137
Fig. 4.17 Programming logic for non-pipelined stage	139
Fig. 4.18 Generated VHD files by MATLAB-to-FPGA toolbox	141
Fig. 4.19 10-machine 39-bus test system.	143
Fig. 4.20 Synchronous machine G1 with/without initialization.	144
Fig. 4.21 Time schedule of 39-bus system.	145
Fig. 5.1 Simple diode structure	149
Fig. 5.2 Equivalent circuit for diode	149
Fig. 5.3 Aggregate model for MMC	151
Fig. 5.4 Discrete mean control block	152

Fig. 5.5 Variable mean control block	154
Fig. 5.6 D axis current control block	156
Fig. 5.7 Update modes for three RAM types	160
Fig. 5.8 Data partition for 32-input multiply-add block	161
Fig. 5.9 Switch design for different topologies	162
Fig. 5.10 Interpolation for periodic and non-periodic function	162
Fig. 5.11 Transfer directions between different speed data	164
Fig. 5.12 PWM Generator	164
Fig. 5.13 PWM signal result comparison	165
Fig. 5.14 HVDC-MMC topology	167
Fig. 5.16 HVDC-MMC simulation results	169
Fig. 5.17 Time schedule of HVDC-MMC system.	169

## LIST OF TABLES

Table 1.1 Typical HVDC Interconnector Projects between UK and Europe	3
Table 1.2 Comparison between National Grid's transmission and distribution network	6
Table 1.3 Installed Capacity of renewable energies in UK	6
Table 1.4 Characteristics of lead-acid batteries versus lithium-ion batteries	7
Table 1.5 Summary of fault types	8
Table 1.6 Typical Transient events	9
Table 1.7 Frequency classification of transients	10
Table 1.8 Price comparison between RTDs and FPGA	14
Table 2.1 Summary of the comparison between different RLC branches	32
Table 2.2 Comparison between lumped and distributed parameter model	35
Table 2.3 Summary of the comparison between addition, sine function and cos function	59
Table 2.4 Summary of Floating-Point Core and Block Memory Generator.	64
Table 2.5 Resource Comparison on A 4-bus Electric Network without SGs	69
Table 2.6 Simulation resource utilization of synchronous machine models of 4 types in ML605	75

Table 2.7 Resource Comparison of Kundur’s System Case Study	77
Table 3.1 Features of Required Variables in EMT Initialization	87
Table 3.2 Programming code for Method 1	91
Table 3.3 Programming code for method 2	93
Table 3.4 Programming code for Method 3	94
Table 3.5 Programming code for Method 4	95
Table 3.6 Real-time Occupation for Initialization Methods	98
Table 3.7 Resource Comparison of Single SG	105
Table 3.8 Timing Comparison of Single SG	106
Table 3.9 Resource Comparison of 4-machine 11-bus Test System	109
Table 3.10 Timing Comparison of Single SG	109
Table 4.1 General format of input data	124
Table 4.2 Generated files details	140
Table 4.3 Resource utilization between 11-bus and 39-bus	141
Table 5.1 Resource utilization of HVDC-MMC system.	169
Table A.1 Latencies of IP COREs	194
Table A.2 Parameters of 4-bus network	195

Table A.3 Synchronous machine parameters [1]	196
Table A.4 IEEE AC4A excitation model parameters [1]	197
Table A.5 Non-reheat steam governor parameters [113]	197
Table A.6 Power system stabilizer parameters [1]	198
Table A.7 External network parameters	198
Table A.8 Transmission lines for 11-bus 4-machines system [1]	199
Table A.9 Transmission lines length for 39-bus 10-machines	200
Table A.10 Parameters of HVDC-MMC Interconnection system	202

## **LIST OF ABBREVIATIONS**

AC	Alternating Current
ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Blocks
CPU	Central Processing Unit
DC	Directive Current
DSP	Digital Signal Processing
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
FSM	Array FSM Finite State Machine
HIL	Hardware-in-the-Loop
HVDC	High Voltage Direct Current
I/O	Input/Output
LUT	Look-Up Tables
MMC	Modular Multilevel Converter
PLL	Phase Locked Loop
PV	Photovoltaics



PWM	Pulse Width Modulation
RAM	Random Access Memory
ROM	Read-Only Memory
RTDS	Real-Time Digital Simulator

# **CHAPTER 1 INTRODUCTION**

This chapter gives the brief background introduction and carries out the literature review of the power system evolution and EMT simulation requirement. Then based on these, the related research gaps and research objectives are presented. To fill the research gap, the thesis contributions are introduced. At the end of the chapter, the thesis outline and the relationship between chapters are provided.

## **1.1 Background**

### **1.1.1 Modern Power System Evolution**

The AC power system development can back to 1880s when AC transformer and transmission line were developed [1]. This motivated the first commercial use of AC transmission line in North America, which was a single-phase line over 21 km distance. The first three-phase transmission line was put into operation in North America in 1883 [2]. For further flexible interconnection, various frequency (30Hz, 40 Hz and 50 Hz) were standardized eventually as 60 Hz and 50 Hz depending on locations.

The purpose of the electric transmission system is the interconnection of the electric energy producing power plants or generating stations with the loads over 100kV [3]. The topologies of transmission network are normally networked and meshed topology infrastructure. For example, the transmission system of UK is managed by National Grid and operated typically

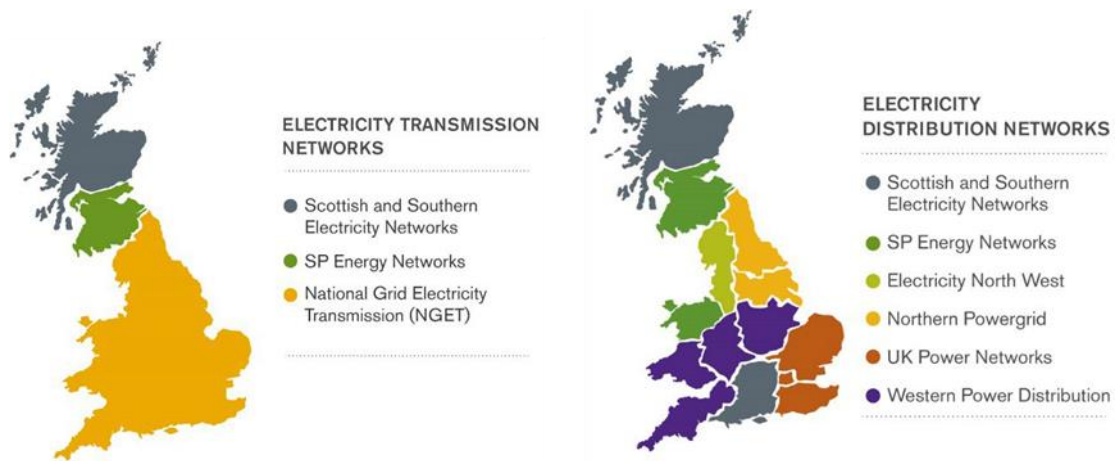
at extra high voltages of typically 275 or 400 kV [4]. Since the invention of mercury-arc rectifiers, high-voltage transmission system became attractive. HVDC can efficiently meet the demand of transmitting power at very long distance [5]. In addition to that, HVDC is able to isolate the frequency between two systems with different nominal frequency, when AC transmission line is not possible. This advantage became more significant than AC transmission line when overhead lines is about 500 km and underground cable is about 50 km [6]. For example, the interconnects between UK and other European countries are via HVDC. Table 1.1 details recent and future HVDC interconnector projects between UK and Europe [7].

Distribution grids deliver electricity to end-users at lower voltage levels [8]. Electric power was originally distributed by radial DC systems for heavy-density load areas. Because of fixed the utilization voltage, the DC system was not suitable for the more extensive lighter-load areas [9]. Since the introduction of AC, these areas were served by overhead AC systems of the radial type [1], where the primary feeders radiate from the distribution substations and branch into sub feeders and laterals. Typical distribution topologies include radial and meshed networks. Since 1950s, distribution voltages rose into 25kV and 35kV. Recently, the distribution voltages for North American ranges from 4kV to 35kV [10]. The UK distribution networks operate at voltages of typically 400V, 11kV, 33kV, 132 kV [11], which are managed by distributed network operators and independent distribution network operators. Fig. 1.1 presents the ownership for electricity transmission and distribution networks in the UK [12].

In the UK, National Grid Company has been working to transmit electricity safely and efficiently over long distances since 1990. Table 1.2 compares some key factors between transmission system and distribution system of National Grid [13].

**Table 1.1 Typical HVDC Interconnector Projects between UK and Europe [7]**

Project name	Connecting country	Capacity	Distance	DC voltage	AC voltage	Delivery date
NSL	Norway	1400MW	720 km	400kV	515 kV	2021
ElecLink	France	1000MW	51 km	400 kV	320 kV	2022
Viking Link	Denmark	1400MW	765 km	400 kV (Jutland), 400 kV (Bicker Fen)	525 kV	2023
Greenlink	Ireland	500MW	200 km	400 kV (UK); 220 kV (Ireland)	320 kV	2023
GridLink	France	1400MW	140 km	440 kV	525 kV	2024
NeuConnect	Germany	1400MW	720 km	380 kV GER/400 kV UK	525 kV	2024
FAB Link	France	1400MW	220km	400kV	320kV	2025



**Fig. 1.1 Maps showing the current ownership landscape for energy network companies in Great Britain [12]**

**Table 1.2 Comparison between National Grid’s transmission and distribution network [13]**

	Electricity transmission system	Electricity distribution system
Map	21,900 steel pylons ranging from 118ft (36m) to 623ft (190m) tall	7.9 million customers across an area 55,500 square kilometres
Cables	4,500 miles of overhead line and 900 miles of underground cable	60,000 miles of overhead line and 83,900 miles of underground cable
Voltages	Electricity is carried at higher voltages of 275kV and 400kV	Electricity is distributed at lower voltages of up to 132kV
Electricity conversion	Over 300 substations to convert electricity into different voltages	188,000 transformers in substations to lower voltage to safe levels

Along with power system evolution, power electronics is also going through a vast evolution and creating potentials for power systems. The modern semi-conductor era started with the invention of diode, thyristor, and gate-turn-off thyristor in 1950s. Especially thyristors are still extensively used in HVDC and inverter-controlled synchronous machine until now. After the insulated gate bipolar transistor (IGBT) was developed in 1985, it soon became attractive for medium voltage level application. Recently, wide bandgap semiconductor components are emerged, such as SiC [14] in 2009, which have higher tolerance for voltage level, switching frequency and junction temperature. Based on existing material, some research works even developed advanced hybrid materials recently, for example, [15] proposed a hybrid silicon carbide (SiC)-silicon (Si) DC-AC topology to improve performance in high power application.

The integration of renewable energies and energy storages also have changed the nature of power systems, which is driven by government targets and national legislations. With the 2030 Climate Target Plan, the EU has set binding climate and energy targets for 2030: reducing greenhouse gas emissions by at least 40%, increasing energy efficiency by at least 32.5%, increasing the share of renewable energy to at least 32% of EU energy use and guaranteeing at least 15% electricity inter-connection levels [16]. By 2030, half of Scotland's total energy demand (electricity, heat and transport) is aimed to be supplied by renewable sources [17]. Projects such as HyDeploy, H21, H100, Aberdeen Vision and Cavendish, pilot approaches to decarbonize gas fuel [18]. National Statistics publication Digest of UK Energy Statistics (DUKES), produced by the Department for Business, Energy and Industrial Strategy (BEIS), has presented renewable energies capacity in UK in Table 1.3 [19].

**Table 1.3 Installed Capacity of renewable energies in UK [19]**

Installed Capacity (MW)	2017	2018	2019	2020	2021
Wind:	19585	21605.62	23886.91	24457.59	25747.88
Marine energy	18.4	20.4	22.4	22.4	22.4
Solar photovoltaics	12760.03	13059.07	13344.84	13578.96	13964.92
Hydro:	1869.64	1877.19	1879.86	1886.15	1891.32
Solid biomass:	3149.51	4592.59	4683.82	4693.42	4702.15
Biogas:	1819.05	1837.13	1843.65	1844.28	1922.7
Energy from waste	1090.93	1136.73	1309.77	1434.93	1450.9
Total Installed Capacity	40292.56	44128.73	46971.25	47917.73	49702.27

To maximize the use of renewable energies, Energy Storage Systems (ESS) are offering economic advantages to customers and providing grid service [20]. ESS are classified into mechanical, electrochemical, chemical, electrical, thermal, and hybrid. Flywheel, secondary electrochemical batteries superconducting magnetic coils, and hybrid ESS are commonly used in power applications [21]. The most commonly used domestic Battery Energy Storage Systems (BESS) are Lead-Acid and Li-on. Table 1.4 shows the characteristics of Lead-Acid and Li-on batteries [22].

**Table 1.4 Characteristics of lead-acid batteries versus lithium-ion batteries [22]**

Type	Energy Density	Voltage V	Electrolyte	Usage	Advantages	Disadvantages
Lead-acid	< 40	2	Sulphatic acid	Mass production for SLI, stationary, fork lift trucks.	Cost effective Mature technology Power capability at low temperature.	Low energy density (high maintenance) Life at elevated temperature Hazardous materials.
	There are different types: Open, SMF (sealed maintenance free), VRLA – AGM or Gel-type.					
Li-ion	100-260	3.2-3.8	Organic solvent	Mass production for portable, stationary, industrial and HEV/EV applications.	High energy power Cycle life efficiency.	Cost Safety Maturity (large format) Temperature range.
	Family of chemistries resulting in different properties. Most common; LCO, NMC, LFP.					

For future power systems, it is necessary to analyse at different time scales, system sizes, voltage levels and component techniques. Since EMT phenomena can happen in tens of microseconds, EMT modelling is particularly significant for power system stability analysis. While the real-time operational analysis for growing power systems and interconnected networks is indeed a challenging research problem.

### 1.1.2 The need for fast EMT modelling

The operation of power systems is affected by faults at different positions of AC systems, converters and DC transmission lines. Table 1.5 gives typical fault types, including single-phase, two-phase, or three-phase grounded or ungrounded fault [23]. There are three reasons why single-phase-to-ground fault is the most significant one in EMT study: First, the single-



phase-to-ground fault can even trigger higher frequency transients compared to the three-phase-to-ground fault [24]. Second, single phase to ground fault (asymmetrical fault) is more influential than a three-phase to ground fault (symmetrical fault) because asymmetrical fault will lead to different voltage drop in each phase [25]. Third, single-phase-to-ground fault occurs more frequently than other types of faults [26]-[27]. Therefore, single-phase-to-ground fault is the researching focus for most EMT analysis.

**Table 1.5 Summary of fault types**

Fault type	Examples
Single-phase-to-ground fault	A-G, B-G or C-G
Two-phase fault	A-B, B-C, or A-C
Two-phase-to-ground fault	A-B-G, B-C-G, A-C-G
Three-phase fault	A-B-C
Three-phase-to-ground fault	A-B-C-G

Transients are outward manifestation of sudden change in power system conditions when a circuit breaker opens and/or a fault occurs [26]. Transients can result in various overvoltage, overcurrent and distorted waveforms. gives typical transients events and frequency ranges [27]. Seen from Table 1.6, different transient phenomenon corresponds to different frequency ranges. To describe transient phenomenon more accurately, it is essential to give a generic classification depending on frequency. One of the most acceptable classifications is proposed by International Electrotechnical Commission (IEC) and CIGRE [28]-[30]. Table 1.7 shows

frequency ranges are divided mainly into four different groups in terms of dynamics of frequency concerned [30].

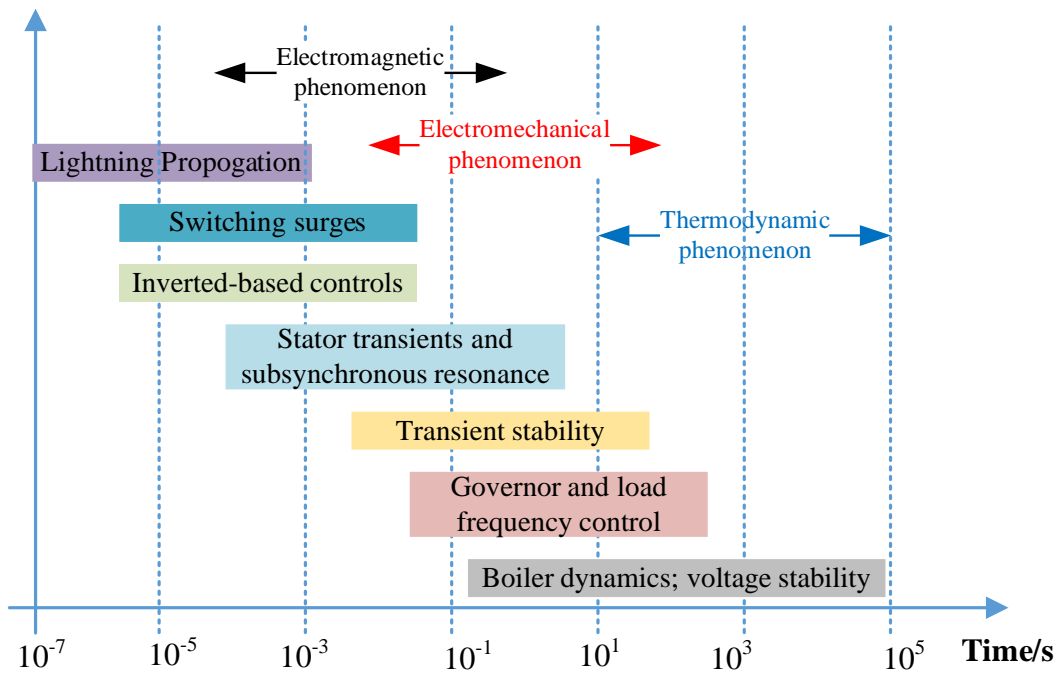
**Table 1.6 Typical Transient events**

Typical transients	Frequency
Ferroresonance	0.1 Hz to 1 kHz
Load rejection	0.1 Hz to 3 kHz
Fault clearing	50 Hz to 3 kHz
Line switching	50 Hz to 20 kHz
Transient recovery voltages	50 Hz to 100 kHz
Lightning overvoltage	10 kHz - 3 MHz

**Table 1.7 Frequency classification of transients**

Transients	Frequency
low-frequency oscillations	0.1 Hz -3 kHz
slow-front surges	50/60 Hz -20 kHz
fast-front surges	10 kHz - 3 MHz
very fast-front surges	100 kHz - 50 MHz

The simulation of transients involves Electromagnetic Transients (EMT) and Electromechanical transients for different power system components. EMT mainly occur in magnetic field and electric field due to voltage and current change, such as lighting strike and switching operations [31] - [32]. Electromechanical transients are mainly affected by the interconnections between mechanical energy stored in rotating machines and electrical energy stored in the network, such as rotor swings [32]. Electromechanical transients are much slower than EMT. Normally, the typical simulation step of electromechanical transient is 10 *ms*, while typical simulation step for EMT is 20-200  $\mu s$  [33] - [34]. Fig. 1.2 gives the time ranges for EMT phenomenon and electromechanical phenomenon [35] - [36]. With the development of computational technologies and techniques, the traditional electromechanical transients tools can be replaced by EMT tools. Depending on particular stability studies, dedicated assumptions are required to be made on component modelling to represent the limited transients bandwidth. Moreover, EMT occur more frequent in modern power systems than before due to the wide use of power electronic converters to support renewable energy integration, therefore it is essential to develop fast and even real-time EMT technologies and techniques.



**Fig. 1.2 Different time scales of power system transients**

The simulation speed is always a key factor for limiting EMT simulation performance. Without losing information, there is a huge potential to improve calculation speed by balancing and eliminating ignorable elements. For example, it is possible to represent the overall performance of multi-arms MMC by equivalent circuit [37]. Therefore, hundreds of nodes within internal submodule will not appear in external admittance matrix, which can improve the calculation speed efficiently for solving real-time EMTs.

The component modelling has to be straightforward and general for large-scale systems. The modelling is required to maintain the calculation manageable and achievable, such as admittance matrix solution. Otherwise, the calculation burden can be an unavoidable barrier for preventing scaling up system. It is necessary to propose general and simple modelling for long-term real-time simulation.

For power electronic devices, IGBT and diode are considered as two-state resistances [38] for most application circuits. It is efficient for system-level simulation. But specific simulation step is ignored, the determination of switching state will take only several microseconds. To model this detailed situation, a small time-step is required for large scale systems, for instance, HVDC system [39]. Therefore, small time step and fast calculation speed is necessary for HVDC to ensure real-time EMT simulation for large scale power systems.

### **1.1.3 The Importance of Simulation Accuracy**

The development of arithmetic operation units, including integer, Floating-Point and Binary values [40], has advanced microprocessors to handle more complex process in hardware. To process reasonable power system accuracy, it is desirable to develop in-depth knowledge in arithmetic calculation, testing techniques and computer-aided design [41]. Considering multi disciplines, the target is to eliminate heavy calculation burden and maintain calculation precision at the same time. Otherwise, there is a potential to damage practical system and components to different degrees.

For power electronic devices, diode, IGBT and MOFETs [42] are widely utilized in modern power system applications, such as converters, inverters and power electronic interfaces, etc. The estimation of power losses and switching states of these semiconductors has become a major barrier for continuous operation when switching frequency is extremely high. Inaccurate estimation can reduce utilization duration and capacities. Moreover, it can lead to break down power electronic control system, such as Pulse-width-modulated rectifiers. Therefore, accurate prediction of power electronic devices is significant for enhancing device maintenance and management. Previous efforts to address the problem of accuracy include using pre-defined

curves and interpolation, which is only for unsuitable for real-time simulation owing to heavy calculation burden.

Once the switching states of power electronic devices are determined, it becomes necessary to solve the dynamic states of other power devices. Uncertain state estimations can also result in oscillations and detect faults unreliably. For power devices, the dependency of angular speed, current, voltage and frequency have to be taken into consideration together. Even slightly difference on one factor can have a huge impact on other factors and remove them from appropriate operation.

The most common method is to solve these problems in a graphical software and hardware. Existing packages, for instance, PSCAD and RTDS [43], are commercial simulation packages. Users can only select existing components in the menu. It is difficult to involve self-developed components. This disadvantage limits user's potential to develop fundamental functionalities of emerging or new EMT components.

#### **1.1.4 Challenges**

Based on introduced background, exiting challenges can be summarized as follows:

- First, the multi-dimension relationship between different factors, such as frequency, voltage and current, is very complicated for different components. This coupling problem is more severe to handle in tiny simulation step and scaling-up system scale.
- Second, simulation software and hardware are always too expensive and cost much volume to for beginners. For example, RTDS, costs more than thousands pounds, is usually unable for undergraduate and postgraduate. Table 1.8 shows the price difference between RTDs and FPGA [44].

**Table 1.8 Price comparison between RTDs and FPGA**

	Price (€)	Savings (%)
FPGA board	5800	75%
RTDS PB5 board	24000	-

- Third, EMT simulation tool requires much longer simulation time and much more efforts for users to develop familiarities. Especially in a graphical software, it is extremely challenging to create self-defined models to be connected with build-in black-box models in consistently stable state.
- Fourth, the integration of renewable energies has changed the nature of modern power systems. Faster EMT simulation is desired for testing protection and control equipment.
- Finally, simulation accuracy is desirable to be improved for different platforms to provide high confidence simulation results.

## **1.2 Literature Review**

### **1.2.1 EMT Solution**

In power system operation, transients may interrupt normal operation, damaging device operation and protection. Typical EMT oscillations consist of energization of capacitors, synchronous machine start-up, fast switching-on breakers, transformer saturation, overhead line grounding. These uncertain strikes can happen at harmonic or fundamental frequency depending on system structure and oscillation type. It is complex to predict coupling voltage, current and frequency at these phenomena. Therefore, accurate digital EMT simulation is

always irreplaceable and significant for design, planning and operation of modern power system.

In order to obtain time responses for Electromagnetic Transients, the general solution for solving arbitrary phases and parameters is firstly proposed by Dommel in 1969 [45]. Different solutions have been proposed for single-phase RLC branch and three-phase coupling transmission line. Considering transient oscillations are more common in non-linear components, a composition method theory to combine time-varying and non-linear components was proposed [46], including switching-off breakers and saturation transformers. After that, he presented the comprehensive theory in EMTP theory book [47]. Even for linear transmission lines, they can be modelled by basic lumped parameter model, traveling wave model and time-varying model. For general-purpose application, three-phase synchronous machines and universal machines were both developed to represent major types of electric motors, such as induction machines and DC motors [48].

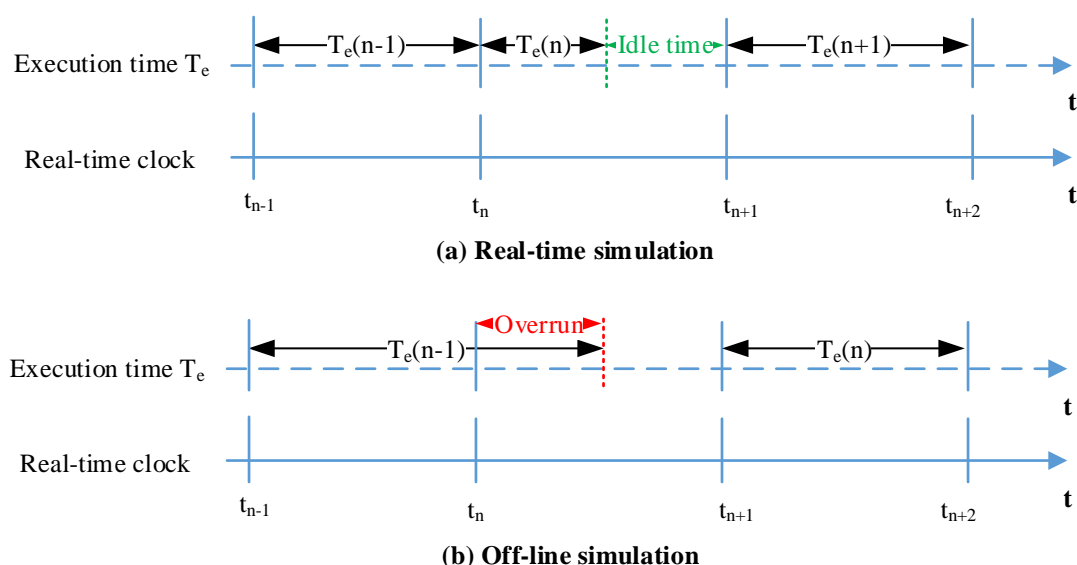
Computer-aided program [49] is unable to solve EMT in continuous state, and it can only provide solutions at discrete time intervals. Without reasonable method, the relative truncation error can increase from step to step and cause divergence from the true solution. Typical numerical methods include trapezoidal rule, back Euler and forward Euler [50]. By the mathematical proof in [51], trapezoidal rule shows the best performance for numerical stability. Recently, these methods are both used in off-line software (such as SIMULINK and PSCAD) and real-time packages (for instance, RTDS) [52]. Different numerical method also possibly leads to different simulation results at specific cases.



## 1.2.2 Simulation Tools

To explore the detailed behaviours of power systems, it is necessary to construct circuits and observe simulation results. It takes a lot of efforts and time for beginners to obtain familiarity with laboratory experiments. Considerable design techniques and analysis skills are normally required. Depending on execution time and simulation step, EMTs solutions consist of off-line simulation method and real-time simulation method. Fig. 1.3 illustrates the difference between real-time simulation and off-line simulation [53]:

- 1) The simulation is considered to be real-time if execution time  $T_e$  is shorter than the selected time step.
- 2) The simulation is considered to be off-line if execution time  $T_e$  is greater than one or more time steps, in which overruns occur.



**Fig. 1.3 Comparison of real-time and off-line simulation [53]**

Off-line software aims to solve mathematical equations as fast as possible, and the actual calculation time at each time step would be normally much longer than the simulation time

step. CPU time required to compute the modelled system's response might take several seconds or minutes, longer than the actual response [54]. Solving speed depends on available computation power and mathematical model complexity [55]. Typical off-line software includes Power System Computer Aided Design (PSCAD), Electromagnetic Transients Program (EMTP) and MATLAB. PSCAD is a power graphical interface for Electromagnetic Transient simulation Engine, allowing users to build up circuits and observe results [56]. Based on PSCAD/EMTPC program, a graphical simulation laboratory was introduced for undergraduates and postgraduates [57]. In [58], an analytic method is proposed to predict core saturation instability for AC and DC operations. Besides, a continuous small-signal model for High voltage direct current transmission line was detailed using PSCAD in [59]. In [60], a phase-disposition sinusoidal pulse-width modulation strategy was presented for MMC operation in PSCAD environment. In [61], a decentralized control strategy based on the two-stage modified droop method was proposed for AC and DC connection.

MATLAB is a mature simulation tool with user-friendly environment, in which SIMULINK is also a flexible toolbox for EMT solutions. In [62], a novel average-value method for modelling six-pulse front-end rectifier was implemented. In addition, a fault ride-through capability was explored for VSC-HVDC EMT model based on SIMULINK [63]. A dynamic model for multi-terminal VSC-HVDC was proposed by compared with assembled model in SIMULINK [64]. Moreover, SIMULINK has great flexibility to interface with other simulators. For example, a general interface was developed between PSCAD and SIMULINK to compare different simulation environments for HVDC benchmark [65]. In addition to that, a close loop between RTDs and SIMULINK was interfaced for implementing automatic voltage control regulator [66].

Real-time simulator needs to solve model equations for one time step within the same time in real-world clock [53]. In other words, faster-than-real-time calculation speed is desirable for real-time simulators [67]. This permits the real-time simulator to perform all operations necessary to make a real-time simulation relevant, including driving inputs and outputs to and from externally connected devices [56]. In 1991, the first commercial real-time digital simulator (RTDS) [68] was demonstrated using DSPs by RTDS Technologies Inc. RTDS was developed specifically for running real-time power system simulations and communicates in real time with the simulation hardware [68]. The structure and performance of RTDS for testing relays was firstly introduced in [69]. A novel control method for PV under real weather conditions was proposed in [70] using RTDS. In addition, a real-time Ultra-High-Voltage model was totally exploited in [71]. Protective relays for super conductive limiters were exploited using RTDs.

However, RTDS is too expensive for beginners owing to extra features. It is essential to overcome this disadvantage by investigating alternative real-time simulators. Therefore, it is of great significance to develop novel high-performance EMT solution technologies.

### **1.2.3 FPGA Technologies**

Field Programmable gate arrays are consisting of integrated configurable logics connected via programmable interconnects in one board. Most FPGA boards, such as ML605 and Nexys 40 [72], are allowed to be reprogrammed time after time. Its rich resources and reprogrammable functionality give designers great opportunities to implement high-level design and tremendous simulation tasks.

FPGA was firstly developed to implement glue logic and state machines in 1980s [73]. Increasing size and complexity of FPGA made it possible to reduce manufacturing time from months to minutes. Moreover, much lower prototype cost also inspired huge market of FPGA in telecommunications during 1990s. At that time, the trade-off relationship between implementation functionality and circuit area were explored in [74]. FPGA routing flexibility was determined by the amount and distribution of switches in interconnection structures in [75]. Therefore, FPGA was also utilized to handle various types of data, including text, audio and image in [76].

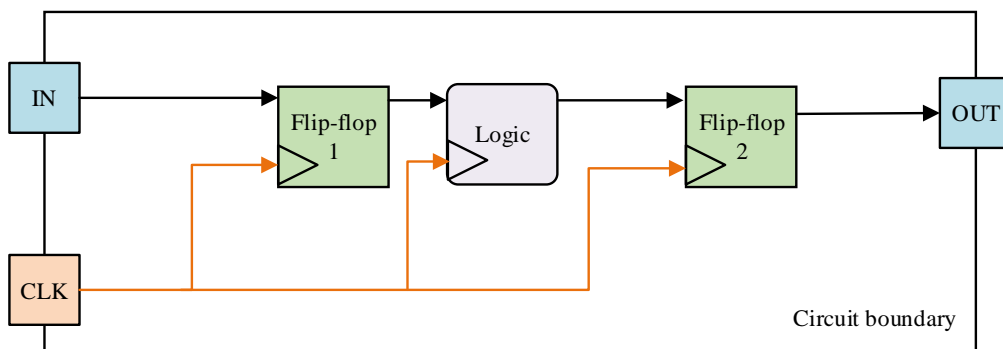
Low simulation cost and short implementation time made them special market to compete with ASICs. In [77], experimental measurements of FPGA and ASIC circuits were presented to give performance comparison in terms of logic density, circuit speed, and cost.

During 2000s, Xilinx S-FPGAs were used to control critical pyrotechnics and control the rover wheel motors. After that, Xilinx Virtex-II was developed with several hundred general pins and hundreds of configurable components [78]. Related fault injection experiments were also derived to get greater input and output performance in [79]. Xilinx 7 Series were released in June of 2010 and fabricated by 28-nm process technology [80].

FPGA design tools were also developed for different versions, including ISE and VIVADO. ISE (Integrated System Environment) [81] was developed in, mainly for previous languages, including VHDL 2008. The latest version of ISE is 14.7 and was replaced by VIVADO since 2013. VIVADO was firstly introduced as a synthesis and analysis for hardware design by Xilinx in 2012, and its high-level compiler even enables C and C++ to be downloaded into

targeted board. In [82], the routing algorithm of VIVADO was totally explored for faster routing delays.

For the purpose of real-time simulation, RTDS uses parallel processing architecture based on a state-of-the-art Digital Signal Processor (DSP) to run power system simulations in real time with a time step of 50-100 microseconds [83]. Similar to DSP, the simulation step of FPGA can be very small in the order of 250 nanoseconds driven by internal clock [55]. This is because an internal clock is responsible to drive the FPGA design and determines the process data speed. Fig. 1.4 presents the typical digital design for FPGA [84]. The clock signal is connected to all flip flops, RAM blocks, and I/O ports, activating them according to the clock frequency [85]. For example, Nexys A7 board is equipped with a 100 MHz system clock, thus the highest data rate of 650Mbps is recommended [86]. Compared with sequential CPU, FPGAs are massively parallel processing devices. The simulation starts immediately once the download is finished. If timing constraints are met for FPGA, overruns can not occur during the EMT model is running [87]. With high-frequency clock, inherent parallel architecture and high density [88], FPGA is capable of implementing real-time EMT solutions.



**Fig. 1.4 A typical digital design [84]**

With significant computational power, FPGA is able to calculate power system devices to the amplitude of nanoseconds. As a result of that, FPGA was firstly as computational engine to implement inductive machine drive in real-time [89]. Solving networks with transmission lines and machines was implemented using arithmetic Floating-Point calculation in [90]. Moreover, high-resolution magnetic circuit was also considered for transformers based on hardware-in-loop structure in [91]. For circuit acceleration, a matrix-inverse technique was introduced for high-frequency power converters in [92]. Besides, FPGA is flexible to be compatible with other platforms, [93] proposed a FPGA-MPSOC platform for power systems using iterative and non-interactive loops.

#### **1.2.4 Research Gap**

FPGA-based EMT simulation is a comprehensive process, different from off-line software which is only focused on accuracy. In addition to improving accuracy, FPGA-based EMT simulation is facing to deal with complex timing constraint, overused resource utilization and calculation sequence exclusively. Therefore, FPGA design and analysis requires in-depth background of hardware technologies, EMT simulation tools. Specifically, the following research gap needs to be considered.

- First, how to develop an optimized hardware design for FPGA based real-time EMT solution by limited timing constraint and resource utilization?
- Second, how to improve simulation accuracy and efficiency compared with existing FPGA-based real-time simulation implementations?
- Third, what is the best trade-off for achieving higher accuracy and using lower system constraints FPGA based real-time EMT simulation?

- Fourth, to what extent, can we reach more flexible and scalable FPGA-based real-time simulation for scaling-up system?
- Fifth, how to integrate hardware programming efficiently into a purely intelligent software-based programming for FPGA based real-time EMT simulation?
- Finally, how to involve high switching frequency electronic devices using limited memory space of FPGA based real-time EMT simulation?

## **1.3 Aim, Objectives, Contributions and Outline of The Thesis**

### **1.3.1 Aim**

The aim of this thesis is to optimize FPGA-BASED EMT simulation, especially improving numerical accuracy, calculation speed and programming efficiency. Single-phase grounding fault and three-phase grounding fault is modelled to observe transient behaviours on transmission systems.

### **1.3.2 Objectives**

Based on the research gaps identified in the literature review, the main objectives of this thesis are to:

- Develop power system mathematical models of linear and non-linear components of different precisions on FPGA based hardware platform.
- Evaluate the effect of different precisions on numerical accuracy of FPGA based real-time simulation.
- Investigate methods to determine the initial steady-state operation point.
- Investigate intelligent toolbox or interface to simplify hardware programming for FPGA based real-time platform.
- Design high-frequency power electronic devices on FPGA based real-time platform.

### **1.3.3 Contributions**

The developed technologies support the application for different FPGA-based studies. Four principal contributions are involved:



- First, Single-Precision and Double-Precision are completely explored for linear and non-linear components for FPGA-based EMTs. The significant phase shift of Single-Precision on rotating process is discovered and analysed. In order to involve merits of Double-Precision and Single-Precision, adaptive Mixed-Precision algorithm is proposed for the first time. Apart from existing Single-Precision algorithm, Double-Precision and Mixed-Precision algorithms become available for users.
- Second, four initialization methods are proposed to allow users to define initial stable operation point for the first time. Time-increasing non-initialized error is no longer reducing global accuracy of FPGA-based EMT. For performance estimation, straightforward code and graphical schematic are demonstrated. With best timing constraint and least resource utilization, Method 4 using COE file becomes optimal for large system initialization.
- Third, a general MATLAB-to-FPGA (MTF) toolbox is proposed to simplify hardware programming. Training and programming efforts for hardware is significantly reduced for beginners. Overall development framework provides initial guidelines for translation and integration. Following specific translation rules, EMT component can be integrated as high-level main controller, low-level memory unit and low-level submodule. MTF toolbox becomes the user-friendly bridge to allow users generate repetitive hardware description in MATLAB environment.
- Fourth, the high-frequency switching of power electronic devices are modelled for FPGA. Frequent topology changes are no longer a barrier for implementing HVDC-MMC on hardware platform. Optimized strategies, such as interpolation and shift memory, simplify hardware design for power electronic devices and control system.

The formulated aggregate model makes it possible to simulate multi-level HVDC MMC, such as 36-level HVDC MMC, on a single FPGA board.

### **1.3.4 Thesis Outline**

The outline of the chapters is as follows:

- In Chapter 2, mathematical models of transmission line, RLC branch and synchronous machine are introduced in detail. Corresponding EMT model and hardware implementation structure are presented for different components. Considering numerical difference, the comparison of Double-Precision and Single-Precision Floating-Point is described by linear and non-linear conversion. Proposed single-Precision, Double-Precision and Mixed Precision are both introduced for non-linear and linear case studies.
- In Chapter 3, four initialization methods: Method 1 using physical interface, Method 2 using signal declaration, Method 3 using signal assignment and Method 4 using COE file are proposed for solving large EMTs based on FPGA. At first, detailed VHDL programming codes and implementation resources are provided for four methods. As a result of that, in-depth schematic structure and time schedules are analysed in direct graphical comparison. Theoretical analysis motivates to build up hardware implementation structure and hardware-in-loop algorithm for Method 4. In case studies, both component-level initialization and system-level initialization is provided to prove the magnificent strength of Method 4.
- In Chapter 4, a MATLAB-to-FPGA toolbox is developed to enhance the programming environment and help beginners to develop fast familiarity with FPGA-based EMT. At

first, a MATLAB-based input environment is introduced with different parameters for different components. Then the numerical conversion between MATLAB and FPGA is developed using flexible built-in M-functions. To optimize data processing in FPGA, straightforward translation for pipelined and unpipelined calculation procedure is presented. Finally, different system-level simulation results have proved the effectiveness of proposed Toolbox in practical EMT.

- In Chapter 5, electronic devices are built up in FPGA in order to solve high switching frequency in real-time FPGA. In the beginning, semiconductors, including Diode, IGBT and MMC are introduced and derived in mathematic models. Moreover, the fundamental blocks for control system are presented, such as PWM control and current control loop. To make hardware design more reliable, optimized strategies are developed, such as shift memory and interpolation for real-time implementation. In simulation analysis, PWM control block and HVDC-MMC system can operate with faster-than-real-time speed and high accuracy, verifying the model effectiveness.
- In Chapter 6, conclusions are drawn for previous chapters and possible applications are proposed for further work.

Chapter 2 develops a power system library for detailed components, which lays the foundation for Chapter 3 - 5.

Chapter 3 presents four initialization methods for FPGA based power system simulation. The initialization methods can be utilized to provide initial steady operating point for large power system simulations in Chapter 4 and 5.

Chapter 4 proposes a toolbox - a universal interface between FPGA and MATLAB. This toolbox can be applied to translate model in Chapter 2 and Chapter 3 from MATLAB to FPGA very efficiently.

Chapter 5 using the electric component library in Chapter 2 and initialization in Chapter 3 to build up power electronic device models in FPGA.

Chapter 6 summarizes the conclusion and presents potential future work.

# **CHAPTER 2 SINGLE-PRECISION, DOUBLE-PRECISION AND MIXED-PRECISION ALGORITHMS FOR FPGA-BASED EMT SOLUTION**

This chapter discusses the numerical integration methods, including Forward Euler, Backward Euler and Trapezoidal Rule in Section 2.1 first. Section 2.2 describes the mathematical modelling of components, including RLC branch, transmission line, transformers and synchronous machines. Section 2.3 discusses the numerical difference between Single-Precision and Double-Precision Floating-Point value in detailed comparison. In order to maintain both high accuracy and low memory, a novel Mixed-Precision algorithm is proposed for solving components with different complexity-level in Section 2.3. In Section 2.4, a number of case studies are carried out to compare the performance of Single-Precision and Double-Precision algorithms and to validate the effectiveness of proposed Mixed-Precision. Finally, Section 2.5 summarizes the major conclusions of this chapter.

## **2.1 Integration Methods**

EMT solutions require simulators to solve complex different components with high accuracy and fast speed. Computer and digital hardware are not able to solve differential equations in continuous period. To meet the simulation requirement, it requires that simulation time step needs to be extremely small to the amplitude of microsecond. Continuous states have to be

solved discretely from current time step to next time step. Therefore, efficient integration methods are required to create purely discrete models for different components. Most basic numerical methods for solving ordinary differential equations include Back Euler, Forward Euler, and Trapezoidal rules.

Back Euler is one of the most basic implicit methods. In this method, only history value at previous step is used to approximate future value for whole time range. Therefore, it is easily to grow error of first order. In order to integrate (2-1) from  $t$  to  $t - \Delta t$ , (2-2) using Back Euler method is given as follows:

$$f(t) = \int_{t-\Delta t}^t u(t)dt \quad (2-1)$$

$$f(t) = f(t - \Delta t) + \Delta t \cdot u(t - \Delta t) \quad (2-2)$$

where  $\Delta t$  is the time step,  $t$  is current time,  $u(t)$  is the input of the integrator,  $f(t)$  is the output of the integrator.

Forward Euler is another implicit method, which is similar to Backward Euler. But this method only uses future value to approach neighbourhood values. At every time step, a local truncation error is induced owing to the truncation of Taylor series. In (2-3), Forward Euler method is applied to discretize (2-1).

$$f(t) = f(t - \Delta t) + \Delta t \cdot u(t) \quad (2-3)$$

In order to prevent amplification errors of the iteration process, Trapezoidal rule is a second-order integral method, which has been proved numerically stable [51]. This means that the previous error of history values will not increase, it will provide damping of errors until coming

back to the stable condition. Therefore, it is widely suitable for high-frequency simulation.

Trapezoidal rule is given as follows:

$$f(t) = f(t - \Delta t) + \frac{\Delta t}{2}(u(t - \Delta t) + u(t)) \quad (2-4)$$

For a long-duration simulation, Trapezoidal Rule has been verified to show better numerical stability than Backward Euler and Forward Euler methods. Therefore, Trapezoidal rule is selected for the following research.

## **2.2 Power System Components Modelling**

### **2.2.1 RLC branch**

Consider the transient behaviours, most basic components only involve linear calculations, such as addition and multiplication, etc. Major linear components consisting of RLC branches are transmission lines and transformers without saturation.

Resistors are used to conduct current. They can represent circuit breaker resistances, grounding faults, lumped transmission lines and transformers in EMT. The relationship between node voltage and flowing current of single R branch is given as follows:

$$i_{km}(t) = \frac{1}{R}(v_k(t) - v_m(t)) \quad (2-5)$$

where  $R$  is the resistance,  $v_k(t)$  and  $v_m(t)$  is the node voltage of terminal  $k$  and terminal  $m$ , and  $i_{km}(t)$  is the current flowing from terminal  $k$  to terminal  $m$ .

Inductances are used to represent any sudden inductive changes in electrical circuits. It is widely representing the inductive part of load and voltage source impedance. Based on trapezoidal rule, an inductance can be replaced by a resistance of value  $\frac{\Delta t}{2L}$  and injected current source. If the history value of the branch is already known, the current at next time step can be generated recursively. Thus, differential relationship between node voltage and flowing current can be discretized as follows [45]:

$$i_{km}(t) = \frac{\Delta t}{2L}(v_k(t) - v_m(t)) + I_L(t - \Delta t) \quad (2-6)$$

$$I_L(t - \Delta t) = \frac{\Delta t}{2L}(v_k(t - \Delta t) - v_m(t - \Delta t)) + i_{km}(t - \Delta t) \quad (2-7)$$

where  $L$  is the impedance,  $i_{km}(t - \Delta t)$  is the current flowing from terminal  $k$  to terminal  $m$  at  $t - \Delta t$ ,  $I_L(t - \Delta t)$  is the equivalent history current source.

Capacitors are usually linked with electric fields to store energy. Capacitor voltage jump is instantly owing to consistent charging and discharging process. Usually, capacitive elements are used to represent snubber breakers and capacitive loads. The discrete equations can be derived similarly as follows [45]:

$$i_{km}(t) = \frac{2C}{\Delta t}((v_k(t) - v_m(t)) + I_C(t - \Delta t)) \quad (2-8)$$

$$I_C(t - \Delta t) = -\frac{2C}{\Delta t}(v_k(t - \Delta t) - v_m(t - \Delta t)) - i_{km}(t - \Delta t) \quad (2-9)$$

where  $C$  is the capacitance,  $I_C(t - \Delta t)$  is the equivalent history current source.

In spite of different connections, any RLC branch can be equivalent to a resistor in parallel with a current source as given in (2-10) - (2-11). Only constant parameters need to be modified



according to the type of a branch. Table 2.1 gives the exact parameter to calculate different current source for single RLC, series RL and series RC.

$$I_{RLC}(t - \Delta t) = J_{RLC}(v_k(t - \Delta t) - v_m(t - \Delta t)) + H_{RLC}i_{km}(t - \Delta t) \quad (2-10)$$

$$i_{RLC}(t - \Delta t) = G_{RLC}(v_k(t - \Delta t) - v_m(t - \Delta t)) + I_{RLC}(t - \Delta t) \quad (2-11)$$

where  $i_{RLC}(t)$  is the branch current,  $I_{RLC}(t - \Delta t)$  is the equivalent history current source for whole branch,  $J_{RLC}$  and  $H_{RLC}$  are constants,  $G_{RLC}$  is the branch equivalent conductance.

**Table 2.1 Summary of the comparison between different RLC branches**

	$G_{RLC}$	$H_{RLC}$	$J_{RLC}$
Single R	$\frac{1}{R}$	0	$\frac{1}{R}$
Single L	$\frac{\Delta t}{2L}$	1	$\frac{\Delta t}{2L}$
Single C	$\frac{2C}{\Delta t}$	-1	$-\frac{2C}{\Delta t}$
Series RL	$\frac{\frac{\Delta t}{2L}}{1 + \frac{\Delta t}{2L}R}$	$\frac{1 - \frac{\Delta t}{2L}R}{1 + \frac{\Delta t}{2L}R}$	$\frac{\frac{\Delta t}{2L}}{1 + \frac{\Delta t}{2L}R}$
Series RC	$\frac{\frac{2C}{\Delta t}}{1 + \frac{2C}{\Delta t}R}$	$\frac{-1 + \frac{2C}{\Delta t}R}{1 + \frac{2C}{\Delta t}R}$	$-\frac{\frac{2C}{\Delta t}}{1 + \frac{2C}{\Delta t}R}$

The real-time hardware implementation can be given by a 6-input Floating-Point accumulator.

The constants, including  $\frac{2C}{\Delta t}$ , 1 and -1, can be saved in ROMs without changes. The variables,

including  $I_C(t - \Delta t)$  and  $v_k(t)$  can be kept in RAMs. The RAM is a dual-port block, which

allows read and written simultaneously. Because each branch is independent of each other, and can be calculated separately. (2-10) and (2-11) can use the same 6-input block for saving resources. Therefore, pipeline strategy is employed for this 6-input computation block.

Although the loads might vary significantly from each other, the total load of a network changes rather slowly in a predictable manner. In addition, loads are usually linked with the terminals of transmission lines to separate from each other. In order to describe the characteristics of a large scale network over long distances under steady-state and transient conditions, it is necessary to establish the traveling wave model of transmission line components to explore the influence of R, L and C along the distributed parameter lines, which can not be considered in the lumped models.

In most transmission lines, the electric and magnetic fields point purely transverse to the direction of propagation [94]. Long transmission line can not be treated as a lumped element. In principle, series cascaded connection of a large number of lumped transmission line equivalent circuits will be close to the transmission line distributed model [95]. Owing to the distributed resistance, voltages along the axis of transmission line are totally different. This is the reason to involve the traveling wave theory to model the profiles of currents and voltages along the transmission line.

### **2.2.2 Transmission line**

The modelling of transmission line consists of distributed parameter model and lumped parameter model [45] - [47]. Distributed parameter model implements three-phase distributed line model with lumped losses based on traveling wave model [45]. The three-phase pi-section

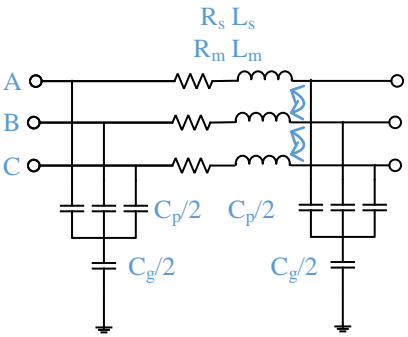
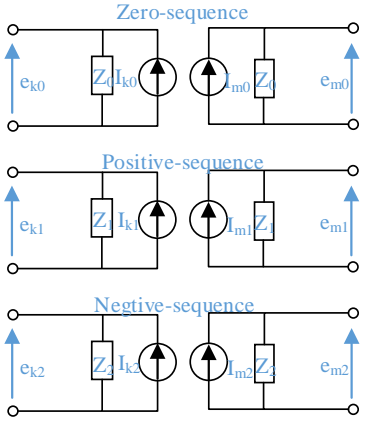
line model implements a balanced three-phase transmission line model with lumped parameters in a pi-section circuit [47]. For steady-solutions, transmission lines can be modelled with lumped nominal pi-section circuit accurately. For transient simulation, travelling wave model is faster and more accurate to describe more complicated dynamics. Table 2.2 details the comparison between lumped parameter model and distributed parameter model.

The travelling wave transmission line model is faster and more accurate than the nominal pi-circuit model.

Considering most transmission lines are longer than 15km, distributed parameter model is widely used for EMT simulations.

To simplify initial calculation, power loss is not involved for transmission lines at first stage. This means the resistance of the transmission line and leakage conductance of lossless transmission line are assumed to be zero. In order to prevent calculation errors from real ones, the assumption is that the operation frequency is high enough to ignore the resistance.

**Table 2.2 Comparison between lumped and distributed parameter model**

	Lumped model	Distributed parameter model
Distance	$\leq 100km$	$\geq 15km$
Solution	Steady solution	EMT solutions
Coupling relationship	Coupled in phase domain	Decoupled in modal domain
Equivalent circuit		
Parameters	<p><math>R_s</math> and <math>R_m</math> are self and mutual resistances, <math>L_s</math> and <math>L_m</math> are self and mutual inductances, <math>C_p</math> and <math>C_g</math> are phase and ground capacitances.</p>	<p><math>Z_0, Z_1, Z_2</math> is the zero-sequence, positive-sequence and negative-sequence surge impedance, <math>e_{k0}, e_{k1}, e_{k2}, e_{m0}, e_{m1}, e_{m2}</math> is the zero-sequence, positive-sequence and negative-sequence voltage at terminal k and terminal m, <math>I_{k0}, I_{k1}, I_{k2}, I_{m0}, I_{m1}, I_{m2}</math> is the corresponding history current source voltage at terminal k and m</p>

According to the traveling wave theory [45], the surge impedance and phase velocity can be calculated as:

$$Z_C = \sqrt{\frac{L'}{C'}} \quad (2-12)$$

$$v = \sqrt{\frac{1}{L'C'}} \quad (2-13)$$

where  $L'$  and  $C'$  is the inductance and capacitance of transmission line per unit length of the transmission line,  $Z_C$  is the surge impedance,  $v$  is the phase velocity.

In addition, the travel time from one terminal to the other terminal of the transmission line is as follows [45]:

$$\tau = \frac{d}{v} = d\sqrt{L'C'} \quad (2-14)$$

where  $\tau$  is the traveling time from one terminal to another, and  $d$  is the distance of the transmission line.

Based on trapezoidal rule, each terminal of the transmission line can also be equal to a resistance equal to  $Z$  in parallel with a history current source. The relationship between node voltages and currents at discrete time intervals can be derived as follows [45]:

$$i_k(t) = \frac{1}{Z}e_k(t) + I_k(t - \tau) \quad (2-15)$$

$$I_k(t - \tau) = -\frac{1}{Z}e_m(t - \tau) - i_m(t - \tau) \quad (2-16)$$

where  $e_k(t)$  and  $e_m(t)$  is the voltage of terminal  $k$  and  $m$  at time  $t$ ,  $I_k(t - \tau)$  is the history current source at terminal  $k$ ,  $i_k(t)$  is the current of terminal  $k$  at time  $t$ ,  $i_m(t - \tau)$  is the current of terminal  $m$  at time  $t - \tau$ .

On most occasions, a leakage conductance can not be ignored. This has led to involve distributed resistances when modelling a transmission line. After multiples of practical experiments, the method of lumped  $R/4$  in the terminals of transmission lines and  $R/2$  in the middle of transmission line shows great accuracy. Considering the distributed resistances, the surge impedance [45] can be further derived as:

$$Z = Z_c + \frac{R}{4} = \sqrt{\frac{L'}{C'}} + \frac{R}{4} \quad (2-17)$$

where  $Z$  is the surge resistance of transmission line,  $Z_c$  is the surge resistance of lossless transmission line,  $R$  is the equivalent lumped resistance.

After back substitution, the relationship of the voltages and currents of terminal  $k$  can be given as follows [45]:

$$i_k(t) = \frac{1}{Z} e_k(t) + I_k(t - \tau) \quad (2-18)$$

$$I_k(t - \tau) = -\frac{1-h}{2} \cdot \frac{1}{Z} \cdot e_k(t - \tau) - \frac{1-h}{2} \cdot h \cdot i_k(t - \tau)$$

$$-\frac{1+h}{2} \cdot \frac{1}{Z} \cdot e_m(t - \tau) - \frac{1+h}{2} \cdot h \cdot i_m(t - \tau) \quad (2-19)$$

$$h = \frac{Z_c - \frac{R}{4}}{Z_c + \frac{R}{4}} \quad (2-20)$$

where  $h$  is a constant as defined in (2-20).

For a multiphase transmission line, every phase of the transmission line is coupled with others mutually. Unlike the three-phase RLC branch, the off-diagonal elements of the multi-phase transmission line conductance matrix are non-zero. Fortunately, this problem could be solved by converting from phase domain to modal domain. For three phases, ABC phases are defined, which can be converted into zero sequence, positive sequence, and negative sequence. In addition, the three phase-modal transformation matrix [45] is given as follows:

$$T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 - M & 1 \\ 1 & 1 & 1 - M \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \quad (2-21)$$

where  $T$  is the phase-to-modal transformation matrix,  $M$  is the number of phases,  $M = 3$ .

Using conversion matrix, multi-phase currents and voltages can be decoupled separately as follows:

$$i_{phase}(t) = T i_{modal}(t) \quad (2-22)$$

$$e_{phase}(t) = T e_{modal}(t) \quad (2-23)$$

where  $i_{phase}(t)$  and  $i_{modal}(t)$  are the currents in phase domain and modal domain,  $e_{phase}(t)$  and  $e_{modal}(t)$  are the voltages in phase domain and modal domain.

Similarly, it is directly to get resistance transformation [45] as following:

$$Z_{phase} = T \cdot Z_{modal} \cdot inv(T) \quad (2-24)$$

where  $inv(T)$  is the modal-phase transformation matrix,  $Z_{phase}$  and  $Z_{modal}$  are the impedance matrix in phase domain and modal domain.

For hardware connection, currents and node voltages can be kept and updated consistently in RAM. The rest parameters, such as  $h$  and  $T$ , are unchanged in ROM. Considering maximum dimension and sequence, similar calculations can be both be implemented using 8-input multiply-add structure. This pipelined structure is able to handle hundreds of transmission lines with a clock cycle.

### 2.2.3 Transformer

Using three single-phase transformers, a three-phase transformer is built up. Complex internal magnetic field is not considered for the transformer, therefore saturation characteristic is not involved in this section. The major connections of two windings transformer include,  $Y$ , Grounded  $Y$ , Delta ( $DI$ ) and Delta ( $DII$ ). If reference  $Y$  voltage phasor is considered to be at 12 am on a clock, then  $DI$  and  $DII$  is at 1 pm and 11 am respectively. This is to say, they lag and lead  $Y$  by 30 degrees correspondingly.

If primary winding connection is  $Y$  connection, secondary winding connection is  $Y$ . Then this transformer is  $Y/Y$  connection. The  $Y/Y$  transformer can be modelled as follows:

$$v_{primary}(t) = n_{YY}v_{second}(t) \quad (2-25)$$

$$i_{primary}(t) = \frac{v_{second}(t)}{n_{YY}} \quad (2-26)$$

where  $v_{primary}(t)$  and  $v_{second}(t)$  are the primary and secondary phase voltages,  $i_{primary}(t)$  and  $i_{second}(t)$  are the primary and secondary phase currents,  $n_{YY}$  is the turn ratio of  $YY$  connection.



Considering the primary winding and secondary winding are series resistances and inductances, the resulting in Table 2.1 can be utilized. If the leakage of the magnetic winding is infinite, the  $Y/Y$  connection transformer can also be simplified as a two-terminal Norton circuit as follows:

$$I_{YY}(t - \Delta t) = G_{YY}(v_p(t - \Delta t) - n_{YY}v_s(t - \Delta t)) + H_{YY}i_{YY}(t - \Delta t) \quad (2-27)$$

$$i_{YY}(t - \Delta t) = G_{YY}(v_p(t - \Delta t) - n_{YY}v_s(t - \Delta t)) + I_{YY}(t - \Delta t) \quad (2-28)$$

$$G_{YY} = \frac{G_{YY1} \cdot G_{YY2}}{G_{YY2} + G_{YY1} \cdot n_{YY}^2} \quad (2-29)$$

$$H_{YY} = \frac{H_{YY1} \cdot G_{YY2} + H_{YY2} \cdot G_{YY1} \cdot n_{YY}^2}{G_{YY2} + G_{YY1} \cdot n_{YY}^2} \quad (2-30)$$

where  $v_p(t)$  and  $v_s(t)$  are the primary terminal and secondary terminal voltages,  $i_{YY}(t - \Delta t)$  is the primary current,  $G_{YY1}$  and  $G_{YY2}$  is the conductance of the primary and secondary series RL,  $H_{YY1}$  and  $H_{YY2}$  are the constants of the primary and secondary series RL,  $I_{YY}(t - \Delta t)$  is the history current source,

For node voltage solution, the admittance matrix and current source vector can be derived for  $Y/Y$  transformer as follows:

$$Y_M^{YY} = \begin{bmatrix} Y_{11}^{YY} & Y_{12}^{YY} \\ Y_{21}^{YY} & Y_{22}^{YY} \end{bmatrix} \quad (2-31)$$

$$Y_{12}^{YY} = Y_{21}^{YY} = -n_{YY} \cdot G_{YY} \quad (2-32)$$

$$Y_{11}^{YY} = G_{YY1} - \frac{G_{YY1}^2 \cdot n_{YY}^2}{G_{YY2} + G_{YY1} \cdot n_{YY}^2} = \frac{G_{YY1} \cdot G_{YY2}}{G_{YY2} + G_{YY1} \cdot n_{YY}^2} = G_{YY} \quad (2-33)$$

$$Y_{22}^{YY} = G_{YY2} - \frac{G_{YY2}^2}{G_{YY2} + G_{YY1} \cdot n_{YY}^2} = \frac{G_{YY1} \cdot G_{YY2} \cdot n_{YY}^2}{G_{YY2} + G_{YY1} \cdot n_{YY}^2} = n_{YY}^2 \cdot G_{YY} \quad (2-34)$$

$$I_V^{YY} = \begin{bmatrix} -I_{YY}(t - \Delta t) \\ n_{YY} \cdot I_{YY}(t - \Delta t) \end{bmatrix} \quad (2-35)$$

where  $Y_M^{YY}$  is the symmetrical admittance matrix,  $Y_{12}^{YY}$  and  $Y_{21}^{YY}$  are mutual conductance,  $Y_{11}^{YY}$  is the primary self-conductance,  $Y_{22}^{YY}$  is the secondary self-conductance,  $I_V^{YY}$  is the current vector.

## 2.2.4 Synchronous Machines

Synchronous machines play an essential role in supplying electric energy and controlling stable terminal voltage. It is always challenging to synchronize all the synchronous machines in the same large-scale network. Therefore, it is significant to model the dynamic properties and behaviours of synchronous machines.

Based on the electric machine theory, synchronous machines consist of magnetic rotors and stators. Take synchronous machine as an example where the armature windings are on the rotor and field windings are set on the stator. Synchronous machines can be classified into salient-pole and round-round pole type. For generic modelling, salient-pole is selected for modelling.

In this part, the mathematical model of a universal machine is built up to represent synchronous machines, induction machines and dc machines. Using compensation method, the rest network, without synchronous machines, can be regarded as a voltage source connected with a resistance. Similar to electrical part, the mechanical torque is treated as current source instead of mass-shaft model.

The core of transient solutions for synchronous machines is to form a whole matrix for integrating the differential equations of the field and armature windings together. Because of the coupling relationship between the electrical part and mechanical part, they can not be calculated simultaneously. To solve this calculation sequence problem, prediction methods become essential and are applied. Considering high-frequency magnetic and electrical changes, the prediction of the mechanical elements is selected to avoid unexpected oscillations.

The history values including currents, voltages and flux are assumed to be known at previous step. To improve global accuracy, the calculation of synchronous machine should follow these steps:

(a) Predict the mechanical and electrical angular speed according to the prediction method based on linearization. Owing to the mechanical angular speed change slowly, linearization can be applied successfully.

$$w_{m\_pre}(t) = 2w_m(t - \Delta t) - w_m(t - 2\Delta t) \quad (2-36)$$

where  $w_m(t - \Delta t)$  and  $w_m(t - 2\Delta t)$  are the real mechanical angular speed at time  $t - \Delta t$  and  $t - 2\Delta t$ ,  $w_{m\_pre}(t)$  is the predicted angular speed after linearization.

(b) According to trapezoidal rule, it is direct to predict mechanical and electrical angle based on the previous predicted angular speed by applying trapezoidal rule [47].

$$\beta_m(t) = \beta_m(t - \Delta t) + \frac{\Delta t}{2}(w_{m\_pre}(t) + w_m(t - \Delta t)) \quad (2-37)$$

$$\beta_e(t) = \beta_m(t) \cdot PolePairs \quad (2-38)$$

where  $\beta_m(t - \Delta t)$  and  $\beta_m(t)$  are the mechanical rotor angle at time  $t - \Delta t$  and  $t - 2\Delta t$ ,  $\beta_e(t)$  is the electric rotor angle at time  $t$ , and  $PolePairs$  is the pole pairs.

(c) Calculate the park transformation matrix [47] according to the updated electrical angle, which can reflect the  $a$ - $b$ - $c$  phase to rotating  $dq0$  axis for decoupling three phases. This has reduced the calculation sophistication because the off-diagonal elements of the inductance matrix are all zero.

$$Park = \frac{2}{3} \cdot \begin{bmatrix} \sin(\beta_e(t)) & \sin(\beta_e(t) - \frac{2}{3}\pi) & \sin(\beta_e(t) + \frac{2}{3}\pi) \\ \cos(\beta_e(t)) & \cos(\beta_e(t) - \frac{2}{3}\pi) & \cos(\beta_e(t) + \frac{2}{3}\pi) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2-39)$$

where  $Park$  is the Park transformation matrix.

(d) Solve the Thevenin equivalent circuit of external network according to the compensation method. Firstly, calculate the open-circuited terminal voltage by assuming synchronous machine is open-circuited. After generating open-circuit voltage and equivalent resistance, the terminal voltages can be converted into  $dq0$  axis as (2-40) – (2-48) [96]. These equations are all calculated in per unit value.

$$v_{abc}(t) = v_{abc\_0}(t) + R_{external\_abc} i_{abc}(t) \quad (2-40)$$

$$v_{dq0}(t) = v_{dq0\_0}(t) + R_{external\_dq0} i_{dq0}(t) \quad (2-41)$$

$$v_{dq0}(t) = Park \cdot v_{abc}(t) \quad (2-42)$$

$$R_{external\_dq0} = Park \cdot R_{external\_abc} \cdot (Park)^{-1} \quad (2-43)$$

$$v_{abc}(t) = [v_a(t) \ v_b(t) \ v_c(t) \ v_f(t) \ 0 \ 0 \ 0]^T \quad (2-44)$$

$$v_{dq0}(t) = [v_d(t) v_q(t) v_0(t) v_f(t) 0 0]^T \quad (2-45)$$

$$i_{abc}(t) = [i_a(t) i_b(t) i_c(t) i_f(t) i_{kd}(t) i_{kq1}(t) i_{kq2}(t)]^T \quad (2-46)$$

$$i_{dq0}(t) = [i_a(t) i_q(t) i_0(t) i_f(t) i_{kd}(t) i_{kq1}(t) i_{kq2}(t)]^T \quad (2-47)$$

$$R_{abc} = \text{diag}[R_a R_a R_a 0 0 0] \quad (2-48)$$

where  $v_{abc}(t)$  and  $v_{dq0}(t)$  are vectors of voltages in the phase domain and modal domain, respectively,  $v_{abc\_0}(t)$  and  $v_{dq0\_0}(t)$  are vectors of the equivalent voltage sources in the phase domain and modal domain, respectively,  $i_{abc}(t)$  and  $i_{dq0}(t)$  are vectors of currents in the phase domain and modal domain, respectively,  $R_{external\_abc}$  and  $R_{external\_dq0}$  are the resistance matrices in the phase domain and modal domain, respectively,  $(Park)^{-1}$  is the inverse Park transformation matrix,  $i_a(t), i_b(t),$  and  $i_c(t)$  are stator currents in A,B C phase domain,  $i_d(t), i_q(t),$  and  $i_0(t)$  are stator currents in modal domain,  $i_f(t)$  is the field current,  $i_{kd}(t), i_{kq1}(t),$  and  $i_{kq2}(t)$  are the damping currents in d axis and q axis,  $v_a(t) v_b(t)$  and  $v_c(t)$  are stator voltages in A,B C phase domain,  $v_d(t) v_q(t)$  and  $v_0(t)$  are stator voltages in modal domain,  $V_f$  is the field voltage.

(e) Calculate the currents total network with synchronous machine representation of armature and field windings. Based on trapezoidal rule, the voltage-current-flux relationship among stator winding, field winding and damper winding can be described as follows [47]:

$$\int_{t-\Delta t}^t v_{dq0}(t) dt = \int_{t-\Delta t}^t \frac{d\varphi(t)_{dq0}}{w_{base} dt} + \int_{t-\Delta t}^t w \varphi(t)_{dq0} dt + \int_{t-\Delta t}^t i(t)_{dq0} R_{SG} t \quad (2-49)$$

$$R_{SG} = \text{diag}[-R_a -R_a -R_a R_{fd} R_{kd} R_{kq1} R_{kq2}] \quad (2-50)$$

$$w = \begin{bmatrix} 0 & -w_e(t) & \text{zeros}(1,5) \\ w_e(t) & 0 & \text{zeros}(1,5) \\ \text{zeros}(5,1) & \text{zeros}(5,1) & \text{zeros}(5,5) \end{bmatrix} \quad (2-51)$$

$L_{SG} =$

$$\begin{bmatrix} -(L_f + L_{md}) & 0 & 0 & L_{md} & L_{md} & 0 & 0 \\ 0 & -(L_f + L_{mq}) & 0 & 0 & 0 & L_{mq} & L_{mq} \\ 0 & 0 & L_0 & 0 & 0 & 0 & 0 \\ -L_{md} & 0 & 0 & (L_l + L_{lfd}) & L_{md} & 0 & 0 \\ -L_{md} & 0 & 0 & L_{md} & (L_l + L_{lkd}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (L_l + L_{lkq1}) & L_{mq} \\ 0 & 0 & 0 & 0 & 0 & L_{mq} & (L_l + L_{lkq2}) \end{bmatrix}$$

(2-52)

where  $R_a$  is the stator resistance per phase,  $R_{fd}$  is the field resistance,  $R_{kd}$  is the damper resistance in d axis,  $R_{kq1}$  and  $R_{kq2}$  are the damper winding resistances in q axis,  $w_e(t)$  is the electric angular speed,  $R_a$  is the stator resistance,  $L_l$  is the leakage inductance,  $L_{md}$  is the direct-axis magnetizing inductance viewed from stator,  $L_{mq}$  is the quadrature-axis magnetizing inductance viewed from stator,  $L_{lfd}$  is the leakage inductance,  $L_{lkd}$  is the d-axis leakage inductance,  $L_{lkq1}$  and  $L_{lkq2}$  are the two q-axis leakage inductances.

By back substitution, the following equation can be obtained [96]:

$$v_{dq0\_0}(t) - v_{hist}(t - \Delta t) = (-R_{external\_dq0} + \frac{2}{w_{base}\Delta t} L_{SG} + w L_{SG} + R_{SM}) \cdot i_{dq0}(t) \quad (2-53)$$

where  $v_{hist}(t - \Delta t)$  is the equivalent voltage sources,  $w_{base}$  is base angular frequency.

By gauss elimination,  $i_{dq0}(t)$  can be solved directly.

(f) Update fluxes and voltages in direct and quadrature axis [45]-[47]

$$v_{dq0}(t) = -i_{dq0}(t)R_{dq0} + \frac{2}{\Delta t \cdot w_{base}} L_{SM} i_{dq0}(t) + w L_{dq0} i_{dq0}(t) + v_{hist}(t) \quad (2-54)$$

$$\varphi_{dq0}(t) = L_{SG} i_{dq0}(t) \quad (2-55)$$

$$\varphi_{dq0}(t) = [\varphi_d(t) \ \varphi_q(t) \ \varphi_0(t) \ \varphi_f(t) \ \varphi_{kd}(t) \ \varphi_{kq1}(t) \ \varphi_{kq2}(t)]^T \quad (2-56)$$

where  $\varphi_{dq0}(t)$  is flux vector,  $\varphi_d(t)$ ,  $\varphi_q(t)$  and  $\varphi_0(t)$  is d-axis, q-axis and 0-axis stator flux linkage,  $\varphi_f(t)$  is field flux linkage,  $\varphi_{kd}(t)$  is d-axis damper flux linkage,  $\varphi_{kq1}(t)$  and  $\varphi_{kq2}(t)$  q-axis damper flux linkage.

(g) Calculate electrical torques by substituting previous currents and voltages [1].

$$T_e(t) = i_q(t)\varphi_d(t) - i_d(t)\varphi_q(t) \quad (2-57)$$

where  $T_e(t)$  is the electrical torque.

(h) Similar to RLC branch, solve the mechanical part of the synchronous machine [47].

$$T_e(t) = i_q(t)\varphi_d(t) - i_d(t)\varphi_q(t) \quad (2-58)$$

$$w_m(t) = \frac{\Delta t}{4H} (T_e(t) - T_m(t)) + w_{hist}(t) \quad (2-59)$$

$$w_{hist}(t) = w_m(t - \Delta t) + \frac{\Delta t}{4H} (T_m(t - \Delta t) - T_e(t - \Delta t)) \quad (2-60)$$

where  $T_m(t - \Delta t)$  and  $T_e(t - \Delta t)$  is the mechanical torque and electrical torque at time  $t - \Delta t$ ,  $w_{hist}(t)$  is the history angular speed source,  $w_m(t - \Delta t)$  is the mechanical angular speed at time  $t - \Delta t$ ,  $H$  is inertia constant.

(i) Update the history values and get back to step (a) to find the solution at next time step until the total simulation time is finished.

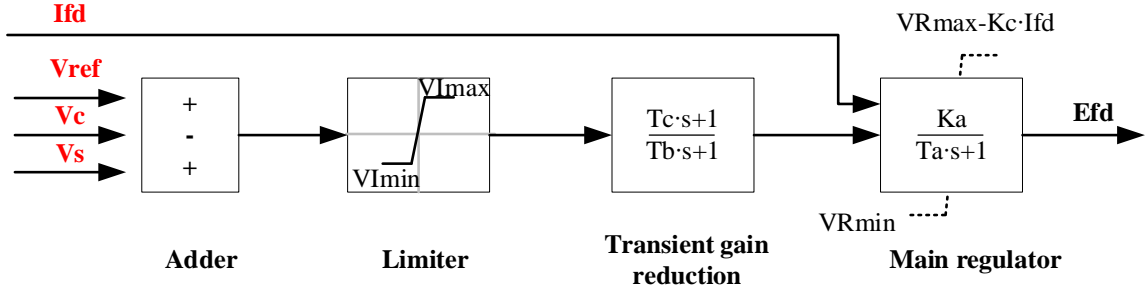
### **2.2.5 Excitation system**

Excitation system is to provide direct current to the field winding for a synchronous machine. Therefore, excitation system can control field voltage and field current, improving the control and protection functions of the synchronous machine. The control functions include voltage control, reactive power flow control, and power stability enhancement. The protective functions ensure the operation of synchronous machine, excitation system and other equipment will not exceed upper limit and lower limit.

After evolution, excitation system can be classified into three main types based on excitation resources, including AC excitation system, DC excitation system and static excitation system. AC excitation system is using alternator to be the main excitation generator power. DC excitation system is using the EC generator to provide DC current. Static excitation system provided excitation system through transformers and rectifiers.

To represent a wide variety of excitation systems, IEEE has standardized 12 models for excitation system currently in use, such as AC1A, DC1A, and AC4A. AC4A type exciter model is using a full-wave bridge circuit thyristor in an alternator-supplied controlled-rectifier excitation system. Fig. 2.1 shows the detail of AC4A system. Therefore, IEEE type AC4A excitation system is selected to provide direct current for the synchronous machine.





**Fig. 2.1 IEEE AC4A exciter system [1]**

To model complete AC4A system, mathematical models can be built up step by step using trapezoidal rule. Detailed calculation steps from step (a) to step (g) are given as follows:

(a) The output of initial adder is given as:

$$v_{e1}(t) = v_s(t) + v_{ref}(t) - v_c(t) \quad (2-61)$$

where  $v_s(t)$  is the stable voltage obtained from PSS,  $v_{ref}(t)$  is the referenced voltage,  $v_c(t)$  is the compensated voltage,  $Efd(0)$  is the initial excitation voltage,  $K_a$  is the overall gain,  $v_{e1}(t)$  is the output voltage at step (a).

(b) Limiter ensures that the output will not exceed upper bound and lower bound. The output of the limiter can be given as:

$$v_l(t) = \begin{cases} v_{Imax}, & \text{if } v_{e1} \geq v_{Imax} \\ v_{e1}(t), & \text{if } v_{Imin} < v_{e1}(t) < v_{Imax} \\ v_{Imin}, & \text{if } v_{e1} \leq v_{Imin} \end{cases} \quad (2-62)$$

where  $v_l(t)$  is the limiter output,  $v_{Imax}$  and  $v_{Imin}$  are the upper bound and lower bound, respectively.

(c) The transient gain reduction is implemented by a PID control block. Discretized calculation is given as follows:

$$v_{e2}(t) = \frac{\left(\frac{\Delta t}{2} + T_C\right)}{\left(\frac{\Delta t}{2} + T_B\right)} v_I(t) + hist\_v_{e2}(t - \Delta t) \quad (2-63)$$

$$w_{hist}(t) = w_m(t - \Delta t) + \frac{\Delta t}{4H} (T_m(t - \Delta t) - T_e(t - \Delta t)) \quad (2-64)$$

where  $T_B$  and  $T_C$  are the transient gain reduction lead and lag time constants, respectively,  $hist\_v_{e2}(t - \Delta t)$  is the equivalent history voltage source,  $v_{e2}(t)$  is the output of transient gain reduction.

(d) Main regulator is implemented by a single time constant with non-windup limits. With the non-windup limits, the output comes off the limits as soon as input re-enters the range within limits. The output and input of the non-windup limit blocks are influenced each other all the time. But the output can not be obtained in advance. Therefore, linear prediction is required to discretize the non-windup limit as follows:

$$E_{fd\_pre}(t) = 2E_{fd}(t - \Delta t) - E_{fd}(t - 2\Delta t) \quad (2-65)$$

where  $E_{fd\_pre}(t)$  is the predicted field voltage,  $E_{fd}(t - \Delta t)$  is the field voltage.

The relationship between the integrator output and non-windup limit output is given as [1]:

$$y(t) = \frac{K_a v_R(t) - E_{fd}(t)}{sT_a} \quad (2-66)$$

$$E_{fd}(t) = \begin{cases} v_{Rmin}, & y(t) \leq v_{Rmin} \\ y(t), & v_{Rmin} < y(t) < v_{Rmax} \\ v_{Rmax} - K_c i_{fd}(t), & v_{Rmax} \leq y(t) \end{cases} \quad (2-67)$$

where  $y(t)$  is the output of integrator,  $v_{Rmin}$  and  $v_{Rmax}$  are voltage regulator limits,  $i_{fd}(t)$  is the field current,  $K_c$  is the time constant.

Applying trapezoidal rule and linearization, calculations for the main regulator can be calculated as follows:

$$y(t) = \frac{K_a \Delta t}{2T_a} v_R(t) - \frac{\Delta t}{2T_a} E_{fd\_pre}(t) + hist(t - \Delta t) \quad (2-68)$$

$$hist(t - \Delta t) = \frac{K_a \Delta t}{2T_a} v_R(t - \Delta t) - \frac{\Delta t}{2T_a} E_{fd}(t - \Delta t) + y(t - \Delta t) \quad (2-69)$$

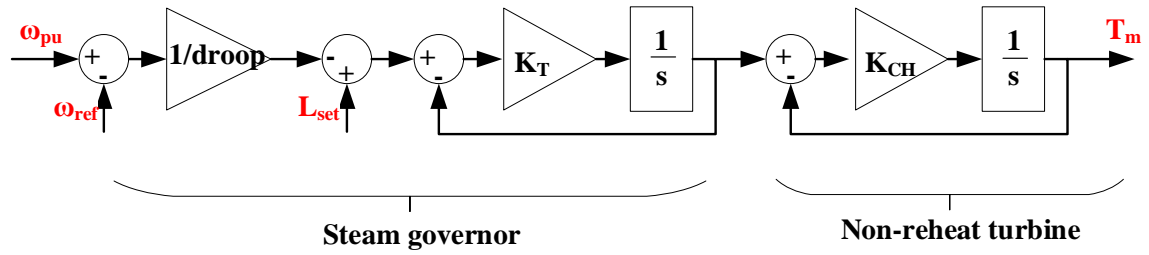
where,  $K_a$  and  $T_a$  are the gain and time constant for the main regulator.

## 2.2.6 Governor and prime mover system

The prime sources of electrical energy are normally derived from kinetic energy and thermal energy. Prime mover converts these sources of energy into mechanical energy, which is converted into electrical energy by a synchronous machine. Prime mover can provide controlling frequency and power, including low-frequency control and automatic generation control.

A variety of prime movers can be seen in power system networks, such as hydraulic turbine and steam turbine. Steam turbine, as one of the most common prime movers, converted high temperature and high pressure into rotating energy. Therefore, a steam turbine in Fig. 2.2 is selected to provide mechanical torque for synchronous machine [1].

Similar to the excitation system, the modelling of a governor could be solved correspondingly from step (a) to step (g).



**Fig. 2.2 Steam governor and non-reheat turbine [1]**

(a) The first step is to calculate the Delta speed:

$$w_1(t) = w_{pu}(t) - w_{ref} \quad (2-70)$$

where  $w_1(t)$  is delta speed,  $w_{ref}$  is referenced speed,  $w_{pu}(t)$  is the input per unit velocity.

(b) The gain of droop can be given by a multiplier:

$$w_2(t) = \frac{w_1(t)}{100droop} \quad (2-71)$$

where  $v_2(t)$  is droop speed,  $droop$  is percentage droop

(c) The delta load setpoint can be given by a subtracter:

$$w_3(t) = L_{set} - w_2(t) \quad (2-72)$$

where  $L_{set}$  is the constant load setpoint,  $w_3(t)$  is the delta load setpoint

(d) The prime mover output is implemented based on negative feedback and integrator.

Applying trapezoidal rule, delta y can be obtained using the following equations:

$$Deltay(t) = \frac{K_T \Delta t}{K_T \Delta t + 2} w_3(t) + Deltay_{hist}(t - \Delta t) \quad (2-73)$$

$$Deltay_{hist}(t - \Delta t) = \frac{K_T \Delta t}{K_T \Delta t + 2} w_3(t - \Delta t) - \frac{K_T \Delta t - 2}{K_T \Delta t + 2} Deltay(t - \Delta t) \quad (2-74)$$

$$K_T = \frac{1}{T_T} \quad (2-75)$$

where  $Deltay(t)$  is the output of prime mover,  $Deltay_{hist}(t - \Delta t)$  is the equivalent history current source,  $K_T$  is the time gain,  $T_T$  is the time constant of governor

(e) Similar to step (d), the modelling of a non-reheat turbine can be established as follows:

$$T_m(t) = \frac{K_{CH}\Delta t}{K_{CH}\Delta t + 2} Deltay(t) + T_{m\_hist}(t) \quad (2-76)$$

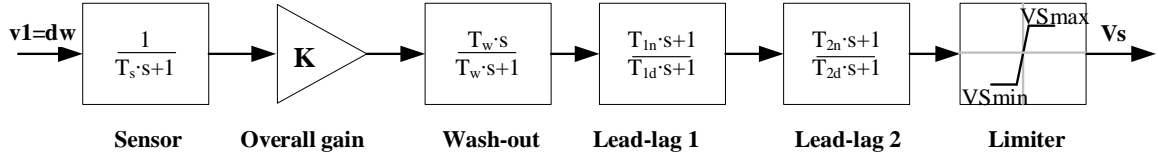
$$T_{m\_hist}(t) = \frac{K_{CH}\Delta t}{K_{CH}\Delta t + 2} Deltay(t - \Delta t) - \frac{K_{CH}\Delta t - 2}{K_{CH}\Delta t + 2} T_m(t - \Delta t) \quad (2-77)$$

$$K_{CH} = \frac{1}{T_{CH}} \quad (2-78)$$

where  $T_m(t)$  is the mechanical torque,  $T_{m\_hist}(t)$  is the equivalent history current source,  $K_{CH}$  is the gain for non-reheat turbine,  $T_{CH}$  is the time constant of main inlet volumes and steam chest.

### 2.2.7 Power system stabilizer

The function of a power system stabilizer (PSS) is to add damping to the rotor oscillations. This is by controlling its excitation using auxiliary stabilizing signal. For the purpose of introducing damping torque component, the deviation of the generator speed is used to control the excitation system for the generator. A generic PSS model [1] is given in Fig. 2.3. The complete modelling can be obtained by following calculation steps using trapezoidal rule:



**Fig. 2.3 Power system stabilizer**

(a) A sensor is implemented by low pass filter block. The relationship between the output and input can be discretized as follows:

$$v_2(t) = \frac{\frac{\Delta t}{2}}{\frac{\Delta t}{2} + T_S} v_1(t) + hist(t - \Delta t) \quad (2-79)$$

$$hist(t - \Delta t) = \frac{\frac{\Delta t}{2}}{\frac{\Delta t}{2} + T_S} v_1(t - \Delta t) - \frac{\frac{\Delta t}{2} - T_S}{\frac{\Delta t}{2} + T_S} v_2(t - \Delta t) \quad (2-80)$$

where  $v_1(t)$  is the deviation generator speed,  $v_2(t)$  is the sensor output,  $T_S$  is the time constant.

(b) The gain block can be obtained by a multiplier.

$$v_3(t) = K v_2(t) \quad (2-81)$$

where  $K$  is the overall gain,  $v_3(t)$  is the gain block output.

(c) The wash-out block is a high pass filter, which can be obtained as follows:

$$v_4(t) = \frac{T_w}{\frac{\Delta t}{2} + T_w} v_3(t) + v_{4\_hist}(t - \Delta t) \quad (2-82)$$

$$v_{4\_hist}(t - \Delta t) = \frac{\frac{\Delta t}{2} + T_w}{\frac{\Delta t}{2} + T_w} v_4(t - \Delta t) - \frac{T_w}{\frac{\Delta t}{2} + T_w} v_3(t - \Delta t) \quad (2-83)$$

where  $T_w$  is the time constraint for wash-out,  $v_4(t)$  is the wash-out output,  $v_{4\_hist}(t - \Delta t)$  is the equivalent history source.

(d) The phase compensation consists of two lead-lag blocks. The first block can be discretized as:

$$hist(t - \Delta t) = \frac{\left(\frac{\Delta t}{2} - T_{1n}\right)}{\left(\frac{\Delta t}{2} + T_{1d}\right)} v_4(t - \Delta t) - \frac{\left(\frac{\Delta t}{2} - T_{1d}\right)}{\left(\frac{\Delta t}{2} + T_{1d}\right)} v_5(t - \Delta t) \quad (2-84)$$

$$v_5(t) = \frac{\left(\frac{\Delta t}{2} + T_{1n}\right)}{\left(\frac{\Delta t}{2} + T_{1d}\right)} v_4(t) + hist(t - \Delta t) \quad (2-85)$$

where  $T_{1n}$  and  $T_{1d}$  is the time constant for lead-lag block,  $v_5(t)$  is the lead-lag output,  $v_{5\_hist}(t - \Delta t)$  is the equivalent history source.

(e) The final limiter can keep the PSS output within limits. The limiter calculation is given as:

$$v_s(t) = \begin{cases} v_{Smax}, & \text{if } v_6(t) \geq v_{Smax} \\ v_6, & \text{if } v_{Smin} < v_6(t) < v_{Smax} \\ v_{Smin}, & \text{if } v_6(t) \leq v_{Smin} \end{cases} \quad (2-86)$$

where  $v_{lmax}$  and  $v_{lmin}$  are the limits,  $v_s(t)$  is the stable voltage output,  $v_6(t)$  is the phase compensation block output.

## 2.2.8 Circuit breaker

The circuit breaker is modelled as a time-controlled resistor [47]. When the circuit breaker is closed, it is represented by the closed resistance  $R_{off}$ . When the circuit breaker is open, it is represented by the open resistance  $R_{on}$ . The calculation of circuit breaker is given as follows:

$$R_{CB}(t) = \begin{cases} R_{on}, & \text{if } t_{on}^{start} \leq t \leq t_{on}^{end} \\ R_{off}, & \text{else} \end{cases} \quad (2-87)$$

where  $R_{CB}(t)$  is the circuit breaker resistance at time  $t$ ,  $t_{on}^{start}$  and  $t_{on}^{end}$  are the closing start time and closing end time respectively.



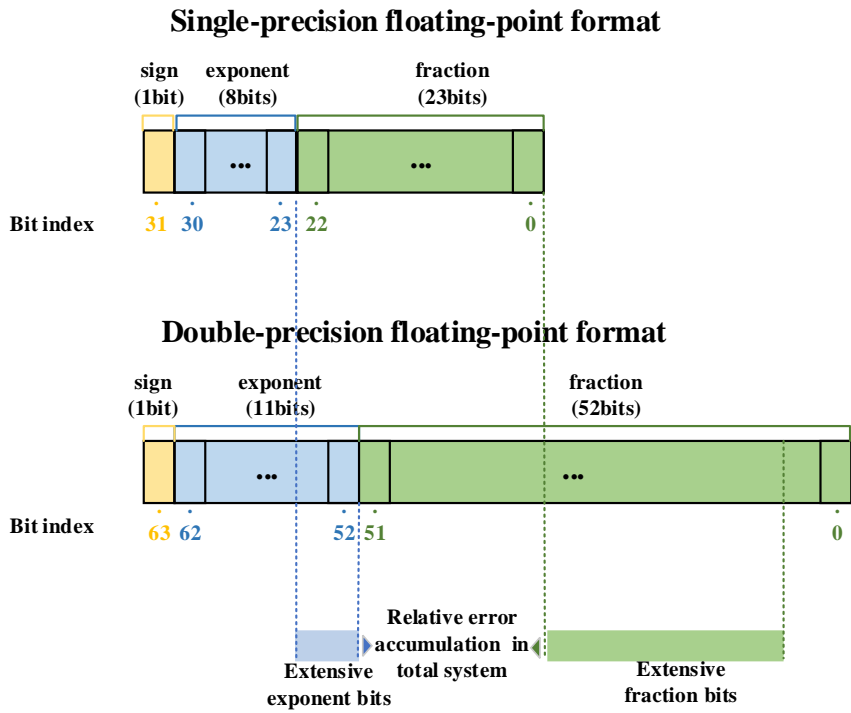
## **2.3 Single-Precision, Double-Precision and Mixed-Precision**

### **Algorithm**

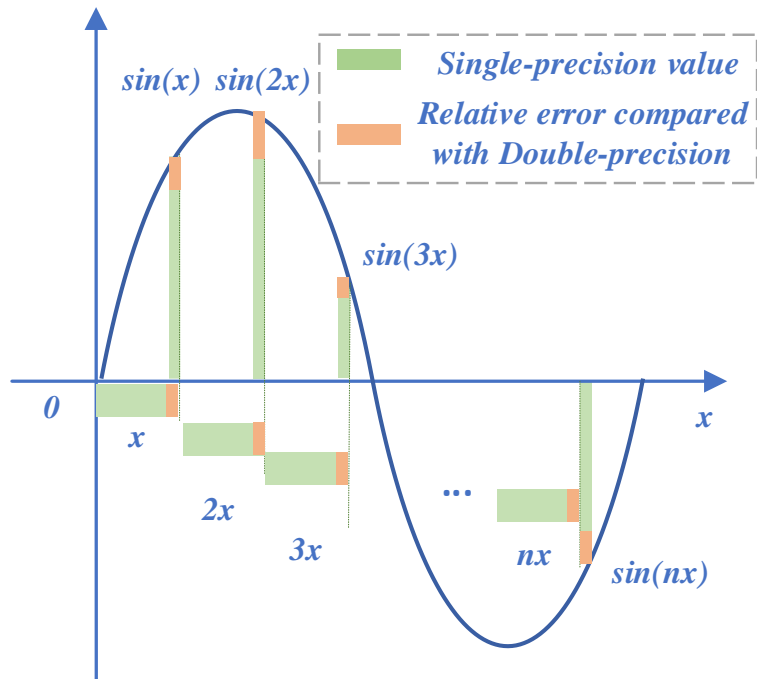
#### **2.3.1 Single-Precision and Double-Precision Comparison**

To achieve high accuracy, major off-line software, such as MATLAB, uses Double-Precision Floating-Point value to implement multiples calculations. The main challenge for real-time hardware is to achieve all calculations in hardware-in-loop structure within a small time step. Therefore, the first choice for FPGA is usually Single-Precision Floating-Point value to finish fast calculations on most occasions. However, it is possible that Single-Precision can reduce accuracy on complex conversion. It is necessary to analyse and assess the difference between Single-Precision and Double-Precision Floating-Point value in detail. Fig. 2.4 gives the comparison between single-precision floating format and double-precision format.

Seen from Fig. 2.4, Single-Precision format allocates 1 bit for sign, 8 bits for exponent and 23 bits for fraction. By contrast, Double-Precision includes 1 bit for sign, 11 bits for exponent and 23 bits for fraction. When storing the same value in Single-Precision for memory, more mantissa will be abandoned than Double-Precision. This is for the purpose of using half memory space, which comes along with slightly accuracy reduction. This relative error is possible to accumulate after repetitive and complex calculation times.



**Fig. 2.4 Double-precision and Single-precision format comparison**



**Fig. 2.5 Relative error accumulation in sine wave function between double-precision and single-precision format**

In order to analyse this accumulation error more explicit, a fixed step addition with increment  $x$ , sine function  $\sin(nx)$  and cos function  $\cos(nx)$  is involved.  $x$  is a constant, equal to 0.1884. For graphical comparison, Fig. 2.5 illustrates the relative error of sine-wave function and fixed step addition. It can be observed that, the relative error between Single-Precision and Double-Precision can vary from the point position. The relative error at different input level is oscillated at various numerical range and sign. Especially in large-scale control system, the influence is unpredictable. This makes it more difficult to insert correction feedback section especially for long-duration simulation. This relative error of single value is uncertain and can not be eliminated by prediction. In particular, the potential influence in the phase angle will be more apparent for a variable-step sine wave function.

Apart from single relative error, it is essential to exploit this error changes after multiples of accumulation. Table 2.3 gives the relative error for a fixed step addition  $nx$ , sine function  $\sin(nx)$  and cos function  $\cos(nx)$ , where iteration time  $n$  vary from 1 to 100000. Seen from Table 2.3, the relative errors are quite small in the initial steps until 1000 steps. This can demonstrate single-precision format is available to be the main format choice for most scenarios in short-duration simulation. However, when it comes to 10000 steps, there is a clear error in adder output, sin wave function output and cos wave function as well. This is because some essential bits are released for limited 32 bits bandwidth as simulation time goes on. And the phase shift problem is clearer and more apparent at step 100000. With time step of 50 microsecond, this error can damage amplitude and phase shift for long-duration calculation to the amplitude of second.

**Table 2.3 Summary of the comparison between addition, sine function and cos function**

$n$	1	100	1000	10000	100000
$nx$	-5.22E-10	2.44E-06	9.59E-06	-0.0082	-1.9243
$\sin(nx)$	-5.21E-10	-7.53E-07	9.59E-06	-0.0082	-0.9382
$\cos(nx)$	9.83E-12	-2.32E-06	-4.60E-11	-3.34E-05	-1.3462

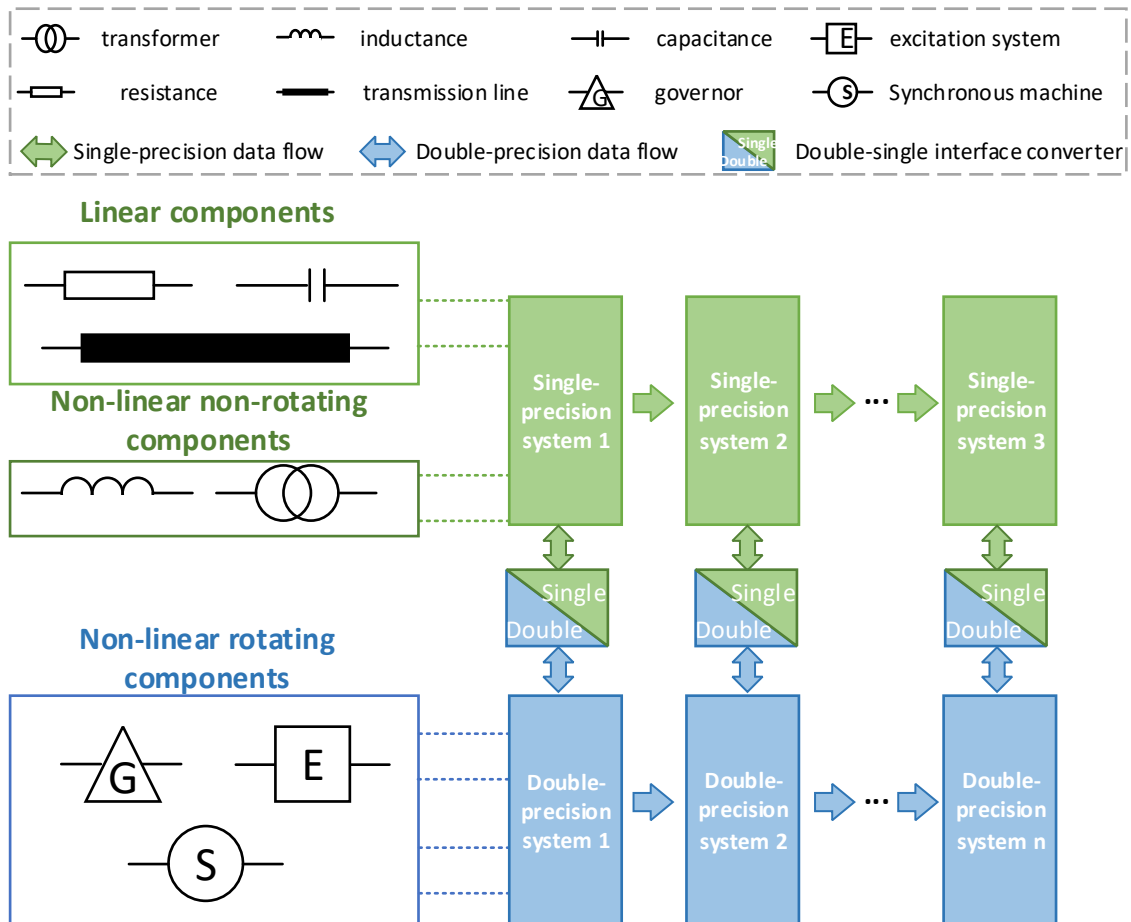
Therefore, how to make the accuracy benefits double-precision formats to insert in a limited memory ranges consisting of single-precision format is a main problem to be addressed. In this paper, a Mixed-precision calculation structure is proposed to combine various types of power system components in a limited memory space with high-performance speed.

### **2.3.2 Mixed-Precision Algorithm**

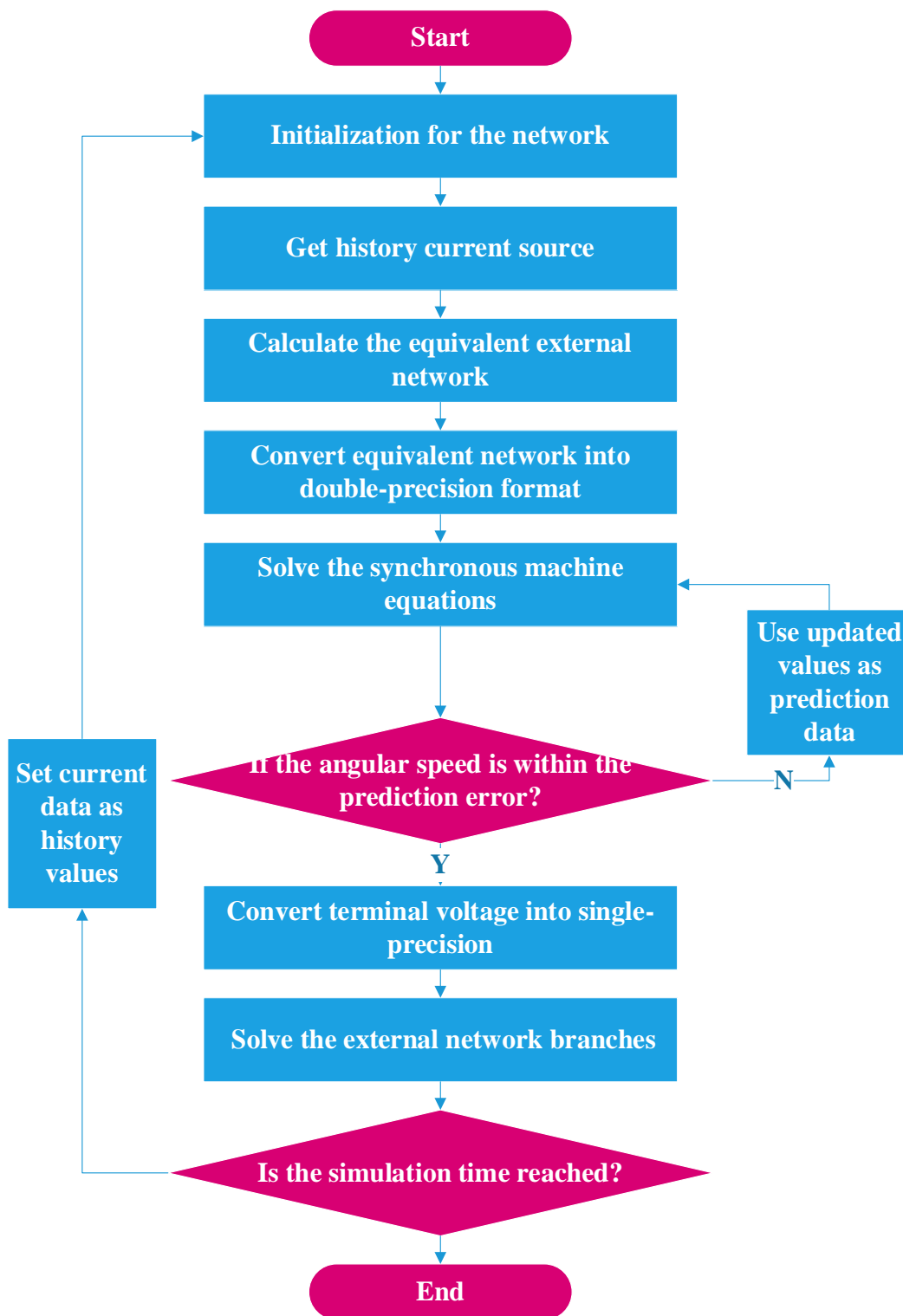
To maintain the advantages of Single-Precision and Double-Precision, Fig. 2.6 illustrates the details large-scale electromagnetic transient system of hybrid single-precision and double-precision in FPGA. In addition, double-precision format is selected for the synchronous machine, excitation system and prime mover system owing to Park's transformation. By contrast, single-precision format is applied for transformer, RLC branch and transmission line owing to their linearity.

And double-to-single and single-to-double block has linked the bridge between linear and non-linear system. This interface can deal with bidirectional dataflow between different arithmetic operation systems reasonably. Thus, the overall simplified mixed-precision system is finally obtained.

The proposed structure has reduced the heavy burden of resource utilization for complete double-precision system as well as emerging mixed arithmetic technology. Therefore, the dynamic characteristics of non-linear components and linear components can be combined with each other in a simplified method even from different numerical systems.



**Fig. 2.6 Mixed-Precision EMTP system framework composed of linear and non-linear components**



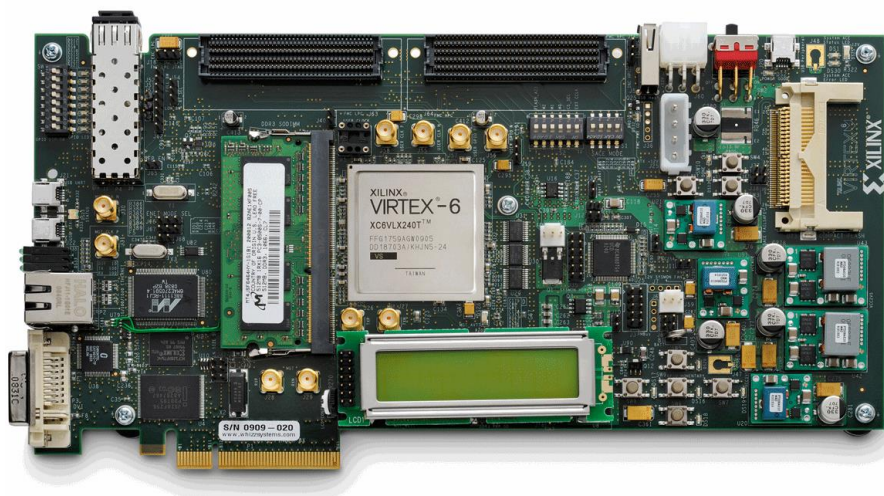
**Fig. 2.7 Simulation algorithm for hybrid double-precision and single-precision floating point**

In order to exploit the architecture including three types of components as well as conversion among various types of numbers, the hybrid solution algorithm ought to be analysed. Fig. 2.7 provided an advanced solution for handling mixed-precision EMTP system, which can be divided into five steps as follows:

- The initial step is to initialize relevant values, for example, calculation index and time step should be set as 0 and 50 us respectively.
- In addition, the second step is to establish equivalent history voltage source models for linear transmission lines, transformers and RLC branches based on classic electromagnetic transient models.
- And the following step is to determine the equivalent terminal voltage at present for each partial non-linear synchronous machine model based on node voltage solution. Therefore, the large-scale system can be simplified to several decoupling partial subsystem to reduce simulation complexity.
- Next, the third step is to give those local synchronous machine models present parameters under steady-state or oscillated-state initial condition. Capturing reasonable angular speed and field voltage by using prediction system is the essential task in this step. If the predicted values are far away from the typical limit range, move to next iteration until the limit is reached.
- And the fourth step is to obtain the single-precision terminal voltage from each local governor by double-to-single conversion block. In addition, this voltage ought to be provided to the linear network system, so that other parameters of the total original network can be finally simulated. For the next simulation step, store the present parameters in the memory space as history data until total simulation time is finished.

## 2.4 Case Study

All the case studies are implemented using single Virtex6 ML605 FPGA board as shown in Fig. 2.8. This board has been equipped with 768 Digital Signal Processors (DSPs), 416 RAMs, 301440 registers and 150720 Look-Up Tables (LUTs) [97]. A LUT is the basic building block of an FPGA and is capable of implementing any logic function of Boolean variables. A DSP slice is a digital signal processing logic element to perform different arithmetic operations on FPGA [98]. 100MHz system clock is selected to control main module and sub-modules in FPGA board. All the calculations are using Intellectual Property Cores (IP Cores). The simulation step for EMT programs is  $50 \mu\text{s}$ .



**Fig. 2.8 Virtex6 ML605 FPGA board**

IP Core is a reusable HDL component for FPGA, allowing users integrate complete implementations using design tools [99]-[100]. The Xilinx Floating-Point Core provides designers with the means to perform Floating-Point arithmetic on FPGA devices, such as



addition and multiplication [98] - [101]. Operation, length, latency and interface can be customized. The Xilinx Block Memory Generator Core provides high-performance memories at up to 450 MHz [101]. Operation mode, data width, memory depth, initialization can be optimized. Table 2.4 summarizes the key features of Floating-Point Core and Block Memory Generator.

**Table 2.4 Summary of Floating-Point Core and Block Memory Generator.**

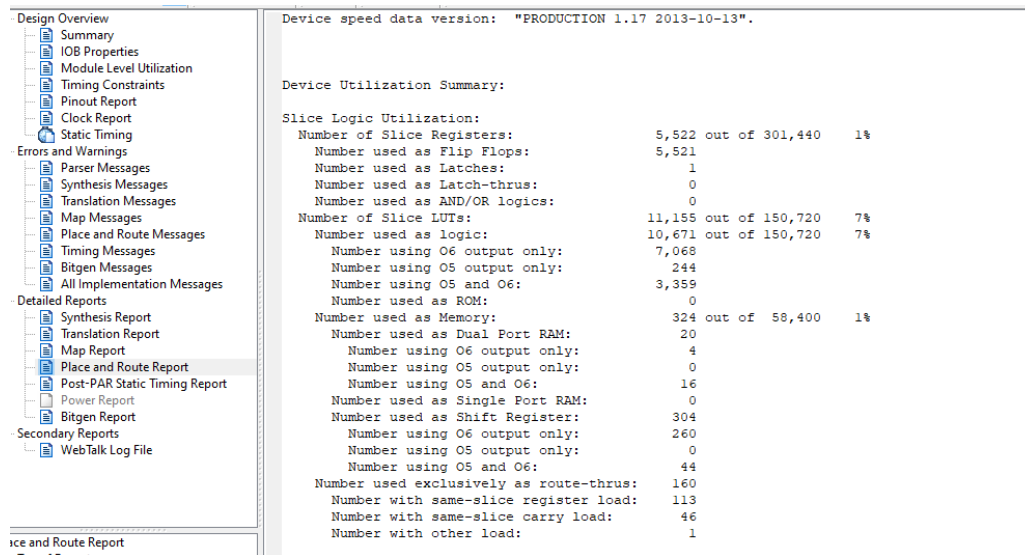
IP COREs	Floating-Point Core	Block Memory Generator
Functionality	Calculations	Memories
Supported operators	Multiply, add, multiply, divide, square-root and etc.	Dual-Port RAM, Single-Port ROM and etc.
Arithmetic	IEEE-754 Standard Floating-Point	IEEE-754 Standard Floating-Point, binary, hexadecimal and etc.
Customizations	Latency, speed, word length and etc.	Data width, memory depth, initialization, operating mode and etc.

### 2.4.1 Hardware validation

In this thesis, hardware performance is validated in three aspects, including resource utilization, accuracy and timing:

- Resource utilization: Resource utilization is to estimate occupied resources on single ML605 FPGA board. The size of a network that can be simulated with an acceptable time step depends on the available FPGA resources [88]. Fig. 2.9 presents that device utilization summary will be displayed in the form of Place and Route report immediately after generating bitstream file [102]. Lower resource utilization means that less hardware is required to do larger and more complex power system simulations. Therefore, all the resource utilization percentage, such as Look-up Table (LUT),

registers and DSP, should be less than 100% to avoid overused resources. Otherwise, the bitstream file, executable program, could not be generated if resources such as LUT, registers and DSP are overused.



**Fig. 2.9 The Place and Route Report**

- Accuracy: According to existing researches, real-time digital simulator results were verified for accuracy by simulating the same system in off-line software, such as EMTP, ATP and MATLAB [103] - [105]. In this thesis, accuracy is compared with a referenced off-line simulation benchmark in MATLAB 2018b in windows 64bits system. The simulation step is 50 us in both MATLAB and FPGA. The simulation results in FPGA are obtained in binary format from Chipscope tool [106], which allows insert logic

analyzer, system analyzer, and virtual I/O low-profile software cores to view any internal signal. Then these data can be converted from binary to double floating-point in MATLAB. Using graphical functions in MATLAB, the results obtained from FPGA can be compared with outputs obtained from MATLAB.

- Timing schedule to ensure real-time implementation: Timing schedule is to analyse the total time schedule so as to determine whether the FPGA based EMT model is implementable for the chosen simulation time step  $\Delta t$  (typically 50  $\mu s$  or smaller). Implementable means execution time should be smaller than the chosen simulation time step. Simulation starts immediately once the download is finished. If the first simulation loop cannot be finished in this time step, timing errors will be displayed at failed timing constraint [88]. Once timing errors in **Fig. 2.10** occur, bitstream file actually cannot be generated and downloaded in FPGA. In other words, to enable real-time processing, the execution time should be smaller than the chosen simulation time step, say 50  $\mu s$ .

Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1 No	TS_Clk100M = PERIOD TIMEGRP "Clk100M" 10 ns HIGH 50%	SETUP	-1.100ns	11.100ns	47	23506
		HOLD	0.002ns		0	0

**Fig. 2.10 Timing errors**

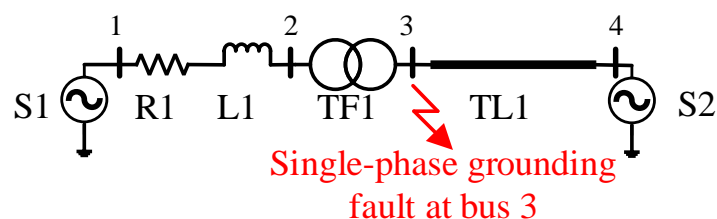
In summary, in order to ensure real-time implementation, resource utilization and timing constraints should be operated within reasonable limits. In other words, bitstream file can not be generated when overused resources or failed timing constraints occur.

The following case studies are all focused on modelling transmission systems, including a 5-bus linear network, synchronous machine and 11-bus 4-machines network. Considering the impact of accuracy, it has been found that a single phase to ground fault (asymmetrical fault) is more influential than a three-phase to ground fault (symmetrical fault) [24], and based on this consideration, a single-phase grounding fault is chosen to demonstrate the significance in 5-bus linear network, synchronous machine and 11-bus 4-machines network, while a three-phase grounding fault is also modelled to enhance the effectiveness on 11-bus 4-machines network.

#### **2.4.2 4-bus linear network**

To compare the effect of Single-Precision and Double-Precision on non-rotating components, a hypothetical 4-bus network is proposed in Fig. 2.11. This case study is to simulate the transmission system with two sub transmission networks and the transformer (TF1) transform is connected between the two sub transmission systems of different voltage levels. The left sub transmission network consists of bus 1-2. RL branch (R1 and L1) represents simplified overhead lines and S1 represents an ideal voltage source, representing an infinite-bus of the power grid, The right-side sub transmission network consists of bus 3-4. TL1 represents a transmission line using a traveling wave model and S2 represents an ideal voltage source, representing another infinite-bus of the power grid. In order to exploit further transient behaviours, a single-phase phase grounding fault is applied at bus 3 from 11.5s to 12.5s. For the purpose of observing long-duration accumulation, simulation time and time step are 20s

and  $50\mu s$ , respectively. To compare accuracy, full Double-Precision and Single-Precision is applied for whole network.



**Fig. 2.11 Simulation topology of 4-bus network**

shows the absolute error comparison of branch current and voltage between SIMULINK and FPGA in respectively. Seen from Fig. 2.12, it demonstrates that the absolute error of double-precision format network model is limited in a range, while that error for single-precision is increasing along with simulation time. For 4-bus system, Single-Precision and Double-Precision can both reduce relative error to less than 1%. The error increment is caused by the lack of data bandwidth. This also verifies that Double-Precision can avoid numerical error to the most degree for long-duration simulation.

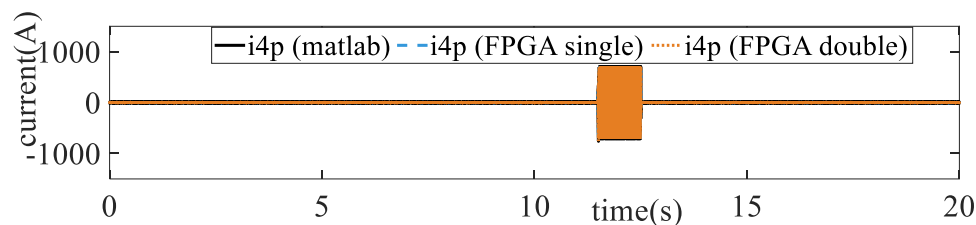
Table 2.5 provides the detailed resource utilization for 4-bus network using Single-Precision and Double-Precision. Compared with Single-Precision scheme, Double-Precision scheme cost more 10% Registers and 7% LUTs. In particular, Double-Precision also utilizes 23% DSP exclusively. This resource utilization comparison also proves that the accuracy improvement of Double Precision is a treat-off in exchange of nearly one-time resource of using Single-Precision. Lower resource utilization means more components can be further involved in real-time simulation. Therefore, Single-Precision algorithm is more scalable by saving more resources.

This is mainly because that the amplitude for abandoned bits is expanding during accumulation. And this also verifies the double-precision data can improve accuracy by avoiding most of accumulation error especially for long-duration simulation.

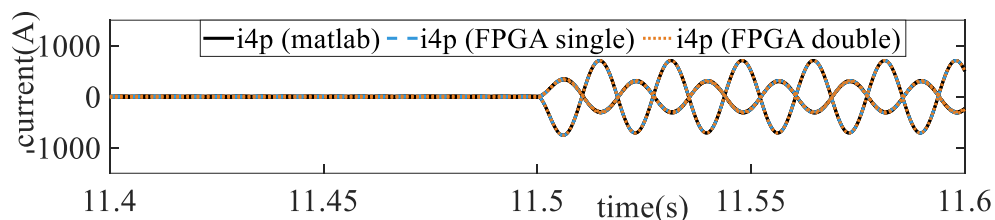
As the rotating process is not including in this model, the accumulation error is single-direction influencing other components, which can not give any feedback for voltage source. Therefore, single-precision format accuracy is still acceptable for this model. Besides, considering the Double-Precision model occupies nearly twice resources for Single-Precision model, Single-Precision format is still recommended for non-rotating components in EMTP system.

**Table 2.5 Resource Comparison on A 4-bus Electric Network without SGs**

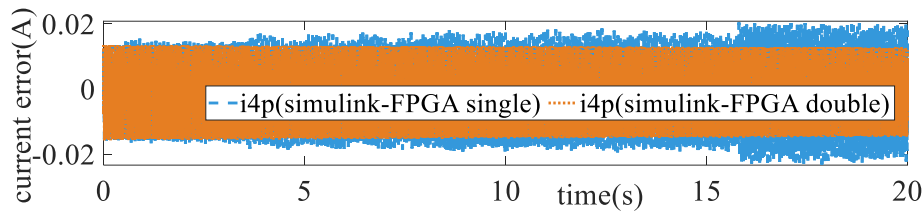
	Registers	LUTs	Memory	DSP
Full Single-Precision	13%	23%	1%	0%
Full Double-Precision	23%	30%	0%	23%



(a) Long-duration comparison



(b) Short-duration comparison



(c) Absolute error comparison

**Fig. 2.12 Simulation results of currents of a 4-bus system. (a) Long-duration comparison (b) Short-duration comparison (c) Absolute error comparison**

### 2.4.3 Synchronous Machine

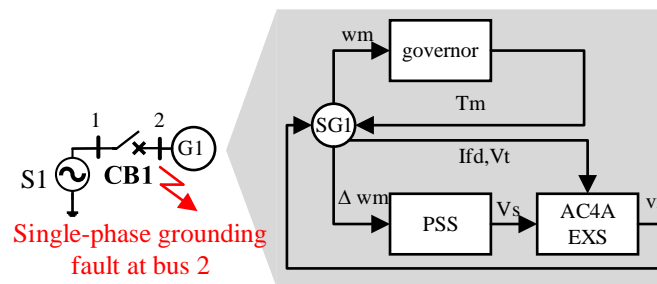
To analyse the effect of precision on rotating procedure, a hypothetical topology is presented for starting-up synchronous machine in Fig. 2.13. S1 represents an ideal voltage source, representing an infinite-bus of the power grid. Circuit breaker CB1 is closed at time 0, connecting the synchronous machine G1 to the power grid. This case study is to simulate starting-up procedure of synchronous machine. For exploring precision effect, four schemes, including Single-Precision, Single-Precision with iteration, Mixed-Precision, and Double-Precision, are proposed as follows:

- Single-precision synchronous machine model: All the floating-point matrix and vectors are stored and calculated in single-precision format.
- Single-precision synchronous machine model with iteration: An iteration steps is added, in which field voltage and angular speed in first calculation step is reset as the

prediction values for the second calculation step at the same time step compared with model (1).

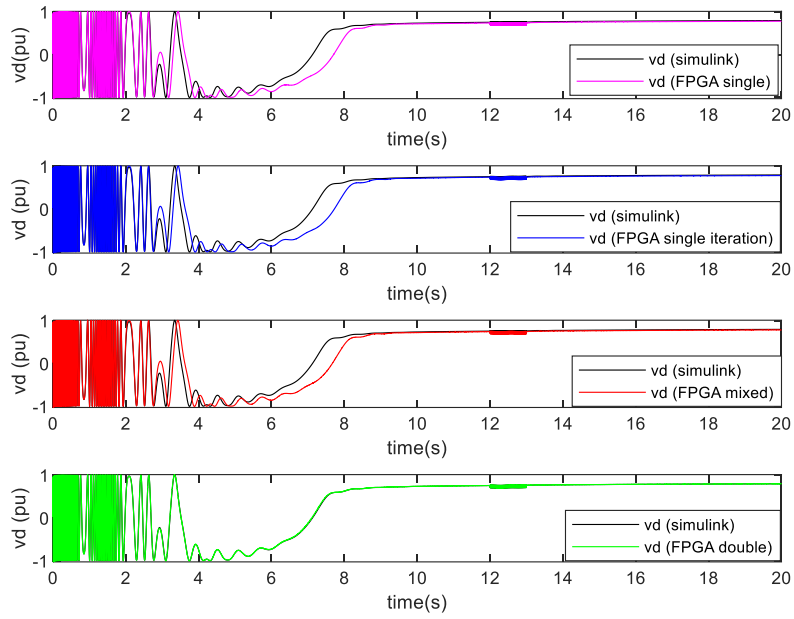
- Mixed-precision synchronous machine model: Electric angle update block is simulated in double-precision format. And the input data flow and output data flow are transferring by double-to-single and single-to-double interface to other single-precision blocks compared with model (1).
- Double-precision synchronous machine model: All the floating-point matrix and vectors are stored and calculated in double-precision format.

The simulation time duration and time step are 20 s and 50 us, respectively. A single-phase grounding fault is applied at node 2 from 12s to 13s. Fig. 2.14 shows the d axis voltage, d axis current, electric torque and mechanical torque comparison and absolute error of synchronous machines models of 4 types, in which MATLAB SIMULINK model is selected as the reference standard model. Table 2.6 gives the resource utilization comparison using different schemes.

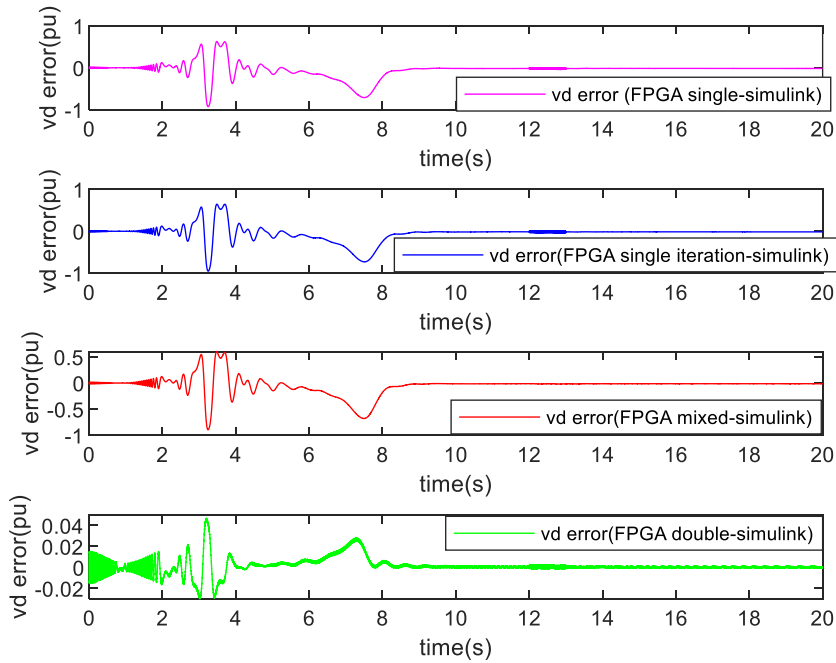


**Fig. 2.13 Synchronous machine with AC4A excitation system, PSS and governor system**

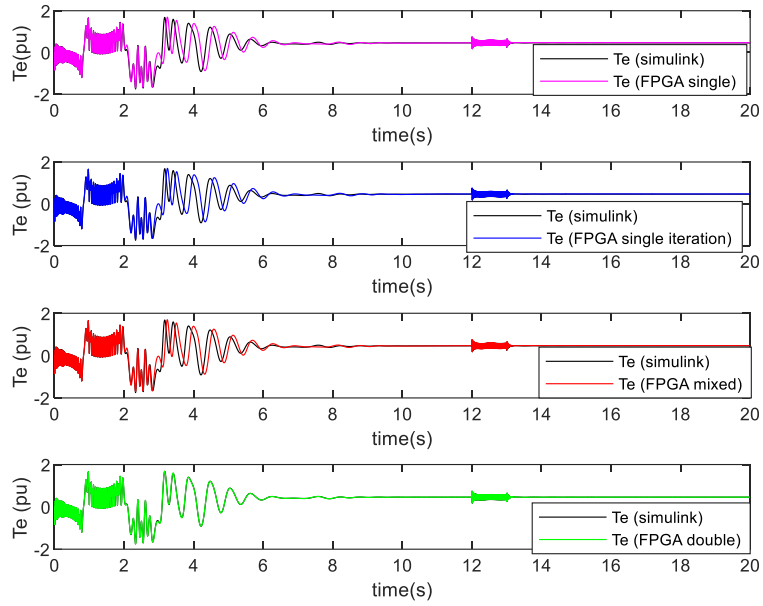




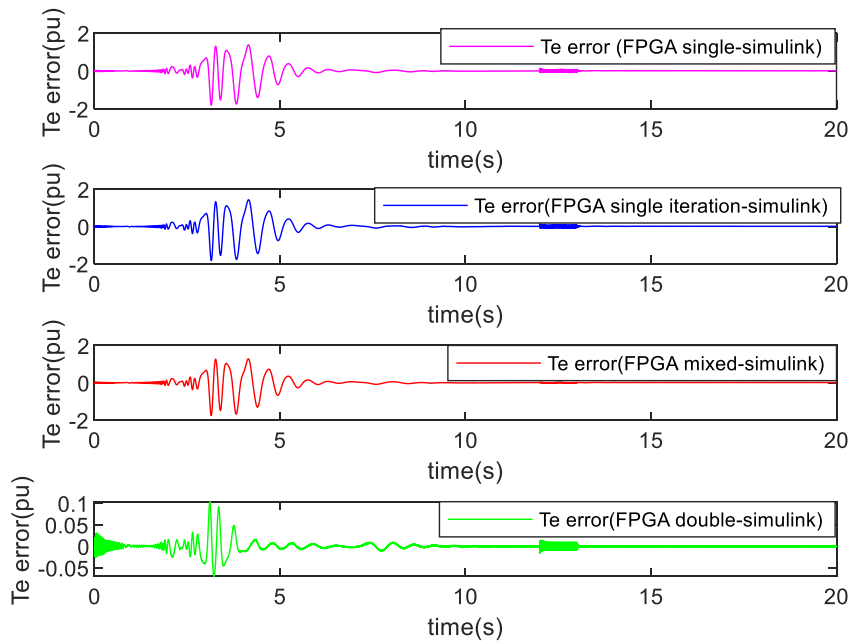
(a) D axis voltage comparison



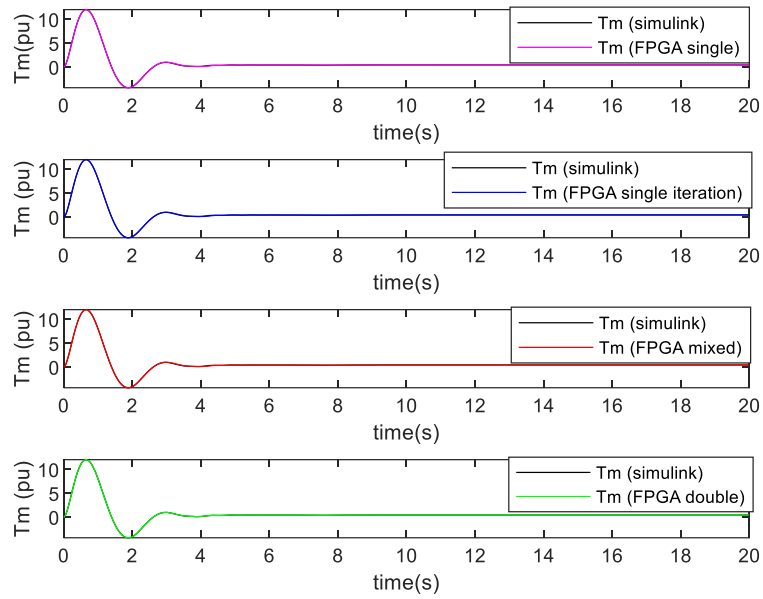
(b) D axis voltage absolute error comparison



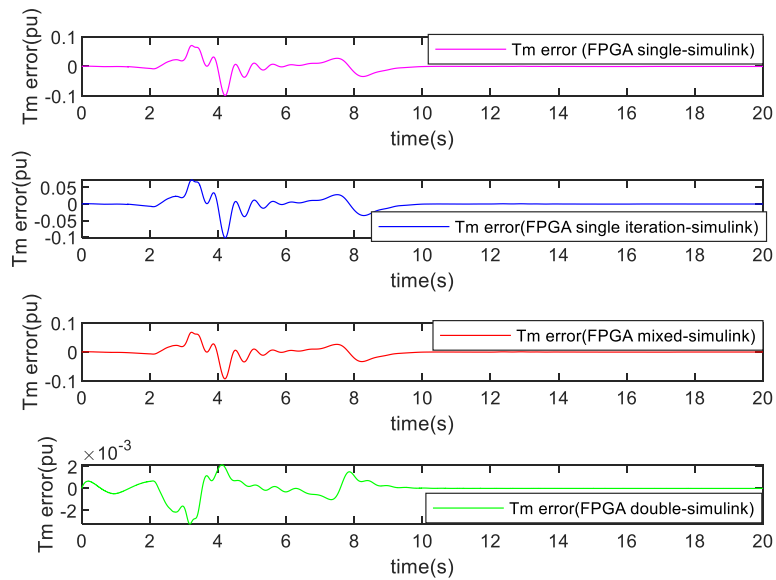
(c) Electric torque comparison



(d) Electric torque absolute error comparison



(e) Mechanical torque comparison



(f) Mechanical torque absolute error comparison

**Fig. 2.14 4 types of synchronous machine simulation results**

Table 2.6 Simulation resource utilization of synchronous machine models of 4 types in ML605

For start-up procedure from 0s to 8s, there is a phase angle problem appearing in the Single-precision, Single-Precision with iteration and Mixed Precision synchronous machine models. Only Double-Precision can match with referenced SIMULINK model with less than 2% relative error. And this due to the half bandwidth of single-precision format. As electric angle and other elements are based on the accumulation, the relative error will increase gradually as data becomes massive. This will cause divergence for swing equation and Park's transformation. The error in start-up process is not able to be eliminated periodically by local high Double-Precision.

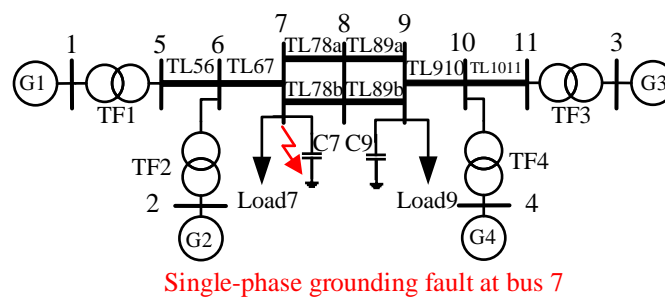
For single-phase fault from 12s to 13s, Mixed Precision can eliminate local error to less than 5% for the synchronous machine. Although the Mixed-Precision scheme can not perform the same high accuracy as the Double Precision for whole simulation period, it still has the potential to provide relatively high accuracy.

Seen from Table 2.6, with the best accuracy, Double Precision scheme still uses nearly twice resources than Single-Precision scheme. Especially, Double-Precision requires 48% DSP resources. Mixed Precision only consumes 1% register than Single-Precision to eliminate local errors to less than 7%.

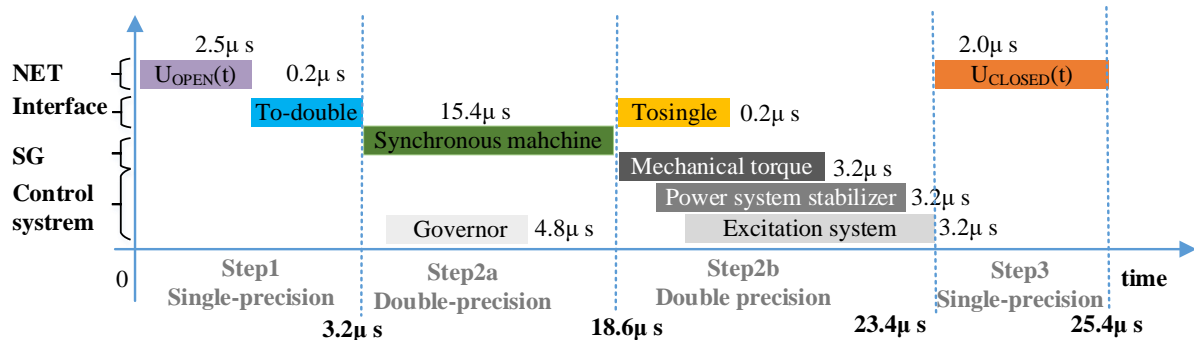
#### **2.4.4 11-bus 4-machines network**

To verify the application of the Mixed-Precision in network, Double-Precision and Single-Precision , respectively, are applied to synchronous machine and other non-rotating

components. For accuracy comparison, Kundur's 11-bus system given in Fig. 2.15 is implemented by Mixed-Precision and Double-Precision respectively. A single-phase grounding fault is applied at bus 7 from 1s to 1.1s. Fig. 2.17 shows the detailed results and absolute error comparisons of electric torque, mechanical torque and current. Fig. 2.16 shows the detailed time schedule for Mixed-Precision Algorithm. Table 2.7 gives the resource utilization comparison.



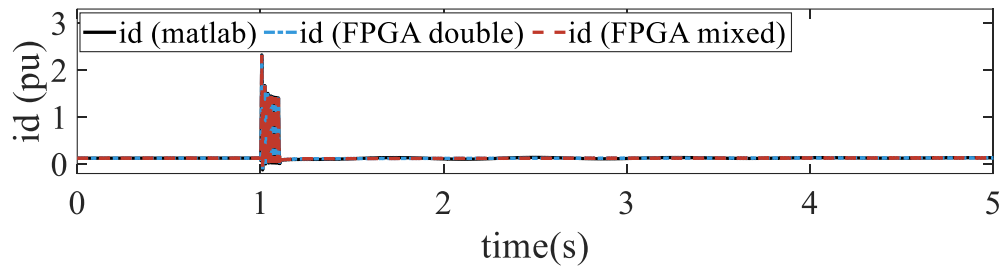
**Fig. 2.15 4-machine 11-bus test system**



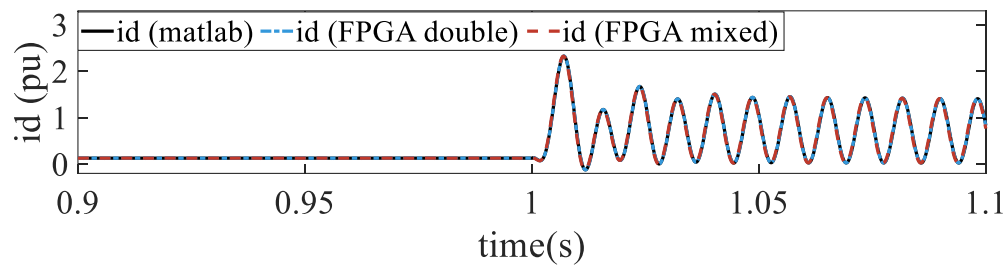
**Fig. 2.16 Time schedule of Mixed-Precision structure**

**Table 2.7 Resource Comparison of Kundur’s System Case Study**

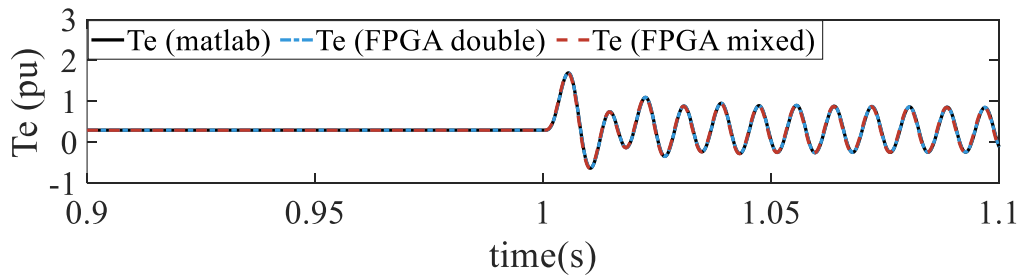
	Registers	LUTs	Memory	DSP
Mixed-Precision	29%	60%	6%	78%
Full Double-Precision	35%	87%	9%	93%



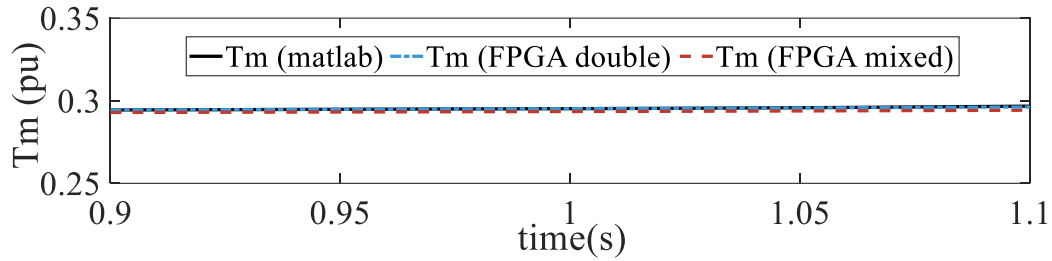
(a) Long-duration d axis current



(b) Short-duration d axis current



(c) Short-duration electric torque



(d) Short-duration mechanical torque

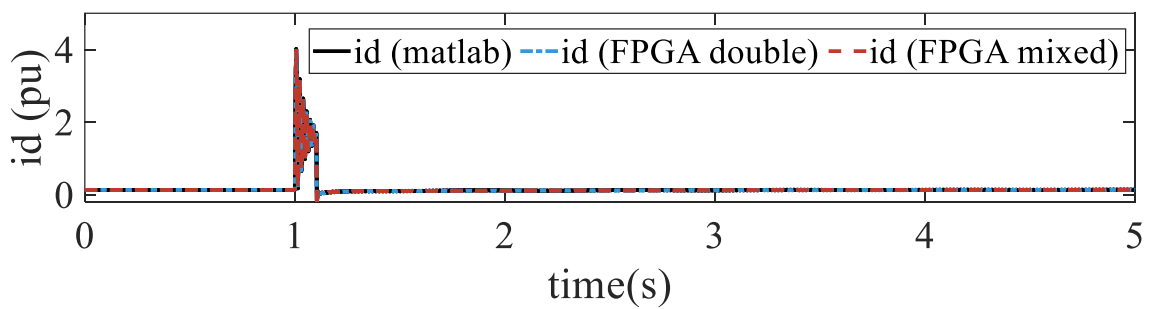
**Fig. 2.17 Results of synchronous machine G1 in Double-Precision and Mixed-Precision when applying single-phase grounding fault**

It can be seen from Fig. 2.17 that the relative error of electric torque, currents and mechanical torque can be maintained less than 5% and 2% respectively. Both of Mixed-Precision and Double-Precision can implement highly accurate transient behaviors for 11-bus network. Within Mixed Precision Scheme, the Park-Transform rotating dynamics and control actions of synchronous machine is totally maintained by local Double-Precision. In addition, the external network change is only acting as single-direction feedback for synchronous machine, and will not influence the internal Park Transform.

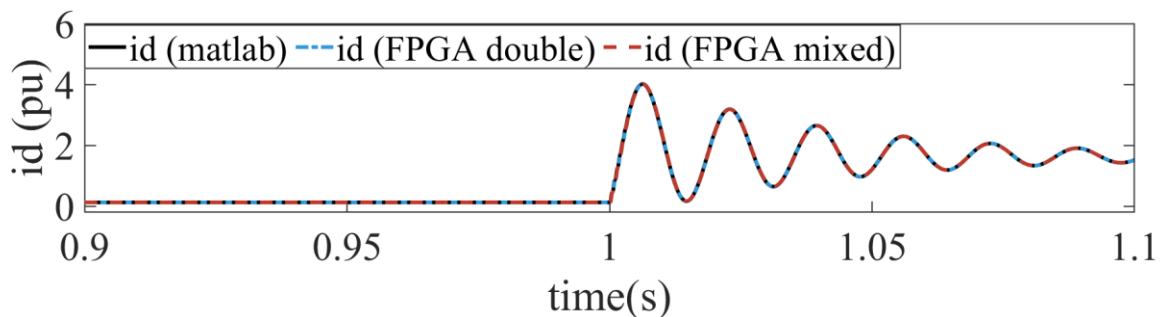
Except for the data conversion between Single-Precision and Double Precision, the time schedule of Single-Precision and Double-Precision is nearly the same as the Mixed-Precision algorithm. Fig. 2.16 demonstrates the total simulation time for Mixed-Precision structure. The total calculation only costs  $25.4 \mu s$  at simulation time step  $50 \mu s$ . Therefore, further Single-Precision and Double-Precision calculation blocks are still allowed at available time slots.

Seen from Table 2.7, it is observed that Mixed precision can save 27% LUT and 15% DSP than fully Double-Precision scheme. Mixed-Precision algorithm is more scalable by saving more resources. For the same board, using Mixed Precision can implement larger system than Double-Precision. This resource-saving advantage make Mixed-Precision algorithm can easily adjust to different FPGA boards with highly portability and flexibility.

Single-phase grounding fault causes unbalanced voltage drop at each phase, while three-phase grounding fault is symmetrical. To demonstrate the simulation capability, a three-phase grounding fault is also applied at bus 7 from 1s to 1.1s in 11-bus 4-machines network. Fig. 2.18 presents the simulation result for G1 when applying three-phase grounding fault.

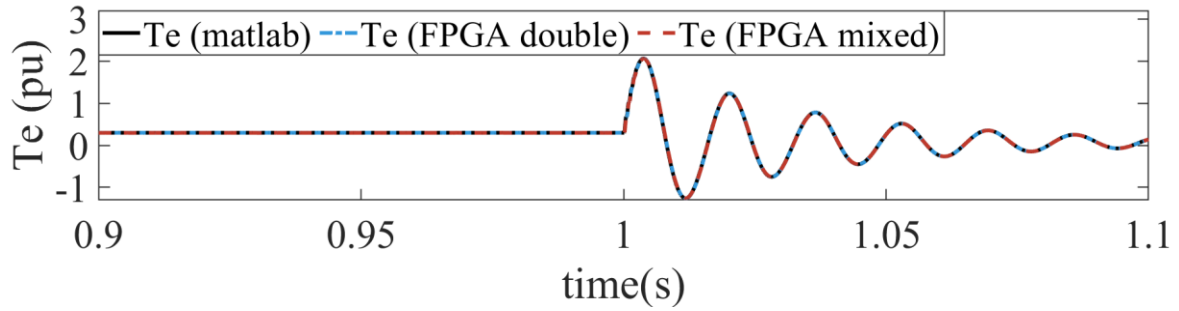


(a) Long-duration d axis current

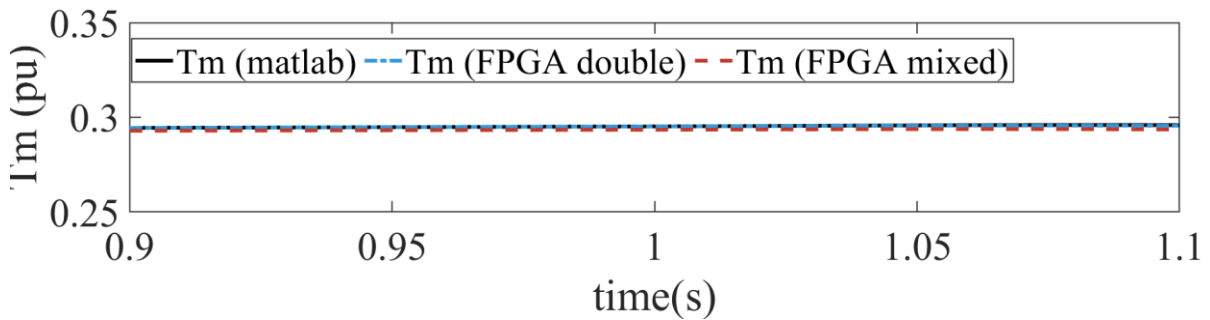


(b) Short-duration d axis current





(c) Short-duration electric torque



(d) Short-duration mechanical torque

**Fig. 2.18 Results of synchronous machine G1 in Double-Precision and Mixed-Precision when applying three-phase grounding fault**

As illustrated, the relative error of electric torque, currents and mechanical torque can be maintained less than 5% and 3% using Mixed-Precision and Double-Precision, respectively. When a three-phase grounding fault occurs, these two algorithms can still match the referenced model in MATLAB. It can be demonstrated that the proposed mathematical models, such as synchronous machine and RLC branch, can well represent the dynamic behaviors under symmetric and asymmetric faults of EMT events.

To trigger higher frequency transients [24], the standard deviation is used to measure the variation of electric torque. In the referenced model, standard deviations of electric torque are 0.3318 and 0.0892, respectively where the impact of the single-grounding fault is much more significant than that of the three-phase grounding fault. This is a solid evidence why single-phase grounding fault is more preferable for EMT simulation than three-phase grounding fault.

## 2.5 Conclusion

This chapter has demonstrated the difference between Single-Precision and Double-Precision for FPGA solver. To maintain the benefits of fast-speed and high accuracy, a Mixed-Precision algorithm is built up for solving large system. To verify the effectiveness and advantages of Single-Precision, Double-Precision and Mixed-Precision, linear component, non-linear component and integrated system simulation are both simulated. By theoretical analysis and simulation results, the following conclusions can be given:

- The dynamic transient behaviors can be well represented by developed mathematical models, such as synchronous machine and RLC branch, when symmetrical and unsymmetrical EMT event occurs.
- For linear components, Single-Precision and Double-Precision can achieve high accuracy for linear non-rotating components.
- For non-linear rotating components, only Double-Precision can achieve the best accuracy for both start-up and transient state. By utilizing twice resources than Single-precision and Single-Precision with iteration, the original phase shift error of can be eliminated.
- For local fault analysis, Mixed-Precision scheme is the most resource-saving scheme to reduce local error to less than 5%. This method has high portability to apply both single machine model and system level model on different boards.

# CHAPTER 3 FOUR INITIALIZATION MEHODS FOR FPGA-BASED EMT SIMULATION

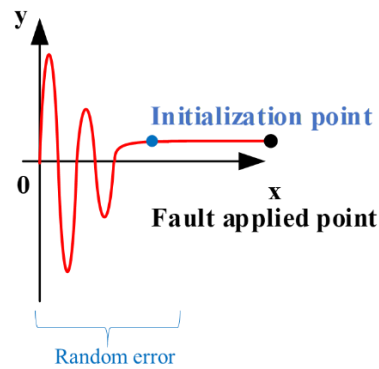
This chapter firstly introduces the motivation for fast initialization for FPGA-based EMTs in Section 3.1. Section 3.2 introduces key issues for initialization, including model type, memory unit and initialization sequence. For generic programming, Section 3.3 provides the programming codes for different initialization methods, including physical interface (Method 1), signal declaration (Method 2), signal assignment (Method 3), COE file. Section 3.4 gives the theoretical comparison in schematic and time schedule, providing ahead-of-time evaluation. Section 3.5 develops implementation structure and algorithm for processing Method 4 efficiently. In Section 3.6, device-level and system-level case studies are both provided to verify the effectiveness of initialization and compare initialization performance. In Section 3.7, the conclusions for FPGA-based initialization are summarized for this chapter.

## 3.1 Motivation

There are four main reasons for FPGA-based initialization:

- For the purpose of high accuracy, uninitialized solution will accumulate uncertain error at hardware environment. Numerical error is easy to accumulate from zero-time point to initialization point, which will influence further calculation, such as fault applied point. This error is random, and unable to be eliminated periodically. This will introduce uncertainties on amplitudes and phases in global system, such as currents and voltages. Fig. 3.1 details accumulated error of non-initialization. Furthermore, this

problem will accelerate with the increase in system scale and simulation time, which is essential for large systems.



**Fig. 3.1 Random error of non-initialized calculation**

- For FPGA processing, non-optimized initialization programming can bring the problem of timing constraint failure and overused resources. This is because FPGA can use global resources to adapt to bad design in small systems but unable to expand non-optimized design for larger systems. But bad design has low portability and unable to expand for larger scale system. Besides, FPGA design is facing timing constraint, resource utilization and accuracy at the same time. This will make this problem even more complicated. This problem will only occur until system reaches the certain level. The FPGA optimization is always facing the problem of managing timing constraint and routing at the same time. Unreasonable initialization is able to ruin original design and structure, making program unable to implement.
- The third reason is to make initialization simple and easy to use for beginners. And system scalability and accuracy will not be impacted by the involvement of initialization.

- The fourth reason is to verify whether signal declaration and signal assignment are still effective for initializing large systems. For initializing single signal, signal declaration and signal assignment are easy to implement. However, EMT initialization for a network, i.e., the 11-bus network, normally consists of massive data. The EMT initialization still remains uncertain whether signal declaration and signal assignment are achievable for large systems, and hence it is necessary to find alternative initialization methods.

Therefore, this Chapter proposes four different initialization methods to give a direct graphical and numerical comparison. This chapter mainly solving following problems:

- Whether device or system is worth to be initialized?
- Even using different initialization methods, is the same performance for scalability?
- Which method is the best for further FPGA programs?

## 3.2 Key Issues of Initialization Models

### 3.2.1 Initialization model type

To update instantaneous values, history values are required to formulate the history vectors in different solutions. Take nodal solution as an example, history current source vector is required to update present voltages as given in Chapter 2. Therefore, history values are essential memory units for providing initial operation point for initialization.

According to the interactive relationship with current values and history values, the initialization of EMT models can be classified into three different types: current type, history point type, and history delay type.

For the present type, initialization is not required. Because this type is calculated by present value directly instead of history values. A variety of present type components include resistance branch and proportion in control system. The exact relationship is given as following:

$$i_R(t) = k_1 v_R(t) \quad (3-1)$$

where  $i_R(t)$  and  $v_R(t)$  is the present output and input,  $k_1$  is the constant parameter.

For the history point type, only history values at last time step are necessary for initialization memory. Components of this kind consist of transformer, synchronous machines and PID control systems, in which values at last time point is able to decide future initialization at next point given as follows:

$$i_{PID}(t) = k_2 v_{PID}(t) + k_3 v_{PID}(t - \Delta t) + k_4 i_{PID}(t - \Delta t) \quad (3-2)$$

where  $i_R(t)$  and  $v_R(t)$  are the present output and input, respectively, and  $k_1$  is the constant parameter.

For the history delay type, initialization is determined by a continuous duration of history values. For example, the terminal current is updated by a duration of history voltage and currents for calculating distributed parameter transmission line model. Detailed calculation based on traveling wave model is given as follows:

$$i_s(t) = k_5 v_s(t) - I_s(t - \tau) \quad (3-3)$$

Where  $\tau$  is the time delay,  $i_R(t)$  and  $v_R(t)$  is the present output and input,  $I_s(t - \tau)$  is the history value before time delay,  $k_5$  is constant parameters.

Above all, Table 3.1 summarizes the required variables and corresponding time sensitivity range for EMT initialization.

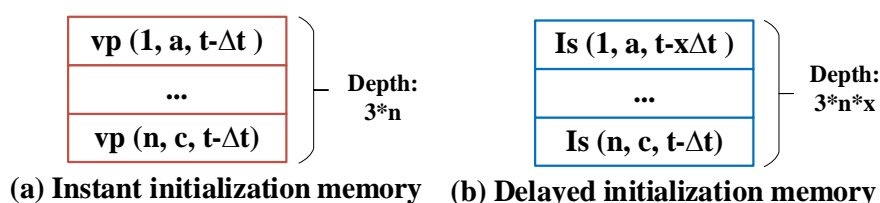
**Table 3.1 Features of Required Variables in EMT Initialization**

	History variable	Time range
RLC	History phase current and phase voltage	t
TL	History modal current, modal voltage and fixed distance history current source	(t- $\tau$ ,t)
SG	History current, voltage, flux and control system history values	t

### 3.2.2 Memory unit



Consider memory usage might exceed the physical memory, initialization memory has to be optimized in advance for different models, avoiding overused memory. To achieve higher memory intensity for hardware, grouped data should be used repetitively. As a result of that, different addresses can be applied to search different values in the same memory unit. Consider different numerical sensitivity, instant and delayed memory units are both developed. Fig. 3.2 demonstrates detailed data structure for these two types.



**Fig. 3.2 Initialization memory data structure**

Instant initialization memory is designed for components only sensitive to previous time step  $t - \Delta t$ , such as transformer and synchronous machine. Consistent initialization memory is for components are related to history values within  $(t - x\Delta t, t - \Delta t)$  such as transmission lines. With the involvement of address table, the same memory unit is able to generate different results along with calculation stages.

### 3.2.3 EMT model initialization sequence

Memory data sequence plays an important role in FPGA-based global optimization. Well-organized initialization sequence is able to achieve much lower timing constraint and resource utilization than random sequence for implementing the same system scale. The optimized sequence is applied universally for both initialization stage and repetitive calculation stages.

To simplify initialization sequence, the relationship between phase, component type and time delay should be integrated in a single dimension. Consider the component dynamics in (3-1) - (3-3), the searching addresses for initialization memory unit can be classified into three different types: instant sequential address, instant random address and delayed address.

When initialization memory sequence is the same as calculation search sequence, instant sequential address is mainly designed. This kind of address includes initializing RLC branch using history currents in (3-2). (3-4) gives the detailed address search for instant sequential address.

$$i_{RLC}(s, t - \Delta t) = i_{RLC}^{INI}(s, t - \Delta t) \quad (3-4)$$

where  $i_{RLC}$  and  $i_{RLC}^{INI}$  are the current variables for calculation and initialization, respectively,  $s$  is the calculation sequence.

When initialization memory sequence is different from search sequence, instant random address is utilized. This type address includes initializing voltage variable in (3-2). Therefore, detailed address search for Instant sequential address is given as following:

$$v_1(s, t - \Delta t) = v_p^{INI}(m(s), t - \Delta t) \quad (3-5)$$

where  $v_1$  and  $v_p^{INI}$  are the phase voltage variable for calculation and initialization, respectively,  $m(s)$  is the look-up function for random sequence.

Delayed addresses are designed for history variables are exclusively related to multi-dimension variables. For example, the address search for transmission line history current source in (3-3) is determined by consistent time duration component number and phase. (3-6) demonstrates

that delayed initialization address function is a multi-factor function sensitive to calculation sequence and time delay.

$$I_{TL}(s, t - y\Delta t) = I_{TL}^{INI}(f(s, y)) \quad (3-6)$$

where  $I_{TL}$  and  $I_{TL}^{INI}$  are the transmission line history current variable for calculation and initialization, respectively,  $y$  is the delayed time slot,  $f(s, y)$  is the combined look-up sequence function.

### 3.3 Four Initialization Methods

To fully explore initialization potential, FPGA-based initialization is flexible to use different initialization data source and memory allocation. The initialization source comes either from real-time FPGA board or off-line software. In addition, memory allocation is able to be implemented in VHDL module signal declaration, signal assignment and off-line memory IP COREs. Therefore, initialization is able to be classified into four different types by physical port (Method 1), signal declaration (Method 2), process structure (Method 3) and COE file (Method 4), respectively. Among them, Method 1 is completely FPGA-to-FPGA initialization, Method 2-4 is software-to-FPGA initialization.

For small scale systems, using different methods might give similar performance. The effect of an optimal method will be more significant for large systems. The computational advantage will accumulate with the increase in system size.

#### 3.3.1 Method 1-Physical Interface

**Table 3.2 Programming code for Method 1**

---

Method 1 (Physical interface in VHDL module):

---

```
MODULE: INI_CODE PORT MAP
```

---

```
(CLK=>CLK, out1=>in1,...,          outQ=>inQ);
```

---

In the design of FPGA-to-FPGA method 1, the steady-state operation power flow is generated in real-time FPGA board A, and then received by another FPGA board B using I/O interfaces. Then FPGA board B has been started from initial steady-state condition for subsequent Electromagnetic transient calculation. Detailed programming code is provided in Table 3.2.

Method 1 presents the detailed initialization technique by physical interface. Method 1 uses *port map* in an architecture, block and configuration to make port connection to other blocks. The ports connections can via named association. The elements of an array port can be connected individually when using named association. Using keyword *open* for a port element can make it unconnected. For example, if we only need to use one output for a 3-output block, the rest output ports can be left unconnected.

The biggest challenge for Method 1 is the lower application range brought by the limitation of physical interface ports. Because there is a huge gap between maximum I/O ports and required data communication amount. If total initialization data bandwidth is  $N$  bits, physical input ports of required FPGA board is  $Q$  bits. Owing to  $Q \ll N$ , this data communication occupies more clock cycles for using same input and output ports repetitively. For instance, ML605 FPGA board only has maximum 916 binary ports, while Double-Precision Floating-Point value occupying 64 bits in memory. Additional decoder complexity and real-time communication time grows rapidly with scaling-up system. This is because multiples of data will go through in the same port, and after that these data will be classified before allocating for different components. The long initialization time is a significant barrier for large-scale system. Therefore, this method is not discussed in case studies.

### 3.3.2 Method 2-Signal Declaration

**Table 3.3 Programming code for method 2**

---

Method 2 (Signal declaration in VHDL module):

---

```
signal INI_signal1: std_logic_vector (width-1 downto 0):= INI_data1;  
  
...  
  
signal INI_signalN: std_logic_vector(width-1 downto 0):= INI_dataN;
```

---

Signal declaration initialization requires initial off-line steady-state solution to be defined as signal default values represented by IEEE 754 Floating-Point. This initialization data can be effectively achieved from off-line CPU-based software, such as PSCAD and MATLAB. This can also make FPOGA-based EMT solution more competitive even compared with powerful off-line software. Programming code is given in Table 3.3 for Method 2 using signal declaration.

Method 2 gives initialization scheme by signal declaration. A signal can have multiples of sequential drivers but only one default value. Signal can get initial default value via its declaration. If a signal can not get a default value, then the value be undefined “Z” for each bit. In scheme 2, total initialization data N are assigned in signal declaration before behaviour structure, therefore this procedure does not cost real-time clock cycle.

This method brings heavy computational burden for concurrency. Because any of EMT components need to be initialized simultaneously and randomly at initial declaration stage. Without creating sub-groups, this method is facing huge challenge for routing placements.

### 3.3.3 Method 3-Signal Assignment

**Table 3.4 Programming code for Method 3**

---

Method 3 (Process structure in VHL module):

---

```
process

    INI_signal1<= INI_data1;

    ...

    INI_signalN<= INI_dataN;

end process;
```

---

In Method 3, initial values are allocated for corresponding signals at processes structures in behavioural VHDL modelling. Rather than intensive declaration in Method 2, signal assignment at process structures should be arranged separately. If the initialization data is massive, this method will require sufficient waiting time for initializing variables in sequence.

Table 3.4 presents the code for using data assignment driver in process structure as a typical example. This is to change different drivers at different calculation stage for each signal. Signal assignment can appear inside a process or in architecture. Sequential signal assignment and concurrent signal assignment can both be used. More specially, simple concurrent signal assignment, conditional signal assignment (*if ... else...*) and selected signal assignment (*with ... select...*) can be utilized. In this part, signal assignment is applied to put a number of

assignments in the same process. Therefore, the value of a signal can be changed sequentially without any overlap.

The complexity of this method is to allocate the sequence for all initialization data. The main task is to avoid heavy concurrent tasks and put non-overlapping assignments. Apart from EMT calculation, signal assignment for initial values always comes with additional drivers. This problem is extremely challenging for timing constraints and resource utilization for scaling up the system. In particular, signals with current values and future values should be listed at specified time ranges.

### 3.3.4 Method 4-COE File

**Table 3.5 Programming code for Method 4**

---

Method 4 (COE file):

---

```
memory_initialization_radix=2;

memory_initialization_vector=INI_data1,

                                     ...,

                                     INI_dataN;
```

---

In Method 4, the steady-state solution is also generated from off-line software. The main difference is that data is kept in separate Coefficient files (COE file). COE file is an ASCII text file, including radix header and value vectors. The radix can be binary, decimal and hexadecimal. Each value is terminated by a semi-colon. The depth and width must match with



memory block components, so that the file can be executed for program. No comments are allowed in the COE file. The COE file must display in the format in Table 3.5, otherwise is not able to be recognized by FPGA IP CORE Generators. In spite of data depth and width, all initialization data N is able to be kept in one COE file if without massive read and write.

When loading massive COE files, COE file can be written by off-line software and loaded into the XILINX CORE Generator. The radix, precision, depth and width can be specified in off-line software, such as MATLAB. Using related functions *coewrite*, the off-line workspace data can be extracted into hardware memory unit. There is no need to change main programming for controllers in hardware, minimizing programming complexity for FPGA.

The most important advantage of Method 4 is the hardware programming simplicity. The main controller can remain the same in spite of initialization point changes. Except for replacing COE files, signal assignment statements, calculation function and delays can remain same. This is owing COE file is directly executed to the IP CORE Generator, independent of main module. By minimizing programming modification and memory usage, Method 4 can improve computational benefits for FPGA. The significant advantage is the specific simplicity for no extra programming modification. This benefit is more important when user is willing to observe EMT behaviour in a new time range and have to use a new initialization point for observation.

### 3.4 Theoretical Comparisons four methods

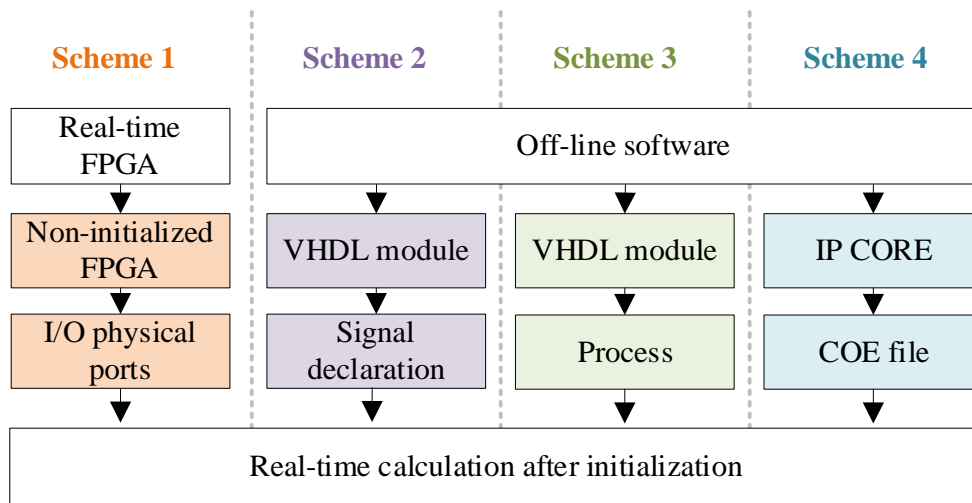
In order to compare detailed performance, procedure and time schedules comparison are compared for four initialization methods in Fig. 3.3 and Fig. 3.4. Seen from Fig. 3.3 and Fig. 3.4, Method 1, 2 and 3 still have to occupy real-time domain in parallel or sequential. Especially, Method 1 is the most difficult to implement, which is owing to the unstable interface and extra port-to-signal decoding process. With no extra module modification, Method 4 has the minimum calculation time and implementation simplicity for initializing FPGA-based EMT.

To evaluate detailed initialization time, the maximum amount of physical transceiver and internal driver is a limit for exchanging massive data. When initializing the same data bandwidth, Table 3.6 provides the detailed real-time clock cycles for four methods where  $d$  is the width for total initialization data,  $e$  is the maximum data exchange speed of physical I/O port,  $f$  is the maximum data assigned speed allowed in behavioural structure. Seen from Table 3.6, Method 3 and 4 costs no clock cycles, while Method 1 cost maximum initialization time owing to less than hundreds of ports. For example, ML605 only has 720 I/O ports, allowing processing less than 20 Double-Precision data at the same time.

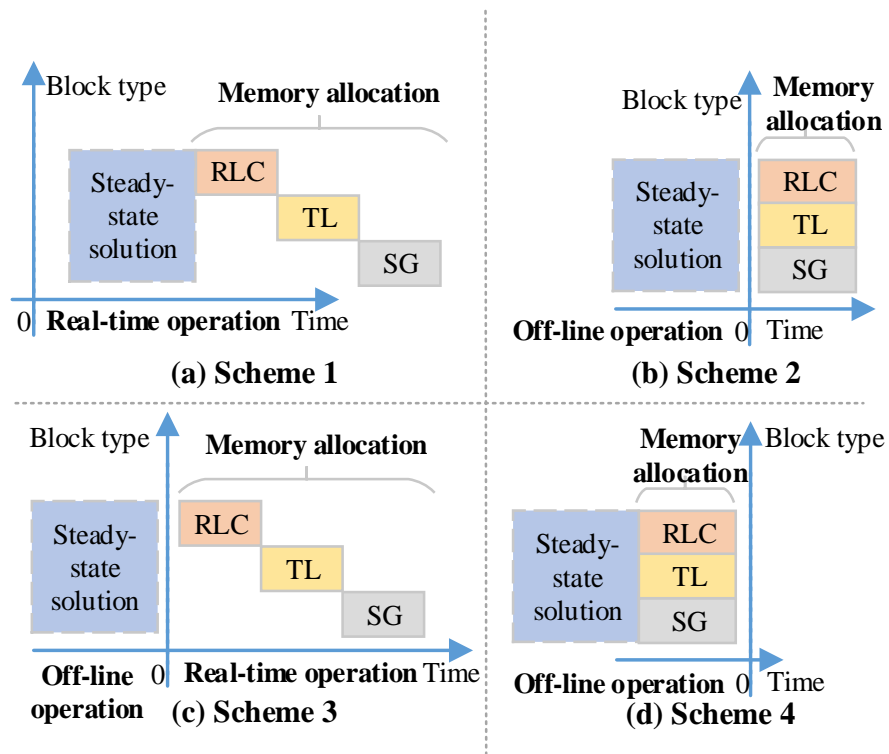
For routing architecture design, Fig. 3.5 gives the schematic comparison for original module after initialized by four methods. Owing to the need for physical communication and signal assignment, Method 1 and 3 requires additional routes. Instead of distributed task in Method 3, intensive memory unit in Method 4 is able to initialize same memory depth using only one output port within memory unit. This demonstrates Method 4 has the most area-efficient routing structure.

**Table 3.6 Real-time Occupation for Initialization Methods**

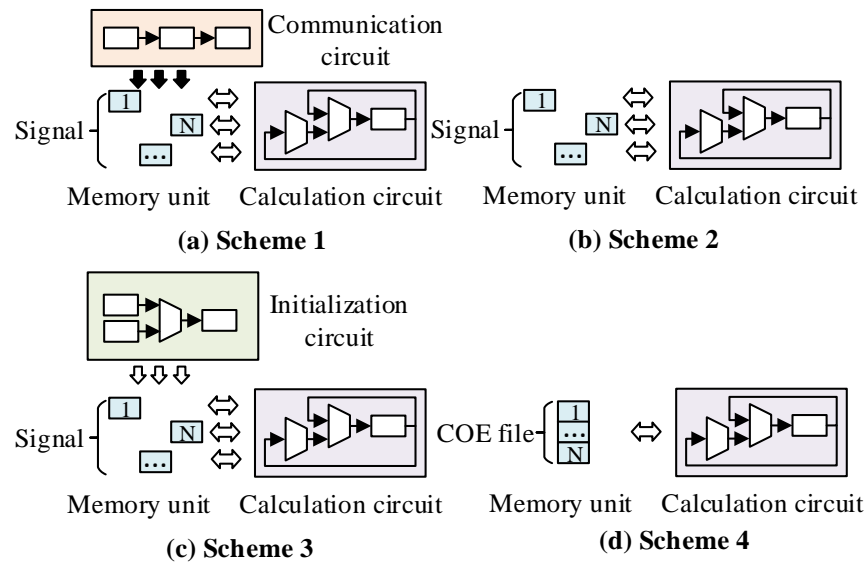
Method	1	2	3	4
Real-time clock cycles	d/e	d/f	0	0



**Fig. 3.3 Initialization procedure comparison**



**Fig. 3.4 Initialization global time schedule comparison**



**Fig. 3.5 Initialization schematic comparison**

### 3.5 Implementation Structure and Algorithm

To enable off-line initialization, it is essential to make cooperative data exchange via local and external memory units. In this Section, flow diagram and hardware structure are introduced for flexible data exchange between off-line software and real-time hardware.

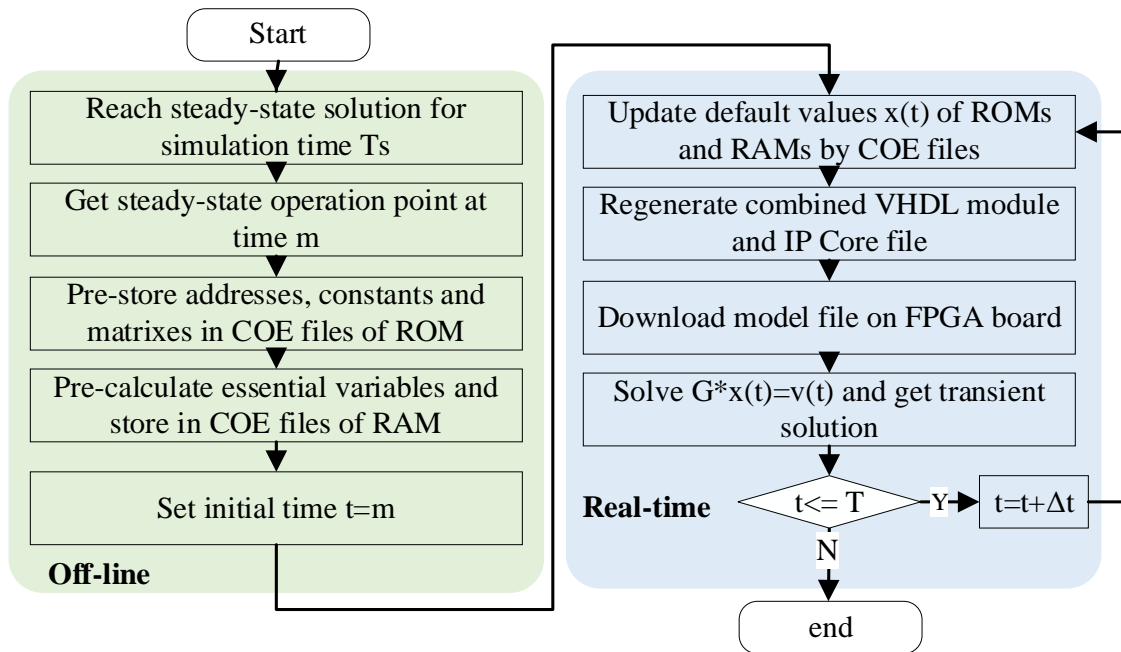
For the purpose of Method 4 acceleration, the calculation algorithm at different processing stages is described in Fig. 3.6. Therefore, the processing acceleration can be obtained when partitioning hardware task and software task.

In the off-line stage, initial operation time point  $T_5$  is set up after finishing steady-state solution. Depending on whether data are updated, separate COE files are used to keep constants and variable. The constant data, including conductance, branch index and PID parameter, should be detected are kept in COE files. The COE file with constant is executed for read-only ROMs. The variables, including voltage, current, rotor angle and i.e., are stored in other COE files. These COE files will be loaded for RAMs and will be updated in next read and write.

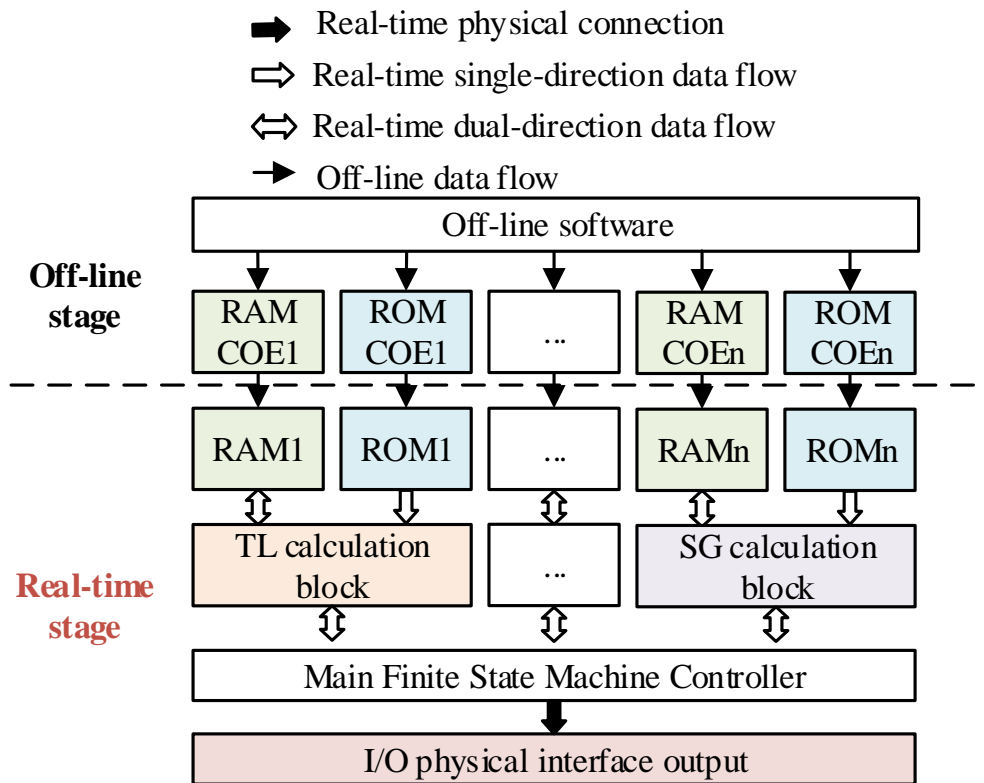
During real-time stage, variables is updated periodically in temporary RAMs at every time step. Therefore, initialization memory can be replaced by updated values. As a result of that, memory access cost and size are minimized for the same memory space. Therefore, initial values and updated values can share the memory access by this optimization. During real-time calculation, FPGA board is able to trigger specific output signal until the simulation time reaches convergence time.

By combining off-line steady-state initialization and real-time calculation, the algorithm allows solving initialized EMT model by fixed and repetitive calculation logic without interrupting existing controllers. The fan-out load can be totally reduced by sharing the same memory

access. Therefore, data placement and replacement [107] can be handled successfully in local memories and hardware memories.



**Fig. 3.6 Processing algorithm for novel initialization method**



**Fig. 3.7 Hardware structure design for off-line data flow initializing real-time module.**

To maximize the task parallelism, pipeline and parallel processing structure should be employed. For highly parallel processing, hardware structure is detailed in Fig. 3.7 for combining initialization and computation efficiently.

In horizontal, constants are converted from off-line software and stored in read-only ROMs permanently. Variables are kept in read-and-write RAMs temporary. The single-direction and bi-direction data direction of ROM and RAM allows efficient update procedures for multiples elements of same type. In addition, Finite State Machine (FSM) assigned “starting” and “ending” time ranges to allocate proper sequence for different components. In addition to that, output signal for physical interface is also set up at specific time point.

In vertical, different component calculation blocks are able to solve independently and simultaneously, helping achieve a faster performance. For memory usage, permanent and temporary memory can be updated separately, reducing the probability of read-error and write-error [108]. Therefore, main FSM controller is able to dispatch memory units and calculation blocks more efficiently and quickly. Coupling horizontally and vertically, the parallel processing and resource sharing is kept at the maximum level for initialized EMT structure.



### 3.6 Case Study

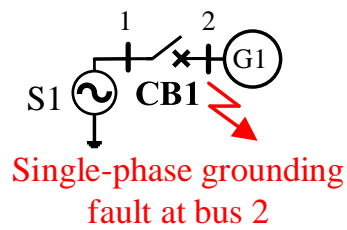
Consider different accumulation, the computational advantage of initialization methods is able to increase with system size. In order to observe the correction effect of initialization method, system scale needs to reach the critical point. There are two main questions remaining to be verified in case studies. The primary question is to investigate whether an initialization for single component is necessary. The secondary question is to compare the practical performances in small and large systems. Therefore, component-level and system-level case studies are both demonstrated accordingly.

For device-level initialization, a synchronous machine model is given to compare the timing constraints difference for Method 2, 3 and 4. And 11-bus system is provided to validate the effectiveness of proposed Method 4 for larger system. Timing, resource utilization and accuracy are considered for comparison. In particular, timing constraint has the higher priority than resource utilization as evaluation indicator owing to its significant impact on global routing. All existing simulation results are compared with referenced MATLAB model respectively.

The initialization application procedure consists of off-line solution and real-time operation. For off-line solution, steady-state initialization data are solved in MATLAB driven by CPU. For real-time operation, the proposed initialization methods for different models are all tested in single Virtex-6 FPGA ML605 Evaluation Kit. It is equipped with 241152 logic cells, 14976 memory, 768 DSP slices and 720 I/O ports. After linking the hardware modules with initialization solutions reasonably, different models can be initialized and executed to FPGA hardware.

### 3.6.1 Device-level Initialization

For individual component, a hypothetical topology, with single synchronous machine, is proposed in Fig. 3.8. S1 represents an ideal voltage source, representing an infinite-bus of the power grid. Circuit breaker CB1 is closed at time 0, connecting the synchronous machine G1 to the power grid. The starting-up procedure is mainly observed for initialization. This device-level component is to verify whether initialization is necessary for a single component and compare the practical performance of four initialization methods for a single component. A single-phase grounding fault is applied at node 1 from 15s to 15.1s. The simulation time is 20s.



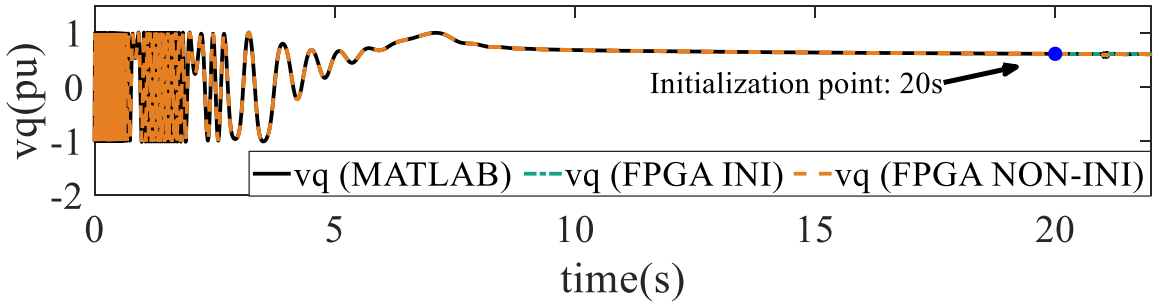
**Fig. 3.8 Synchronous machine simulation model**

**Table 3.7 Resource Comparison of Single SG**

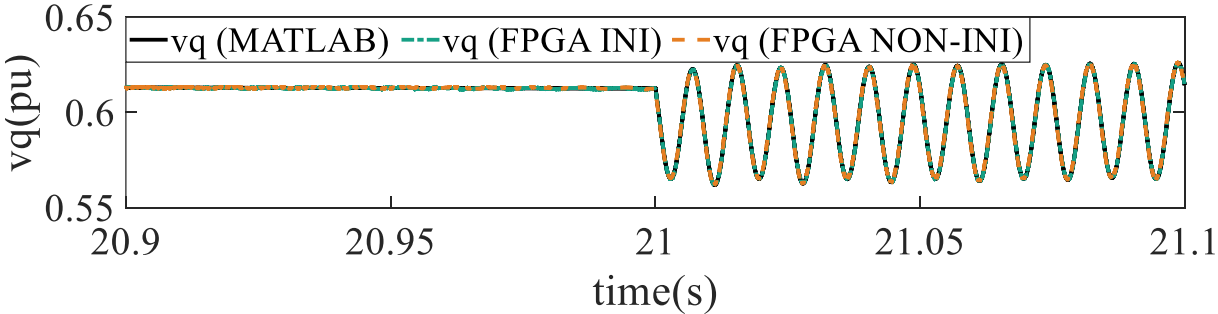
	registers	LUTs	RAM	DSP
Non-initialized	33%	52%	3%	48%
Initialized by Method 2	34%	58%	3%	51%
Initialized by Method 3	33%	60%	3%	51%
Initialized by Method 4	28%	55%	4%	51%

**Table 3.8 Timing Comparison of Single SG**

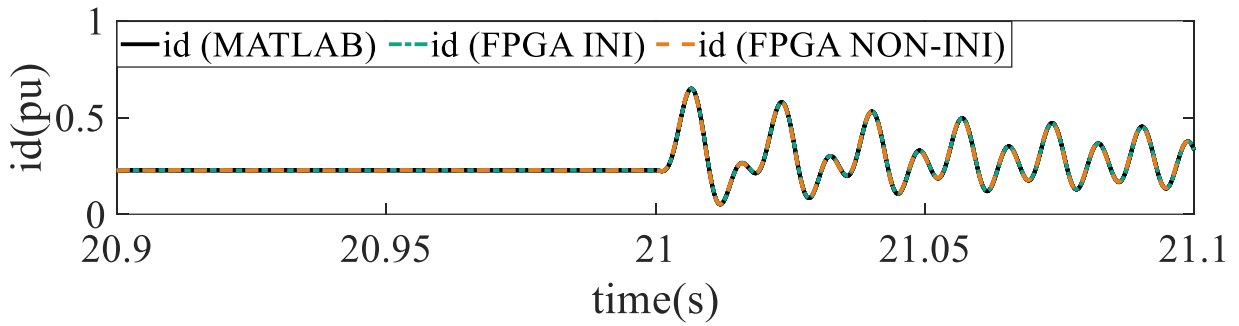
	Timing constraint	Initialization time
Non-initialized	9.414ns	0 Clock cycles
Initialized by Method 2	10.892ns	0 Clock cycles
Initialized by Method 3	9.985 ns	50 Clock cycles
Initialized by Method 4	9.414 ns	0 Clock cycles



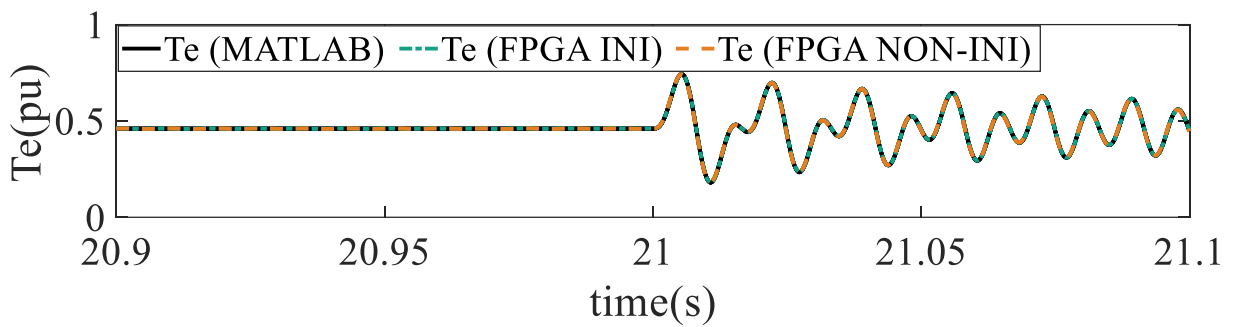
(a) Long-duration q axis voltage



(b) Q axis voltage



(c) D axis current



(d) Electric torque

**Fig. 3.9 Results of synchronous machine G1 with initialization and without initialization.**

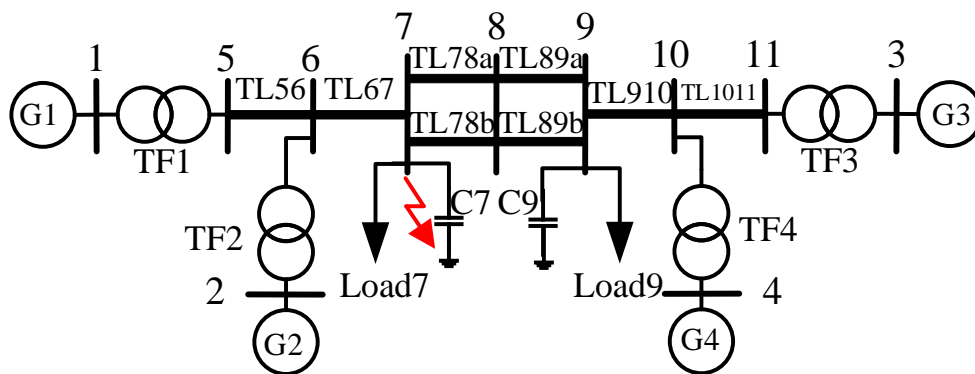
Fig. 3.9 illustrates the electric torque, voltage and current comparison with initialization schemes for single SG. Seen from Fig. 3.9, Method 2, 3, and 4 are all able to initialize with high accuracy, less than 4% error compared with referenced MATLAB model. With initialization, the relative error can be reduced from 4% to 2%. After successful initialization within the FPGA, the numerical accuracy is the same no matter which initialization scheme is used. Consider the slightly improvement, the effect of initialization is not significant for single device. This is the relative error have not been accumulated to a certain level to force amplitude and phase shift occur.

Table 3.7 and Table 3.8 provides the resource utilization and timing for different initialization schemes. Seen from Table 3.7, that Method 3 occupies longest real-time latency 50 clock cycles owing to sequential signal assignment. Method 2 has the longest timing constraint for 10.892 ns because of intensive signal declaration. By comparison, Method 4 using least timing constraint 9.414 ns and no extra latency can achieve the same initialization accuracy. Besides, Method 4 just occupies 3% LUT and 3% DSP, while saving 5% registers. Consider timing constraint has a huge impact on routing and translating, Method 4 with best timing constraint is selected as primary initialization Method.

Therefore, Method 4 shows the best performance in both accuracy and speed for initializing single SG. In order to explore whether Method 4 is extensible and initialization is necessary, it is essential to exploit potential application on 11-bus system in next section.

### **3.6.2 System-level Initialization**

For system-level initialization, a two-area 11-bus system is given in Fig. 3.10. This case study is to make sure whether initialization is essential for network and verify initialization Method 4 is scalable for network. A single-phase grounding fault is applied at node 7 from 15s to 15.1s at bus 7 for compare high-frequency transient behaviours.



Single-phase grounding fault at bus 7

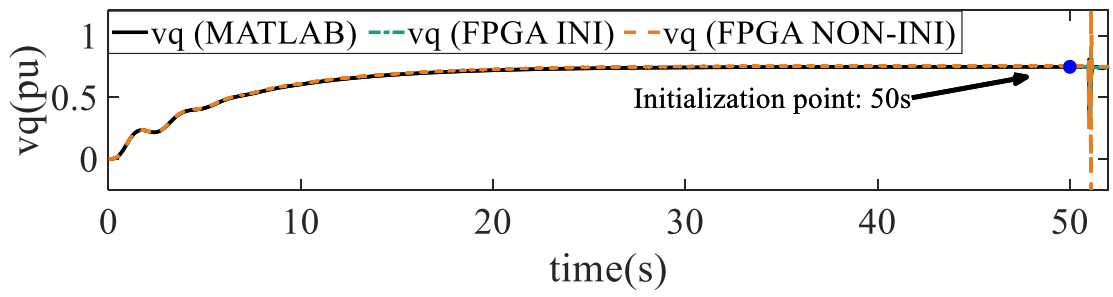
Fig. 3.10 4-machine 11-bus test system.

Table 3.9 Resource Comparison of 4-machine 11-bus Test System

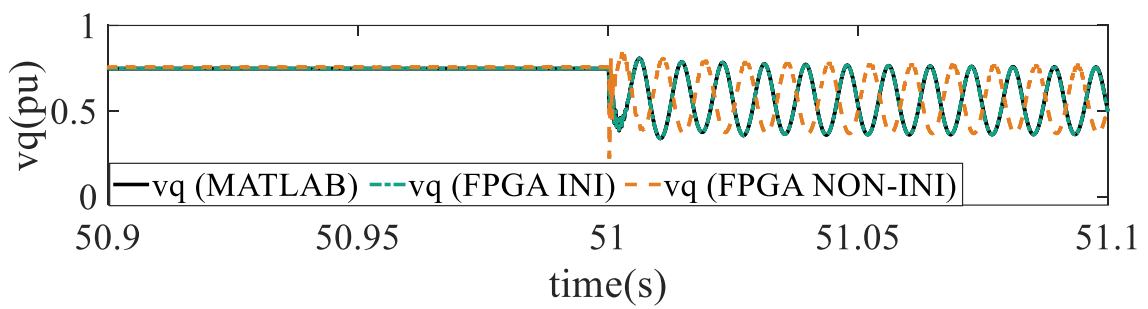
	registers	LUTs	RAM	DSP
Non-initialized	34%	78%	12%	91%
Initialized by Method 4	34%	80%	12%	92%

Table 3.10 Timing Comparison of Single SG

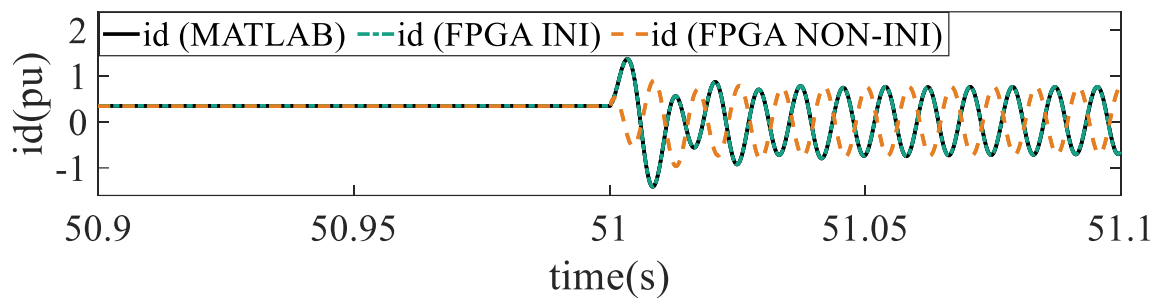
	Timing constraint	Initialization time
Non-initialized	9.975 ns	0 Clock cycles
Initialized by Method 4	9.980 ns	0 Clock cycles



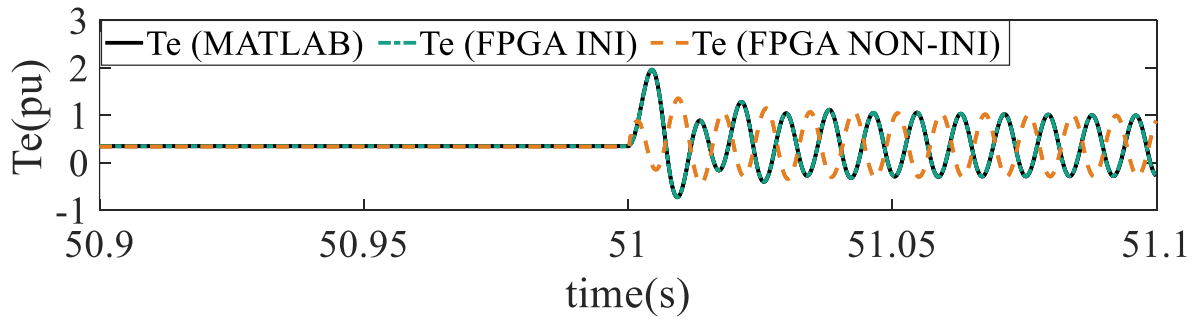
(a) Long-duration q axis voltage



(b) Q axis voltage



(c) D axis current



(d) electric torque

**Fig. 3.11 Results of synchronous machine G2 with initialization and without initialization.**

To analyse the effect of initialization on network accuracy, Fig. 3.11 provides detailed comparison between initialized FPGA model by Method 4, non-initialized FPGA model and referenced model. Seen from Fig. 3.11, initialization Method 4 is able to improve much higher accuracy than non-initialized model, reducing relative error efficiently to 5%. By initialization Method 4, significant error accumulated in non-initialized procedure owing to simulation phenomenon and distributed solver is almost eliminated. The powerful effect of Method 4 can effectively eliminate the accumulation error in amplitude and phase shift. Therefore, the long-duration accuracy of FPGA-based is still competitive with off-line software.

For hardware resource utilization, Table 3.9 and Table 3.10 gives related resource utilization and timing constraint comparison. Seen from Table 3.9, Method 4 only consumes additional 2% LUT and 1% DSP for initialization, maximizing resource sharing rate. Seen from Table 3.10, Method can fully initialize 11-bus system by just additional ns timing constraint without real-time operation time.

Therefore, this novel initialization Method 4 has dramatic computational advantage for initializing medium system and eliminating accumulated error.



## 3.7 Conclusion

This chapter has proposed four initialization methods for the first time. In addition, the initialization necessity and optimal initialization methods are fully explored for FPGA-based EMT. By comparing device-level and component-level simulation results, the following conclusions can be drawn:

- Unreasonable initialization method, including Method 2-3, can prevent original program from scaling up, which is owing to additional routes, resources and timing.
- On most occasions without specific accuracy requirement, device-level single SG is able to use uninitialized model. This is because uninitialized is already accurate enough with average error less than 5%. Initialization method can only limit the average error to 2%.
- For achieving same accuracy, Method 4 has the best initialization performance for minimum timing constraint, maximum calculation speed and minimum resource utilization. It is extensible for both device-level and component-level initialization.
- For system-level application, initialization is necessary for eliminating accumulation error of amplitude and phase shift. This is because the relative error of multiples of components will accumulate much faster than single device-level application. The computational advantage of Method 4 accumulates with system size and simulation time.

## **CHAPTER 4 MATLAB-TO-FPGA TOOLBOX**

This chapter firstly introduces the motivation for developing programming toolbox for FPGA using purely MATLAB programming environment in Section 4.1. For long-term sustainability, Section 4.2 introduces the overall development framework, design features and platform requirements. For flexible data conversion, Section 4.3 gives general data format, structure and partition. To utilize configurable logics, Section 4.4 develops different translation and reutilization process for sub-modules. To integrate low-level submodules, Section 4.5 presents how to formulate main controller using FSM. In Section 4.4, device-level and system-level case studies are both provided to verify the effectiveness of initialization and compare initialization performance. In Section 4.5, a 39-bus test system is provided to verify the effectiveness of this MATLAB-to-FPGA toolbox. In Section 4.6, the conclusions for FPGA-based initialization are summarized for this chapter.

## 4.1 Motivation

There are four main reasons to develop intelligent toolbox for FPGA-based EMT solutions:

- 1) The first reason is to reduce the programming complexity of FPGA. Compared with sequential programming, parallel programming is more complex to deal with. This is because the parallel programming of FPGA needs to take time schedule management, routing design, resource utilization estimation and numerical accuracy at the same time. These competing tasks are hard to be completed by new users.
- 2) The second reason is that existing translation software is unable to design EMT programs. This is because EMT program is complicated, optimisation in pipeline and parallel processing is required. Otherwise, straightforward translation will make translated code unable to implement on hardware owing to overused resource and timing constraint.
- 3) The third reason is that existing EMT toolboxes, such as PSCAD and EMTP, are commercial simulation packages. Due to the Intellectual Property issues, users can only use the existing library components on the menu of the platform. While internal calculation algorithms and detailed modelling are not transparent for users.
- 4) The fourth reason is to improve the scalability of existing programming. Pure FPGA programming is independent of network topology. Expanding topology will take a lot of repetitive efforts, such as data input, data conversion and programming, etc, which is a time-consuming procedure. Hence it is essential to propose a generic toolbox to program similar processes, for instance, RLC branches, in most efficient manner.

Therefore, there is an urgent need to develop a generic toolbox under user-friendly environments. This box can be utilized where it does not require a background of hardware, and this will lead to significant savings of efforts and time for beginners.

## 4.2 Development Framework

### 4.2.1 Overall framework

Pure hardware design is always time-consuming for beginners in terms of resource utilization, timing, and power efficiency. It is essential to make this procedure more intelligent. Fig. 4.1 presents the difference between original programming and hardware programming. As seen from that, an intelligent toolbox is expected to do generic tasks for different topologies. Users only need to input data for the intelligent toolbox.

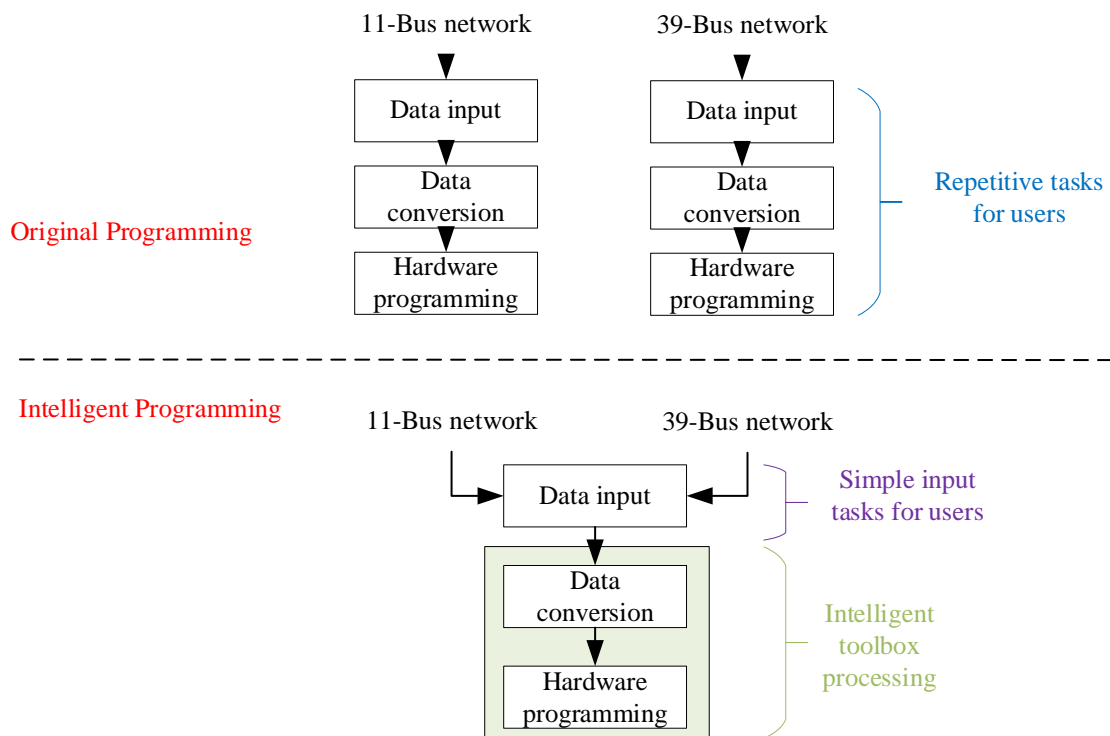
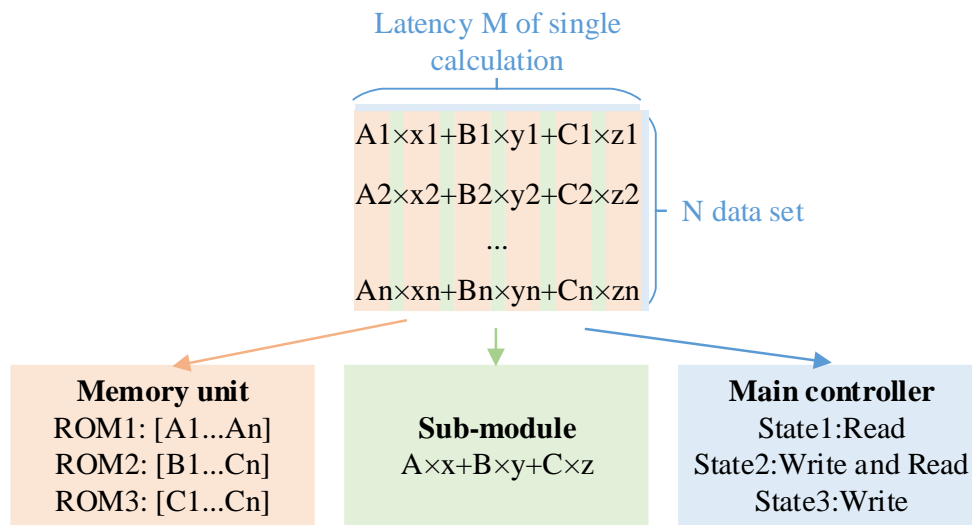


Fig. 4.1 Difference between intelligent and independent programming

MATLAB offers powerful built-in functions to accomplish diverse calculations, such as matrix inverse and rank. Moreover, MATLAB is able to read and write files with a variety of formats, such as .VHD, .COE and .OUT. More importantly, MATLAB is frequently used in engineering teaching and research, which is easy to be accepted by undergraduates and postgraduates. Therefore, MATLAB is used to pre-calculate so as to accelerate and simplify hardware. This thesis proposed a MATLAB-to-FPGA (MTF) toolbox to simplify FPGA-based EMT. This MTF toolbox can integrate the user-friendly MATLAB environment and real-time FPGA calculation.

This MTF toolbox can analyse and partition EMT models into three types of files, including memory unit, sub-module and main controller. For example, Fig. 4.2 presents the functionality of data formatter to calculate  $n$  RLC branch using (2-10). As seen from that, original  $n$  calculations can be partitioned into three parts: (a)  $[A1, \dots, An]$ ,  $[B1, \dots, Bn]$ ,  $[C1, \dots, Cn]$  are kept in ROMs. (b) 6-input multiply-adder is added in sub-module. (c) Data depth and latency are reflected in the read state, read-and-write state and write state in the main controller. Thus, there is no need to deal with RLC branch one by one. User's task is only at input stage instead of global design. Data formatter can significantly simplify repetitive calculations.



**Fig. 4.2 Data formatter for repetitive calculations**

To improve portability, different data can be extracted from a variety of sources: table files, databases, vectors and parameters. Moreover, the format of these data commands and various data declarations should be supported by software and hardware. Therefore, it is essential to allow users to define set and parameter that can be used to construct signal declaration, signal assignment and port map for FPGA.

To promote data management, the entire framework and dataflow of proposed toolbox is illustrated in Fig. 4.3. As shown in Fig. 4.3, the translated system will routinely run the data formatter to analyse and partition data sources. Input data will be partitioned into memory unit, sub-module, and main controller for different EMT components. Memory units, including RAM and ROM, are used to keep variables and constants. Sub-modules are generic calculations, which consist of basic IP Cores, such as add and multiply. Main controllers use different states of FSM to arrange time schedules for memory units and main modules. Users can run this MTF toolbox to obtain a variety of .COE, .VHD and IP files for hardware

operation. By combining these files, a variety of graphical and numerical results are able to be generated for further system operation and transient analysis.

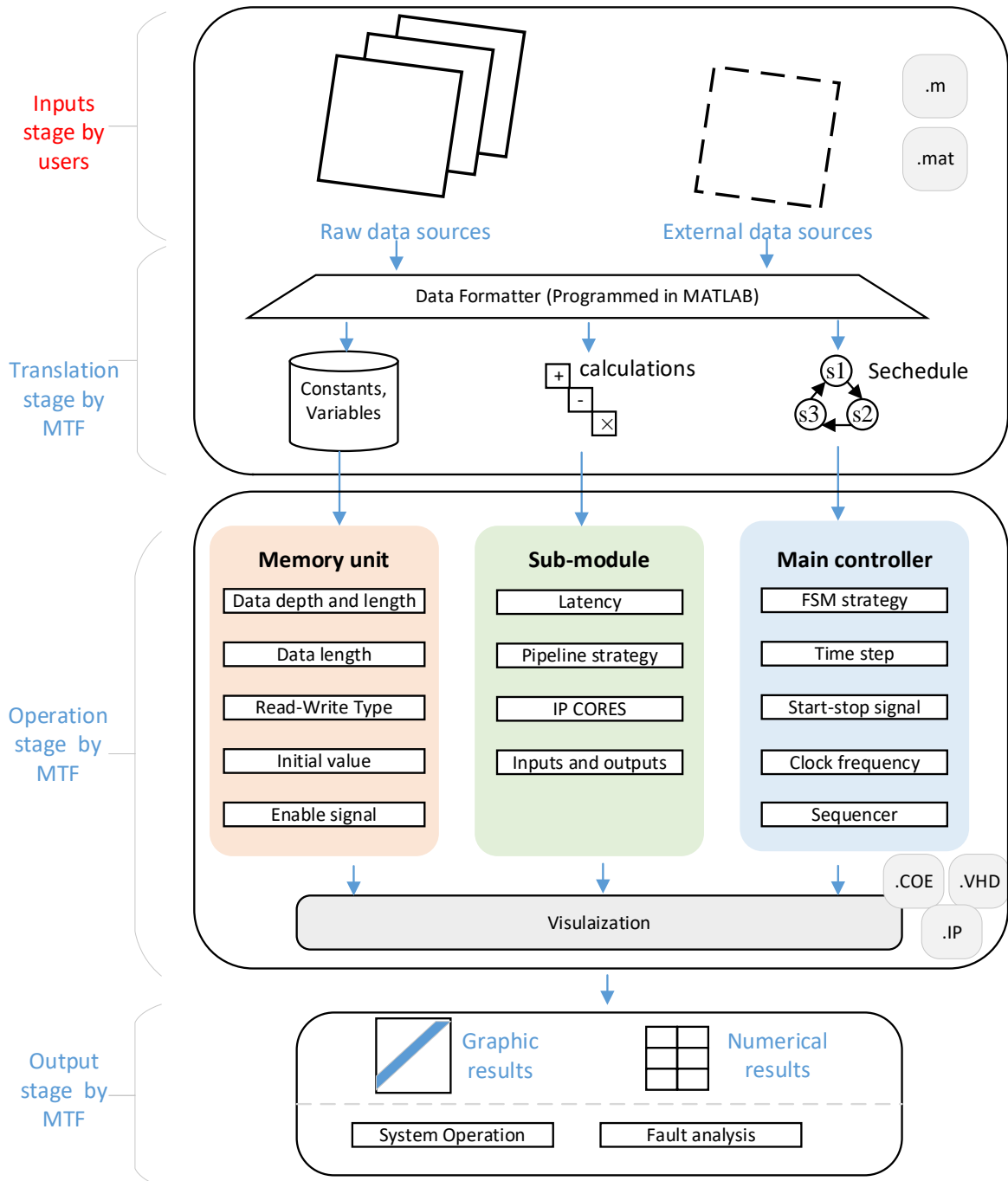
In addition to that, users are allowed to execute external data and user-defined models to FPGA, improving greater expandability. The whole system is still open to develop new functionalities. Without any background of hardware design, users can just follow the instructions to define arbitrary topology, releasing heavy hardware programming burden for users. Just read instructions can make a quick start example.

Pure FPGA programming has been replaced by data definition in MATLAB. Users only need to define data at input stage. High-performance structure is no longer limited for one component, which can be easily built up for similar calculations. The toolbox can be reutilized for different topologies. The repetitive and heavy programming burden is totally reduced.

The aim of MTF toolbox is to accelerate, simplify and scale up FPGA-based EMT system. The contribution of this toolbox can be summarized as follows:

- 1) First, MTF can achieve high-performance FPGA-based EMT design with much fewer lines of MATLAB codes compared with pure FPGA programming.
- 2) Second, MTF automatically analyses EMT models and picks up essential information into sub-modules, main controllers and memory units.
- 3) Third, MTF is capable of pre-calculating parameters, latencies and initialization, minimizing resource utilization and eliminating non-supported operations for FPGA.
- 4) Fourth, optimized designs for EMT components, based on hardware resources and timing constraints, can be generated and scaled up flexibly.
- 5) Finally, make data definition more readable, removing complicated binary pattern.





**Fig. 4.3 Global workflow for the proposed toolbox**

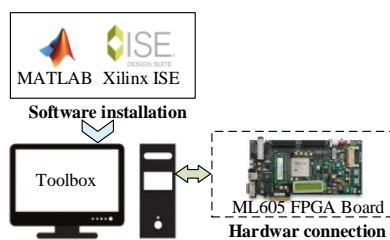
### 4.2.2 Design features

The key objective is to simplify FPGA design in user-friendly MATLAB environment. This is owing to the fact that high-quality FPGA design requires long design time and efforts to optimize. The competing factors, such as timing constraint, routing placement and resource sharing, are preventing bad-quality design from expansion. In order to avoid heavy computational burden, the high-level toolbox should involve the following features:

- 1) Broad data categories are involved. The data resources can be defined in .txt, .mat and .m files. Internal data type can come from common arithmetic, including decimal, hexadecimal, Single-Precision Floating-Point and so on.
- 2) Fully utilizing built-in functions of MATLAB. For data conversion, different data can be achieved flexibly by doing binary-to-decimal and Double-to-Single conversions. For data storage, the length and depth of memory units can be generated automatically.
- 3) Optimising design for pipeline and parallel processing in advance. The designed calculation structure is supported to multiples of extensions, such as involve more elements.
- 4) Automatic translation for simple blocks, including port map and time delay. For example, 4-input multiply-add block for frequent use should be generated automatically.
- 5) Scientific visualization. The observed output signal and type can be selected by users. Both graphical and numerical result is available for users.

### 4.2.3 Software-to-hardware Platform requirements

To ensure successful implementation of this toolbox, requirements for hardware and software should be both set up properly. The detailed setting up process is given in Fig. 4.4. For software, the host PC requires to be equipped with MATLAB and Xilinx Vivado or Xilinx ISE. For hardware connection, the PC should be connected with FPGA board via USB port. In addition, the detailed connection and installation for host PC should be given.



**Fig. 4.4 MATLAB-to-FPGA hardware platform.**

In host PC, data files are collected from MATLAB environment and downloaded into FPGA. In general, MATLAB is able to execute hardware designs and address files in purely MATLAB environment. By data file exchange, XILINX ISE and VIVADO provides low-level routines to integrate these files.

To implement the dataflow, these source files should be included:

- The input file for EMT system in TXT file, MAT file or M.file.
- The data conversion files that measures the size, depth and length of arbitrary input file.
- Calculation translation files that can directly count required IP cores, fill in required delays and detect pipeline strategies.
- Main controller design file with high expandability. If the same type elements are added, only constants for start-stop time point are required.

- General description file to FPGA interface in M-language. It declares block memory and the functions to execute hardware designs. It can manage arbitrary functions and files
- The bitstream files generated after translation by grouped modules. The netlist contains FPGA configuration is automatically mapped the design into FPGA using Xilinx tool.

### **4.3 Memory unit**

In order to obtain memory units acceptable for FPGA, constants are required to be preconditioned in advance. For user-friendly programming environment, a data file consisting of EMT system should be defined at first. The data is defined at purely MATLAB environment for flexible modification. These data are easy to be recognized for users in direct representation rather than binary representation. Even without the knowledge of hardware, beginners are still eligible to change the topology of EMT system. Therefore, this section describes general data format at first, detailed inputs for different EMT models, and data conversion for FPGA.

#### **4.3.1 Data Format**

For user-friendly data handling, MATLAB can accelerate computations. Therefore, the data format and arithmetic are implemented in MATLAB, completely using MATLAB programming language.

To allow flexible conversion, all the data are based on a default representation of Double-Precision Floating-Point. This help users to minimize truncation and rounding errors. Moreover, this also allows to flexible data conversion for other radices, including binary and hexadecimal.

All the data of same type are allowed to be defined in a n-dimension array, allowing intensive processing for further applications.

### 4.3.2 Data Structure

For quick processing, a n-dimension array is defined as a general structure. All type of information, such as connection types, connected buses and distances, are contained in the same array. This can minimize the amount of debug and maximize intensive processing. A general table is given in Table 4.1. The name vector contains all the name information, which is defined as cell array. As shown in Table 4.1, buses, decision variables and values are all defined in the same table. Although buses and decision variables are integer and zero-one integer, they are all represented in Double-Precision in MATLAB.

If these values are defined initially in VHDL, additional data format conversions are required, such as float-to-fixed conversion and fixed-to-integer conversion. Moreover, n-dimension matrix will have to be partitioned into one-dimension memory unit, making it more difficult to define consistent data. A general array in MATLAB can help promote computational efficiency by avoiding multiples of data conversion and one-dimension vectors.

**Table 4.1 General format of input data**

---

Data structure in MATLAB

---

```
Name={'BUS',DECISION_YES', 'VALUE'};

cell=[ 26, 1, 0.5;
       28, 0, 1.0;
       ...];
```

---

The size and dimension of array is dependent on the EMT component. The EMT models are based on the mathematic equations in Chapter 2. To form the entire EMT simulation system, the definition consists of bus and connected components. The connected components include RLC branch, transmission line, transformers, synchronous machines. All the components are optional for users, disabled component array will not influence the processing. By entering data in a new row, users can make new components, for instance, a synchronous machine, connected to existing network, enhancing scalability. Detailed parameters for different EMT models are provided as follows:

Synchronous machine is using the universal machine model in Chapter 2 [47]. Default parameters of SGs are using same constants [1], where rated power  $P_n$ , line-to-line voltage  $V_n$  and nominal frequency  $f_n$  are 555MW, 24kV, and 60 Hz, respectively. And each synchronous machine is equipped with an AC4A control system [1]. The location of the synchronous machine is determined by the connected bus and components. The general structure for defining the synchronous machine parameters is given in Fig. 4.5. Except for nominal values, other resistance and inductance are represented in per unit values. Depending on user's preference, parameters, such as nominal power, voltage and frequency, can be changed flexibly.

SG No.	BUS	Pn	Vn	fn	...
					...

**Fig. 4.5 Data structure for synchronous machine**

Ideal voltage source is to represent infinite external network at specific node. The parameters of ideal voltage, such as amplitude, frequency, and phase shift, can be defined in the general structure as shown in Fig. 4.6. The amplitude is in Volts, frequency is in Hz and phase shift is in degree.

VS No.	BUS	Amp	Phase	f

**Fig. 4.6 Data structure for ideal voltage source**

For RLC branches, locations are determined by connected start bus and connected end bus. If RLC branch is grounded, start bus or end bus will be the same as ground bus 0. Different combinations of RLC values are flexible for representing different components. For example, RL branch can be represented for simplified overhead lines and transformers when C is 0. In addition to that, RC branches can represent filters when L is 0. These combinations can be generally defined in detailed structure of a RLC branch in Fig. 4.7, avoiding repetitive tasks. RLC branch number, start bus and end bus are integer values, while R, L and C values are Floating-Point type.

RLC No.	R	L	C	Start_BUS	End_BUS

**Fig. 4.7 Data structure for RLC branch**

Transmission line is using distributed parameter model [45]. Its location is determined by connected buses, while the conductance is determined by its RLC values and distance. Lossless and loss type can be optional for users by changing R values. Data structure of transmission

line is detailed Fig. 4.8. Branch number, start bus and end bus are integer type, while distance and RLC are Floating-Point type.

TL No.	Distance	rlc	Start_BUS	End_BUS

**Fig. 4.8 Data structure for transmission line**

Apart from these components, buses, including ground bus and load bus, are defined in an integer table. This is to formulate the network completely at calculation stages.

### 4.3.3 Data Partition

Floating-Point parameters, such as distance and RLC values, are used to calculate the parameters, such as conductance. Therefore, these parameters can be pre-calculated as constants for memory unit.

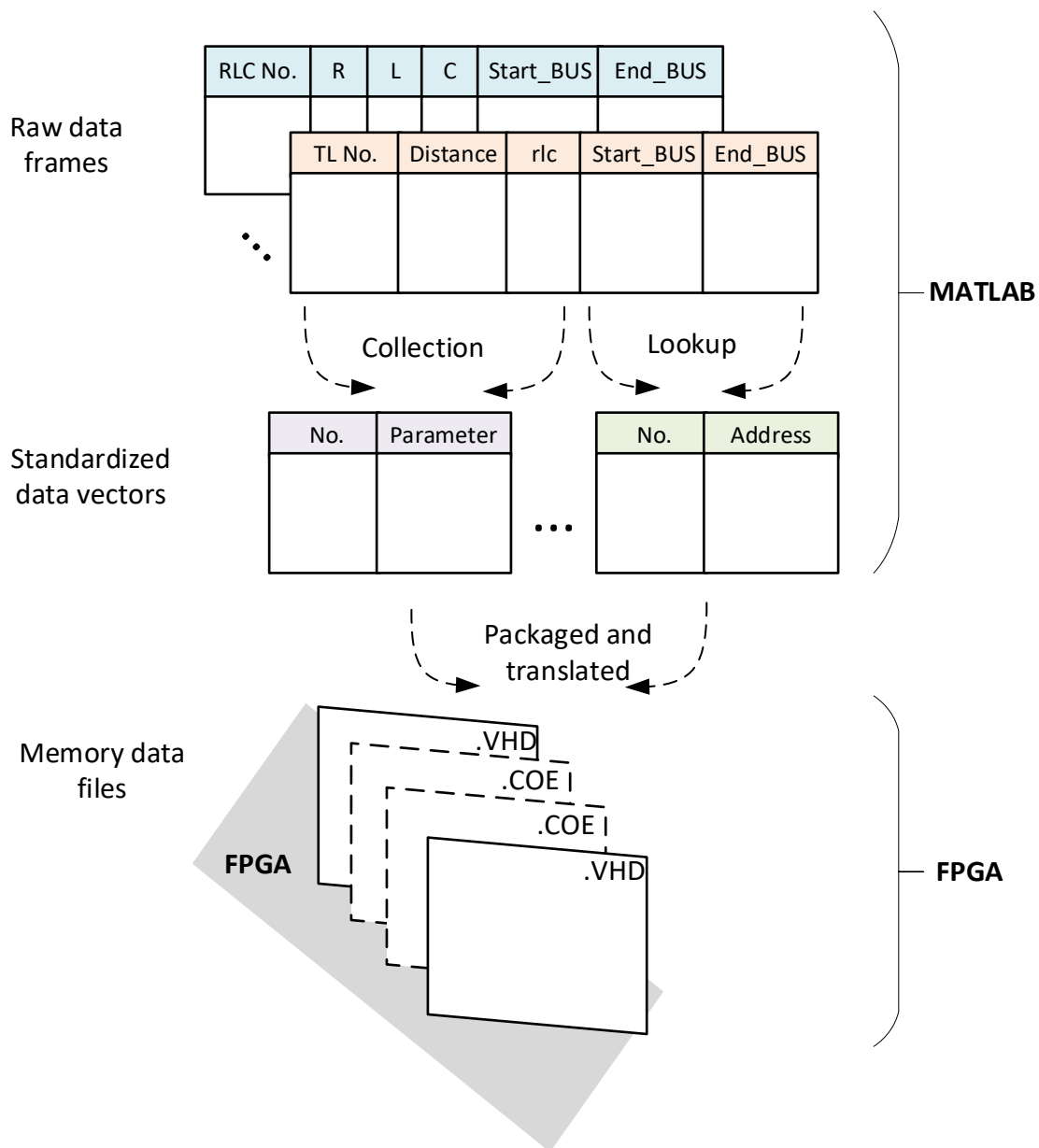
Users only need to define the basic parameters in default Floating-Point type, such distance and RLC values. MTF toolbox will pre-calculate these basic parameters into further intermediate parameters, such as conductance. Therefore, only intermediate parameters are kept in hardware memory units so as to improve hardware process speed and save memory space. For example, transmission lines just need to calculate the A, B, C, D and conductance in for hardware processing. After using in MATLAB, the required parameters can be converted into IEEE 754 Floating-Point. With additional declaration, these converted parameters can be kept in .VHD or .COE files for FPGA processing.

Integer parameters, such as start bus and end bus, are used to point to the exact physical address of the memory unit. This is because the values of the same type will be kept in one memory unit for hardware. The requirement for quick addressing is necessary. Although the currents



and voltages of branches always change, while the addresses of start buses and end buses remain unchanged. The unchanged memory addresses make it possible to adjust different calculations for high-frequency pipeline design. The address table uses *find* function to return the linear indices for different variables, such as start node. This can minimize the error for choosing the wrong value.

For hardware processing, more direct and simpler calculation is better to accommodate FPGA design, accelerating routing and translation. Parameters of any components and order can be searched in one clock cycle using address table. To achieve high-performance processing, it is necessary to use pipeline technologies to accelerate EMT calculations. Therefore, the parameters and search addresses are required to be standardized and suitable for pipeline strategies. The MTF toolbox can standardize any parameters and addresses automatically. This is implemented by picking up and locating the required parameters in the original data table. The standardization process is repetitive and general for individual components. The global structure for standardizing data from MATLAB array to hardware memory unit is described in Fig. 4.9.



**Fig. 4.9 Data standardization**

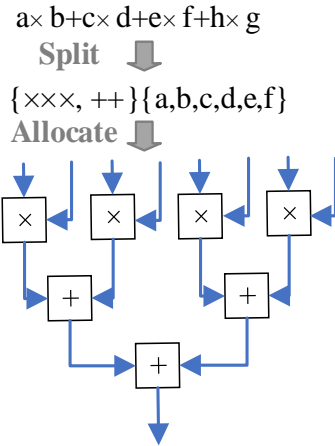
As seen from Fig. 4.9, various data can be converted into .VHD and .COE file in spite of data length and depth. Lack of intelligent functions, this procedure is unable to be achieved in pure FPGA programming environment. With the help of basic MATLAB functions, this procedure is programmed in MTF toolbox. By standard data structure, as shown in Fig. 4.9, essential information is kept in fixed columns for batch processing. Parameters, such as distance and

RLC values, can be picked up and precalculated as conductance in parameter table. Addresses, such as connected bus, can be picked up and numbered in address table for high-frequency search. Because plenty of parameters and addresses for whole calculation, pre-calculated data, obtained in MTF toolbox, can be used in hardware processing for millions of times. Therefore, this data standardization can help greatly reduce resource utilization and improve processing speed. Hardware computation and programming burden is efficiently reduced by MTF toolbox.

## **4.4 Sub-module**

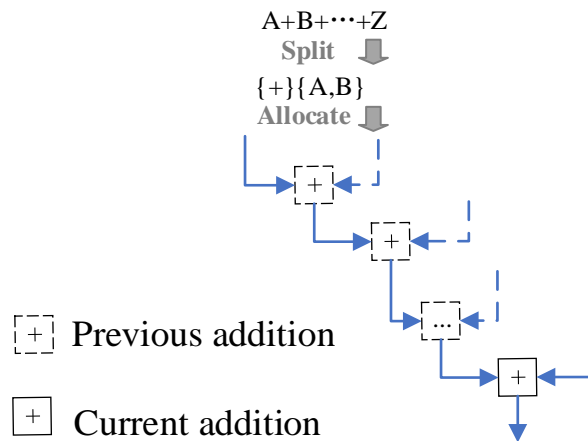
### **4.4.1 Translation**

For simple calculation, such as 4-input multiply-add block, the sub-module can be processed directly by splitting string. These simple calculations are well suited for pipeline strategies with no overlap. The detailed calculation code is treated as a string in MATLAB. Using *split* function for string can pick up the ports and IP COREs. Additional registers can be added to set up delays for synchronize whole data procedure. For example, Fig. 4.10 shows that 4-input multiply-add block requires 4 multipliers and 3 adders. In the same latency, all the calculators are used for once. This can help to solve asynchronous process for different modules.



**Fig. 4.10 Simple pipelined translation**

For complex calculations, the FSM controller can be applied to translate complex procedure. The FSM sub-module is not suitable for pipeline strategy. Different from simple calculations, the first step is to determine the IP CORE types, because each calculator will be reutilized for saving resources. Then, *split* function will determine the exact sequence and times for a single calculator. For example, a 32-input adder is using one multiplier to finish 32 repetitive additions. Fig. 4.11 details the non-pipelined translation for 32-input addition. The previous result is the input for next addition, the inputs of adder will be reallocated every latency. Except for these adders, other inputs will interrupt the whole calculation. Therefore, this translation is not supported for pipeline processing. Wating time will be put as different states for FSM. By trade-off for longer latency, complex structure can be implantable by reusing calculators.



**Fig. 4.11 Complex non-pipelined translation**

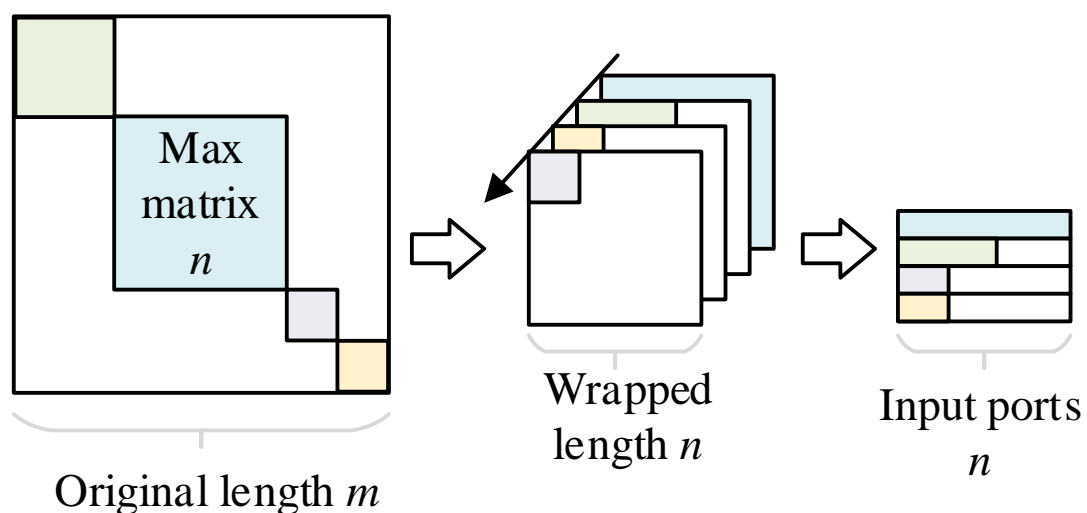
#### **4.4.2 Reutilization and simplification**

For reducing computation burden, some similar blocks are available for reutilization. For different size, 2-input multiply-add block can utilize 3-input multiply-add block by leaving zero for unnecessary ports. For different signs, adder and subtractor can use the same adder. Therefore, it is necessary to explore that possible steps can be reutilized in the entire EMT calculations.

Unlike MATLAB, matrix multiplication is not built in VHDL. It is necessary to build up entire matrix calculation based on matrix elements. To simplify matrix calculation, it is essential to find out whether a large matrix is full rank. Consider an EMT model is connected with multi transmission lines, the admittance matrix is sparse and symmetric. This motivates us to partition original matrix to several small matrices. Detailed matrix partition and maximized structure is given in Fig. 4.12. Therefore, the matrix calculations have been replaced by smaller vector calculations so as to minimize input dimension.

After partitioning a matrix into vectors, it is essential to be further reutilize calculation blocks. Fig. 4.13 gives the detailed reutilization strategy for different vectors. Every line in the same matrix is able to use the same calculation module. The sub-module calculation block is determined by the maximum length. For any vector of smaller size, unessential ports can be left zero. Fig. 4.13 shows the explicit hardware design for 8-input vector multiplier. Seen from Fig. 4.13, the processing vectors for 8-input and 4-input multipliers can be  $[A_{11}; x_1; A_{12}; x_2; A_{13}; x_3; A_{14}; x_4; ]$  and  $[C_{11}; y_1; C_{12}; y_2; 0; 0; 0; 0; ]$ , respectively, using the same module. Moreover, this design uses pipeline strategies to accelerate the whole matrix calculations.

Therefore, previous  $m \times m$  matrix calculations have been reduced into  $n \times 1$  vector calculations. This design can not only help improve calculation efficiency but also synchronize data processing.



**Fig. 4.12 Matrix decomposition structure.**

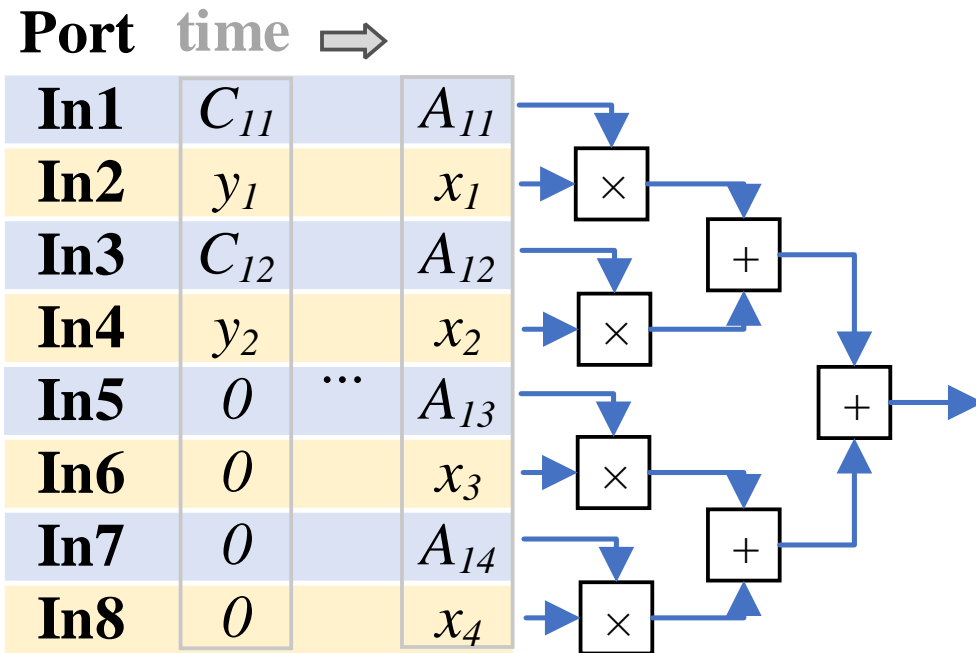


Fig. 4.13 Hardware design for 8-input vector multiplication.

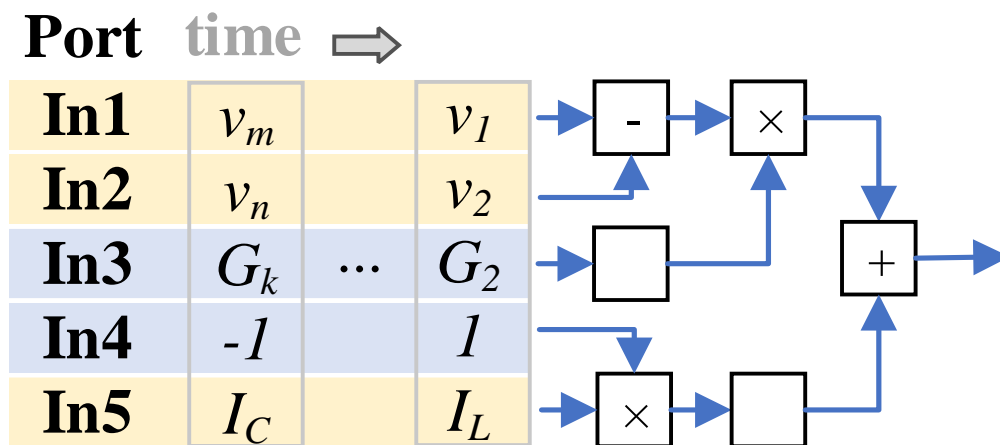


Fig. 4.14 Hardware connection of RLC branch current.

Apart from size, different calculations can also use the same block by reverting signs. For example, the main calculation difference between  $L$  and  $C$  is the positive sign and negative sign

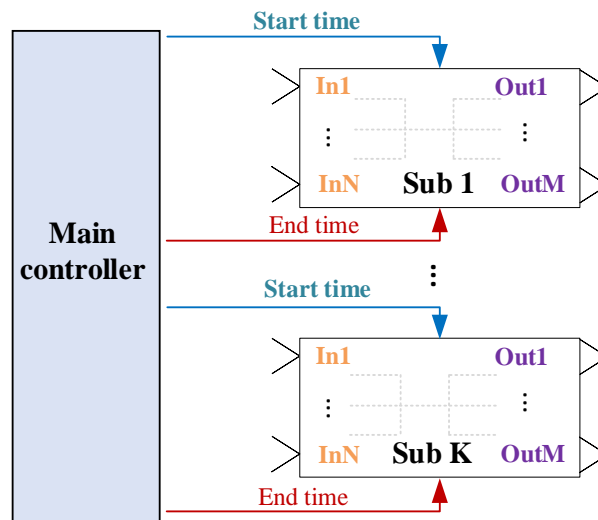
of history current. If the sign of history current is pre-stored to multiplied constants, then all types of RLC branches can use the same calculation block. For detailed sign conversion, Fig. 4.14 provides the detailed of hardware connection for *RLC* branch. As illustrated, the processing vectors for *C* branch and *L* branch can be  $[v_m; v_n; G_k; -1; I_c; ]$  and  $[v_1; v_2; G_2; -1; I_L; ]$ , respectively, using the same module. In this way, real-time computation can be simplified further by reverting sign.



## 4.5 Main controller

### 4.5.1 FSM controller structure

The main controller is to set up the start and stop signal to control the processing for different submodules. The objective is to allow the main controller be generalized for any size network. To maximize calculation flexibility, a general controller structure is developed in M-files to generate real-time VHD modules. Therefore, essential structure is provided in Fig. 4.15. At a high-level, main controller set up start and end time to control the operation of sub modules. At a low-level, sub-module builds up input-to-output routing for different calculations, for instance, for a transmission line. This general controller structure allows users to involve further calculation stages using similar structures. In this way, the universality can be improved to a new level despite of calculation amount.



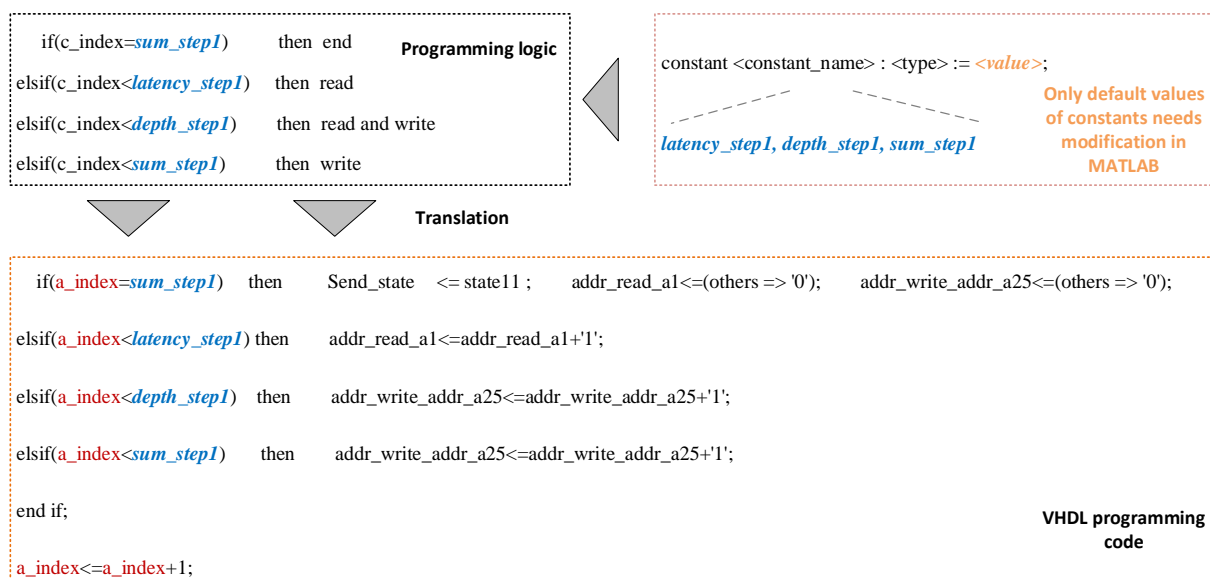
**Fig. 4.15 Essential translation structure for main controller and sub-modules.**

With the involvement of pipeline strategies, the stop-end control needs to be modified adaptively. This is because non-pipelined calculations just require start-end signal for each

state. Pipelined structures need to consider the data depth owing to intensive calculations at one stage. Compared to non-pipelined structure, the start signal needs to be further divided as read-start-signal, write-start-signal, read-end-signal and write-end-signal.

### 4.5.2 Pipelined stage

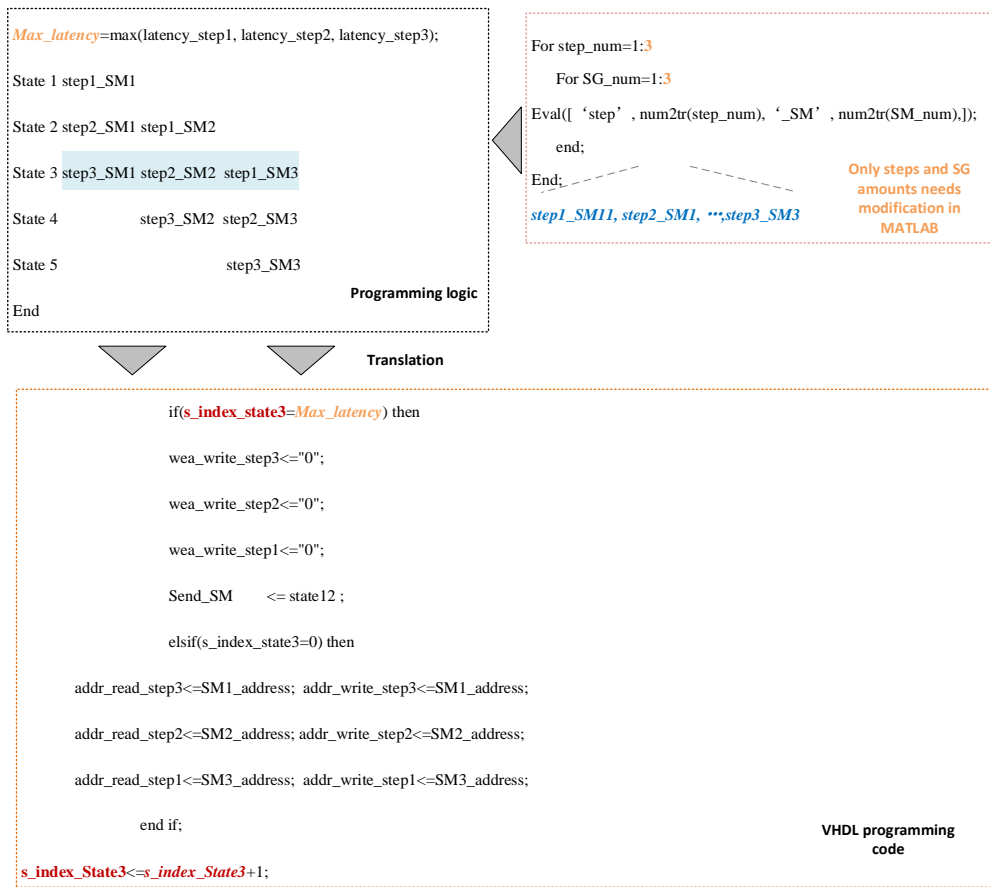
For pipelined module, the control is determined by the data depth at each stage. For correct data write and read, the latency and data depth are all defined as constants at signal declaration stage. To maximize flexibility of pipeline technology, read-write time of different pipeline stages is able to be adjusted adaptively for a scaling-up system. Fig. 4.16 gives the detailed programming logic for a pipelined design. For the scaling-up system, only constant time parameters, such as *sum\_step3*, *depth\_step3* and *latency\_step3*, need to be modified in constant declaration stage. The controller original design can stay the same. The corresponding read-write time is determined by data depth and sub-module latency. By built-in function *size*, the reading and writing time can be tracked with network scale automatically.



**Fig. 4.16 Programming logic for pipelined stage**

### **4.5.3 Non-pipelined stage**

For non-pipelined modules, main controller needs to make sure that the module inputs will not start next calculation unless previous calculation is finished. This is because non-pipelined structure reuses the same calculator for one time calculation. When using the same module, it is essential to put calculations in a sequential time slot. For different modules, it is possible to put next calculation for free modules. By processing different modules at the same time slot, the required states and entire calculation time can be significantly reduced. If we are going to calculate four synchronous machines, each synchronous machines occupies 3 states. Fig. 4.17 provides the programming logic for non-pipelined stage. Without parallel processing, original design need 9 FSM states and 9 latencies. With the help of parallel processing, only 5 states and 5 latencies are required. By parallelizing different modules, the calculation efficiency can be highly improved.



**Fig. 4.17 Programming logic for non-pipelined stage**

## 4.5 Case study

In the following experiments, the basic calculations in FPGA are implemented by Xilinx Floating-Point Operator, which is full compliance with IEEE-754 standard Floating-Point values. The Xilinx ISE Tool is utilized for synthesizing VHD-files generated from MATLAB. To validate the effectiveness of proposed toolbox, the 11-bus system and 39-bus system are simulated using this toolbox. The calculation time and number of generated files are detailed in Table 4.2. The generated VHD files are displayed in Fig. 4.18. The topologies of 11-bus and 39-bus are displayed in Fig. 3.10 and Fig. 4.19.

**Table 4.2 Generated files details**

Network	VHD source file generation time	Number of VHDL files
11-bus	50s	89
39-bus	300s	89

6step0_1e4_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_2e5_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_3e6_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_4e7_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_5e8_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_6e9_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm11_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm12_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm13_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm21_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm22_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm23_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm31_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm32_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB
6step0_SG_ram_64_8_rm33_vec_64bits	09/09/2022 14:01	Hard Disk Image F...	2 KB

#### Fig. 4.18 Generated VHD files by MATLAB-to-FPGA toolbox

Seen from Table 4.3, the file generation time varies from the system size, where the 39-bus network needs longer time than 11-bus network. This is because 39-bus network needs more information to define topology. But the total number of the files are the same for the 11-bus and 39-bus. This is because once the same sub-module can be utilized repetitively as long as it is involved. The number of files always stays the same. The system size will just change the data depth and length in each system.

The comparison of the resource utilization of the 11-bus system and 39-bus system is given in Table 4.3. As shown, the 39-bus network just costs more 2% LUT, 8% registers and 2% DSP than 11-bus, which is around 3-time of the expansion of the 11-bus. This verifies the benefit of resource sharing of the proposed method. This translation benefits improve along with the increase in the system size.

**Table 4.3 Resource utilization between 11-bus and 39-bus**

	registers	LUTs	RAM	DSP
11-bus	34%	78%	12%	91%
39-bus	36%	86%	19%	93%

Consider the transients behaviours are already analysed in Chapter 2 and Chapter 3. This chapter is mainly focused on the transient behaviours of the 39-bus network in Fig. 4.19. A

single-phase grounding fault is applied at bus 6 from 1.0s to 1.1s. G1 is the observed for transient results. Fig. 4.20 provides detailed voltage, current, electric torque for 39-bus system.

Seen results shown in Fig. 4.20, the generated results are highly matched with referenced MATLAB model. The relative errors of voltage, current and electric torque are less than 5%. Even fault is applied from 1.0s to 1.1s, the dynamic behaviours still highly matched with the referenced model. Moreover, the generated FPGA model can automatically settle down to steady state once the fault is removed. These high-performance accuracy and resource utilization have validated the reliability and flexibility of the proposed toolbox.

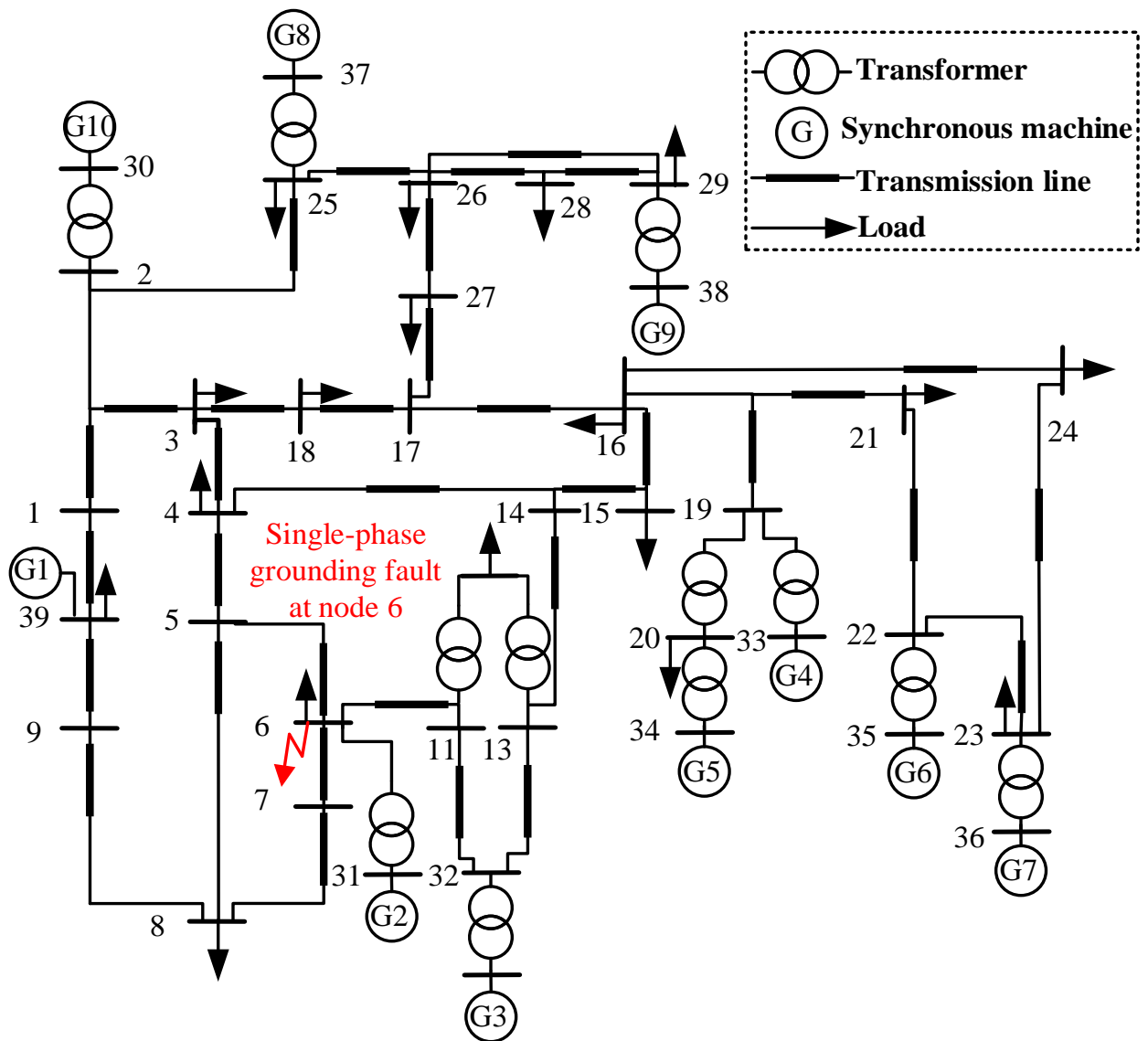
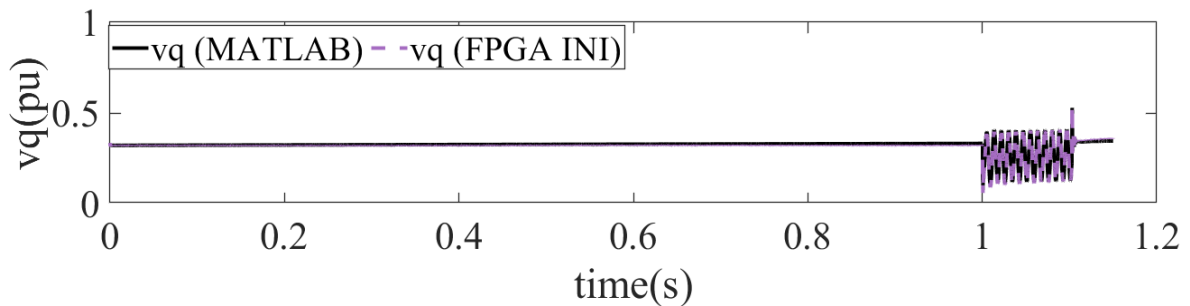
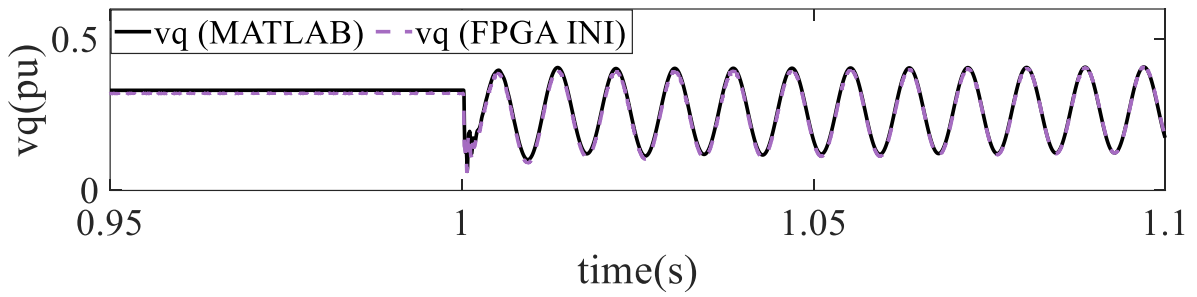


Fig. 4.19 10-machine 39-bus test system.

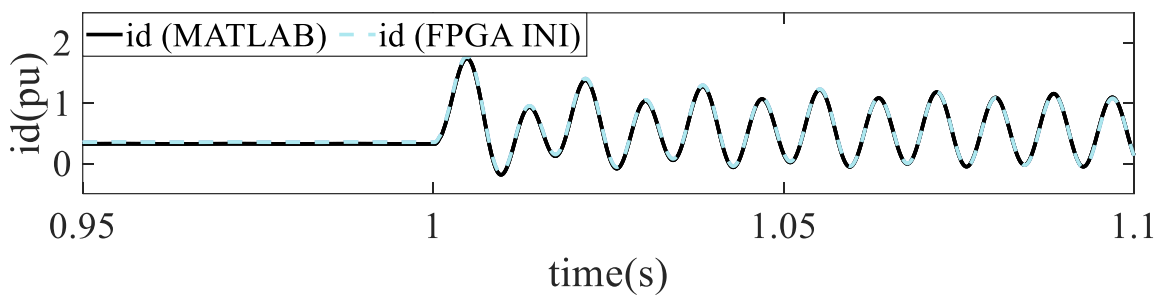


(a) Long duration q axis voltage

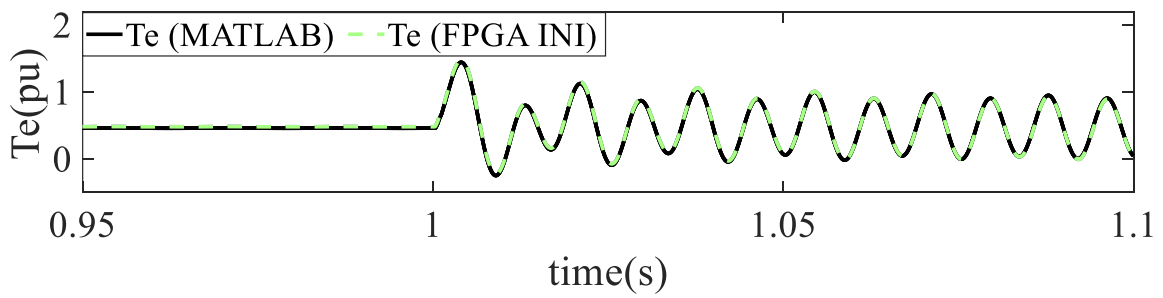




(b) q axis voltage

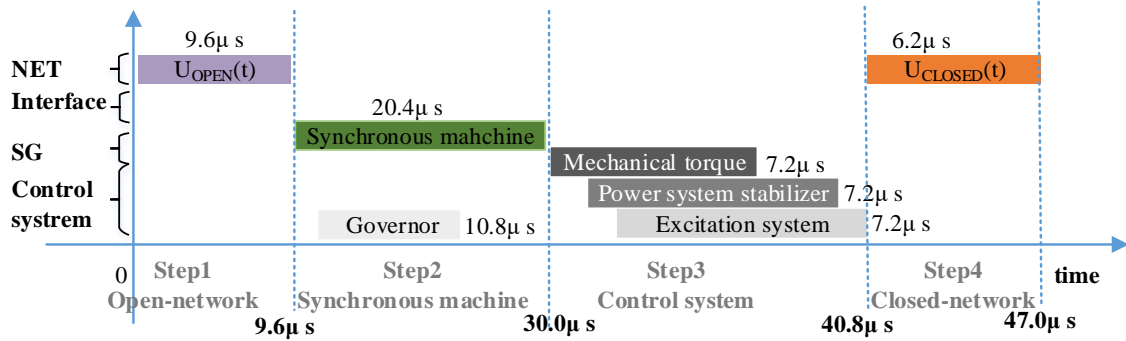


(c) d axis current



(d) Electric torque

**Fig. 4.20 Synchronous machine G1 with/without initialization.**



**Fig. 4.21 Time schedule of 39-bus system.**

Seen from, the time schedule of THE 39-bus system cost 47.0 $\mu$ s at the simulation step 50 $\mu$ s. With the help of optimized FSMs and pipelines, calculating the EMT of the 39-bus system is able to be finished in one simulation time step. This means that the FPGA model for the 39-bus system is real-time implementable.

## 4.6 Conclusion

The chapter has proposed a novel MATLAB-to-FPGA toolbox for FPGA-based EMT for the first time. The design objective is to allow the users even unfamiliar with hardware able to generate high-performance for FPGA-based EMT solutions. The significant benefit lies on the fully utilization of MATLAB built-in functions to finish all programming, which releases the heavy burden of FPGA programming. The global framework is divided into detailed data definition, sub-module calculation and main module calculation. By standardization, memory units can improve memory utilization efficiency and accommodate pipeline designs. Submodules, using reutilization and simplification, can simplify original complex calculations for hardware acceleration. Main controller, using FSM strategies, can arrange time schedule for both pipelined and non-pipelined designs automatically, the EMT dynamics of the 39-

network can be calculated within real-time  $47\mu\text{s}$ , which is less than one step of  $50\mu\text{s}$ . This means that the FPGA model for the 39-bus system is real-time implementable.

The intelligent toolbox can obtain essential files for the 39-bus network and 11-bus network using 300s and 50s, respectively. Case studies, including the 11-bus system and 39-bus system, also have verified that the MTF toolbox is accurate (with less than 5% relative error), resource-savings and scalable. In the future, this toolbox can be applied to test new models in other large benchmark systems.

# **CHAPTER 5 FPGA-Based Power Electronic Devices and Control System Simulation**

This chapter firstly introduces the motivation for high-frequency electronic devices and control system in Section 5.1. For transient behaviours, Section 5.2 gives the mathematical models for electronic devices, including IGBT, Diode and MMCs. For efficient control actions, Section 5.3 details control system for grid-side converters into sub-blocks, such as variable mean and PID control. To reduce implementation complexity, Section 5.4 gives optimized strategies to implement real-time EMT models in Section 5.2 and 5.3 in hardware. For practical application, Section 5.5 provides PWM control and HVDC-MMC case study. Finally, Section 5.6 summarizes the major work and gives conclusion.

## **5.1 Motivation**

Chapter 2-4 are focused on dynamic behaviours of power devices. The topologies changes are led by the actions of circuit breakers. In practical operations, the switching frequency of power electronic devices, such as IGBT (more than 1kHz), which is much faster than circuit breakers. This chapter is mainly focused on implementation of power electronic devices and control system. There are two main reasons for this research work proposed here:

- 1) The first reason is that whether multi-level HVDC-MMC is possible to be implemented on a single FPGA board. Existing research work [109], using 2 PB5 board, is expensive to implement. A single FPGA board can be an alternative method for cheaper cost.

- 2) The second reason is that high-frequency switching is always challenging to be modelled and simulated for FPGA-based applications. The operation of power electronics is much faster than traditional circuit breakers. It is very challenging to switch multiple topologies driven by a clock frequency of 100Mhz. Especially for HVDC-MMC, the internal half-bridge MMC is influencing the operation of the other arms. This will make the implementation of HVDC-MMC system more difficult.

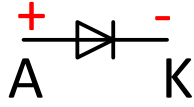
Therefore, it is necessary to provide cheaper and flexible solutions for the HVDC-MMC system using FPGA board.

## 5.2 Power Electronic Device Modelling

### 5.2.1 Diode

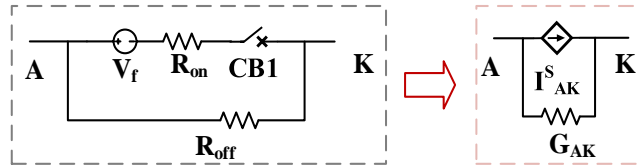
Different converters are composed of semiconductors, such as IGBT and diode. To formulate converters efficiently, single models of diode, IGBT and MMCs need to be built up at first. Compared to uncontrollable diodes, IGBT and MMC can be controlled by external signals.

Diode is a semiconductor component, which can only be controlled by internal voltage and current. For practical experiment, the difficulty of diodes lies on the nonlinearity and single direction flowing. Fig. 5.1 gives the detailed diode structure, which can only conduct from node A to node K. It is controlled by its own current  $I_{AK}$  and voltage  $V_{AK}$ . When  $V_{AK} > 0$  or  $I_{AK} > 0$ , then the diode is forward biased with a small forward voltage  $V_f$ . When  $I_{AK} \leq 0$ , then the diode will reverse biased and turn off.



**Fig. 5.1 Simple diode structure**

For switching operation, the diode block can normally be represented by a series resistor, an inductor, a forward voltage source and a switch. For the sake of simplifications, the leakage current in the blocking state and the reverse-recovery current are not considered. For the discrete-time modelling, inductor  $L_{on}$  is forced to zero owing to the simulation time step. Therefore, Fig. 5.2 shows diode, composed of a resistor, a forward voltage source and a switch, can be represented by Norton circuit.



**Fig. 5.2 Equivalent circuit for diode**

According to Norton's Theorem and the Norton Equivalent Circuit, the corresponding resistor and current source can be replaced as:

$$G_{AK} = \begin{cases} G_{on} + G_{off}, & v_{AK} > 0 \text{ or } i_{AK} > 0 \\ G_{off}, & \text{else} \end{cases} \quad (5-1)$$

$$I_{AK}^S = \begin{cases} G_{on} \cdot V_f, & v_{AK} > 0 \text{ or } i_{AK} > 0 \\ 0, & \text{else} \end{cases} \quad (5-2)$$

where  $R_{off}$  and  $G_{off}$  are the resistance and conductance of snubber, respectively,  $R_{on}$  and  $G_{on}$  are the resistance and conductance of forward biased resistor, respectively,  $V_f$  is the forward

voltage,  $v_{AK}$  and  $i_{AK}$  are the voltage and current, , respectively, between the diodes,  $I_{AK}^S$  and  $G_{AK}$  are the equivalent current source and conductance for the diode, , respectively.

## 5.2.2 IGBT

IGBT is a semiconductor can be controlled by external gate signal. The turn-on and turn-off switch is determined by the combination of internal state and external gate signals. When the gate signal is positive and the voltage is over forward voltage, the IGBT turns on. Otherwise, it stays turn-off. Similar to a diode, the equivalent circuit for an IGBT can be represented as follows:

$$G_{AK} = \begin{cases} G_{on} + G_{off}, (g > 0, v_{AK} > 0) \text{ or } (g > 0, i_{AK} > 0) \\ G_{off}, \text{ else} \end{cases} \quad (5-3)$$

$$I_{AK}^S = \begin{cases} G_{on} \cdot V_f, (g > 0, v_{AK} > 0) \text{ or } (g > 0, i_{AK} > 0) \\ 0, \text{ else} \end{cases} \quad (5-4)$$

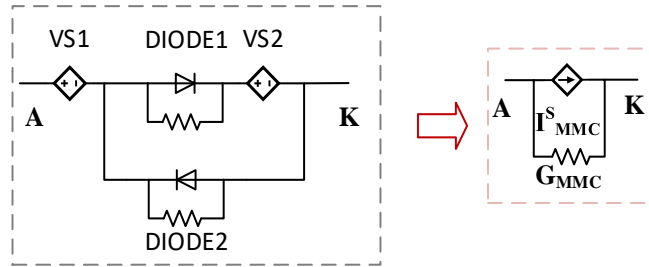
where  $g$  is the external gate signal.

## 5.2.3 Half-Bridge MMC

Half bridge Modular multilevel converter normally includes multiples of series-connected power modules. In each power module, there are diode/IGBT pairs and one capacitor on the DC side. Control signal can be given as inputs to determine the turn-on and turn-off state for each power module. For simplification, an aggregated model is chosen to use the equivalent circuit to represent all the power modules. More importantly, the aggregated model can run much faster than the detailed model, and the former is more suitable for real-time simulation.

To simplify MMC modelling, Fig. 5.3 gives the detailed circuit for the aggregated model [37] of the half-bridge MMC. The diode  $d1$  is connected with a controlled voltage source  $VS2$  in

parallel with reversed d2, and in series with the voltage-controlled voltage source VS1. Because of MMC is applied at amplitude of kV, diode 1 and 2 can be able to be replaced as a variable resistor.



**Fig. 5.3 Aggregate model for MMC**

Based on trapezoidal rule, the conductance is determined by a variable resistance and a simulation time step. Because only one diode can be derived at the same step, the reversed conductance is the same despite of switching state. According to Norton circuit theory, the relationship between open-circuit voltage and current can be summarized as follows:

$$(i - G_{D1} \cdot VS_2) \cdot (G_{D1} + G_{D2}) + VS_1 = v \quad (5-5)$$

$$i \cdot (G_{D1} + G_{D2}) + (VS_1 - G_{D1} \cdot (G_{D1} + G_{D2}) \cdot VS_2) = v \quad (5-6)$$

Therefore, the open-circuit voltage and current of half bridge MMC can be given as:

$$G_{MMC} = G_{D1} + G_{D2} \quad (5-7)$$

$$I_{MMC}^S = VS_1 - G_{D1} \cdot (G_{D1} + G_{D2}) \cdot VS_2 \quad (5-8)$$



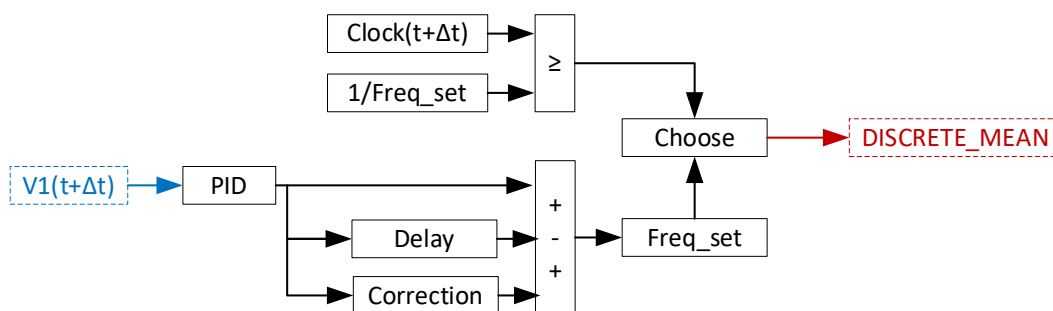
where  $G_{D1}$  and  $G_{D2}$  are the conductance for two diodes, respectively,  $I_{MMC}^S$  and  $G_{MMC}$  are the equivalent current source and conductance for the diode, respectively,  $V_{S1}$  and  $V_{S2}$  are the voltage sources determined by external signals.

### 5.3 Control System Modelling

Different control systems, such as Phase Lock Loop (PLL), are implemented by fundamental blocks. Therefore, modelling of fundamental control block is necessary, including variable mean block, discrete mean block, current control loop and PWM control.

#### 5.3.1 Discrete mean value

Discrete mean block is to calculate mean value after a fixed time delay [110]. Once the time delay is set, it will continue for the whole simulation duration. Therefore, a fixed range of memory space is required to delay input signals.



**Fig. 5.4 Discrete mean control block**

The mean block is able to get the mean value of the inputs over a running window at the specified frequency. Fig. 5.4 provides the detailed control connections for the discrete mean block. For the first cycle, the output will be kept at initial constant value. For the rest cycles, the output will be updated consistently. Mean block can be considered as the combination of integrator, delay unit, subtractor and switch.

$$T_{set} = \text{round}\left(\frac{F_{fun}}{\Delta t}\right) \quad (5-9)$$

where  $T_{set}$  is the time delay step,  $F_{fun}$  is the specified delay frequency.

Based on trapezoidal rule, integrator can be formed as:

$$V_1(t + \Delta t) = V_1(t) + \frac{\Delta t}{2} (V_{dc}(t) + V_{dc}(t + \Delta t)) \quad (5-10)$$

where  $V_{dc}(t)$  is the input direct voltage,  $V_1(t)$  is the integrated signal.

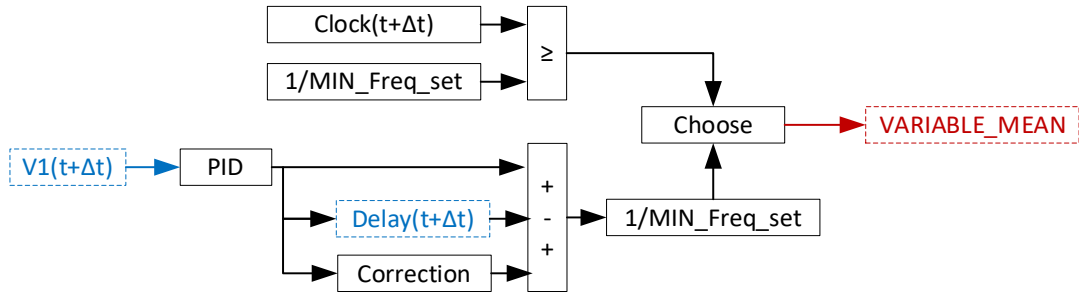
Conditional output is determined at the current time step:

$$V_2(t + \Delta t) = \begin{cases} V_{dc}^{nom}, & \text{if } t < T_{set} \cdot \Delta t \\ F_{fun} \cdot (V_1(t + \Delta t) - V_1(t + \Delta t - T_{set} \cdot \Delta t)), & \text{else} \end{cases} \quad (5-11)$$

where  $V_{dc}^{nom}$  is the nominal direct voltage,  $V_2(t)$  is the mean DC voltage over running cycle.

### 5.3.2 Variable mean value

Variable mean block can help to calculate the mean value over a running window with a variable delayed frequency [110]. A maximum frequency will be pre-defined to decide the maximum buffer size in memory area. Variable delay can be input for the whole simulation duration. Fig. 5.5 details control block for variable mean value.



**Fig. 5.5 Variable mean control block**

At the first step, integrator based on trapezoidal rule is applied.

$$V_1(t + \Delta t) = V_1(t) + \frac{\Delta t}{2} (V_q(t) + V_q(t + \Delta t)) \quad (5-12)$$

where  $V_q(t)$  is the per unit q axis voltage,  $V_1(t)$  is the integrated signal.

At the second step, the input frequency is limited within the upper limit and lower limit.

$$F_{freq}(t + \Delta t) = \begin{cases} F_{MAX}^{PLL}, & \text{if } Freq(t + \Delta t) > F_{MAX}^{PLL} \\ F_{MIN}^{PLL}, & \text{if } Freq(t + \Delta t) < F_{MIN}^{PLL} \\ Freq(t + \Delta t), & \text{else} \end{cases} \quad (5-13)$$

where  $F_{MAX}^{PLL}$  and  $F_{MIN}^{PLL}$  are the upper limit and lower limit of input frequency, respectively,  $Freq(t + \Delta t)$  is the input frequency,  $F_{freq}(t + \Delta t)$  is the limited frequency.

At the third step, a correction block is applied for q axis voltage.

$$N_0(t + \Delta t) = 1/(F_{freq}(t + \Delta t) \cdot T_s) \quad (5-14)$$

$$F_0(t + \Delta t) = \text{ceil}(N_0(t + \Delta t)) \quad (5-15)$$

where  $N_0(t + \Delta t)$  is the number of samples per cycle,  $T_s$  is the sample time step,  $F_0(t + \Delta t)$  is the integer sample time step.

At the fourth step, a variable delay unit is applied to delay input signal at a specified delay input. When the input delay is not integer sample time step, linear interpolation is utilized. The maximum frequency can help to determine the buffer size for storage. For integer samples, the output can be defined as follows:

$$F_1(t + \Delta t) = \begin{cases} V_2^{ini}, & \text{if } t < F_0(t + \Delta t) \\ V_1(t + \Delta t - F_0^{MAX} \cdot \Delta t), & \text{if } F_0(t + \Delta t) > F_0^{MAX} \\ V_1(t + \Delta t - F_0(t + \Delta t) \cdot \Delta t), & \text{else} \end{cases} \quad (5-16)$$

where  $V_2(t + \Delta t)$  is the output after variable time delay,  $V_2^{ini}$  is the pre-set initial condition value,  $F_0^{MAX}$  is the allowed maximum time delay.

$$F_2(t + \Delta t) = (V_1(t + \Delta t) - F_1(t + \Delta t)) \cdot F_{freq}(t + \Delta t) \quad (5-17)$$

where  $F_2(t + \Delta t)$  is the mean value.

To eliminate dynamic error, the mean value will be corrected by a correction system:

$$C_0(t + \Delta t) = \frac{N_0(t+\Delta t) - F_0(t+\Delta t)}{N_0(t+\Delta t)} \cdot \left( \frac{(V_q(t) - V_q(t+\Delta t))(N_0(t+\Delta t) - F_0(t+\Delta t))}{2} + V_q(t + \Delta t) \right) \quad (5-18)$$

where  $C_0(t + \Delta t)$  is the corrected factor for mean value.

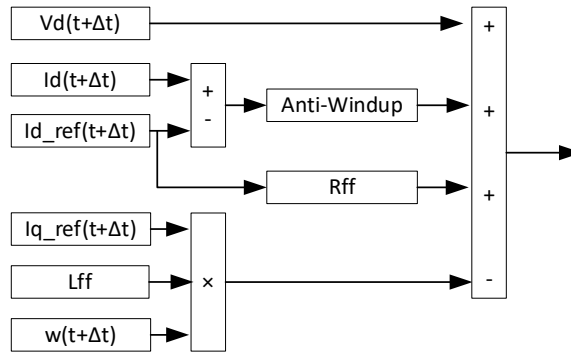
To avoid the initial oscillation, the variable time delay is only allowed to generate external inputs from the initial set period. The final output can be given as follows:

$$V_3(t + \Delta t) = \begin{cases} V_{INI}, & \text{if } t < T_{INI} \\ F_2(t + \Delta t) + C_0(t + \Delta t), & \text{else} \end{cases} \quad (5-19)$$

where  $V_3(t + \Delta t)$  is the final output of variable delay unit,  $V_{INI}$  is the block initial condition,  $T_{INI}$  is the initial frequency.

### 5.3.3 Current control loop

The current control loop, voltage control loop and power control loop are similar for HVDC-MMC. The same strategy relies on using PID control and limiter to make the measured value close to the referenced value as much as possible. Therefore, only d current control loop is introduced, other control loop can be built up similarly. Fig. 5.6 provides d axis current control loop.



**Fig. 5.6 D axis current control block**

At the first step, d axis current error is computed by measured current minus referenced current at the same time step.

$$I_1(t + \Delta t) = I_d^{ref}(t + \Delta t) - I_d(t + \Delta t) \quad (5-20)$$

where  $I_d^{ref}(t + \Delta t)$  is the referenced d axis current,  $I_d(t + \Delta t)$  is the measured d axis current,  $I_1(t + \Delta t)$  is the absolute error.

At the second step, d axis current error will be accumulated by different constants.

$$I_2(t + \Delta t) = K_p^{Ireg} \cdot I_1(t + \Delta t) \quad (5-21)$$

$$I_3(t + \Delta t) = I_3(t) + K_i^{Ireg} \cdot \frac{\Delta t}{2} \cdot (I_1(t + \Delta t) + I_1(t)) \quad (5-22)$$

where  $I_2(t + \Delta t)$  and  $I_3(t + \Delta t)$  are intermediate signals with different gain,  $K_p^{Ireg}$  and  $K_i^{Ireg}$  are constant gains, respectively.

At the third step, Anti-windup block can be treated as the combination of integrator and limiter. Moreover, anti-windup block can also keep integrated values within specified levels. When the integral is between the upper bound and lower bound, the integrator will turn on. If the integral is greater than the upper bound and input is positive, the saturation will be held at the upper bound. If the integral is lower than the lower bound and input is negative, the saturation will be held at the lower bound.

$$I_4(t + \Delta t) = I_5(t) + I_3(t + \Delta t) \cdot \Delta t \quad (5-23)$$

$$I_5(t + \Delta t) = \begin{cases} I_{MAX}^{Ireg}, & \text{if } I_4(t + \Delta t) > I_{MAX}^{Ireg} \text{ and } I_3(t + \Delta t) > 0 \\ I_{MIN}^{Ireg}, & \text{if } I_4(t + \Delta t) < I_{MIN}^{Ireg} \text{ and } I_3(t + \Delta t) < 0 \\ I_4(t + \Delta t), & \text{else} \end{cases} \quad (5-24)$$

where  $I_4(t + \Delta t)$  is the integral signal,  $I_5(t + \Delta t)$  is the output after saturation,  $I_{MAX}^{Ireg}$  and  $I_{MIN}^{Ireg}$  are the upper bound and lower bound of the current regulator, respectively.

At fourth step, an adder is applied for adding the integral output with the previous gained output.

$$I_6(t + \Delta t) = I_3(t + \Delta t) + I_5(t + \Delta t) \quad (5-25)$$

At the fifth step, a limiter is applied to make the output within the specified limit.

$$I_7(t + \Delta t) = \begin{cases} I_{MAX}^{Ireg}, & \text{if } I_6(t + \Delta t) > I_{MAX}^{Ireg} \\ I_{MIN}^{Ireg}, & \text{if } I_6(t + \Delta t) < I_{MIN}^{Ireg} \\ I_6(t + \Delta t), & \text{else} \end{cases} \quad (5-26)$$

where  $I_7(t + \Delta t)$  is the output after certain limit.

At the final step, an adder is use to integrate d axis voltage, integral current and rotating q axis current as proportion to the control referenced PWM signal.

$$I_8(t + \Delta t) = V_d(t + \Delta t) + R_{ff} \cdot I_7(t + \Delta t) + L_{ff} \cdot I_q(t + \Delta t) \cdot w(t + \Delta t) \quad (5-27)$$

where  $I_8(t + \Delta t)$  is the output of d axis control loop,  $R_{ff}$  and  $L_{ff}$  are constant gain values, respectively,  $I_q(t + \Delta t)$  is the measured q axis current,  $w(t + \Delta t)$  is the angular speed based on frequency.

### 5.3.4 PWM control

PWM control is using referenced signal to compare with carrier signal to generate pulses [110] - [111]. A fixed shift degree will be generated between different bridges. The time shift can be obtained using following equation:

$$Offset(k) = \frac{(k-1)}{n_{level}} \cdot ST_{PWM} \quad (5-28)$$

where  $Offset(k)$  is the offset time for bridge  $k$ ,  $ST_{PWM}$  is a shift time constant,  $n_{level}$  is the number of bridges (power modules) of the multilevel converter.

Time accumulation effect for different bridges can be generated as follows:

$$P_1(k, t + \Delta t) = (Clock(t + \Delta t) + Offset(k)) \cdot F_c \quad (5-29)$$

$$P_2(k, t + \Delta t) = P_1(k, t + \Delta t) - fix(P_1(k, t + \Delta t)) \quad (5-30)$$

where  $P_1(k, t + \Delta t)$  and  $P_2(k, t + \Delta t)$  are intermediate PWM signal,  $F_c$  is carrier frequency.

The interpolation for the sequential carrier can be generated as follows:

$$C_r(k, t + \Delta t) = \begin{cases} 1 - |4 P_2(k, t + \Delta t) - 2|, & \text{if } 0 \leq P_2(k, t + \Delta t) \leq 1 \\ -1, & \text{else} \end{cases} \quad (5-31)$$

where  $C_r(k, t + \Delta t)$  is the carrier signal.

$$P_3(k, t + \Delta t) = \begin{cases} 1, & \text{if } C_r(k, t + \Delta t) > U_{ref}(t + \Delta t) \\ 0, & \text{else} \end{cases} \quad (5-32)$$

$$P_4(t + \Delta t) = \sum P_3(k, t + \Delta t) \quad (5-33)$$

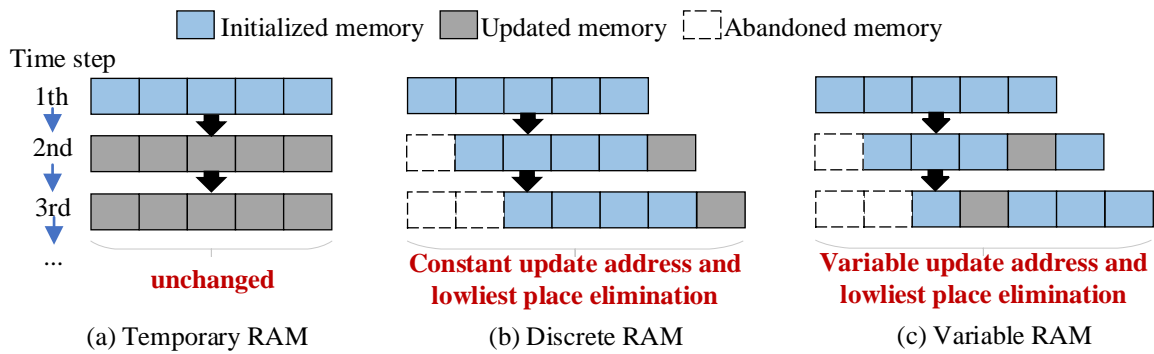
where  $P_3(k, t + \Delta t)$  is the single PWM signal for bridge k,  $P_4(t + \Delta t)$  is the total PWM signal for the whole arm.

## 5.4 Hardware Implementation Strategies

### 5.4.1 Shift memory unit

FPGA memory space is relatively limited for massive data. To keep necessary data for a long duration, it is desirable to develop efficient memory utilization for massive data. Take different dynamics, memory strategies are required to be optimized adaptively. The memory blocks can be divided into three different types. Temporary type is updated completely at every time step, for instance, RLC branch. Discrete type is updated proportionally with a constant address at every time step, such as discrete time mean value. Variable type is updated proportionally with a variable address, such as variable mean value. Fig. 5.7 shows the update mode of three different RAMs in details.



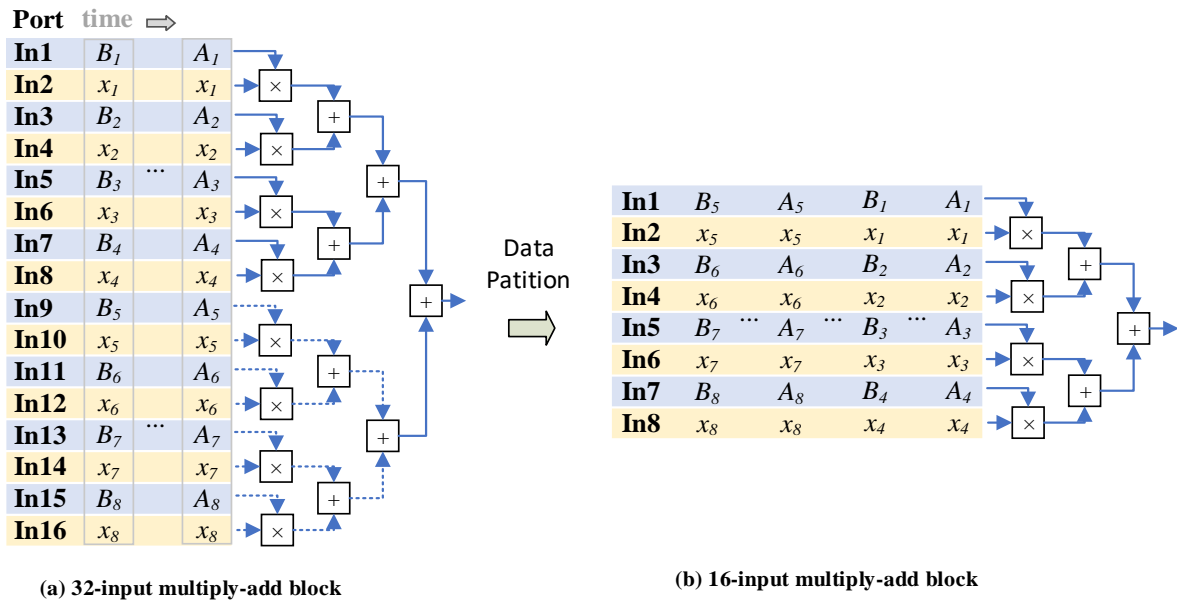


**Fig. 5.7 Update modes for three RAM types**

As shown from Fig. 5.7, temporary and discrete RAM are using constant addresses for whole simulation period. By comparison, variable RAM is using different address for writing data. Moreover, the depth of three types of RAMs is all constant. Thus, the memory utilization will stay the same instead of increasing along with simulation duration. This can help to avoid overused memory and improve memory utilization efficiency.

### 5.4.2 Partitioned pipeline design

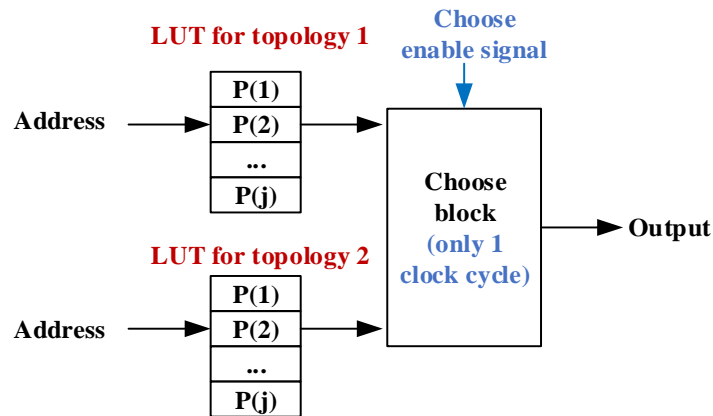
Consider the network with converter is complicated with more electric nodes, the calculation burden will grow. Especially for a large admittance matrix, ports and computation units will be even doubled. To avoid high-fanout routing, large calculations can be partitioned into small pipelined designs. The only thing to do is to use a longer input address and waiting latency. For example, 32-input multiply-add block in Fig. 5.8 will cost 32 multipliers and 31 adders in the original design. By partition, it can be built up by a 16-input multiply-add block, which only costs 16 multipliers and 15 adders. Shows the partition process with the pipelined designs. Therefore, resource utilization is improving efficiently.



**Fig. 5.8 Data partition for 32-input multiply-add block**

### 5.4.3 Switch block

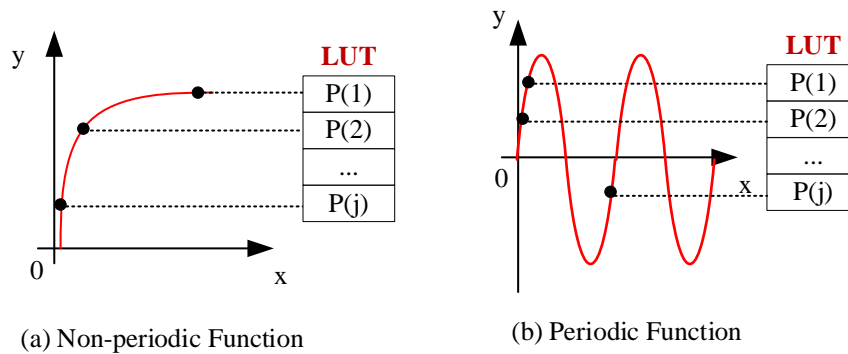
To solve the changing topologies, the possible topologies need to be pre-calculated in advance. LUT switch design is implemented in Fig. 5.9 for frequent topologies changes. Therefore, in real-time calculation, the topologies change will only cost 1 clock cycle and will not increase additional time. In this way, different topologies can switch very fast and suitable for real-time design. This design can help to place single-phase fault, three-phase fault and other circuit breaker actions. Provides detailed switch hardware design when switching between different topologies.



**Fig. 5.9 Switch design for different topologies**

### 5.4.4 Interpolation

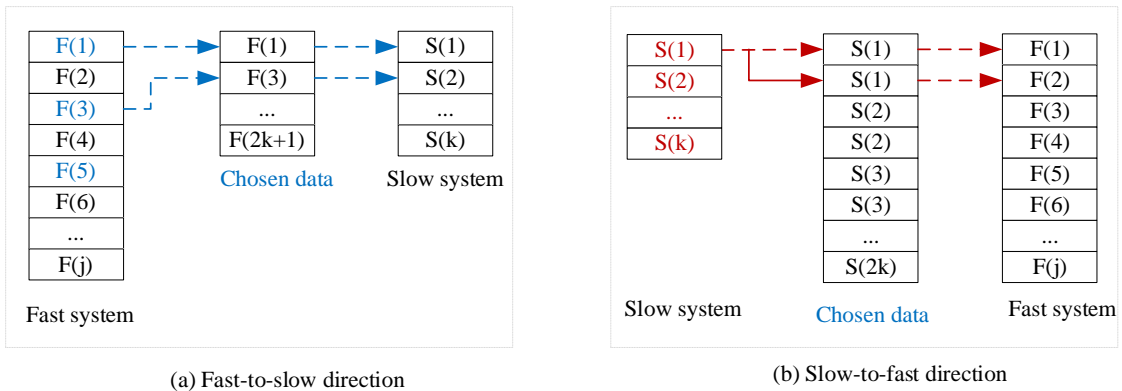
FPGA are only equipped with basic calculations, such as adder and multiplier. Exponent and sine function are not involved in IP COREs. To address the problem in real-time, interpolation in Fig. 5.10. For periodic function, such as sine wave function, discrete points of one cycle memory are kept for the whole duration of the simulation. For non-periodic function, such as exponent function, a limited range is predicted for the whole duration of the simulation. Once the input is given, the search address can be generated automatically. An address function and memory LUT can interpolate the function in real-time, which gives detailed interpolation process for periodic function and non-periodic function. The main difference is that the memory depth for non-periodic function needs to be predicted in advance. To avoid overflow of data, the upper bound and lower bound needs to be determined for the whole duration of the simulation.



**Fig. 5.10 Interpolation for periodic and non-periodic function**

### 5.4.5 Different time step interface

When the control system is extremely complex, the calculations need longer time step than electric systems. In order to transfer data efficiently, data interface should be built up between sub systems using different time steps. Fig. 5.11 shows the detailed data transfer directions for different system speed. The transfer directions are composed of fast-to-slow direction and slow-to fast direction. For example, the time step for a fast system and a slow system are 20 us and 40 us, respectively. For the fast-to-slow direction, the slow system will only allow the original fast data every two time steps. For the slow-to-fast direction, the data from the slow system will be extended to the last two time steps for the fast system. Therefore, efficient waiting and processing can help to couple the fast system and slow system effectively. This strategy can be even utilized for coupling different energy systems, such as fast electric system and slow thermal system.

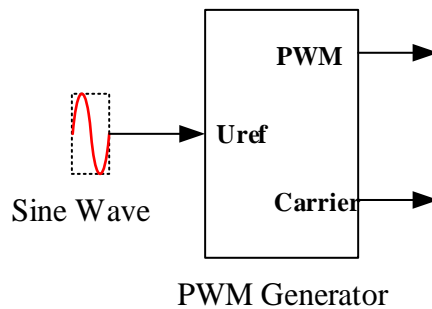


**Fig. 5.11 Transfer directions between different speed data**

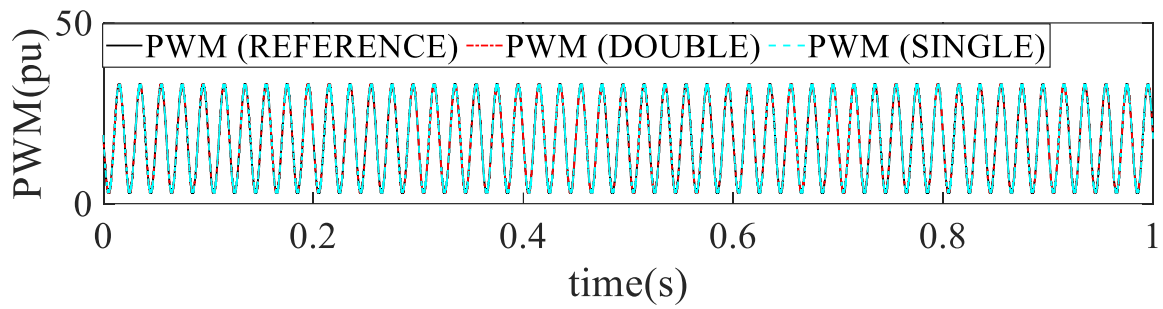
## 5.5 Case study

### 5.5.1 PWM control block

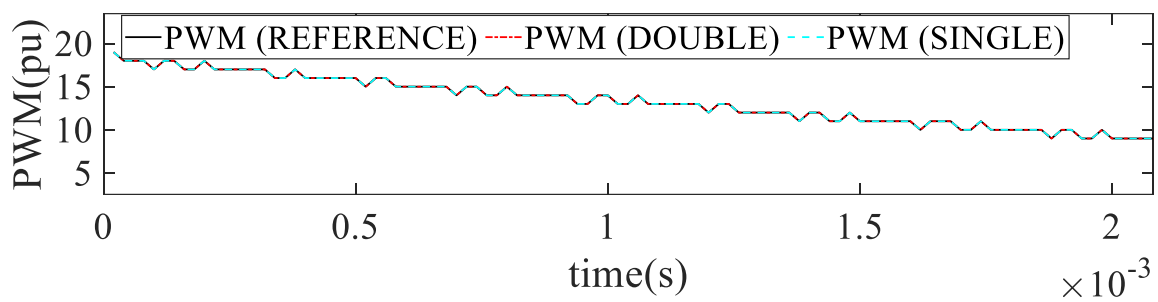
To analyse the precision impact for PWM, Single-Precision and Double-Precision are both applied for single PWM generator block. A 36-level PWM control in Fig. 5.12 is tested. A same model in MATLAB is selected to compare accuracy. Fig. 5.13 provides PWM signal and error simulation results



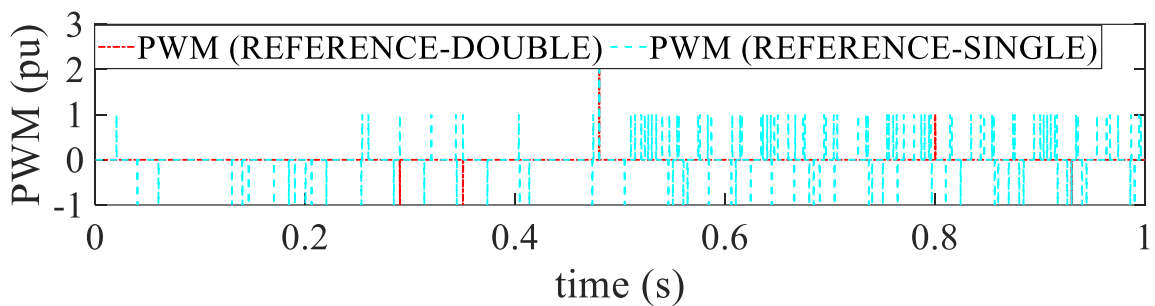
**Fig. 5.12 PWM Generator**



(a) Long-duration comparison



(b) Short-duration comparison



(c) Error comparison

**Fig. 5.13 PWM signal result comparison**

Seen from Fig. 5.13, Single-Precision leads to inaccurate results at more discrete time points. Double-Precision can reduce the average relative error from 8% to less than 4%, compared with Single-Precision. This is because PWM control is operating at high frequency. Single-Precision, lack of efficient bandwidth, will lead to low accuracy for comparison between

referenced signal and carrier signal. This is more significant for zero-crossing point. This problem can be more significant for multi-level PWM control. Therefore, Double-Precision is chosen for the whole MMC system to avoid uncertain control signal errors.

### **5.5.2 HVDC-MMC control system**

In order to test the performance of MMC aggregated model on FPGA, a 36-level HVDC-MMC interconnection system [112] in Fig. 5.14 is simulated. The nominal power and DC voltage is 1000MW and 320kV. MMC converter is implemented using an aggregated model to simulate 36 power modules per arm, where capacitance is assumed with balanced voltage. The various blocks of the control system [112] is depicted in Fig. 5.15. Per arm modulation references are calculated from external measurements, including DC voltage  $V_{dc}$ , secondary current  $I_{sec}$ , and secondary voltage  $V_{sec}$ . DC side converter will be turned on from 1s. Simulation time step is 50 $\mu$ s. The results for voltage and current are displayed in Fig. 5.16. Timing design for HVDC-MMC system is depicted in Fig. 5.17. Table 5.1 details resource utilization for HVDC-MMC system. The same MATLAB model is set as a referenced model for the comparison.

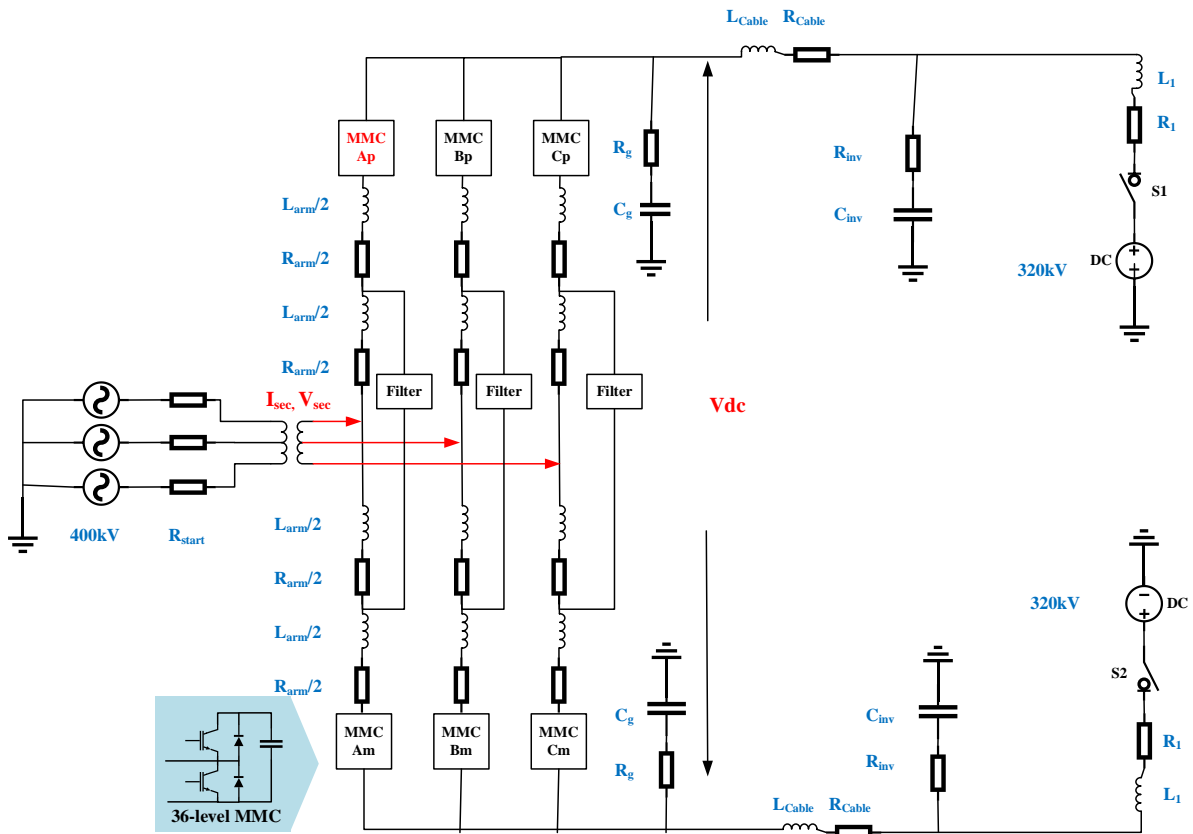


Fig. 5.14 HVDC-MMC topology [112]

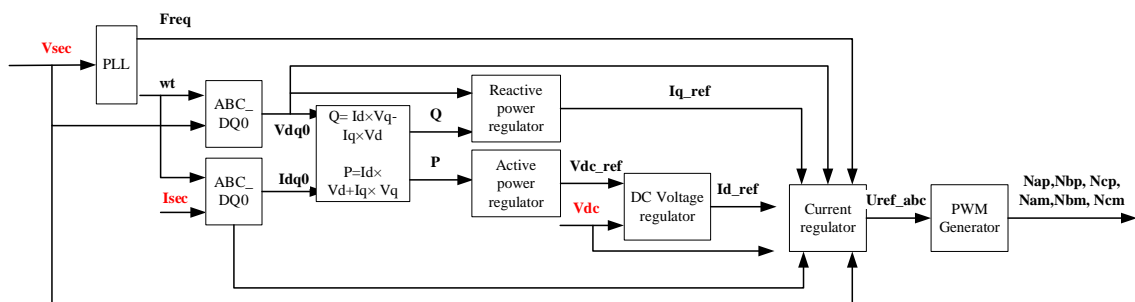
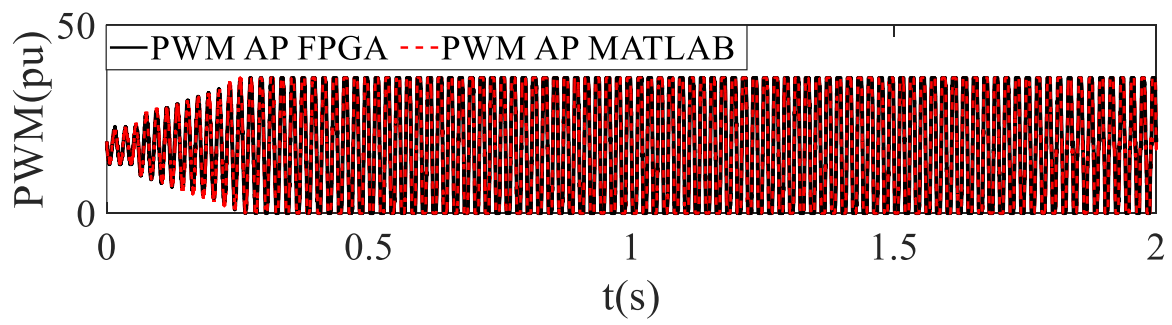
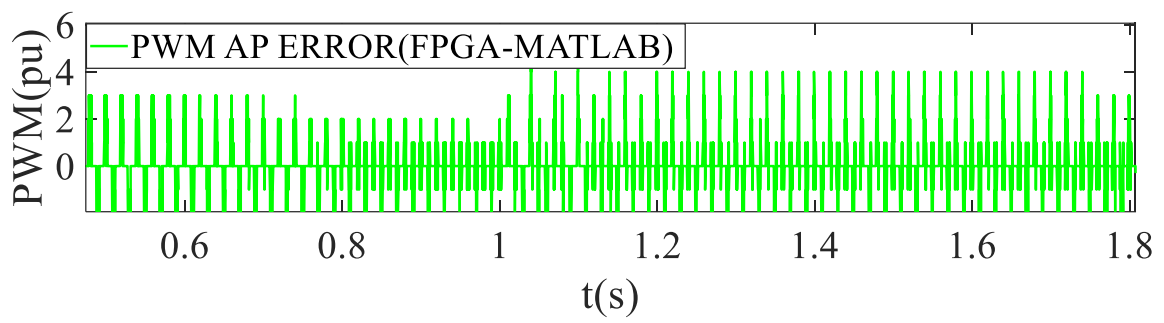


Fig. 5.15 Block diagram of HVDC-MMC control loop [112]

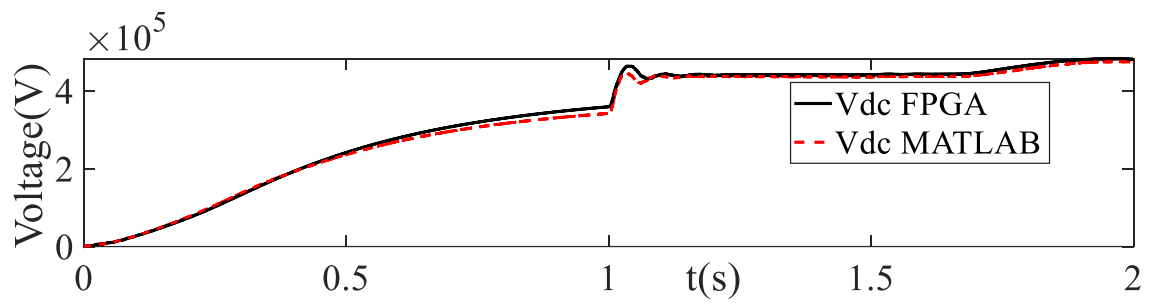




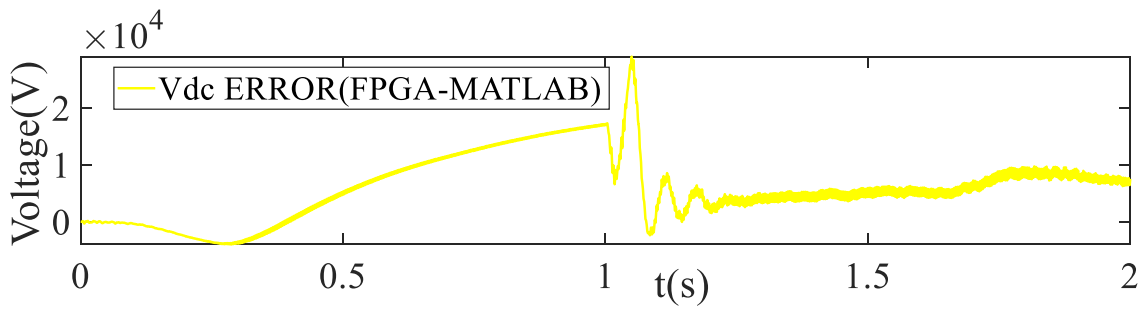
(a) PWM control signal for AP



(b) PWM control signal error for AP

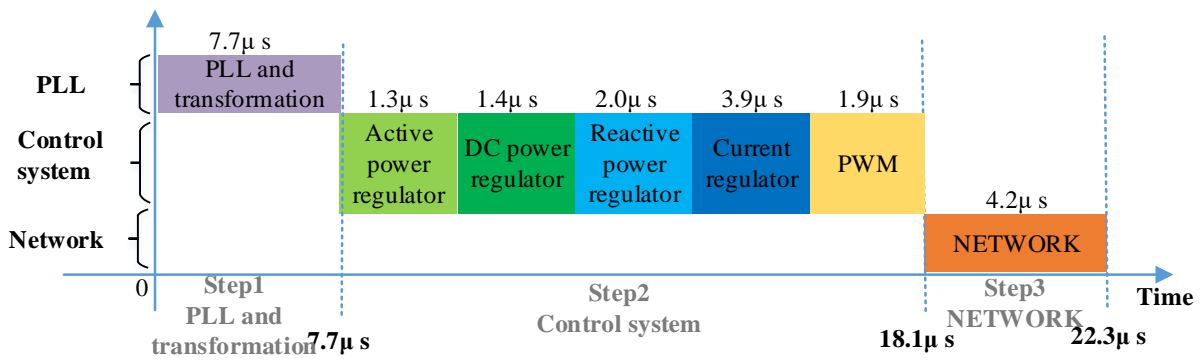


(c) Vdc voltage



(d) Vdc voltage error

**Fig. 5.16 HVDC-MMC simulation results**



**Fig. 5.17 Time schedule of HVDC-MMC system.**

**Table 5.1 Resource utilization of HVDC-MMC system.**

	Registers	LUTs	RAM	DSP
HVDC-MMC	36%	77%	16%	93%

For accuracy analysis, the PWM control signal and DC voltage are still matched with the referenced model as shown in Fig. 5.16. For DC voltage, the average relative error is less than 5%. After DC converter turns on at 1s, the relative error increased suddenly at peak value at 1.1s, and faded until 1.5s. The major error is caused by the interpolation blocks in hardware, which is a treat-off for fast speed in hardware. For PWM control signal, the average relative error is instant at discrete time point. PWM is implemented by comparing calculated values with the referenced values to determine 0 and 1. Owing to the difference between MATLAB and FPGA IP Core, very close values can get different results at zero-crossing point. With this aggregated model, control system dynamics and circulating voltages phenomena are all well-represented on FPGA.

To evaluate timing performance, Fig. 5.17 shows that total calculation time only costs 22.3 $\mu$ s. Therefore, free time slots are still available for other components and control system. Moreover, smaller simulation time steps, such as 40 $\mu$ s, can also be applied for this HVDC-MMC system.

Table 5.1 shows the detailed percentage of resource utilization. As seen from that, all resources, such as DSP and LUT, are all below 100%. With the help of the shift memory unit, the RAM utilization percentage is still below 16%. Therefore, the whole HVDC-MMC system using the aggregated model can be implemented successfully on a single FPGA board.

## 5.6 Conclusion

This chapter mainly introduced the mathematical models for high-frequency power electronic devices and control, such as MMC and PWM control. To simplify FPGA-based implementation, optimized strategies for lower memory and faster speed are also presented.

The following conclusions can be drawn:

- Owing to efficient bandwidth, Double-Precision is more recommended for high-frequency PWM controls than Single-Precision, reducing relative error to less than 4%.
- Optimized memory strategies for hardware, such as shift memory shift, can help to achieve HVDC-MMC system and control system on single FPGA board using low-memory design, with less than 16% RAM.
- For faster calculation speed, time switch strategy and interpolation can efficiently accelerate data transfer and calculation on FPGA, using only 22.3 $\mu$ s for the whole calculation.
- For hardware application, control dynamics and voltage phenomenon of HVDC-MMC system can be well-represented using the aggregated model on FPGA, reducing relative error to less than 5%. Using the aggregated model, complex calculations of frequent changing topologies are simplified by equivalent voltage sources.

# CHAPTER 6 CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusions

The major work of this thesis is to find more accurate and simplified solutions for FPGA-based EMT. With absolute numerical stability, the whole work is completely using trapezoidal rule to discretize continuous model. The conclusions can be drawn as follows:

- **More accurate and resource-saving precision algorithms:** To improve accuracy, Floating-Point precision plays an important role in hardware computation. Default Single-Precision algorithm, used in most researches, will lead to phase shifts on rotating transformation compared with Double-Precision. To eliminate these possible errors, three algorithms, including Single-Precision, Double-Precision and Mixed-Precision, are established for different application scales. Mixed-Precision algorithm is built up by using Single-Precision for non-rotating component (such as RL branch) and Double-Precision for rotating component (such as synchronous machine) in the same system. For linear components, Single-Precision and Double-Precision can both reduce relative error of 4-bus system to less than 1%. For non-linear components, only Double-Precision algorithm can eliminate significant phase shift error, led by Single-Precision, to less than 2%. For integrated system, Mixed-Precision can not only reduce relative error to less than 5% but also save 27% LUT and 15% DSP. Mixed-Precision can be an efficient alternative for Double-Precision when resource utilization is very limited.
- **Faster and more accurate initialization:** By providing initial steady-state, initialization is also significant for improving accuracy of FPGA-based EMT. Without

initialization, it is easy to accumulate from non-initialized time point to steady-state time point, which takes up to tens of seconds. This has not been mentioned by recent researches in FPGA-based EMT field. For efficient initialization, four initialization methods, including physical interface (Method 1), signal declaration (Method 2), signal assignment (Method 3) and COE files (Method 4), are proposed for the first time. Theoretical and graphical comparisons show Method 4 has the most significant advantages, including simplest routing and lowest timing. With developed structure and algorithm, massive dataflow can be transferred from off-line software to real-time hardware. Device-level case study shows that Method 2 and 3 will increase timing constraint to from 9.414 ns to 10.892ns and 9.985ns, while Method 4 remains 9.414ns. Method 2 and 3, although simple for single signal, are still unsuitable for large system initialization. Without any initialization time, Method 4, can initialize 11-bus 4-machines system using only 2% LUT and 1% DSP resources, which reduces relative error to less than 5%. Method 4, with best routing, resource utilization and timing, is highly recommended for further FPGA-based EMT initialization.

- **Intelligent MATLAB-to-FPGA toolbox:** Independent programming for different scale systems, consisting of similar components, are time-consuming and effort-consuming on FPGA. To simplify hardware programming, MTF toolbox is presented for FPGA-based EMT. MATLAB, with powerful computational functions and flexible file access, is well suited for developing this proposed toolbox. The only task for users is to define data sources, while MTF toolbox can automatically extract from sufficient data into submodule, memory unit and main controllers. Extracted constants and variables are kept in memory unit, basic operators are mapped in sub-modules, and time

schedules are arranged in main controllers. This procedure is easy-to-use and generic, avoiding repetitive efforts. This MTF toolbox can generate 89 files for 11-bus and 39-bus network within 50s and 300s. After synthesizing these files, the relative error of 39-bus network is still less than 5%. Compared with 11-bus network, 39-bus network costs more 2% LUT and 2% DSP. This MTF toolbox can also be used for testing benchmarks.

- **High-frequency HVDC-MMC model:** Power electronic devices with high-frequency switching, such as half-bridge MMC, will lead frequent topologies on FPGA. Existing researches, such as RTDs, are using more than 2 PB5 or FPGA board to implement. For single FPGA board, aggregate model of MMC and control systems is derived based on trapezoidal rule, replacing frequent topologies with equivalent voltage source. Optimized strategies, such as shift memory unit and interpolation, can help to reduce lower resource utilization, timing and memory space. PWM control case study illustrates the significance of Double-Precision by reducing error to less than 4%. With the help of optimized strategies, a 36-level HVDC-MMC system using the aggregated model is still achievable on single FPGA board with less than 5% error and 22.3 $\mu$ s calculation time. This HVDC-MMC model can interface with new components on single FPGA board.

With adaptive precision and initialization, the accuracy is much more enhanced for general FPGA-BASED EMT simulation. With intelligent MTF toolbox, hardware programming becomes more easy-to-use for beginners. With high-frequency electronic devices, modelling capability is enhanced significantly for FPGA-based EMT. Owing to developed works, future FPGA-based researches can be started from a more accurate and simplified beginning.

## 6.2 Future works

Future work can be divided into following three aspects:

- **Variable simulation time step:** Variable simulation time step can be treated as a solution for integrating multi-vector energy systems. For example, thermal energy and cooling power systems can be involved with power system in real-time, which can be combined as a Combined Cooling, Heat and Power (CCHP) system. Consider transient dynamics, thermal energy and cooling energy are much slower than power system. Using variable simulation steps can simulate slow component and fast component in the same system in real-time. It is beneficial to implement similar case studies for multi-energy system operation.
- **Renewable energies:** Wind turbine with complex converter control system is also another prospective aspect for future research. Considering grid forming and grid following types are required for different scenarios. How to solve wind turbines with short distances in large system is still the key problem for real-time simulation. The interface between electric energy and cooling energy can also be investigated in FPGA platform.
- **Precision analysis:** The Mixed-Precision can be applied in other systems, such as HVDC-MMC system. The application boundary between Single-Precision and Double-Precision can also be investigated for both component internal structure and global



network. This thesis mainly uses Floating-Point values, Fixed-Point values can also be considered as possible arithmetic calculations.

- Accuracy improvement: Compared with MATLAB model, the accuracy is still within 5%. To reduce further error, more intelligent functions can be fast implemented in FPGA platform. For example, more rounding functions can be developed, similar to the *fix* and *round* function in MATLAB. This attempt can make FPGA calculation more competitive compared with MATLAB.
- Detailed MMC model: This thesis assumes that the voltages of capacitors are balanced for aggregate model, which is not realistic. Detailed MMC model, with unbalanced capacitor, is much closer to practical operations. The modelling is much more complicated. It is worth to investigate the possibility to implement detailed model of a 36-level HVDC-MMC on FPGA platform.
- Multi-FPGA simulation: All the case studies are implemented on single board. In the future, the communication between multi-FPGA boards can be investigated. Large-scale system, such as 100-bus, can be partitioned on different boards.
- Integration method: This thesis applies trapezoidal rule for all the EMT calculations. In the future, other integration methods can be improved with corrected actions, making it suitable for EMT simulation. The numerical difference of using different integration methods can also be modelled and simulated.
- Initialization: Device-level component initialization is only used for synchronous machine model in starting-up procedure. In the future, initialization can also be tested for other rotating components (such as wind turbines) and non-rotating components (such as MMC).

- Intelligent programming: This thesis built up MTF toolbox using MATLAB for accelerating FPGA-based EMT. Other languages, such as python, can also be investigated to simplify FPGA programming. Relevant factors, such as programming efforts and generation time, can be compared with MATLAB.

## REFERENCES

- [1] P. Kundur, *Power System Control and Stability*. New York, NY, USA:McGraw-Hill, 1994.
- [2] A. T. De Almeida, F. J. T. E. Ferreira and A. Q. Duarte, "Technical and Economical Considerations on Super High-Efficiency Three-Phase Motors," *IEEE Transactions on Industry Applications*, vol. 50, no. 2, pp. 1274-1285, March-April 2014, doi: 10.1109/TIA.2013.2272548.
- [3] Gonen, T., *Electric Power Transmission System Engineering*, Wiley, New York, 1986.
- [4] N. H. Dandachi, M. J. Rawlins, O. Alsac, M. Prais and B. Stott, "OPF for reactive pricing studies on the NGC system," in *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 226-232, Feb. 1996, doi: 10.1109/59.486099.
- [5] M. W. Stoddart, "The role of HVDC in the electrical development of Canada," in *Electrical Engineering*, vol. 82, no. 10, pp. 602-609, Oct. 1963, doi: 10.1109/EE.1963.6540928.
- [6] N. G. Hingorani, "Transient Overvoltage on a Bipolar HVDC Overhead Line Caused by DC Line Faults," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 4, pp. 592-610, April 1970, doi: 10.1109/TPAS.1970.292606.
- [7] Ofgem. Existing and future interconnector projects, Mar 2023, [Online]. Available: <https://www.ofgem.gov.uk/electricity/transmission-networks/electricity-interconnectors>
- [8] Gonen, T., *Electric Power Distribution System Engineering*, Wiley, New York, 1986
- [9] W. H. Kersting, *Distribution System Modelling and Analysis*. New York, NY: CRC Press, 2018
- [10] A. A. Sallam and O. P. Malik, *Electric Distribution Systems*. John Wiley & Sons, 2018.

- [11] C. Zachariades, V. Peesapati, R. Gardner and O. Cwikowski, "Electric Field and Thermal Analysis of 132 kV Ceramic Oil-Filled Cable Sealing Ends," in *IEEE Transactions on Power Delivery*, vol. 36, no. 1, pp. 311-319, Feb. 2021, doi: 10.1109/TPWRD.2020.2977728.
- [12] Ofgem. Electricity Networks Strategic Framework: Enabling a secure, net zero energy system, Aug 2022, [Online]. Available: [https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/1096283/electricity-networks-strategic-framework.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1096283/electricity-networks-strategic-framework.pdf)
- [13] National Grid, "What's the difference between electricity transmission and distribution", Sep 2022, [Online]. Available: <https://www.nationalgrid.com/stories/energy-explained/electricity-transmission-vs-electricity-distribution>
- [14] J. A. Schrock et al., "Failure Modes of 15-kV SiC SGTO Thyristors During Repetitive Extreme Pulsed Overcurrent Conditions," in *IEEE Transactions on Power Electronics*, vol. 31, no. 12, pp. 8058-8062, Dec. 2016, doi: 10.1109/TPEL.2016.2574625.
- [15] C. Liu et al., "Hybrid SiC-Si DC-AC Topology: SHEPWM Si-IGBT Master Unit Handling High Power Integrated With Partial-Power SiC-MOSFET Slave Unit Improving Performance," in *IEEE Transactions on Power Electronics*, vol. 37, no. 3, pp. 3085-3098, March 2022, doi: 10.1109/TPEL.2021.3114322.
- [16] Jensen, L. EU Climate Target Plan Raising the Level of Ambition for 2030. Briefing—Towards Climate Neutrality. European Parliament. 2020. Available online: [https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659370/EPRS\\_BRI\(2020\)6\\_59370\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2020/659370/EPRS_BRI(2020)6_59370_EN.pdf)

- [17] M. Child, R. Ilonen, M. Vavilov, M. Kolehmainen, and C. Breyer, "Scenarios for sustainable energy in Scotland, " *Wind Energy*, vol. 22, no. 5, pp. 666–684, May 2019, doi: 10.1002/we.2314.
- [18] Castek, R.; Harkin, S. Evidence Review for Hydrogen for Heat in Buildings. 2021. Available online: <https://www.climateexchange.org.uk/media/4982/cxc-evidence-review-for-hydrogen-heat-in-buildings-august-2021.pdf>
- [19] Department for Business, E. I. S. Digest of UK Energy Statistics (DUKES): renewable sources of energy, July 2022, [Online]. Available: <https://www.gov.uk/government/statistics/renewable-sources-of-energy-chapter-6-digest-of-united-kingdom-energy-statistics-dukes>
- [20] G. Renn, G. Ma, and N. Cong, "Review of electrical energy storage system for vehicular applications, " *Renew. Sustain. Energy Rev.*, vol. 41, pp. 225–236, Jan. 2015.
- [21] M. A. Hannan, M. M. Hoque, A. Mohamed, and A. Ayob, "Review of energy storage systems for electric vehicle applications: Issues and challenges, " *Renew. Sustain. Energy Rev.*, vol. 69, pp. 771–789, Mar. 2017.
- [22] Intertek, Domestic Battery Energy Storage Systems: A review of safety risks, Oct 2020, [Online]. Available: <https://www.gov.uk/government/publications/domestic-battery-energy-storage-systems>
- [23] M. H. J. Bollen and L. D. Zhang, "Different methods for classification of three-phase unbalanced voltage dips due to faults, " *Electr. Power Syst. Res.*, vol. 66, no. 1, pp. 59–69, Jul. 2003
- [24] Y. Xia, Y. Chen, Y. Song, S. Huang, Z. Tan and K. Strunz, "An Efficient Phase Domain Synchronous Machine Model With Constant Equivalent Admittance Matrix," *IEEE*

- Transactions on Power Delivery, vol. 34, no. 3, pp. 929-940, June 2019, doi: 10.1109/TPWRD.2019.2897612.
- [25] M. H. J. Bollen, Understanding Power Quality Problems, Voltage Sags and Interruptions. New York, NY: IEEE Press, 2000
- [26] A. Greenwood, Electrical Transients in Power Systems, 2nd ed. New York: Wiley, 1991.
- [27] J. Martinez-Velasco, Power System Transients: Parameter Determination. Boca Raton, FL, USA: CRC, 2009
- [28] IEC 60071-1, Insulation co-ordination—Part 1: Definitions, principles and rules, January 2006. 5. CIGRE WG 33.02, Guidelines for Representation of Network Elements when Calculating Transients, CIGRE Technical Brochure no. 39, 1990.
- [29] P. Heine and M. Lehtonen, "Voltage sag distributions caused by power system faults," in IEEE Transactions on Power Systems, vol. 18, no. 4, pp. 1367-1373, Nov. 2003, doi: 10.1109/TPWRS.2003.818606.
- [30] IEC 60071-4, Insulation co-ordination—Part 4: Computational guide to insulation co-ordination and modelling of electrical networks, June 2004.
- [31] A. Morched, B. Gustavsen, and M. Tartibi, "A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables, " IEEE Trans. Power Del., vol. 14, no. 3, pp. 1032–1038, Jul. 1999.
- [32] N. Watson and J. Arrillaga, Power Systems Electromagnetic Transients Simulation. Stevenage, U.K.: IET, 2003.
- [33] B. Salarieh, H. M. J. De Silva, A. M. Gole, A. Ametani and B. Kordi, "An Electromagnetic Model for the Calculation of Tower Surge Impedance Based on Thin

- Wire Approximation," in IEEE Transactions on Power Delivery, vol. 36, no. 2, pp. 1173-1182, April 2021, doi: 10.1109/TPWRD.2020.3003250.
- [34] Tian Fang, Yue Chengyan, Wu Zhongxi and Zhou Xiaoxin, "Realization of Electromechanical Transient and Electromagnetic Transient Real Time Hybrid Simulation in Power System," 2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific, Dalian, China, 2005, pp. 1-6, doi: 10.1109/TDC.2005.1546932.
- [35] J. Vega-Herrera, C. Rahmann, F. Valencia and K. Strunz, "Analysis and Application of Quasi-Static and Dynamic Phasor Calculus for Stability Assessment of Integrated Power Electric and Electronic Systems," in IEEE Transactions on Power Systems, vol. 36, no. 3, pp. 1750-1760, May 2021, doi: 10.1109/TPWRS.2020.3030225.
- [36] National Academies of Sciences, Engineering, and Medicine. Analytic Research Foundations for the Next-Generation Electric Grid. Washington, DC, USA: The National Academies Press, 2016.
- [37] U. N. Gnanarathna, A. M. Gole and R. P. Jayasinghe, "Efficient Modeling of Modular Multilevel HVDC Converters (MMC) on Electromagnetic Transient Simulation Programs," in IEEE Transactions on Power Delivery, vol. 26, no. 1, pp. 316-324, Jan. 2011, doi: 10.1109/TPWRD.2010.2060737.
- [38] Y. Xu, C. N. M. Ho, A. Ghosh and D. Muthumuni, "An Electrical Transient Model of IGBT-Diode Switching Cell for Power Semiconductor Loss Estimation in Electromagnetic Transient Simulation," in IEEE Transactions on Power Electronics, vol. 35, no. 3, pp. 2979-2989, March 2020, doi: 10.1109/TPEL.2019.2929113.

- [39] A. A. van der Meer, M. Gibescu, M. A. M. M. van der Meijden, W. L. Kling and J. A. Ferreira, "Advanced Hybrid Transient Stability and EMT Simulation for VSC-HVDC Systems," in *IEEE Transactions on Power Delivery*, vol. 30, no. 3, pp. 1057-1066, June 2015, doi: 10.1109/TPWRD.2014.2384499.
- [40] N. Shirazi, A. Walters and P. Athanas, "Quantitative analysis of floating point arithmetic on FPGA based custom computing machines," *Proceedings IEEE Symposium on FPGAs for Custom Computing Machines*, Napa Valley, CA, USA, 1995, pp. 155-162, doi: 10.1109/FPGA.1995.477421.
- [41] C. S. Brett, C. M. Saldanha and D. A. Lowther, "Interval mathematics for knowledge-based computer aided design in magnetics," in *IEEE Transactions on Magnetics*, vol. 26, no. 2, pp. 803-806, March 1990, doi: 10.1109/20.106439.
- [42] M. Rahimo et al., "Characterization of a Silicon IGBT and Silicon Carbide MOSFET Cross-Switch Hybrid," in *IEEE Transactions on Power Electronics*, vol. 30, no. 9, pp. 4638-4642, Sept. 2015, doi: 10.1109/TPEL.2015.2402595.
- [43] F. J. Ferreira Pereira, J. M. Rodrigues, K. L. Miranda de Oliveira, D. R. Ribeiro Penido Araujo and L. R. de Araujo, "Simulations and analysis of distribution systems with aspects of smart grids using MICQ, RTDS and PSCAD," *2013 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America)*, Sao Paulo, Brazil, 2013, pp. 1-8, doi: 10.1109/ISGT-LA.2013.6554416.
- [44] RTDS Technologies, June 2023, [Online]. Available: <https://www.rtds.com/>
- [45] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single-and Multiphase Networks," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 4, pp. 388-399, April 1969, doi: 10.1109/TPAS.1969.292459.



- [46] H. W. Dommel, "Nonlinear and Time-Varying Elements in Digital Simulation of Electromagnetic Transients," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, no. 6, pp. 2561-2567, Nov. 1971, doi: 10.1109/TPAS.1971.292905.
- [47] H. W. Dommel, *EMTP Theory Book*. Portland, OR: Bonneville Power Admin., Aug. 1986.
- [48] V. Brandwajn, W. S. Meyer and H. W. Dommel, "Synchronous Machine Initialization For Unbalanced Network Conditions Within An Electromagnetic Transients Program," *IEEE Conference Proceedings Power Industry Computer Applications Conference*, 1979. PICA-79., Cleveland, OH, USA, 1979, pp. 38-41, doi: 10.1109/PICA.1979.720043.
- [49] A. Vaccarella, E. De Momi, A. Enquobahrie and G. Ferrigno, "Unscented Kalman Filter Based Sensor Fusion for Robust Optical and Electromagnetic Tracking in Surgical Navigation," in *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 7, pp. 2067-2081, July 2013, doi: 10.1109/TIM.2013.2248304.
- [50] N. Mohan, W. P. Robbins, T. M. Undeland, R. Nilssen and O. Mo, "Simulation of power electronic and motion control systems-an overview," in *Proceedings of the IEEE*, vol. 82, no. 8, pp. 1287-1302, Aug. 1994, doi: 10.1109/5.301689.
- [51] H. W. Dommel and N. Sato, "Fast Transient Stability Solutions," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-91, no. 4, pp. 1643-1650, July 1972, doi: 10.1109/TPAS.1972.293341.
- [52] Minwon Park and In-Keun Yu, "A novel real-time simulation technique of photovoltaic generation systems using RTDS," in *IEEE Transactions on Energy Conversion*, vol. 19, no. 1, pp. 164-169, March 2004, doi: 10.1109/TEC.2003.821837.

- [53] M. D. Omar Faruque et al., "Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis," in *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 2, pp. 63-73, June 2015, doi: 10.1109/JPETS.2015.2427370.
- [54] M. Matar, M. Abdel-Rahman, and A. Soliman, "FPGA-based real-time digital simulation," in *Proc. Int. Conf. on Power System Transients*, Montreal, QC, Canada, Jun. 2005, pp. 1–6.
- [55] P. Venne, J.-N. Paquin, and J. Bélanger, "The what, where and why of real-time simulation," in *Proc. 2010 IEEE PES General Meeting, Tutorial\_04*, 2010, pp. 37–49.
- [56] A. Masoom, J. Mahseredjian, T. Ould-Bachir and A. Guironnet, "MSEMT: An Advanced Modelica Library for Power System Electromagnetic Transient Studies," in *IEEE Transactions on Power Delivery*, vol. 37, no. 4, pp. 2453-2463, Aug. 2022, doi: 10.1109/TPWRD.2021.3111127.
- [57] A. M. Gole, O. B. Nayak, T. S. Sidhu and M. S. Sachdev, "A graphical electromagnetic simulation laboratory for power systems engineering programs," in *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 599-606, May 1996, doi: 10.1109/59.496082.
- [58] R. S. Burton, C. F. Fuchshuber, D. A. Woodford and A. M. Gole, "Prediction of core saturation instability at an HVDC converter," in *IEEE Transactions on Power Delivery*, vol. 11, no. 4, pp. 1961-1969, Oct. 1996, doi: 10.1109/61.544283.
- [59] C. Osauskas and A. Wood, "Small-signal dynamic modeling of HVDC systems," in *IEEE Transactions on Power Delivery*, vol. 18, no. 1, pp. 220-225, Jan. 2003, doi: 10.1109/TPWRD.2002.803843.

- [60] M. Saadifard and R. Iravani, "Dynamic Performance of a Modular Multilevel Back-to-Back HVDC System," in *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2903-2912, Oct. 2010, doi: 10.1109/TPWRD.2010.2050787.
- [61] N. Eghtedarpour and E. Farjah, "Power Control and Management in a Hybrid AC/DC Microgrid," in *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1494-1505, May 2014, doi: 10.1109/TSG.2013.2294275.
- [62] S. Chiniforoosh et al., "Steady-State and Dynamic Performance of Front-End Diode Rectifier Loads as Predicted by Dynamic Average-Value Models," in *IEEE Transactions on Power Delivery*, vol. 28, no. 3, pp. 1533-1541, July 2013, doi: 10.1109/TPWRD.2013.2262316.
- [63] S. I. Nanou and S. A. Papathanassiou, "Grid Code Compatibility of VSC-HVDC Connected Offshore Wind Turbines Employing Power Synchronization Control," in *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 5042-5050, Nov. 2016, doi: 10.1109/TPWRS.2016.2515504.
- [64] L. M. Castro and E. Acha, "A Unified Modeling Approach of Multi-Terminal VSC-HVDC Links for Dynamic Simulations of Large-Scale Power Systems," in *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 5051-5060, Nov. 2016, doi: 10.1109/TPWRS.2016.2527498.
- [65] M. O. Faruque, Yuyan Zhang and V. Dinavahi, "Detailed modeling of CIGRE HVDC benchmark system using PSCAD/EMTDC and PSB/SIMULINK," in *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 378-387, Jan. 2006, doi: 10.1109/TPWRD.2005.852376.

- [66] H. T. Nguyen, G. Yang, A. H. Nielsen and P. H. Jensen, "Hardware- and Software-in-the-Loop Simulation for Parameterizing the Model and Control of Synchronous Condensers," in *IEEE Transactions on Sustainable Energy*, vol. 10, no. 3, pp. 1593-1602, July 2019, doi: 10.1109/TSTE.2019.2913471.
- [67] M. Matar and R. Iravani, "The Reconfigurable-Hardware Real-Time and Faster-Than-Real-Time Simulator for the Analysis of Electromagnetic Transients in Power Systems," in *IEEE Transactions on Power Delivery*, vol. 28, no. 2, pp. 619-627, April 2013, doi: 10.1109/TPWRD.2012.2229723.
- [68] RTDS Technology Inc., Winnipeg, MB, Canada. (2014). Real-Time Simulation. [Online]. Available: <http://www.rtds.com>
- [69] P. G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht and L. Arendt, "A real time digital simulator for testing relays," in *IEEE Transactions on Power Delivery*, vol. 7, no. 1, pp. 207-213, Jan. 1992, doi: 10.1109/61.108909.
- [70] Minwon Park and In-Keun Yu, "A novel real-time simulation technique of photovoltaic generation systems using RTDS," in *IEEE Transactions on Energy Conversion*, vol. 19, no. 1, pp. 164-169, March 2004, doi: 10.1109/TEC.2003.821837.
- [71] B. Wang, X. Dong, Z. Bo and A. Perks, "RTDS Environment Development of Ultra-High-Voltage Power System and Relay Protection Test," in *IEEE Transactions on Power Delivery*, vol. 23, no. 2, pp. 618-623, April 2008, doi: 10.1109/TPWRD.2008.915818.
- [72] X. Ma, C. Yang, X. -P. Zhang, Y. Xue and J. Li, "Real-Time Simulation of Power System Electromagnetic Transients on FPGA Using Adaptive Mixed-Precision Calculations," *IEEE Transactions on Power Systems*, 2022, doi: 10.1109/TPWRS.2022.3199181. <https://ieeexplore.ieee.org/document/9858646>

- [73] S. M. S. Trimberger, "Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology: This Paper Reflects on How Moore's Law Has Driven the Design of FPGAs Through Three Epochs: the Age of Invention, the Age of Expansion, and the Age of Accumulation," in *IEEE Solid-State Circuits Magazine*, vol. 10, no. 2, pp. 16-29, Spring 2018, doi: 10.1109/MSSC.2018.2822862.
- [74] J. Rose, R. J. Francis, D. Lewis and P. Chow, "Architecture of field-programmable gate arrays: the effect of logic block functionality on area efficiency," in *IEEE Journal of Solid-State Circuits*, vol. 25, no. 5, pp. 1217-1225, Oct. 1990, doi: 10.1109/4.62145.
- [75] J. Rose and S. Brown, "Flexibility of interconnection structures for field-programmable gate arrays," in *IEEE Journal of Solid-State Circuits*, vol. 26, no. 3, pp. 277-282, March 1991, doi: 10.1109/4.75006.
- [76] N. K. Ratha, K. Karu, Shaoyun Chen and A. K. Jain, "A real-time matching system for large fingerprint databases," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799-813, Aug. 1996, doi: 10.1109/34.531800.
- [77] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203-215, Feb. 2007, doi: 10.1109/TCAD.2006.884574.
- [78] C. Chang, J. Wawrzyniek and R. W. Brodersen, "BEE2: a high-end reconfigurable computing system," in *IEEE Design & Test of Computers*, vol. 22, no. 2, pp. 114-125, March-April 2005, doi: 10.1109/MDT.2005.30.
- [79] Mei-Chen Hsueh, T. K. Tsai and R. K. Iyer, "Fault injection techniques and tools," in *Computer*, vol. 30, no. 4, pp. 75-82, April 1997, doi: 10.1109/2.585157.

- [80] A. Ruan, J. Yang, L. Wan, B. Jie and Z. Tian, "Insight Into a Generic Interconnect Resource Model for Xilinx Virtex and Spartan Series FPGAs," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 60, no. 11, pp. 801-805, Nov. 2013, doi: 10.1109/TCSII.2013.2278129.
- [81] K. Dang Pham, E. Horta and D. Koch, "BITMAN: A tool and API for FPGA bitstream manipulations," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland, 2017, pp. 894-897, doi: 10.23919/DATE.2017.7927114.
- [82] T. Feist, "Vivado design suite, " White Paper, vol. 5, 2012.
- [83] C. Dufour, H. Le-Huy, J. Soumagne, and A. E. Hakimi, "Real-time simulation of power transmission lines using marti model with optimal fitting on dual-DSP card, " IEEE Trans. Power Del., vol. 11, no. 1, pp. 412–419, Jan. 1996.
- [84] Introduction to FPGA Acceleration, Stemmer Imag. GmbH, Puchheim, Germany, 2017, pp. 1–3. [Online]. Available: <https://www.commonvisionblox.com/en/technical-tips/introduction-to-fpgaacceleration/>
- [85] S. Kilts, Advanced FPGA Design: Architecture, Implementation, and Optimization. New York, NY, USA: Wiley, 2007.
- [86] Digilent, Nexys A7 FPGA Board Reference Manual, Revised July 10, 2019. Available: [https://digilent.com/reference/\\_media/reference/programmablelogic/nexys-a7/nexys-a7\\_rm.pdf](https://digilent.com/reference/_media/reference/programmablelogic/nexys-a7/nexys-a7_rm.pdf)
- [87] P. M. Menghal and A. J. Laxmi, "Real time simulation: A novel approach in engineering education," in Proc. 3rd Int. Conf. Electron. Comput. Technol., Kanyakumari, India, Apr. 2011, pp. 215–219.

- [88] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP, " *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892–902, Apr. 2009
- [89] G. G. Parma and V. Dinavahi, "Real-Time Digital Hardware Simulation of Power Electronics and Drives," in *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 1235-1246, April 2007, doi: 10.1109/TPWRD.2007.893620.
- [90] Y. Chen and V. Dinavahi, "Digital Hardware Emulation of Universal Machine and Universal Line Models for Real-Time Electromagnetic Transient Simulation," in *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1300-1309, Feb. 2012, doi: 10.1109/TIE.2011.2157296.
- [91] J. Liu and V. Dinavahi, "Detailed Magnetic Equivalent Circuit Based Real-Time Nonlinear Power Transformer Model on FPGA for Electromagnetic Transient Studies," in *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1191-1202, Feb. 2016, doi: 10.1109/TIE.2015.2477487.
- [92] A. Hadizadeh, M. Hashemi, M. Labbaf and M. Parniani, "A Matrix-Inversion Technique for FPGA-Based Real-Time EMT Simulation of Power Converters," in *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1224-1234, Feb. 2019, doi: 10.1109/TIE.2018.2833058.
- [93] Z. Shen, T. Duan and V. Dinavahi, "Design and Implementation of Real-Time Mpsoc-FPGA-Based Electromagnetic Transient Emulator of CIGRÉ DC Grid for HIL Application," in *IEEE Power and Energy Technology Systems Journal*, vol. 5, no. 3, pp. 104-116, Sept. 2018, doi: 10.1109/JPETS.2018.2866589.
- [94] P. C. Magnusson, *Transmission Lines and Wave Propagation*. Boston, MA: Allyn & Bacon, 1965, pp. 118–130.

- [95] X. P. Zhang, C. Rehtanz, and B. Pal, Flexible AC Transmission Systems: Modelling and Control. New York: Springer, 2006.
- [96] Y. Chen, "Large-scale real-time electromagnetic transient simulation of power systems using hardware emulation on FPGAs," Ph.D. dissertation, Dept. Elect. Comp. Eng., Univ. Alberta, Edmonton, AB, Canada, 2012.
- [97] "ML605 Hardware User Guide", UG534 (v1.5) Feb. 15, 2011, [http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug534.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug534.pdf)
- [98] "LogiCORE IP Floating Point Operator v5.0, DS335" Xilinx Corp., March 2011.
- [99] J. D. Cappello and D. Strenski, "A practical measure of FPGA floating point acceleration for High Performance Computing," 2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors, Washington, DC, USA, 2013, pp. 160-167, doi: 10.1109/ASAP.2013.6567570.
- [100] M. F. Tolba, M. E. Fouda, H. G. Hezayyin, A. H. Madian and A. G. Radwan, "Memristor FPGA IP Core Implementation for Analog and Digital Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 8, pp. 1381-1385, Aug. 2019, doi: 10.1109/TCSII.2018.2882496.
- [101] "Distributed Memory Generator v8.0, PG063" Xilinx Corp., Nov 2015.
- [102] K. Babulu and K. S. Rajan, "FPGA Implementation of USB Transceiver Macrocell Interface with USB2.0 Specifications," 2008 First International Conference on Emerging Trends in Engineering and Technology, Nagpur, India, 2008, pp. 966-970, doi: 10.1109/ICETET.2008.77.



- [103] X. Wang and R. M. Mathur, "Real-time digital simulator of the electromagnetic transients of transmission lines with frequency dependence," in *IEEE Transactions on Power Delivery*, vol. 4, no. 4, pp. 2249-2255, Oct. 1989, doi: 10.1109/61.35654.
- [104] 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.
- [105] P. G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht, and L. Arendt, "A real-time digital simulator for testing relays," *IEEE Trans. Power Del.*, vol. 7, no. 1, pp. 207–213, Jan. 1992.
- [106] Xilinx, San Jose, CA, ChipScope Software and ILA Cores User Manual, v. 1.1 ed., June 2000.
- [107] J. Protic, M. Tomasevic and V. Milutinovic, "Distributed shared memory: concepts and systems," in *IEEE Parallel & Distributed Technology: Systems & Applications*, vol. 4, no. 2, pp. 63-71, Summer 1996, doi: 10.1109/88.494605.
- [108] K. Kawabata, M. Tanizawa, K. Ishikawa and Y. Inoue, "Micro magnetic simulation of write error probability in STT-MRAM," 2008 International Conference on Simulation of Semiconductor Processes and Devices, Kanagawa, Japan, 2008, pp. 53-56, doi: 10.1109/SISPAD.2008.4648235.
- [109] K. Ou et al., "Research and application of small time-step simulation for MMC VSC-HVDC in RTDS," 2014 International Conference on Power System Technology, Chengdu, China, 2014, pp. 877-882, doi: 10.1109/POWERCON.2014.6993901.
- [110] MATLAB, MathWorks, Natick, MA, USA, 2012

- [111] M.-F. Tsai, T. P. Quy, B.-F. Wu, and C.-S. Tseng, "Model construction and verification of a BLDC motor using MATLAB/Simulink and FPGA control," in Proc. 2011 6th IEEE Conf. Ind. Electron. Appl., Jun. 2011, pp. 1797–1802
- [112] B. Jacobson, P. Karlsson, G. Asplund, L. Harnefors, and T. Jonsson, "VSC–HVDC transmission with cascaded two-level converters," presented at the Cigre Session, Paris, France, vol. B4-110, 2010.
- [113] IEEE committee report, "Dynamic models for steam and hydro turbines in power system studies," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-92, No. 6, 1973, pp. 1904-1915.
- [114] PSCADTM IEEE 39 Bus System, Reversion 1, Manitoba HVDC Res. Centre, Winnipeg, MB, Canada, 2010.
- [115] CIGRE WG B4-57, "Guide for development of models for HVDC converters in a HVDC grid," Int. Council Large Elect. Syst., Paris, France, Rep. 604, Dec. 2014.

# APPENDIX TEST SYSTEMS

## A.1 FPGA IP Cores Latency

The global clock frequency for ML605 FPGA board is selected as 100 MHz. The detailed latency for basic operators is given in Table A.1:

**Table A.1 Latencies of IP COREs**

Operator	Clock cycles
Add/Subtract	4
Multiply	4
Divide	20
Square-root	6
Max	2
Float-to-Fixed	2
Fixed-to-Float	2
Float-to-Float	3
Data-read	1
Data-write	1

## A.2 4-bus network

4-bus network is a proposed hypothetical topology. RL branch represents simplified overhead lines, ideal voltage source represents infinite power system network. The parameters of 4-bus network in Chapter 2 are given in Table A.1:

**Table A.2 Parameters of 4-bus network**

Number of phases	3
Frequency for RLC specification	60
Resistance per unit length (Ohms/km)	$r_1 = 0.0001, r_0 = 0.0002$
Inductance per unit length (H/km)	$x_1 = 0.0001, x_0 = 0.0002$
Capacitance per unit length (F/km)	$b_1 = 0.00175, b_0 = 0.000875$
TL1 length (km)	25
Transformer primary voltage and secondary voltage (kV)	23/200
Transformer nominal power (MVA)	900
R1 (ohm)	1
L1 (H)	0.001
Ideal voltage source frequency (Hz) and phase-to-phase amplitude (kV)	60Hz, 20kV

RL branch is the representation of simplified overhead lines.

### **A.3 Synchronous machine and control system**

The single synchronous machine system in Chapter 2 and Chapter 3 is a proposed hypothetical benchmark. The synchronous machine in Chapter 2, Chapter 3 and Chapter 4 are using per unit model [1]. For nominal values, nominal power is 555MVA, line-to-line voltage RMS value is 24kV, nominal power is 60 Hz. The inertia coefficient is 3.525 S, friction factor is 0 pu, and pole pair is 1. Other winding resistance and leakage inductance are listed in per unit values as follows:

**Table A.3 Synchronous machine parameters [1]**

$R_s$	0.003
$R_f$	0.0006
$R_{kd}$	0.0284
$R_{kq1}$	0.00619
$R_{kq2}$	0.02368
$L_l$	0.15
$L_{md}$	1.66
$L_{mq}$	1.61
$L_{lfd}$	0.165
$L_{lkd}$	0.1713
$L_{lkq1}$	0.7252
$L_{lkq2}$	0.125

The excitation system is using AC4A exciter as shown in Table A.4 [1].

**Table A.4 IEEE AC4A excitation model parameters [1]**

$K_A$	200.0
$T_A$	0.04
$T_B$	12.0
$T_C$	1.0
$V_{RMAX}$	5.64
$V_{RMIN}$	-4.53
$K_C$	0
$V_{IMAX}$	1.0
$V_{IMIN}$	-1.0

Prime mover and governor system is using non-reheat steam turbine. The parameters of prime mover and governor [113] is given in Table A.5.

**Table A.5 Non-reheat steam governor parameters [113]**

$droop$	5
$v_{ref}$	0.998
$L_{set}$	0.5
$T_T$	0.2
$T_{CH}$	0.3

The parameters of power system stabilizer are given in Table A.6 [1].

**Table A.6 Power system stabilizer parameters [1]**

$T_S$	0.03
$K$	20
$T_w$	2
$T_{1n}$	0.05
$T_{1d}$	0.02
$T_{2n}$	3
$T_{2d}$	5.4
$V_{Smin}$	-0.15
$V_{Smax}$	0.15

The network parameters in Section 2.4.3 Synchronous Machine is given in Table A.7.

**Table A.7 External network parameters**

	60
Ideal voltage source frequency (Hz)	60
Ideal voltage source amplitude (kV)	24
Resistance R1 (Ohms)	0.01

## **A.4 11-bus 4-machines network**

Synchronous machines are using the same parameter in A.1.

11-bus 4 machines network is a standard benchmark [1]. The parameters of 11-bus 4-machines network in Chapter 2, Chapter 3 and Chapter 3 are using the same system [1]. Transmission lines are using same values per unit length, parameters of different transmission line in 11-bus 4-machines [1] network are shown as follows:

**Table A.8 Transmission lines for 11-bus 4-machines system [1]**

Number of phases	3
Frequency for RLC specification	60
Resistance per unit length (Ohms/km)	$r_1 = 0.0001, r_0 = 0.0002$
Inductance per unit length (H/km)	$x_1 = 0.0001, x_0 = 0.0002$
Capacitance per unit length (F/km)	$b_1 = 0.00175, b_0 = 0.000875$
TL56, TL1011 length (km)	25
TL67, TL910 length (km)	10
TL78, TL89 length (km)	110

For transformers, nominal power is 900 MVA, frequency is 60 Hz, primary winding voltage is 20kV, secondary winding is 230kV, and connection is D1/Yg connection.

Grounding resistance for bus 7 fault is 0.001 (pu).

## **A.5 39-bus 10-machines network**

Synchronous machines are using the same parameter in A.1.



The 39-bus 10-machines in Chapter network is IEEE standard benchmark [114]. Based on typical line data in A.2, the approximate line length of the transmission line can be summarized as follows:

**Table A.9 Transmission lines length for 39-bus 10-machines [114]**

From Bus	To Bus	Length (km)
1	2	43.4838
1	39	26.45
2	3	15.9758
2	25	9.0988
3	4	22.5354
3	18	14.0714
4	5	13.5424
4	14	13.6482
5	6	2.7508
5	8	11.8496
6	7	9.7336
6	11	8.6756
7	8	4.8668
8	9	38.4054
9	39	26.45
10	11	4.5494
10	13	4.5494

13	14	10.6858
14	15	22.9586
15	16	9.9452
16	17	9.4162
16	19	20.631
16	21	14.283
16	24	6.2422
17	18	8.6756
17	27	18.3034
21	22	14.812
22	23	10.1568
23	24	37.03
25	26	34.1734
26	27	15.5526

For transformers, nominal power is 1000 MVA, frequency is 60 Hz, primary winding voltage is 22kV, secondary winding is 345kV, and connection is D1/Yg connection.

## **A.6 HVDC-MMC interconnection**

The simulation is operated by use of an aggregate MMC model [115].

The parameters of HVDC-MMC Interconnection (1000-MW, +/- 320 kV) [115] is a standard benchmark from CIGRE and details are given:

**Table A.10 Parameters of HVDC-MMC Interconnection system [115]**

Parameters	Values
Nominal system frequency/Hz	50
Converter rated power/MVA	10e9
Nominal primary voltage/V	400000
Nominal secondary voltage/V	333000
Number of power modules per arm	36
DC nominal voltage/V	640000
Power module capacitor/F	0.001758
Filter capacitor value/F	4.78e-5
Filter resistance value/ohm	1996
Filter frequency/Hz	1000
Startup resistance $R_{start}$ /ohm	400
Cable $R_{cable}$ /ohm	0.5
Cable $C_{cable}$ /F	0.0015
Grounding $R_g$ /ohm	100
Grounding $C_g$ /F	50e-9
$L_{arm}$ /H	0.0529
$R_{arm}$ /ohm	0.1663

Inverter Capacitance $R_{inv}/ohm$	0.001
Inverter Resistance $C_{inv}/F$	1.4650e-04
$L_1/H$	0.0001
$R_1/ohm$	0.01
Active Power Regulator $K_p^{Preg}$	0.167
Active Power Regulator $K_i^{Preg}$	1.0
Active Power Regulator Limits [ $MAX^{Preg}, MIN^{Preg}$ ]	[1.2, 0.8]
Reactive Power Regulator $K_p^{Qreg}$	0.167
Reactive Power Regulator $K_i^{Qreg}$	1.0
Reactive Power Regulator Limits [ $MAX^{Qreg}, MIN^{Qreg}$ ]	[0.25, -0.25]
Direct Voltage Regulator $K_p^{VDCreg}$	4
Direct Voltage Regulator $K_i^{VDCreg}$	100
Direct Voltage Regulator Limits [ $MAX^{VDCreg}, MIN^{VDCreg}$ ]	[2, -2]
Current Regulator $K_p^{Ireg}$	4
Current Regulator $K_i^{Ireg}$	100
Current Limits [ $MAX^{Ireg}, MIN^{Ireg}$ ]	[2, -2]