energies

MDPI

Article

FPGA-Based Real-Time Simulation of Dual-Port Submodule MMC–HVDC System

Zhao Jin [1,*], Yanjie Wu [2], Shuyuan Wang [1] and Bingda Zhang [1]

[1] The Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China; noahdream@163.com (S.W.); 1017234013@tju.edu.cn (B.Z.)
[2] State Grid Handan Electric Power Company, Handan 056035, China; 18222901154@163.com
* Correspondence: jinzaho@tju.edu.cn; Tel.: +86-182-2228-5930

Abstract: Aiming at the problems of the high switch numbers, complex working mechanisms, and complicated real-time simulation of modular multilevel converters (MMCs) composed of dual-port submodules, in this study, we designed a unified equivalent model of the multiple submodule network by analyzing the combination of parallel submodules in the bridge arm. The proposed model decouples the submodules that do not affect each other in the subnetwork calculation process, thereby reducing the number of prestored parameters in the subnetwork simulation. In the Xilinx Virtex-7 FPGA VC709 (Xilinx Corporation, San Jose, CA, USA) development board, we replaced the inline computation combined with the prestorage of parameters with the proposed equivalent model to optimize the execution unit structure and redesigned the FPGA-Based Real-Time Digital Solver (FRTDS). Taking the P-FBSM-based MMC–HVDC system as the simulation object, we performed a real-time simulation with a step size of 10 µs, which verified the effectiveness of the proposed model and the improvement in the hardware. We compared the results with the offline MATLAB/Simulink simulation results to verify the accuracy of the simulation.

Keywords: modular multilevel converter (MMC); real-time simulation; electromagnetic transient modeling; prestored parameters; field-programmable gate array (FPGA)

Citation: Jin, Z.; Wu, Y.; Wang, S.; Zhang, B. FPGA-Based Real-Time Simulation of Dual-Port Submodule MMC–HVDC System. *Energies* **2022**, *15*, 4624. https://doi.org/10.3390/en15134624

Academic Editor: Panos Kotsampopoulos

Received: 20 May 2022
Accepted: 22 June 2022
Published: 24 June 2022

1. Introduction

Energy and environmental problems are becoming increasingly severe, and the energy structure is transforming toward clean and low-carbon energy. Flexible DC transmission technology has many advantages in long-distance power transmission and large-scale renewable energy grid integration and thus is playing an important role in energy transformation [1–3]. Owing to their scalability, low harmonic content, and strong fault ride-through capability, MMCs have gradually become the mainstream flexible HVDC converter structure. The control and protection system has an important function in the secure and reliable operation of the power grid. The hardware-in-loop (HIL) test based on real-time simulation is used to verify the correctness of the protection devices' action and the effectiveness of the control strategy in HVDC flexible transmission [4–6]. As the transmission voltage level increases, the switching element groups sometimes need to be used in parallel. In this case, a new topology can be constructed with the same components to add new functions to the submodules. As such, researchers have designed an MMC topology that provides capacitor voltage self-balancing and bridge arm current-sharing [7–9]. The parallel full-bridge submodule (P-FBSM) topology [10] is a dual-port structure that reduces switching frequency and power loss and performs the sensorless operation of capacitor voltage, providing practicality and advantages in high-voltage MMC applications. The dual-port submodule has four outlets, and its current path is complex and changeable, requiring the Thevenin equivalent model, which is applicable to real-time simulations of the single-port submodule [11]. Previously proposed is the electromagnetic transient

simulation model suitable for the topology of the dual-port submodule, but this method requires iterative calculation and involves a large number of matrix operations, so it is not suitable for real-time simulation due to high calculation cost. The many high-frequency interrupted power electronic devices used in MMCs require a small simulation step size for real-time simulation, but a real-time simulation model that can reflect the diverse working modes and complex current paths of dual-port submodules has not been designed. A hardware platform with strong computing power is also required for real-time simulations of the MMC–HVDC system.

At present, the mainstream real-time simulators used in the electrical engineering field include RTDS and RT-LAB [12–14]. RT-LAB integrates the dynamic system mathematical model established by MATLAB/Simulink and performs real-time simulations. It is mainly used for the simulation of a single-power electronic device and less so for system-level simulation. The RTDS hardware used for computing tasks includes several racks and multiple processors for parallel computing to increase the simulation scale. However, the underlying computing hardware is still a serial device in essence, which leads to limitations in small-step simulation. In addition, the existing platform has strict hardware configuration requirements, so the equipment cost is too high for most practical situations [15]. The field-programmable logic gate array (FPGA) has a fully configurable parallel hardware structure, with distributed memory and deep pipeline structures, and also is low cost and small. As such, FPGA has gradually become the main hardware used for the real-time simulation of power systems [16]. A specialized calculation module was constructed on an FPGA based on the mathematical model of the power system, which improved the real-time simulation speed [17–19]. However, the dedicated modules often remain in an idle state due to limitations of the calculation process, resulting in a waste of FPGA resources. Zhang et al. [20] designed a real-time digital solver (FRTDS) based on FPGA and an instruction stream. FRTDS breaks the design concept of designing specific hardware circuits for each function involved in the simulated objects and provides a new idea for the real-time simulation of electromagnetic transients in power systems. FRTDS was designed for traditional AC power systems; thus, when used for real-time simulation of MMC–HVDC systems, it needs to be modified according to the characteristics of the calculation process. To perform real-time simulations of the dual-port submodule in MMC–HVDC systems, in this study, we optimized both the simulation models and the simulation platform. Section 2 introduces the basic structure of the FPGA-based real-time digital solver (FRTDS), focusing on the operation mechanism of the general computing components; Section 3 describes the method of replacing computation with storage, which is often used for real-time simulations to reduce the computation amount. In Section 4, we propose an equivalent model of the PFBSM subnetwork from the perspective of storage capacity optimization and analyze the computation amount. In Section 5, we describe appropriate improvements to the original FRTDS according to the calculation characteristics of the MMC–HVDC system; in Section 6, we verify the effectiveness and accuracy of the proposed simulation model and design method by describing a study case of the MMC–HVDC system; Section 7 outlines the studies conclusions.

2. FPGA-Based Real-Time Digital Solver (FRTDS)

The FPGA-Based Real-Time Digital Solver (FRTDS), which uses instruction stream-driven architecture, is to design a high reusability calculation unit based on the simulation method, the data address and the operation type of the calculation unit are given by instruction to realize the calculation. FRTDS was designed to simulate traditional AC power systems, adopting the node voltage analysis method. Its calculation process and commonly used arithmetic operations are shown in Figure 1. If each calculation process is designed as a specific function module, it not only wastes FPGA resources but also increases the simulation calculation time. According to the typical arithmetic operations of the calculation process, when the operating components of an FRTDS are designed, the following seven arithmetic operations are defined as basic operations: $Y = A \times B$, $Y = A + B$, $Y = A/B$,

Y = A × B + C × D, Y = A/B, Y = A × B/C, and Y = A × B/C + D; all the other arithmetic operations can be split into these basic operations. For example, Y = A × B/C + D/E + F can be split into two basic operations: Y = A × B/C + Y1 and Y1 = 1 × D/E + F. To save the DSP resources in an FPGA, all basic operations are implemented with two adders, two multipliers, and one divider in the general operation component.

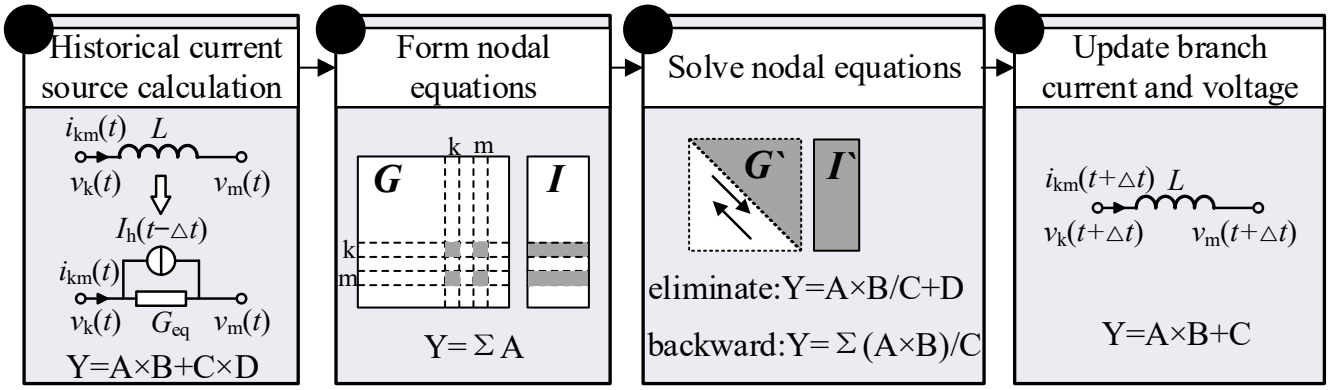


Figure 1. Computational features of node analysis.

FRTDS describes the data memory address and the selected word of the basic operation involved in completing a certain operation in the specified instruction format and saves them in the program's memory. The arithmetic component takes the instructions from the program memory at the specified operating frequency, and transmits them to the read, write, and selection controllers. The read controller obtains the operation data from the data memory through the multiport read-and-write circuit, the arithmetic unit performs calculation according to the operation formula determined by the select controller, and the write controller stores the calculation result in the data memory through the multiport read-and-write circuit. In each step, the arithmetic components continuously and cyclically execute the calculation process according to the instructions; the specific structure of the universal computing component is shown in Figure 2. To enable the arithmetic components to work in a pipeline, an instruction buffer queue is added to the read, write, and selection controllers to implement a delayed output of data memory addresses and selection words. Several arithmetic components can be formed into a microprocessor core with stronger computing power. The arithmetic components in the core communicate through shared memory, and the microprocessor cores communicate through the straight-through wires of the arithmetic components via message passing. When allowed by available hardware resources, as many microprocessor cores as possible should be built so that the FRTDS will have powerful parallel computing capabilities. The data exchange between the microprocessor cores adopts the way of "hand in hand and data pipeline", which means that two adjacent microprocessor cores share data storage units while two non-adjacent microprocessor cores exchange data through the data pipeline.

When used for HIL testing of real equipment, the time when the control signals of the switching elements such as IGBTs sent by the controller reach the FRTDS is uncertain; therefore, these signals are first stored in the backup data memory. At the start of each simulation step, the backup data memory becomes the running data memory of the arithmetic components, and the external information is provided to the arithmetic components. The output data of the arithmetic components cannot be directly provided to the peripheral equipment. First, the output data are saved in the backup data memory, and at the beginning of each simulation step, the backup data memory is changed to the running data memory of the peripheral equipment to ensure that the voltage and current peripherals are used at the same time. The ping-pong operation of the data memory improves the working efficiency of the real-time simulation platform.

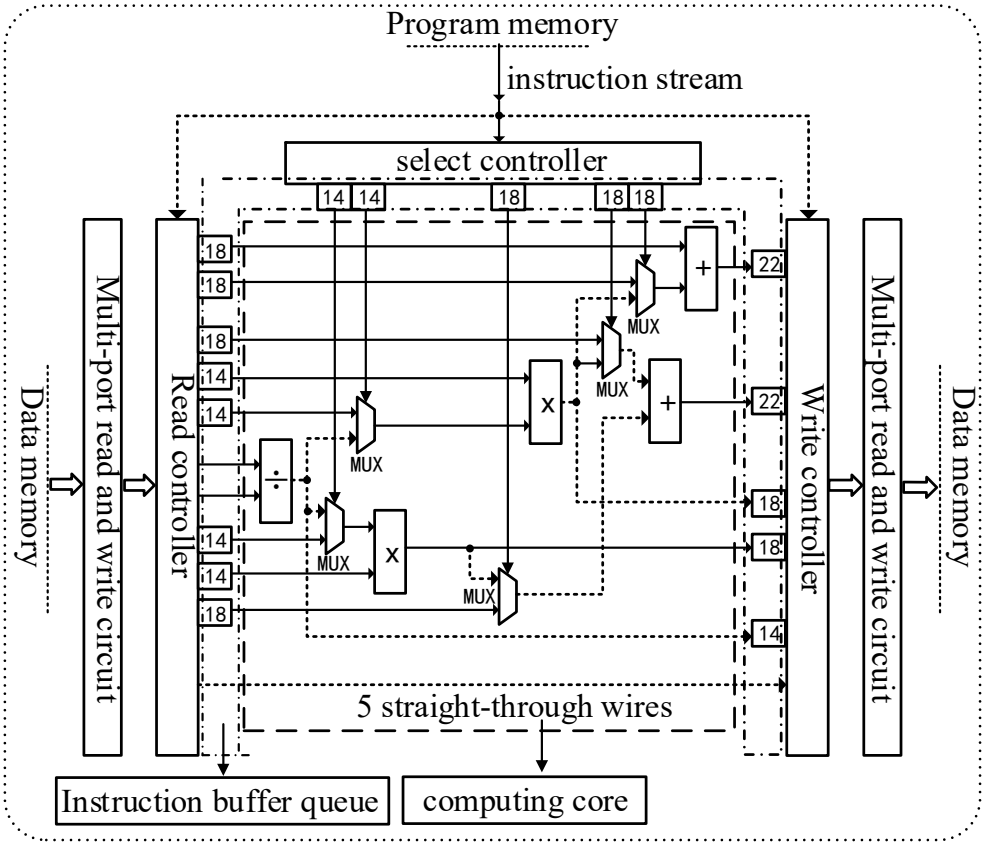


Figure 2. Structure of universal arithmetic components.

During the simulation calculations, the switching element is modeled as a binary resistance, and the nonlinear elements are calculated in a piecewise linear manner. The existence of a switching element and a nonlinear element leads to many simulation parameters having various values. To facilitate the selection of the current value of a multivalued parameter, all the values of a multivalued parameter are stored as an array, in which each element represents the value of a multivalued parameter for a certain case. The information describing the multivalued parameters is stored in the boot word, which includes the starting address of the multivalued parameter array, the address of the affect word, and the decoding method. After obtaining the state of the affect word, FRTDS calculates the address offset according to a certain decoding method and adds this offset to the starting address to obtain the real address of the current value of the multivalued parameter. The specific process is shown in Figure 3. The affect word is divided into internal and external affect words. The state of the internal affect word is determined by the calculation result of the arithmetic component, and the state of the external influence word is determined by the input of the external device.

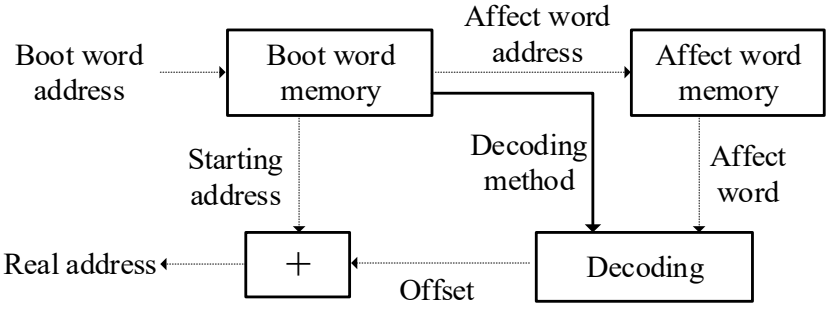


Figure 3. Indirect addressing workflow.

The graphical simulation software provided by the FRTDS includes the graphical modeling of simulation objects and instruction-level assignment of computing tasks. The graphical modeling of the simulation object automatically provides the specific simulation calculation process according to the characteristics of the electrical equipment and their connection relationships. The instruction-level assignment of computing tasks automatically generates a directed acyclic graph describing the dependencies of computing tasks according to the positions of variables in the computing expression. For each single clock cycle, the task with the least amount of time slack among the ready computing tasks is found, and the cost to arrange it in every arithmetic component is calculated. Then, the task is arranged in the arithmetic component with the least cost until all computing tasks are scheduled, or the arithmetic component cannot take on more computing tasks.

3. Method of Replacing Inline Computation with Parameters Prestorage

Real-time simulations require the calculation time to be strictly synchronized with real-time; therefore, the computing power per unit time of the simulation platform is determined, so an algorithm with a small calculation amount must be selected to ensure that the hardware has sufficient ability to complete the calculation task within the expected time. Some calculation tasks can be calculated offline and then stored in the simulation platform. As such, this method of replacing computation with storage reduces the amount of calculation. Large-scale simulation objects are usually divided into multiple subnetworks, and an upper-layer network can be created after the external equivalent circuit of each subnetwork is obtained. After solving the upper-layer network, which has fewer nodes, the input variables of each subnetwork port are obtained, and the internal subnetworks are solved. Some calculation tasks during the equivalence and solution process of each subnetwork are appropriate for being replaced with prestored parameters. A discrete linear multiport network is equivalent to k Thevenin equivalent ports and m Norton equivalent ports. The input variable of the Thevenin ports is the current vector $\boldsymbol{i}_{\mathrm{A}} = [i_1 \ \ldots \ i_k]^T$, and the input variable of the Norton equivalent ports is the current vector $\boldsymbol{u}_{\mathrm{B}} = [i_1 \ \ldots \ i_m]^T$. The equivalent voltage source vector $\boldsymbol{U}_{\mathrm{eq}}$ and equivalent current source vector $\boldsymbol{I}_{\mathrm{eq}}$ in the external equivalent circuit of the subnetwork can be expressed as a linear combination of the historical power vector $[\boldsymbol{U}_h \ \boldsymbol{I}_h]$ and the independent power vector $[\boldsymbol{U}_s \ \boldsymbol{I}_s]$ inside the subnetwork. The solution formula is:

$$\begin{bmatrix} \boldsymbol{U}_{\mathrm{eq}} \\ \boldsymbol{I}_{eq} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_h & \boldsymbol{R}_h \\ \boldsymbol{G}_h & \boldsymbol{H}_h \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_h \\ \boldsymbol{I}_h \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_s & \boldsymbol{R}_s \\ \boldsymbol{G}_s & \boldsymbol{H}_s \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_s \\ \boldsymbol{I}_s \end{bmatrix} \tag{1}$$

According to the network connection relationship, after simultaneously solving the port input variables of each subnetwork, the $\boldsymbol{x}$ to be solved in each subnetwork can be further solved as follows:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{i}_{\mathrm{A}} \\ \boldsymbol{u}_{\mathrm{B}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_s \\ \boldsymbol{I}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_h \\ \boldsymbol{I}_h \end{bmatrix} \tag{2}$$

In formula (2) $[\boldsymbol{A}_1 \ \boldsymbol{A}_2]$ is the coefficient matrix associated with the input variables, $[\boldsymbol{B}_1 \ \boldsymbol{B}_2]$ is the coefficient matrix associated with the independent sources, and $[\boldsymbol{C}_1 \ \boldsymbol{C}_2]$ is the coefficient matrix associated with the historical sources. $\boldsymbol{K}_h$, $\boldsymbol{R}_h$, $\boldsymbol{G}_h$, $\boldsymbol{H}_h$, $\boldsymbol{K}_s$, $\boldsymbol{R}_s$, $\boldsymbol{G}_s$, and $\boldsymbol{H}_s$ in (1), and $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, $\boldsymbol{C}_1$, and $\boldsymbol{C}_2$ in (2) are sets of known constants after the element states within the subnetwork are determined, which can be precalculated and stored in the hardware of the simulation platform. When calculating in each step, the prestored coefficients are removed and used in the vector multiplication and addition operations of Equations (1) and (2) to complete the equivalence and solution of the subnetwork. The state variable method can also be used in the subnetwork equivalence and solution process. The voltage at the port of the subnetwork is taken as the external input vector of the subnetwork, the current at the port of the subnetwork is taken as the output vector of the subnetwork, and the independent voltage source or current source inside the subnetwork is taken as

the internal input vector of the subnetwork. Then, the state and output equations of the subnetwork can be written as:

$$\begin{cases} \dot{\boldsymbol{x}}_a = \boldsymbol{A}_a \boldsymbol{x}_a + \boldsymbol{B}_a \boldsymbol{w}_a + \boldsymbol{E}_a \boldsymbol{u} \\ \boldsymbol{i}_a = \boldsymbol{C}_a \boldsymbol{x}_a + \boldsymbol{D}_a \boldsymbol{w}_a + \boldsymbol{F}_a \boldsymbol{u} \end{cases} \tag{3}$$

where $\boldsymbol{i}$ is the current vector at the port of the subnetwork, $\boldsymbol{u}$ is the voltage vector at the port of the subnetwork, $\boldsymbol{s}$ is the independent power vector inside the subnetwork, and $\boldsymbol{x}$ is the state variable vector of the subnetwork. Use the implicit Euler method to differentiate Equation (3); then, the following formula is derived:

$$\begin{cases} \boldsymbol{x}_a(t) = \boldsymbol{A}_a^* \boldsymbol{x}_a(t - \Delta t) + \boldsymbol{B}_a^* \boldsymbol{w}_a(t) + \boldsymbol{E}_a^* \boldsymbol{u}(t) \\ \boldsymbol{i}_a(t) = \boldsymbol{C}_a^* \boldsymbol{x}_a(t - \Delta t) + \boldsymbol{D}_a^* \boldsymbol{w}_a(t) + \boldsymbol{F}_a^* \boldsymbol{u}(t) \end{cases} \tag{4}$$

Equation (4) shows that the state variable and output vectors of the subnetwork at time t can also be expressed as the linear combination of the state vector of the subnetwork at time $t - \Delta t$, the independent power vector in the subnetwork at time t, and the voltage vector at the port of the subnetwork at time t. Combining the first two terms of the right-hand side of the output equation in Equation (4), the Norton equivalent circuit expression of the subnetwork is obtained:

$$\boldsymbol{i}_a(t) = \boldsymbol{P}_a(t) + \boldsymbol{F}_a^* \boldsymbol{u}(t) \tag{5}$$

where $\boldsymbol{P}_a(t) = \boldsymbol{C}_a^* \boldsymbol{x}_a(t - \Delta t) + \boldsymbol{D}_a^* \boldsymbol{w}_a(t)$ is the same as the multiport equivalent method based on a discrete network. The coefficient matrices $\boldsymbol{A}^*$, $\boldsymbol{B}^*$, $\boldsymbol{C}^*$, $\boldsymbol{D}^*$, $\boldsymbol{E}^*$, and $\boldsymbol{F}^*$ are precalculated and stored in the simulation hardware, and in each step, the vector multiplication and addition operations are performed according to Equations (4) and (5) to complete the equivalence of the subnetwork and update the state variables.

4. P-FBSM Subnetwork Model

The MMC topology based on P-FBSM is shown in Figure 4. Each bridge arm of the converter is composed of a bridge arm reactor L_{arm} and N parallel full-bridge submodules (PFBSM) connected in series. The upper and lower bridge arms of each phase form a phase unit, and the MMC bridge arm is still a single-port structure by shorting the two terminals of sub-modules at the head and end of the bridge arm. The P-FBSM is obtained by symmetrically inverting the full-bridge submodule, which is a typical dual-port submodule. The binary resistor is modeled, the capacitor is then differentiated by the Euler method, and the accompanying PFBSM circuit shown in Figure 4 is obtained. Using a binary resistor to model the entire switch group of the IGBT and diode in parallel with the PFBSM and differentiating the capacitor by the implicit Euler method, the companion circuit shown in Figure 4 can be obtained. For a submodule, the three conduction modes for the switch groups on the left and right sides of the capacitor are: positive (G1, G2 on), negative (G3, G4 on), and parallel (G1, G4 or G2, G3 on).

When simulating the MMC converter composed of a P-FBSM, if the node voltage analysis is directly used to solve the problem, owing to the large number of neutron modules in the bridge arm, the dimensions of the node voltage equation are large, which makes it difficult to quickly solve the problem. Therefore, the bridge arm needs to be divided into multiple subnetworks, and the method of replacing computation with storage needs to be used to perform real-time simulation. The currently used external equivalent circuit of the PFBSM subnetwork still has a dual-port structure, as shown in Figure 5. Each subnetwork contains n submodules; u_{ex1}, u_{ex2}, u_{ex1}, and u_{ex2} are the input voltages of the subnetwork ports; G_{12}, G_{13}, G_{14}, G_{23}, G_{24}, and G_{34} are the six equivalent conductances in the equivalent model; and J_{in1}, J_{in2}, J_{in3}, and J_{in4} are equivalent current sources. The equivalent current source vector $\boldsymbol{J} = [J_{in1}\ J_{in2}\ J_{in3}\ J_{in4}]^T$ can be obtained by the linear combination of

the historical value of each capacitor voltage in the subnetwork according to coefficient matrix $\boldsymbol{C}$:

$$\boldsymbol{J}(t) = \boldsymbol{C}_{4\times n}\cdot\boldsymbol{u}_c(t-\Delta t) \tag{6}$$

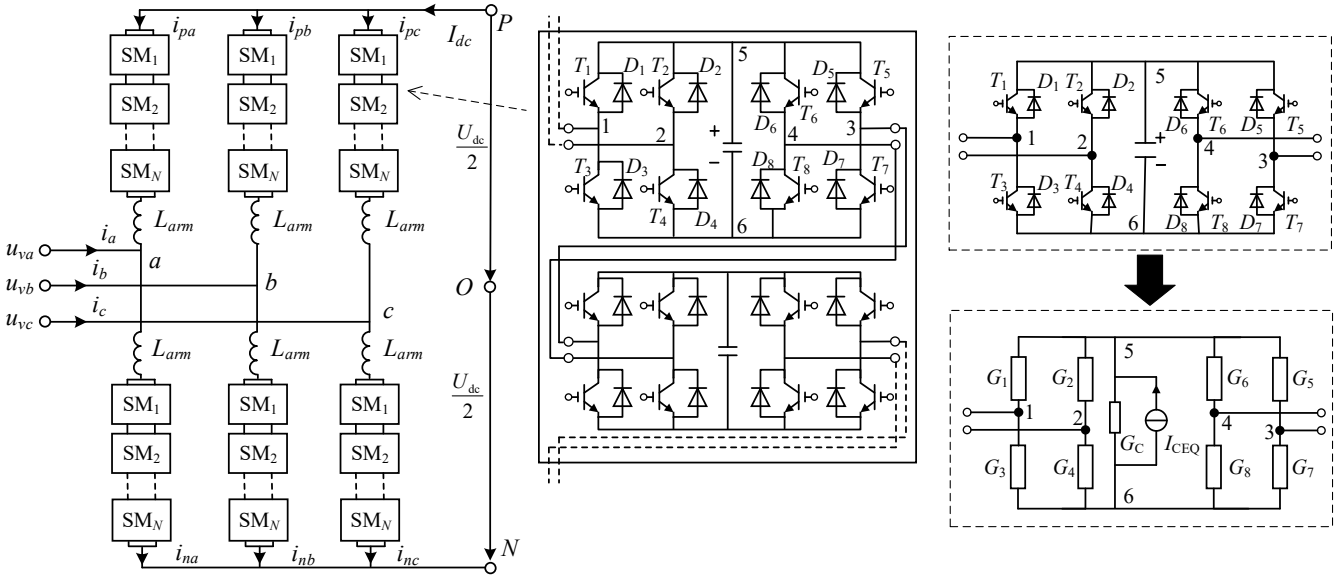


Figure 4. Topology of MMC based on P-FBSM.

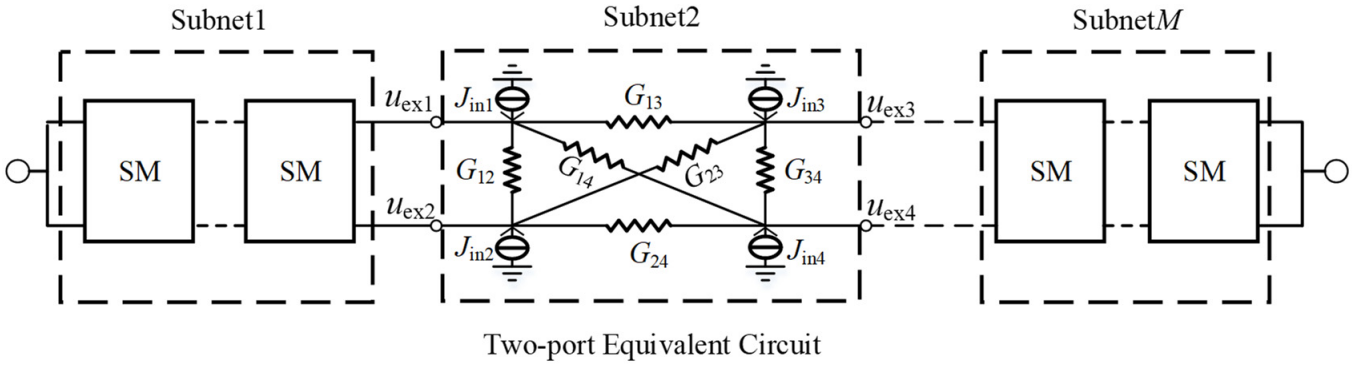


Figure 5. P-FBSM dual-port equivalent circuit.

By connecting the equivalent circuits of each subnetwork and eliminating the internal nodes, the equivalent circuit of the entire bridge arm is obtained, then the node voltage equation of the whole simulation object can be formed and solved. After finding the input voltage at the interface of each subnetwork, the voltage of each capacitor in the subnetwork can be calculated and updated. Each capacitor voltage can be expressed as the linear combination of the voltage vector $\boldsymbol{u}_{ex} = [u_{ex1}\ u_{ex2}\ u_{ex1}\ u_{ex2}]^T$ and the historical value of each capacitor voltage in the subnetwork:

$$\boldsymbol{u}_c(t) = \boldsymbol{A}_{n\times n}\cdot\boldsymbol{u}_c(t-\Delta t) + \boldsymbol{B}_{4\times n}\cdot\boldsymbol{u}_{ex}(t) \tag{7}$$

The coefficient matrices $\boldsymbol{A}_{n\times n}$, $\boldsymbol{B}_{4\times n}$, $\boldsymbol{C}_{4\times n}$ and the six equivalent conductances of the subnetwork in Equations (6) and (7) can be prestored. The coefficients in these matrices are related to the conduction mode of the switch groups on the left and right sides of the capacitor in each submodule. However, an analysis of the operation mode shows that if the on-off of the IGBT is controlled by the control pulse during normal operation or if only the on-off of the diode is controlled during the blockade of the whole station after startup or

failure, only five connection modes will exist between two adjacent submodules, as shown in Figure 6. The number of parameters that each subnet needs to prestore is:

$$Mem = 3^2 \times 5^{n-1} \times (n^2 + 8n + 6) \tag{8}$$

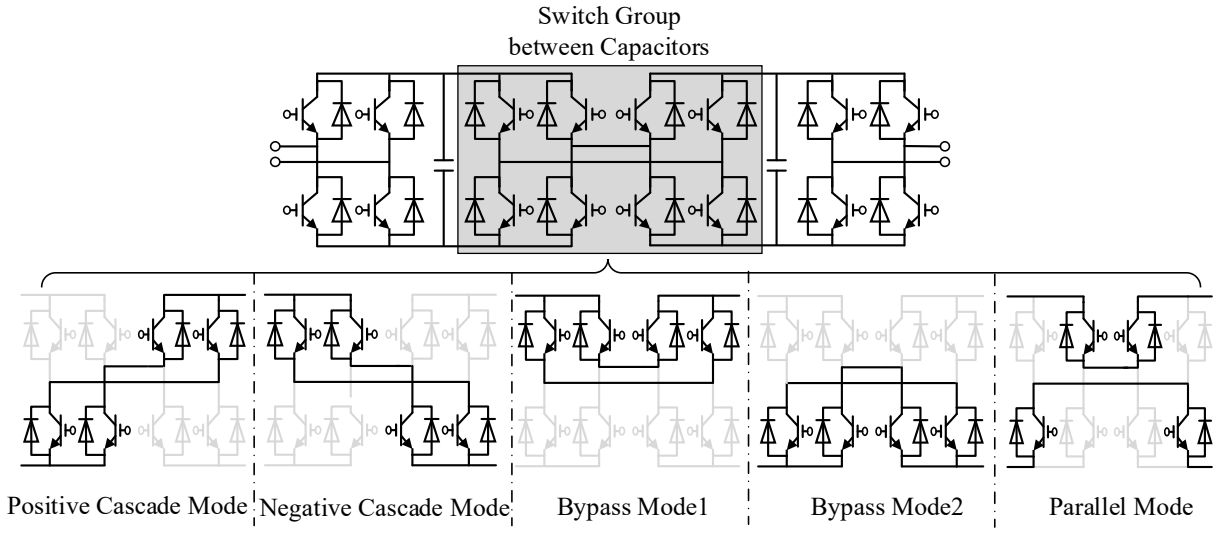


Figure 6. Connection (mode) of adjacent submodules.

The number of prestored parameters exponentially increases as the number of submodules in the subnetwork increases. Practically applying these parameters in an FRTDS without optimizing them is difficult. Given the parallel state of the newly added submodules in PFBSM–MMC, the concept of a submodule segment was proposed [21], in which the parallel submodule combination is regarded as a submodule segment. The segmented structure of a bridge arm is shown in Figure 7a. The submodule segment in the bridge arm has four different working states, as shown in Figure 7b: positive voltage input, negative voltage input, and two bypass states. A single submodule is also regarded as a submodule segment, and the number of submodules in the segment dynamically changes with the control signal.

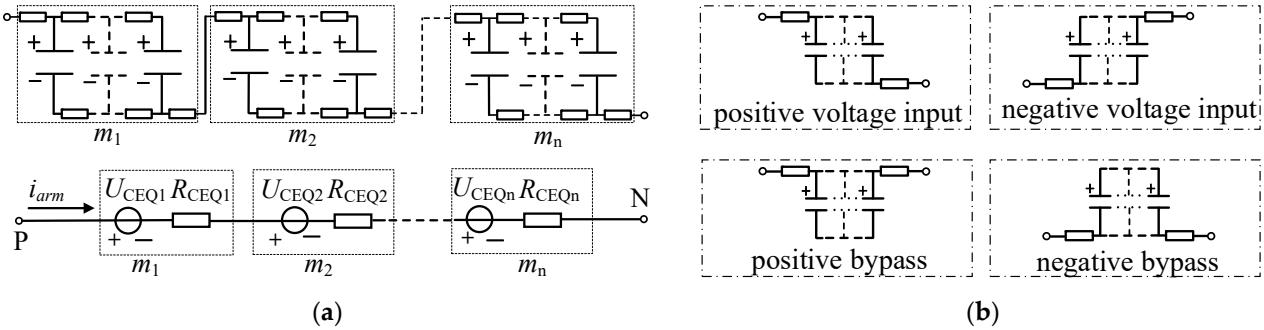


Figure 7. Submodule segment in PFBSM–MMC bridge arm: (**a**) segmented structure and equivalent model of bridge arm; (**b**) four working states of the submodule segment.

By considering the submodule segment as a subnetwork, it is equivalent to a single-port Thevenin circuit. Then, the entire bridge arm is composed of the Thevenin equivalent circuits of n submodule segments in series. The resistance R_{ceq} of a Thevenin equivalent circuit is related to the working mode and length of the submodule segment. The voltage source U_{ceq} in a Thevenin equivalent circuit can be expressed as the linear combination of the historical value of the capacitor voltage of each submodule in the segment:

$$U_{ceq}(t) = \boldsymbol{C}_{1\times l} \cdot \boldsymbol{u}_c(t - \Delta t) \tag{9}$$

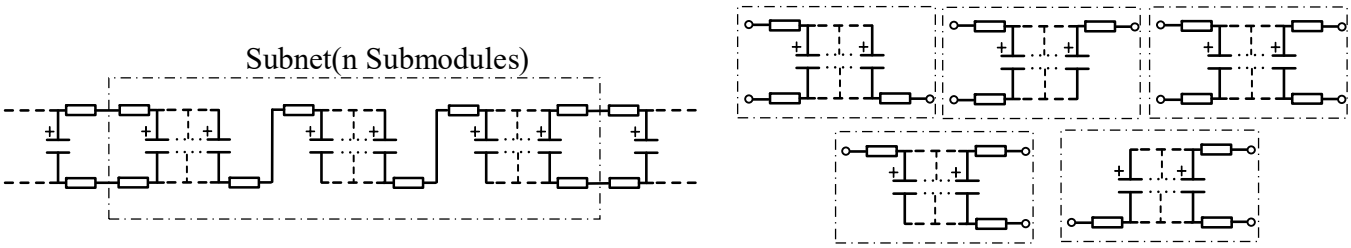
The value of element c_i in coefficient matrix $\boldsymbol{C}_{1\times l}$ of the historical capacitance voltage is related to the working mode of the submodule segment, the length of the submodule segment, and the position of the ith submodule in the segment. The Thevenin equivalent circuit of the whole bridge arm can be obtained by summing the equivalent voltage source and the equivalent resistance of each submodule segment in the bridge arm. Then, the upper-layer network node equation of the whole system can be formed and solved. After calculating the bridge arm current i_{arm}, it is regarded as the input variable of each subnetwork. The submodule capacitor voltage in each segment can be updated by the linear combination of the current submodule capacitor voltage in each segment and i_{arm}:

$$\boldsymbol{u}_{\mathrm{c}}(t) = \boldsymbol{A}_{l\times l}\cdot\boldsymbol{u}_{\mathrm{c}}(t-\Delta t) + \boldsymbol{B}_{l\times 1}\cdot i_{arm}(t) \tag{10}$$

The value of element a_{ij} in coefficient matrix $\boldsymbol{A}_{l\times l}$ of the historical capacitance voltage is determined by four factors: the working mode of the submodule segment, the length of the submodule segment, the position of the ith submodule in the segment, and the position of the jth submodule in the segment. The value of element b_i in coefficient vector $\boldsymbol{B}_{1\times l}$ of the bridge arm current is determined by three factors: the working mode of the submodule segment, the length of the submodule segment, and the position of the ith submodule in the segment. When the length of the module group is l, $\boldsymbol{A}_{l\times l}$, $\boldsymbol{B}_{1\times l}$, $\boldsymbol{C}_{1\times l}$, and R_{ceq} occupy $4l^2$, $4l$, $4l$, and 4 storage spaces, respectively, where l ranges from 1 to the total number of submodules in bridge arm m. When the submodule segment is used as the subnetwork, the number of parameters that need to be prestored is:

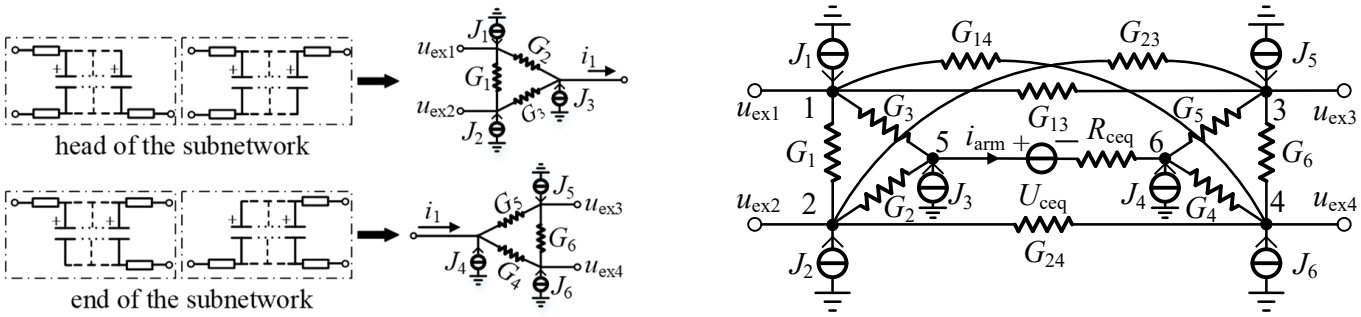
$$Mem = \sum_{l=1}^{m}\left(4l^2+4l+4l+4\right) = \frac{4}{3}\mathrm{m}^3 + 6\mathrm{m}^2 + \frac{26}{3}\mathrm{m} \tag{11}$$

At this point, the number of prestored subnetwork parameters quadratically increases with the number of submodules in the subnetwork, which are substantially fewer compared to the number of subnetwork parameters calculated according to Equation (3). However, if the submodule segment is directly used as the subnetwork, the calculation process changes after the segmentation situation changes, so the control instructions of the FRTDS involve conditional branching. The submodule segmentation status is determined by the control signal from the controller, and the user cannot obtain the segmentation status in real-time, which prevents switching the control instructions in real-time. To ensure that the instructions in FRTDS remain unchanged, only a fixed number of adjacent submodules in the topology can be delineated as a subnetwork, as shown in Figure 8a. Depending on the switch state, a subnetwork may contain several submodule segments or all submodules in the subnetwork may be connected in parallel and belong to one submodule segment. The parallel combination of submodules in the subnetwork, in addition to the four submodule segment forms in Figure 7b, has five added forms, as shown in Figure 8b. The equivalent circuit for the parallel combination of three-terminal submodules appears at the head or end of the subnetwork. When all the submodules in the subnetwork are connected in parallel, the entire subnetwork is a dual-port structure. The equivalent circuit for the parallel combination of three-terminal submodules appearing at the head or end is shown in Figure 9a. Combining the single-ended Thevenin equivalent circuit of the submodule segment and the dual-port equivalent circuit of all submodules in parallel, the P-FBSM subnet equivalent model, shown in Figure 9b, is finally obtained. When solving the model, first, the values of each parameter in the figure are obtained by looking up the table or calculating according to the current operation of the subnetwork, then internal nodes 5 and 6 can be eliminated; finally, the model is simultaneously solved with other subnetworks. When the subnetwork does not contain a certain type of submodule parallel structure at the current time, the unified model of six nodes remains unchanged, and the part of the model that reflects the parallel combination of this type takes the maximum value of the conductance and the minimum value of the resistance.



(**a**) (**b**)

Figure 8. Subnetwork divided by a fixed number of submodules: (**a**) connection of submodules in the subnetwork; (**b**) added submodule parallel combination forms.



(**a**) (**b**)

Figure 9. Equivalent model considering the parallel combination of submodules: (**a**) equivalent circuit of three-terminal submodule parallel combination; (**b**) unified equivalent model of PFBSM subnetworks.

The data storage capacity of the equivalent model in Figure 9b is the sum of the prestored parameters in the case of a single-port Thevenin circuit, two three-terminal equivalent circuits, and a dual-port equivalent circuit with all submodules connected in parallel. The storage capacity of a single-port Thevenin circuit can be calculated by Equation (11). For a subnetwork with n submodules, when all the submodules are connected in parallel, the equivalent current sources J_1, J_2, J_5, and J_6 are calculated by Equation (6), and the voltage of each capacitor in the subnetwork is updated by Equation (7). When the number of submodules in the subnetwork is determined, each element in the coefficient matrix $\boldsymbol{A}_{\mathrm{n}\times\mathrm{n}}$, $\boldsymbol{B}_{4\times\mathrm{n}}$, $\boldsymbol{C}_{4\times\mathrm{n}}$, and the six equivalent conductances $G_1(G_{12})$, G_{13}, G_{14}, G_{23}, G_{24}, and $G_6(G_{34})$ in the subnetwork are all definite values; the storage capacity is $n^2 + 8n + 6$.

For a three-terminal structure, taking the head as an example, the update of the capacitor voltage of the submodule in the parallel combination can be calculated by:

$$\boldsymbol{u}_{\mathrm{c}}(t) = \boldsymbol{A}'_{l\times l}\cdot\boldsymbol{u}_{\mathrm{c}}(t-\Delta t) + \boldsymbol{B}'_{l\times 2}\cdot[u_{\mathrm{ex1}}(t)u_{\mathrm{ex2}}(t)]^T + \boldsymbol{C}'_{l\times 1}i_{arm} \tag{12}$$

which needs to store the historical capacitor voltage value coefficient matrix $\boldsymbol{A}'_{l\times l}$, the port input voltage coefficient matrix $\boldsymbol{B}'_{l\times 2}$, and the bridge-arm current coefficient matrix $\boldsymbol{C}'_{l\times 1}$. The equivalent current source vector $\boldsymbol{J} = [J_1\ J_2\ J_3]$ is obtained by the linear combination of the capacitor voltages of the submodule in parallel combination, for which the voltage coefficient matrix $\boldsymbol{D}'_{3\times l}$ and the three conductances G_1, G_2, and G_3 need to be stored. The value of l may range from 1 to n, and two parallel combinations of submodules corresponding to this model are possible. In summary, the total storage capacity of the three terminals at the head is $\frac{2}{3}n^3 + 7n^2 + \frac{37}{3}n$, and the total storage capacity of the three terminals at the end is the same as that of the head end. The number of prestored parameters required for the equivalent model of the six-node P-FBSM subnetwork is:

$$Mem = \left(\frac{4}{3}n^3 + 6n^2 + \frac{26}{3}n\right) + 2 \times \left(\frac{2}{3}n^3 + 7n^2 + \frac{37}{3}n\right) + \left(n^2 + 8n + 6\right) = \frac{8}{3}n^3 + 21n^2 + \frac{124}{3}n + 6 \quad (13)$$

When the total number of submodules in the bridge arm is N, suppose the subnetwork contains n submodules, and the entire bridge arm is divided into N/n subnetworks. The amount of calculation for elimination and back-substitution between two adjacent subnetworks is fixed as E_1, which needs to be performed $(N/n - 1)$ times. Inside the P-FBSM model, each conductance value can be obtained by looking up the table. However, the six equivalent injection current sources J_1–J_6, the equivalent voltage source, and the resistance of the single-port Thevenin equivalent circuit all need to be calculated according to the prestored parameters. To ensure that the calculation formula remains unchanged under the model's various working states, the single-port Thevenin equivalent circuit part of the model is obtained by adding the single-port Thevenin equivalent circuits to the n submodule segments. Only when all the submodules in the subnetwork are connected in series will n submodule segments be formed. In other cases, the number of submodules in each segment is determined by the actual segmentation situation; then, the resistance of the Thevenin equivalent circuit is obtained by looking up the table and the power supply of the Thevenin equivalent circuit is calculated according to the prestored parameters. The resistance of the Thevenin equivalent circuit in the nonexistent segment is set to the minimum value of 10^{-7} Ω. The calculation formula of this part of the model is:

$$\begin{aligned} R_{\text{ceq}} &= \sum_{i=1}^{n} R_{\text{ceq}i} \\ U_{\text{ceq}} &= \sum_{i=1}^{n} C_i^{''} u_i(t - \Delta t) \end{aligned} \quad (14)$$

According to the two operational formulas of A × B + C × D and A + B + C + D, the calculation amount is determined. The calculation amount of Equation (13) is $0.75n$. The calculation formulas for the six equivalent injection current sources are the same:

$$J_h = \sum_{i=1}^{n} D_i^{''} x_i(t - \Delta t), \quad h = 1, 2, \cdots 6 \quad (15)$$

The amount of calculation for Formula (14) is $3n$, and the updated formula of the internal capacitor voltage of the subnetwork is:

$$\begin{aligned} x_i(t) = &\sum_{j=1}^{n} A_j^{''} \times x_j(t - \Delta t) + B^{(u_{\text{ex1}})} \times u_{\text{ex1}}(t) + B^{(u_{\text{ex2}})} \times u_{\text{ex2}}(t) + B^{(u_{\text{ex3}})} \times u_{\text{ex3}}(t) \\ &+ B_{\text{LT}}^{(u_{\text{ex4}})} \times u_{\text{ex4}}(t) + B^{(i_{\text{arm}})} \times i_{\text{arm}}(t), \quad i = 1, 2, \cdots, n \end{aligned} \quad (16)$$

The amount of calculation of Formula (15) is approximately $0.5n^2 + 3n$. In addition to the parameter calculation in the model, the subnetwork six-node model needs to eliminate the two internal nodes, 5 and 6, with a fixed amount of calculation E_2. In summary, the total calculation amount for a bridge arm is:

$$Cal = (0.5n^2 + 6.75n + E_2) \times \frac{N}{n} + (\frac{N}{n} - 1) \times E_1 = N \times (0.5n + \frac{E_1 + E_2}{n}) + 6.75N - E_1 \quad (17)$$

Equation (17) shows that when the number of submodules n in each subnetwork is determined, the amount of calculation linearly increases with the total number of submodules in the bridge arm. When the total number N of submodules in the bridge arm is determined by taking the derivative of n in Formula (17), its value is smallest when $n = \sqrt{2(E_1 + E_2)}$. In actual simulations, a close integer number of submodules can be used to divide the subnetwork of the bridge arm. After division, the number of prestored param-

eters of the subnetwork needs to be calculated. If they cannot be stored, then the number of submodules in the subnetwork should be reduced to the amount that can be stored.

5. Adaptive Transformation of FRTDS Platform

The original FRTDS is mainly oriented to traditional AC systems, and the amount of calculation is mainly due to the formation and solution of the node voltage equation. The data dependencies between the calculation tasks are complex, and the data flow is heavily coupled. For example, when forming the injection current source for the node equation, the injection current source of a node needs to sum the historical current sources on all its related branches. When a node is eliminated, the self-admittance and mutual-admittance of multiple adjacent nodes are affected. This results in the diverse arithmetic operations and the data reading and writing processes being irregular when FRTDS performs simulation, necessitating read-and-write address instructions and operation type selection instructions at every moment to control the data flow and arithmetic operation selection. When using FRTDS for PFBSM–MMC simulation, the unified subnetwork model proposed in Section 2 is used, in which the external equivalent and internal solutions of the bridge arm adopt the matrix–vector multiplication operation. The execution process of this calculation is shown in Figure 10. The data flow direction in the inner and outer loops is determined, and only the fixed operation formula $Y = \Sigma(A \times B + C \times D)$ is executed. By storing the data involved in the operation in the storage area in order, sequential reading and writing can be achieved.

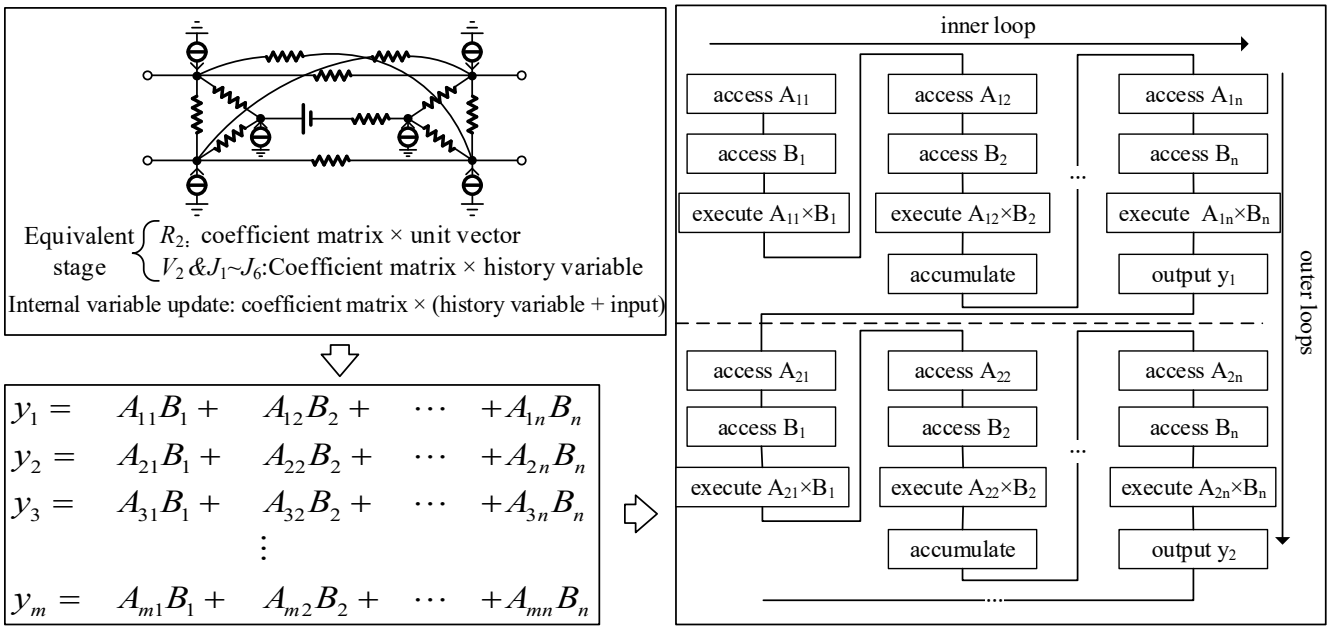


Figure 10. Computational features of matrix–vector multiplication.

The multiply accumulating operation used by matrix–vector multiplication accounts for a large proportion of the whole calculation process, and a dedicated execution unit was designed to perform this part of the calculation. The floating-point adder needs five clock cycles to obtain the calculation result. To ensure the data input involved in the accumulation is continuous, the floating-point number is converted into a fixed-point number, and the fixed-point number adder is used to complete the accumulation calculation in one clock cycle. The completion of each accumulation is determined by device S1 with a 1-bit control signal. The dedicated execution unit does not need a select controller to select the operation type of each clock. The instruction format used to drive the dedicated execution unit to run is addr(A, B, C, D, Y) + ctrl. The input and output data are arranged in an orderly manner according to the inner and outer loops, and the coefficient address

continuously increases from the first address during the entire calculation process. In the inner loop, the data address is continuously increased from the first address, and the value is refetched from the first address after each inner loop. The output address is increased from the first address after each inner loop; the control command outputs 1 in the inner loop and 0 after each inner loop completes. Therefore, the instruction storage area dedicated for executing unit instruction can store only one start instruction, start(AS,BS,CS,DS,YS) +m,n; that is, the start address of each port and the number of inner loops n and the number of outer loops m. When the coefficients involved in matrix–vector multiplication are multivalued parameters, each instruction only needs to calculate the offset once. An instruction loop unit, shown in Figure 11, was designed to generate the instructions required in the calculation process of the dedicated execution unit. The +1 and −1 operations in the instruction loop unit consume one clock; thus, a delay of one clock needs to be set for the data selected in synchronization. Channels A and C are coefficient channels, and channels B and D are data channels. The instruction loop unit reduces the storage resource consumption of the instruction storage area of the dedicated execution unit, leaving FRTDS more storage resources for prestored parameters. When forming the microprocessor core, several universal arithmetic components are replaced with dedicated execution units, thereby reducing the resource consumption of each microprocessor core and enabling more microprocessor cores to be placed in the FPGA. The computing tasks performed by the general-purpose computing components and the dedicated execution units in the same microprocessor core still communicate in a shared memory manner. The group of multiply accumulating operations is not split and is handled by a dedicated execution unit, which does not involve inter-core communication. After the calculation of each step starts, according to all the switch states of the current bridge arm, combined with the current direction of the bridge arm, the parallel combination mode of every submodule segment in each subnetwork and the number of submodules in the segment are determined.

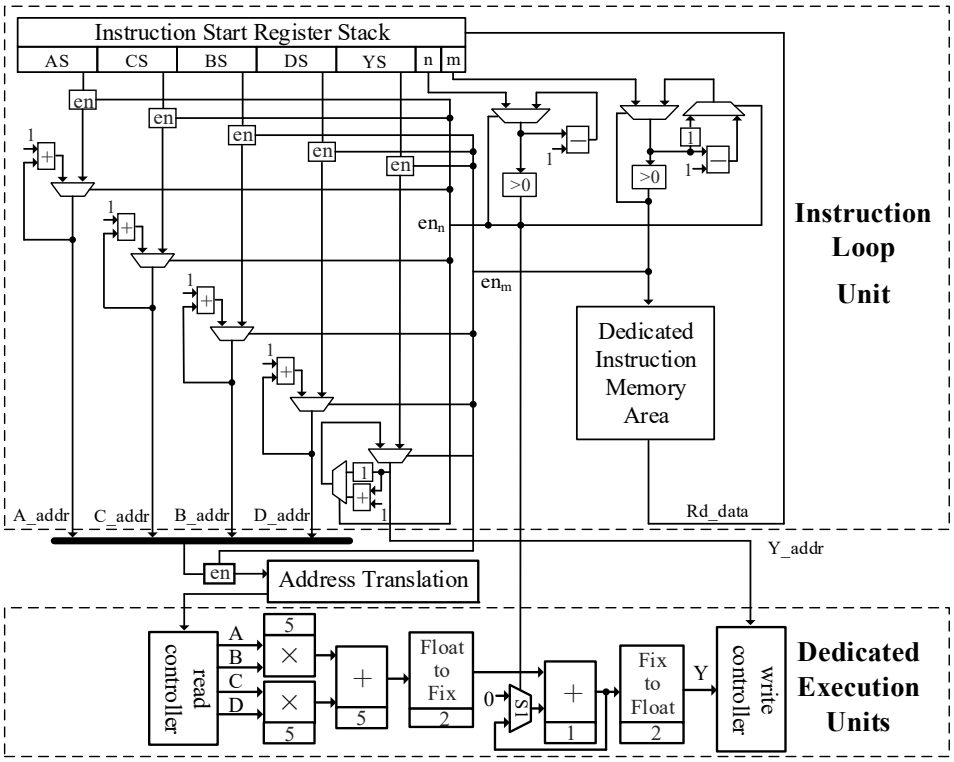


Figure 11. Structure of dedicated execution units and instruction loop unit.

6. Example Verification

6.1. Simulation Platform

FRTDS uses the Virtex-7 FPGA VC709 as the development board for the solver, and the FPGA chip on the board is XC7VX690T-2FFG1761. The chip contains 693,120 logic units, 108,300 configurable logic modules, 3600 DSP slices, and 1470 36KB dual-port BRAMs. The original FRTDS contains four universal arithmetic components in each microprocessor core. The newly designed FRTDS replaces two universal arithmetic components in each computing core with dedicated execution units. Table 1 shows the resources consumed by placing different numbers of microprocessor cores in the FRTDS in VC709. Limited by the configurable logic block (CLB), the original scheme only holds five microprocessor cores at most. The new design scheme has six microprocessor cores. Because the dedicated execution unit consumes less instruction storage space, in the new design scheme, the storage resources used for instruction storage in the original design can be saved for storing the parameters by replacing online computation with parameter prestorage. Finally, we designed the FRTDS according to the new design scheme with six microprocessor cores, working at a clock frequency of 150 MHz, and each microprocessor core is able to store 8000 prestored parameters in the form of double-precision floating points.

Table 1. Resource consumption of the improved FRTDS design.

Number of Microprocessing Cores	Resource Consumption				
	CLB		DSP		RAM
	New FRTDS	Original FRTDS	New FRTDS	Original FRTDS	
3	42.63%	51.25%	18.7%	25.86%	50.03%
4	53.83%	70.94%	24.56%	34.11%	64.18%
5	67.14%	92.62%	30.42%	42.36%	78.35%
6	84.6%	/	36.28%	/	92.53%

The graphical simulation software and task scheduling software are developed by the QT C++ platform. The graphical modeling of the simulation object automatically provides the specific simulation calculation process according to the characteristics of the electrical equipment and their connection relationships. The instruction-level assignment of computing tasks automatically generates a directed acyclic graph (DAG) describing the dependencies of computing tasks according to the positions of variables in the computing expression. The task scheduling software schedules the DAG graph to get an instruction stream, which can be stored or downloaded to FRTDS by the upper computer. FRTDS performs the corresponding operations according to each instruction to complete the simulation calculation. We connected the designed FRTDS to the digital relay protection device through a common ethernet switch to form a hardware-in-loop real-time simulation platform, as shown in Figure 12. We adopted the UDP communication protocol between the FRTDS and the upper computer responsible for command download and system operation monitoring, and we used the IEC61850 protocol to connect the real digital protection device. In particular, the IP address of FRTDS and the upper computer should be set on the same LAN. The message analyzer communicated with the FRTDS using SV messages and is able to draw and display real-time simulation waveforms. For another development board, VC707, we simulated the control system of the flexible DC transmission system to control the Aurora protocol of Xilinx Company between the controller and the FRTDS instead of the real controller.

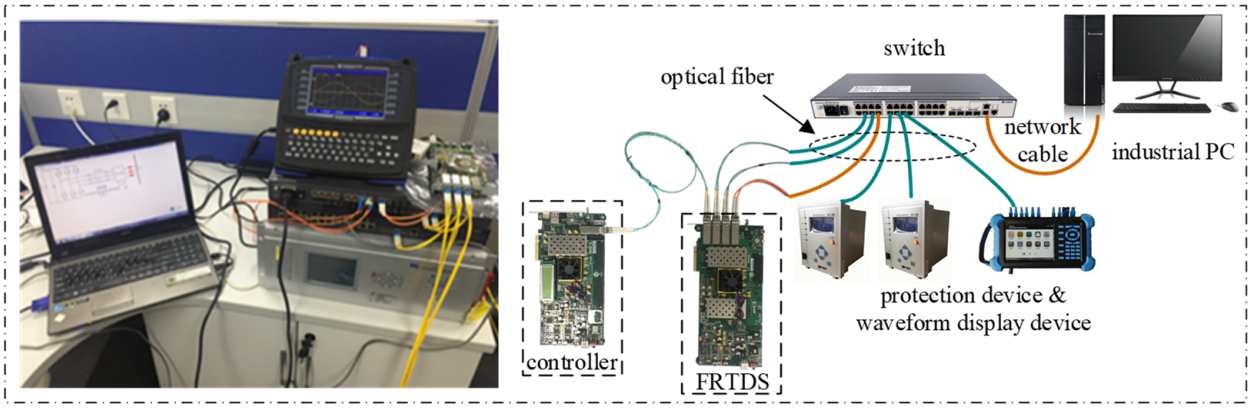


Figure 12. Hardware-in-loop system based on FRTDS.

6.2. Simulation Objects and Results

As the simulation object, we adopted the double-terminal MMC–HVDC system shown in Figure 13, where the AC grid voltage was 230 kV, and the DC bus voltage was 400 kV. The topology of the converter submodule was P-FBSM, in which each bridge arm contained 80 submodules, the submodule capacitance was 2 mF, the bridge arm reactor was 50 mH, and the converter capacity was 350 MVA. For the inverter-side MMC1, we adopted the control mode of fixed active power and fixed reactive power; for the rectifier-side MMC2, we adopted the control mode of fixed DC voltage and fixed reactive power and adopted the previously described low-switching-frequency modulation strategy [22]. The simulation step was set to 10 μs. If using the previously proposed calculation method of eliminating the internal nodes of all dual-port submodules in the MMC bridge arm and the interconnection nodes between modules in cyclic iteration [11], the calculation of one step would cost 64.8 μs, which is insufficient for real-time simulation. In the newly designed FRTDS, the calculation amount $E1$ of elimination and back-substitution between subnetworks is 38 unit quantity, and the calculation amount $E2$ of eliminating nodes 5 and 6 in the six-node subnetwork model is 16 unit quantity. Therefore, we divided the bridge arm into subnetworks, with each subnetwork containing 10 submodules. According to Equation (13), the number of parameters that the subnetwork needed to prestore was 5186, and the FRTDS was capable of storing these prestored parameters. After the subnetwork was divided, we used the unified model of the 6-node PFBSM subnetwork and performed the real-time simulation in FRTDS with a step of 9.62 μs using the method of replacing inline computation with the parameters' prestorage. We built the same simulation object using MATLAB/Simulink, and we compared the offline running results with the real-time calculation results produced by FRTDS:

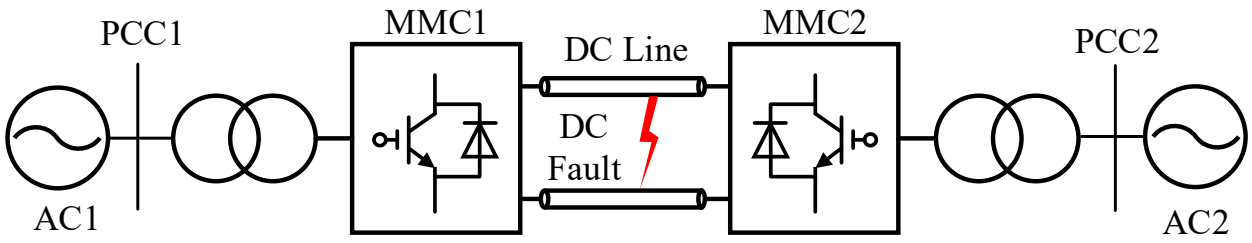


Figure 13. Double-ended MMC–HVDC system.

(1) Trouble-free operation

Figure 14a–d shows the A-phase AC current I_{sa} on the inverter side, the DC-side voltage U_{dc}, the capacitor current I_c of the No. 1 submodule on the upper bridge arm of the rectifier side in A phase, and the capacitor voltage of each submodule of the upper bridge arm in A phase when the MMC–HVDC system operates in a steady-state, respectively.

Figure 14a–c shows that when the MMC simulation system ran without fault, the simulation waveform in FRTDS using the proposed modeling method had a high degree of coincidence with the simulation waveform in the detailed model in MATLAB. The error was mainly reflected in the high-frequency irregular burrs, and the maximum relative error was about 0.89% in the DC-side voltage. The result verified that the simulation accuracy of the proposed modeling method is high. From the waveform of the bridge arm capacitor voltage of the FRTDS shown in Figure 14d, we found that the capacitor voltage was balanced over time, and some capacitors had equal voltage at some moments, which demonstrated the parallel voltage equalization characteristics of P-FBSM.

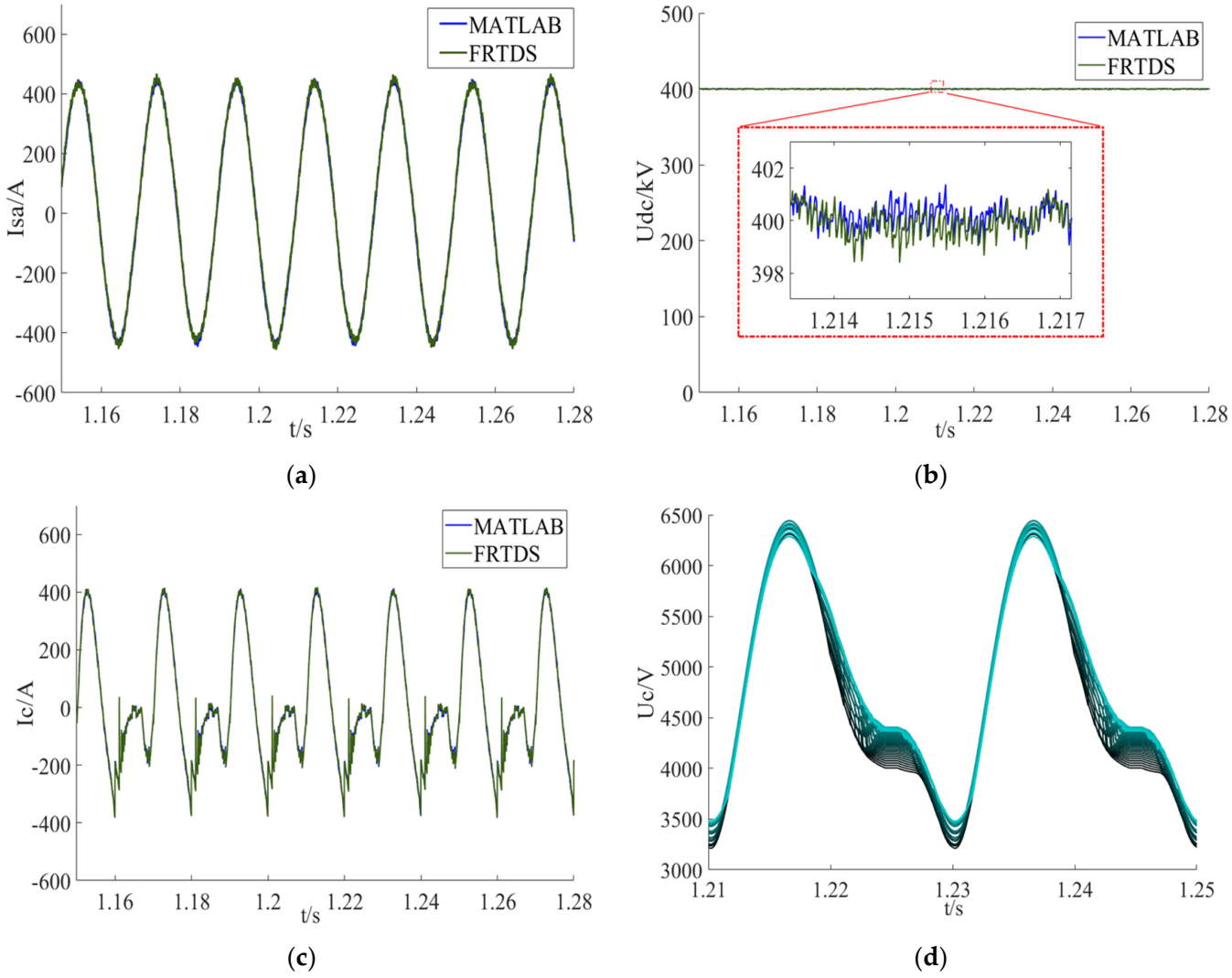


Figure 14. Steady-state operation waveform of the MMC simulation system: (**a**) A-phase AC current waveform of inverter side; (**b**) DC-side voltage waveform; (**c**) waveform of the capacitor current of the upper bridge arm in phase A on the rectifier side; (**d**) capacitor voltage of the upper bridge arm in phase A on the rectifier side.

(2) DC bipolar fault

We set a DC bipolar short-circuit fault at t = 1.295 s, the whole station was blocked 5 ms after the fault occurred, which unblocked at t = 1.365 s. Figure 15a–d shows the AC current I_{sa} of the inverter side in phase A under the fault, the capacitor voltage U_c of the first submodule of the upper bridge arm in phase A on the rectifier side, the DC-side voltage U_{dc}, and the DC side current I_{dc}. Figure 15 shows that the simulation accuracy of FRTDS was still high during the bipolar fault occurrence in the MMC simulation system

before and after the blocking, and the maximum relative error was about 1.09% on the DC-side voltage.

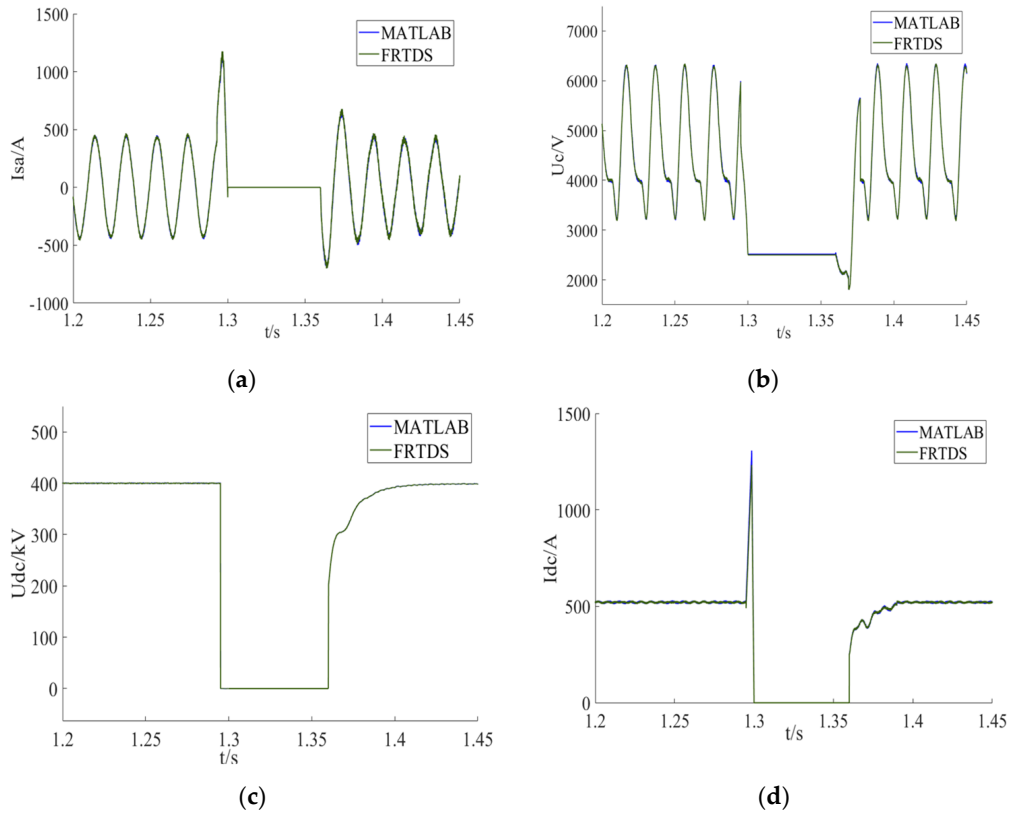


Figure 15. Bipolar fault transient waveform diagram of the MMC simulation system: (**a**) A-phase AC current waveform of the inverter side; (**b**) waveform diagram of the capacitor voltage of the upper bridge arm of phase A on the rectifier side; (**c**) DC-side voltage waveform; (**d**) DC-side current waveform.

7. Conclusions

(1) We found that the proposed unified equivalent model of the P-FBSM subnetwork, after analyzing the working mechanism of the submodules, requires less computation by replacing inline computation with parameter prestorage and reduces the storage capacity required for prestored parameters, making it suitable for application in FRTDS.

(2) After dividing the subnetwork by a fixed number of submodules, we found a relationship between the amount of simulation calculation and the number of submodules in the subnetwork. Accordingly, the subnetwork division method with the least amount of calculation could be determined.

(3) After improving the structure of the computing components, we found that the FRTDS real-time simulation platform has improved resource use, requires less storage space for prestored parameters, and is more suitable for real-time simulations of dual-port submodule MMC–HVDC systems by replacing inline computation with the prestorage of parameters.

When subnetwork multivalued parameters are prestored, for the convenience of hardware query design, the current method does not consider the problem where different prestored parameters may have the same value. In future research, the amounts of computation and storage should be further optimized, a matching multivalued parameters query circuit should be designed in FRTDS, and the improved method and platform should be applied to the real-time simulation of MMC–HVDC using other types of dual-port submodule topologies.

Author Contributions: Conceptualization, Z.J.; methodology, B.Z. and Z.J.; software, Y.W.; validation, S.W. and Z.J.; formal analysis, B.Z.; investigation, B.Z. and Z.J.; resources, B.Z.; data curation, Y.W. and S.W.; writing—original draft preparation, S.W.; writing—review and editing, B.Z. and S.W.; visualization, Y.W.; supervision, B.Z.; project administration, B.Z.; funding acquisition, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 51477114).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful for the experimental site provided by the Key Laboratory of the Ministry of Education for Smart Grid of Tianjin University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bresesti, P.; Kling, W.L.; Hendriks, R.L.; Vailati, R. HVDC Connection of Offshore Wind Farms to the Transmission System. *IEEE Trans. Energy Convers.* **2007**, *22*, 37–43. [CrossRef]
2. Gomis-Bellmunt, O.; Sau-Bassols, J.; Prieto-Araujo, E.; Cheah-Mane, M. Flexible Converters for Meshed HVDC Grids: From Flexible AC Transmission Systems (FACTS) to Flexible DC Grids. *IEEE Trans. Power Deliv.* **2020**, *35*, 2–15. [CrossRef]
3. Tang, L.; Dong, X.; Shi, S.; Qiu, Y. A high-speed protection scheme for the DC transmission line of a MMC–HVDC grid. *Electr. Power Syst. Res.* **2019**, *168*, 81–91. [CrossRef]
4. Resch, S.; Friedrich, J.; Wagner, T.; Mehlmann, G.; Luther, M. Stability Analysis of Power Hardware-in-the-Loop Simulations for Grid Applications. *Electronics* **2022**, *11*, 7. [CrossRef]
5. Jeon, J.H.; Kim, J.Y.; Kim, H.M.; Kim, S.K.; Cho, C.; Kim, J.M.; Ahn, J.B.; Nam, K.Y. Development of Hardware In-the-Loop Simulation System for Testing Operation and Control Functions of Microgrid. *IEEE Trans. Power Electron.* **2010**, *25*, 2919–2929. [CrossRef]
6. Bruno, S.; Giannoccaro, G.; Iurlaro, C.; La Scala, M.; Rodio, C. Power Hardware-in-the-Loop Test of a Low-Cost Synthetic Inertia Controller for Battery Energy Storage System. *Energies* **2022**, *15*, 3016. [CrossRef]
7. Ilves, K.; Taffner, F.; Norrga, S.; Antonopoulos, A.; Harnefors, L.; Nee, H.P. A submodule implementation for parallel connection of capacitors in modular multilevel converters. *IEEE Trans. Power Electron.* **2015**, *30*, 3518–3527. [CrossRef]
8. Li, Z.; Lizana, R.; Sha, S.; Yu, Z.; Peterchev, A.V.; Goetz, S.M. Module implementation and modulation strategy for sensor-less balancing in modular multilevel converters. *IEEE Trans. Power Electron.* **2019**, *34*, 8405–8416. [CrossRef]
9. Xu, J.; Zhang, J.; Li, J.; Shi, L.; Jia, X.; Zhao, C. Series-parallel HBSM and two-port FBSM based hybrid MMC with local capacitor voltage self-balancing capability. *Int. J. Electr. Power Energy Syst.* **2018**, *103*, 203–211. [CrossRef]
10. Goetz, S.M.; Peterchev, A.V.; Weyh, T. Modular multilevel converter with series and parallel module connectivity: Topology and control. *IEEE Trans. Power Electron.* **2015**, *30*, 203–215. [CrossRef]
11. Xu, J.; Fan, S.; Zhao, C.; Gole, A.M. High-Speed EMT Modeling of MMCs With Arbitrary Multiport Submodule Structures Using Generalized Norton Equivalents. *IEEE Trans. Power Deliv.* **2018**, *33*, 1299–1307. [CrossRef]
12. Sharma, A.; Srivastava, S.C.; Chakrabarti, S. Testing and validation of power system dynamic state estimators using real time digital simulator (RTDS). *IEEE Trans. Power Syst.* **2016**, *31*, 2338–2347. [CrossRef]
13. Yu, D.; Zhou, N.; Luo, Y.; Dong, L.; Jia, Z. Research on the Cross-Platform Co-Simulation Strategy of Power Systems Based on the Model-Segmentation Algorithm. *Electronics* **2021**, *10*, 3185. [CrossRef]
14. Yuan, J.; Samson, S.Y.; Zhang, G.; Lim, C.P.; Trinh, H.; Zhang, Y. Design and HIL Realization of an Online Adaptive Dynamic Programming Approach for Real-Time Economic Operations of Household Energy Systems. *IEEE Trans. Smart Grid* **2022**, *13*, 330–341. [CrossRef]
15. Zi, P.; Li, Y.; Tan, B.; Chen, X.; Jia, L.; Wang, M. Current situation of largescale power grid simulation tools and their popularization and application in North China Power Grid. *Electr. Power Autom. Equip.* **2019**, *39*, 199–205.
16. Chen, Y.; Dinavahi, V. Digital hardware emulation of universal machine and universal line models for real-time electro-magnetic transient simulation. *IEEE Trans. Ind. Electron.* **2012**, *59*, 1300–1309. [CrossRef]
17. Liu, J.; Dinavahi, V. A Real-Time Nonlinear Hysteretic Power Transformer Transient Model on FPGA. *IEEE Trans. Ind. Electron.* **2013**, *61*, 3587–3597. [CrossRef]
18. Bai, H.; Luo, H.; Liu, C.; Paire, D.; Gao, F. A Device-Level Transient Modeling Approach for the FPGA-Based Real-Time Simulation of Power Converters. *IEEE Trans. Power Electron.* **2020**, *35*, 1282–1292. [CrossRef]
19. Matar, M.; Iravani, R. Massively Parallel Implementation of AC Machine Models for FPGA-Based Real-Time Simulation of Electromagnetic Transients. *IEEE Trans. Power Deliv.* **2010**, *26*, 830–840. [CrossRef]

20. Zhang, B.; Jin, X.; Tu, S.; Jin, Z.; Zhang, J. A New FPGA-Based Real-Time Digital Solver for Power System Simulation. *Energies* **2019**, *12*, 4666. [CrossRef]
21. Shi, L.; Zhao, C.; Xu, J.; Xu, Y. Research on voltage self-balancing operating characteristics of paralleled full-bridge submodules based MMC. *Proc. CSEE* **2018**, *38*, 2419–2428+2551.
22. Xu, J.; Li, J.; Zhang, J.; Shi, L.; Jia, X.; Zhao, C. Open-loop voltage balancing algorithm for two-port full-bridge MMC–HVDC system. *Int. J. Electr. Power Energy Syst.* **2019**, *109*, 259–268. [CrossRef]