

Article

FPGA-Based Real-Time Digital Solver for Electro-Mechanical Transient Simulation

Bingda Zhang *, Yang Wang, Sijia Tu and Zhao Jin

The Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China; leowang94@tju.edu.cn (Y.W.); tusijia@tju.edu.cn (S.T.); 1017234013@tju.edu.cn (Z.J.)

* Correspondence: bdzhang@tju.edu.cn; Tel.: +86-022-2740-4101

Received: 10 September 2018; Accepted: 2 October 2018; Published: 4 October 2018



Abstract: A field-programmable gate array (FPGA)-based digital solver for real-time electromechanical transient simulation is designed in this paper. The solving process for a device or sub-network in an electromechanical transient simulation is packaged into the orders in soft function solvers. The orders are reused by soft function solvers that are invoked by microprocessor cores. The data exchange between the microprocessor cores and soft function solvers is enhanced through explicit and implicit channels. The orders of the microprocessor cores are stored in the synchronous dynamic random access memory on the FPGA board, which solves the problem of insufficient storage space for the orders in electromechanical transient simulation. A real-time digital solver for electromechanical transient simulation, whose feasibility is verified by a simulation of part of the power system in East China, is successfully constructed by applying a small number of microprocessor cores and multiple soft function solvers.

Keywords: real-time electromechanical transient simulation; field-programmable gate array (FPGA); FPGA-based real-time digital solver (FRTDS); soft function solvers; data cluster transmitter

1. Introduction

Real-time simulation of power systems is a form of simulation where the simulation process is consistent in time with the actual power system. These simulations can be used for hardware-in-the-loop tests, which play an important role in the design, testing, and detection of power system automation and protection systems, as well as in the teaching and training process of power systems [1–3]. According to the classification of dynamic processes, real-time simulation includes both real-time electromagnetic and real-time electromechanical transient simulations. Real-time electromagnetic transient simulation, the time step of which is in the microsecond range, is used to test equipment with high working frequency, or equipment that is designed based on instantaneous voltage and current value. Real-time electromechanical transient simulation, the typical time step of which is 10 milliseconds, is used to test equipment such as united power flow controllers (UPFC) and energy management systems (EMS). It can also be applied to dispatcher training systems (DTS).

The commonly used real-time electromechanical transient simulation platforms include personal computer (PC) clusters and graphics processing unit (GPU) and CPU collaborative real-time simulation platforms. The PC cluster, with CPU as the computing core, is composed of multiple PCs and a commercial high-speed network. PC clusters are cost-effective, easy to upgrade, and easy to expand, so they have been widely used for real-time electromechanical transient simulation [4–6]. A real-time simulation system based on electromechanical transient simulation has been proposed [7], which introduced a hardware-in-the-loop real-time simulation of a system with virtual excitation regulators of simulated generators and connected virtual relay protection devices. A complex fault parallel computing method that could be used for digital simulation of large-scale power systems was

proposed [8], which effectively reduced parallel computation and communication, and was conducive to expanding the scale of real-time simulation. However, the relatively narrow communication bandwidth and long communication delay of the PC cluster constrained its simulation scale. As an emerging parallel computing device, GPU has attracted the attention of power system simulation researchers since its creation [9–11]. The GPU has a large number of computing cores, providing lightweight thread based on hardware switching and a multi-level storage structure that can perform massive parallel computation. The GPU simulation program model [12,13] allowed electromechanical transient simulation with multiple granularities on multi-GPU clusters. The cu-basic linear algebra subprograms (cuBLAS) library is only suitable for symmetric matrix calculation. The calculation of each matrix element in the LU decomposition is allocated to a single thread, which improves the versatility of the parallel computation [14]. The GPU performs parallel computation with fine granularity, but it cannot independently control the process or schedule the data in the simulation, and must collaborate with a CPU. However, the time-consuming data transmission between the GPU and CPU constrains its ability to perform simulation in real-time.

The field-programmable gate array (FPGA) has a fully configurable parallel hardware structure, distributed memory structure, and deep pipeline structure, which performs highly parallel numerical calculations, and is inexpensive and small. FPGAs have already been used for signal recognition [15], transmission line fault detection [16], and harmonic analysis in power systems [17]. It is widely used for real-time simulation of power systems [18–21]. Based on the idea of modular design, a multi-FPGA hardware designed for real-time electromagnetic transient simulation of large-scale power systems was proposed [22]. Motor models, distributed power supplies, modular multi-level converters (MMC), and LLC resonant converters, which are suitable for FPGA, have been reported [23–26]. Real-time simulation of active distribution networks was achieved using FPGA hardware and input/output (I/O) interface design [27,28]. Studies of transient co-simulation of FPGA and a real-time digital solver (RTDS) were reported [29,30]. A FPGA-based real-time digital solver (FRTDS) was designed [31,32]. The hardware design and the orders generator of FRTDS have been improved [33], so that FRTDS could perform multi-value parameter pre-storage and multi-rate simulation. The overall structure of a hardware-in-the-loop real-time simulation platform of a smart substation was proposed [34]; the formation and resolution of the process layer network communication message were realized in FRTDS. However, most current research into FPGA real-time simulation including FRTDS is focused on electromagnetic transient simulation, and little research has addressed real-time electromechanical transient simulation.

FRTDS is able to perform real-time electromagnetic transient simulation with multiple granularities when constructing multiple microprocessor cores with the same structure in it. However, electromechanical transient simulation has a long time step, so the FPGA storage resources are insufficient for such a large amount of orders. To address this problem, a real-time digital solver with a small number of microprocessor cores and multiple soft function solvers is constructed in the FPGA board in this paper. The orders for the microprocessor cores are stored in the synchronous dynamic random access memory (SDRAM). Soft function solvers, which store the orders for the solving process, can calculate the devices and sub-networks in electromechanical transient simulations. The microprocessor cores invoking the soft function solvers enable the reuse of the orders, which reduces the orders storage space requirement for real-time electromechanical transient simulation.

The original FRTDS hardware design for real-time electromagnetic transient simulation is introduced in Section 2. Section 3 describes the new FRTDS hardware design for electromechanical transient simulation. Section 4 presents the design of the orders in the soft function solvers. The feasibility of the novel FRTDS for electromechanical transient simulation is verified in Section 5 using a case study. The novel FRTDS is discussed in Section 6 and our conclusions are outlined in Section 7.

2. FRTDS Hardware Design

With pipeline technology, arithmetic expressions and functions are encapsulated in the processing elements (PEs) at a frequency of 200 MHz in FRTDS. Guide words and the influencing words are used to indirectly modify the simulation parameters. The pipelining operation of the PEs is described as orders, like assembly language. The compilation software for translating the simulation script to orders can prevent users from FPGA programming [31–34]. The overall FRTDS structure is shown in Figure 1. Several microprocessor cores exist in FRTDS. Data exchange between the microprocessor cores is performed by means of “hand in hand and data pipeline”. The “Ping-Pong” operation is used for data exchange between microprocessor cores and external devices, whose frequency is controlled by the time step controller. FRTDS is equipped with a small form-factor pluggable/enhanced small form-factor pluggable (SFP/SFP+) interface and peripheral component interconnect express (PCI-E) interface. The SFP/SFP+ interface is connected with signal conversion devices for interacting with actual devices in the power system. The PCI-E interface is used for data exchange between the solver and the industrial computer.

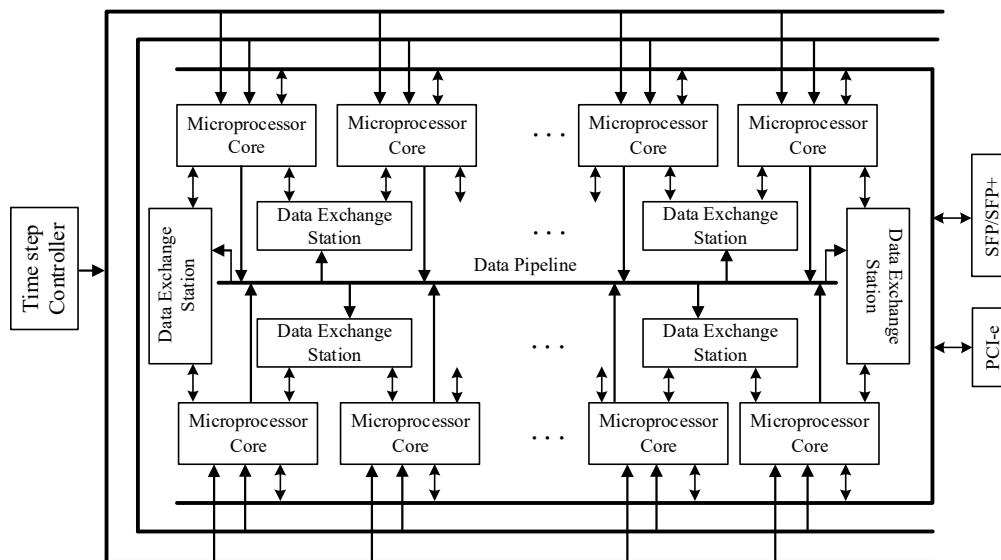


Figure 1. The overall structure of a field-programmable gate array (FPGA)-based real-time digital solver (FRTDS).

A microprocessor core consists of processing elements (PEs), a data storage unit, a control unit, an order allocation unit, and multiplex switches. The structure of a microprocessor core is shown in Figure 2. PEs are applied for arithmetic, logic, and comparison calculations. The data storage unit is used for storing data; the parts for interacting with the communication circuits are set as two sets. The Ping-Pong operation mechanism ensures that the modification of system parameters by external devices is uniformly updated at the beginning of each time step. The control unit determines what calculations are performed in PEs according to the orders, and it controls the status of the multiplex switches to ensure the ports of the PEs are correctly connected with the data storage unit. The order allocation unit takes orders from memory and transforms them into the data structure required by the control unit to guarantee the PEs work as ordered.

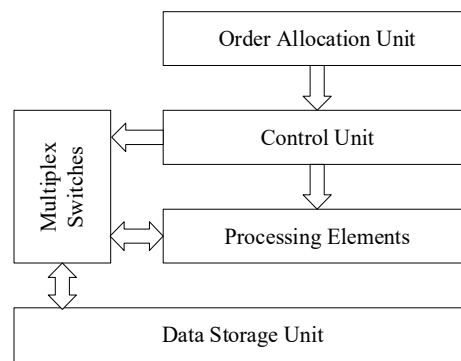


Figure 2. The structure of a microprocessor core.

The PEs include a three-level series arithmetic expression circuit that can execute specified mixed calculations of addition, subtraction, multiplication, and division according to the selection words. Logic expression circuits and comparison expression circuits are constructed in the PEs for logic and comparison calculations. There are several data channels in the PEs that are used for data transmission between different blocks in the data storage unit. The function circuits are built in the PEs to solve some calculation processes that cannot be solved by the arithmetic expression circuits, including exponential function circuits, logarithmic function circuits, and trigonometric function circuits. These expression circuits are constructed for specified requirements with pipeline technology used. The structure of the PEs is shown in Figure 3.

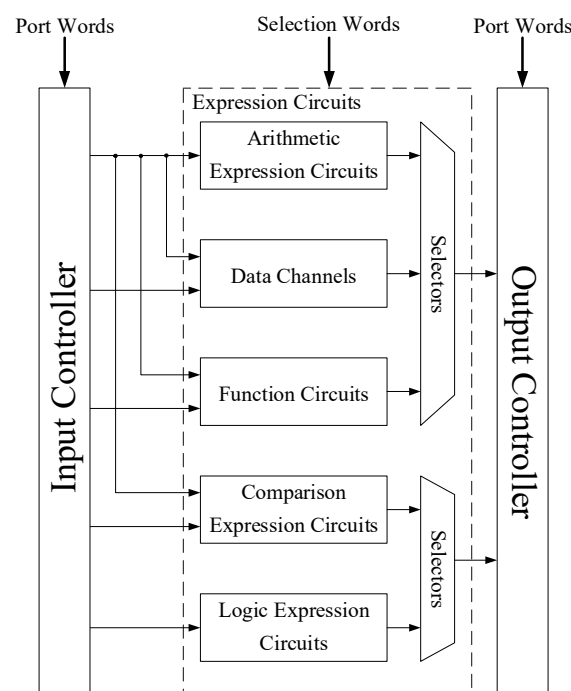


Figure 3. The structure of the processing elements (PEs).

As the logic expression circuits, comparison expression circuits and function circuits are not frequently used. Some input ports of the PEs are shared by arithmetic expression circuits, comparison expression circuits, function circuits, and data channels. Some output ports of the PEs are shared by logic expression circuits and comparison expression circuits through selectors, whereas some of the other output ports are also shared by arithmetic expression circuits, data channels, and function circuits through selectors. Therefore, although the computing capability of the PEs is weakened slightly, fewer PE ports are required. However, the resource consumption of multiplex switches is much lower.

The calculation tasks of FRTDS are described as orders. The orders include information about the status of the selectors and ports in the PEs at each clock in the pipelining work. The addresses of the data used by the ports are also contained in the orders. Each order, whose width is 512 bits, includes a control part and a data address part. The width of the control part is 128 bits, and it includes both selection and port words. The selection words determine the selectors in the PEs which input should be selected as the output, whereas the port words determine whether or not each port receives the data address. The data address part, with a width of 384 bits, can be filled with 24 16-bit addresses. These addresses are arranged in the sequence of the ports that receive data. When there are fewer than 24 addresses, the invalid address “0xFFFF” is used to fill the gap.

Some parameters have various possible values in the simulation. These parameters are called multi-valued parameters for FRTDS. For instance, there are two possible values for the mutual admittance between two nodes that are connected by a switch, described as a binary resistance model. This mutual admittance is a multi-valued parameter. All possible values of a multi-valued parameter are consecutively pre-stored in the data storage unit. A dedicated addressing circuit is built in the control unit for searching the actual address of the parameter. As some parameters have the same possible values, memory space of the data storage unit can be saved using the addressing circuit.

Different kinds of data, including multi-valued parameters, historical data, iterative loop data, and output data, are stored in different ranges in the data storage unit so that a multi-valued parameter can be recognized by its data address. The given address is the actual address if the data do not refer to a multi-valued parameter; otherwise, the actual address should be obtained by the guide word and the addressing circuit.

Each microprocessor core should be given an order at each time point to ensure the pipelining work is continuous in the simulation. The frequency of the PEs is 200 MHz, so that 10,000 orders are required for the 50 μ s time step, which is a typical time step for electromagnetic transient simulations. However, 2,000,000 orders are required for a 10 ms time step, which is a typical time step for electromechanical transient simulation. It is difficult to realize real-time electromechanical transient simulation on FRTDS because the FPGA chip memory space is insufficient for constructing the storage to store the orders.

3. FRTDS for Electromechanical Transient Simulation

There are two methods that can solve the problem of insufficient storage space for orders:

- (1) Expanding the storage space for orders. There is SDRAM on the FPGA board so that orders can be stored in SDRAM. However, only a small number of microprocessor cores can be constructed due to the limited SDRAM resources.
- (2) Reducing the orders. The loop function and jump function can be used to describe the repeated calculation processes. However, the application of these functions violates the design concept of the order system based on the pipeline mechanism, and the order allocation unit would be complicated.

Considering the existing problems in these two methods, a novel FRTDS consisting of a small number of microprocessor cores and multiple soft function solvers is designed in this study, as shown in Figure 4.

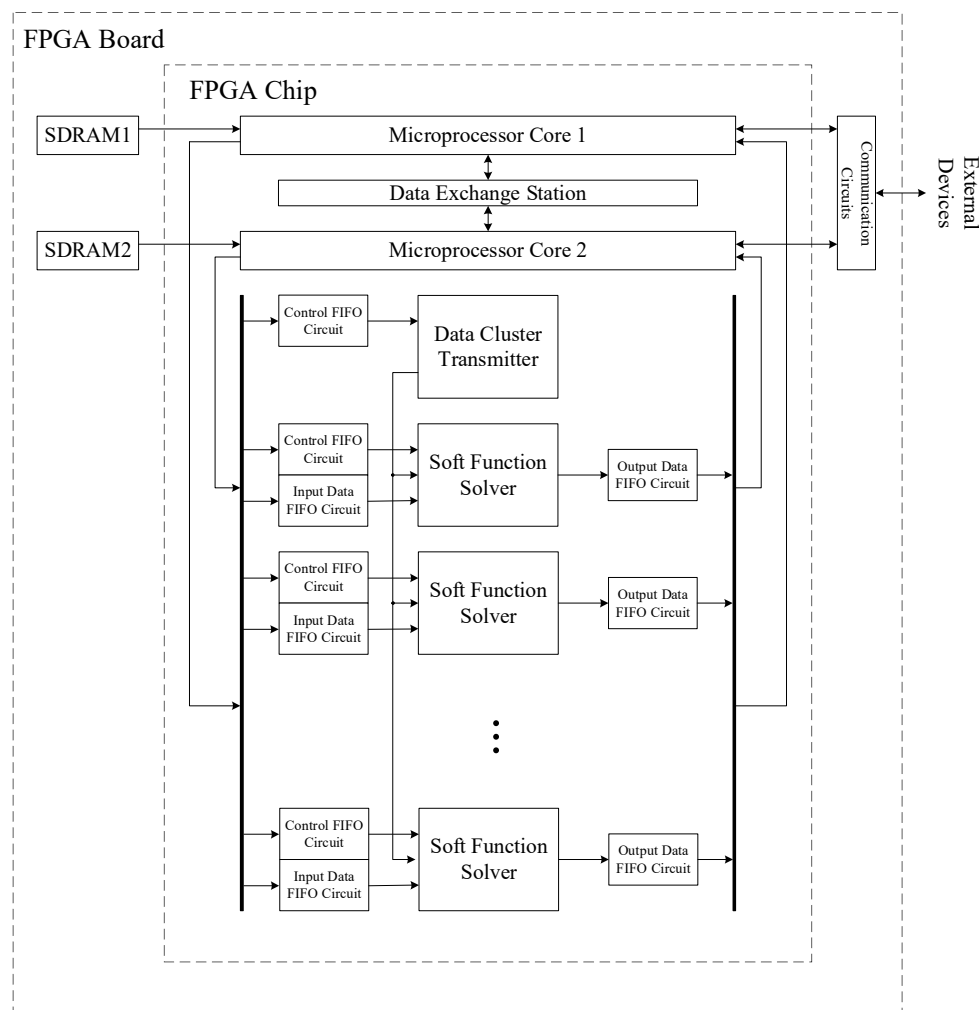


Figure 4. The structure of FRTDS for electromechanical transient simulation.

The microprocessor cores whose orders are stored in SDRAM are used for process control and data dispatch. The soft function solvers whose orders are stored in the FPGA chip are used for process calculation. The reuse of the orders is accomplished by the microprocessor cores using soft function solvers, which reduces the requirement for order storage space for real-time electromechanical transient simulation. The data cluster transmitter is used to enhance the ability of data exchange between microprocessor cores and soft function solvers.

3.1. Use of SDRAM

A Virtex-7 FPGA VC709 FPGA (Xilinx, San Jose, CA, USA) board was used to construct the FRTDS. There are two independent pieces of double-data-rate 3 (DDR3) SDRAM with 4 GB storage space, whose data width is 64 bits, and the maximum operating frequency is 933 MHz.

The memory interface generator (MIG) core was used as the data exchange interface between the PEs and the SDRAM, whose first in first out (FIFO) circuit enable the PEs and the SDRAM to work at different frequencies. SDRAM can be read and written on both the rising and falling edges of the clock, whereas the PEs can only perform all operations on the rising edge of the clock. The actual frequency of the SDRAM is 8 times that of the PEs, when the main clock frequency of SDRAM is set to 4 times that of PEs. Eight 64-bit data can be read from a single piece of SDRAM in a clock of the PEs, which can exactly form a complete 512-bit order. As such, two microprocessor cores were constructed in the Virtex-7 FPGA VC709 board.

3.2. Soft Function Solvers

The function circuits in the PEs were designed according to the pipeline mechanism. The hardware resources consumption would be extremely large when the function circuits are used to complete the solving processes of the devices and sub-networks in electromechanical transient simulation.

In the soft function solvers, the solving process of a device or a sub-network is transformed into a set of orders, so the complex solving process is accomplished through the orders. Unlike the function circuits, a soft function solver calculates the functions in turn, which means the next set of orders are executed only after the current set of orders is completely executed. However, the microprocessor cores can perform other tasks while the invoked soft function solvers are working. The orders for the soft function solvers and microprocessor cores are executed concurrently without breaking the pipeline mechanism of the orders system in microprocessor cores. In a soft function solver, multiple sets of orders can be stored, and different functions are solved at different times, so the soft function solvers have strong versatility.

The hardware structure of the soft function solvers, which is similar to that of microprocessor cores, is composed of PEs, a data storage unit, a control unit, an order storage unit, and multiplex switches. However, the PEs of the soft function solvers are relatively simple and consume less hardware resources, so multiple soft function solvers can be constructed.

The amount of orders in microprocessor cores depends on the time step, whereas the number of orders in soft function solvers depends on the calculation tasks. So, the orders in the soft function solvers do not consume much storage space and can be stored in the FPGA chip. The composition of the orders in the soft function solvers, similar to that in microprocessor cores, includes a control part and a data address part. Since the number of ports and selectors in the soft function solvers is less than in the microprocessor cores, orders with a width of 256 bits are used. The width of the control part is 64 bits, and the width of data address part is 192 bits, equaling a length of 16 12-bit data addresses.

The data storage unit of the soft function solver is divided into a public storage area and a private storage area. Common data used by many objects and intermediate variables appearing in the simulation of the soft function solver are stored in the public storage area. Private data, which is arranged according to the sequence of the objects, are stored in the private storage area. The offset address can be discovered in the data address part of the orders in the soft function solvers, and the base address can be captured when the microprocessor cores invoke the soft function solver. The actual addresses of the data can be obtained by the base address and the offset address. There are two sets of private storage areas that are used for the soft function solvers and the data cluster transmitter according to the Ping-Pong operation mechanism.

3.3. Data Exchange between Microprocessor Cores and Soft Function Solvers

The data exchange between the soft function solvers and microprocessor cores can be completed through both explicit channels and the implicit channel. The explicit channels for data exchange are the FIFO circuits between soft function solvers and microprocessor cores. Each soft function solver has a control FIFO circuit, an input data FIFO circuit, and an output data FIFO circuit. The control FIFO circuit and input data FIFO circuit are written by microprocessor cores and read by the soft function solver. The output data FIFO circuit is written by the soft function solver and read by microprocessor cores. The width of each FIFO circuit is 64 bits.

The content of the control FIFO circuit indicates the entry address of the function, the object index, and the set number of the private storage area. After the soft function solver receives the message from the control FIFO circuit, it begins to execute the orders specified by the entry address of the function and stops when encountering the termination order. The object index is used as the base address for addressing.

Microprocessor cores can write the preparation data in the input data FIFO circuit for the next object or even the next two objects, which means task arrangement is flexible in the microprocessor cores. However, microprocessor cores are not suitable for situations where the soft function solver

needs a large amount of data provided by microprocessor cores, because orders are needed when microprocessor cores write data in the input data FIFO circuit.

The implicit channel for data exchange is the data cluster transmitter. There is only a control FIFO circuit but no input data FIFO circuit, output FIFO circuit, or orders in the data cluster transmitter. The content of the control FIFO circuit indicates the source address of the transmission data in the microprocessor core, the destination address of transmission data in the soft function solver, the amount of transmission data, and the set number of the private storage area. After the data cluster transmitter receives a message from the control FIFO circuit, it transmits the data from the source address in microprocessor cores to the destination address in soft function solvers until all the data with specified amount are completely transmitted.

In order to ensure that the data cluster transmitter work does not affect the microprocessor core, the data cluster transmitter only operates on the image of the microprocessor cores, in which multi-valued parameters, guide words, internal influencing words, external influencing words, and the addressing circuit in microprocessor cores are copied. In order to ensure that the work of the data cluster allocator does not affect the soft function solvers, the data cluster transmitter only operates on the specified set number of the private storage area according to the message from the control FIFO circuit.

4. Design of Orders in Soft Function Solvers

4.1. Process of Electromechanical Transient Simulation

There are electrical devices such as generators and loads in electromechanical transient simulation, as well as control devices including power system stabilizers (PSS), excitation systems, and prime movers with their speed controllers. The input-output relationship among various devices and the power system is shown in Figure 5.

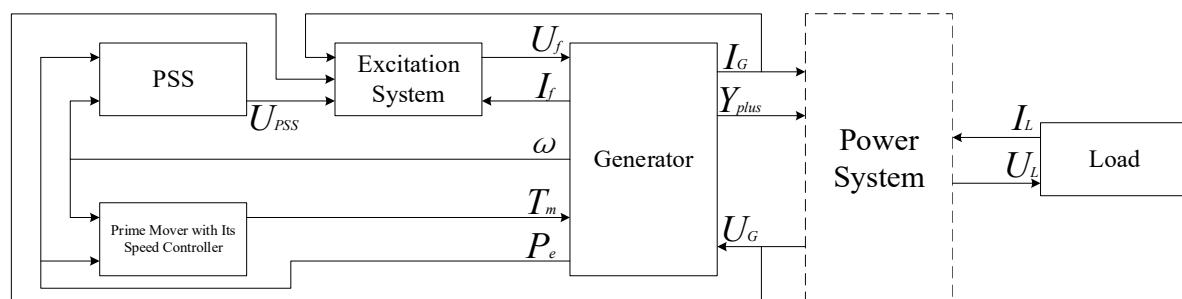


Figure 5. The input-output relationship among various devices and the power system. ω is the rotor angular velocity of the generator, P_e is the electromagnetic power of the generator, T_m is the mechanical torque provided by the prime mover, U_{PSS} is the output of the power system stabilizers (PSS), U_f is the exciting voltage, I_f is the exciting current, I_G and U_G refer to the equivalent current source and the terminal voltage of the generator, respectively, Y_{plus} is the additional admittance of the generator to the network, and I_L and U_L refer to the equivalent current source and the terminal voltage of the load, respectively.

The steps for real-time electromechanical transient simulation using the sequential method are as follows:

- (1) Update the input from the actual external devices, and use the solution of the previous step as the forecast solution.
- (2) Calculate the status of the control devices including PSS, excitation systems, and prime movers with their speed controllers according to the forecasting ω , I_f , P_e , I_G , and U_G .

- (3) Calculate the status of electrical devices, such as generators and loads, according to the forecasting of I_G , U_G , I_L , and U_L , as well as the output of control devices including U_f and T_m .
- (4) Calculate the value of the current source injected to power system of generators and loads. Modify the network equation coefficient matrix.
- (5) Solve the network equation and obtain the voltage of each bus.
- (6) Calculate other parameters such as the current of the branches.
- (7) If the maximum number of iterations is reached, the calculation of this time step ends. Otherwise, update the forecast solution by the solution of this iteration, and return to step 2 to execute next iteration.

In real-time electromechanical transient simulation, the solving process of devices is executed multiple times, and some solving processes in the network solving are repeated. The iterations also cause repeated solving processes. In the original FRTDS, due to the limitation of the calculation tasks, a large amount of memory space is used for the orders to describe these repeated solving processes. In the novel FRTDS for electromechanical transient simulation, the solving processes of the devices and sub-networks are regarded as functions, which effectively enhances the order storage ability. The design of the orders in the soft function solvers is introduced, with examples of the generator and the sub-network, in the next two subsections.

4.2. Design of Generator Orders

The mathematical model of the generator is represented by a sixth-order differential equation:

$$\begin{cases} \frac{dE'_q}{dt} = [E_{fq} - k_d E'_q + (k_d - 1) E''_q] / T'_{d0} \\ \frac{dE''_q}{dt} = [E'_q - E''_q + (X'_d - X''_d) I_d] / T''_{d0} \\ \frac{dE'_d}{dt} = [-k_q E'_d + (k_q - 1) E''_d] / T'_{q0} \\ \frac{dE''_d}{dt} = [E'_d - E''_d + (X'_q - X''_q) I_q] / T''_{q0} \\ \frac{d\delta}{dt} = \omega - 1 \\ \frac{dw}{dt} = (-Dw + T_m - T_e) / T_J \end{cases} \quad (1)$$

The first four differential equations describe the change in transient and sub-transient electromotive forces in the generator, and the last two equations are the rotor motion equations of the generator. E'_q , E'_d , E''_q , and E''_d refer to the transient and sub-transient electromotive forces on the q and d-axis of the generator. I_d and I_q refer to the currents on the d and q-axis, respectively. E_{fq} refers to the exciting voltage, δ refers to the power-angle of the generator, ω refers to the rotor angular velocity, and k_d , k_q , T'_{d0} , T'_{q0} , T''_{d0} , T''_{q0} , X'_d , X'_q , X''_d , X''_q , and D are constants. T_m refers to mechanical torque inputting to the generator, and T_e refers to the output electromagnetic torque.

The balance equation of stator circuit voltage in the generator can be represented as:

$$\begin{bmatrix} U_d \\ U_q \end{bmatrix} = \begin{bmatrix} F_d \\ F_q \end{bmatrix} - \begin{bmatrix} R_a & -X''_{qm} \\ X''_{dm} & R_a \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} \quad (2)$$

where U_d and U_q refer to the terminal voltages on the d and q-axis of the generator, respectively; R_a refers to the leakage reactance of stator; X''_{dm} and X''_{qm} are constants and they are determined by the transient and sub-transient reactance of the generator, respectively; and F_d and F_q can be obtained by solving the first four equations in Equation (1) via the numerical integration method. Convert the

d- and q-axis components of the generator voltage and current into the x and y components in the synchronous rotation coordinate via coordinate transformation:

$$\begin{bmatrix} \sin \delta & -\cos \delta \\ \cos \delta & \sin \delta \end{bmatrix} \begin{bmatrix} U_x \\ U_y \end{bmatrix} = \begin{bmatrix} F_d \\ F_q \end{bmatrix} - \begin{bmatrix} R_a & -X''_{qm} \\ X''_{dm} & R_a \end{bmatrix} \begin{bmatrix} \sin \delta & -\cos \delta \\ \cos \delta & \sin \delta \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix} \quad (3)$$

The current sources can be calculated using Equation (3):

$$\begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} G_{xx} & -B_{xy} \\ B_{yx} & G_{yy} \end{bmatrix} \begin{bmatrix} F_d \sin \delta + F_q \cos \delta - U_x \\ -F_d \cos \delta + F_q \sin \delta - U_y \end{bmatrix} \quad (4)$$

where:

$$\begin{cases} G_{xx} = [r_a + (X''_{qm} - X''_{dm}) \sin \delta \cos \delta] / (r_a^2 + X''_{qm} X''_{dm}) \\ B_{xy} = -(X''_{qm} \sin^2 \delta + X''_{dm} \cos^2 \delta) / (r_a^2 + X''_{qm} X''_{dm}) \\ B_{yx} = -(X''_{qm} \cos^2 \delta + X''_{dm} \sin^2 \delta) / (r_a^2 + X''_{qm} X''_{dm}) \\ G_{yy} = [r_a - (X''_{qm} - X''_{dm}) \sin \delta \cos \delta] / (r_a^2 + X''_{qm} X''_{dm}) \end{cases} \quad (5)$$

Move the part related to the terminal voltage of the generator in Equation (4) to the other side of the network equation. G_{xx} , B_{xy} , B_{yx} and G_{yy} are regarded as additional admittance, and they are added to the self-admittance elements in the network admittance matrix. Thus, the current source injected to the power system of the generator changes to:

$$\begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} G_{xx} & -B_{xy} \\ B_{yx} & G_{yy} \end{bmatrix} \begin{bmatrix} F_d \sin \delta + F_q \cos \delta \\ -F_d \cos \delta + F_q \sin \delta \end{bmatrix} \quad (6)$$

Constants and some iterative loop data in the solving process of the generator are not affected by other solving processes. These data are regarded as the private data of the generator and are stored in a part of the private storage area of the data storage unit by a certain sequence.

For the generator orders, some parameters need to be provided by microprocessor cores including the terminal voltage, the exciting voltage, and the mechanical torque, whereas some other parameters need to be transmitted to microprocessor cores including the electromagnetic power, the rotor angular velocity, the power-angle, the exciting current, the additional admittance, and the current source injected to power system. The amount of input and output data is relatively small; the data FIFO circuits are applied to transmit these data.

Other devices also contain a large amount of private data and a small amount of input and output data, whose orders can be designed using the same method as the generators.

4.3. Design of Sub-Network Orders

The network equation for electromechanical transient simulation can be represented as:

$$AU = I \quad (7)$$

where A refers to the network equation coefficient matrix, U refers to the bus voltage column vector, and I refers to the injection current source column vector. The part related to the sub-network is selected in the network equation, and the rows related to the boundary buses that are connected with the external network are put in the bottom. Thus, the network equation of the sub-network can be represented as:

$$\begin{bmatrix} A_{KK} & A_{KT} \\ A_{TK} & A_{TT} \end{bmatrix} \begin{bmatrix} U_K \\ U_T \end{bmatrix} = \begin{bmatrix} I_K \\ I_T \end{bmatrix} \quad (8)$$

where K refers to non-boundary buses in the sub-network that are not connected with external network, and T refers to boundary buses. Eliminate the network equation of the sub-network, and when it is eliminated to boundary buses, the network equation can be represented as:

$$\begin{bmatrix} \tilde{A}_{KK} & \tilde{A}_{KT} \\ 0 & \tilde{A}_{TT} \end{bmatrix} \begin{bmatrix} U_K \\ U_T \end{bmatrix} = \begin{bmatrix} \tilde{I}_K \\ \tilde{I}_T \end{bmatrix} \quad (9)$$

where \tilde{A}_{KK} is a upper triangular matrix, \tilde{A}_{TT} is the sub-matrix of the boundary buses, and \tilde{I}_T refers to the injection current source column vector of boundary buses after the elimination. \tilde{A}_{TT} and \tilde{I}_T are not eliminated and are transmitted to the main network or the upper-level sub-network. The voltages of all boundary buses in the sub-network are back substituted into Equation (9) after being solved. Thus, the voltages of non-boundary buses can be solved in the sub-network.

The elimination and the back substitution of the sub-network are executed in the soft function solvers. Elements of the network equation coefficient matrix and injection current sources of buses are needed in the elimination. Each element of the network equation coefficient matrix, which is a typical multi-valued parameter, has various possible values because the status of the power system changes, whereas injection current sources of buses are calculated by devices in the power system. Therefore, all these data need to be transmitted from microprocessor cores to a soft function solver. Many network equation coefficient matrix elements exist in the sub-network, so these elements are transmitted via the data cluster transmitter. The number of injection current sources of buses is relatively small, so these injection current sources can be transmitted via the input data FIFO circuit. After the elimination, the network equation coefficient matrix and the injection current sources of the boundary buses are transmitted to microprocessor cores via the output data FIFO circuit, which is applied to solve the main network or the upper-level sub-network. The network equation of the sub-network for back substitution is stored in the private storage area of the soft function solver, which facilitates the subsequent bus voltages calculation. After the voltages of boundary buses are solved, they are transmitted from the microprocessor cores to the soft function solver via the input data FIFO circuit. Then, the voltages of non-boundary buses are transmitted to the microprocessor cores after the back substitution.

The sub-networks with the same topology have the same expressions of elimination and back substitution. Therefore, selecting more sub-networks with the same topology while subnetting can further improve the reuse of the sub-network orders.

5. Case Study

The FRTDS for electromechanical transient simulation was built on a Virtex-7 FPGA VC709 board. Two micro-processing cores were constructed in FRTDS, whose orders were stored in the DDR3 SDRAM on the FPGA board with 244.1 MB storage space consumed. Sixteen soft function solvers were constructed in order to cooperate with the microprocessor cores to perform the real-time electromechanical transient simulation with a 10 ms time step. A data transmitter and multiple FIFO circuits were constructed to complete the data exchange between the microprocessor cores and soft function solvers. The constructed FRTDS had a utilization rate of 90% for the configurable logic block (CLB) in the FPGA chip, 81% for the block random access memory (RAM), and 85% for the digital signal processor (DSP). The FPGA hardware resources were fully utilized and well-proportioned for FRTDS.

A part of the power system in Fujian Province, East China was simulated as the example power system, which contains 106 generators, 257 buses, and 280 branches, as shown in Figure 6. The mathematical model of the generator is represented by a sixth-order model mentioned in Equation (1). The mathematical models of loads, excitation systems, power system stabilizers and prime movers with their speed controllers are same with models in PSD-BPA software developed by China Electrical Power Research Institute, Beijing, China [35].

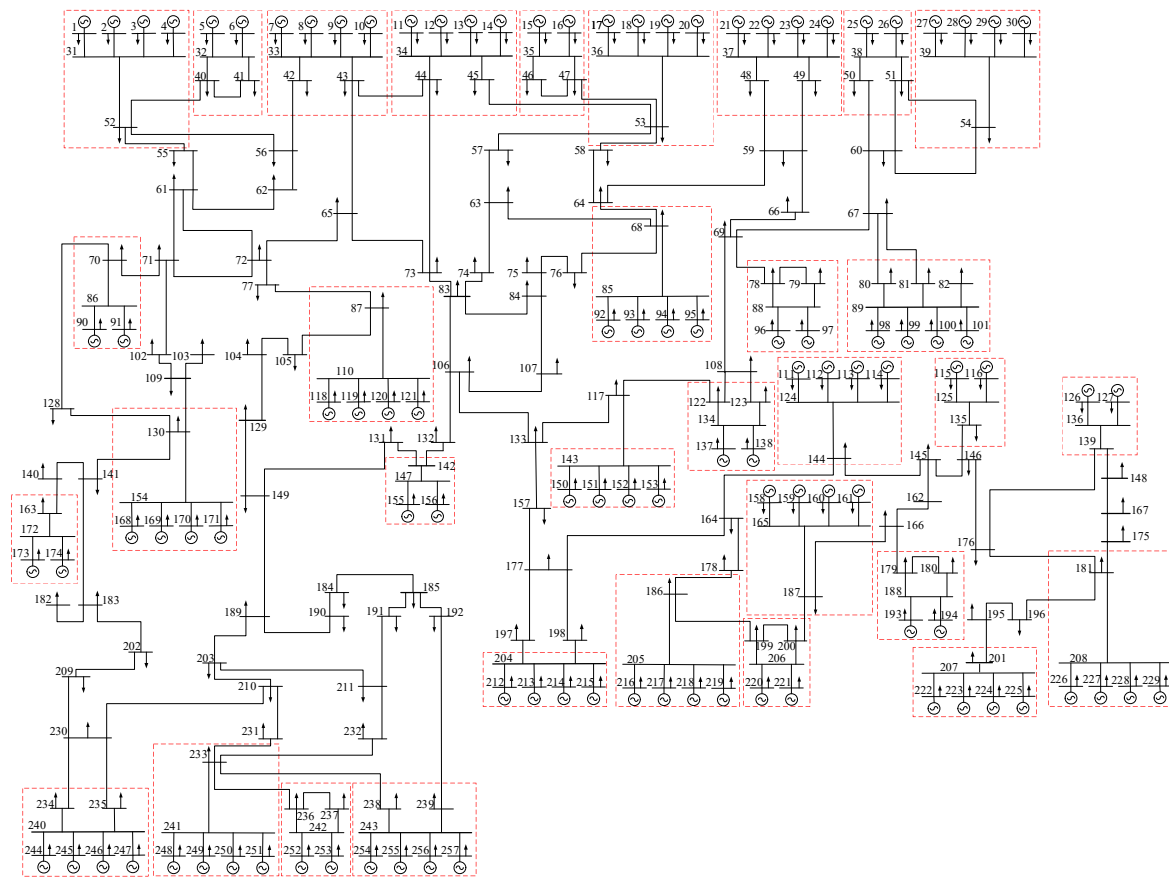


Figure 6. Online diagram of the example power system in East China.

The generator set and the related high-voltage bus were regarded as non-boundary buses of the sub-networks in the system, so 33 sub-networks were selected in the power system, as marked by the red dashed box in Figure 6. These sub-networks were solved by the soft function solvers. Each sub-network contained 2 or 4 generators, and 1, 2, or 3 boundary buses, so all these sub-networks correspond to 6 specific topologies. The elimination and back substitution of these 6 specific topologies and various devices in the system were solved in the soft function solvers, whereas data dispatch and solving of the main network were executed in the microprocessor cores.

The simulation of the example system was performed on a PC with the same simulation algorithm. The processor of the PC was an Intel Core i7-4790 (Intel, Santa Clara, CA, USA), which contains four cores, and the frequency of each core was 3.6 GHz. The PC was also equipped with 16 GB DDR3 RAM (Samsung, Seoul, Korea). The simulation script was coded by c++ language in Visual Studio 2013.

Five parallel real-time electromechanical transient simulation experiments of the example system were performed, to verify the feasibility of the novel FRTDS by comparison. The simulation was performed on the PC in experiment 1, whereas it was performed in FRTDS in experiments 2–5. Only microprocessor cores without a soft function solver were applied in experiment 2. The soft function solvers were applied to solve devices, and the microprocessor cores were applied to solve the whole network without any sub-network solved in the soft function solvers in experiment 3. The soft function solvers were used to solve both devices and sub-networks, but only FIFO circuits were used for the data exchange between soft function solvers and microprocessor cores in experiment 4. The soft function solvers were used to calculate both devices and sub-networks, and explicit and implicit channels were used to perform the data exchange in experiment 5, which is the model introduced in this paper.

The results of offline simulation showed the example power system is convergent with five iterations in a time step when the initial parameters resulted from the power flow calculation. The orders for these experiments were generated from the order generation software developed by the

key laboratory of smart grid of the ministry of Education, Tianjin University, China. The computation time for a single time step in experiments 2–5 was obtained from the analysis results of the orders for FRTDS. The computation time for the PC to complete the simulation of a single time step in experiment 1 resulted from dividing the total computation time by the amount of time steps. The computation times for a single time step in the five experiments are shown in Table 1.

Table 1. The computation time for a single time step in five parallel experiments.

Experiment Number	1	2	3	4	5
Calculation Time (ms)	214.62	22.87	11.43	9.07	8.36

The example system was modeled on SIMULINK (2017a, MathWorks, Natick, MA, USA) as well. Compared with the simulation result of the SIMULINK model, the accuracy of the result obtained from FRTDS was verified.

When the system run stably, a three-phase grounding short-circuit fault, with fault grounding resistance at $2\ \Omega$, was set on bus 57, which was cleared after 0.5 s. The voltage amplitude of bus 57, the voltage phase angle of bus 57, the power-angle, and the rotor angular velocity of the generator connected to bus 11 in the SIMULINK model and FRTDS are shown in Figures 7–10, respectively. The simulation results of PC are also provided as a reference.

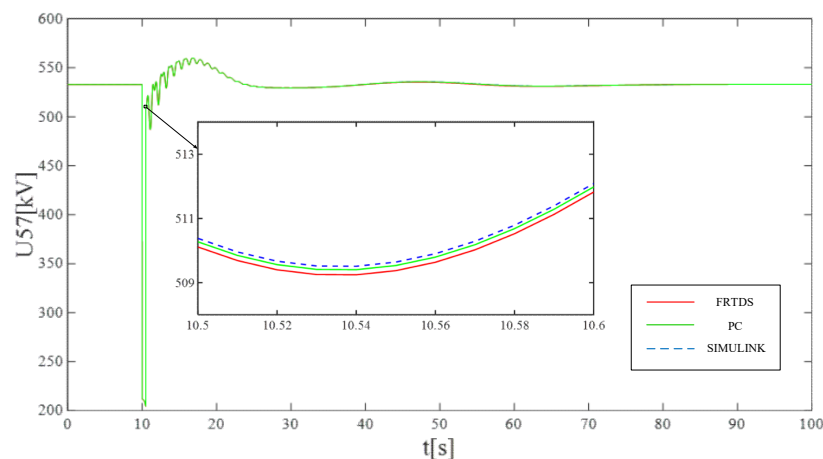


Figure 7. The voltage amplitude of bus 57.

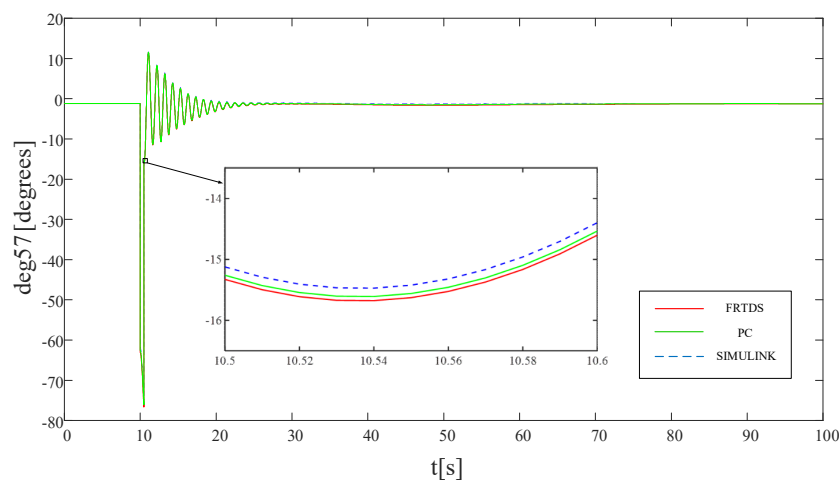


Figure 8. The voltage phase angle of bus 57.

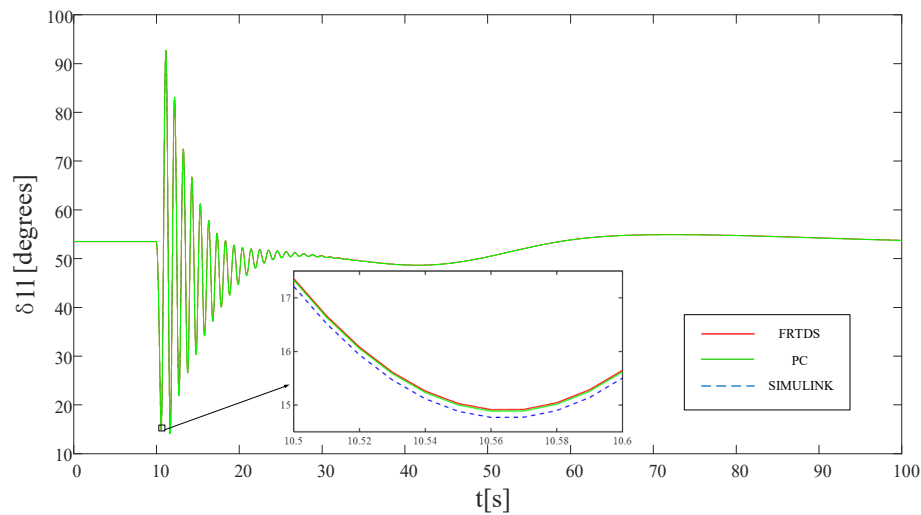


Figure 9. The power-angle of the generator connected to bus 11.

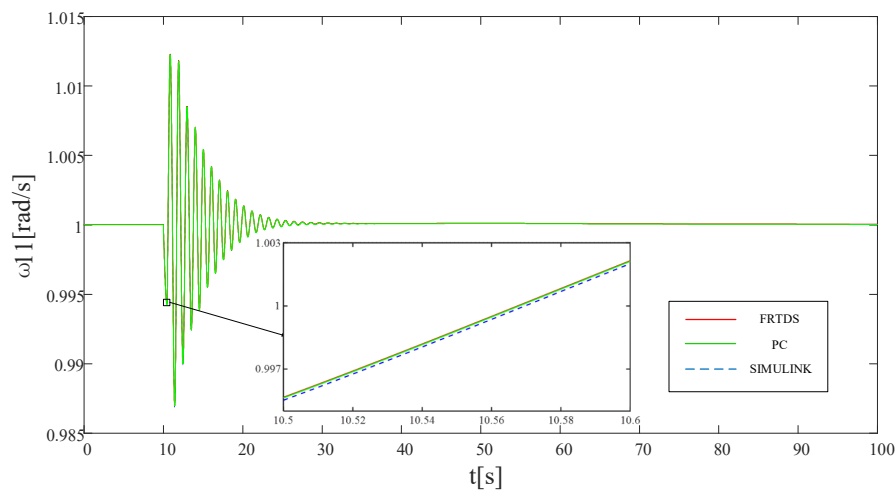


Figure 10. The rotor angular velocity of the generator connected to bus 11.

When the system ran stably, a three-phase grounding short-circuit fault, with fault grounding resistance at 0.01Ω , was set on bus 27, which was cleared after 0.5 s. Bus 27 was connected to a generator. The voltage amplitude and the voltage phase angle of bus 27, the power-angle, and the rotor angular velocity of the generator connected to bus 27 in the SIMULINK model and FRTDS are shown in Figures 11–14, respectively. The simulation results of PC are also provided as a reference as well.

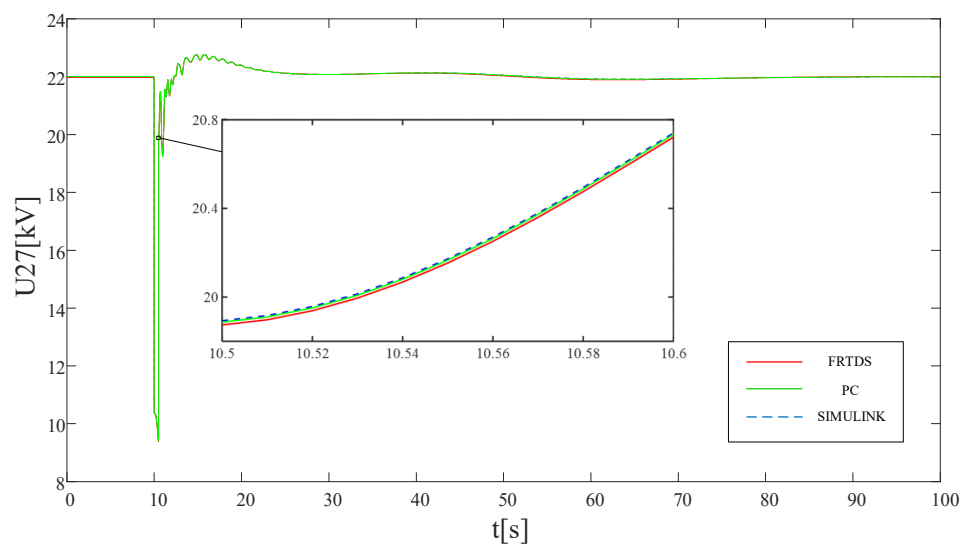


Figure 11. The voltage amplitude of bus 27.

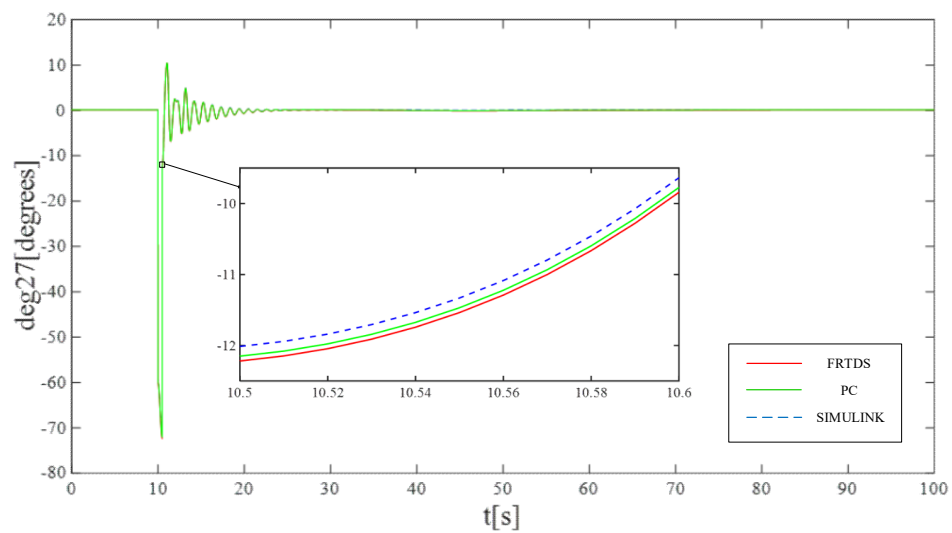


Figure 12. The voltage phase angle of bus 27.

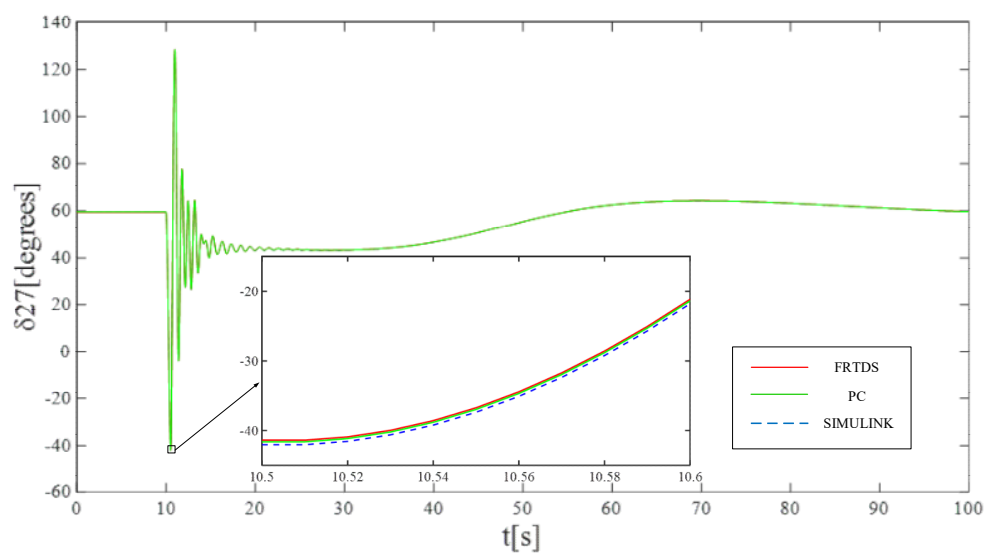


Figure 13. The power-angle of the generator connected to bus 27.

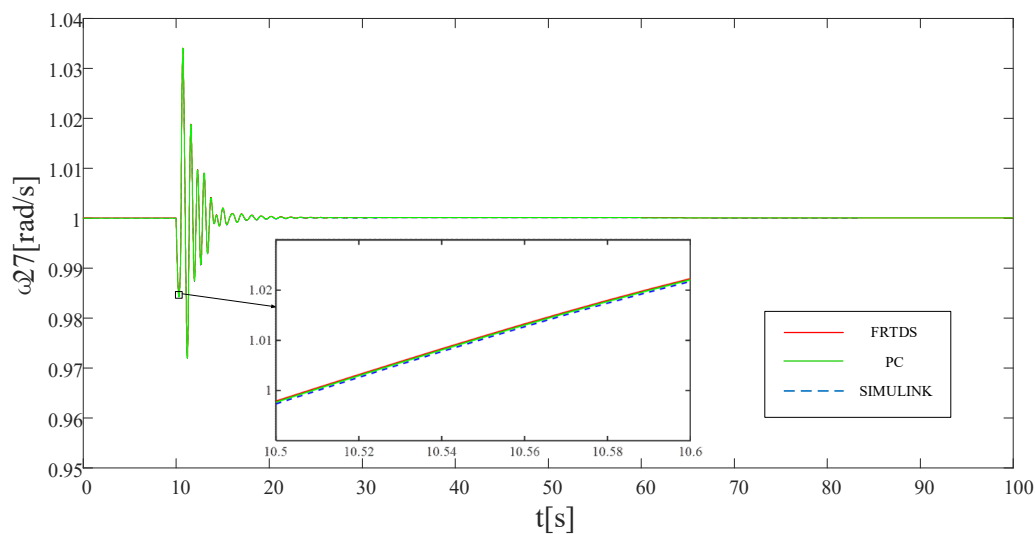


Figure 14. The rotor angular velocity of the generator connected to bus 27.

6. Discussion

The results of comparison showed that the computation time for a single time step in experiment 1 was far longer than that in experiments 2–5, which exceeded 10 ms. This indicated that the computation capability of FRTDS is beyond that of the PC, and the PC was not able to perform real-time simulation for the example system. The reason for this phenomenon is that the calculations are performed serially in a single core of CPU, whereas FPGA has considerable parallel computing capability and a deep pipeline mechanism that can perform parallel computation with multiple granularities. Thus, the parallelism of simulation was improved, and the simulation time was reduced.

In FRTDS, the simulation could not be performed in experiments 2 and 3 because the computation time for a single time step exceeded 10 ms in both experiments. Real-time simulation was performed in experiments 4 and 5. Compared with experiment 4, the computation time in experiment 5 was shorter, which indicated that the computing capability of the hardware in experiment 4 was the best in experiments 2–5. The computation time decreased from experiment 2–5, which indicated that the computing capability was effectively improved by applying soft function solvers, using soft function solvers to solve sub-networks, and using implicit channels for data exchange.

The partially enlarged Figures 7–14 show that the maximum error between results from FRTDS and the SIMULINK model was less than 1%, which verifies the accuracy of our novel FRTDS in both stable and critical situations. The error occurred due to the difference in data format and the simplification of the mathematical model of some devices in FRTDS.

In FRTDS, microprocessor cores were used for process control, data dispatch, and solving the main network. Soft function solvers were applied to solve the repeated calculation processes like devices and sub-networks. The relationship between the microprocessor cores and soft function solvers was similar to the relationship between CPUs and GPUs. The computing capability of the GPU-CPU collaborative real-time simulation platforms are limited because the synchronization overhead is inevitable, since CPU and GPU use different architectures, and the data transmission overhead is very large because many elements in the matrix must be transmitted between the CPU and GPU. Electromechanical transient simulation was performed collaboratively on the GPU and CPU [12], and the speedup ratio compared to the simulation using a CPU alone was high in large power systems. However, the real-time simulation of a power system with only 10 generators and 39 buses could not be performed in the simulation platform [12]. In FRTDS, microprocessor cores and soft function solvers were all constructed in the FPGA chip, and their calculation tasks were arranged uniformly, so there was no synchronization overhead. The large number of elements in the matrix can be pre-transmitted to soft

function solvers with only one order via the data cluster transmitter. Therefore, the data transmission overhead is significantly reduced and the computing capability of FRTDS is enhanced.

FRTDS has a small volume, and it is inexpensive and easy to install. The Virtex-7 FPGA VC709 board used for FRTDS only costs about 5000 dollars. FRTDS can be plugged into an industrial computer through the PCI-E interface, and high-speed Ethernet communication can be completed with actual devices through the SFP/SFP+ interface, so FRTDS can be easily installed into a real-time simulation platform.

At present, the novel FRTDS for electromechanical transient simulation still has some disadvantages that require further improvement. (1) The orders generation software needs to be improved. When using FIFO circuits for data exchange between the microprocessor cores and soft function solvers, the “first in, first out” principle must be met, which is a new restriction on task arrangement and creates more orders. Therefore, the orders generation software must be optimized to reduce the amount of the orders. (2) Research on co-simulation of multi-FPGA is necessary, especially the communication and task allocation among FPGA boards. The hardware resource requirement is considerably greater in multiple FPGA boards, so the scale of the simulated power system can be enlarged.

7. Conclusions

In this paper, a novel FRTDS for electromechanical transient simulation was described that solves the insufficient storage space problem for the orders in the FPGA chip in electromechanical transient simulation. Some new hardware designs applied in this paper are as follows: (1) SDRAM frequency multiplication was applied to ensure that the microprocessor cores can obtain 512-bit orders in the pipelining operation. (2) Soft function solvers were constructed in FRTDS. The solving processes of various devices and sub-networks were regarded as functions that can be solved in soft function solvers, which enabled the reuse of the orders. (3) The data was exchanged between microprocessor cores and soft function solvers through explicit and implicit channels, which improved the flexibility of the orders arrangement of the microprocessor cores and the capability of data exchange in FRTDS.

The elements in the network equation coefficient matrix of sub-networks were transmitted via the data cluster transmitter, whereas the injection current sources of the buses and the data of back-substitution and the devices were transmitted via FIFO circuits. The feasibility and accuracy of the novel FRTDS for electromechanical transient simulation were verified using part of a power system in East China as an example.

Given the new energy revolution, the smart grid has become a trend in the electric power industry. Smart grids have higher levels of safe and stable operation and highly intelligent grid dispatching. Some devices, such as united power flow controllers and energy management systems, play important roles in smart grids. Real-time electromechanical transient simulation is used for hardware-in-the-loop tests of these devices, and is used in power system training, online warning, and decision support. FRTDS has strong computing capability for real-time electromechanical transient simulation, economic advantages, and versatility. It has potential for application in the electrical power industry.

Author Contributions: Conceptualization, B.Z.; methodology, B.Z. and Y.W.; software, Y.W.; validation, Y.W., S.T. and Z.J.; formal analysis, B.Z.; investigation, Y.W. and Z.J.; resources, B.Z.; data curation, S.T.; writing—original draft preparation, Y.W.; writing—review and editing, B.Z.; visualization, Y.W.; supervision, B.Z.; project administration, B.Z.; funding acquisition, B.Z.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 51477114.

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

References

1. Strasser, T. Real-Time Simulation Technologies for Power Real-Time Simulation Technologies for Power System Design, Testing, and Analysis. *IEEE Power Energy Technol. Syst. J.* **2015**, *2*, 63–73. [[CrossRef](#)]

2. Tian, F.; Song, R.; Zhou, X.; Wu, Z.; Li, Y. Test and Analysis on Closed-Loop Simulation of Advanced Digital Power System Simulator and HVDC Control and Protection Devices. *Power Syst. Technol.* **2010**, *34*, 57–62. [[CrossRef](#)]
3. Tian, F.; Huang, Y.; Shi, D.; Xia, T.; Qiu, W.; Hu, X.; Li, Y.; Tang, L.; Zhou, X. Developing Trend of Power System Simulation and Analysis Technology. *Proc. CSEE* **2014**, *34*, 2151–2163. [[CrossRef](#)]
4. Dong, Y.; Wang, Y.; Han, J.; Li, Y.; Miao, S.; Hou, J. Review of High Efficiency Digital Electromagnetic Transient Simulation Technology in Power System. *Proc. CSEE* **2018**, *38*, 2213–2231. [[CrossRef](#)]
5. Li, Y.; Zhou, X.; Wu, Z. Personal Computer Cluster Based Parallel Algorithms for Power System Electromechanical Transient Stability Simulation. *Power Syst. Technol.* **2003**, *27*, 6–12. [[CrossRef](#)]
6. Xue, W.; Shu, J.; Yan, J.; Wang, X.; Han, Y. Cluster-Based Parallel Simulation for Power System Transient Stability Analysis. *Proc. CSEE* **2003**, *23*, 38–43. [[CrossRef](#)]
7. Zhai, P.; Yu, Q.; Li, Z.; Huang, Z.; Tong, L. Real-Time Electromechanical Transient Simulation System of Power System. *Power Syst. Technol.* **2008**, *32*, 42–45. [[CrossRef](#)]
8. Li, Y.; Zhou, X.; Wu, Z. A Parallel Complex Fault Computation Algorithm for Large Scale Power System Digital Simulation. *Proc. CSEE* **2003**, *23*, 1–5. [[CrossRef](#)]
9. Singh, J.; Aruni, I. Accelerating Power Flow Studies on Graphics Processing Unit. In Proceedings of the 2010 Annual IEEE India Conference, Kolkata, India, 17–19 December 2010.
10. Guo, C.; Jiang, B.; Yuan, H.; Yang, Z.; Wang, L.; Ren, S. Performance Comparisons of Parallel Power Flow Solvers on GPU System. In Proceedings of the 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Seoul, Korea, 19–22 August 2012.
11. Rahman, A.; Venayagamoorthy, G.K. Dishonest Gauss Newton Method Based Power System State Estimation on a GPU. In Proceedings of the 2016 Clemson University Power Systems Conference, Clemson, SC, USA, 8–11 March 2016.
12. Jalili-Marandi, V.; Dinavahi, V. SIMD-Based Large-Scale Transient Stability Simulation on the Graphics Processing Unit. *IEEE Trans. Power Syst.* **2010**, *25*, 1589–1599. [[CrossRef](#)]
13. Jalili-Marandi, V.; Zhou, Z.; Dinavahi, V. Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1255–1266. [[CrossRef](#)]
14. Zhang, N.; Gao, S.; Zhao, X. A Fine Granularity Parallel Algorithm for Electromechanical Transient Stability Simulation Based on Graphic Processing Unit. *Autom. Electr. Power Syst.* **2012**, *36*, 54–60. [[CrossRef](#)]
15. Pavlatos, C.; Vita, V. Linguistic Representation of Power System Signals. In *Electricity Distribution, Energy Systems Series*, 1st ed.; Karampelas, P., Ekonomou, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 285–295, ISBN 978-3-662-49432-5.
16. Pavlatos, C.; Vita, V.; Dimopoulos, A.C.; Ekonomou, L. Transmission lines' fault detection using syntactic pattern recognition. *Energy Syst.* **2018**, 1–22. [[CrossRef](#)]
17. Gao, C.; Han, R.; Li, W.; Liu, M. A Novel Power System Harmonic Analyzer Based on ARM and FPGA. In Proceedings of the 2011 International Conference on Electrical and Control Engineering, Yichang, China, 16–18 September 2011.
18. Razzaghi, R.; Mitjans, M.; Rachidi, F.; Paolone, M. An Automated FPGA Real-Time Simulator for Power Electronics and Power Systems Electromagnetic Transient Applications. *Electr. Power Syst. Res.* **2016**, *141*, 147–156. [[CrossRef](#)]
19. Jayalakshmi, K.; Ramanarayanan, V. Real-Time Simulation of Electrical Machines on FPGA Platform. In Proceedings of the India International Conference on Power Electronics, Chennai, India, 19–21 December 2006.
20. Chen, Y.; Dinavahi, V. Digital Hardware Emulation of Universal Machine and Universal Line Models for Real-Time Electromagnetic Transient Simulation. *IEEE Trans. Ind. Electron.* **2012**, *59*, 1300–1309. [[CrossRef](#)]
21. Dufour, C.; Jalili-Marandi, V.; Bélanger, J. Real-Time Simulation using Transient Stability, ElectroMagnetic Transient and FPGA-based High-Resolution Solvers. In Proceedings of the 2012 SC Companion: High-Performance Computing, Networking, Storage and Analysis, Salt Lake City, UT, USA, 10–16 November 2012.
22. Chen, Y.; Dinavahi, V. Multi-FPGA Digital Hardware Design for Detailed Large-Scale Real-Time Electromagnetic Transient Simulation of Power Systems. *IET Gener. Transm. Distrib.* **2013**, *7*, 451–463. [[CrossRef](#)]
23. Matar, M.; Iravani, R. Massively Parallel Implementation of AC Machine Models for FPGA-Based Real-Time Simulation of Electromagnetic Transients. *IEEE Trans. Power Deliv.* **2011**, *26*, 830–840. [[CrossRef](#)]

24. Wang, C.; Ding, C.; Li, P.; Wang, Z.; Lin, D.; Xing, F. FPGA-based Real-time Transient Simulation of Photovoltaic Generation System. *Autom. Electr. Power Syst.* **2015**, *39*, 13–20. [[CrossRef](#)]
25. Gong, W.; Zhu, Z.; Xu, S.; Li, X. A Parallel Simulation Model for Electromagnetic Transient of Diode Clamped Modular Multilevel Converter based on Equivalent Transformation. *South Power Syst. Technol.* **2015**, *9*, 30–37. [[CrossRef](#)]
26. Ji, F.; Fan, H.; Sun, Y. Modelling a FPGA-based LLC converter for real-time hardware-in-the-loop (HIL) simulation. In Proceedings of the 2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), Hefei, China, 22–26 May 2016.
27. Wang, Z.; Zeng, F.; Li, P.; Wang, C.; Fu, X.; Wu, J. Kernel Solver Design of FPGA-Based Real-Time Simulator for Active Distribution Networks. *IEEE Access* **2018**, *6*, 29146–29157. [[CrossRef](#)]
28. Li, P.; Zeng, F.; Wang, Z.; Wang, C.; Ding, C.; Zhao, J. I/O Interface Design for FPGA Based Real-time Simulator of Active Distribution Networks. *Proc. CSEE* **2017**, *37*, 3409–3417. [[CrossRef](#)]
29. Sloderbeck, M.; Meka, R.; Faruque, M.O.; Langston, J.; Steurer, M. Co-simulation of an FPGA-based electromagnetic transient model and a small time-step model in the RTDS real-time digital simulator. In Proceedings of the 2012 Grand Challenges in Modeling and Simulation, GCMS 2012, Part of SummerSim 2012 Multiconference, Genoa, Italy, 8–11 July 2012.
30. Wang, X.; Zhang, B.; Chen, M. Multi-Rate Real-Time Simulation Method Based on RTDS and FPGA Co-Simulation Platform. *Autom. Electr. Power Syst.* **2016**, *40*, 144–150. [[CrossRef](#)]
31. Wang, X.; Zhang, B.; Chen, X. Implementation of Fine Granularity Parallelization in Power System Real-Time Simulation. *J. Tianjin Univ. (Sci. Technol.)* **2016**, *49*, 513–519. [[CrossRef](#)]
32. Zhang, B.; Wang, L. Real-time Simulation Training System for Substation Based on FPGA. *Power Syst. Prot. Control* **2017**, *45*, 55–61. [[CrossRef](#)]
33. Zhang, B.; Fu, S.; Jin, Z.; Hu, R. A Novel FPGA-Based Real-Time Simulator for Micro-Grids. *Energies* **2017**, *10*, 1239. [[CrossRef](#)]
34. Zhang, B.; Wu, Y.; Jin, Z.; Wang, Y. A Real-Time Digital Solver for Smart Substation Based on Orders. *Energies* **2017**, *10*, 1795. [[CrossRef](#)]
35. Han, S.; Xu, Z.; Wu, X.; Jin, X. Comparison of Mathematical Models for Transient Stability between PSD-BPA and PSS/E. *South Power Syst. Technol.* **2010**, *4*, 67–71. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).