FPGA-Based Real-Time Electromagnetic Transient Simulation of Substation

Bingda ZHANG[a], Xiong CHEN[b]

Key Laboratory for Smart Grid of the Ministry of Education, Tianjin University, Naikai District Tianjin 300072, China

[a]email: bdzhang@tju.edu.cn, [b]tju_cx@163.com

Keywords: Electromagnetic Transients; Real-Time Simulation; Local Iterative Method; Instruction Stream Arithmetic Unit; decoding search method; FPGA

Abstract. At present, Real-time digital simulation of substation play an important role in analysing the characteristic of the substation, testing relay protection equipment, and building substation training simulation system. In order to create a high degree of similarity with the actual substation, this paper designs a FPGA-based real-time electromagnetic transient simulation of substation. A new substation simulation model is proposed to deal with nonlinear components based on the analysis of substation network structure. To take full advantage of FPGA parallelism and pipeline structure, a special arithmetic unit and decoding search method are shown. Simulation test shows that a simulation model of substation with more than 500 nodes could be processed in an EP4CGX150 chip within 40μs simulation timestep.

Introduction

Real-time digital simulation of substation play an important role in analyzing the characteristic of the substation, testing relay protection equipment, and building substation training simulation system, which has been widely used in the relevant departments of electric power[1]. In order to completely reconstruct an actual substation, all kinds of fault settings, switch operations and incident managements need to be achieved, which means that substation simulation model should involve disconnectors, grounding points, and fault setting points. Therefore, real-time electromagnetic transient simulation of substation primary system is not an easy task.

In recent years, the FPGA (field-programmable gate array) become a very effective algorithm accelerated device for its high degree of parallel processing ability and deeply pipelined design. Literature [2] achieves FPGA-based real-time simulation of transmission system which includes frequency-dependent transmission line model and conventional component models by parallel algorithm; nonlinear components are deal with the Newton's method, and the models of general motor and general transmission line are proposed in Literature [3-4], respectively; Literature [5-6] demonstrates that the timestep of converter real-time simulation can be smaller than 100ns. At present, resources in a FPGA chip become more and more abundant, but the costs become more and more lower, which makes FPGA-based real-time simulation of power system have a very good application prospect.In this paper, based on the FPGA platform, real-time electromagnetic transient simulation of substation considering nonlinear excitation is researched.

Simulation model of substation

To simulate magnetizing inrush current caused by transformer no-load closing and ferroresonance caused by magnetic saturation of voltage transformer, iron core magnetic saturation characteristic must be considered. For the network including nonlinear elements, it can be turned into the corresponding adjoint network with adjoint circuit as shown in Fig.1. On the basis of above, the nodal equation of adjoint network can be obtained, and the approximations of branch voltage, branch current at time t iteration (m-1) will be computed. The computations of equivalent conductance G^{m-1} and history current source $I_{his}^{(m-1)}$ at time t iteration (m) depend on the above

approximations, and the new approximations of branch voltage, branch current can be computed as the same way. When the approximations satisfy specified requirements, the iterative computations are completed, and the new iterative computations will begin from time $(t+\Delta t)$.

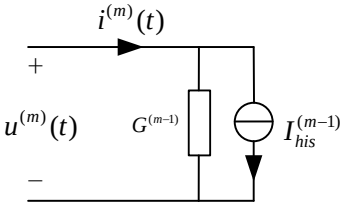

Fig.1: Adjoint circuit

For linear elements, the values of equivalent conductance and history current source at time t do not change with the number of iterations, and the formula for computing is relatively simple, not repeat them here.

For excitation inductances of the main transformer and voltage transformer, flux ψ in the inductor can be expressed as a monotone increasing function of excitation current *i*, and the tangent method is described by dynamic characterization of the inductor $L = d\psi / di$ and the remaining flux $\varphi = \psi - Li$. Thus, equivalent conductance and history current source of excitation inductance in the adjoint circuit can be expressed as:

$$\begin{cases} I_{his}^{(m-1)} = G^{(m-1)}u(t-\Delta t) + \alpha i(t-\Delta t) + \beta \\ G^{(m-1)} = \Delta t / (2L^{(m-1)}) \end{cases} \tag{1}$$

Where $\alpha = L^{(0)} / L^{(m-1)}$， $\beta = (\varphi^{(0)} - \varphi^{(m-1)}) / L^{(m-1)}$。

For convenience, the node just connecting with linear element is called FC-Node (fixed conductance node), the node connecting with nonlinear element called VC-Node (variable conductance node). According to the principle of VC-Nodes arranged after FC-Nodes, all the nodes are sorted, and the nodal equation at time *t* iteration (*m*) is expressed as:

$$\begin{bmatrix} G_{XX}^{(m-1)} & G_{XY}^{(m-1)} \\ [G_{XY}^{(m-1)}]^T & G_{YY}^{(m-1)} \end{bmatrix} \begin{bmatrix} u_X^{(m)}(t) \\ u_Y^{(m)}(t) \end{bmatrix} = \begin{bmatrix} I_X^{(m-1)} \\ I_Y^{(m-1)} \end{bmatrix} \tag{2}$$

Where X represents FC-Nodes, Y represents VC-Nodes.

The conductance G in Formula (2) has symmetry, so the Gauss reduction procedure is given as:

$$\begin{cases} G_{ij}^{'} = G_{ij} - G_{kj}G_{ki} / G_{kk} \\ I_i^{'} = I_i - I_k G_{ki} / G_{kk} \end{cases} \tag{3}$$

After reducing the number of FC-nodes, Formula (2) become:

$$\begin{bmatrix} G_{XX}^{'(m-1)} & G_{XY}^{'(m-1)} \\ 0 & G_0^{(m-1)} + G_{YY}^{(m-1)} \end{bmatrix} \begin{bmatrix} u_X^{(m)}(t) \\ u_Y^{(m)}(t) \end{bmatrix} = \begin{bmatrix} I_X^{'(m-1)} \\ I_0^{(m-1)} + I_Y^{(m-1)} \end{bmatrix} \tag{4}$$

Where $G_0^{(m-1)}$ 和 $I_0^{(m-1)}$ are called the equivalent conductance and equivalent current source, respectively.

$$G_{XX}^{'(m-1)} u_X^{(m)}(t) = I_X^{'(m-1)} - G_{XY}^{'(m-1)} u_Y^{(m)}(t) \tag{5}$$

$$(G_0^{(m-1)} + G_{YY}^{(m-1)}) u_Y^{(m)}(t) = \left[I_0^{(m-1)} + I_Y^{(m-1)} \right] \tag{6}$$

Although the $G_{XX}^{(m-1)}$、$G_{XY}^{(m-1)}$、$I_X^{(m-1)}$ in the Formula (2) have something to do with the switch state and fault setting, but which has no influence on the nodal voltage vector. Thus, there is no need to compute $G_{XX}^{(m-1)}$、$G_{XY}^{(m-1)}$、$I_X^{(m-1)}$ at each iteration, and making a computation at the end of the iteration is enough. In this case, the iterative computations according to Formula (2) are turned into computing the equivalent conductance and equivalent current source in the adjoint network composed by FC-Nodes; then the VC-Node voltage vector is computed by local iterative method according to the Formula (6); finally the FC-Node voltage vector is computed according to the Formula (5).

Computational complexity of node elimination is proportional to the node degree, a method of independence set based on minimum degree is introduced to reduce the computations. Because the elimination computations of each node in the independent set have nothing to do with each other, it is easy to achieve parallel computing by this way. Therefore, the method of minimum degree with independence set is used to arrange node elimination order. I.e., the node whose degree is minimum is selected as the first elimination node while its adjacent nodes are marked as have been visited, then choose the node whose degree is minimum from the nodes that have not been visited as the next elimination node until all the remaining nodes is visited so far, finally the elimination nodes form an independent set. From the remaining nodes, using the same way to establish an another independent set of elimination nodes until no node left.

Computing Platform of FPGA

FPGA has become a kind of effective algorithm accelerating device for its inherent highly parallel architecture and pipelining technique [7]. Now, FPGA chip with large capacity has millions of LEs (logic elements), but to achieve a real-time electromagnetic transient simulator of substation involving nonlinear excitation still need to consider the effective use of FPGA internal resources.

Computations of substation simulation model can be divided into 6 parts: ① computation of adjoint circuit history current source; ② computation of adjoint network injecting current; ③ a Gauss reduction procedure of nodal equation; ④ back-substituting of nodal equation; ⑤ computation of branch voltage; ⑥ computation of branch current. There is no need to design special arithmetic circuit for each of the 6 computation blocks due to the strong data dependence among them [8-9].

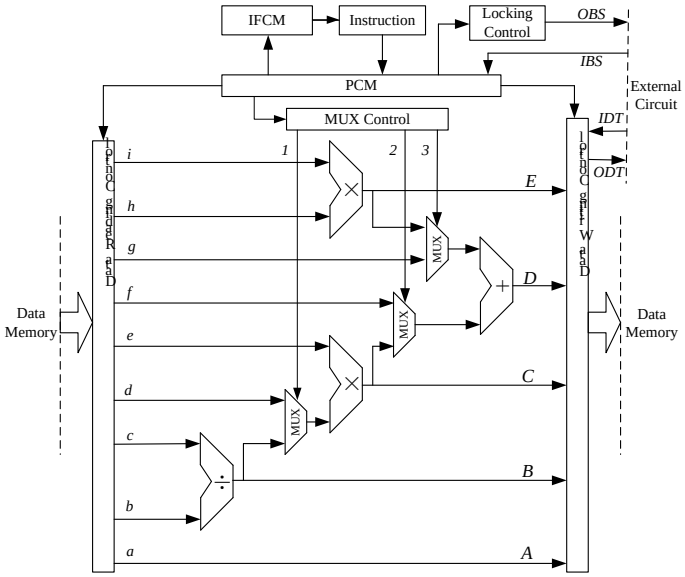

Fig.2: Instruction stream arithmetic unit

Since computation of some formulas are uncertain (such as the nodal current injections), it is not advisable to design an arithmetic circuit whose operation data in the formula are given all at once. For formulas such as $b/c\times e+g$ and $d\times e+h\times i$ are widely used for computing simulation model of substation, an arithmetic unit is designed in this paper as shown in Fig.2. As shown in Fig.2, b/c、$d\times e$、$f+g$、$b/c\times e+g$、$d\times e+h\times i$ single-output formulas and (b/c、$b/c\times e$、$b/c\times e+g$), ($d\times e$、$h\times i$) multi-output formulas can be achieved by controlling the MUX (multiplexer).

According to the computation steps for simulation model of substation, controls for MUX and reading/writing are achieved in the PCM (process control module). PCM need to be redesigned when the computation steps are modified, which means a high cost. Therefore, the computation steps for substation simulation model are described by instruction stream. Because format of instruction stream is user-defined and has nothing to do with the FPGA programming language, users can easily modify instruction stream. In order to save memory space of instruction stream, all

kinds of instruction are described by no more than 128-bit binary number. Thus, the main task of PCM is changed from providing computation steps into instruction translation.

In the process of real-time simulation, data input or data output occurs in the fixed time interval, so it is not necessary to stop a local iteration by detecting branch voltage or branch current before the maximum number of iteration coming. Therefore, jump instruction in the instruction stream which is executed in the IFCM (instruction fetch control module) has nothing to do with branch voltage or branch current, and IFCM only involves the instruction stream itself.

To fully exploit the parallel architecture of the FPGA, there are some instruction stream arithmetic units created in a FPGA chip. To ensure fast information exchange between instruction stream arithmetic units, OBS (output bitstream) and IBS (input bitstream) ports whose wide is 16-bit are set for instruction synchronization, ODT (output data transmission) and IDT (input data transmission) ports whose wide is 64-bit are set for data transmission.

FPGA resources can be divided into 6 kinds, programmable input/output unit, basic programmable logic element, embedded block RAM, routing resources, embedded functional units in bottom layer and embedded special hardcore.

In general, register based on basic programmable logic element is used for storing small-sized blocks of data that are frequently accessed or temporary accessed; dual-port RAM based on embedded block RAM is used for storing large-sized blocks of data that are sequential accessed or less frequently accessed. Therefore, data storage in Fig.2 is divided into register area and dual-port RAM area. If the data in a dual-port RAM need to be frequently accessed, the data should be transferred to the specified register in advance, which take full advantage of register. In order to achieve rapid data exchange between registers and dual-port RAMs, a direct line *a-A* in Fig.2 is set. The direct line *a-A* is associated with ODT port, which realizes data interaction between each instruction stream arithmetic unit.

Controls for data reading/writing in Fig.2 is one stage pipeline, but the multiplier is five stages pipeline, the divider is ten stages pipeline, the adder is seven stages pipeline. To make all the length of pipeline in the instruction stream arithmetic unit are shortest, nine ports for data reading, six ports for data writing and three ports for MUX control in Fig.2 are equipped with a FIFO, respectively, and the length of FIFO shown in Table 1.

Table 1: Length of FIFO

Name	*A*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
Length	1	1	1	11	11	16	16	11	11
Name	*A*	*B*	*C*	*D*	*E*	*IWR*	*1*	*2*	*3*
Length	2	12	17	24	17	2	11	16	16

The FIFOs are filled with “no operation” codes during initialization. After initialization, a code is written to a FIFO and, simultaneously, a code is read from a FIFO at every clock. Codes read from FIFO is used for controlling of data reading/writing and MUX, codes written to FIFO are translated from the current instruction and will control data reading/writing and MUX. The code buffer mechanism makes that reading/writing conflict checking is no longer a single instruction code, instead of codes from FIFOs in 9 ports for data reading and 6 ports for data writing. If you have multiple writing operations to the same address or reading/writing operations to the same RAM block more than two addresses, reading/writing conflict will happen.

There is a computation block for nodal equation, which contains Gauss reduction procedure and back-substituting. If a computation formula need more than one instruction to describe, it may be easy to cause data invalid because the output data of prior instruction is used as the input data of latter instruction before the output data has been written to the data memory. In order to prevent data invalid, it is forbidden to get code from the FIFOs in 9 ports for data reading which is same as the code written to the FIFOs in 6 ports for data writing in the last clock cycle.

Simulation Parameters

Using 2-RB (resistance branch with two values) to simulate switch and fault. If switch is opening

or fault is free, the value of 2-RB is large; If switch is closing or fault happens, the value of 2-RB is small. The piecewise linearization method is used for dealing with the magnetization curve of excitation inductance, and dynamic inductance is described by magnetization section. Inquiry method is adopted to get simulation parameters, and simulation parameters can be divided into the IV (inquiry value) and the DV (data value). The IV related to 2-RB and magnetization section is called the SW (state word). The DV represents the actual values of simulation parameters, and all the possible actual values belong to the same simulation parameter are sequential storage. In order to make the SW correspond to the actual value of simulation parameters, simulation parameters is divided into some kinds and each kind is given a special decoding mode. Therefore, in addition to the SW, the IV also contains a decoding mode and base address as shown in Fig.3.

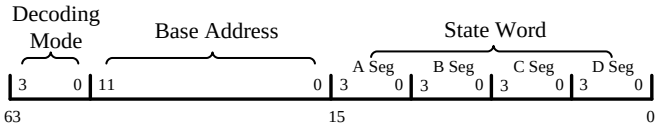

Fig.3: Structure of inquiry value

The process to find actual value of simulation parameters: the SW is translated into offset address according to the decoding mode in the IV, and the base address in the IV adds offset address to get the physical address of the DV, finally reading actual value of simulation parameters by physical address.

Here are two kinds of typical definition for state word and offset address computation.

① The bits in *A*, *B* segments are used for storing state of the 2-RBs which have the same characteristic, the bits in *C* segment are used for storing state of the remaining 2-RBs, the bits in *D* segment are used for storing the magnetization section number of the excitation inductance. Formula to computing offset is shown as follows:

$$Offset = XC_{max}D_{max} + CD_{max} + D \tag{7}$$

Where X is the number of the 1 existing in *A*, *B* segments, C_{max} is the maximum of *C* segment + 1, D_{max} is the maximum of *D* segment + 1.

② The bits in *A*, *B*, *C* segments are used for storing the magnetization section number of the three-phase excitation inductance, the bits in *D* segment are used for storing the 2-RBs. Formula to computing offset is shown as follows:

$$Offset = (2Y_3 + Y_2(Y_2 + 1) + Y_1(Y_1 + 1)(2Y_1 + 4)/6) + D \tag{8}$$

Where Y_1、Y_2、Y_3 are the maximum, the median, and the minimum of the *A*, *B*, *C* segment.

Simulation parameters influenced by switch state, fault setting, magnetization section are described by the IW (influence word). Obviously, the IW contains the information about self-conductance and mutual conductance G_i, G_j, G_{ij} of associated nodes *i*, *j*.

In general, the value of the 2-RBs become large only when branch current equal zero, but the value of the 2-RBs become small can occur in any moment. In order to avoid unnecessary computation of branch current, the information about node voltage u_i, u_j, branch voltage u_{ij}, and branch current i are added to the IW of switch state and fault setting, shown in Fig.4(a). The bit in the IV is affected by the IW, and the value range from 0 to 15.

At the beginning of each simulation, the branch resistance that is required to be small directly to be small, but the branch resistance that is required to be large need to compute branch voltage and branch current, only when the branch current equal zero, the resistance can be large. Once the value of resistance branch are changed, the bits in the IV will be modified according to its IW.

The IW of magnetization section also involves α, β in Formula(1) as shown in Fig.4(b). The segments in IV are affected by the IW of magnetization section, and the value range from 0 to 15.

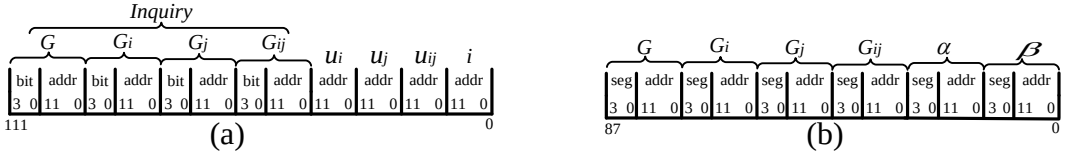

Fig.4: Influence word

At each step of the local iteration, the IW of magnetization section is used for modifying the SW. The IW of magnetization section and the SW are all stored in registers due to the influence of the frequent operation.

Simulation Test

The main wire in substation is shown in Fig.5, two main transformers are SFPSZ7-120000/220; double bus in 220kV side with 4 feeders, 110kV and 35kV sides are single bus with a tie which has 6 feeders. Phase to phase fault and grounding fault can be set in the simulation model of the bus, transformer, breaker and disconnector, voltage transformer, etc. The feeder is divided into three segments, disconnection fault can be set between each segment, and short circuit fault can be set in the head and tail of the feeder, the total number of simulation nodes up to 542.

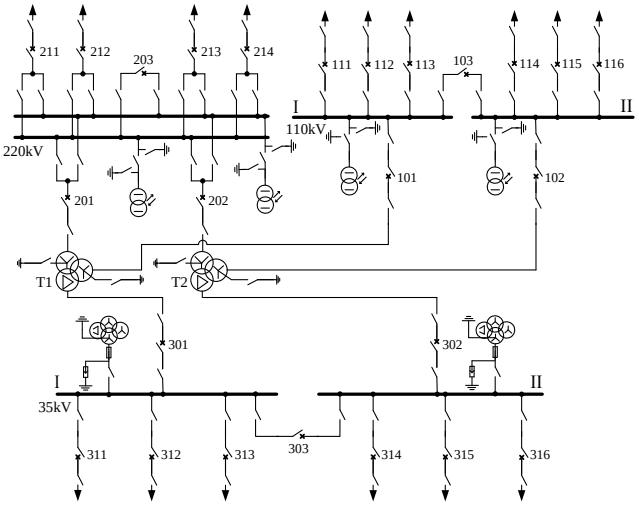

Fig.5: Main wire in substation

The FPGA-based real-time simulation for substation is implemented on a Altera Cyclone IV GX Development Board. A FPGA EP4CGX150 chip used in this board has the following features: 150K logic elements (LEs); 360 DSP blocks based on 18×18 multiplier; 6480K total RAM bits; 8 3.125-Gbps transceivers. There are 6 instruction stream arithmetic units created in a EP4CGX150 chip, which achieves the real-time electromagnetic transient simulation of substation as shown in Fig.5, with a timestep of 40μs.

①The magnetizing inrush current caused by main transformer no-load closing is shown in Fig.6, and the current scale coefficient is 800: 1.

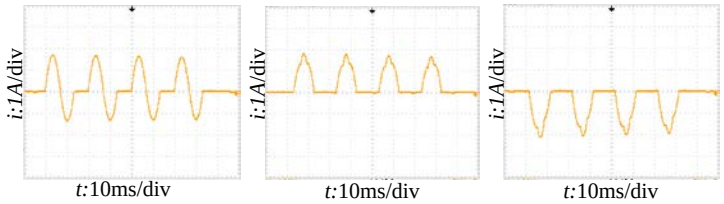

Fig.6: Current waveforms of no-load closing

②When segment I of the bus in the 35kV side only running with feeder 311, and grounding fault of the phase A last 0.06 seconds. At this point, segment I of the bus in the 35kV side happens ferromagnetic low frequency oscillation, shown in Fig.7, the voltage scale coefficient is 1000: 1.

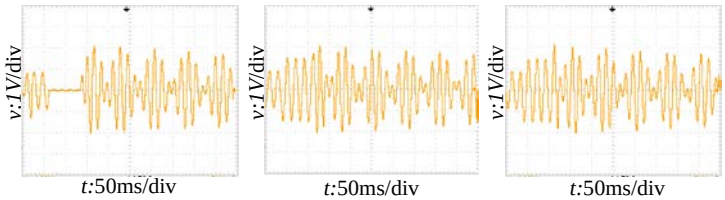

Fig.7: Voltage waveforms of ferroresonance

The simulation results show that the simulation waveforms generated by the FPGA-based

instruction stream arithmetic unit is almost consistent with simulation waveforms generated by using Matlab/Simulink simulation tool.

Conclusion

(1) The local iterative method only involves nonlinear element and independence set method based on minimum degree are used for arranging node elimination order. The values of resistance branch current are computed only when the values of resistance turn into large, which effectively reducing the amount of computation.

(2) The ideas of multiple-input/multiple-output and code buffer mechanism achieve highly parallel computation and shortest pipeline in the instruction stream arithmetic unit.

(3) The decoding search method not only reduces the time to determine the simulation parameters, but also rational use of the FPGA resources.

(4) For a substation simulation model with 542 nodes, can be simulated in real-time with a timestep of 40μs on an EP4CGX150 chip. Thus, a larger scale and smaller timestep real-time simulation can be implemented on a larger FPGA chip.

Acknowledgement

In this paper, the research work was supported by National Natural Science Foundation of China (Project No. 51477114) and Science and Technology Plan Projects of Tianjin (Project No. 13TXSYJC40400).

References

[1] Huang S, Liang XB, Gao XH. Design and realization of real time simulator for substation automation system. [J]. International Conference on Electricity Distribution, 2008, 3: 1-5.

[2] Chen Y, Dinavahi V. FPGA-based real-time EMTP. [J]. Journal of IEEE Transactions on Power Delivery, 2009, 24(2):1301-1309.

[3] Chen Y, Dinavahi V. An iterative real-time nonlinear electromagnetic transient solver on FPGA. [J]. Journal of IEEE Transactions on Industrial Electronics, 2011, 58(6): 2547-2555.

[4]Chen Y, Dinavahi V. Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation. [J]. Journal of IEEE Transactions on Industrial Electronics, 2012, 59(2): 1300-1309.

[5]Parma G, Dinavahi V. Real-time digital hardware simulation of power electronics and drives. [J]. Journal of IEEE Transactions on Power Delivery, 2007, 22(2): 1235-1246.

[6]Matar M, Iravani R. FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients. [J]. Journal of IEEE Transactions on Power Delivery, 2010, 25(2): 852-860.

[7]Myaing A, Dinavahi V. FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics. [J]. Journal of IEEE Transactions on Industrial Electronics, 2011, 58(1): 358-368.

[8]Luo Z, Martonosi M. Accelerating pipelined integer and floating point accumulations in configurable hardware with delayed addition techniques. [J]. Journal of IEEE Transactions on Computers, 2000, 49(3): 208-218.

[9]Bachir TO, David JP, Dufour C, Bélanger J. Effective FPGA-based electric motor modeling with floating-point cores. [C]. Annual Conference of the IEEE Industrial Electronics Society, 2010: 829-834.