International Conference on Applied Energy 2019
Aug 12-15, 2019, Västerås, Sweden
Paper ID: 0411

A DESIGN OF FPGA-BASED REAL-TIME SOLVER FOR ACTIVE DISTRIBUTION NETWORKS BASED ON BTF MATRIX BLOCK ALGORITHM

JIN Shuo[1], WANG Zhiying[1*], ZENG Fanpeng [2], FU Hao[1], WANG Xin[1], LI Peng[1]

1 Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China

2 State Grid Tianjin Dongli Electric Power Supply Company, Tianjin 300300, China

ABSTRACT

Linear equation solver of real-time simulators for active distribution networks (ADNs) requires significant computation power. Field programmable gate array (FPGA) has many merits such as highly parallel hardware structure and programmable characteristics, which shows great potential for the real-time solving of linear equations. Considering the sparsity of the linear equation, a parallel solver is designed in this paper based on block triangular form (BTF) matrix block algorithm. The design of off-line processing on host PC and online calculation on FPGA are proposed in detail. Simulation results are compared with PSCAD/EMTDC to validate the correctness and effectiveness of the parallel solver design.

Keywords: active distribution network (ADN), real-time simulation, field programmable gate array (FPGA), block triangular form (BTF), parallel solver design

NONMENCLATURE

Abbreviations	
ADN	Active Distribution Network
FPGA	Field Programmable Gate Array
BTF	Block Triangular Form
RES	Renewable Energy Sources
DG	Distributed Generator
COLAMD	Column Approximate Minimum Degree
ROM	Read-Only Memory
CSC	Compressed Sparse Column
PE	Processing Element
ADC	Associated Discrete Circuit

Symbols	
$\boldsymbol{i}$	Vector of the injected node currents
$\boldsymbol{G}$	Nodal conductance matrix
$\boldsymbol{u}$	Vector of the node voltages
$\boldsymbol{G}'$	Block upper triangular matrix of the nodal conductance matrix
$\boldsymbol{Q}$	Column permutation matrix
$\boldsymbol{P}$	Row permutation matrix
$\boldsymbol{G}'_{\boldsymbol{kk}}$	Matrix block
$\boldsymbol{P}'_{\boldsymbol{k}}$	Row permutation matrix of the matrix block
$\boldsymbol{Q}'_{\boldsymbol{k}}$	Column permutation matrix of the matrix block
$\boldsymbol{G}''_{\boldsymbol{kk}}$	Transformed matrix block
$\boldsymbol{P}'$	Row permutation matrix
$\boldsymbol{Q}'$	Column permutation matrix
$\boldsymbol{G}''$	Pre-processing node conductance matrix
$\boldsymbol{P}''$	Reordering matrix of the vector of the injected node currents
$\boldsymbol{Q}''$	Reordering matrix of the vector of the node voltages
$\boldsymbol{i}''$	Reordered vector of the injected node currents
$\boldsymbol{u}''$	Reordered vector of the node voltages
$\boldsymbol{L}''_{\boldsymbol{kk}}$	Upper triangular matrix of the transformed matrix block
$\boldsymbol{U}''_{\boldsymbol{kk}}$	Lower triangular matrix of the transformed matrix block
N_k	Number of the solving subtasks
N	Number of the updating subtasks
T_{Lki}	Startup time of the forward subtask
T_{Uki}	Startup time of the backward subtask
T_{Gki}	Startup time of the updating subtask

1. INTRODUCTION

The integration of renewable energy sources (RES) is changing the traditional distribution networks to active distribution networks (ADNs). The brought devices, such as distributed generator (DG) prototype and its controllers, have to be tested before use in the hardware-in-the-loop environment using real-time simulators. However, the complex dynamic characteristics and increasing scales of ADNs poses higher requirements for the real-time simulation [1]. Especially, solving the linear equations resulting from the mathematical models of ADNs is time-consuming and computational demanding, which restricts the performance of the real-time simulators.

Field programmable gate array (FPGA) has many merits such as highly parallel hardware structure and programmable characteristics, which provides a feasible solution to the linear equation solver. Some research efforts have been conducted to accelerate the solving of linear equations using FPGA. In reference [2], the method of pre-storing the inverse matrix of the nodal conductance matrix is adopted to meet the demand for simulation speed. In reference [3], the directly LU decomposition method is realized based on FPGA. Although the linear equations can be solved in existing simulators, for large-scale ADNs, the performance of the real-time solver still needs improving.

Considering the sparsity of linear equations, a parallel solver is designed in this paper based on BTF matrix block algorithm. The paper is organized as follows. Section 2 introduces the BTF matrix block algorithm. Section 3 proposes the detailed design of the parallel solver. In Section 4, a typical ADN with photovoltaics (PVs) are simulated in Section 4, and the correctness and effectiveness of the parallel solver design proposed in this paper are fully verified. Conclusions are stated in Section 5.

2. BTF MATRIX BLOCK ALGORITHM OF THE SPARSE NODAL CONDUCTANCE MATRIX

In this paper, the node analysis method [4] is adopted to form the simulation framework of the FPGA-based real-time simulator. The nodal equations are shown in equation (1), that need to be solved in each time-step.

$$\boldsymbol{i} = \boldsymbol{G}\boldsymbol{u} \tag{1}$$

Considering $\boldsymbol{G}$ is a sparse matrix, LU decomposition method is generally adopted, which consumes less storage space and computing resources. In order to increase the solving speed and make full use of parallelism, the matrix is divided into blocks, and each block is independent from others. Each matrix block can be solved by LU decomposition method.

BTF matrix block algorithm permutes the square sparse matrix $\boldsymbol{G}$ into block upper triangular matrix $\boldsymbol{G}'$. It firstly finds a column permutation matrix $\boldsymbol{Q}$ to give $\boldsymbol{GQ}$ a zero-free diagonal [5]. Then it finds a row permutation matrix $\boldsymbol{P}$ to put $\boldsymbol{PGQ}$ into the block upper triangular matrix [6]. The resulting permutation equation is shown in (2).

$$\boldsymbol{G}' = \boldsymbol{PGQ} \tag{2}$$

BTF algorithm has the following advantages: (a) Each diagonal block is independent from each other, and it can be solved in parallel. (b) The elements below the diagonal block are zeros, which simplifies the subsequent updating process. (c) The fill factors are not generated in the part above the diagonal block, which does not burden the updating process.

Taking the parallelism of FPGA hardware structure, the resulting blocks can be mapped into different hardware resources and solved in parallel.

3. DESIGN OF THE PARALLEL SOLVER

The parallel solver of the FPGA-based real-time simulator consists of the off-line processing part on host PC and the online calculation part on FPGA.

3.1 Off-line processing part

3.1.1 Matrix block

As shown in equation (2), BTF matrix block algorithm converts $\boldsymbol{G}$ to $\boldsymbol{G}'$. For each matrix block $\boldsymbol{G'}_{\boldsymbol{kk}}$ on the diagonal of $\boldsymbol{G}'$, $\boldsymbol{G'}_{\boldsymbol{kk}}$ is transformed using the column approximate minimum degree (COLAMD) algorithm. $\boldsymbol{G''}_{\boldsymbol{kk}}$, $\boldsymbol{P'}_{\boldsymbol{k}}$ and $\boldsymbol{Q'}_{\boldsymbol{k}}$ are obtained, as shown in (3). By using the COLAMD algorithm, the number of fill factors can be effectively decreased, and therefore the burden of online calculation will be reduced.

$$\boldsymbol{G''}_{\boldsymbol{kk}} = \boldsymbol{P'}_{\boldsymbol{k}}\boldsymbol{G'}_{\boldsymbol{kk}}\boldsymbol{Q'}_{\boldsymbol{k}} \tag{3}$$

According to the position of $\boldsymbol{G''}_{\boldsymbol{kk}}$ in $\boldsymbol{G}'$, $\boldsymbol{P'}_{\boldsymbol{k}}$ of each $\boldsymbol{G''}_{\boldsymbol{kk}}$ is combined into $\boldsymbol{P}'$. $\boldsymbol{Q}'$ is obtained in the same manner. $\boldsymbol{G}''$ can be obtained according to equation (4). Substituting equation (2) and equation (4) can be rewritten as (5).

$$\boldsymbol{G}'' = \boldsymbol{P}'\boldsymbol{G}'\boldsymbol{Q}' \tag{4}$$

$$\boldsymbol{G}'' = \boldsymbol{P}'\boldsymbol{PGQQ}' \tag{5}$$

Two reordering matrices $\boldsymbol{P}''$ and $\boldsymbol{Q}''$ are calculated by (6) and (7), which are stored in the corresponding read-only memories ROM_P and ROM_Q in FPGA.

$$\boldsymbol{P''} = \boldsymbol{P'P} \qquad (6)$$

$$\boldsymbol{Q''} = \boldsymbol{QQ'} \qquad (7)$$

Substituting equations (5)-(7) into equation (1), the transformed linear equations can be rewritten as (8)-(9),

$$\boldsymbol{P''i} = \boldsymbol{P''GQ''Q''^{-1}u} \qquad (8)$$

$$\boldsymbol{i''} = \boldsymbol{G''u''} \qquad (9)$$

where $\boldsymbol{i''}$ and $\boldsymbol{u''}$ can be expressed as (10) and (11).

$$\boldsymbol{i''} = \boldsymbol{P''i} \qquad (10)$$

$$\boldsymbol{u''} = \boldsymbol{Q''^{-1}u} \qquad (11)$$

3.1.2 LU decomposition

The LU decomposition algorithm is adopted to decompose $\boldsymbol{G''_{kk}}$ into $\boldsymbol{L''_{kk}}$ and $\boldsymbol{U''_{kk}}$, expressed by (12). $\boldsymbol{L''_{kk}}$ and $\boldsymbol{U''_{kk}}$ are respectively the coefficient matrices of the forward and backward substitution.

$$\boldsymbol{G''_{kk}} = \boldsymbol{L''_{kk}U''_{kk}} \qquad (12)$$

The non-zero elements of $\boldsymbol{L''_{kk}}$ and $\boldsymbol{U''_{kk}}$ are in the compressed sparse column (CSC) format. They are stored in the corresponding memories ROM_{Lk} and ROM_{Uk} respectively in FPGA. The Non-zero elements on the non-diagonal blocks in $\boldsymbol{G''}$ are processed in the same manner, and stored in ROM_{Gk}.

3.1.3 Solving startup time of subtasks

The forward and backward substitution of $\boldsymbol{G''_{kk}}$ are respectively divided into solving subtasks, and N_k is equal to the number of columns of the corresponding $\boldsymbol{G''_{kk}}$. The startup time of the forward and backward subtasks are T_{Lki} and T_{Uki} respectively, which can be obtained by performing the topology analysis of $\boldsymbol{L''_{kk}}$ and $\boldsymbol{U''_{kk}}$.

The calculation of the non-zero elements on the non-diagonal block of $\boldsymbol{G''}$ is assigned into the updating subtasks, and N is the dimension of $\boldsymbol{G''}$. The startup time of the updating subtasks is T_{Gki}, which can be obtained by performing the topology analysis of $\boldsymbol{G''}$.

T_{Lki}, T_{Uki} and T_{Gki} are stored in ROM_{Lki}, ROM_{Uki} and ROM_{Gki} in FPGA respectively, which will be used in the online calculation.

3.2 *Online calculation part*

The framework of the online calculation part is shown as Fig. 1. In the online calculation part, the calculation is carried out by (9)-(11).The function of the solving processing unit is to perform the forward and backward substitution of each $\boldsymbol{G''_{kk}}$. The updating processing unit deals with the non-zero elements on the non-diagonal blocks of $\boldsymbol{G''}$, and continuously updates the current vector $\boldsymbol{i''}$.

The processing elements (PEs) in each unit have the same structure, as shown in Fig. 2. The process is illustrated with the forward substitution as an example. When a subtraction timer counts to '1', the enable signal $ena[K]$ of the corresponding subtask K is set to '1'. While solving the unknown solution $y[K]$, the Kth column non-zero elements of $\boldsymbol{L''_{kk}}$ and the corresponding row number $L\_row[K]$ are transmitted into the PE in the form of a pipeline. After the floating point operations with $\boldsymbol{i''}$, the result is assigned to the register $i\_temp[K]$ and $\boldsymbol{i''}$ is updated.

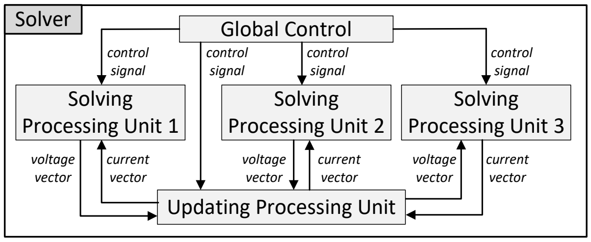

Fig 1 Framework of the online calculation part of the solver

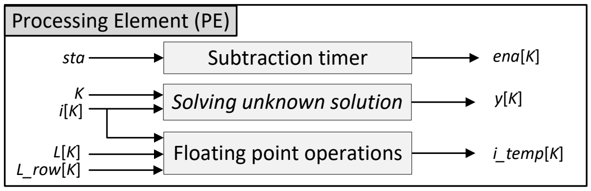

Fig 2 Processing Element

4. CASE STUDY

Four Stratix® V GX 530 development kits from Altera® are utilized as the carrier of the real-time simulator. The simulator is driven by a 125MHz clock.

4.1 *Test case*

In this paper, the IEEE-33 system with three PVs is simulated, as shown in Fig. 3. The time-step is 4μs. At 3.2 second, the irradiance arises from 500W/m^2 to 1000W/m^2.

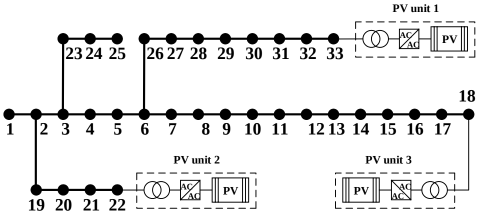

Fig 3 Test case

4.2 *Analysis of solving speed and scale*

The dimension of the linear equation resulting from the testing system is 120 and the number of non-zero elements is 342. After employing BTF algorithm, there are three 38-dimensional matrix blocks and six independent elements on the diagonal. After employing COLAMD algorithm, the formed matrix is shown in Fig. 4.

In the solver, the number of solving processing units is set to three, and the number of updating processing units is set to one.

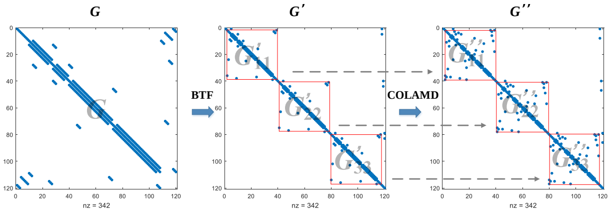

Fig 4 The formed matrix

The resource utilization and time consumption compared with the pre-storing inverse matrix method are shown in Table 1. It can be seen that the parallel LU decomposition method proposed in this paper consumes less resources obviously. Although the solution time of proposed method is longer, it can also well meet the requirement of real-time simulation. Moreover, this proposed method has potential advantages for larger linear equations.

Table 1 Comparison of resource utilization and time consumption

Solver	Resource utilization			Time consumption
	Logic	DSP	Memory	
Parallel LU decomposition	53 %	4 %	1.2%	5.848μs
Pre-storing inverse matrix	79 %	31%	16.7%	1.200μs

4.3 Analysis of solving accuracy

The simulation results are compared with PSCAD/EMTDC. Phase-A current and the DC voltage of PV unit 2 are plotted in Fig. 5 and Fig. 6.

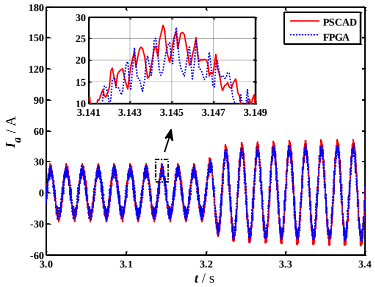

Fig 5 Phase-A current of PV unit 2

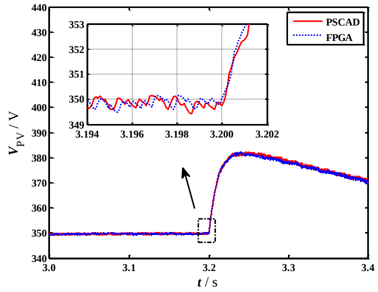

Fig 6 DC voltage of PV unit 2

It can be seen that the simulation results of the two simulation tools are nearly the same, which verifies the correctness of the proposed parallel solver. The errors mainly comes from the different modeling methods of power electronic devices. In PSCAD/EMTDC, large-small resistance is adopted, while in FPGA, it is represented by the associated discrete circuit (ADC) [7].

5. CONCLUSIONS

The FPGA-based real-time simulator for active distribution networks makes full use of the parallel computation of FPGA, which well meets the demand of real-time simulation of ADNs. This paper aims to realize the deep parallel solving of the sparse linear equations formed by the node conductance matrix. The parallel solver design is carried out in detail. The correctness and effectiveness of the proposed parallel solver are verified by the simulation cases of ADN. The design lays a foundation for improving the performance of real-time simulator for large-scale ADNs based on FPGA.

ACKNOWLEDGEMENT

This research is supported by the National Natural Science Foundation of China (51577127).

REFERENCE

[1] Jalili MV, Lok FP, Dinavahi V. Real-time simulation of grid-connected wind farms using physical aggregation. IEEE Transactions on Industrial Electronics, 2010, 57(9): 3010-3021.

[2] Wang CS, Ding CD, Li P, et al. Real-time transient simulation for distribution systems based on FPGA, Part I: module realization. Proceedings of the CSEE, 2014, 34(1): 161-167.

[3] Liu JD, Dinavahi V. Detailed magnetic equivalent circuit based real-time nonlinear power transformer model on FPGA for electromagnetic transient studies. IEEE Transactions on Industrial Electronics, 2016, 63(2): 1191-1202.

[4] Dommel HW. Digital computer solution of electromagnetic transients in single-and multiphase networks. IEEE Transactions on Power Apparatus and Systems, 1969, PAS-88(4): 388-399.

[5] Duff IS. Algorithm 575: permutation for a zero-free diagonal. ACM Transactions on Mathematical Software, 1981, 7(3): 387-390.

[6] Duff IS, Reid JK. Algorithm 529: permutations to block triangular form. ACM Transactions on Mathematical Software, 1978, 4(2): 189-192.

[7] Maguire T, Giesbrecht J. Small time-step (< 2μsec) VSC model for the real-time digital simulator. IPST International Conference on Power Systems Transients. 2005, 1-6.