AN FPGA-BASED SIMULATION SYSTEM FOR PHOTOVOLTAIC POWER GENERATION

Aravind R¹, BoreGowda H B², Manasa M G³

¹, ², ³Assistance Professor, Dept. of ECE, ¹, ²GSSS Institute of Engineering & Technology for Women, ³Maharaja Institute of Technology, Mysuru, India,
aravindr@gsss.edu.in, boregowda@gsss.edu.in, manasamg.ec@gmail.com³

Abstract— The real-time simulation of photovoltaic systems requires microsecond level simulation step size and it's hard to implement. In this paper, a real-time simulation system based on FPGA was designed and implemented which can realize the real-time simulation of photovoltaic power systems and the simulation step reached 1μs. System design and implementation method was introduced in detail, including the design of real-time simulation IP core. The whole process from system model to real-time simulation was discussed. And the function of the system was verified by conducting modeling and simulation of photovoltaic power system.

Keywords: photovoltaic power generation; real-time simulation; FPGA;

I. INTRODUCTION

The time-step of real time simulation for the electromagnetic transient of power system is generally selected in the range of 10~50μs, which is enough for the calculation of power system electromagnetic transient simulation in which there is no fast switches. However, lots of switches are existed in the photovoltaic generation systems, which comprise voltage source converter (VSC) that can transform DC to AC. The switching frequency of these switches may reach several tens of thousands Hz. As a rule, the simulation step frequency should be up to 20 times than the switching frequency for high speed switches so that the power system can be correctly simulation when the switch states change. So the time step of real time simulation for the photovoltaic generation systems should be smaller than 5μs which cannot be accomplished by x86 CPU or DSP.

The FPGA is reconfigurable logic device which has inherent parallelism and deep pipeline. The great computational power can be provided to implement the electromagnetic transient real time simulation for the photovoltaic generation systems by FPGA. The time step which is smaller than 5μs can be achieved. References [1] described the modeling method and the real time simulation of switch element, and implement time steps of microsecond level and sub-microsecond level, respectively. But, there are still many difficulties to implement the small step real time simulation for power system based on FPGA. The first of all is power system model must be described in hardware describe language (HDL) for FPGA-based simulation. This will be a very hard work for most of PV researcher. Secondly, development of PV small step model based FPGA is very time consuming, however, when you intend to simulate another power circuit, the corresponding model must be rebuilt in HDL [2, 3, 4].

In this paper, we present a novel implementation of FPGA real time simulation platform in small time step for PV generation system. In this platform, a general linear system solver is designed which can be used for calculating discrete state equations which size may be various. Otherwise, to simulate converter in PV system, a T type converter model is built in FPGA simulation platform. T type converter is prevalent in PV system.

II. THE SYSTEM ARCHITECTURE OF REAL TIME SIMULATION PLATFORM

In the FPGA-based simulation platform presented, as shown in Fig. 1, a Xilinx ML605 FPGA board is adopted for calculation and simulation as a simulator. A DSP TMS320- F2812 board is used to generate PWM signals as a controller. Simulation model should be built by...
utilizing SimPower- System blocksets in Simulink environment on host computer. In addition, to observe the outputs from real time simulator on oscilloscope and to implement converter controlled in close-loop, an IO chassis is introduced in which various kinds of IO cards are contained, such as AD, DA, DI, DO cards. The Host PC in Fig. 1 is used to model the PV system and controller, as well as takes the tasks of producing C code for DSP from controller model and obtaining matrix and parameters from the PV system. The ML605 FPGA board is interfaced to the Host PC through PCIe bus where data can be transferred at a speed of 5Gbps. In addition, a fiber link based on Aurora protocol is built between FPGA and IO chassis. As shown in Fig. 1, the proposed simulation process consists of three steps. The first step is to model controller using Simulink block sets which will send out fire pulses to simulator, and PV system models utilizing Sim Power System toolbox in the same model file. And then, the second step of model analyze will be automatically implemented by carrying on the background program, Embedded Coder and auto analyze program for PV system model. Embedded Coder is software tool provided by simulink, which can translate Simulink block diagram to embedded C code for DSP. The auto-analyze program is made in MATLAB m code, which can analyze PV system and then obtain the state equation and the corresponding parameters about the PV system model. Finally, in the step of model running in real time, the C code and state space equations from the model analyze step will execute on DSP and be solved on FPGA in real time, respectively.

III. THE ALGORITHM IMPLEMENTATION OF REAL TIME SIMULATION PLATFORM

The PV generation system is generally comprised of linear part which is comprised of lumped capacitances, inductances and resistances, as well as nonlinear converter apparatus which contains many switches, IGBTs and diodes. In this paper, a linear system solver and a converter simulation unit are designed and implemented, respectively.

A. THE PRINCIPLE AND IMPLEMENTATION METHOD OF LINEAR SYSTEM SOLVER

There are basically two ways to describe linear power system, state space equation and the nodal voltage analysis method. The nodal voltage method is the principle of The EMTP class algorithms. The state space equations also can be used to describe linear power system which only comprises of linear element.

Compared with the method of state space, the nodal voltage method has same effect with the state space method in which nodal voltages are adopted as state variables, while the correlations between nodal voltages make computation burden heavier than the state space method in which the state variables are irrelevant.

In this paper, we adopted the state space method to solve the linear power system. State variables represent the inductor currents and
capacitor voltages. Formula (1) is the general form of differential equations of linear system.

\[
\begin{align*}
\dot{X}(t) &= AX(t) + Bu(t) \\
Y(t) &= CX(t) + Du(t)
\end{align*}
\]

(1)

Where \( X \) is state vector, \( A \) is linear system state matrix, \( B \) is input matrix, \( C \) is output matrix, and \( D \) is direct matrix. Formula (2) is a general discrete form of state space equation. When back-euler integration algorithm is adopted to disperse state space equation, we can obtain (3). Moreover, the state space equation may be expressed in matrix form (4).

\[
\begin{align*}
X(n) &= A_m X(n-1) + B_m u(n) \\
Y(n) &= C_m X(n-1) + D_m u(n)
\end{align*}
\]

(2)

Where \( A_m, B_m, C_m, D_m \) can be presented as follows. In (3), \( h \) is the time-step of simulation.

\[
\begin{align*}
A_m &= (I - hA)^{-1} \\
B_m &= (I - hA)^{-1}Bh \\
C_m &= C(I - hA)^{-1} \\
D_m &= C(I - hA)^{-1}Bh + D
\end{align*}
\]

(3)

\[
\begin{align*}
X(n) &= A_m B_m X(n-1) + B_m u(n) \\
Y(n) &= C_m D_m X(n-1) + D_m u(n)
\end{align*}
\]

(4)

ABCD matrix, while RAM B is used to save a subset of state variables. It can be seen that the results of MUs are send to adder tree, the number of adder in adder tree is determined by the number of MUs. There is an accumulator at the end point of adder tree. The outputs of accumulator are comprised of the results of simulation sent to Output RAM and state variables at next time-step which will be fed back to RAM B in MU.

Due to fully exploit the parallel feather of FPGA, this approach will greatly accelerate matrix-vector multiplication.

B. THE ARCHITECTURE OF WHOLE SIMULATION ALGORITHM

As shown in Fig. 3, in addition to linear system solver and three level inverter unit, the whole algorithm flowchart for real simulation in FPGA includes the central controller, autosorting unit, signal sources and data distribution unit. This algorithm has an obvious advantages, which is universality, linear system with various linear elements and T type inverter can be real time simulation at time step of 1μs. As shown in Fig. 3, the outputs of three-level inverter are delayed to next time-step, and then combine them and the outputs of signal resource to a vector, at the same time, the new vector is resorted automatically according to the need of linear system solver, whereas the outputs of linear system solver are distributed automatically to three-level inverter unit and Output RAM which can be seen in Fig.2.
C. THE IMPLEMENTATION OF FPGA CORE OF REAL-TIME SIMULATION FOR PV SYSTEM

System Generator is XilinxFPGA graphical development tool. In this paper, we adopt System Generator to implement the simulator algorithm. As shown in Fig.5, the top diagram of simulator IP core contains two major modules, the “T” type three-level inverter block and the linear system solver block. In addition, the PWM control pulses interface are mapped to the inport of FPGA to receive the control signal from outside the FPGA chip. The control interface contains Start, Enable, Stepsize and Reset etc, which are mapped to the AXI-Lite bus registers in FPGA SoPC that can be modified by Host computer software in simulation process. The computing results are saved in the output Double Port RAM (DPRAM). Moreover, the output DPRAM is connected to AXI bus. The DMA controller can deliver the output data from the output DPRAM to Host computer or IO chassis in time.

In addition, in order to calculate the results of various systems which have different topology, the input and output of linear system solver is designed to transfer data in serial form, this approach greatly improves the flexibility of simulation calculation. At the same time, in order to improve the efficiency of calculation, the parallel algorithm, deep pipeline, multi-core multiplier and adder tree structure greatly improves the efficiency of calculation, and effectively reduces the simulation time-step.

To verify the real time simulation platform, we built a offgrid PV generator system simple model in Fig. 6. In this case, the photovoltaic cell is equivalent to a DC source. The output of three-phase three-level inverter to provide power to the load through RLC filter. Figure 6 shows the schematic of a PV system, the parameter is set as follows, DC source voltage Ud=600V, inductance L=0.1mH, capacitance C1=C2=1000μF, resistance R1=R2=0.001Ω, R3=10-4Ω, R=20Ω, the frequency of PWM control pulses fs=30KHz. The controller adopts open-loop control method, and the frequency of sinusoidal modulation wave is 50 Hz. Moreover, it can be seen in Fig. 6 that Measurement1 and Measurement2 are measurement points to measure line voltage and phase current before and after RLC filtering, respectively. In this experiment, offline simulation of the PV generator system is realized by using SimPowerSystem toolbox in Simulink environment, and the implementation of the realtime simulation is achieved by using the FPGA-based realtime simulator proposed in this paper, respectively. Both the time-steps of off-line and real-time simulation are set to 2μs.

IV. SYSTEM VERIFICATION FOR PV REAL TIME SIMULATION

![Fig. 4 The graphical block diagram of real-time simulation IP core.](image)

In real-time simulation, auto-analysis program is used to analyze the model of PV system as mentioned in section two, the matrix and parameters obtained by auto-analysis program are transferred to FPGA-based simulator via PCIe bus. The controller algorithm is also implemented in Simulink, and then using Embedded Coder to generate C code, finally, the compiled code is downloaded to DSP TMS320F2812 chip.

The SPWM signals are produced to control the inverter unit in simulator. The offline simulation based on Simulink and real time simulation in
FPGA are implemented, respectively. In Fig. 6 and Fig. 7, the waveforms of offline and real time simulation are shown. Compared with offline simulation waves and realtime simulation waves, good consistence can be observed from the waveforms. So the performance of proposed FPGA-based simulator for PV system has been verified.

![Waveform Diagram]

a, three-phase inverter output line voltage.

b, three-phase current.

c, three-phase line voltage of load side.

Fig. 6 Three-level inverter output waveforms in offline simulation

![Waveform Diagram]

b, three-phase current.

c, three-phase line voltage of load side.

Fig. 7 The waveforms of offline and real-time simulation

V. CONCLUSIONS

This paper presents a novel approach to simulate PV system, which may implement real time simulation in time step of 1μs. This is very important to analyze electromagnetic transient of PV system. The FPGA-based small time-step simulator in real-time is developed based on fully parallel algorithm for solving linear system that exploit the parallel processing capacity of FPGA. Moreover, the proposed linear system solver is very generalize, which is suitable for various kinds of topology system where the dimensions of state matrix might be changed. The time step of 1μs or less in real time for PV system simulation can be achieved. Another core module in proposed simulator is three-level “T” type inverter computation unit, which can solve the PV inverter model in 7 FPGA clock cycles in a time step.

A study case of real-time simulation using the proposed FPGA simulator for PV generator system has been presented to demonstrate the performance of the simulator. The results from FPGA-based real-time simulator show good agreement with the off-line simulation in Simulink environment. The proposed design can be used as a stand-alone solver in all FPGA real-time simulation or as a part of a whole simulator including PC-based and FPGA-based simulation units, and the two parts can run at different time-step, such as a time-step of 50μs for PC and 1μs for FPGA.
REFERENCES


