Hardware Emulation Building Blocks for Real-Time Simulation of Large-Scale Power Grids

Yuan Chen, *Member, IEEE*, and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—This paper proposes digital hardware building block concept for emulating power system networks on field-programmable gate arrays (FPGAs) in real time. Basic hardware emulation building blocks (HEBBs) for machines, transmission lines, nonlinear elements, and loads are presented to demonstrate how real-time simulation can be achieved for realistic power systems. All of the hardware modules were developed using the VHDL language for portability and extensibility. Employing multiple FPGAs, a large-scale power system consisting of 1260 three-phase buses is modeled in detail in real time to show both electromagnetic transients and electromechanical dynamics in the system. The advantage of HEBB-based modeling is that the design and development of new technologies can be accelerated by utilizing massive real-time digital simulators capable of modeling multiscale and multidomain dynamics.**

***Index Terms*—Field-programmable gate arrays (FPGAs), large-scale systems, parallel processing, power system simulation, real-time systems, smart grids.**

I. Introduction

MODERN electrical power grids are evolving into large and complex interconnected systems containing myriad equipment. Historically, power systems were planned and operated in a monolithic fashion from generation to utilization and contained only a minimum number of sensors and controllers. The bulk integration of disparate energy sources at multiple levels of the grid is creating system complexities and operational challenges that can only be resolved by the pervasive use of real-time digital computation, intelligent sensors, and advanced communication technologies integrated into the smart grid [1]–[3]. The design and implementation of smart grid technologies on a wide area requires the development of novel, real-time, parallel, and reconfigurable low-latency, high-bandwidth simulators. In particular, when a new controller, protective device, smart meter, or a security protocol needs to be tested and tuned in the hardware-in-the-loop (HIL) scenario before commissioning in the field, there is no alternative to a real-time digital simulator [4]–[8]. Transient network analyzers (TNAs) were the predecessors of fully digital real-time simulators; however, realization of large-scale power systems with a high level of complexity and fidelity using TNAs is practically impossible. There is a need for massive real-time simulators for the planning, design, implementation, and testing of smart grid technologies on a large-scale using detailed simulation models for grid components. Such simulators need to be able to model the grid dynamics on a multiscale basis. While the application of real-time digital simulators spans the entire frequency spectrum of power system studies, two of the most computationally demanding studies for large-scale power grids are the simulation of electromagnetic transients and electromechanical dynamics. In both of these studies, the resulting system model is a highly complex, large-dimension, nonlinear model. The real-time simulation should be able to seamlessly transition between slow and fast transients of various grid components.

The inter-operability of power systems, control, and communication also brings to the fore another challenge: multidomain simulation where different modeling philosophies, solution algorithms, and numerical methods coalesce to provide a high-fidelity simulation of the smart power systems [3], [9]. Existing simulators employing sequential x86 processors or digital signal processors (DSPs) have limited computational power, bandwidth, and the simulation algorithms for such inter-operability to exist. For example, real-time implementation of advanced multiplexing schemes, coding algorithms, communication channel emulation, and synchronization requires very fast signal-processing hardware. Field-programmable gate arrays (FPGAs) are ideally suited to meet these challenges in real time. FPGAs are high-density digital programmable logic devices containing millions of logic elements, distributed embedded memories, advanced arithmetic blocks, and high-bandwidth transceivers to emulate almost any physical system. FPGAs are currently being used for a wide variety of real-time control and simulation applications [10]–[20]. Their main advantages are fully parallel hardwired architecture, reconfigurability, fast clock speed, and sophisticated design tools, among many others. Currently available FPGAs from Altera and Xilinx can accommodate millions of logic gates at up to 1-GHz clock frequency allowing large-scale grid emulation.

Using FPGAs as the core computational engine, this paper proposes the concept of hardware emulation building blocks (HEBBs) for the real-time simulation of large-scale power grids. High-fidelity real-time hardware emulation of the fundamental power system apparatus such as generator, linear passive elements, transmission lines, transformers, nonlinear elements, and controls is presented. IEEE 32-bit floating-point precision is used in the hardware emulation for high accuracy. Multiple

Manuscript received September 17, 2012; revised December 10, 2012; accepted January 24, 2013. Date of publication January 30, 2013; date of current version December 12, 2013. This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC). Paper no. TII-12-0678.

Y. Chen is with RTDS Technologies Inc., Winnipeg, MB, Canada R3T 2E1 (e-mail: yuanchen@rtds.com).

V. Dinavahi is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2V4 (e-mail: dinavahi@ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2013.2243742

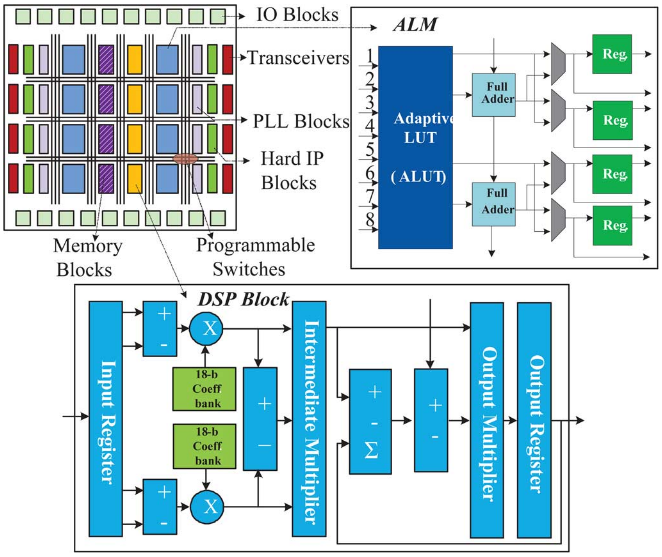


Fig. 1. FPGA hardware architecture.

FPGAs are employed to realize a detailed simulation of a large-scale system up to 1260 three-phase buses in real time showing both the slow and fast dynamics. This paper is organized as follows. Section II gives an overview of the FPGA design process. Section III presents the HEBB concept and details about how to build a large-scale power system using HEBBs. Section IV presents a detailed real-time case study of a large power system on a multi-FPGA architecture along with the real-time simulation results and discussion. Finally, conclusions are given in Section V.

II. FPGA Design Procedure

The generic FPGA is a two-dimensional (2-D) array of programmable logic cells interconnected by a matrix of wires and programmable switches. Each logic cell performs a basic logic function, while the programmable switches control the configuration of logic cells and the interconnection of the wires, thus achieving field programmability. For example, Fig. 1 illustrates the advanced 28-nm Altera Stratix V FPGA hardware architecture and its features [21]. The logic cells in this FPGA are logic array blocks (LABs) which are composed of several adaptive logic modules (ALMs). Each ALM contains an eight-input lookup-table (LUT) which are divided between two adaptive LUTs (ALUTs), two dedicated embedded adders, and four dedicated registers, providing sufficient resources for efficient implementation of logic, arithmetic, and register functions. Beside the basic logic blocks, modern FPGAs also provide massive embedded memory blocks, and other dedicated circuits such as the DSP blocks and the phase-locked loop (PLL) blocks. For example, the Stratix 5SGSD8 device has up to 1963 DSP blocks with up to 3926 18 × 18 bit multipliers for efficient implementation of high-performance (500 MHz) and computationally intensive signal-processing functions. Furthermore, the device also provides a unique array of integrated hard intellectual property (IP) blocks and serial transceivers (up to 28.05 Gbps).

The FPGA is a space-oriented logic device which enables full hardwired parallelism to be achieved to the extent permitted by the implemented user model and algorithm. A large number of customized parallel processing units can be easily configured. The integrated massive memory blocks can be partitioned into many independent types such as RAM, ROM, FIFO, single-port, or dual-port user memory units through which multiple data can be accessed simultaneously. Although multiple data can be processed in parallel in the FPGA, due to resource limitation, it is difficult to achieve the ideal parallelism (one data component per hardware module). In such cases, the *pipelining* technique has to be used. In a pipeline scheme, a function is divided into several stages by inserting registers between stages. Thus, multiple data can be processed at different stages at once, allowing a high computational throughput. The FPGA design procedure involves the design entry step using a hardware description language such as VHDL and the implementation step which involves mapping, placing, and routing on the physical device to generate the downloadable bitstream.

III. HEBB Concept for Real-Time Simulation

The HEBB is a modular parallel hardwired implementation of a specific type of power system component on an FPGA using the principles of parallel processing, deep pipelining, and distributed memory management. Each HEBB is a fully self-contained standalone digital hardware component with requisite interfaces that can be quickly replicated to be build large systems with several hundreds or thousands of power system components. The HEBB consumes a finite amount of digital hardware resources in terms of logic blocks, DSP blocks, and memory space and has a finite input–output latency. Therefore, by extension, the resource consumption for a large power system and the overall latency or the simulation time-step can be explicitly calculated. Depending on the FPGA resources available, and the required accuracy of the real-time simulation, multiple components of the same type can be pipelined through each HEBB.

A. Linear Lumped Passive Element HEBB

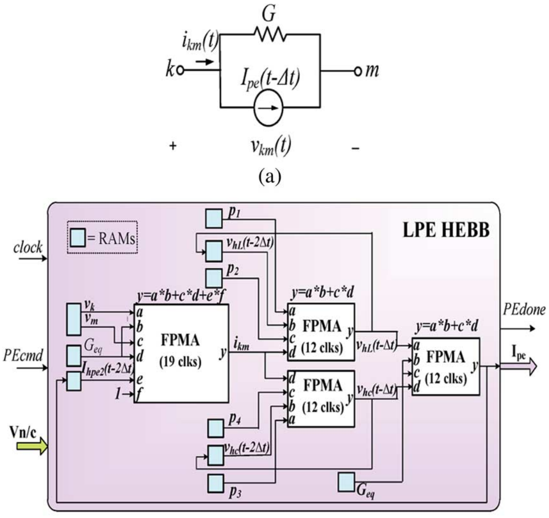
The linear lumped R, L, C elements or any combinations of them are used to represent loads, transformers, and other equivalent impedances in power systems. By applying the Trapezoidal rule of integration, the continuous-time differential equations for such elements can be represented by a common discrete-time model [22], as shown in Fig. 2(a), whose $v - i$ characteristic is given as

$$i(t) = Gv(t) + i_{pe(t-\Delta t)}, \tag{1}$$

where G is equivalent conductance, and $i_{pe}(t - \Delta t)$ is equivalent history current which is updated at each time-step. The LPE HEBB is designed for hardware emulation of linear lumped passive elements, as shown in Fig. 2(b). The main calculation of this HEBB is to update the history currents $i_{pe}(t - \Delta t)$. As can be seen in this figure, the parameters are stored in various RAMs and accessed in parallel for processing by the floating-point multiply-add (FPMA) units. These FPMA units are pipelined to achieve high throughput.

B. Electric Machine HEBB

Rotating machines are modeled using the universal machine (UM) model [23]. The UM model is a sufficiently generalized machine model which can accurately represent several types



(a)

(b)

Fig. 2. (a) Discrete-time model of lumped linear passive elements. (b) LPE HEBB.

of rotating machines for electromagnetic transient studies. The number of windings on the stator and rotor and the mechanical parts are fully customizable to model any specific machine. In this paper, an eighth-order UM model is employed to represent synchronous machines equipped with the exciter power system stabilizer (PSS) control systems. The discrete-time equation for the machine electrical side in the synchronously rotating $dq0$ frame is given as

$$\boldsymbol{v}_{dq0}(t) = -\boldsymbol{R}\boldsymbol{i}_{dq0}(t) - \frac{2}{\Delta t}\boldsymbol{\lambda}_{dq0}(t) + \boldsymbol{u}(t) + \boldsymbol{v}_{\text{hist}} \tag{2}$$

where $\boldsymbol{v}_{dq0}$, $\boldsymbol{i}_{dq0}$, $\boldsymbol{\lambda}_{dq0}$, $\boldsymbol{R}$, and $\boldsymbol{u}$ are vectors of voltages, currents, flux linkages, resistances, and speed voltages of the windings, Δt is the simulation time-step, and $\boldsymbol{v}_{hist}$ is the voltage history term. The mechanical dynamics for the UM are given as

$$T_m = J\frac{d\omega}{dt} + D\omega + T_e \tag{3}$$

where J, D, ω, T_m, and T_e are the moment of inertia, damping coefficient, rotor speed, load torque, and air gap torque, respectively. In the UM, the mechanical part is modeled as an equivalent electric network using lumped *RLC* elements, as seen in Fig. 3. The UM model is interfaced with the EMT network solution using the compensation method. As shown in Fig. 3, when the node voltage $\boldsymbol{v}_n$ without machines is ready, it is transferred into the $dq0$ frame, and the machine equations are solved. Then, the machine currents are transferred back into the abc frame and superimposed into the network to iteratively solve for the compensated voltage $\boldsymbol{v}_{c1}$. The iteration process terminates when the difference between the calculated and predicted ω is within the given tolerance.

The MAC HEBB is designed to emulate the UM model in the FPGA, as shown in Fig. 4. It mainly consists of six functional units. The Speed and Angle unit is responsible for predicting and calculating rotor speed ω and rotor angle β. The FrmTran unit transforms the UM quantities between the abc and $dq0$ frames. The Comp$\boldsymbol{i}_{dq0}$ unit is used to calculate the machine current $\boldsymbol{i}_{dq0}$. The flux linkages $\boldsymbol{\lambda}_{dq0}$ and torque T_e are solved in the Flux and Torque unit. The Update unit updates machine history terms $\boldsymbol{v}_{hist}$ and the history terms of the equivalent mechanical network. Finally, the CompVc unit calculates the complete voltages of the network $\boldsymbol{v}_{c1}$. Parallelism is used in the MAC HEBB in all of the functional units, for example, in the Comp$\boldsymbol{i}_{dq0}$ unit, a parallel Gauss–Jordan elimination (GJE) method is used to eliminate the rows of the matrix concurrently and then factorize.

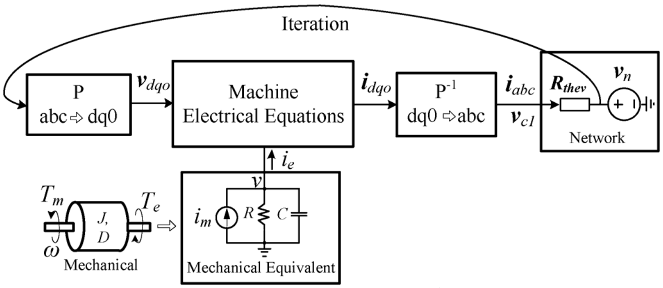


Fig. 3. UM model representation.

C. Transmission-Line HEBB

The transmission lines are modeled using the phase-domain frequency-dependent universal line model (ULM) [24]. The ULM is recognized as the most accurate and robust model for both overhead lines and underground cables under symmetrical and asymmetrical operating conditions. This model is first formulated in the frequency domain through the characteristic admittance matrix $\boldsymbol{Y}_c$ and propagation function matrix $\boldsymbol{H}$. In order to implement the model in the time domain, the elements of $\boldsymbol{Y}_c$ and $\boldsymbol{H}$ are approximated using finite-order rational functions. The (i,j) element of $\boldsymbol{Y}_c$ is expressed as

$$\boldsymbol{Y}_{c,(i,j)}(s) = \sum_{m=1}^{N_p} \frac{\boldsymbol{r}_{Y_c,(i,j)}(m)}{s - \boldsymbol{p}_{Y_c}(m)} + \boldsymbol{d}_{(i,j)} \tag{4}$$

where N_p is the number of poles and $\boldsymbol{r}_{Y_c}$, $\boldsymbol{p}_{Y_c}$, and $\boldsymbol{d}$ are the residues, poles, and proportional terms, respectively.

The (i,j) element of $\boldsymbol{H}$ is expressed as

$$\boldsymbol{H}_{(i,j)}(s) = \sum_{k=1}^{N_g}\left(\sum_{n=1}^{N_{p,k}} \frac{\boldsymbol{r}_{H,(i,j),k}(n)}{s - \boldsymbol{p}_{H,k}(n)}\right) e^{-s\tau_k} \tag{5}$$

where N_g denotes the number of propagation modes, $N_{p,k}$ and τ_k are the number of poles and time delays used for fitting the kth mode, and $\boldsymbol{r}_{H,k}$ and $\boldsymbol{p}_{H,k}$ are residues and poles for the k^{th} propagation mode. The resulting time-domain model of the ULM is shown in Fig. 5(a), where $\boldsymbol{G}$ is the equivalent conductance matrix, and $\boldsymbol{i}_{tlc\_k}, \boldsymbol{i}_{tlc\_m}$ are the history currents at sending-end "k" and receiving-end "m' of the line, which are calculated as follows:

$$\begin{aligned} \boldsymbol{i}_{tlc\_k} &= \boldsymbol{Y}_c * \boldsymbol{v}_k(t) - 2\boldsymbol{H} * \boldsymbol{i}_m(t-\tau) \\ \boldsymbol{i}_{tlc\_m} &= \boldsymbol{Y}_c * \boldsymbol{v}_m(t) - 2\boldsymbol{H} * \boldsymbol{i}_k(t-\tau) \end{aligned} \tag{6}$$

where the symbol * denotes the matrix-vector convolution.

The $\text{T}-\text{Line/Cable}$ HEBB is designed to emulate the ULM model in the FPGA, as shown in Fig. 5(b). Depending on the number of phases of the transmission line and the number of

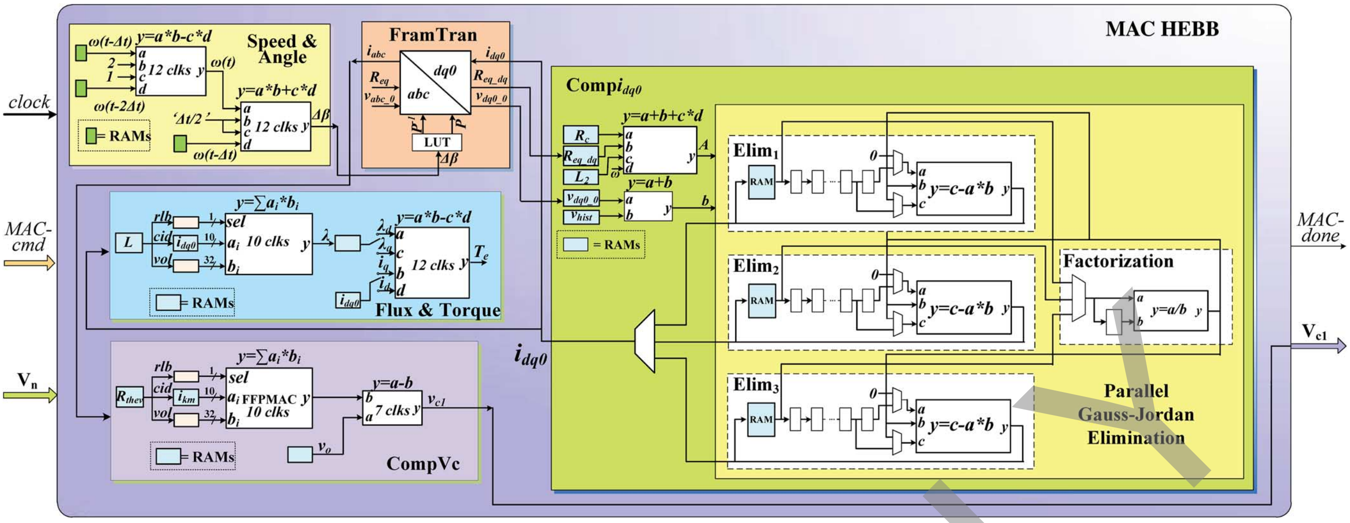
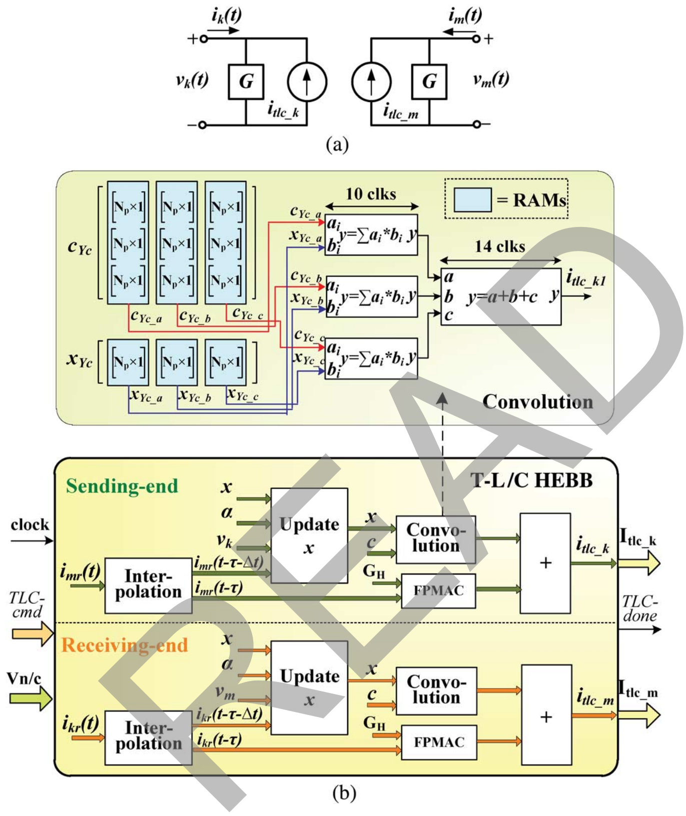


Fig. 4. MAC HEBB.



(a)

(b)

Fig. 5. (a) Discrete-time equivalent of ULM model. (b) T − Line/Cable HEBB.

poles of fitted rational functions, the convolution is in general quite expensive computationally. Parallel computation is essential to improve the computational efficiency. First, due to the traveling time delay of the ULM model, the sending and receiving ends of a transmission line are decoupled. Thus, the calculations at both ends can be conducted fully in parallel. Furthermore, for multiphase transmission lines, the calculation in each phase can be executed simultaneously.

D. Nonlinear Element HEBB

The commonly occurring nonlinear elements in power systems are nonlinear inductances and surge arresters [25]. Based on the different application requirements, the nonlinearity can be represented using either a piece-wise linear approximation or direct analytical nonlinear function. In both cases, Newton–Raphson (NR) method can be used in a piece-wise manner (PNR) or continuously (CNR) to iteratively arrive at the solution [15]. The compensation method is also employed here to solve the linear

$$\boldsymbol{v} = \boldsymbol{v}_n - \boldsymbol{R}_{thev}\boldsymbol{i} \qquad (7a)$$

and nonlinear

$$\boldsymbol{v} = \boldsymbol{f}(\boldsymbol{i}) \qquad (7b)$$

simultaneously, as shown in Fig. 6(a), where $\boldsymbol{v}_n$ is the node voltage computed without the nonlinear elements, $\boldsymbol{R}_{thev}$ is the Thévenin equivalent resistance seen from nonlinear elements' port, and $\boldsymbol{f}$ is the nonlinear function relating $\boldsymbol{v}$ and $\boldsymbol{i}$.

Applying the NR method to (7), the current $\boldsymbol{i}$ is solved using

$$\boldsymbol{J}(\boldsymbol{i}^{k+1} - \boldsymbol{i}^{k}) = \boldsymbol{v}_n - \boldsymbol{R}_{thev}\boldsymbol{i}^{k} - \boldsymbol{f}(\boldsymbol{i}^{k}) \qquad (8)$$

where $\boldsymbol{J}$ is the Jacobian matrix and $\boldsymbol{i}^{k+1}$ and $\boldsymbol{i}^{k}$ are the current vectors at the $(k+1)$th and kth iterations, respectively. To solve (8), a parallel GJE method is employed [15]. After the current $\boldsymbol{i}$ has been obtained, it is superimposed on the linear network to solve for the compensated voltage $\boldsymbol{v}_{c2}$.

The NE HEBB is designed to emulate nonlinear elements in the FPGA, as shown in Fig. 6(b). It consists of four main hardware submodules: 1) NLFunc (evaluating nonlinear function $\boldsymbol{f}(\boldsymbol{i}_{km})$ and $\partial \boldsymbol{f}(\boldsymbol{i}_{km})/\partial \boldsymbol{i}_{km}$; 2) CompJF (computing $\boldsymbol{J}$ and $-\boldsymbol{F}(\boldsymbol{i}_{km})$); 3) parallel GJE (GJE) for solving (8) for $\boldsymbol{i}_{km}$; and 4) CompVc (computing the compensated node voltage $\boldsymbol{v}_{c2}$.

E. Circuit Breaker HEBB

The circuit breaker in the power system is modeled as an ideal time-controlled switch. It opens at time t_{op} or closes at time t_{cl}, as shown in Fig. 7(a).

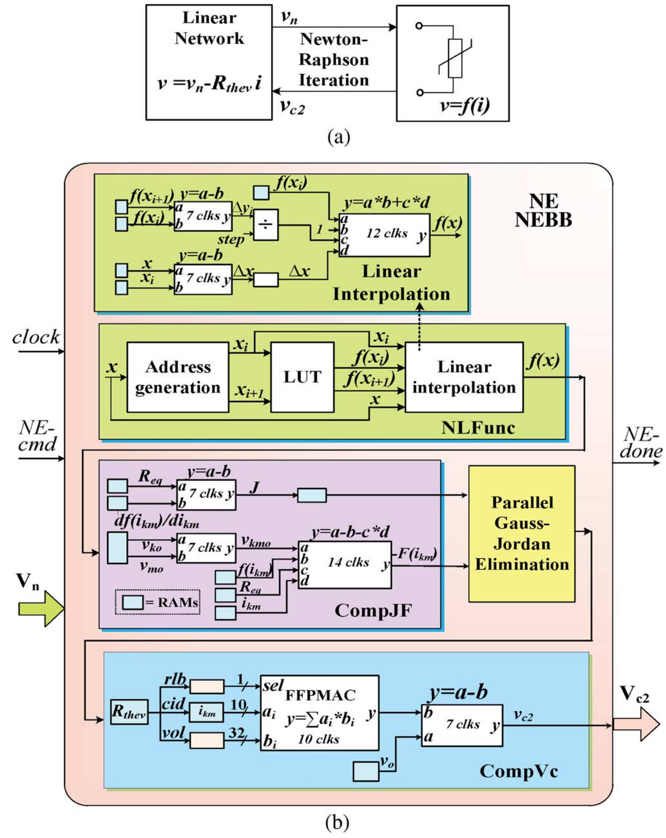


Fig. 6. (a) Nonlinear element solution method. (b) NE HEBB.

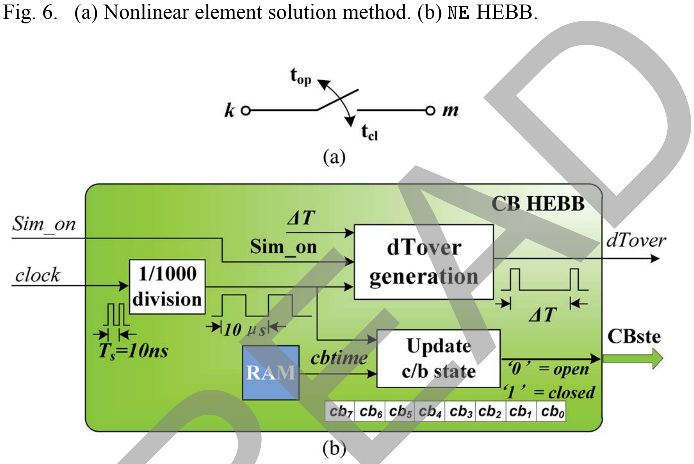


Fig. 7. (a) Circuit breaker model. (b) CB HEBB.

Since the circuit breaker is time-controlled, the core of CB HEBB is a real-time clock generator. As shown in Fig. 7(b), the input system clock frequency (100 MHz) is divided by 1000 to generate a 10-μs-period clock signal. This clock signal is counted and compared with the switch operation time $cbtime$ (integer of 10 μs) saved in a RAM. Once the circuit breaker's $cbtime$ is reached, the corresponding circuit breaker state bit in $CBste$ register is inverted using 0/1 for circuit breaker open/closed.

Another important function of this HEBB is to generate the $dTover$ signal, which indicates the end of the real simulation time-step Δt. When a simulation step is finished, the $dTover$ signal is checked. If it is not "1," the simulation step is finished within Δt, and thus the real-time simulation is achieved. Otherwise, the simulation step takes longer time than Δt, and the real-time constrain is not met.

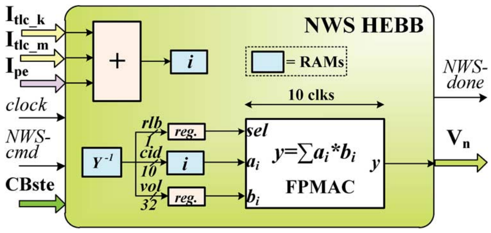


Fig. 8. NWS HEBB for linear network solver.

F. Linear Network Solver HEBB

The admittance matrix $\boldsymbol{Y}$ of the system is assembled based on the discrete-time equivalents of all the system functional components. Then, the system nodal equation

$$\boldsymbol{Y}\boldsymbol{v}_n = \boldsymbol{i} \tag{9}$$

is solved at every simulation time-step Δt for the node voltage vector $\boldsymbol{v}_n$ (without the machines and nonlinear elements). The current injection vector $\boldsymbol{i}$ is assembled from $\boldsymbol{i}_{tlc}$ and $\boldsymbol{i}_{pe}$. Equation (9) is solved by using a sparse matrix vector multiply of the inverse system nodal admittance matrix $\boldsymbol{Y}^{-1}$ with the current injections $\boldsymbol{i}$. While in this method the $\boldsymbol{Y}^{-1}$ has to be calculated *a priori* for all combinations of the circuit-breaker switches, it has the advantage of reducing memory utilization and increasing the calculation efficiency significantly. A formal sparse LU decomposition and substitution method can also be used to solve (9), provided there are sufficient hardware resources available.

As shown in Fig. 8, the current vector $\boldsymbol{i}$ is first assembled by an adder. Then, a floating-point multiply-accumulator (FPMAC) unit is used for sparse matrix-vector multiplication for calculating node voltage $\boldsymbol{v}_n$.

G. Overall Power System HEBB

To integrate aforementioned HEBBs into a single FPGA, a power system hardware emulation building block (PS HEBB) was built, as shown in Fig. 9. A control module was designed to coordinate the operation of other HEBBs to fulfill the power system transient simulation. The finite state machine of the PSHEBB is shown in Fig. 10. A real-time simulation time-step involves three stages: solving the network for voltages $\boldsymbol{v}_n$ without taking into account machines and nonlinear elements (S_0, S_1), solving machine equations and nonlinear equations for compensated voltages $\boldsymbol{v}_{c1}$ and $\boldsymbol{v}_{c2}$ (S_2, S_3), and finally updating history terms (S_4, S_5).

To accommodate large network sizes or reduce the simulation time-step for high-frequency transients, multiple HEBBS of various component types can be implemented depending on the available hardware resource of the FPGA. The aim is to implement more parallel HEBBs of the same type to maximize the resource utilization of the FPGA and minimize hardware latency. In this design, the Altera Stratix III EP3SL340 FPGA is used. Table I lists the main logic resources utilized by each type of HEBB. As can be seen, in one PS HEBB, two HEBBs

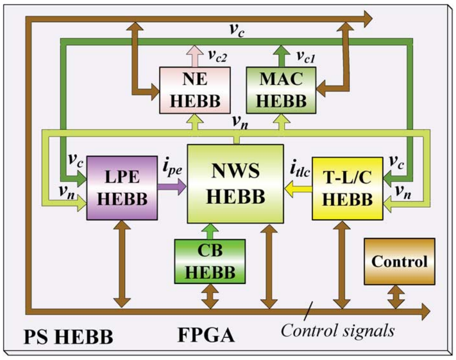


Fig. 9. Power system HEBB.

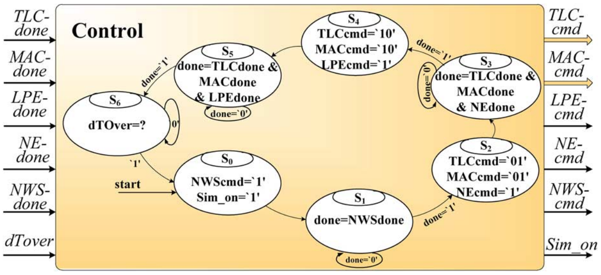


Fig. 10. Finite-state machine (FSM) for the power system HEBB.

TABLE I
FPGA RESOURCES UTILIZED BY INDIVIDUAL HEBBs

HEBB	Combinational ALUTs (Total: 270,400)	DSP 18-bit multipliers (Total: 576)	Memory (Kbits) (Total: 16,272)
1×LPE	33,481 (12.38%)	64 (11.11%)	842.16 (5.18%)
2×T-L/C	89,737 (33.19%)	192 (33.33%)	1,407.74 (8.65%)
2×MAC	49,951 (18.47%)	176 (30.56%)	1,745.18 (10.73%)
2×NE	23,384 (8.65%)	32 (5.56%)	406.80 (2.50%)
1×CB	50 (0.02%)	0 (0%)	408 (2.50%)
1×NWS	35,212 (13.02%)	64 (11.11%)	808.28 (4.97%)
Utilized	231,814 (85.73%)	528 (91.67%)	5617.54 (34.53%)

are implemented for each of the T − Line/Cable, NE, and MAC, while one HEBB each is implemented for LPE, CB, and NWS components. When the utilized resources exceed the available resources, pipelining can be used to accommodate more components at the cost of a higher latency and thus a larger time-step. A larger FPGA would be able to accommodate more HEBBs of the same type.

To emulate a large-scale power system network in real time, hardware parallelism and pipelining need to be exploited on a large scale, which can be realized by using multiple hardware functional modules; however, as seen from Table I, this is hard to achieve on a single FPGA due to the limitations of its logic resources. Multiple FPGAs have to be employed invariably to enable massive parallelism and pipelining for large-scale systems.

IV. MULTISCALE REAL-TIME SIMULATION OF A LARGE POWER GRID

For real-time simulation of large-scale power grids, multiple PS HEBBs have to be utilized. Practically, a large power system can be partitioned into smaller subsystems interconnected using transmission lines [20]. Due to the inherent travel time delay of the transmission line, each subnetwork can be simulated in a PS HEBB independently. At the end of each simulation time-step, the node voltages between the neighboring PS HEBBs have to be exchanged.

In this design, the multi-FPGA platform as shown in Fig. 11 is populated with ten Altera Stratix III EP3SL340 FPGAs arranged in a 2 × 5 matrix. Ten PS HEBBs are implemented in these ten FPGAs. As shown in Fig. 11, the dual-port RAM (DPRAM) is designed to send and receive the nodal voltages between neighboring HEBBs. The results of the real-time emulation were captured on the oscilloscope via a 125-MHz four-channel 14-bit digital-to-analog converter (DAC) card. To ensure fidelity, various verifications based on the behavior, function, and timing analysis were carried out throughout the design process.

A large-scale power grid is simulated in real time to validate the proposed power system HEBB simulator. This system is built using the modified IEEE New England 39-bus test system, as shown in Fig. 12. Each PS HEBB is assigned an equally partitioned subnetwork. The number of system components modeled in each subnetwork are shown in Table II. The subnetworks in neighboring PS HEBBs are interconnected using two transmission lines each. Note that the transformers are modeled using equivalent *RLC* elements. All transmission lines are modeled using ninth-order rational functions for $\boldsymbol{Y}_c$ and $\boldsymbol{H}$, and all synchronous machines utilize the eighth-order UM model. A time-step of $\Delta t = 45\ \mu$s was used for the real-time simulation of the large-scale grid. For capturing EM transients of up to 10 kHz in real time, a time-step of $\Delta t = 50\ \mu$s is acceptable according to Nyquist criterion. Based on Table II, the total number of three-phase components emulated in real time within a $\Delta t = 45\ \mu$s time-step for high-fidelity EM transient simulation include 300 generators, 1170 transmission lines, 990 linear passive elements, and 90 nonlinear surge arresters. Of course, on an FPGA with higher resources, either the time-step can be decreased significantly for a given system size or the number of emulated components can be scaled up for a fixed time-step. This choice is primarily determined by the needs of the specific real-time EM study being carried out, and can be achieved by emulating multiple parallel HEBBs of each component type on the FPGA comprising the PS HEBB.

Fig. 13 shows a sampling of real-time transient results captured on the oscilloscope side by side with offline simulation results from EMTP. A three-phase fault is simulated at Bus 2 (Fig. 12. The fault is initialized at $t_1 = 0.5$ s and cleared at $t_2 = 0.6$ s. Fig. 13(a) shows the three-phase voltages at Bus 3 over a period of 200 ms. The high-frequency electromagnetic transients can be seen clearly at both t_1 and t_2 at fault initiation and fault recovery, respectively. Fig. 13(b) shows the zoomed view of the voltage transients around fault recovery at time t_2. As can be seen, the differences between the real-time simulation and offline simulation are minor, and they can be mainly attributed to the limited resolution (14-bit fixed-point) of the DACs. When the 32-bit floating-point real-time results were

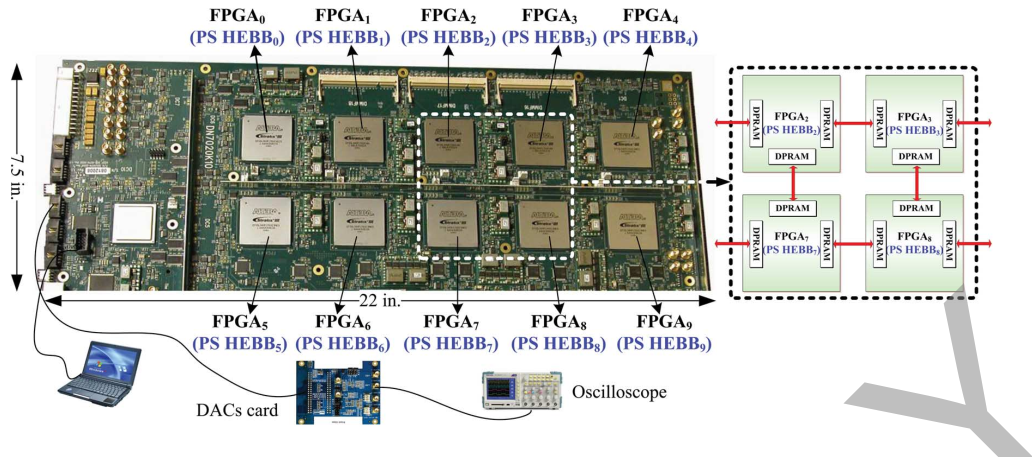


Fig. 11. Ten PS HEBBs emulated on a ten-FPGA hardware architecture.

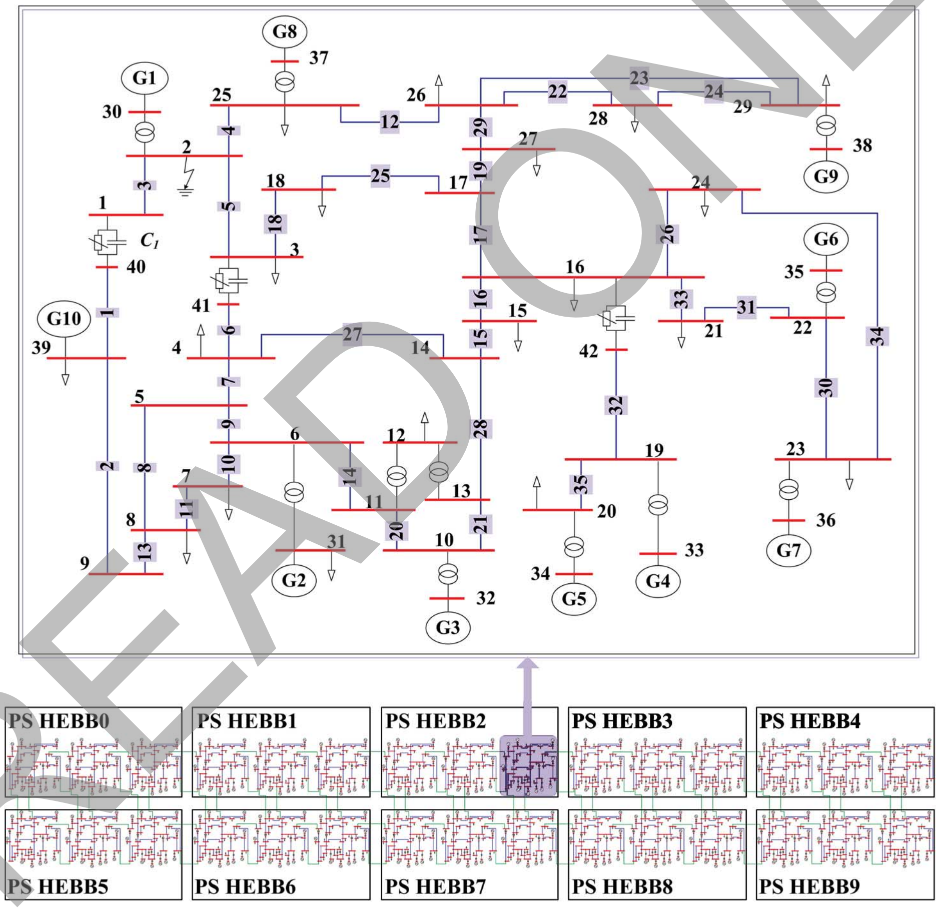
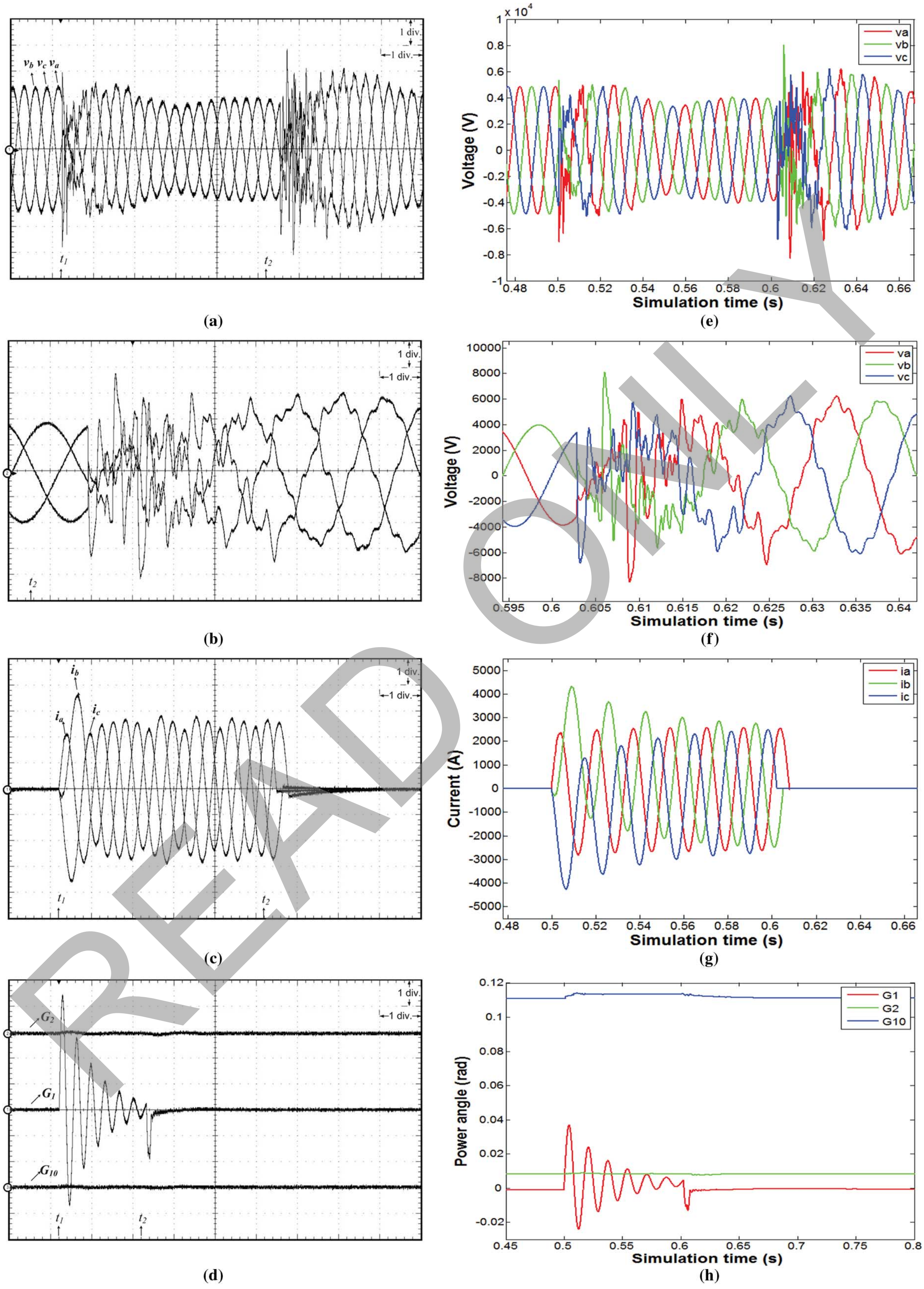


Fig. 12. Large-scale power system used in the real-time simulation case study.

TABLE II
NUMBER OF COMPONENTS IN EACH SUBNETWORK (PS HEBB) IN ONE FPGA

Three-phase component	Number
Linear lumped passive elements	99
Generators	30
Transmission lines	117
Nonlinear elements	9
Buses	126
Total single phase nodes	378

downloaded from the FPGA RAMs and plotted, they were almost identical to the offline simulation results. Fig. 13(c) shows the three-phase fault currents over a period of 200 ms at Bus 2, with the maximum fault current of 4.3 kA in phases b and c. The slower electromechanical dynamics of the synchronous generators over a period of 500 ms are shown in Fig. 13(d), which depicts the power angle δ oscillations of $\boldsymbol{G}_1$, $\boldsymbol{G}_2$ and $\boldsymbol{G}_{10}$. As can be seen from Fig. 12, $\boldsymbol{G}_1$ is the nearest generator to the fault, while $\boldsymbol{G}_2$ and $\boldsymbol{G}_{10}$ are further away from the fault. During the



(a)

(e)

(b)

(f)

(c)

(g)

(d)

(h)

Fig. 13. (a) Real-time waveforms for Bus 3 voltages during a three-phase fault at Bus 2. x-axis: 1 div. = 20 ms; y-axis: 1 div. = 60 mV ($\equiv$ 1.8 kV). (b) Zoomed view of Bus 3 voltages during a three-phase fault at Bus 2. x-axis: 1 div. = 5 ms, y-axis: 1 div. = 60 mV ($\equiv$ 1.8 kV). (c) Real-time waveforms of fault currents during a three-phase fault at Bus 2. x-axis: 1 div. = 50 ms; y-axis: 1 div. = 60 mV ($\equiv$ 1.2 kA). (d) Real-time results of power angles δ for generator G_1, G_2, and G_{10} during a three-phase fault at Bus 2. x-axis: 1 div. = 50 ms; y-axis: 1 div. = 30 mV ($\equiv$ 0.0082 rad). (e)–(h) Offline simulation results.

fault, the oscillations of $\boldsymbol{G}_1$ are more pronounced compared with those of $\boldsymbol{G}_2$ and $\boldsymbol{G}_{10}$.

V. Conclusion

Real-time digital simulators can play a key role in modern power grid projects by enabling rapid testing of new technologies and analyzing their impact on the system. They must however be capable of detailed modeling of power grid components with a sufficiently small time-step to capture dynamic events over a wide frequency bandwidth. This paper proposes the concept of hardware emulation building blocks to emulate in real-time power system components on FPGAs. The HEBB architecture for common power system elements such as linear passive elements, transmission lines, machines, nonlinear elements, and circuit breakers are presented to show how a realistic power system network can be modeled in detail on a FPGA. For emulating large-scale power systems, multiple FPGAs are employed for digital hardware emulation using the natural travel time delay of a transmission line to decompose and map the smaller subsystems to individual FPGAs for transient simulation. Real-time oscilloscope results are provided as a proof-of-concept to display both slow and fast dynamics in the system. The HEBB concept is modular and can be extended to emulate any number of multiphysical and multidomain components to integrate them into real-time power system simulation.

References

[1] D. Anderson, C. Zhao, C. Hauser, V. Venkatasubramanian, D. Bakken, and A. Bose, "A virtual smart grid," *IEEE Power Energy Mag.*, vol. 10, no. 1, pp. 49–57, Jan.–Feb. 2012.

[2] P. McLaren, O. Nayak, J. Langston, M. Steurer, M. Sloderbeck, R. Meeker, X. Lin, M. Yu, and P. Forsyth, "Testing the "smarts" in the smart T & D grid," in *Proc. IEEE Power Syst. Conf. Expo.*, Mar. 2011, pp. 1–8.

[3] J. Nutaro, "Designing power system simulators for the smart grid: Combining controls, communications, and electro-mechanical dynamics," in *Proc. IEEE PES Gen. Meeting*, Jul. 2011, pp. 1–5.

[4] X. Mamo, M. McGranaghan, R. Dugan, J. Maire, and O. Devaux, "A roadmap for developing real time distribution system simulation tools for the smart grid," in *Proc. SmartGrids for Distribution*, Jun. 2008, pp. 1–7.

[5] M. O. Faruque and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1146–1158, Apr. 2010.

[6] M. Brenna, E. De Berardinis, L. D. Carpini, P. Paulon, P. Petroni, G. Sapienza, G. Scrosati, and D. Zaninelli, "Real time simulation of smart grids for interface protection test and analysis," in *Proc. Int. Conf. Quality of Power*, Sep. 2010, pp. 1–6.

[7] P. Tatcho, Y. Zhou, H. Li, and L. Liu, "A real time digital test bed for a smart grid using RTDS," in *Proc. IEEE Int. Symp. Power Elect. for Distrib. Gen. Syst.*, Jun. 2010, pp. 658–661.

[8] R. Podmore and M. R. Robinson, "The role of simulators for smart grid development," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 205–212, Sep. 2010.

[9] K. Mets, T. Verschueren, C. Develder, T. L. Vandoorn, and L. Vandevelde, "Integrated simulation of power and communication networks for smart grid applications," in *Proc. IEEE Int. Workshop Comput. Aided Mod. and Design of Commun. Links and Networks*, Jun. 2011, pp. 61–65.

[10] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Inf.*, vol. 7, no. 2, pp. 224–243, May 2011.

[11] F. Salewski and S. Kowalewski, "Hardware/software design considerations for automotive embedded systems," *IEEE Trans. Ind. Inf.*, vol. 4, no. 3, pp. 156–163, Aug. 2008.

[12] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892–902, Apr. 2009.

[13] A. Myaing and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems With detailed representation of device characteristics," *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 358–368, Jan. 2011.

[14] A. Gomperts, A. Ukil, and F. Zurfluh, "Development and implementation of parameterized FPGA-based general purpose neural networks for online applications," *IEEE Trans. Ind. Inf.*, vol. 7, no. 1, pp. 78–89, Feb. 2011.

[15] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2547–2555, Jun. 2011.

[16] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1300–1309, Feb. 2012.

[17] S. Solano, M. Brox, E. del Toro, P. Brox, and I. Baturone, "Model-based design methodology for rapid development of fuzzy controllers on FPGAs," *IEEE Trans. Ind. Inf.*, vol. PP, no. 99, pp. 1–10, 2012.

[18] B. Alecsa, M. N. Cirstea, and A. Onea, "Simulink modeling and design of an efficient hardware-constrained FPGA-based PMSM speed controller," *IEEE Trans. Ind. Inf.*, vol. 8, no. 3, pp. 554–562, Aug. 2012.

[19] T. Atalik, M. Deniz, E. Koc, C. O. Gercek, B. Gultekin, M. Ermis, and I. Cadirci, "Multi-DSP and -FPGA-based fully digital control system for cascaded multilevel converters used in FACTS applications," *IEEE Trans. Ind. Inf.*, vol. 8, no. 3, pp. 511–527, Aug. 2012.

[20] Y. Chen, "Large-scale real-time electromagnetic transient simulation of power systems using hardware emulation on FPGAs," Ph.D. dissertation., Dept. Electr. Comput. Eng., Univ. of Alberta, Edmonton, AB, Canada, Mar. 2012.

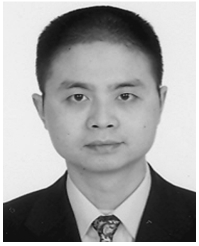
[21] *Stratix V Device Handbook*. : Altera Corp. Inc..

[22] H. W. Dommel, "Digital computer solution of electromagnetic transients in single and multiphase networks," *IEEE Trans. Power Apparatus and Systems*, vol. 88, no. 4, pp. 388–399, April 1969.

[23] H. K. Lauw and W. Scott Meyer, "Universal machine modeling for the representation of rotating electric machinery in an electromagnetic transients program," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 6, pp. 1342–1350, Jun. 1982.

[24] A. Morched, B. Gustavsen, and M. Tartibi, "A universal model for accurate calculation of electromagnetic transients on overhead lines and underground cables," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1032–1038, Jul. 1999.

[25] H. W. Dommel, "Nonlinear and time-varying elements in digital simulation of electromagnetic transients," *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 4, pp. 2561–2567, Jun. 1971.



Yuan Chen (M'12) received the B.Sc. and M.Sc. degrees in electrical engineering from Hunan University, Hunan, China, in 1992 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2012.

From 2000 to 2007, he was an Electrical Engineer in China. Currently, he is with RTDS Technologies Inc., Winnipeg, MB, Canada. His research interests include real-time simulation of power systems and field-programmable gate arrays, and high performance computation.

Venkata Dinavahi (SM'08) is a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronic systems, large-scale system simulation, and parallel and distributed computing.