Low-Latency Distance Protective Relay on FPGA

Yifan Wang, *Student Member, IEEE*, and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—The need for high-speed multi-function protective relays in both traditional transmission systems and the new emerging paradigm of the smart grid is growing. As a widely used protective scheme for transmission lines, a distance relay's high speed and reliable operation to clear faults is essential. This paper proposes a real-time low-latency hardware digital distance protective relay on the field programmable gate array (FPGA). Taking advantage of inherent hard-wired architecture of the FPGA, the proposed hardware distance relay design is paralleled and fully pipelined to achieve low latencies in various relay modules which are developed in textual VHDL language. This low-latency feature allows fast operating and data throughput so that the relay can handle high-frequency sampled data and reach higher computational efficiency. In addition, the parallelism and hardwired architecture of the FPGA makes the design more reliable in computation than the sequential software-based numeric relay. The FPGA-based distance relay can operate on both phasor-based signals and instantaneous signals with 2.09 μs and 0.35 μs latency respectively based on the clock frequency of 100 MHz. The hardware relay is tested in real-time by feeding it with generated faulted current and voltage data for typical faults and the relay response recorded. The results demonstrate the speed and effectiveness of the hardware distance relay.**

***Index Terms*—Digital hardware distance relay, field programmable gate arrays, parallel processing, power system protection, real-time systems.**

I. Introduction

PROTECTIVE relay technologies have been evolving over the last 40 years concomitantly with the developments in analog and digital computing technologies, and the requirements of power systems. But never before have they faced such unprecedented challenges as in the present times. The traditional notion of power systems is mutating into the smart grid concept with the widespread use of distributed generation and smart loads, which pose significant demands on power system protection and security [1]–[8]. Advanced concepts such as adaption, self-healing, wide-area monitoring, agent-based, transient operation, parameter recognition, and artificial intelligence need computationally powerful devices to be implemented in real-time applications [9]–[12]. There is a need for a high-capacity, high-bandwidth protective relay that can cope with the demand of signal processing, intelligence, and communication functions that such advanced concepts entail. In such applications, field programmability (FP) is a desirable feature to have in a protective relay. A FP device, whether analog or digital, is an integrated electronic circuit whose firmware can be reprogrammed in the field after manufacture. Currently, FPGAs are making significant inroads in many applications in industrial and commercial systems [13], and also in the real-time hardware emulation of power systems [14]–[17]. The characteristics of the FPGA that are germane for its use in protection relay application are:

- inherent parallel hard-wired architecture allowing an ultra-low latency realization of complex algorithms;
- very large capacity devices comprised of millions of logic building blocks to provide substantial hardware resources for even the most resource intensive models and algorithms;
- mature and advanced design and development tools for rapid prototyping, and tight integration with mathematical software packages such as Matlab/Simulink, allowing users the choices of written textual (VHDL or Verilog), or schematic design entry methods;
- fast clock speeds and high-speed transceivers to communicate with external devices.

In the industry, protective relays have experienced mainly three generations of evolution [18], [19]: electromechanical (EM) relays, solid-state (SS) relays, and digital relays. EM relays were based on moving parts to perceive abnormal changes of current or voltages to generate the mechanical torque. SS relays based on analog electronic devices such as transistors, diodes, and other electronic components, were the static replacements of EM relays. Currently, most commercial relays are digital numeric relays based on microprocessor technology, and sequential software programmability. The sequentiality of software architecture inherently limits the operating speed of the designs. Recently new types of protective relay hardware are reported in the literature. In [20] a distance protective relay implementation in a field-programmable analog array (FPAA) is presented. This approach using a combination of FPAA and digital technologies has the potential to provide better performance than the present-day numerical relays; however, the use of analog blocks in the relay may have limitations on device size, switching noise, and sample-rate related bandwidth problems compared with digital technology. In the literature, few works have been reported with hardware design of the protective relay on FPGA. Reference [21] proposed a hybrid protection scheme for HVDC line implementation, and [22] implemented the wavelet-based directional relay for transformer protection. In these works, the implementation is carried out using vendor-specific schematic blocks, which preclude a generalized and portable floating-point design. Furthermore, the hardware design details are omitted from their description.

Manuscript received January 23, 2013; revised May 08, 2013 and July 01, 2013; accepted August 13, 2013. This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC). Paper no. TSG-00049-2013.

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: yifan17@ualberta.ca; 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/TSG.2013.2278697

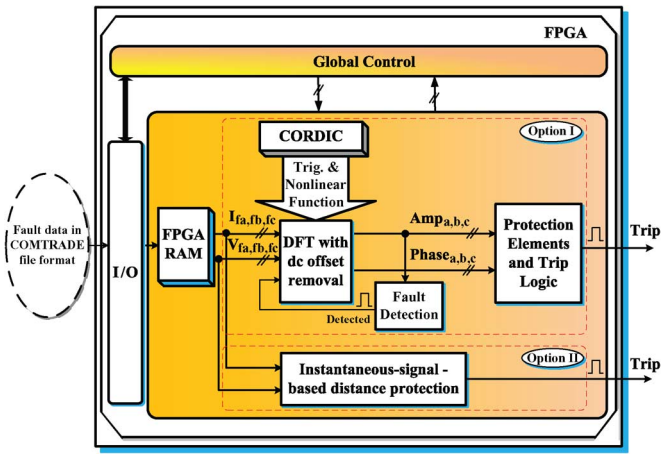


Fig. 1. Overall architecture of the FPGA-based hardware distance relay: functional block diagram.

In this work, an FPGA-based low-latency high-resolution distance protective digital relay with both phasor-based and instantaneous-signal-based processing is designed. The hardware design consumes a small percentage of the Xilinx Virtex-7 FPGA resources, and it can process data within a few microseconds. It uses the inherent parallelism of the FPGA hardware to increase execution speed as well as provide higher design reliability. The organization of this paper is as follows. The operation philosophy and design details of the hardware relay modules on the FPGA are provided in Section II. Section III presents the case studies for testing the designed hardware distance relay, and experimental results are shown to verify the relay operation. Finally, Section IV presents the conclusion of the paper.

II. Distance Protective Relay Hardware Architecture and Parallel Modules

The overall architecture of the FPGA-based hardware distance relay is shown in Fig. 1. It provides two options of operating modes. Option I is DFT-based distance relay for phasor signal processing which consists of four main hardware modules: the discrete Fourier transform (DFT) processing module, the COordinated Rotation Digital Computer (CORDIC) processing module, the fault detection module, and the protection elements module. In this mode, by taking in fault voltage and current data through the I/O interface at first, the DFT module estimates fundamental amplitude and phase of the signals. The CORDIC processing provides trigonometric and nonlinear function values that are needed for arithmetic computations. The fault detection module utilizes the DFT results of the currents to detect the inception of a fault based on an over-current protection mechanism, while the protection element module calculates the fault impedance and decides the trip logic. If a fault is detected, and the calculated impedance falls into the protection zone, a trip signal for the associated circuit breaker is sent to isolate the fault. In addition, a global control module is designed to control all the operations in the whole design. The Option II is the instantaneous-signal-based distance relay module which can process instantaneous signals to achieve a much faster response. In the following subsections, each module within the proposed relay is discussed and its hardware design details are presented.

A. DFT Module With DC Offset Removal

The widely-used digital distance relay relies on the fundamental phasor estimation of the voltage and current signals at the relay location that can derive the fault impedance. Thus, the very first step is to obtain the signal fundamental frequency components. Among the existing filtering algorithms [23]–[28] for the fundamental extraction in digital distance relays, the DFT which has improved harmonic immunity is one of the most popular methods to obtain the quantities of interest. However, dc offset transients contained in fault signals can affect accurate estimation of phasors through DFT. As a result, transmission line relays tend to maloperate by overreaching or underreaching the setting value. Therefore, the dc offset component has to be removed. Here the algorithm put forward by [26] is adopted as it only requires one cycle plus two samples to finish full-cycle DFT (FCDFT) calculation with dc offset removal. The effect of the additional two calculations due to the two samples can be offset by the low-latency of the FPGA hardware module.

The conventional FCDFT algorithm calculates the fundamental component of a sinusoidal discrete time signal can be described as:

$$X_{(1)} = \frac{2}{N}\sum_{n=0}^{N-1} x(n) \times (\cos\omega_1 n\Delta T - j\sin\omega_1 n\Delta T) = \frac{2}{N}\sum_{n=0}^{N-1} x(n) \times \left(\cos\frac{2\pi n}{N} - j\sin\frac{2\pi n}{N}\right), \quad (1)$$

$$A_1 = \sqrt{X_{1real}^2 + X_{1imag}^2} \quad (2)$$

$$\theta_1 = \arctan\left(\frac{X_{1imag}}{X_{1real}}\right) \quad (3)$$

where $x(n)$ is the discrete-time sinusoidal input signal, $\mathbf{N}$ is the number of samples in a fundamental period, with ω_1 being the fundamental angular frequency, and ΔT the sampling interval. Then, with real and imaginary parts of the fundamental phasor known, the amplitude A_1 and phase θ_1 of the phasor could be worked out as in (2) and (3). For dc offset removal strategy in, the input signal contains a decaying dc offset $Ae^{-t/\tau}$ (time constant $\tau = q\Delta T$). After computing $X_{real(N)}$, $X_{real(N+1)}$, $X_{real(N+2)}$ which are real parts of three FCDFT fundamental phasor calculation results, the parameters of the dc offset are obtained as:

$$e^{-1/q} = \frac{\left(X_{real(N+2)} - X_{real(N+1)}\right) cos\left(\frac{2\pi}{N}\right)}{\left(X_{real(N+1)} - X_{real(N)}\right) cos\left(\frac{4\pi}{N}\right)}, \quad (4)$$

$$A = \frac{0.5N\left(X_{real(N+1)} - X_{real(N)}\right)}{cos\left(\frac{2\pi}{N}\right) e^{-1/q}(e^{-N/q} - 1)}. \quad (5)$$

The dc offset can be then subtracted from the sampled signal $x(k)$ as:

$$z(k) = x(k) - Ae^{-k/q}. \quad (6)$$

Fig. 2 shows the FPGA hardware design of DFT module with dc offset removal. The DFT calculation module gets the *set_offset* control signal from the finite state machine controller which has four transition states going from $S1$ to $S4$. Under the control strategy, the controller first senses the fault *detected*

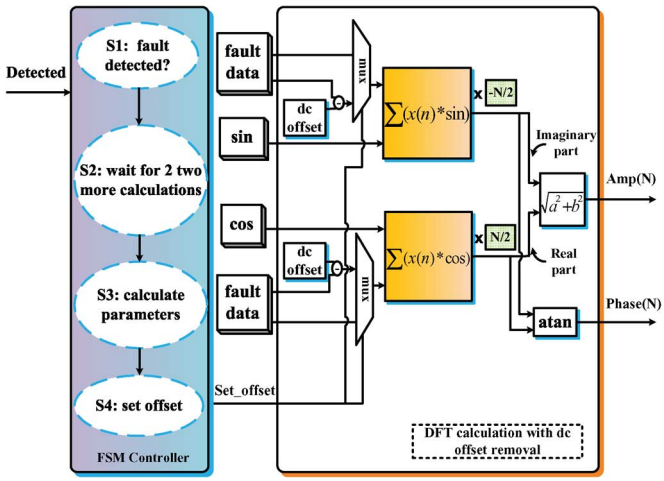


Fig. 2. Pipelined floating-point DFT module with dc offset removal.

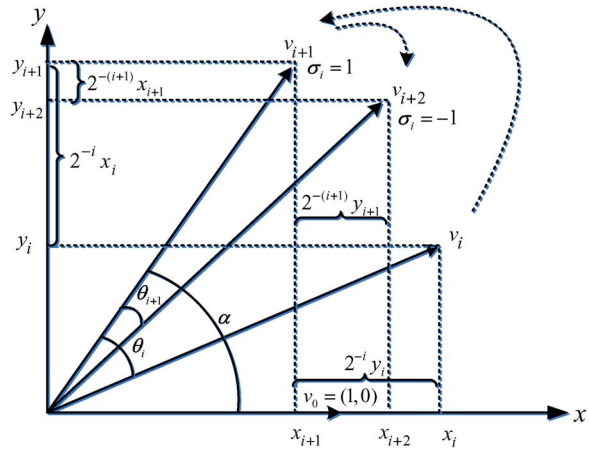


Fig. 3. CORDIC rotation mode for the circular coordinate system.

signal, and waits for two more sample calculation cycles to finish dc offset parameters calculation. Once the calculation is done, a *set_offset* signal is sent out to the DFT calculation module. In each DFT computation cycle, with collection of one full-cycle window of fault data, sin and cos values, imaginary and real part of the fundamental phasor are calculated. After taking square root of the sum of imaginary and real part squares, and taking the *arctangent* of imaginary part over real part, the amplitude and phase of the set of data are generated. After this calculation is completed, the data window moves forward by one sampling point to do the next computation cycle. Within the module, all the computation elements use IEEE single-precision floating-point 32-bit format for higher precision and are fully pipelined to enhance throughput.

B. CORDIC-Based Trigonometric and Nonlinear Function Evaluation Module

The trigonometric and nonlinear function values required by the DFT module are generated from the iterative CORDIC module. The CORDIC method [29]–[31] has the advantage of higher precision over the conventional look-up-table (LUT) based method, because usually the desired trigonometric or nonlinear function values are stored in LUTs whose length and precision are limited by the volume of ROM limits.

The rotation mode and vectoring mode are generally the two ways to implement the CORDIC algorithm, which result in computations of different trigonometric or hyperbolic functions. The general form of CORDIC iteration is as follows:

$$\begin{aligned} x_{i+1} &= x_i - m \cdot \sigma_i \cdot y_i \cdot 2^{-i} \\ y_{i+1} &= y_i + \sigma_i \cdot x_i \cdot 2^{-i} \\ z_{i+1} &= z_i - \sigma_i \cdot \alpha_{m,i} \\ i &= i + 1. \end{aligned} \tag{7}$$

In each rotation, the plane vector $v_i = (x_i, y_i)^T$ rotates to $v_{i+1} = (x_{i+1}, y_{i+1})^T$. z_i tracks the angle at each rotation. For example, in the rotation mode shown in Fig. 3, when estimating the trigonometric values of the given angle that is initialized at first, z_i rotates toward 0 with a micro rotation angle of θ_i which has the tangent value of 2^{-i} (or hyperbolic tangent value of 2^{-i} in hyperbolic coordinates) in each iteration. In the end, when the sum of rotated angles reaches the input angle, the x and y coordinates indicate the sin and cos values of the input angle. On the other hand, in the vectoring mode the coordinates of a vector are given a priori while the magnitude and angular argument of the original vector are computed. Therefore, after n iterations, CORDIC module can obtain the trigonometric and hyperbolic functions based on the mode it is operating on. The parameter descriptions of (7) and the CORDIC mode operation summary are given in Table I.

TABLE I
CORDIC ITERATION PARAMETERS AND FUNCTION DESCRIPTION

	Circular coordinate ($m = 1$)	Hyperbolic coordinate ($m = -1$)
Parameters	$\alpha_{1,i} = tan^{-1}(2^{-i})$	$\alpha_{-1,i} = tanh^{-1}(2^{-i})$
Rotation: $z_i \to 0$ $\sigma_i = sign(z_i)$	$sin(x)$, $cos(x)$	$tan^{-1}(x)$
Vectoring: $y_i \to 0$ $\sigma_i = -sign(x_i y_i)$	$sinh(x)$, $cosh(x)$	$tanh^{-1}(x)$

With the computation results of the elementary functions through CORDIC, the nonlinear functions in (6) can be calculated as (9). Taking $e^{-1/q}$ in (6) as variable m, and k as variable n, the nonlinear function m^n is decomposed as:

$$m^n = e^{n*ln(m)}. \tag{8}$$

Note that the exponential function e^x and natural logarithm function $ln(x)$ could be obtained from:

$$\begin{aligned} e^x &= sinh(x) + cosh(x), \\ ln(x) &= 2tanh^{-1}\left|\frac{x-1}{x+1}\right|. \end{aligned} \tag{9}$$

The design of the CORDIC algorithm on FPGA has a pipelined architecture to improve speed and throughput. Fig. 4 shows how each CORDIC iteration is implemented in hardware. Arctangent and hyperbolic arctangent values of 2^{-i} are pre-calculated and stored in memory. Then, in each iteration, there are only addition/subtraction and shifting operations (multiplication by 2^{-i} can be done by bit shifting to the right), which can be easily achieved in hardware. The iterative stage n can be set manually. In this design, it is set to 20. Besides,

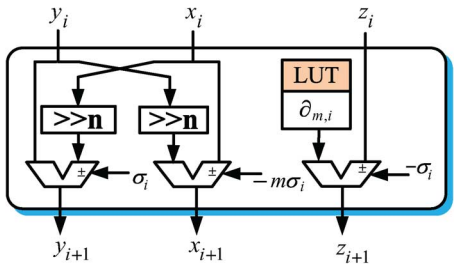


Fig. 4. FPGA design of CORDIC module in one iteration.

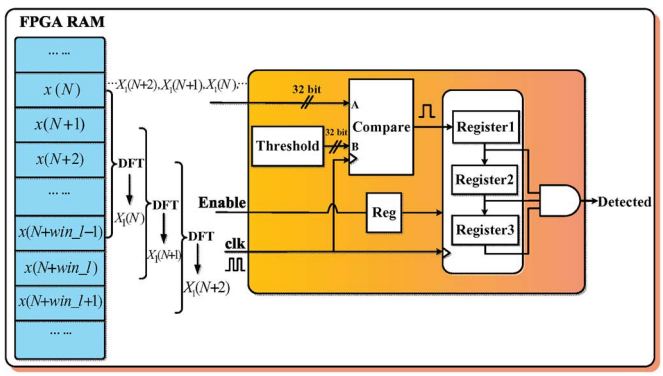


Fig. 5. Fault detection hardware module.

TABLE II
IMPEDANCE EQUATIONS BASED ON DIFFERENT FAULT TYPES

Relay elements	Impedance formula
a-g	$V_A/(I_A + 3k_0 * I_0)$
b-g	$V_B/(I_B + 3k_0 * I_0)$
c-g	$V_C/(I_C + 3k_0 * I_0)$
a-b	$(V_A - V_B)/(I_A - I_B)$
b-c	$(V_B - V_C)/(I_B - I_C)$
c-a	$(V_C - V_A)/(I_C - I_A)$

it is worthwhile to mention that since the iterative CORDIC algorithm is based on fixed-point data, all the incoming data to the module will be converted to fixed-point numbers and computation results to floating-point numbers by a float-to-fix module and a fix-to-float module respectively in order to interface with other floating-point data based modules.

C. Fault Detection

The fault detection module detects the initiation of the fault and trigger the different zone timers. It is based on the over-current starting method. This detection algorithm uses the traditional FCDFT filtering results of fundamental amplitude to decide the detected signal. Although the algorithm suffers from some drawbacks such as its sensitivity to even harmonics and decaying dc components in the fault signals, the presence of harmonics does not significantly affect the fault detection decision [32]. The strategy is illustrated as: if the fundamental amplitude of any phase current exceeds a certain threshold value (usually set according to maximum load current in the phases) three times consecutively, this shows that the transmission line is exposed to an abnormal situation, and a fault detection signal is sent out.

The hardware implementation details of the fault detector are shown in Fig. 5. Previous DFT amplitude results enter into the module, and are compared with the threshold value. The signal *enable* comes from DFT module output indicating validation of each DFT computation result. It enables the comparison and registering process. The register inserted after the *enable* signal is for synchronization purpose. If three successive amplitudes exceed the threshold, the three single-bit registers which record the comparison results will become "1." Then a fault detection signal would be generated.

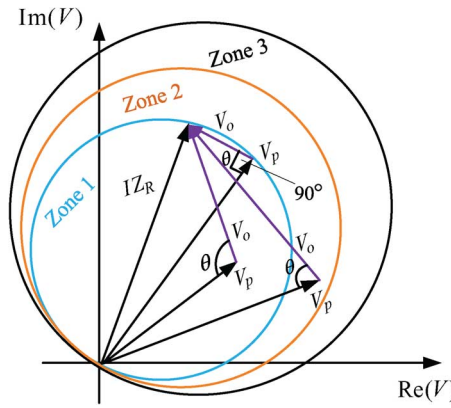


Fig. 6. Mho characteristic with three zone protection.

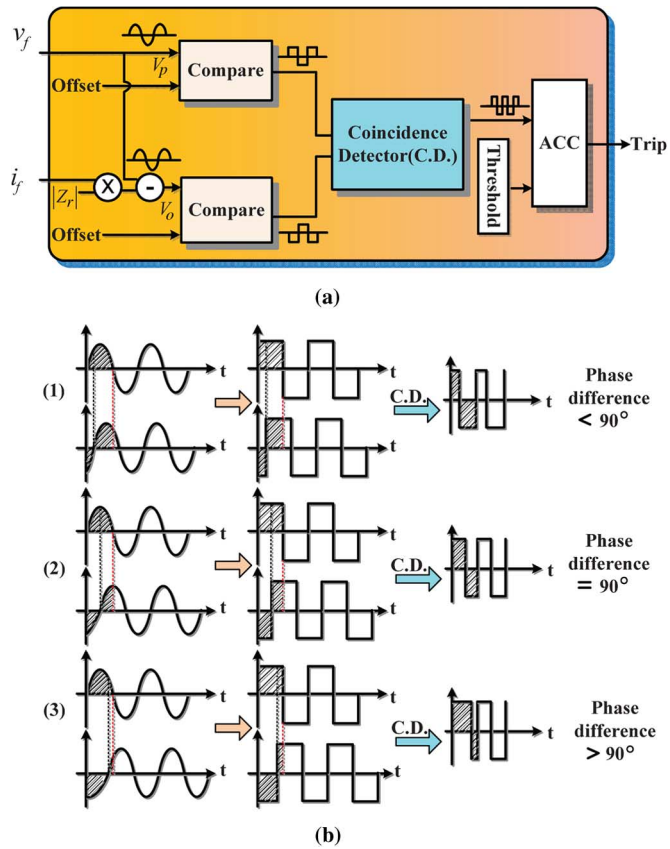


Fig. 7. Instantaneous-signal-based distance relay: (a) Functional block diagram of the hardware design. (b) Principle of operation.

D. Distance Protection Elements

The FPGA-based hardware digital relay was designed to protect a three-phase transmission line with six impedance measuring elements: three phase-to-ground relays and three

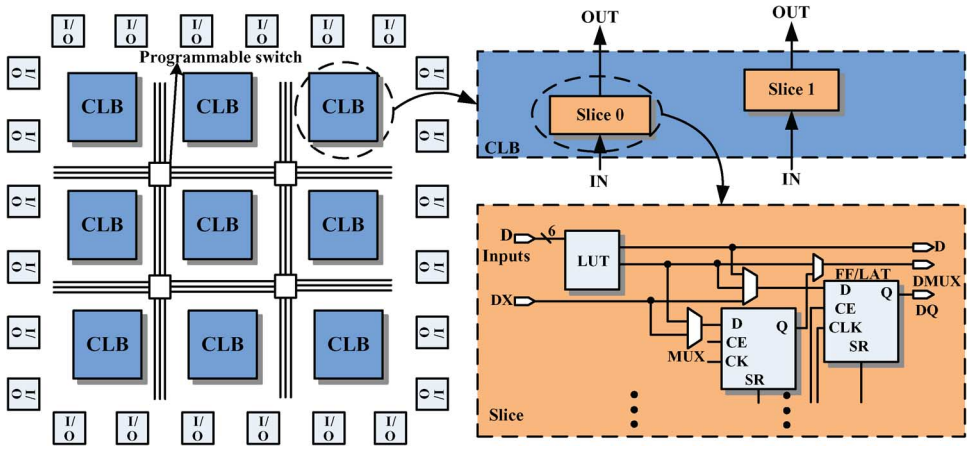


Fig. 8. FPGA hardware architecture, and configurable logic blocks.

phase-to-phase relays. Table II presents apparent impedance calculations for different fault types.

The impedance calculated from these six relay elements is then processed by a mho characteristic [19] relay element that is based on phasor comparison to decide which protection region it is in. Fig. 6 shows the mho relay characteristic with three protection zones.

Applying different zone protection can be achieved by setting different diameter of the circles, and coordination between different zones by setting different corresponding time delays. The module calculates the angle between polarizing vector V_p and operating vector $V_o = IZ_R - V_p$, where V_p equals to fault voltage acting as reference, I is fault current and Z_R is the relay setting. As seen from Fig. 6, if the angle between the two vectors is greater than or equal to 90°, the fault impedance locates within or on the characteristic circle meaning the measured impedance is under reaching the zone. Then the relay will trip the corresponding circuit breaker of the transmission line. The FPGA implementation of mho elements can be achieved by applying complex number computation and angle comparison.

E. Instantaneous-Signal-based Distance Relay Element

To provide a faster tripping, this work also gives another option of distance protection that is based on instantaneous signal processing. The basic trip strategy of this option is the same as the mho phase comparator introduced in the previous subsection (D), however, unlike the DFT distance relay option relying on the fundamental phasor extraction illustrated before, the instantaneous-signal-based algorithm processes instantaneous voltages and current signals [19], [33]. As a result it can shorten the operating time to less than half a fundamental cycle while the DFT method requires collecting one full cycle of data to calculate the phasors and then perform further protection logic. In this module, the input waveforms are first filtered to remove any high-frequency components that could influence the accurate decision making of the relay.

The FPGA hardware implementation of the instantaneous-signal-based method is shown in Fig. 7(a). The filtered fault signal goes into the module and is compared with an offset value (here set as "0.") to produce a square wave corresponding to the input sinusoidal value. The square wave carry the phase information of the sinusoidal wave. Then, the square waves of the two inputs (the polarizing vector V_p and operating vector V_o) are compared with each other using a coincidence detector. A "-1" is produced during the times both square waves agree in polarity while a "$+1$" is produced during times the two square waves have opposite polarity; thus a coincidence detection is achieved resulting in a double frequency square wave. If the two input waves have a phase difference of $\pm 90°$, the double frequency square wave has equal positive and negative pulses as in Fig. 7(b) (2). In other cases, for instance a fault, the coincidence is biased that can be clearly seen in Fig. 7(b) (1 or 3). The ACC block takes in the coincidence results and accumulates the biased part points in the double frequency waveform. Once the accumulation number exceeds a set threshold, a trip output signal is sent to isolate the fault.

III. Test Case and Experimental Results

A. Test Setup and Hardware Utilization

The full hardware distance relay design was targeted to the Xilinx Virtex-7 XC7VX485T FPGA, using Xilinx ISE14.1 tools to synthesize and implement the architecture. The FPGA is generally composed of three types of configurable logic components: configurable logic block (CLB), configurable input/output block (IOB), and programmable interconnections as shown in Fig. 8. All the user-defined functional elements are achieved by CLB which are connected to a programmable switch matrix. In the platform (Xilinx Virtex-7 FPGA), one CLB contains a pair of logic slices that are composed of 6-input LUTs and storage elements. The FPGA has the following main features: 1955 K logic cells, 68 Mb block RAM, 2800 DSP48E1, and 1200 I/O pins [34].

The entire hardware resource consumption of the distance relay design is given in Table III. In order to observe output waveforms, a 16-bit 4-channel DAC board is connected to the FPGA board with an FMC-DAC-Adapter. A500 MHz Tektronix DPO7054 4-channel oscilloscope is used to capture the output of the DACs. Fig. 9 depicts the experimental test setup.

To test the effectiveness of proposed distance relay hardware design, several typical faults were simulated on a test power system using PSCAD/EMTDC to generate the fault data. These data are then fed into the target hardware distance relay for test

TABLE III
FPGA HARDWARE RESOURCE USAGE

Modules	Slice Registers (607,200 available)	Slice LUTs (303,600 available)	DSP blocks (2800 available)
DFT with dc offset removal	41096 (6%)	20446 (6%)	24 (1%)
CORDIC	6429 (1.05%)	4946 (1%)	2 (0.07%)
Fault detection	180 (0.03%)	427 (0.1%)	0
Mho relay element	5892 (1%)	322 (0.1%)	9 (0.3%)
Instantaneous process element	7992 (1%)	1843 (0.6%)	15 (0.5%)
Total usage	61589 (9%)	27984 (8%)	50 (2%)

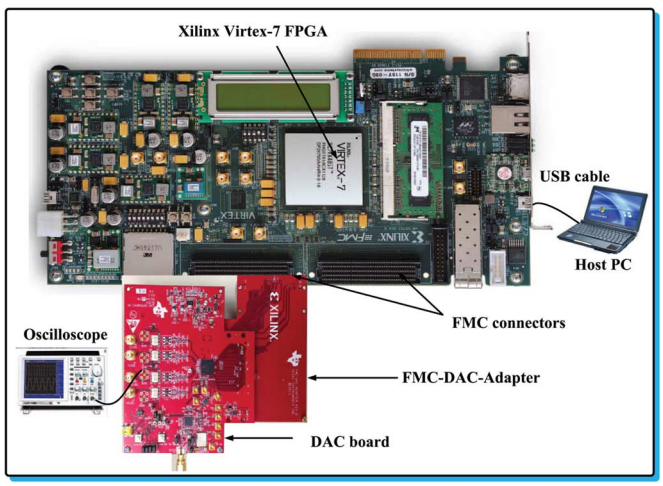


Fig. 9. Experimental setup.

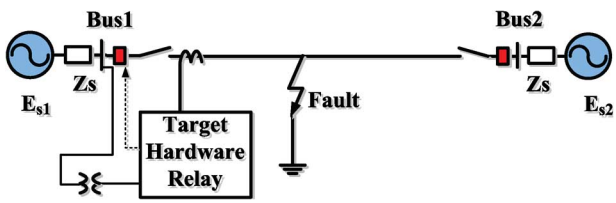


Fig. 10. Single line diagram of test power system.

and validation. The tested power system consists of two synchronous generators and a 210 km long transmission line is shown in Fig. 10 with parameters given in the Appendix.

The whole design which includes the two operational options has a total latency of 2.09 μs and 0.35 μs respectively based on an FPGA clock frequency of 100MHz. According to the breakdown of each module's latency shown in Fig. 11, for the phasor-based option, the DFT module consumes the largest latency of 83 clock cycles, while the fault detection module has the smallest latency of 3 clock cycles. In addition, the CORDIC module, the dc offset parameter calculation and the mho relay element utilize 30, 51, and 42 clock cycles respectively. For instantaneous-signal-based option, the overall latency is greatly reduced to 35 clock cycles due to the elimination of fundamental phasor calculations. The low latencies of the design can be attributed to the hardware parallelism and deep pipelining employed in various hardware modules.

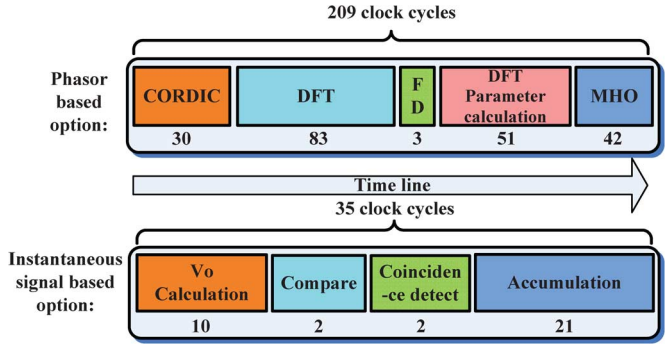


Fig. 11. Hardware module latency breakdown.

B. Results and Discussion

1) Hardware Experimental Results: The designed hardware distance relay was tested under four types of fault conditions. The first fault event is a single-phase-to-ground fault which occurs on the transmission line 90 km away (42.8% of the line length) from the relay location. The oscilloscope traces real-time simulation results of fault data transients at the relay location given in Fig. 12 from t $= 0.05$ s to t $= 0.5$ s. The sustained fault in phase-a is initiated at time $t_1 = 0.2$ s, when the consequent decrease in v_a, and increase in i_a, and its dc offset can be observed. In the DFT processing results of the fault phase current, the amplitude of Segment I stays zero until one full cycle of fault data collection is finished. During Segment II, the i_a amplitude value stays constant under normal operating condition until at t $= 0.2$ s when a ground fault occurred. Due to the inception of the fault, according to the DFT calculation, the i_a amplitude starts to increase after a small transient oscillation lasting about 0.005 s. The fault detection module senses the abnormal current increase and operates to send the fault detected signal at $t = 0.207$ s. The smooth Segment III indicates the effectiveness of the dc offset removal of fault current in the faulted phase. As for the final trip signals, the two available options which are phasor-based and instantaneous-signal-based give two trip signals at $t = 0.225$ s and $t = 0.208$ s respectively.

Fig. 14 shows the real-time trajectory of the apparent impedance seen by the relay during the phase-to-ground fault. The real-time mho circle was obtained by coupling the time-based x and y coordinates for the relay impedance setting, while the real-time impedance trajectory was derived by coupling the calculated time-varying R and X signals displayed on the oscilloscope. The mho zone reach is 80% of the line length, which is $5.998 + j85.31\ \Omega$. Before the fault at the initial state $(-624.8 + j2.7\ \Omega)$, the impedance is outside the mho characteristic circle on the remote left side. After the fault, the impedance locus moves to the right and enters into the circle. Finally, the impedance trajectory converged to its final state of $9.227 + j44.38\ \Omega$, which is 53% of the zone reach and 42.4% of the line length. After the fault detection, the relay calculates the phasor comparison criteria and trips since the operating point is within the trip zone.

The next fault condition presented is a $b - c$ phase-to-phase fault shown in Fig. 13. The waveforms follow the same description pattern as in Fig. 12. The fault currents of phase-b and phase-c increase during fault and then stay at the constant values

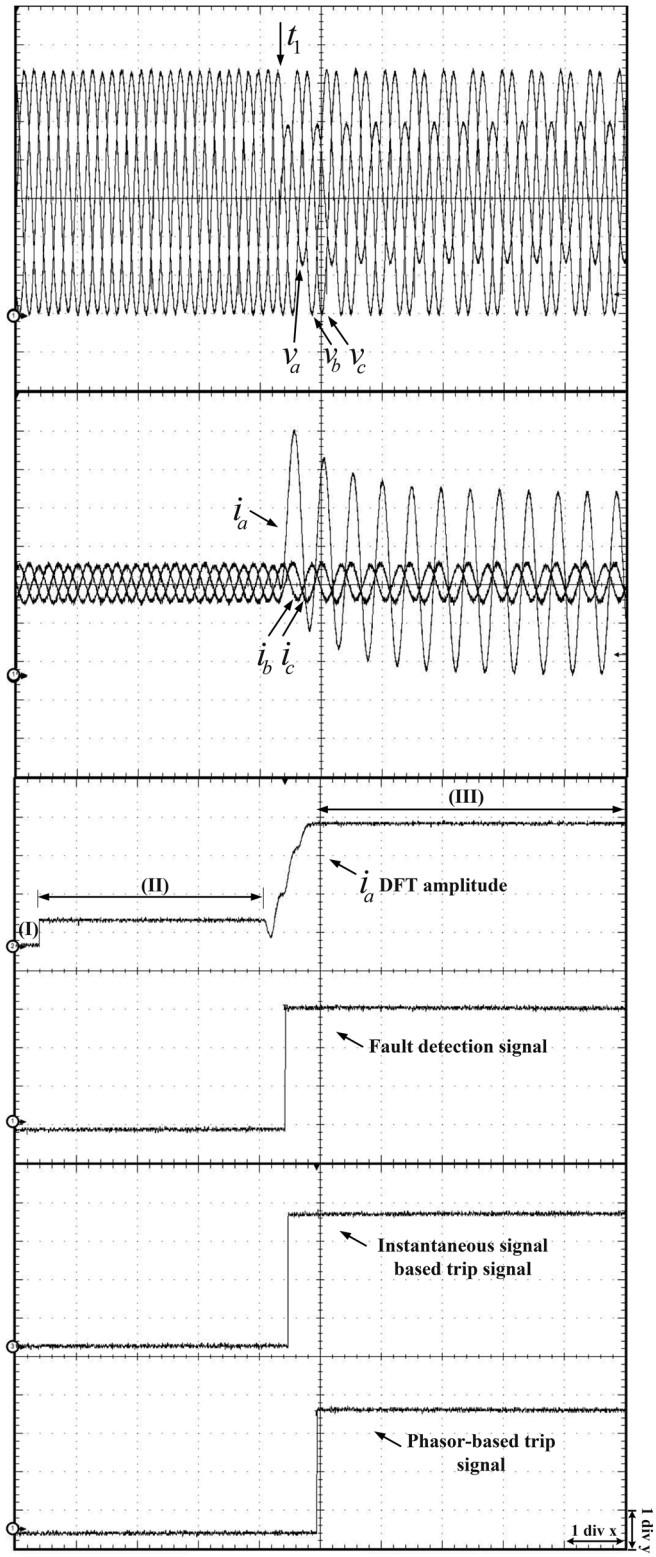


Fig. 12. Real-time fault data waveforms, DFT and trip signals for a Phase A to ground fault. Scale [time: 1 div $x = 45$ ms; voltage: 1 div $v = 0.31$ p.u.; current: 1 div $i = 1.7$ p.u.].

of 5.72 p.u. and 5.7 p.u respectively. A much faster trip offered by instantaneous-signal-based methods can also be observed in Fig. 13. The locus of the apparent b-c phase impedance is given in Fig. 15. The mho zone setting is 80% of the whole line length as well. The apparent impedance moves from the initial state $(-630.3 + j4.24\ \Omega)$ on the left side to the top of the mho circle. The converged impedance point is $3.236 + j46.70\ \Omega$, which is around the 43.8% of the line impedance and 54.7% of the zone reach.

The trip time results of a double-phase-to-ground fault and a three-phase-to-ground fault are given in Table IV. According to these results, the proposed hardware digital distance relay has achieved its fundamental objectives. While an exact comparison of the FPGA-based distance relay with existing numerical relays might be somewhat misleading since the implemented algorithms and the design logic might not be exactly the same, to put this work into context, a few examples from existing relays in the industry are provided. ABB's REL 650 line distance protection relay has the distance measuring typical operate time of 30 ms, and 24 ms for the REL 670 platform. In SEL's 421 protection, automation, and control system, the high-speed elements operate around 0.5 fundamental cycle, while the standard-speed elements longest operating times is close to 1.5 fundamental cycles. These operating times are similar to the ones shown in Table V and Table VI for the designed hardware distance relay. This is because, using FCDFT, one can not initiate the Fourier algorithm until a full cycle of data has been collected. Furthermore, it is worth mentioning that the initial chosen sampling rate for a full-cycle for this hardware design is 68. However, the overall low-latency of the hardware design leads to high-speed computation on the FPGA, and thus higher sampling rates can be chosen (more recent numerical relays use sampling rates that are as high as 96 samples per period [35]). In the proposed hardware design, when the sampling rate is 334 (50 μs time interval) per cycle, results in Table V for the DFT-based and Table VI for the instantaneous-signal-based options show that the fault detection time or trip time could be shortened by over 10%.

2) Advantages of FPGA-Based Hardware Relay Emulation: In addition to the above merit, based on our experience with the designed hardware relay, we found the following advantages related to design reliability, maintainability, power consumption, and flexibility that make FPGAs suitable for protective relay applications.

- Since the FPGA is composed of configurable logic components, all the user-defined functional elements can be achieved by the hard-wired architecture. Each task is allocated its own hardware resources, such as the parallel distributed memory throughout the device, and runs independently with less interference between two functional blocks. However, in a numerical relay with a DSP software design, the executing algorithm relies on transferring information to and from memory, and the sequential process scheme. As the DSP is designed to be constantly operated for many different tasks and its resources are shared by all tasks, it creates lots of opportunities for the tasks to interact in unexpected ways. Consequently, if a small mistakes occurs, there is high probability that the whole program might break down. Therefore, the FPGA-based hardware design has a higher reliability than a software program running on a DSP.
- The VHDL code for the designed hardware relay on FPGA has a modular structure and was developed in a very clean

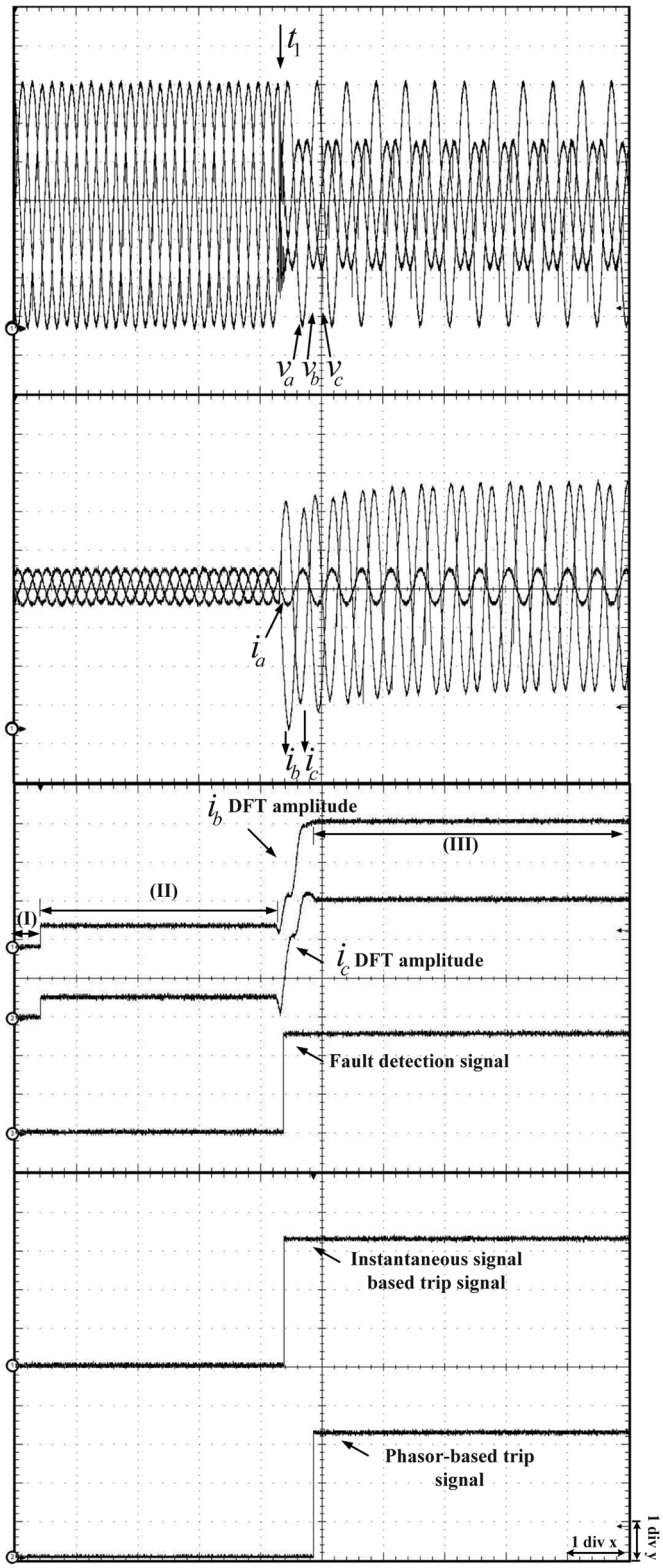


Fig. 13. Real-time fault data waveforms, DFT and trip signals for a $b-c$ phase-to-phase fault. Scale [time: 1 div $x = 45$ ms; 1 div $v = 0.31$ p.u.; current: 1 div $i = 1.7$ p.u.].

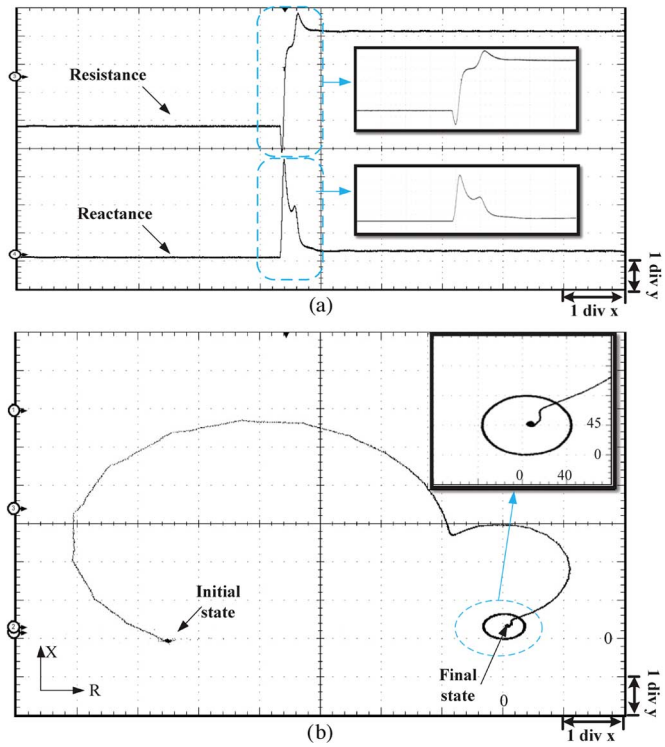


Fig. 14. Real-time impedance trajectory during phase-a-to-ground fault: (a) Real-time resistance (R) and reactance (X) values with respect to time, [time: 1 div $x = 45$ ms; 1 div $y = 190\ \Omega$]; (b) x-y mode trace of impedance, [1 div $x = 120\ \Omega$; 1 div $y = 140\ \Omega$].

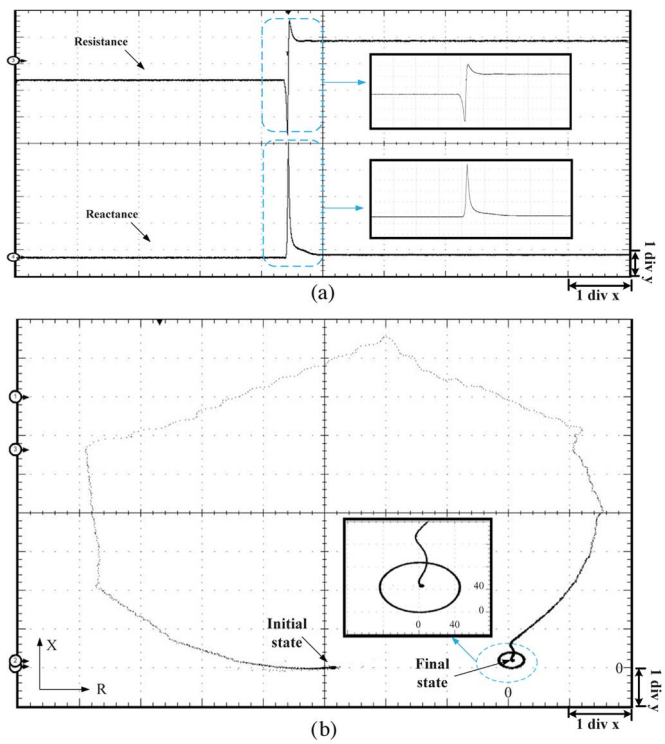


Fig. 15. Real-time impedance trajectory during b-c phase-to-phase fault: (a) Real-time resistance (R) and reactance (X) values with respect to time, [time: 1 div $x = 45$ ms; 1 div $y = 400\ \Omega$], (b) x-y mode trace of impedance, [1 div $x = 220\ \Omega$; 1 div $y = 240\ \Omega$].

level hierarchy. The VHDL modules were realized in hardware circuits distributed on the chip and connected with each other by input and output ports. The connections between the various modules are hardwired. Testing the entire design involved testing every functional circuit and then testing the top-level design connected together. This makes the VHDL program very easy to troubleshoot. However, in a very long C code (typically thousands of lines of

TABLE IV
TRIP TIMES FOR DIFFERENT TYPES OF FAULTS

Faults	Phasor-based (ms)	Instantaneous-signal-based (ms)
a-g	25.12	8.46
b-c	22.89	6.22
b-c-g	22.39	6.21
a-b-c-g	22.14	5.47

TABLE V
FAULT DETECTION TIMES COMPARISON BASED ON DIFFERENT SAMPLING RATES, (DFT-BASED)

Faults	Detection time (ms) Lower sample rate (68/cycle)	Detection time (ms) Higher sample rate (334/cycle)	Reduction (ms)
a-g	7.35	6.80	0.55
b-c	5.15	4.49	0.66
b-c-g	5.39	4.89	0.50
a-b-c-g	4.42	3.84	0.58

TABLE VI
FAULT TRIP TIMES COMPARISON BASED ON DIFFERENT SAMPLING RATES, (INSTANTANEOUS-SIGNAL-BASED)

Faults	Trip time (ms) Lower sample rate (68/cycle)	Trip time (ms) Higher sample rate (334/cycle)	Reduction (ms)
a-g	8.46	6.58	1.88
b-c	6.22	5.59	0.63
b-c-g	6.21	5.79	0.42
a-b-c-g	5.47	4.69	0.78

code) on the DSP, it is very difficult to maintain and troubleshoot the code when a fault occurs since there are subparts connected together, and functions calling and being called every now and then in a sequential way. High-level synthesis tools exist to automatically convert C code to VHDL with minimum user intervention, thereby enabling novice designers to emulate hardware functions on FPGAs.

- The designed FPGA-based distance relay has a very low hardware latency while operating at a much lower clock frequency (100 MHz for FPGA versus 1 GHz for DSP) and consumes very low hardware resources (9%). Thus, a much higher sample rate can be used for the FPGA-based relay which leads to better reproduction of the original signals and better transient analysis. Furthermore, new protection algorithms based on transient signals can be developed on the FPGA. Even if the DSP can perform the high resolution computation within a few hundreds of nanoseconds (which is unlikely based on the specifications of commonly available DSPs), it will occupy large chip resources and heavy power consumption which can cause the device to overheat. But in our design, with very small percentage of hardware usage, the power consumption is very low so that the life-time of the device is increased and maintenance cost is reduced.
- The Xilinx Virtex-7 device has abundant hardware resources and peripheral devices. High-speed transceivers are available to communicate with external devices for future expansion of the hardware relay functionality.
- The partial reconfiguration (PR) feature is available on the FPGA. It can modify an operating FPGA design by loading a partial configuration bit file, which provides the flexibility of on-site re-programming. The partial bit files can be downloaded to modify reconfigurable regions in the FPGA without affecting other static logic part on the board. This feature provides us the flexibility in the choices of algorithms or protocols available to an application.

IV. CONCLUSION

This paper presents the digital hardware distance relay on the FPGA. All the relay submodules were developed in VHDL which can be easily transplanted to different development environments. Taking advantages of inherent parallel architecture of FPGA, the proposed hardware is paralleled and fully pipelined to achieve high operating efficiency and speed. The case studies show the operational effectiveness of the designed distance relay with a low hardware latency. With this low latency, a much higher sampling rate can be achieved and the fault detection time would be shortened. Furthermore, depending on the size of the system and the time-step required for a certain transient, more submodules can be replicated or added to the same FPGA to make it a relay cluster that can handle several different protection scenarios. With plenty of hardware resources available on the FPGA board, this approach can also be updated to a multi-function hardware relay with smart metering capabilities once more functional elements are added in the future.

APPENDIX

Test Power System Parameters

$V_{base} = 230$ kV, $S_{base} = 100$ MVA, fault impedance $= 0.01\ \Omega$.

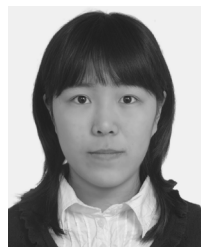
Source parameters: $Z_s = 9.2 + j52\ \Omega$, $E_{s1} = 230\angle 0^\circ$ kV, $E_{s2} = 230\angle 20^\circ$ kV.

Transmission line sequence impedance (Ω/km): $Z_0 = 0.363 + j1.326$, $Z_1 = 0.0357 + j0.5078$, $Z_2 = 0.0357 + j0.5078$.

REFERENCES

[1] S. M. Amin, "Toward more secure, stronger and smarter electric power grids," in *Proc. IEEE PES Gen. Meet.*, Jul. 2011, pp. 1–4.
[2] 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.
[3] 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 *IEEE Power Syst. Conf. Expo. (PSCE)*, Mar. 2011, pp. 1–8.
[4] 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.
[5] H. Qi, X. Wang, L. M. Tolbert, F. Li, F. Z. Peng, P. Ning, and M. Amin, "A resilient real-time system design for a secure and reconfigurable power grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 770–781, Dec. 2011.
[6] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, "Smart transmission grid: Vision and framework," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 168–177, Sep. 2010.

[7] L. Yang, P. A. Crossley, A. Wen, R. Chatfield, and J. Wright, "Performance assessment of a IEC 61850-9-2 based protection scheme for a transmission substation," in *Proc. 2nd IEEE PES Int. Conf. Exhib. Innov. Smart Grid Technol. (ISGT Eur.)*, Dec. 5–7, 2011, pp. 1–5.
[8] L. Wang, J. Suonan, S. He, and G. Song, "The development and perspective of relay protection technology," in *Proc. IEEE Innov. Smart Grid Technol.—Asia (ISGT Asia)*, May 21–24, 2012, pp. 1–4.
[9] R. J. Yinger, S. S. Venkata, and V. A. Centeno, "Southern California Edison's advanced distribution protection demonstrations," *IEEE Trans. Smart Grid*, vol. 3, pp. 1012–1019, Jun. 2012.
[10] H. Liu, X. Chen, K. Yu, and Y. Hou, "The control and analysis of self-healing urban power grid," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1119–1129, Sep. 2012.
[11] P. Mahat, Z. Chen, B. Bak-Jensen, and C. L. Bak, "A simple adaptive overcurrent protection of distribution systems with distributed generation," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 428–437, Sep. 2011.
[12] J. F. Borowski, K. M. Hopkinson, J. W. Humphries, and B. J. Borghetti, "Reputation-based trust for a cooperative agent-based backup protection scheme," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 287–301, Jun. 2011.
[13] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.
[14] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems," *IET Gener., Transm., Distrib.*, vol. 7, no. 5, pp. 451–463, May 2013.
[15] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. Ind. Electr.*, vol. 58, no. 6, pp. 2547–2555, Jun. 2011.
[16] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892–902, Apr. 2009.
[17] G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235–1246, Apr. 2007.
[18] S. H. Horowitz and A. G. Phadke, *Power System Relaying*, 3rd ed. Hertfordshire, U.K.: Res. Studies Press Ltd./Wiley, May 2008.
[19] "Network protection & automation guide," ALSTOM, 2012.
[20] M. R. D. Zadeh, T. S. Sidhu, and A. Klimek, "Field-programmable analog array based distance relay," *IEEE Trans. Power Del.*, vol. 24, no. 3, pp. 1063–1071, Jul. 2009.
[21] X. Liu, A. H. Osman, and O. P. Malik, "Real-time implementation of a hybrid protection scheme for bipolar HVDC line using FPGA," *IEEE Trans. Power Del.*, vol. 26, no. 1, pp. 101–108, Jan. 2011.
[22] S. P. Valsan and K. S. Swarup, "Protective relaying for power transformers using field programmable gate array," *IET Elect. Power Appl.*, vol. 2, no. 2, pp. 135–143, Mar. 2008.
[23] A. A. Girgis, "A new kalman filtering based digital distance relay," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 9, pp. 3471–3480, Sep. 1982.
[24] H. J. Altuve Ferrer, I. Diaz Verduzco, and E. Vazquez Martinez, "Fourier and Walsh digital filtering algorithms for distance protection," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 457–462, Feb. 1996.
[25] N. T. Stringer, "The effect of DC offset on current-operated relays," *IEEE Trans. Ind. Appl.*, vol. 34, no. 1, pp. 30–34, Jan./Feb. 1998.
[26] J. C. Gu and S. L. Yu, "Removal of DC offset in current and voltage signals using a novel Fourier filter algorithm," *IEEE Trans. Power Del.*, vol. 15, no. 1, pp. 73–79, Jan. 2000.
[27] T. S. Sidhu, X. Zhang, F. Albasri, and M. S. Sachdev, "Discrete-fourier-transform-based technique for removal of decaying DC offset from phasor estimates," *IEE Proc. Gener., Transm., Distrib.*, vol. 150, no. 6, pp. 745–752, Nov. 2003.
[28] A. H. Osman and O. P. Malik, "Transmission line distance protection based on wavelet transform," *IEEE Trans. Power Del.*, vol. 19, no. 2, pp. 515–523, Apr. 2004.
[29] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, no. 3, pp. 330–334, Sept. 1959.
[30] J. S. Walther, "A unified algorithm for elementary functions," in *Proc. AFIPS Spring Joint Comput. Conf.*, 1971, vol. 38, pp. 379–385.
[31] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Process. Mag.*, vol. 9, no. 3, pp. 16–35, Jul. 1992.
[32] T. S. Sidhu, X. Zhang, and V. Balamourougan, "A new half-cycle phasor estimation algorithm," *IEEE Trans. Power Del.*, vol. 20, no. 2, pp. 1299–1305, Apr. 2005.
[33] P. M. Anderson, *Power System Protection*. New York: IEEE Press, 1999.
[34] Xilinx Inc., Xilinx Virtex-7 user guide 2013, .
[35] Power System Relaying Committee, Report of Working Group I-01, "Understanding microprocessor-based technology applied to relaying," Jan. 2009.



Yifan Wang received the B.E. degree in electrical engineering from the China Agricultural University in Beijing, in 2011. Currently she is pursuing her M.Sc. degree at the University of Alberta, Edmonton, Canada. Her research interests include power system protection, FPGA-based hardware design, power system simulatio,n, and computation.

Venkata Dinavahi received the Ph.D. degree from the University of Toronto, Canada, in 2000. He is a Professor in the Department of Electrical & Computer Engineering at the University of Alberta, Edmonton, Canada. His research interests include real-time simulation of power systems and power electronic systems, large-scale system simulation, and parallel and distributed computing.