Open Source Heterogeneous Constrained Edge Computing Platform for Smart Grid Measurements

Ashish Joglekar, Gurunath Gurrala, *Senior Member, IEEE*, Puneet Kumar, Francis C Joseph, Kiran T S, K. R. Sahasranand, Himanshu Tyagi, *Senior Member, IEEE*

***Abstract*—This paper presents a low cost open source heterogeneous resource constrained hardware platform called "Parallella" as a measurement platform for edge computing applications research in smart grid. The unique hardware architecture of the Parallella provides a multitude of edge compute resources in the form of a Zynq SoC (dual core ARM + FPGA) and a 16 core co-processor called Epiphany. A multi-functional IED design is demonstrated to show case the capabilities of the platform. A custom I/O board has been developed for the desktop and embedded versions of Parallella which can interface external daughter boards for measurements and custom peripherals. One such daughter board is an analog sensing board which can measure 3 phase voltages and 4 line currents using a 16 bit synchronous ADC set at 32 kHz. The ADC samples are synchronized with a GPS PPS time clock as the global time reference. This 7 channel high rate raw waveform data is sent to a cloud server over a bandwidth limited communication channel using a custom anomaly aware data compression algorithm implemented on ARM. A Phasor measurement algorithm using Teager Enegy Operator is implemented on FPGA. A parallel power quality measurement algorithm is implemented on the 16-core co-processor Epiphany.**

***Index Terms*—Parallella, Edge Computing, Phasor Measurement Unit (PMU), Power Quality (PQ), Signal Compression, Waveform Recording**

I. Introduction

Mobile computing architectures have improved the capabilities of edge devices that are used in smart grid applications. Multi core processors, Graphic processing units (GPUs), Field Programmable Gate Arrays (FPGA), co-processors etc. are being integrated in mobile/edge compute platforms at an ever increasing pace. Moreover in recent years many of these platforms are being made available as development boards to the research community thus enabling unique edge compute applications. Some popular platforms are NVIDIA-Jetson, Adapteva Parallella, Zynq Z-board, Intel Agilex etc. for industrial/product developments and Rasberry Pi, Beaglebone etc. for hobbyists/proof of concept developments [1].

Many measurement platforms with multiple functionalities, called as intelligent electronic devices (IED) have been proposed in the literature for recording and analyzing measured signals. [2] described a micro-controller with two dsPIC33 processors, one for sampling data at $7.68\ kHz$, computations and the other for communication/device control, to satisfy the PQ measurement requirements of the Brazilian power system. [3] proposed a synchronized $10\ kHz$ data recorder with FFT for feature extraction. [4] discusses the need to have low cost measurement device for research purposes; the developed OpenPMU captures 6 channels data $6.4\ kHz$. [5] proposed a PMU for substation based on FPGA board sampling at $12.8\ kHz$. [6] developed a multi purpose board based on Mini-ITX Motherboard with peripherals for $25.6\ kHz$ signal capture; the device performed PMU, PQ, signal capture and event logging functions. [7] developed a $12.8\ kHz$ sampling A/D board that was interfaced to notebooks to communicate data to upstream servers. The notebooks also performs power quality indices computations. In [8], the authors used a $7.68\ kHz$ recorder developed on FPGA & ARM Cortex M4 board. The device performed a compression based on detection of a novelty in the waveform using DWT, to reduce the size of transmission. [9] developed a PMU on a BeagleBone Black, which features an ARM core and 2 Programmable Real-time Units (PRU). The PRUs were responsible for sampling at $12.8\ kHz$. The authors proposed use of either a discrete PLL ASIC or a software based PLL with reduced accuracy for GPS disciplining of the ADC clock. The ARM performed the frequency and phasor estimation and communication. [10] proposed an FPGA based PMU which sampled data at $32\ kHz$ using a NovTech IoT Octopus board using an iterative Interpolated DFT for estimation. [11] provided a review of commercial PMUs, open source PMUs and integrated PMU with DPR along with PMU. [12] discusses a $50\ kHz$ IED intended for prototyping HVDC protection on real-time simulators, for research. Built on ZedBoard with modular peripherals, the IED performs 16 channel signal sampling on one of its peripheral module and GPS synchronization on another. [13] shows a very high frequency data capture, 2^{15} samples per cycle but requires a potent desktop for real-time signal processing and frequency estimation. This paper focuses on the development of a measurement platform for smart grids based on a unique hardware platform developed by Adapteva, advertised as a credit card sized super computer Parallella. This platform has 3 compute units: a Zynq SoC (industry proven SoC), housing 2 ARM (Cortex A9) cores + FPGA, and Epiphany, a 16 core co-processor unit. Parallella is available in three variants viz. server, desktop and embedded versions, consuming less than $5\ W$ starting at a low cost of $99. This heterogeneous resource constrained compute architecture combines features of CPU, DSP, ASSP on a single platform and makes the Parallella unique. Parallella does not come pre-programmed with any external sensor I/O interface options. The internal ARM ADC is disabled on all versions of the Parallella. The default FPGA binary file on Parallella does not allow external sensor interface to the FPGA GPIOs. The co-processor Epiphany can only be accessed from the ARM through a proprietary e-Link protocol module implemented on the FPGA. [14].

In this paper, a custom I/O board for the Parallella desktop and embedded versions is developed to make it amenable for peripheral sensors integration. One such peripheral, an Analog

This work was supported by the Bosch Research and Technology Centre, Bengaluru, India and by the Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, Bengaluru, India, (under Project E-Sense: Sensing and Analytics for Energy Aware Smart Campus)

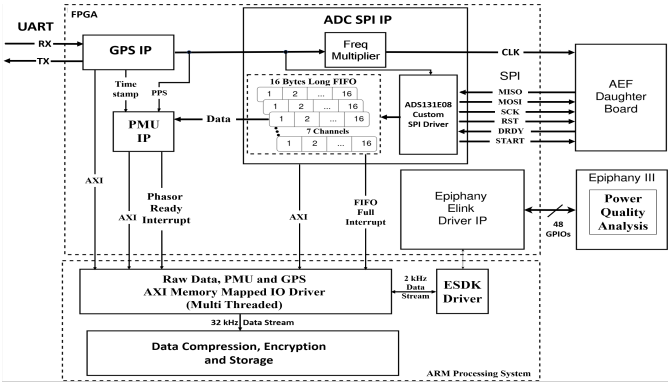


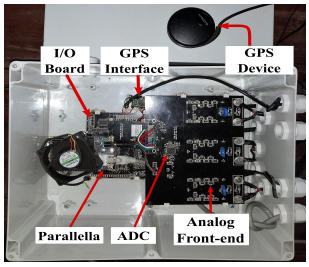
Fig. 1: Proposed IED Architecture on Parallella

Front End (AFE) daughter board for analog measurements, using a synchronous sigma-delta ADC at a maximum sampling rate of 64 kHz, is developed for Parallella. A technique to synchronize the ADC's clock to the GPS time clock PPS reference is also provided [9]. A custom SPI ADC driver and a GPS driver to synchronize to GPS time clock reference have been developed on FPGA. A custom FPGA binary has been compiled to port the developed custom drivers and algorithms and also to expose the Zynq I/Os without loosing the default e-link driver which enables a high speed communication interface between the ARM and the Epiphany co-processor. Three applications are demonstrated, a custom anomaly aware data compression algorithm on the ARM to enable transfer of high frequency sampled waveform data to a server, a Teager Energy Operator (TEO) based phasor measurement algorithm proposed in [15] on the FPGA and a parallel power quality measurement algorithm [16] on 16-core epiphany. This is a first of its kind effort in the literature, to the best of our knowledge, that explores the applicability of an open source edge computing platform under 5 W, with three varieties of compute modules for a smart grid measurement platform.

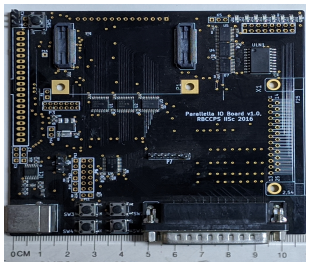
II. Multi Functional IED Hardware Design

Fig.1 shows the internal architecture of the developed multi-functional IED. A 5 W, 18-core credit card sized microcomputer called Parallella-Desktop is used as the edge compute module. The heart of this microcomputer is the Zynq-7010 SoC which features a 7 series FPGA programmable logic. The FPGA features $28k$ Logic cells, 2.1 Mb of block RAM, 80 DSP slices, and 150 I/O pins. The SoC also features a Dual-core ARM Cortex-A9 MPCore clocked at 650 MHz. The Parallella features a 16 core co-processor called Epiphany. Cores can exchange data with each other via a low latency mesh network on chip [14]. The Epiphany architecture allows programming of each core using typical C code [17]. An SRAM alongside each core is required to be managed considering the lack of cache. At 2 W, the Epiphany SoC provides promising power efficiency (performance/watt). The Epiphany talks with the Zynq over e-Link implemented on the FPGA. The Epiphany e-Link driver can be accessed from the ARM's user space over a memory mapped AXI interface.

Parallella has been housed on a custom I/O board which provides the necessary hardware interfaces for making peripherals compatible with the Zynq SoC. One such peripheral is a custom AFE daughter board designed for sensing 3-line voltages, 3-line currents and 1-neutral current. Other peripherals like GPS, environmental sensors are also interfaced. Fig.2(a) shows the developed prototype IED showcasing all the subsystems. The AFE provides 8 synchronous differential analog channels for the IED. The device can sample simultaneously across these 8 channels with zero phase error, 16 bit resolution and a maximum sampling rate of 64 kHz. The ADC's sampling clock is synchronized to the GPS universal time clock reference. The IED provides 8 high drive strength GPIOs for actuation of external circuit breakers or relays. The IED connects to a 10 $Mbps$ Ethernet backhaul. The device also has 1 GB SDRAM and 32 GB of on-board SD card storage.



(a) Prototype IED



(b) I/O Board

Fig. 2: I/O board and Level Converter

The efficient utilization of the available compute modules is a very important step in edge compute devices like Parallella [18]. In this paper, FPGA is proposed to be used for sensing and compute intensive real-time algorithms. Hardware drivers for GPS (UART) and ADC (SPI) are ported on the FPGA to meet the communication bandwidth requirements when the data is sampled at 32 kHz. Provision is made to sample the data at different data rates upto 64 kHz depending on the requirement. The ARM processor is intended to be used for communication with external world, file management on SD card and security algorithms. The ARM acts as an arbitrator to send sampled data obtained from the FPGA based ADC driver to the multi core co-processor. The Epiphany is used for data crunching applications like computation of power quality on the measured signals. For want of space, security algorithms implemented on Parallella are not discussed, [19] can be referred for this.

A. I/O Board

The e-link Epiphany interface utilizes 48 GPIOs of the FPGA. This leaves us with only 24 external GPIOs on the desktop edition of the Parallella. However these are not exposed on the Parallella for external use which limits the usability of Parallella for sensor integration. A custom I/O board is developed to provide a high speed, level shifted GPIO interface to make the FPGA GPIOs accessible for external peripherals like analog sensing and environmental sensors etc. This is one of the major contributions of the paper. Pin mappings for custom GPIOs were added in the Xilinx Design Constraints (XDC) file. This file maps physical pins on the FPGA to signals that interface with the peripheral driver IP. The electrical property standards are also defined for each GPIO in this file. The pin mappings for the Epiphany E-Link driver are also present in the XDC file. Besides FPGA GPIOs, ARM GPIOs can also be directly routed to the physical pins using the Extended Multiplexed I/Os (EMIOs) routed on

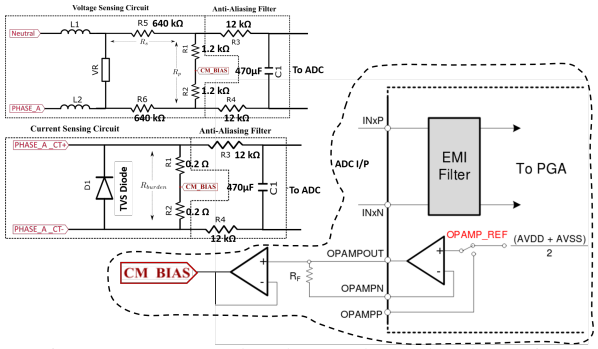
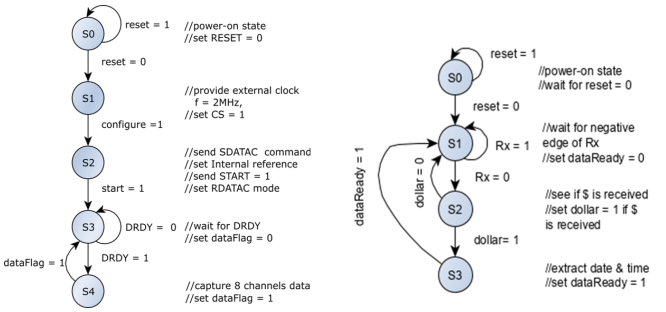


Fig. 3: Current and Voltage Channel Schematics



(a) ADC Driver (b) GPS Driver

Fig. 4: ADC and GPS State Diagrams

FPGA fabric. The I/O board uses high speed level converters to convert the Parallella's 1.8 V level GPIOs to a 3.3 V/5 V level GPIO. The schematic of the developed I/O board is shown in Fig.2. 8 GPIOs are interfaced with a Darlington driver to improve their current sourcing, sinking capability for connection with external relays/circuit breakers. Additional interfaces for future expandability have also been brought out on the custom I/O board. For example, BOSCH XDK - The Sensor X-perience (environmental sensors) has been interfaced with the I2C port of the ARM over extended multiplexed IOs (EMIO). Other peripherals provided by Parallella, Ethernet port, SD card slot and USB port are kept intact. AXI addresses are assigned to custom peripherals using the address editor in Vivado. AXI memory mapped peripherals like the ADC's SPI driver are declared in the device tree file located on the ARM's boot partition. The entry in the device tree specifies the physical address allocated to the peripheral, and the interrupts (if any) that the device will trigger. The entry also contains the corresponding linux kernel driver that will be loaded on boot. The FIFO full AXI GPIO is mapped as a Userspace IO framework (UIO) device in the device tree. This makes the FIFO full interrupt available to any userspace code on the ARM processing subsystem.

B. Analog Front End (AFE)

The Zynq's inbuilt two channel 12-bit ADC is insufficient for 7 channel simultaneous sensing. So, an 8-channel AFE daughter board using a 16-bit synchronous ADC (ADS131E08) at a maximum sampling rate of 64 kHz (currently set to 32 kHz) is developed. The board can be directly interfaced to a 1 A/5 A CT (0.2Ω burden) and upto 440 V RMS input voltage terminals. A common mode DC bias is provided to allow measurement of bidirectional voltages and currents. A technique to synchronize the ADC's clock to a GPS time clock PPS reference to enable synchronous sampling with a global time reference is implemented. The schematics of the differential CT secondary current and line voltage channels are shown in Fig.3.

The ADC on the AFE daughter board communicates over an serial peripheral interface (SPI) interface with the FPGA. The ADC cannot communicate directly with the Zynq's ARM processor as the ARM SPI peripheral driver does not provide the BW needed when 7 channels are sampled at 32 kHz. As a result, a custom ADC SPI driver IP is developed for the FPGA. The custom driver IP has the necessary code to configure the ADC, read and buffer sampled data in synchronization with the GPS time clock reference. Fig.1 shows the internal subsystems of the ADC SPI driver IP developed in the FPGA along with the peripheral interconnections. This driver is realized using a state diagram shown in Fig.4(a) and the states are described below:

1) Default power-on state is S0. Set RESET = 0 on next clock edge after power-on.
2) In state S1 start external clock at 2 MHz and then set chip select (CS) = 1.
3) In state S2, the ADC registers are configured.
4) S2: send Stop Data Continuous (SDATAC) command since device wakes up in Read Data Continuous (RDATAC) mode.
5) S2: set sampling rate and set reference as 2.4 V and internal amplifier operation for common mode bias voltage.
6) S2: send START command to start conversions and send RDATAC to put device back in RDATAC mode.
7) In state S3, set dataFlag = 0 indicating data is not ready. As soon as DRDY becomes 1, state is changed to S4.
8) In state S4, data for 8 channels are received. Then data flag is set to 1, indicating data is ready. After saving the data state is changed to S3.

Nomenclature for Fig.4(a): capital is for command/signals sent to or received from ADC while small letters are for signals generated within SPI driver to control the state machine. To allow synchronization across ADCs, the ADC is configured to accept an external clock signal.The internal ADC reference is set to 2.4 V and the common mode bias, $V_{DD}/2 = 1.5\ V$. The differential ADC channels are set up with unity PGA gain.

C. Interface of ADC with GPS

AFE ADC (ADS131E08) is an 8 channel simultaneous-sampling sigma-delta ADC. The ADCs across IEDs need to be synchronized to enable synchronous PMU functionality. A mechanism to correct oscillator frequency drift for a SAR ADC based AFE was proposed in [20]. Sigma-Delta ADCs are known to be less sensitive to clock jitter compared to SAR ADCs. In a setup similar to [21], the ADC master clock is generated using an FPGA based custom clock generator module that uses the GPS universal clock reference for drift correction. If the START pulse and ADC clock signal are synchronized with the edge of the PPS clock reference, sampling across ADCs is guaranteed to be in sync as per the device datasheet. The reference clock is used to trigger start of conversion and also to generate the ADC's external CLK signal. A frequency multiplier block generates the ADC's external clock reference of 2 MHz from the GPS PPS clock reference on the FPGA.

D. Global Positioning System (GPS)

GPS device gives Coordinated Universal Time (UTC) date and time of the day information. Garmin GPS 18x LVC-5m which transmits this information using Universal Asynchronous Receiver-Transmitter (UART) protocol is used. This

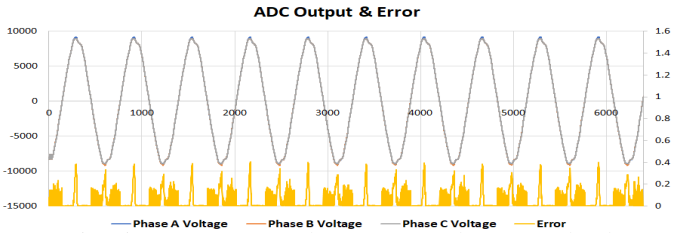


Fig. 5: ADC output when connected to same load

device also gives measurement pulse output (also called PPS) after reporting valid and accurate position fix for at least previous 4 seconds. Rising edge of this pulse is aligned to start of each GPS second within 1 μs which is highly accurate. Before plugging GPS to PMU processor, baud rate for the UART communication is set to 38400. Also GPS device gives many output sentences relating to date, time, day and many more; but for PMU device date and time information is enough. Hence only GPRMC output sequence is made available which can give desired information. Since GPS device gives output at RS-232 level and PMU processor accepts data at TTL-level, RS232 to TTL UART level converter, MAX-232 is used as a link between GPS device and PMU processor. As in the case of ADC, an equivalent GPS driver to acquire GPS data is developed. Output sequence which need to be decoded is as follows:

$\$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12>*hh<CR><LF>$

where $<1>$ contains UTC time in hhmmss format and $<9>$ gives UTC date in ddmmyy format. State diagram of the GPS driver implementation is shown in Fig.4(b), which can be summarized as:

1) Default power-on state is S0. When reset = 1 go-to state S1.
2) In state S1, wait for negative edge of Rx (or Tx from GPS device). After that, reset dataReady and go-to S2.
3) In state S2, check if the first character is $ (=2A hexadecimal). If "$" is received then set dollar = 1 and go-to S3, else reset dollar = 0 and return to state S1.
4) Extract UTC time and date information from obtained GPRMC sentence. After this, set dataReady = 1 and go-to state S1.

Receiver (Rx) of GPS driver is tied to transmitter (Tx) of GPS device (through MAX-232). Every signal that is input to ADC driver and GPS driver from outside the Parallella Board i.e. ADC device and GPS device, has been passed through signal de-bouncers. The performance of the simultaneous sampling on the IED was evaluated in a similar way as in [9]. All the 3-phases of the AFE are connected to a single phase load powered by 50 Hz power supply. The integer output values of the ADCs are shown along with the maximum error among the signals in Fig.5. The plot shows 10 cycles of power frequency waveform captured in 6400 data points, hence the sampling frequency would be 32 kHz. The error is found to be under 0.05% with respect to the 16-bit ADC.

E. Data Pipeline

The ADC is configured to sample 7 channels at 32 kHz (640 samples per line cycle) with 16 bit resolution. The data rate can be calculated using the equation below.

$$DR\,(Kbytes/sec) = (r \times f_s \times c)/1024/8 \qquad (1)$$

where DR is the data rate in $KBytes/sec$, r is the bits per sample, f_s is the sampling rate and c is the number of channels sampled. Accordingly the data rate for the IED is

$$DR = (16 \times 32000 \times 7)/1024/8 = 437.5\ \ KB/s \qquad (2)$$

The FPGA ADC driver IP provides a 16 samples wide FIFO for each channel. For a 16 samples wide FIFO a PL fabric *FIFO full* interrupt is raised at 2 kHz $(32000/16)$. This interrupt is serviced by a dedicated ARM core. The CPU affinity flag is set in the OS to attach a single core to this interrupt. The interrupt service routine transfers the data stored in the FIFO over a memory mapped AXI interface at a transfer rate of $2.4GB/s$. Thus the FIFO is flushed before the next sample set. Raw data sampled at 32 kHz is compressed by a thread running on the ARM core using a custom anomaly aware compression algorithm. 32 kHz data stream is fed to the custom PMU IP block within FPGA. The PMU IP block generates a phasor ready interrupt that is processed by the ARM core. Calculated phasors per channel are transferred from the FPGA PMU IP to the ARM over a memory mapped AXI interface. The 32 kHz data stream is also fed to the Epiphany co-processor over the e-Link interface with the ARM acting as an arbitrator. The Epiphany is tasked with power quality measurement.

III. Applications

A number of power system specific applications can be implemented on the IED. This paper describes three such applications on each compute module, a) anomaly aware data compression algorithm on ARM; b) TEO based PMU algorithm on FPGA; and c) Power quality measurement on the Epiphany parallel processor. Here ARM processor also acts as a communication processor, which facilitates data transfer to an external server.

A. Anomaly-Aware Compression Algorithm

The IED would have throughput of 35 $Mbps$, as given in (2), for the sampled raw values. Such a high data rate is formidable and is not amenable for onboard storage or even transmission. Therefore, it is desirable to subsample in an automated fashion, keeping in mind the "frequency content" of the underlying data. The proposed algorithm performs the required adaptive sampling and bitpacking needed while being feasible in a resource constrained hardware. The algorithm prepares a bit-packing of the samples after filtering and uses the number of bits used in the bit-pack as a measure of "informativeness" of the samples. Highly informative samples are retained completely, while less informative ones are further subsampled. In effect, the algorithm detects anomalous behavior (such as fault, power quality disturbances or switching) and chooses between lossless compression and lossy compression (obtained using subsampling), adaptively based on configurable thresholds. This view allows the use of a single, streaming data pipeline for compression, fault-detection, and subsampling for different operations which would have required separate features to be computed in traditional signal processing. The compression algorithm runs on the ARM processor and takes the channel streams as input from the FPGA.

Formally, let $x(t), t \geq 0$, denote the real-valued, continuous-time power signal, and let $x[i] = x(i/f_s)$, $i \in \mathbb{N}$, be the corresponding sampled sequence acquired at sampling

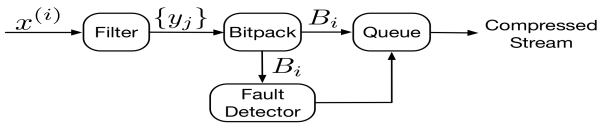


Fig. 6: Anomaly Aware Compression Algorithm

frequency f_s and represented using r-bits per sample. Consecutive samples are grouped into blocks $x^{(1)}, x^{(2)}, \ldots$, with block $x^{(i)}$ comprising samples $x[n(i-1)+1], ..., x[ni]$. Thus, each block x consists of n signed integers taking values in $[-(2^r-1), 2^r]$. In the IED application, block-size $n = 16$ and resolution parameter $r = 16$, are used.

The proposed compression algorithm operates block-by-block and maps the block $x^{(i)}$ to a bit sequence B_i using a fixed quantizer function $Q : x \mapsto Q(x)$. The quantizer Q can be a variable length quantizer which uses different lengths B_is for different is. Note that the length information itself must be included in B_i for signal reconstruction. The goal is to reduce the number of bits used per block, while ensuring that it is possible to construct an estimate $\hat{x}[i]$ of the sampled sequence $x[i]$ from the bit stream $B_1, B_2, \ldots$. It is desirable to have First In First Out (FIFO) recovery, that is, the compressed bit sequences $B_1, ..., B_i$ should suffice to recover the estimate $\hat{x}[i]$ of $x[i]$. The accuracy of recovery is measured by the *normalized mean-squared error* (NMSE) for recovering the sampled sequence up to time N, given by

$$\frac{\sum_{i=1}^{N}(\hat{x}[i] - x[i])^2}{\sum_{i=1}^{N} x[i]^2}. \tag{3}$$

The performance of the algorithm at a fixed NMSE level will be measured by the *compression ratio* (CR) defined as

$$CR = \frac{rnT}{\sum_{i=1}^{T} |B_i|} \tag{4}$$

calculated over T blocks and $|B|$ denotes the length of the bit sequence B. While NMSE is a standard metric in theory, it need not be an appropriate metric for error in power applications. Indeed, it can be observed that the quality of recovered waveform is much better than what is reflected by the NMSE.

A schematic summary of the proposed algorithm is shown in Fig. 6; a preliminary version of the algorithm appeared in [22], the adaptation to the IED environment is described in this work. The first step in the algorithm is to pass each block of data x (comprising n samples) through a linear filter. The filter used is an iterated Δ filter where the output $y = (y_1, ..., y_n)$ is given by $y_1 = x[1]$, $y_2 = x[2] - x[1]$, and for $k = 3, \ldots, n$, $y_k = x[k] - 2x[k-1] + x[k-2]$. The vector y is converted into a bit block B via a bit packing procedure wherein each sample is stored using ℓ bits where ℓ is the maximum number of bits needed to represent any sample within the block. The value of ℓ (represented using $\log_2 r$ bits) is also included in B. An anomaly detector uses the block B and the corresponding ℓ to determine whether to label the block as faulty or normal. In particular, if $\ell > \tau_H$, a fixed threshold, the block is labeled faulty; normal otherwise. Normal blocks are set to be stored in a lossy fashion, but not immediately since it is necessary to retain a few cycles of data before and after the fault. Towards this, the bit block stream $B_1, B_2, ..., B_T$ resulting from the input stream $x^{(1)}, ..., x^{(T)}$ is passed to a FIFO queue of length `BUF` which is set to the number of cycles intended to be retained around the detected anomaly.

Once the block $x^{(i)}$ is set to be stored in a lossy fashion, the bits corresponding to the first sample in $x^{(i)}$ is retained and others dropped. This is repeated at a higher level as well. Namely, if there are m consecutive blocks with only the first sample retained, these first samples are passed through the same linear filter and the output is used to decide whether or not to retain all of the m values or just the first one among them. In particular, if the number of bits per sample resulting from this filter, $\ell > \tau_B$, a fixed threshold, all m values are retained. Otherwise, all values except the first one are dropped. This process of downsampling the data first by n, and then by m if needed, results in a hierarchical subsampling. In effect, the algorithm *tests* whether or not high frequencies are present and adjusts the sampling frequency accordingly. Reconstruction is performed by using splines to interpolate the dropped data points. The IED is connected to measure the author's laboratory input power supply. The measured dataset comprises 5039 files of 1 second duration. The maximum, minimum, and average values of CR and NMSE are calculated for all the 5039 files by varying the threshold settings τ_H and τ_B. Table I provides CR and NMSE for two cases corresponding to a medium and a high compression with $n = 16$, $m = 4$, and `BUF` $= 40$. For the medium case, $\tau_H = 11$ for both the voltage and current channels and $\tau_B = 7$ for voltage channels and $\tau_B = 6$ for the current channels are used. For the high case, $\tau_H = 15$ and $\tau_B = 14$ is used for all the channels. It was observed that an average CR of around 1.42 for current channels and 1.64 for voltage channels can be obtained if all data has to be retained losslessly. From Table I, it can be seen that an average CR of upto around 45 can be configured for waveform recording based on the accuracy requirements for the specified n, m, and `BUF`. Fig. 7 and Fig. 8 show the heat map of compression ratio for a snapshot of voltage and current channel respectively for the two settings in the table. Here, it can be noted that higher compression ratio is attained around the smooth part of the signal and lower around the edges, illustrating adaptive compression. It was observed that the compressed data is captured without any

TABLE I: Maximum, minimum, and average CR and NMSE for two different settings of thresholds for lossy compression and anomaly detection with $n = 16$, $m = 4$, and `BUF` $= 40$

Ch.	CR			NMSE		
	max.	min.	avg.	max.	min.	avg.
Medium Compression						
V_A	16.20	05.58	14.53	0.006222	0.000099	0.000438
V_B	15.70	05.64	14.14	0.004215	0.000083	0.000403
V_C	15.64	05.73	14.05	0.005118	0.000145	0.000485
I_C	15.19	03.13	11.50	0.031000	0.003036	0.011898
I_N	15.17	05.03	13.16	0.030456	0.003435	0.011852
High Compression						
V_A	47.20	17.10	45.75	0.011313	0.002498	0.003303
V_B	46.18	14.33	43.49	0.011204	0.002266	0.003008
V_C	45.52	13.76	42.62	0.014279	0.002453	0.003263
I_C	45.51	15.25	44.25	0.077090	0.012552	0.040798
I_N	45.58	15.29	44.25	0.078481	0.012431	0.038999

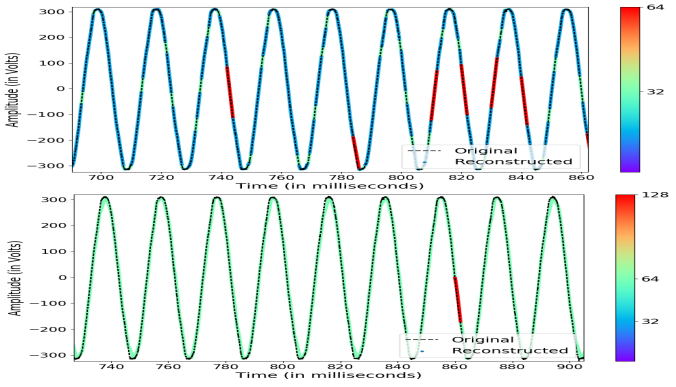


Fig. 7: Heat map of block-wise CR for a voltage channel (top fig: Medium Compression, bottom fig: High Compression)

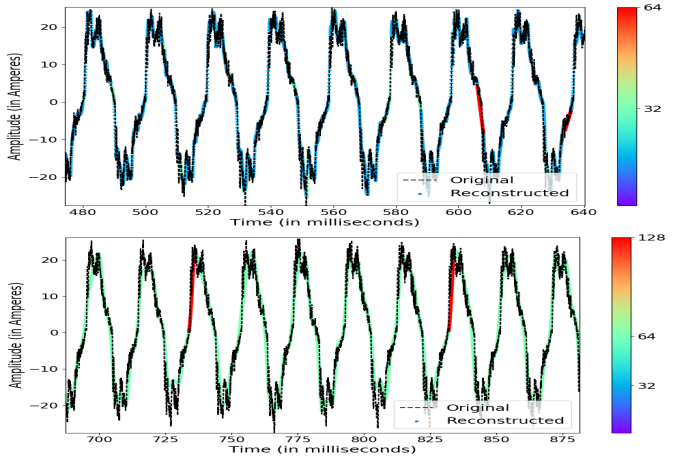


Fig. 8: Heat map of block-wise CR for a current channel (top fig: Medium Compression, bottom fig: High Compression)

loss and without overloading the network bandwidth. The data collected and the computed data by the IED is streamed to the local server for storage and visualization. The user can access the streamed timestamped data of the applications via a web browser.

B. Teager Energy Operator (TEO) based PMU algorithm

Fig.9 shows the block diagram of the TEO based technique proposed in [15] implemented on the FPGA. As shown in the figure, the PMU IP block is fed with a 32 kHz raw data stream. The measurement involves four main stages: 1. Band pass filter (BPF), 2. Teager energy operator, 3. Low pass filter (LPF) and 4. Moving Average filter (MAF). Only a brief explanation of the stages are discussed here due to space constraints. The raw data signal is first passed through an adaptive band pass filter. The adaptive BPF is based on a recursive discrete Fourier transform (RDFT). It was observed that updating the number of samples N dynamically is not numerically stable in a recursive floating point implementation [23]. For practical implementation N is fixed and calculated as $N = f_s/f_0$, where f_s, f_0 are sampling and fundamental system frequency respectively. When previous estimated frequency $\hat{f}[n-1]$ drifts from fundamental (f_0), the phase and amplitude error are calculated as discussed in [24] and accordingly compensation is done for the new sample $x(n)$. Using the calculated amplitude one has to normalize the instantaneous fundamental component of the input signal.

The TEO concept was introduced by Teager and Kaiser to calculate the energy needed to generate speech signals [25].

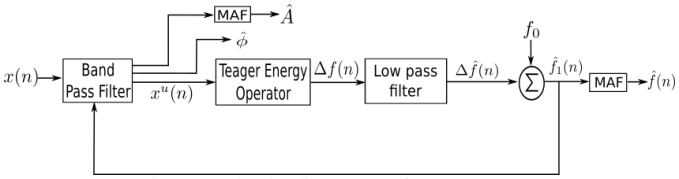


Fig. 9: Block diagram of RDFT-TEO

TABLE II: Resource Utilization for PMU algorithm

Resource	Available	Utilization		Utilization %	
		SE	Total	SE	Total
FF	35200	6885	16575	19.56	47.09
LUT	17600	9251	14296	52.56	81.23
Memory LUT	6000	140	592	2.33	9.87
BRAM	60	7	7.5	11.67	12.5
DSP48	80	17	17	21.25	21.25

TEO is a non-linear operator and can be used to estimate energy of a sinusoidal signal. The TEO block takes the normalized fundamental component as input and outputs the frequency deviation from nominal. TEO is given as:

$$E[x(n)] = x^2(n) - x(n-1) * x(n+1) \quad (5)$$

For a sinusoidal signal, $x(n) = A\cos\left(2\pi\frac{f}{f_s}n+\phi\right)$, the Teager energy is given as: Hence, it can be shown

$$E[x_1^u(n)] = \sin^2\left(2\pi\frac{f}{f_s}\right) \quad (6)$$

where $x_1^u(n)$ represents normalized version of $x_1(n)$. It is worth noting that only three consecutive samples are required to find the frequency deviation from nominal $\Delta f(n)$ [15] using TEO as shown below:

$$\Delta f(n) = \frac{\sqrt{\{x_1^u(n)\}^2 - x_1^u(n-1)x_1^u(n+1)} - S_0}{2\pi C_0} * f_s \quad (7)$$

where $S_0 = \sin(2\pi f_0/f_s)$ and $C_0 = \cos(2\pi f_0/f_s)$. This first order infinite impulse response (IIR) LPF restricts the bandwidth of the estimated amplitude normalized fundamental signal component. Cutoff frequency of LPF is calculated using, [15]

$$\omega_{cut} \leq 5/T_{RDFT} \quad (8)$$

For a 50 Hz system, $\omega_{cut} \leq 250$ rad/s, since window size of BPF is one fundamental time period. To further filter the estimated output, a moving average filter is used so as to bring the errors within the PMU standards [26]. Window size for this filter is taken as one fundamental period of nominal frequency (50 Hz) and its transfer function is given by:

$$H_{maf}(z) = \frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}} \quad (9)$$

where M is the filter order.

1) Implementation on hardware: RDFT-TEO algorithm is implemented on the FPGA. Xilinx Vivado HLS v2014.3 tool chain was used to convert the "C" code to "HDL". The following points need to be followed to ensure optimal resource utilization within device limits [27], [28]:

- Use of $float32$ instead of $float64$
- Use of directives such as #PRAGMA
  Ex: *#pragma HLS allocation instances=fadd limit=1 operation*
- Avoid use of dummy variables to optimally utilize memory elements.

The final resource Utilization of the TEO-PMU algorithm is shown in Table II, where "SE" represents resource utilization of the AXI compatible synchrophasor estimator IP and "Total" represents resource utilization of SE-IP along with SPI ADC driver IP, GPS driver IP, AXI Interconnect and Epiphany driver IP. It is clear that resource utilization has been efficient with

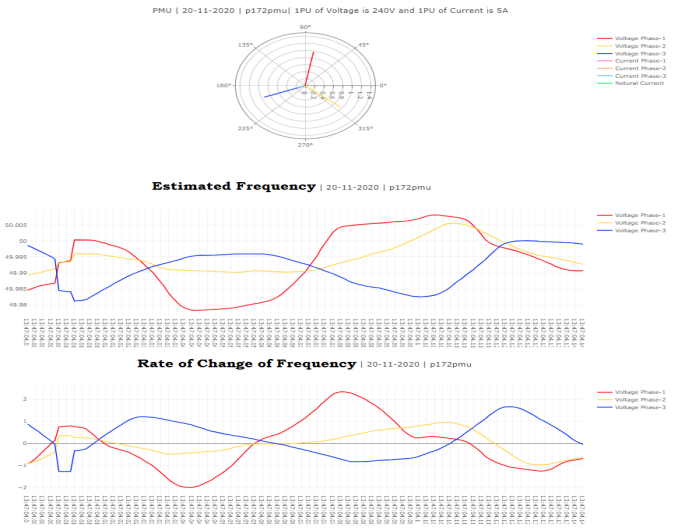


Fig. 10: Real-time PMU-Dashboard

room for further expansion. The PMU data is also sent to the server for real-time display and storage. A snapshot of real-time PMU dashboard is shown in Fig.10. Voltage phasors, corresponding estimated frequencies and rate of change of frequencies for each channel are displayed.

C. Measurement of Power Quality

Power quality (PQ) refers to a wide variety of electromagnetic phenomena that characterize the voltage and current at a given time and at a given location on the power system [29]. In this work, the harmonic and inter-harmonic content of the captured voltage and current waveforms is calculated. The power quality parameters, magnitude, phase, crest factor, Total Harmonic Distortion (THD) and Total Demand Distortion (TDD), are used to define the harmonic content. The basic measurement time interval for calculating these parameters is a 10-cycle time interval for $50\ Hz$ power system as per the STD IEC 61000-4-30 [29]. For power system applications upto 50th Harmonic (i.e $2.5\ kHz$) would need to be captured [16]. Hence the captured waveform is down-sampled to $6.4\ kHz$ by dropping samples. Considering the nominal case, 10 cycle at $6.4\ kHz$ would lead to 1280 samples per signal. The nearest 2 power would be 1024, hence another stage of re-sampling is achieved by using Newton divided difference interpolation. The required anti aliasing filter is applied at the interpolation stage before passing through Fourier Transform. Considering the frequency resolution, the Fast Fourier Transform (FFT) would be the suitable choice since the computation complexity would be $\mathcal{O}(n\ log\ n)$ where n is the no of data points [30].

Usage of Epiphany: The measurement of power quality is divided into 4 tasks as shown in Fig.11. The 16 core Epiphany co-processor, which is fabricated as a $4x4$ mesh, is used to calculate the PQ parameters by efficiently mapping these tasks to one or more cores called as workgroups (W1 to W4). The row and column indices form the core number. Fig.11 shows the cores that are grouped with other adjacent cores in the same row; this reduces the latency of core to core communication, as message is passed along the row then across the column [17]. The first task, the decimation and channel segregation, is assigned to work group W1 having the core (2,0). The second task, interpolation and buffering for 10 cycles data, is assigned to work group W2 consisting of cores (2,1) and (2,2).

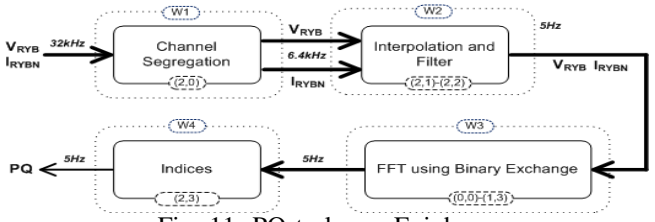


Fig. 11: PQ tasks on Epiphany

One core processes the voltage and another the current, the core processing the voltages acts as master. The anti aliasing filter is implemented at $5.12\ kHz$ which corresponds to the sampling interval for 1024 point FFT. Since $200\ ms$ of data comprising of seven channels lead to nearly $35\ KB$ block, this is stored in the $32\ MB$ shared space. Separation of the cores into work groups forces the data sharing only from the shared space or host [17]. This leads to utilization of flag based handshake between the adjacent workgroups and host for process management.

A parallel FFT algorithm is used in the Epiphany, in which the data is methodically divided among the work group W3 consisting of processors (0,0) to (1,3). The Binary Exchange algorithm [30] is one such parallel FFT method which is based on the CooleyTukey algorithm, and can be deployed on any 2^x processors. Alternatively, the 2D Transpose FFT [30] can be used but the number of processors are limited to even powers of 2. Since pre-processing and other tasks are to be accommodated in the Epiphany cores, the maximum cores that can be used for 2D Transpose is 4 whereas for the Binary Exchange it can be 8. Hence the data space used by 2D transpose is twice that of Binary Exchange. However, the space available for instruction and data for each Epiphany is only $32\ KB$ [14]. So Binary Exchange is utilized for FFT deployment. The FFT work group W3, writes back the results to another block in the 32 MB shared space. Apart from FFT, W3 computes the true RMS and max value per channel required for the indices calculation.

The PQ parameters (indices) are calculated in work group having core (2,3). ARM can access the computed indices from W4 and sends to a server. The Epiphany SDK [17] coding framework allows loading different codes for each workgroup; which helps in keeping smaller codes for each task rather than a single code enforced on all groups. The resource utilization and average run time for each task is provided in Table.III. The timing is done on the data captured for a $2\ s$ duration by the IED. It can be observed that the computation time (including communication time) is within the output interval of each workgroup. Fig.12 shows the real-time PQ-dashboard showing max values, peak values, crest factor, K-factor, THD and TDD.

TABLE III: Epiphany Usage

Module	Image Size	Avg Time	Output Interval
Segregation	3.9KB	0.064ms	2.5ms
Interpolation	12.5KB	134.846ms	200ms
FFT	25.4KB	47.83ms	200ms
Indices	10.9KB	2.804ms	200ms
One Run of Fig.11	-	185.986ms	200ms

The total hardware cost of the proposed IED is approximately INR.60000 ($760 USD). Since Parallella is an open platform and incorporates industrial Zynq platform at $< 5\ W$ power, it makes a great choice for even commercial product development at affordable cost.

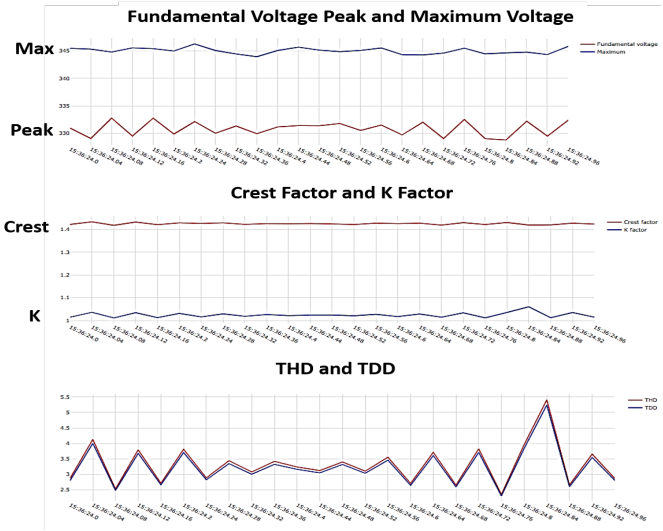


Fig. 12: Real-Time PQ-Dashboard (V_A shown)

IV. Conclusion

This proper introduced a low power heterogeneous edge computing platform called Parallella as a measurement platform for smart grid measurements. A custom FPGA binary file and an I/O board development is detailed which enables access to Zynq I/Os without loss of existing functionalities and allows the platform to integrate with the AFE and GPS device. The computation load has seen offloaded to the FPGA and co-processors hence the ARM cores are used mainly for compression and communication. A phasor measurement algorithm on FPGA, a parallel power quality measurement algorithm on epiphany and an anomaly aware compression algorithm on ARM for waveform recording have been demonstrated on the platform.

References

[1] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi, "Edge Computing Perspectives: Architectures, Technologies, and Open Security Issues," *Proceedings - 2019 IEEE International Conference on Edge Computing, EDGE 2019 - Part of the 2019 IEEE World Congress on Services*, pp. 116–123, 2019.

[2] G. P. Colnago, J. L. De Freitas Vieira, G. C. D. Sousa, and J. R. Macedo, "Real time power quality monitoring system according to New Brazilian regulation policies," *ICHQP 2010 - 14th International Conference on Harmonics and Quality of Power*, no. September, 2010.

[3] H. Maass, H. K. akmak, W. Suess, A. Quinte, W. Jakob, K. Stucky, and U. G. Kuehnapfel, "First evaluation results using the new electrical data recorder for power grid analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 9, pp. 2384–2390, 2013.

[4] D. M. Laverty, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti, and D. J. Morrow, "The openpmu platform for open-source phasor measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 4, pp. 701–709, 2013.

[5] P. Castello, P. Ferrari, A. Flammini, C. Muscas, and S. Rinaldi, "A new ied with pmu functionalities for electrical substations," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 12, pp. 3209–3217, 2013.

[6] T. Atalik, I. Čadirci, T. Demirci, M. Ermis, T. Inan, A. S. Kalaycioglu, and Ö. Salor, "Multipurpose platform for power system monitoring and analysis with sample grid applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 3, pp. 566–582, 2014.

[7] H. Maaß, H. K. Cakmak, F. Bach, R. Mikut, A. Harrabi, W. Süß, W. Jakob, K. U. Stucky, U. G. Kühnapfel, and V. Hagenmeyer, "Data processing of high-rate low-voltage distribution grid recordings for smart grid monitoring and analysis," *Eurasip Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–21, 2015.

[8] L. R. Silva, E. B. Kapisch, C. H. Martins, L. M. Filho, A. S. Cerqueira, C. A. Duque, and P. F. Ribeiro, "Gapless Power-Quality Disturbance Recorder," *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 862–871, 2017.

[9] X. Zhao, D. M. Laverty, A. McKernan, D. J. Morrow, K. McLaughlin, and S. Sezer, "GPS-Disciplined Analog-to-Digital Converter for Phasor Measurement Applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2349–2357, 2017.

[10] P. Romano, M. Paolone, T. Chau, B. Jeppesen, and E. Ahmed, "A high-performance, low-cost PMU prototype for distribution networks based on FPGA," *2017 IEEE Manchester PowerTech, Powertech 2017*, pp. 1–6, 2017.

[11] D. Schofield, F. Gonzalez-Longatt, and D. Bogdanov, "Design and Implementation of a Low-Cost Phasor Measurement Unit: A Comprehensive Review," *2018 7th Balkan Conference on Lighting, BalkanLight 2018 - Proceedings*, pp. 1–6, 2018.

[12] I. Jahn, F. Hohn, G. Chaffey, and S. Norrga, "An open-source protection ied for research and education in multiterminal hvdc grids," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2949–2958, 2020.

[13] J. Kitzig, S. Schlaghecke, and G. Bumiller, "Power quality measurement system with pmu functionality based on interpolated sampling," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 4, pp. 1014–1025, 2019.

[14] *Epiphany Architecture Reference*. [Online]. Available: https://www.adapteva.com/docs/epiphany_arch_ref.pdf

[15] S. Reza, M. Ciobotaru, and V. G. Agelidis, "Single-phase grid voltage frequency estimation using teager energy operator-based technique," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 3, no. 4, pp. 1218–1227, 2015.

[16] "IEEE recommended practice for monitoring electric power quality," *IEEE Std 1159-2009 (Revision of IEEE Std 1159-1995)*, pp. 1–94, June 2009.

[17] *Epiphany SDK Reference*. [Online]. Available: https://adapteva.com/docs/epiphany_sdk_ref.pdf

[18] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward Computation Offloading in Edge Computing: A Survey," *IEEE Access*, vol. 7, pp. 131 543–131 558, 2019.

[19] B. P. Robert, P. A. Babu, N. Kashyap, and G. Gurrala, "Practical approaches towards securing edge devices in smart grid," in *2019 8th International Conference on Power Systems (ICPS)*, 2019, pp. 1–6.

[20] W. Yao, L. Zhan, Y. Liu, M. J. Till, J. Zhao, L. Wu, Z. Teng, and Y. Liu, "A novel method for phasor measurement unit sampling time error compensation," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1063–1072, 2016.

[21] Y. Kononov, M. Ospishchev, and V. Rudnev, "Current and voltage synchronous measurements using ni crio system with sigma delta adc ni 92 and sea gps modules," *20th IMEKO TC4 Symposium on Measurements of Electrical Quantities: Research on Electrical and Electronic Measurement for the Economic Upturn, Together with 18th TC4 International Workshop on ADC and DCA Modeling and Testing, IWADC 2014*, pp. 25–30, 2014.

[22] K. R. Sahasranand, B. Rout, A. Joglekar, G. Gurrala, and H. Tyagi, "Fault-aware compression for high sampling rate data acquisition in smart grids," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: ACM, 2019, pp. 422–424.

[23] H. A. Darwish and M. Fikri, "Practical considerations for recursive dft implementation in numerical relays," *IEEE Transactions on Power Delivery*, vol. 22, no. 1, pp. 42–49, Jan 2007.

[24] Maohai Wang and Yuanzhang Sun, "A practical method to improve phasor and power measurement accuracy of dft algorithm," *IEEE Transactions on Power Delivery*, 2006.

[25] J. F. Kaiser, "On a simple algorithm to calculate the 'energy' of a signal," in *International Conference on Acoustics, Speech, and Signal Processing*, 1990.

[26] "IEEE standard for synchrophasor measurements for power systems – amendment 1: Modification of selected performance requirements," *IEEE Std C37.118.1a-2014 (Amendment to IEEE Std C37.118.1-2011)*, pp. 1–25, April 2014.

[27] Xilinx, *Vivado Design Suite User Guide: High-Level Synthesis*. [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug902-vivado-high-level-synthesis.pdf

[28] ——, *Vivado Design Suite Tutorial: High-Level Synthesis*. [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug871-vivado-high-level-synthesis-tutorial.pdf

[29] "Electromagnetic compatibility (emc) part 4-30: Testing and measurement techniques power quality measurement methods," *IEC 61000 4-30 (2003)*, 2003.

[30] A. Grama, V. Kumar, A. Gupta, and G. Karypis, *Introduction to Parallel Computing*, ser. Pearson Education. Addison-Wesley, 2003.