High-Performance Computing Architecture for Sample Value Processing in the Smart Grid

LE SUN, LEIRE MUGUIRA, JAIME JIMÉNEZ, ARMANDO ASTARLOA, AND JESÚS LÁZARO
Department of Tecnología Electrónica, UPV/EHU, 48013 Bilbao, Spain
Corresponding author: Jesús Lázaro (jesus.lazaro@ehu.eus)

This work was supported in part by the Ministerio de Economía y Competitividad of Spain under Project TEC2017-84011-R, in part by FEDER funds, and in part by the Department of Education of the Basque Government within the fund for Research Groups of the Basque University System under Grant IT978-16.

ABSTRACT The digitalization of the Electric Grid is a continuous process, both for operational and user networks. For example, in the energy transport infrastructures, the Ethernet broadcasting of digitalized current and voltage values for control and protection applications is a reality to share data with the newest digital power substations. However, emerging applications such as Distributed Energy Resources (DER) system coordination, Electric Transmission Lines continuous monitoring, and Power Quality assessment demand virtual technologies capable of processing a significant amount of these current and voltage streams measured in multiple and distributed locations in real-time. Concurrently, High-Performance Computing (HPC) increases its capabilities and reduces its costs, making it suitable for semi-distributed systems, such as the Smart Grid (SG). This article presents an innovative solution to accelerate the computation of hundreds of streams, combining a custom-designed silicon Intellectual Property (IP) and a new generation Field Programmable Gate Array (FPGA)-based accelerator card. This solution overcomes the computation and networking limitations of the state-of-the-art solutions based on distributed Intelligent Electronic Devices (IED) or Central Processing Unit (CPU)-based servers and uncovers the Sample Measured Value (SMV) processing complexity. It offers a high-level interface for the application designers.

INDEX TERMS Field programmable gate arrays, high-performance computing, Smart Grid (SG).

I. INTRODUCTION
At present, many devices that collect and process Sample Values (SVs) are designed based on Central Processing Units (CPUs). On the other hand, the most advanced systems are deployed utilizing high-performance Field Programmable Gate Arrays (FPGAs). In comparison, performance and reliability in the former will be an order of magnitude worse than in the latter. Furthermore, processing Sampled Values is risking susceptible to latency; for this, CPU-based equipment is also worse than that based on FPGA. Conversely, the price, development cost, and complexity of FPGA systems are higher. The traditional method of establishing communication between FPGA and the main CPU needs to deploy a Direct Memory Access (DMA) bridge between them, which is hard to be developed. Challenges and general requirements that should be met by a SG communication infrastructure, as well as several technologies and networks are explained in [1]. As Sample Measured Value (SMV) messages carry hundreds of bytes per frame, each sensor generates more than 80 messages per second. There are millions of such sensors in the smart grid. The data they generate is massive. Therefore, processing and collecting it requires high network bandwidth, reliability, and performance. In order to meet these challenges, the setup in this article leverages Xilinx’s new FPGA accelerator adaptive computing framework to speed up the program under the OpenCL structure. It can accelerate a C program and the Hardware Description Language (HDL) coprocessor core running on an FPGA of the framework. This article will deploy the HDL acceleration core in an FPGA coprocessor that works with the main CPU to obtain the advantages of the above two hardware architectures. For a high-speed, high-throughput, low-latency system, such a hardware design has extraordinary significance. The purpose of our experiment is to clarify the number of SV processing cores that can be deployed on a single FPGA accelerator and its data throughput, latency characteristics, and system power consumption.
As a summary, the main contributions of this paper are:

- Exploiting High-Performance Computing to process big data in the Smart Grid.
- An innovative FPGA architecture to accelerate the processing of hundreds of SV streams.
- An alternative SG network structure to take advantage of the above-mentioned architecture.
- A system of high throughput and low latency.

The paper is organized as follows. Section II provides information about the related work. The traditional approach to computing in the SG is described in Section III. Section IV shows the proposed architecture and system. The comparison between the traditional and proposed approaches is described in Section V. The paper ends with some conclusions in Section VI.

II. RELATED WORK

A. NEW APPLICATIONS IN THE SMART GRID

The SG combines power systems, communication, information, and control. One key element is the Intelligent Electronic Devices (IED), which is a device that collects and processes SVs. In contrast with the traditional approach, SG uses a two-way flow of data to obtain a distributed and automated energy delivery system where the efficiency of all the elements of the power system are supervised, managed and improved (see Fig. 1). Challenges and general requirements that should be met by a SG communication infrastructure, as well as several technologies and networks are explained in [2] Information technologies face multiple events or situations, achieving a more efficient, flexible, reliable, resilient, safe, and secure delivery of energy, improving the economic performance [3].

Once this digitalization has gained acceptance in the sector, new applications and use-cases arise. As an example, the Industry has identified that the Fault Detection on cable sections in the transition from aerial to underground high-voltage lines can benefit from this digitalization [4]. In scenarios where the number of measurement points is low (for instance, three pylons) and the longest distance to the furthest pylon is less than 20 km, a solution based on Optical Flexible Transformers and Analog Merging Units can be engineered. However, a centralized computation solution is desirable, if this approach is scaled to the whole high-voltage line or broader sections of the SG. This solution can receive SMV streams using a high-bandwidth data backbone that enables a new range of applications, while obtaining a significant reduction of overall costs. If this approach is scaled in several strategic Grid locations, the number of SMV streams that would need to be transmitted and processed increases drastically. Conventional IEDs cannot achieve this challenge, due to their constrained networking and computation capabilities.

Another representative example of the emerging applications based on the analysis of precise values of the Grid was the Power Quality analysis done at Sundom Smart Grid pilot of ABB Oy [5]. The emerging spread renewable energy sources identified as Distributed Energy Resources (DER) and Microgrids demand real-time response. This actuation needs to be evaluated based on real-time measurements and high-speed networking. In this context, Big Data analysis, Machine Learning (ML), and Artificial Intelligence (AI) play a vital role.

These new technologies need to manage a large volume of data and real-time requirements in some use-cases. The following analysis quantifies an actual potential set-up: the maximum size of an SMV frame is 140 B, and a typical sampling rate is 4 kHz. Therefore, the required data bandwidth for each SMV is 4.48 Mbit/s. According to the Wind Europe report 2019, the number of Wind installations in Europe (on-shore and off-shore) is close to 190,000 [6]. In a potential on-premises aggregation of information from 200 DERs in a small region, assuming 80 measurement points in each, the number of SMV that should be processed in real-time would be 16,000. In total, approximately 72 Gbit/s of net data would need processing under real-time conditions.

These magnitudes are far from the typical computing capacity of IEDs, which are typically capable of processing less than 10 SMV per unit. Furthermore, dedicated CPU processors such as Intel Quad-Core i5 saturate with 80 SMV streams, as shown in [5]. Therefore, the need for hardware acceleration would help to enable this new generation of Big Data Analysis applications.

B. HPC IN THE SMART GRID

As mentioned in the previous section, the requirements of real-time control and monitoring with bidirectional communication and electricity flows have led to the development of SGs. Furthermore, future SGs are expected to have reliable,
efficient, secured, and cost-effective power management to implement distributed architecture. Apart from the emerging applications based on the intensive and real-time analysis of SMV streams, many cloud computing applications for the smart-grid architecture reported in academia [7] highlight their challenges and problems. As an example, the use of cloud computing applications in SG is one emerging technology to solve conventional power grid management issues, despite some technical challenges inherent to cloud computing [8]. In this environment, the concept of fog computing and the Internet of Energy (IoE) appears. In this scenario, the computation is neither done entirely on edge nor in the cloud [9]. Fog computing is further possible by simplifying the hardware required for such architecture, such as Graphics processing unit (GPU) or FPGAs.

A summary of the benefits and challenges of cloud computing for SG application can be summarized as [10]:

- Security.
- Privacy.
- Scalability.
- Flexibility.
- Real-time response.
- Processing power.
- Data Storage.
- Cost.
- Interconnectivity.
- Compatibility.

In [11], a classification of devices used in the SG based on their computational capabilities was introduced. Due to the addition of smart measurement devices such as smart meters, PMUs, and other advanced sensors into the power grid, a huge amount of data is being generated and must be sent, shared, stored, analyzed, and understood in real-time [12]. This processing demands the use of new and faster computational methods, more intelligent and efficient algorithms, and techniques for storing and transmitting data [12]. High-Performance Computing (HPC) increases the performance of these solutions. Green et al. [13] review how HPC has been used in the field of power systems, understanding the SG as a singular Cyber-Physical System (CPS) that needs the cooperation of HPC and the electric power system [12]. They analyzed the work that has been performed to include HPC in power systems to improve real-time analysis, reliability, and security. In [12], it is reviewed how HPC is used in the SG and its CPS are also explored.

The aspiration to reach more significant power of computation and collaboration leads to changes in HPC, which can be divided into processing hardware, communication, and connectivity. Hardware has evolved from a single processor, then combined with the accelerator, to form compute clusters, multi-core systems, grid, and cloud computing [12]. HPC allows accelerating sequential applications that run in different domains through parallel architectures. Therefore, the required time is decreased. Ahmed et al. [11] pointed out the HPC’s contributions in accelerating applications and its cybersecurity. The major challenges of security in the SG are related to availability, data integrity, confidentiality, and reliability.

Multiple studies that explain SG cybersecurity-related issues and several works that achieve very high speedup based on parallelizing encryption algorithms using HPC resources are also referenced in [6]. In some of them, separate accelerators and the CPU are used, such as GPU or FPGA. Nevertheless, the tendency in the architecture of processors shows an approach to heterogeneous processors and MPSoCs. The changes in connectivity and communication make reaching the highest throughput and increasing accessibility possible.

Heterogeneous Computing systems are the most efficient supercomputers of today.

In some aspects of HPC, FPGAs have been proposed as powerful accelerators. In [14] the FPGAs potential in the context of HPC is examined focusing on performance, energy, efficiency, and programmability. They have considerable capacity for HPC and parallel applications of data-centric [15]. Moreover, considering that these devices provide a considerable volume of parallelism, they have less power consumption than that typically used by GPU and CPU. They also provide a substantial quantity of bandwidth of internal memory [15]. However, their programming complexity can be a problem. So multiple higher-level solutions have been developed [16]. OpenCL is an open programming standard for heterogeneous computing that provides a unified C programming model. Fujita et al. [16] proposed a so-called Communication Integrated Reconfigurable Computing System (CIRCUS) to facilitate the high-speed interconnection of FPGAs from OpenCL. They explained that it is possible to accelerate HPC applications on FPGA by combining computation and communication.

Modern FPGA devices are contemplated for coming generation HPC solutions. Besides, FPGA clusters have been opened to the public in some computing centers [15], [17]. Korcyl et al. [15] provided an overview of the principal advantages of the FPGA devices, emphasizing crucial aspects from the perspective of high-performance, and they provided that power-efficient HPC systems could be implemented using only FPGA devices. They described an FPGA implementation of an HPC-specific algorithm on three Xilinx Alveo U250 accelerator hardware platforms. In [17], a software package that implements the Conjugate Gradient algorithm for the Xilinx U280 Alveo accelerator card is presented. In [18], the Xilinx Alveo U250 FPGA architecture was investigated as the infrastructure capable of developing complex numerical simulations. The FPGA global memory bandwidth was saturated, and the kernel version was optimized using the OpenCL standard. More examples where Alveo accelerator cards have been used for different HPC applications are listed next. In [19], a Xilinx Alveo U200 board hosted by a 40-core Xeon server was used to evaluate HPC design; in [20], the design has been benchmarked on two platforms equipped with Xilinx FPGAs: a Xilinx Alveo U200 and an Alpha-Data ADM-PCIE-7V3, and in [21], a C-DAC developed
accelerator card with Xilinx FPGA and Xilinx Alveo U200 was used to increase the speed of critical computations.

Heterogeneous HPC platforms’ most significant optimizing purposes are energy and performance [22]. System-level and application-level solutions can be found for bi-objective optimization. In the first case, the energy and performance of the environment are improved when the applications are performed. In the second solution method, application-level parameters are used to fix the optimization problem.

It is difficult to obtain the required performance on computing systems, and it is more complicated with the fast modifications and growing sophistication of modern computing systems. A deep comprehension of the underlying computing system and an appropriate implementation are required to exploit computation power. Some performance profiling tools soften this problem, for example, HPCToolkit or NVProf; but few guidelines to optimize the code are provided. Many high-quality HPC tools are referenced in [23]. Furthermore, a framework named Egeria is proposed. Egeria is an auto-synthesizer of advising tools for HPC, which provides users with a list of significant optimization guidelines. In one of the experiments, advising tools were generated for OpenCL programming on AMD GPUs.

C. ALVEO

The Xilinx Alveo U50 data center accelerator card [24] is a single slot, low profile form factor passively-cooled card. It supports PCI Express (PCIe) Gen3 ×16 or dual Gen4 ×8, is equipped with 32 units, an overall 8 GB of High-Bandwidth Memory (HBM), and Small Form-factor Pluggable (SFP) Ethernet for optical-fiber networking capability. The Alveo U50 is designed to accelerate memory-bound, compute-intensive applications. The U50DD is an Engineering sample card with two Double Density SFP (SFP-DD) interfaces instead of the Quad Small Form-factor Pluggable (QSFP) of the production card.

The design is based on OpenCL heterogeneous computing platform [25]. Using its Application Programming Interfaces (APIs), users can query the device, affect the PCIe transfer, make the kernel execute, . . . Xilinx Runtime Library (XRT) needs to be run on the host CPU. It is a critical component of the adaptable FPGA heterogeneous computing platform. Therefore, the HDL Intellectual Properties (IPs) can be executed in parallel and controlled by the CPU. In addition, an optimized parallel design can dramatically benefit the performance of the entire system.

Thanks to the reconfiguration capabilities of the Xilinx Alveo, combined with the network capabilities of the card, it is feasible to support not only legacy networking but high availability and deterministic Ethernet. Therefore, it can be used both in on-premises infrastructures of the SG or in the Internet cloud. Examples are the zero-delay recovery time protocols, Parallel Redundancy Protocol (PRP) —International Electrotechnical Commission (IEC) 62439-3-Clause 4—, and High-availability Seamless Redundancy (HSR) —IEC 62439-3-Clause 5— are used in the modern Digital Power substations to communicate Generic Object Oriented Substation Events (GOOSE) and SMV values. Additionally, TSN [26], the new generation Ethernet with support for real-time traffic, is also considered in the Sector. All these specific protocols are integrable on Alveo FPGA via IPs [27] and [28].

The Alveo acceleration card has two parts [29]: the static and the dynamic regions (see Fig. 2).

- The static region is the part of infrastructure offered by the system such as the Host Interface (HIF), Direct Memory Access (DMA), Clock, Reset and Isolation (CRI), Card Management Peripheral (CMP), Card Management Controller (CMC) and Embedded RunTime Scheduler (ERT).
- The dynamic region constitutes the area where the designer can implement the computing kernels.

The static region encompasses all the required hardware to make the data from the CPU available to the designer’s computing kernels. The DMA engine is in charge of the transactions between PCIe and HBM. Thus, making accessible the data from the computer’s main memory to the user-generated computing kernels. The static part also controls the overall system, generates clocks, resets, and other required signals.

The dynamic region houses the user-generated computing kernels. The Register-Transfer Level (RTL) kernel needs to read and write data to a specified HBM address to interchange data with the HBM banks. The address pointer is provided by an Advanced eXtensible Interface (AXI)-Lite control module that connects both the static and dynamic regions.

Table 1 depicts different HPC solutions based on heterogeneous computing deployed in Alveo accelerators.

III. TRADITIONAL APPROACH

A. NETWORK ARCHITECTURE

IEC 61850 [30] enables an interoperable digitalization of power substations and many Smart-Grid premises [31], [32]. It defines the data models, services, and communication protocols. Since 2004, when the 14 parts of the standard were
TABLE 1. HPC solutions based on heterogeneous computing deployed in Alveo accelerators.

| Reference | HPC application                                      | Hardware | Description                                                                 |
|-----------|-----------------------------------------------------|----------|-----------------------------------------------------------------------------|
| [15]      | Conjugate Gradient for Lattice Quantum Chromodynamics | U250     | Performance comparison to the standard CPU architectures (BRAM/DSP/FPLUT/URAM) |
| [17]      | Conjugate Gradient for Lattice Quantum Chromodynamics | U280     | A software package which implements the CG algorithm for FPGA-based accelerator cards which can serve as a backbone for many applications which are expected to gain a significant boost factor on FPGA accelerators. Performance and energy efficiency of FPGA cards for Computational Fluids Dynamic codes. Performance achieved for all the CFD kernels using a throughput expressed in GB/s and power dissipation using performance per watt metric. |
| [18]      | Computational Fluids Dynamic                        | U250     | Multiple algorithm-architecture co-optimizations. The training speed and accuracy evaluation based on three standard, large benchmark graphs. Execution time measurement to characterize the speedup in performance when compared to the software version of a tool used to perform genome alignment. The analysis of speedup and power consumption of the Geant4 code implementation on the FPGA. |
| [19]      | An accelerator for training Graph Convolutional Networks on CPU-FPGA systems | U200     |                                                                                     |
| [20]      | Genome Sequence analysis                             | U200     |                                                                                     |
| [21]      | Biological dose calculation in cancer therapy based on Monte Carlo calculations | U200     |                                                                                     |

released, most of the digitalization effort in the Electrical field has followed this approach.

IEC 61850 based substations are divided into three logical levels as shown in Fig. 1: station, bay, and process levels. The process bus interconnects the IED at the primary equipment level. On the other hand, the station bus connects IEDs at the bay level and station level. The devices at the process level are used to measure the status of the substation and actuate if required. The devices located at the bay level are used to control, monitor, and protect the substation. The station level provides remote access to the substation to allow remote configuration and supervision.

In this traditional approach, control is distributed. The digital frames, SMV, are typically publishing current and voltage values from the four phases (A, B, C, and N). The original application for the SMV was simplifying the cabling infrastructure and improving the availability at the Process Bus of Power Substations. In this context, the grid protection mechanisms are managed by means of the fault analysis of the current and voltage magnitudes. The IEDs do this analysis by means of the SMV frames broadcasted by the MU installed on the secondary of the transformers. A typical example of this setup is represented in Fig. 3. In these setups, the number of SMV streams broadcasted on the network usually is not high. Thus, the embedded computers of IED can handle them, fulfilling the tight real-time requirements enforced by IEC 61850 for Grid protection. Relying on the processing of the SMV done by the IEDs, they can actuate over the substation, opening a relay, for example.

The communication requirements in the process bus have very stringent time constraints due to the critical nature of the data. Layer-2 messages in the IEC 61850 provide services that require high-speed and low-delay message delivery. Specifically, these messages must be generated, transmitted, received, and processed by the receiver in less than 3 ms in the most exigent Power Substations, as shown in table 2. In order to reduce protocol overhead and achieve required performance levels, IEC 61850 defines that real-time services must be mapped directly to the Data Link layer (layer-2).

B. DRAWBACKS

This distributed approach was thought of in the context of digital substations, where the number of nodes to measure current and voltage was reduced. In the same way, the architecture of the digital substation was based on the traditional Station, Bay, and Process Level, very focused on control and automation processes only.

As presented in the Introduction to this paper, the complete digitalization and OT/IT convergence in the SG makes this traditional approach insufficient to face new applications exploiting the massive use of data combined with the real-time operation.

The main limitations are constrained computing and networking capabilities of IEDs, low data bandwidth of the in-premises network infrastructures, few scalability and flexibility, cost, and security aspects [10].

In this traditional approach, control is distributed. The digital frames, SMV, are typically publishing current and voltage values from the four phases (A, B, C, and N). The original application for the SMV was simplifying the cabling infrastructure and improving the availability at the Process Bus of Power Substations. In this context, the grid protection mechanisms are managed by means of the fault analysis of the current and voltage magnitudes. The IEDs do this analysis by means of the SMV frames broadcasted by the MU installed on the secondary of the transformers. A typical example of this setup is represented in Fig. 3. In these setups, the number of SMV streams broadcasted on the network usually is not high. Thus, the embedded computers of IED can handle them, fulfilling the tight real-time requirements enforced by IEC 61850 for Grid protection. Relying on the processing of the SMV done by the IEDs, they can actuate over the substation, opening a relay, for example.

The communication requirements in the process bus have very stringent time constraints due to the critical nature of the data. Layer-2 messages in the IEC 61850 provide services that require high-speed and low-delay message delivery. Specifically, these messages must be generated, transmitted, received, and processed by the receiver in less than 3 ms in the most exigent Power Substations, as shown in table 2. In order to reduce protocol overhead and achieve required performance levels, IEC 61850 defines that real-time services must be mapped directly to the Data Link layer (layer-2).

B. DRAWBACKS

This distributed approach was thought of in the context of digital substations, where the number of nodes to measure current and voltage was reduced. In the same way, the architecture of the digital substation was based on the traditional Station, Bay, and Process Level, very focused on control and automation processes only.

As presented in the Introduction to this paper, the complete digitalization and OT/IT convergence in the SG makes this traditional approach insufficient to face new applications exploiting the massive use of data combined with the real-time operation.

The main limitations are constrained computing and networking capabilities of IEDs, low data bandwidth of the in-premises network infrastructures, few scalability and flexibility, cost, and security aspects [10].

FIGURE 3. Distributed architecture based on IED and MU in a digital Power Substation.
IV. HPC APPROACH
A. NETWORK ARCHITECTURE
The HPC approach wants to simplify the outer nodes and put all the computing capabilities in a single more powerful node [10]. Fig. 4 depicts a simplified diagram of a possible network architecture.

![Network diagram](image)

**FIGURE 4.** The network structure for SV HPC infrastructure.

In this scenario, sensors and actuators are dummy devices without intelligence. Instead, they send information or receive commands to operate their actuators. Network infrastructure is in charge of channeling the information towards the accelerator-enabled server. The decision-making is done in a single point, and the result is channeled back to the actuators.

B. HARDWARE ARCHITECTURE
There has always been difficulty developing FPGA equipment because the method is entirely different from other programs. Moreover, its development tools are not efficient for higher-level applications. To solve this problem, Xilinx provides a new application acceleration framework. They packaged the necessary technical infrastructure in the Alveo FPGA accelerator into a static region of the system. For example, on-board HBM memory controller, PCIe bridge controller, system clock…Such programs based on C or C++ language can be more conveniently run on the FPGA accelerator after being compiled by High-Level Synthesis tool (HLS) and automatically connect the system resources of the static area.

1) EXPERIMENTAL SETUP
Our experimental hardware design obtains source data from the main CPU, the Sample Valued data set (see Fig. 5). The data is obtained using a program that generated IEC 61850 traffic. This data is then sent to the Alveo accelerator through the PCIe bridge. As the Alveo accelerator has two SFP ports, these data sets are sent from the Tx of each SFP optical communication interface to the Rx of the other SFP interface by a loopback wiring. The loopback wiring means

![Experimental setup diagram](image)

**FIGURE 5.** Overall experimental setup.

to plug the SFP cable from SFP port 0 to port 1 on the same device. Each SFP interface opens four lanes with a total of 10 Gbps. The two SFP interfaces have a total theoretical throughput of 20 Gbps. When the SFP interface receives data from the Rx, it will temporarily store the data in the HBM memory associated with it. Under the control of the OpenCL API, these data are copied to the HBM memory bank associated with the idle SV processing cores. Then the SV processing core will process these data and send the result back to the main CPU through the PCIe bridge. In addition, a channel for sending SV core status to the main CPU is also established.

2) SYSTEM ARCHITECTURE
An Alveo system is composed of a PC and an Alveo card. In our case, the system design is based on an Intel Core i7-9700K CPU and an Alveo U50DD FPGA accelerator card. Our research project is not based only on HLS. On the one hand, only specific algorithms can be deployed in HLS and cannot activate the SFP optical communication controller. On the other hand, using the SV processing core directly deployed on the silicon chip, designed using HDL, can achieve higher performance. To achieve this, we must manually deploy the relevant RTL kernel in the dynamic area of the Alveo accelerator and design the necessary system ports, clocks, reset, and memory controller connections in advance which are connected with the static region’s ports. Additionally, generate a script file for the system tools to know how to connect to the kernels. The SFP optical communication core is also packaged into an RTL core. Therefore, an optimized architecture must be created to obtain a better performance, larger throughput, and lower latency. An overall block diagram of the design is depicted in Fig. 6.

As it can be seen, in the dynamic region of the Alveo card, two sets of kernels exist:
- SFP Kernel: It is in charge of receiving the data from the SFP-DD and storing it in HBM.
- SV computing kernel: It is in charge of reading data from the HBM and feeding it to the SV IP Core.

3) KERNEL ARCHITECTURE
The SVs processing core RTL kernel was designed using a traditional approach. In this case, the computing element is the SV Processing Core. Its primary interfaces are built using AXI: AXI4–stream to transfer data and AXI-Lite for configurations. So two HBM bank connection buses need to
be implemented: One for the data transfer and the other for configuration and status transfer. The overall structure can be seen in Fig. 7.

The MicroBlaze CPU is in charge of the overall control and software processing of the outgoing data. It will receive configuration information from one channel and configure and initialize the SV Processing Core. Then, in every kernel execution, this onboard CPU reads and stores the status of the SV Processing Core into the HBM1. The CPU also performs data processing of the final results. When the CPU’s works finish, the status data will be transfer back to the specified address of HBM. Meanwhile, a configuration interface finish signal will be generated.

When both the configuration and data interface finish signals have been generated, the entire kernel execution is considered finished.

The finishing mark will be reflected on the AXI-Lite control module so that the host application will be conscious of the end of the execution of the kernel. A new execution will be performed if there is new data to be processed.

On the other hand, the SFP kernel allows data to enter the system through the SFP ports. Due to the constraints imposed by the Alveo design, the Ethernet Subsystem cannot be directly accessed by other kernels. The Ethernet Subsystem needs to be packaged as an RTL kernel. Then it must be deployed into the device dynamic region and it will make use of the HBM to interchange information. As the Ethernet Subsystem requires a high-speed data connection, a wide and high-speed AXI4 connection to the HBM bank is necessary.

Two —out of four— SFP transfer cores lanes are enabled, each of them configured for a 10 Gbps connection of each SFP port. We are limited to two lanes because the Alveo U50DD card we are using is an Engineering sample and has two SFP instead of a QSFP — Single SFP vs. Quad SFP. The SFP kernel structure is depicted in Fig. 8.

The control operation of kernels, the data transmission between PCIe, and the HBM memories of the accelerator are all controlled by the OpenCL framework. The above steps for the accelerator’s initialization process are described next. First, the system retrieves the list of Alveo accelerator devices in the system through clGetPlatformIDs(). Then, it obtains the cl_device_id instance of the device available in the system and uses clCreateContext() to create a context for each device to store the command list. It uses clCreateCommandQueue() to create a command list for each accelerator device. The following steps are used to program the xclbin file into the accelerator using clCreateProgramWithBinary(). The xclbin is the firmware that contains the FPGA core design. Buffer instances are created with clCreateBuffer() in the system memory to store the data to interact with the accelerator. The connection for each kernel interface is set and mapped with buffers using clSetKernelArg(). At this moment, clEnqueueWriteBuffer() can be called to transfer data with the accelerator, but only the data is transferred, and the kernel is not started. clEnqueueTask() must be called to run the kernel. Before the completion of the kernel operation signal is sent, the program will block in clFinish(). If the kernel runs smoothly, its data can be safely read through clEnqueueReadBuffer(). The above process completes the control of a single core (see List 1).

C. RESULTS
As mentioned above, the AXI_lite controller connected to the static area will control the RTL core to start, stop,
LISTING 1. Control process of a single core.

```c
clGetDeviceInfo(devices[i], CL_DEVICE_NAME, 1024, cl_device_name, 0);
context = clCreateContext(0, 1, &device_id, NULL, NULL, &err);
commands = clCreateCommandQueue(context, device_id,
    CL_QUEUE_PROFILING_ENABLE | CL_QUEUE_OUT_OF_ORDER_GLSFENCE_MODE_ENABLE,
    &err);
program = clCreateProgramWithBinary(context, 1, &device_id, &mem, (const unsigned char **)&kernelbinary, &status, &err);
kernel_svm[i] = clCreateKernel(program, "svm_core", &err);
cSetKernelArg(kernel_svm, 0, sizeof(cl_mem), &svm_sfc0_ptr0);
sfm_axi00_ptr0 = clCreateBuffer(context,
    CL_MEM_READ_WRITE | CL_MEM_EXT_PTR_XILINX, DATA_BU0, gLs[FF]ER_LENGTH_BYTE,
    &mem_ext_sfp, &err);
err = clEnqueueWriteBuffer(commands, sfm_axi00_ptr0, CL_TRUE, 0,
    DATA_BU0, gLs[FF]ER_LENGTH_BYTE, b_data, 0, NULL, NULL);
cFinish(commands);
cEnqueueReadBuffer(commands, sfm_axi00_ptr0, CL_TRUE, 0, 16384, sfm__output, 0, NULL, &sfm_read_finish_event_1);
cReleaseProgram(program);
```

Because there are many identical SV processing cores in the system, in order to increase their parallelism, it is not necessary to check their completion status every time they are completed. The parallelism will increase system throughput and reduce latency. Nevertheless, they need to be detected and synchronized at an appropriate time to avoid errors. So in Fig. 10, there is a gap in the queue where many kernels are running. It means that the system is waiting for all the kernels to complete to synchronize. These idle states will waste running time, but it is necessary.

1) TIMING
The different stages in the system execution are (see Fig. 10):
1) Reception of data to the SFP.
2) Execution of the SFP kernel.
3) Transfer of data from the SFP kernel to the SV processing kernels through HBM.
4) Execution of all the SV processing kernels.
5) Gathering the results by the PC from every SV processing kernel using the PCIe through HBM.

From here, we can make the stages execute in parallel with the OpenCL APIs. The key to improve the performance is to make kernel usage as high as possible. After optimizing the control program, the performance of the accelerator overall has increased at least 50%.

In this design, the kernel read/write was included in the kernel’s execution, and all the tasks spend the same magnitude of time; therefore, they can run in parallel more conveniently. All the above processes need to be parallel to make the system more efficient. The higher the task usage, the better performance gain. Therefore, multiple buffers were implemented to apply the parallel execution. In some cases, the kernel execution can be faster than their data transfer in the parallel execution. This discrepancy may cause a read-before-write error which will lead the application to hang. The solution should be to synchronize the transfer and execution tasks during some safe execution queues. As shown in Fig. 10, there is a queue blank for synchronization.

In table 3, we can see how the throughput is rising by incrementing the Computing Units (CUs). Once we have 7 SV processing units enabled, the increase in throughput is almost negligible, and the total kernel usage comes to a standstill. In our case, the bottleneck is the bandwidth of SFP port communication. Notice that the U50DD is an engineering sample card and has limited SFP connectivity. The latency is calculated using the following formula:

$$\text{Latency} = \frac{t_{\text{CPU}} - t_{\text{Result}}}{N}$$

(1)
TABLE 3. Timing results from different implementations. SFP and SV kernel usage as well as total SV usage (taking into account the number of CUs).

| CU  | Throughput (Gbit s⁻¹) | Latency (ms) | SFP (%) | SV (%) | Total SV (%) |
|-----|-----------------------|--------------|---------|--------|-------------|
| 2   | 4.96                  | 1.69         | 21.45   | 93.08  | 186.16      |
| 4   | 10.06                 | 1.67         | 43.00   | 89.43  | 357.72      |
| 6   | 13.38                 | 1.88         | 54.10   | 82.57  | 495.42      |
| 7   | 13.97                 | 2.10         | 56.19   | 71.86  | 503.02      |
| 8   | 14.17                 | 2.36         | 56.01   | 63.02  | 504.16      |
| 10  | 14.28                 | 2.93         | 56.03   | 51.01  | 510.10      |
| 12  | 14.30                 | 3.50         | 56.10   | 42.60  | 511.20      |

where $t_{CPU}$ is the time of the first CPU to SFP transfer, $t_{Result}$ is the time of the last result transfer to the CPU, and $N$ is the number of execution times.

FIGURE 11. Graphical representation of the relationship between CUs and throughput.

Fig 11 depicts the relationship between CU and the throughput. Once the SFP interface cannot provide more data, the SV usage drops, and the throughput remains constant. The throughput is calculated using the following formula:

$$\text{Throughput} = \frac{D}{t_{CPU} - t_{Result}}$$

where $D$ is the amount of data sent through the SFP, $t_{CPU}$ is the time of the first CPU to SFP transfer, and $t_{Result}$ is the time of the last result transfer to the CPU.

If we focus our analysis on the microgrids in a small region, assuming 80 measurement points in each DER [34] (for a microgrid of 500 kW), a typical SV frame of 156 B, a signal frequency of 50 Hz and 40 samples per cycle, the throughput of each DER can be calculated as:

$$\text{DER}_{Th} = 156 \text{ B} \cdot 50 \text{ Hz} \cdot 80 \cdot 40 = 24.960 \text{ Mbit/s}$$

In this scenario, the proposed architecture is capable of supporting more than 500 DERs:

$$N = \frac{\text{Throughput}}{\text{DER}_{Th}} = \frac{14.3 \text{ Gbit/s}}{24.960 \text{ Mbit/s}} = 572.9$$

In this example, the total power managed by our solution would be around 280 MW (572.9 \times 500 kW).

2) AREA

The system utilization with One SFP kernel and 12 SV kernels is shown in table 4. For every execution, the transfer size is 512 kB for all the kernels. The transfer size will affect the resource utilization, especially that of BRAM and URAM.

TABLE 4. Utilization report in percentage.

| Resource | SLR0 (%) | SLR1 (%) |
|----------|----------|----------|
| LUT      | 43       | 45       |
| FF       | 38       | 34       |
| BRAM     | 65       | 64       |
| URAM     | 28       | 46       |

The limiting resources are control sets and BRAMs. If the usage of control sets is increased, the timing may fail. The CUs have to be implemented in one of the SLRs. Because the FPGA is divided into two SLRs, it makes more sense to examine the resource usage by SLR.

The system consumes resources from both the static region, which contains system infrastructures, and the dynamic one containing all the CUs.

The clock used for AXI memory read and write, and the DMA engine was limited to 300 MHz. The kernel clock used for the SV processing units was limited to 350 MHz. The HBM bank clock was set to 450 MHz, which is a typical clock speed. Under these frequencies, the entire load power of the FPGA processor is 58 W. The power cost is in the order of 5 times lower compared to a CPU/GPU based accelerator.

V. COMPARISON

In this section, we will compare the approach in this paper with others present in the literature. SV processing system is usually implemented by software running on CPU or hardware on an FPGA or ASIC. All cores are designed as dispersed models in order to meet the standards of heterogeneous computing. Moreover, doing so can easily control and expand the number of cores. We have also considered designing SV and SFP cores together, and the disadvantage of a massive core is that the system lacks scalability and maintainability. In order to explore the limits of SV processing cores that can be deployed in one Alveo accelerator, this dispersed model is also meaningful. Table 5 shows the results for different approaches. Hardware implementation architectures typically have better throughput and latency. Comparing this work with the FPGA SoC like Zynq-7020 [35] based architecture, our system’s latency is higher. A similar design can also be used to deploy the SFP controller in an FPGA chip with many programmable logic resources on one chip and then make a custom circuit board with SFP ports [36]. Nevertheless, the difficulty of development and the investment of resources is much more significant in this way. Another option can also use the traditional CPU to connect to the PCIe SFP communication card and a software method to achieve similar results. The critical point in this solution is the computing capability of the CPU and the resulting latency issues. Because our approach is based on heterogeneous computing,
our structure is more complicated. Our latency is still much lower than the 3 ms requirement of the standard. Additionally, due to the SFP optical connection and the formidable computing power of the Alveo accelerator, its throughput is hugely higher than others.

VI. CONCLUSION
This article shows a novel hardware architecture that uses the new FPGA adaptive computing design framework and brings the advantages of traditional FPGA design. Finally, the performance of the entire system and its power consumption is very satisfactory. On the other hand, this FPGA accelerator-based hardware system can quickly expand its scale, such as incorporating more accelerators on other PCIe interfaces quickly, reflecting its highly adaptable characteristics. Such a system is highly efficient, energy-saving, scalable, maintainable, and has moderate development difficulties. The entire system can process as high as 14 Gbit/s IEC 61850 SV data with a latency as low as 0.84 ms, while, at the same time, consuming only 58 W. This article presents a novel architecture, but its implementation has led to new challenges. The future work includes improving the latency by further pipelining the executions kernels as well as improving the PC ↔ Alveo communications.

REFERENCES

[1] Y. Zhao, M. Li, Y. Zhang, Q. Lin, and Z. Chen, “Research on FPGA timing optimization methods with large on-chip memory resource utilization in PCIe DMA,” in Proc. CIE Int. Conf. Radar (RADAR), Oct. 2016, pp. 1–4, doi: 10.1109/radar.2016.8059429.

[2] A. Ghasempour, Advanced Metering Infrastructure in Smart Grid: Requirements, Challenges, Architectures, Technologies, and Optimizations (Energy Science, Engineering and Technology). Annandale, VA, USA: Nova, Oct. 2017, pp. 77–127.

[3] M. Ghorbanian, S. H. Doldabadi, M. Masjedi, and P. Siano, “Communication in smart grids: A comprehensive review on the existing and future communication and information infrastructures,” IEEE Syst. J., vol. 13, no. 4, pp. 4001–4014, Dec. 2019, doi: 10.1109/jsyst.2019.2928090.

[4] Fault Detection on Cable Sections in Mixed High Voltage. Accessed: Nov. 10, 2021. [Online]. Available: https://www.arteche.com

[5] P. Hovila, A. Monot, and H. Laaksonen, “Advanced utilization of big data for real-time monitoring and data analytics in sundom smart grid,” in Proc. 25th Int. Conf. Electr. Distrib., 2019, pp. 1–5.

[6] ENTSO Europe. (2019). ENTSO-E Area Wind Europe Report 2019. [Online]. Available: https://www.entsoe.eu

[7] S. Bera, S. Misra, and J. J. P. C. Rodrigues, “Cloud computing applications for smart grid: A survey,” IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1477–1494, May 2015, doi: 10.1109/tpds.2014.2321378.

[8] S. N. Kumar and A. Vajpayee, “A survey on secure cloud: Security and privacy in cloud computing,” Am. J. Syst. Softw., vol. 4, no. 1, pp. 14–26, 2016. [Online]. Available: http://pbs.sciencepubsys.com/ajss/4/1/2

[9] M. H. Y. Moghaddam and A. Leon-Garcia, “A fog-based internet of energy architecture for transactive energy management systems,” IEEE Internet Things J., vol. 5, no. 2, pp. 1055–1069, Apr. 2018, doi: 10.1109/jiot.2018.2805899.

[10] N. Chaichi, J. Lavoie, S. Zarrin, R. Khalifa, and F. Sie, “A comprehensive assessment of cloud computing for smart grid applications: A multi-perspectives framework,” in Proc. Portland Int. Conf. Manage. Eng. Technol. (PICMET), Aug. 2015, pp. 2541–2547, doi: 10.1109/picmet.2015.7273227.

[11] W. Ahmed, “Evaluating high performance computing (HPC) requirements of devices on the smart grid for increased cybersecurity,” in Proc. 2nd Int. Conf. Anti-Cyber Crimes (ICACC), Mar. 2017, pp. 103–107, doi: 10.1109/anti-cybercrime.2017.7905722.

[12] R. C. Green, L. Wang, and M. Alam, “Applications and trends of high performance computing for electric power systems: Focusing on smart grid,” IEEE Trans. Smart Grid, vol. 4, no. 2, pp. 922–931, Jun. 2013, doi: 10.1109/tsg.2012.2225646.

[13] R. C. Green, L. Wang, and M. Alam, “High performance computing for electric power systems: Applications and trends,” in Proc. IEEE Power Energy Soc. Gen. Meeting, Jul. 2011, pp. 1–8, doi: 10.1109/pes.2011.6039420.

[14] T. Nguyen, C. MacLean, M. Siracusa, D. Doerfler, N. J. Wright, and S. Williams, “FPGA-based HPC accelerators: An evaluation on performance and energy efficiency,” Concurrency Comput., Pract. Exper., vol. 2021, Aug. 2021, Art. no. e6570, doi: 10.1002/cpe.6570.

[15] G. Korcyl and P. Korcyl, “Towards lattice quantum chromodynamics on FPGA devices,” Comput. Phys. Commun., vol. 249, Apr. 2020, Art. no. 107029, doi: 10.1016/j.cpc.2019.107029.

[16] N. Fujita, R. Kobayashi, Y. Yamaguchi, T. Ueno, K. Sano, and T. Boku, “Towards lattice QCD on FPGA,” in Proc. IEEE Int. Symp. on FPGA, Apr. 2020, pp. 450–459, doi: 10.1109/ispf.2020.00083.

[17] G. Korcyl and P. Korcyl, “Optimized implementation of the conjugate gradient algorithm for FPGA-based platforms using the Dirac-Wilson operator as an example,” 2020, arXiv:2001.05218.

[18] K. Rojec, K. Haliabina, and L. Kuczynski, “CFF code adaptation to the FPGA architecture,” Int. J. High Perform. Comput. Appl., vol. 35, no. 1, pp. 33–46, Nov. 2020, doi: 10.1177/1094342020972461.

[19] A. Zeni, M. Crespi, L. Di Tucci, and M. D. Santambrogio, “An FPGA-based computing infrastructure tailored to efficiently scaffold genome sequences,” in Proc. IEEE 27th Annu. Int. Symp. Field-Program. Custom Comput. Mach. (FCCM), Apr. 2019, p. 33, doi: 10.1109/fccm.2019.00074.

[20] A. Zeni, M. Crespi, L. Di Tucci, and M. D. Santambrogio, “An FPGA-based computing infrastructure tailored to efficiently scaffold genome sequences,” in Proc. IEEE 27th Annu. Int. Symp. Field-Program. Custom Comput. Mach. (FCCM), Apr. 2019, p. 33, doi: 10.1109/fccm.2019.00074.

[21] Y. Abhyankar, S. Dev, O. Sarun, A. Saxena, and R. Joshi, “Monte Carlo processing on a chip (MCoaC) preliminary experiments toward the realization of optimal-hardware for TOPAS/Geant4 to drive discovery,” Phys. Medica, vol. 64, pp. 166–173, Aug. 2019, doi: 10.1016/j.fccm.2019.06.016.

[22] H. Khaledizadeh, M. Fahad, A. Shahid, R. R. Manumachu, and A. Lastovetsky, “Bi-objective optimization of data-parallel applications on heterogeneous HPC platforms for performance and energy through workload distribution,” IEEE Trans. Parallel Distrib. Syst., vol. 32, no. 3, pp. 543–560, Mar. 2021, doi: 10.1109/tpds.2020.3027338.

TABLE 5. Comparison of the proposal with those present in the literature.

| Reference | Implementation | Max. Latency (ms) | Delivery Time Usage (%) | Max. Throughput | Fixed Latency |
|-----------|----------------|-------------------|------------------------|----------------|---------------|
| [37] SW   | 4,000,000      | 133,000           | -                      | No             |               |
| [37] HW   | 1,917,000      | 63,900            | -                      | Yes            |               |
| [38] SW   | 6,000,000      | 200,000           | -                      | No             |               |
| [39] SW   | 942,000        | 31,400            | -                      | No             |               |
| [39] SW   | 3,560,000      | 118,700           | -                      | No             |               |
| [40] SW   | 242,000        | 8,070             | -                      | No             |               |
| [35] HW   | 0.006          | 0.230             | >1 Gbit s⁻¹            | Yes            |               |
| This work | HW             | 0.840             | 26,000                 | No             | >14 Gbit s⁻¹   |
LEIRE MUGUIRA received the Ph.D. degree in telecommunications engineering from the University of the Basque Country (UPV/EHU), Spain, in 2015. In 2018, she started with UPV/EHU. She has participated in 11 research projects and three research contracts. She is the author or coauthor of a patent, seven JCR articles, a book chapter and 23 papers in scientific conferences.

JAIME JIMÉNEZ received the M.S. and Ph.D. degrees from the University of the Basque Country (UPV/EHU), in 1991 and 2005, respectively. He has participated in 45 competitive research projects supported by public institutions and 39 private research contracts. He is the author or coauthor of 26 articles in scientific international journals. His research interests include high-speed circuits on reconfigurable devices and communications devices.

ARMANDO ASTARLOA received the M.Sc. and Ph.D. degrees in electrical engineering from the University of the Basque Country (UPV/EHU), Spain, in 1999 and 2005, respectively. After several years of developing his professional career as a Hardware Engineer in the private sector, in 2001, he joined the staff of the Faculty of Engineering of Bilbao, UPV/EHU, as a full-time Researcher and a Lecturer. He is currently a Founding Member of the Research Applied Electronics Research Team (APERT). He is the author of dozens of technical contributions in international scientific journals, conferences, book chapters, and patents. His research interests include reliable electronic design for reconfigurable devices, industrial networking, and cybersecurity. He collaborates as a Committee Member in IEC working groups. In 2008, he enrolled with the Institute of Microelectronics and Wireless Systems, Ireland, as a Visiting Researcher. In 2010, he launched System-on-Chip engineering S.L. (SoC-e.com) Business Project, and he held the CEO position at this company, from 2016 to 2020. In 2021, he achieved a full professor position with UPV/EHU.

JESÚS LÁZARO received the M.S. and Ph.D. degrees in telecommunications engineering from the University of the Basque Country (UPV/EHU), Spain, in 2001 and 2005, respectively. In 2001, he started with the Telecommunications Department, University of the Basque Country, as a Researcher and a Lecturer. Since 2019, he has been a Full Professor. In 2010, he was a Research Visitor with the Configurable Computing Laboratory, Virginia Tech. He is currently a member of the Applied Electronics Research Team and the Co-Founder and an Entrepreneur of System-on-Chip engineering S.L. (SoC-e.com). He has participated in 50 competitive research projects supported by public institutions and in 47 private research contracts, in 21 of them as the Leading Researcher. He is the author or coauthor of four patents, 31 articles in international scientific magazines, and 76 papers in international conferences. His research interests include high-speed circuits on reconfigurable devices and communications devices.

LE SUN received the M.S. degree in photovoltaic systems from the University of the Basque Country, Spain, in 2014. He is currently pursuing the Ph.D. degree in high-speed circuits based on reconfigurable devices and communications devices with the Department of Electronics Technology, UPV/EHU.

F. Andren, R. Brundlinger, and T. Strasser, “IEC 61850/61499 control of differential and integrity in IEC 61850 GOOSE messages,” IEEE Trans. Power Del., vol. 35, no. 5, pp. 2565–2567, Oct. 2020, doi: 10.1109/tpwd.2020.2990760.

V. Chikhale, N. Wankhede, and S. Hande, “Development of optical communication framework with Xilinx gigabit transceivers,” in Proc. Int. Conf. Nascent Technol. Eng. (ICNTE), Jan. 2019, pp. 1–7, doi: 10.1109/icnte.2019.8954995.

R. Aguilar and J. Ariza, “Testing and configuration of IEC 61850 multi-vendor protection schemes,” in Proc. PES T&D, Apr. 2010, pp. 1–8, doi: 10.1109/hdc.2010.5484210.

M. Rodriguez, J. Lazarro, U. Bidarte, J. Jimenez, and A. Astarloa, “A fixed-latency architecture to secure GOOSE and sampled value messages in substations systems,” IEEE Access, vol. 9, pp. 51646–51658, 2021, doi: 10.1109/access.2021.3069088.

S. M. Farooq, S. M. S. Hussain, and T. S. Ustun, “Performance evaluation and analysis of IEC 62351-6 probabilistic signature scheme for securing GOOSE messages,” IEEE Access, vol. 7, pp. 32343–32351, 2019, doi: 10.1109/access.2019.29002571.

S. Hussain, S. Farooq, and T. S. Ustun, “A method for achieving confidentiality and integrity in IEC 61850 GOOSE messages,” IEEE Trans. Power Del., vol. 35, no. 5, pp. 2565–2567, Oct. 2020, doi: 10.1109/tp-wrd.2020.2990760.