System-on-Chip FPGA Devices for Complex Electrical Energy Systems Control

I. Introduction

DIGITAL electronics has become a standard for controlling electrical systems. This is due to the constant improvement of the digital devices, whether in terms of density, performance, flexibility of use or cost reduction [1]. This paper looks into System-on-Chip (SoC) Field Programmable Gate Array (FPGA) for controlling complex electrical energy systems. These devices encompass multicore floating point microprocessors embedded with standard peripherals together with an FPGA fabric that allows the design of custom peripherals and specific hardware accelerators. Thus, SoC FPGA devices can be regarded as a good compromise between "super" microcontrollers (very fast in terms of computation but with a fixed micro-architecture) and pure FPGAs (ideal for specific concurrent micro-architectures but limited in terms of density).

SoC FPGA architectures are discussed and compared with state-of-the-art DSP-controllers, since they can also be qualified a s S oC d evices a s t hey a re i ntegrating fl oating po int mi croprocessor co res and substantial peripherals. The main differences between these two groups of devices lies in the opportunity offered to the designer by the SoC FPGAs to customize the SoC device via its internal FPGA fabric. Two case studies demonstrate that with SoC FPGAs one can go beyond standard control by introducing new auxiliary functions that enhance market competitiveness. The first a pplication c oncerns a f uel cell hybrid electric system controlled by passivity-based power management associated with an aging prognosis algorithm. For this application, it is shown that the time and cost constraints justify the use of a soft processor core to implement the controller.

The second application concerns the maximization of the electrical power production of a PhotoVoltaic (PV) field operating in mismatched conditions through the dynamical reconfiguration of the PV modules. This application allows to illustrate the ability of SoC FPGA to solve a complex optimization problem in a time that is so short that the PV field operating conditions can be considered as constant. Secondly, it shows the benefits of implementing C/C++ High Level Synthesis-based (HLS) hardware accelerators by significantly simplifying the design space exploration phase.

Finally, to generalize the lessons learned from these, an analysis of recent and inspiring controllers for complex electrical energy systems is presented from which key principles for designing the next generation of SoC FPGA-based smart controllers are derived.

II. Embedded Digital Controllers and System-on-Chip - Evolution and Trends

Due to their ability to execute control algorithms of ever increasing complexity in a very short time, using cheap components, digital controllers took preference over the analogue ones. Microcontrollers and DSPs are used [2], however, FPGA-based controllers also have some advantages [3]. DSPs and microcontrollers are flexible (C-based p rogramming), l ow c ost, a nd w ith a h ighly-performing floating point Arithmetic Logic Unit (ALU). DSP controllers integrate a high number of peripherals, all well fitting w ith t he c ontrol o f p ower e lectronics a nd d rives. T he m ain d isadvantage o f s uch d evices i s that

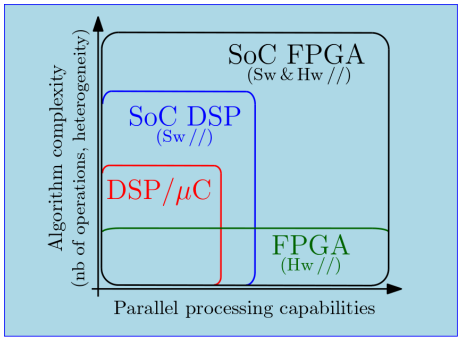

Fig. 1: Ability for each device technology to handle the algorithm complexity and concurrency

they are based on a fixed micro-architecture which prevents to concurrently execute tasks that could be executed in parallel. This significantly limits their timing performance, leading to the introduction in the controller of one sampling period delay that reduces the control system's bandwidth and introduces more chattering into direct control of power converters.

Initially designed as a simple fabric of lookup tables and flip-flops, FPGAs have then integrated DSP units and memory banks and lately the end user has been able to easily synthesize 32-bit RISC processors within the FPGA fabric [4]. FPGAs are attractive for controlling industrial systems mainly because they are allowing the design of dedicated controllers that are the 'hardware copies' of the source control algorithms, thus including the entire potential parallelism of these algorithms and, as a consequence, accelerating significantly their real-time executions. FPGAs can also handle the control of systems with a high number of I/Os, such as multilevel converters. Indeed, the parallelism can be inside the control algorithm but it can also be intrinsic to the system to be controlled, like for multiphase motors. As no additional delay is introduced, the FPGA-based controller increases the bandwidth of the designed control loops, thus being ideal for the direct control of power converters [3], including power electronics using the recently introduced wide band-gap power switches that are commonly driven with a switching frequency above 100kHz [5]. Computational demanding algorithms like Model Predictive Control (MPC) are also good

candidates for FPGA-based implementations because of their parallelized and highly pipelined architecture [6]. The main drawbacks of FPGAs are the lack of performing internal ADCs and limited size, making floating point arithmetic architecture design problematic. However, Intel-FPGA has introduced 32-bit floating-point DSP units [7].

System-on-chip devices were introduced around a decade ago, mainly due to the benefits brought to mobile phones and, more recently, to the Internet-of-Things [8]. They also impacted control applications because of their impressive computational power; the parallelism of the computing tasks can also be obtained by running several tasks simultaneously on different processor cores, with the possibility to also embed a real-time Linux operating system. Thus, SoC can help expand the domain of traditional control algorithms (Fig. 1) and brings convergence between the worlds of DSP controllers and FPGAs.

The TI Dual Delfino device [9] (see Fig. 2.a) represents a natural SoC evolution of traditional DSP controllers. It is based on dual 32-bit floating-point DSP cores, with always more peripherals and dedicated arithmetic units like VCU (Viterbi, Complex Math unit) and TMU (Trigonometric Math Unit), which can be regarded as specific hardware accelerators (Fig. 2.a). With the TMU, a Park's transformation can be executed in about 100ns, comparable to what can be achieved with an FPGA. Also, parallel computing

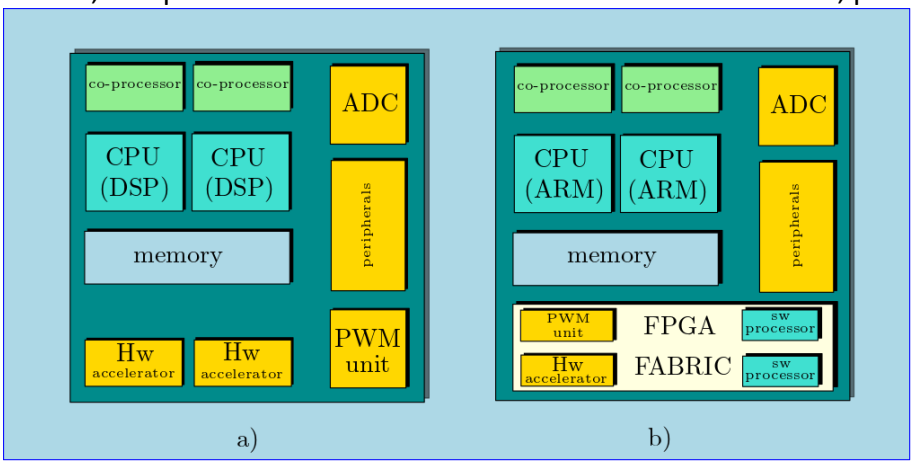

Fig. 2: Digital control architectures: (a) SoC DSP-based controllers, (b) SoC FPGA controllers.

is now possible since 4 tasks can be executed simultaneously, one on each DSP core and one in each of the two Control Law Accelerators (CLA) cores. So, its clock frequency is 200MHz but as it is a multicore architecture it can reach up to 800 MIPS. CLAs alleviate the DSP cores of low level but very time constrained tasks, like an FPGA current/voltage controller would do. Most IGBT-based inverter switching control functions in the 10 kHz frequency range can therefore be achieved.

SoC FPGAs (Xilinx Zynq, Intel-FPGA Arria 10 or Cyclone V devices [7]) include a dual core ARM A9 along with powerful coprocessors like the Single Instruction Multiple Data (SIMD) NEON, a set of peripherals to communicate with other boards and high performing FPGA fabric (Fig. 2.b). The latter offers the designer the possibility to add custom peripheral and/or specific h ardware accelerators adapted to a given

application. The 32-bit ARM A9 microprocessors are intended to run a powerful OS like embedded Linux. However, these processors can also be used for bare metal applications that are more adapted to standard control solutions for electrical systems. Running at 667MHz, they feature high computing power and a high quality internal bus, used for controlling either a simple peripheral via its internal registers or for exchanging a stream of data at high rate with the processors, an FPGA-based hardware accelerator [10]. SoC FPGA components can easily implement 32-bit RISC processor cores within the FPGA fabric (Microblaze for Xilinx, Nios II for Intel-Altera and ARM Cortex M1 or M3 [11]). These features offer huge flexibility to the designer who can, thanks to the FPGA fabric, integrate specific peripherals and/or hardware accelerators plus additional 32-bit RISC processor cores into the SoC architecture. Table 1 summarizes pros and cons of the different types of SoC devices.

III. CASE STUDIES

A. Control, estimation and prognosis of an hybrid fuel cell system

Because of their large number of components like PEM fuel cells, electrolysers associated with hydrogen tanks for long term storage [12], PV arrays and power converters, and because of emerging possibilities in terms reinforcement of reliability offered by multi-stack fuel cells and interleaved converters [13], modern fuel cell hybrid power systems can be considered as very complex. To cope with this complexity, controllers are rapidly evolving by including always more new functionalities such as power sharing [14] impedance spectroscopy for data-based diagnosis [15], prognosis and fault system control [16], as well as weather and power consumption forecasting.

TABLE 1 – ADVANTAGES AND DISADVANTAGES OF THE DIFFERENT DIGITAL TECHNOLOGIES

CRITERIA	DSP/µC	FPGA	SoC DSP	SoC FPGA
ALGORITHMIC PERSPECTIVE	●○○	●○○	●●○	●●●
Algorithm complexity management	●○○	●○○	●●○	●●●
Rapidity, possibility of parallelism	●○○	●●○ *(Hw parallelism: concurrency & pipeline)*	●●○ *(Multi cores Sw parallelism)*	●●● *(both Hw & Sw parallelism)*
Accuracy (floating point capability)	●●●	●●○	●●●	*Sw:* ●●● *Hw:* ●●○
CONNECTIVITY	●○○	●○○	●●○	●●○
Analog interface (ADC, DAC)	●●○	●○○	●●○	●○○
Digital interface (number of I/O)	●●○	●●●	●●○	●●●
Embedded peripherals	●●○	●●○	●●●	●●●
Embedded Operating System (Internet access,...)	●○○	●○○	●○○	●●●
FLEXIBILITY OF USE	●●○	●●○	●●○	●●○
Coding facilities	●●●	●●○	●●●	●●○
Micro-architecture adaptation, Sw processor core implementation	○○○	●●●	●○○	●●●
Learning curve	●●●	●●○	●●○	●○○
COST	€○○	€€○	€○○	€€○

Score: ○○○ bad, ●○○ medium, ●●○ good, ●●● very good

With SoC FPGAs, the hardware processor cores and the FPGA fabric are tightly coupled for such control of complex electrical systems, so that the data communication is achieved with low latency. Therefore, one critical point that needs attention is the priority interrupts management. A Vectored Interrupt Controller (VIC) integrated in the soft-core processor NIOS II (Intel/Altera) or hard-core processors ARM Cortex-R and M is mandatory to ensure the lowest interrupt latency and constant low jitters for real-time applications, compared with general ARM Cortex-A [17]. Thanks to the VIC unit of the Cortex-R5 of Xilinx Zynq Ultrascale+, this powerful component is ready to handle critical real-time applications and due to the integration a quad-core Cortex-A53, it is also highly adapted to high computing applications. However, considering the reduction of the costs, a soft-core processor solution, such as the NIOS II, may sometimes be a better option rather than using an over-sized SoC FPGA owed to their interesting properties: low interrupt latency and hardware adaptability to the system to be controlled.

A proof of concept system, shown in Fig. 3, was implemented to validate the performance of a SoC FPGA-based smart controller for a hybrid Fuel Cell (FC) system composed of a FC stack and SuperCapacitors (SCs). It is worth mentioning that this plant is emulated in the DS1006 and DS5203 dSPACE boards [14]. All the corresponding blocks in Fig. 3 are in solid blue lines. The modules related to the

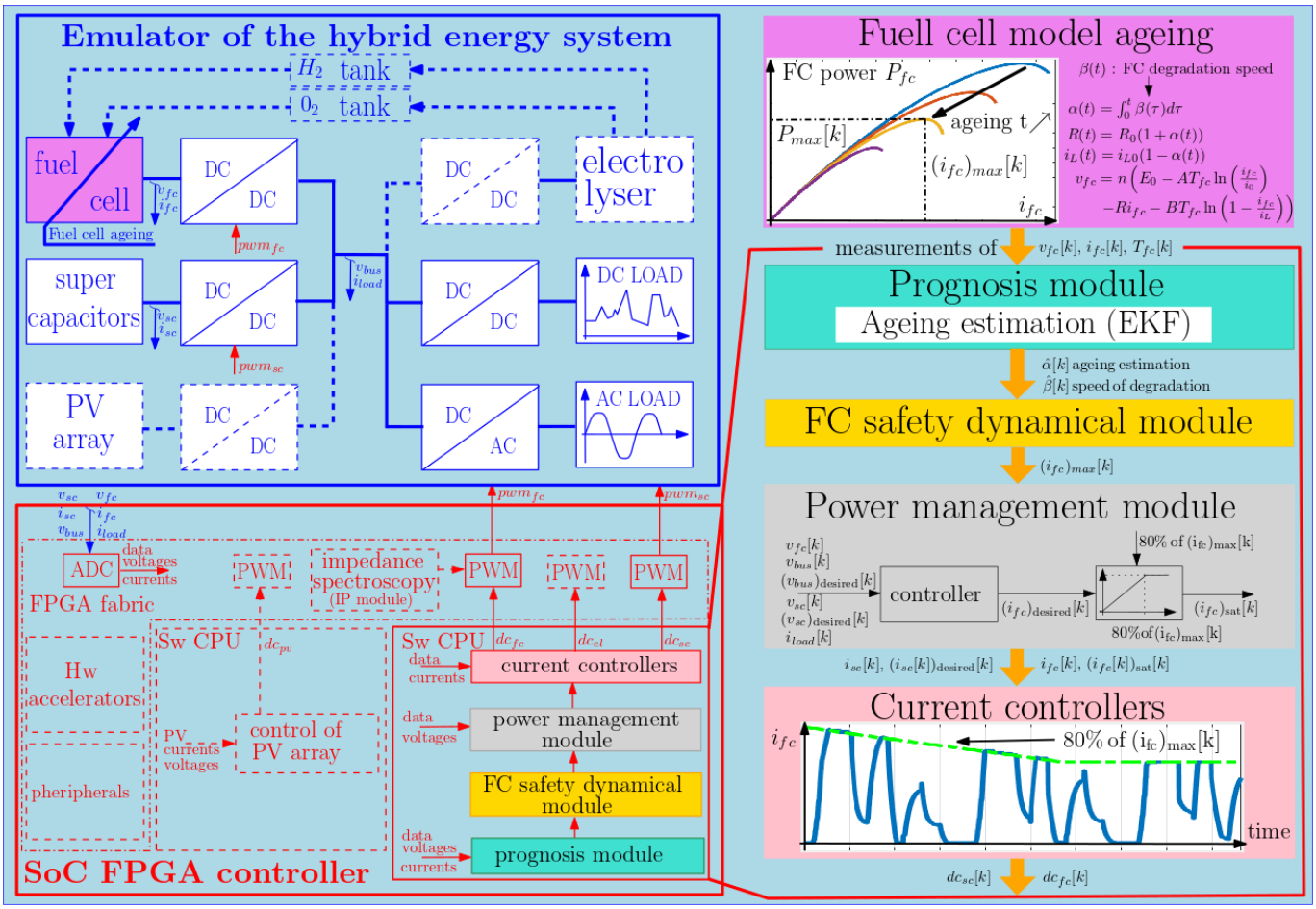

Fig. 3: Hybrid fuel cell system architecture.

SoC FPGA-based controller are in solid red lines in Fig. 3. Among them, the FC control and prognosis algorithms have been implemented in a NIOS II on a low-cost Cyclone V board (DE1 SoC Intel/Altera). Finally, all the modules in dashed lines, both within the plant or within the SoC FPGA-based controller, are not present in the current study but can be included for next prospects, thus showing the high level of scalability of the presented SoC FPGA-based control framework. The complete Hw/Sw system represents a Hardware-In-the-Loop (HIL) platform to validate the algorithms in real-time [14]. The SoC FPGA architecture is composed of: Two PWM units, an acquisition unit of 6 ADCs and a Soft core base on a NIOS

II. The algorithms are executed in three Interrupt Service Routines (ISR) based on three synchronized timers events configured with a sampling time equal to 50 μs for the current loops and PWMs, 500 μs for the power management, and 1s for a Prognosis and Health Management (PHM) algorithms. The 3 ISRs use vectorized interrupts with a highest priority (0) for the current controllers (ISR0) and then priority 1 for the power management module (ISR1). The computation times are equal to 7.20 μs, 9.84 μs and 117 μs respectively [14].

Fig. 4 shows all the main data computed in the emulated system (blue curves) and in the NIOS II processor (red curves), these colors correspond to those chosen in Fig. 3. Ageing $\alpha(t)$ of the PEMFC that has been emulated in the FC model is estimated on-line ($\hat{\alpha}[k]$) by the PHM algorithm (here an Extended

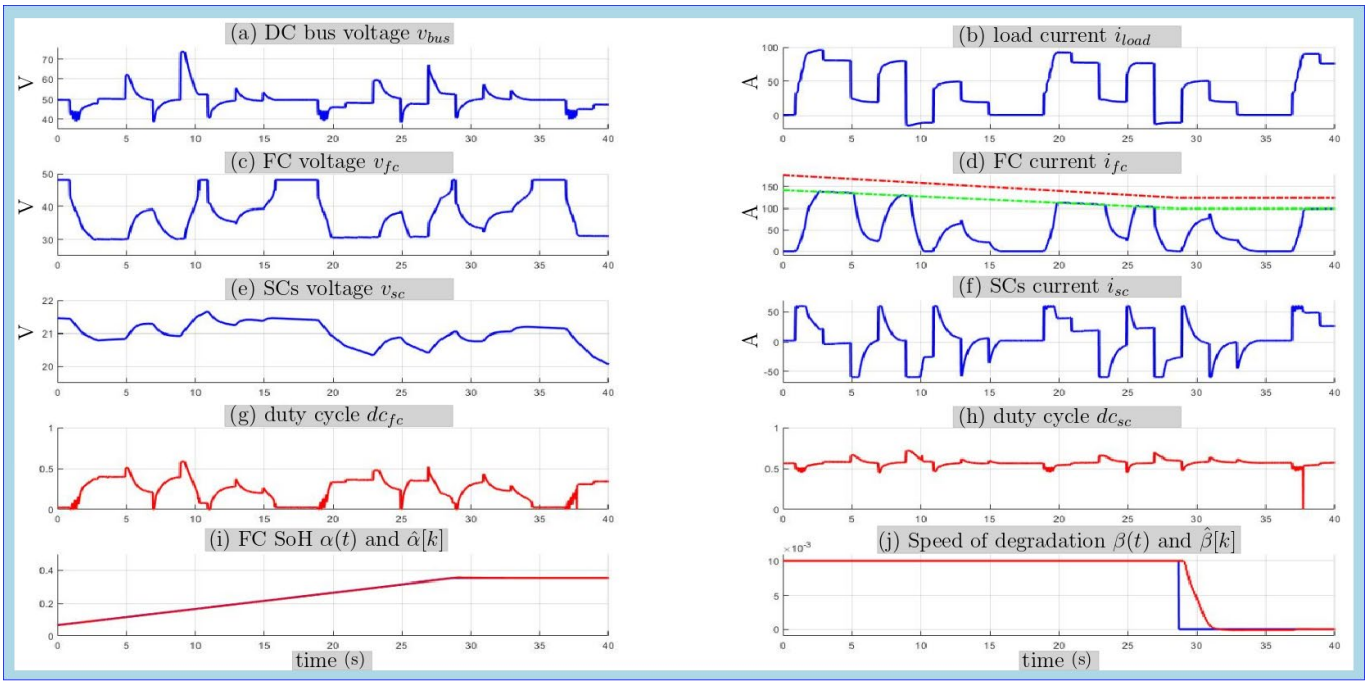

Fig. 4: Hybrid fuel cell HIL results.

Kalman Filter) [14], [18]. The FC safety dynamical module computes the maximum FC current $(i_{fc})_{max}[k]$ value that must not be exceeded. It can be noticed that the FC current i_{fc} is well controlled by the SoC FPGA-based controller since it does not exceed the defined maximum current fixed to 80% of $(i_{fc})_{max}[k]$ (see Fig. 4.(d)). This means that both the speed of degradation $\beta(t)$ (see Fig. 4.(j)) and the aging $\alpha(t)$ (see Fig. 4.(i)) are well estimated by the observer implemented within the NIOS II soft-core processor [14]. Moreover, as the current controllers and peripherals implemented within the FPGA Fabric (PWM, ADC) need to be tightly coupled, it appears that the soft-core processor is a valuable option to provide

deterministic interrupt, minimum jitters, many possibilities of evolution of the proposed hardware/software architecture and it reduces the risk of obsolescence.

B. Dynamical reconfiguration of PV modules

Shadowing significantly affects PV arrays electrical power production and may lead to the conduction of the modules bypass diodes. Consequently, more than one Maximum Power Point (MPP) appear in the string Power vs. Voltage (P-V) and Current vs Voltage (I-V) curves [19]. Depending on the actual shadowing pattern, the adoption of a system allowing to change the electrical connections among the PV modules through a suitable switching matrix [20] is useful. The reconfiguration has to be performed dynamically, because the shading pattern changes during the day, and in a short time interval during which the irradiance level received by the PV cells does not change significantly.

In [21], a theoretical analysis of the problem was proposed, and in [22], an Evolutionary Algorithm (EA) aimed at dynamically determining the best electrical configuration of the PV modules in a plant formed by more strings was presented.

Fig. 5 shows a fixed shadow affecting the PV array and two EA individuals, each corresponding to a specific electrical connection of modules, to form the two parallel connected strings. The green P-V curve

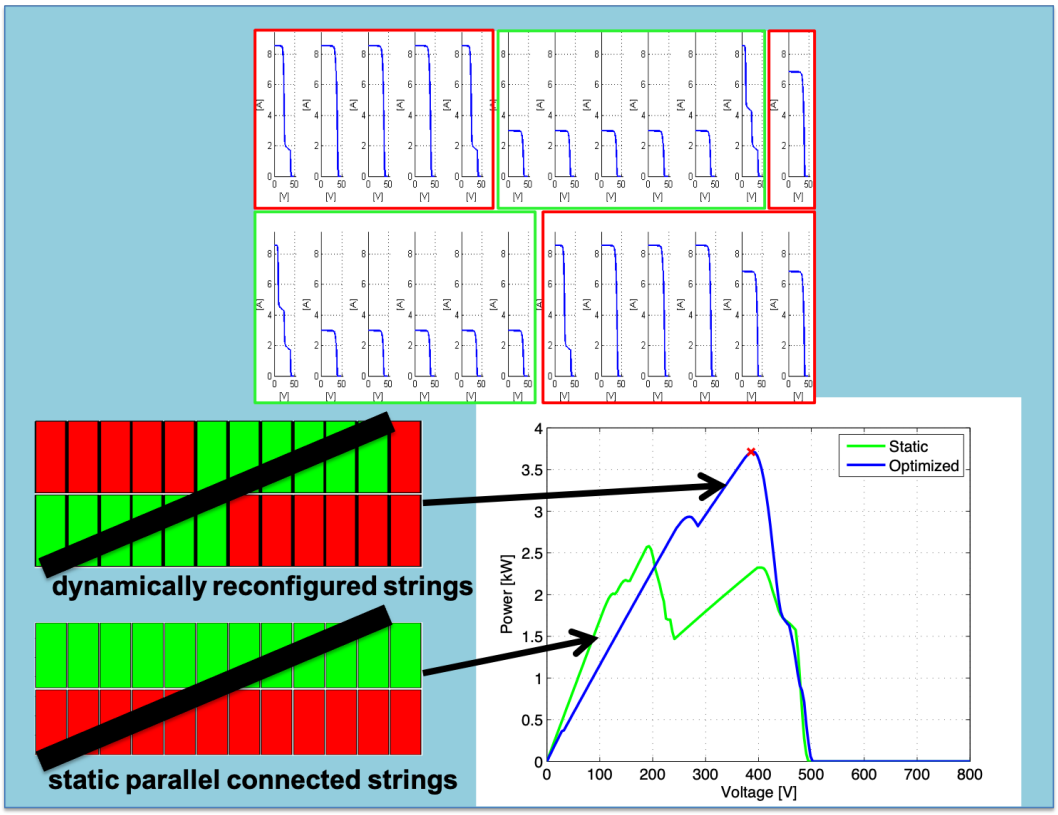

Fig. 5: Dynamic reconfiguration of a PV array of 2 parallel connected strings of 12 modules each. Top: modules I-V curves. Bottom left: strings affected by an oblique shadow. Green and red modules are series connected. Bottom right: P-V curves corresponding to the static and to the reconfigured PV field.

corresponds to the static connection, showing a peak power lower than the one the EA determines and corresponding to the configuration with blue P-V curve.

The conjoint Hw/Sw SoC FPGA-based implementation [23] (Fig. 6) consists of the core of the EA, implemented in Sw on bare metal ARM A9 core, and of the fitness function instances that are executed in a couple of dedicated Intellectual Property (IP) modules within the FPGA fabric. The 12-bit fixed-point representation ensures a good trade off between the FPGA fabric consumed area and the loss of accuracy, which is less than 1% compared to the reference case based on a 32-bit floating-point representation. The fitness function IP module is written in C++ and the architectural design space is explored by using the High Level Synthesis (HLS) approach [24]. HLS allows to design the hardware accelerator through high level languages, e.g. C/C++, by generating production-quality register transfer level (RTL) code that is optimized for the targeted FPGA. The synthesis process transforms automatically a C/C++ source code in a hardware descriptive language such as VHDL or System Verilog. HLS accelerates verification time over RTL by raising the abstraction level for FPGA hardware design. HLS designs are typically verified orders of magnitude faster than RTL ones. The algorithm is preliminary optimized in order to put into evidence the subroutines to be run in parallel and by using a counting sort algorithm, allowing to save up to 80% of computation time with respect to the use of a standard bubble sort algorithm. The reduction on the size of the fixed-point divider leads to a 20% reduction of the latency of the fitness function.

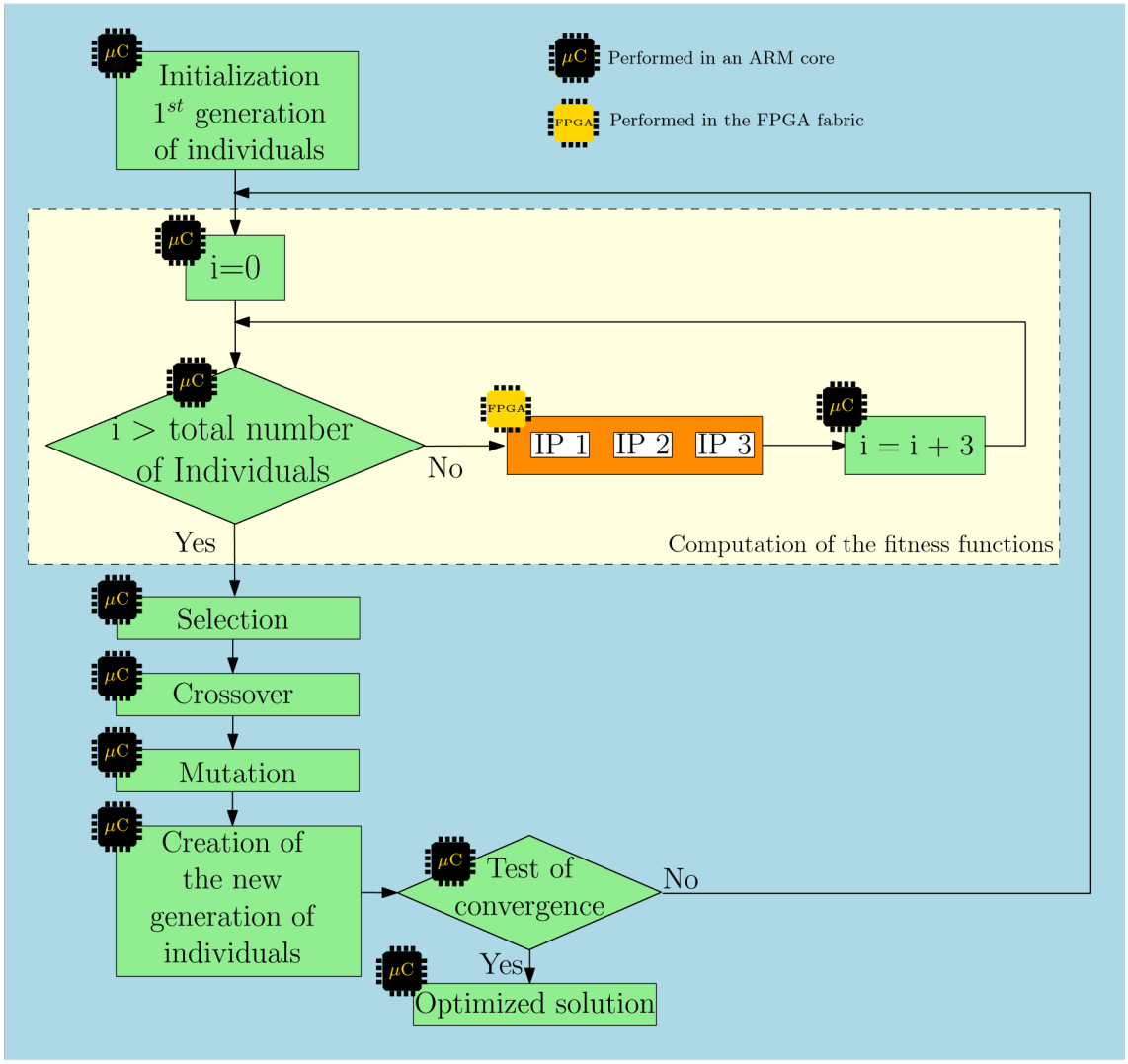

Fig. 6: Hw/Sw implementation of the PV dynamic reconfiguration algorithm.

Two practical cases, with 100 and 25 samples per module I-V curve respectively, were implemented on the low-end Zynq-based board Zybo from Digilent at an FPGA clock frequency of 125 MHz. The PV field has 24 modules divided in 2 parallel connected strings. The EA runs on a population of 48 individuals, for a maximum of 100 generations. The experiments revealed that, if 100 samples per curve are used, two fitness function Hw accelerators can be integrated in parallel in the FPGA fabric. Instead, three IPs modules can be embedded in the 25 samples per curve case. The acceleration rate for the 100 sample case is of 2.46 compared to an optimized full Sw implementation based on a bare metal ARM A9 core running at 667MHz, thus leading to a total execution time of 13.218 s. Instead, the acceleration rate for the 25 sample case is of 2.80, with a total execution time of 2.374 s.

IV. Next Generation of Smart Controllers for Electrical Energy Systems

The electrical energy sector will be pushed at European level by the EU Green Deal [25] and the EU Recovery Plan [26], also in view of its integration with other energy sectors [27], [28] and with digital technologies for achieving the de-carbonisation goal.

The significant contributions SoC FPGA can bring to the key future developments of renewable generators and hydrogen technology are discussed through the two applications presented in the former section. SoC FPGA will facilitate meeting the EU expectations and targets in other fields, too, such as battery management and diagnostic systems (e.g. [29], [30]).

Monitoring and diagnostic functions will benefit from the decentralized high computational potential SoC FPGAs offer, enabled by the use of the model-based approach for PV systems [31] and even running data driven approaches (e.g. [32] for fuel cells applications and [33] for PV systems).

Smart power management area will also profit from SoC FPGAs, especially by introducing in the controller Digital Twins (DTs) of the used static power converters. DTs have several benefits such as the possibility to make online diagnosis [34], or to study with fine details the power losses of a complex structure like a Modular Multilevel Converter (MMC) [35].

Finally, the optimized economic dispatching of a microgrid is also a good example of the ongoing mutation in terms of control algorithmic needs for modern complex electrical energy systems [36].

These recent works are good illustrations of what could be the next generation of smart controllers for complex electrical energy systems. Beyond the standard control functions (still implemented), these smart controllers will also include additional tasks like diagnosis, fault tolerant capabilities, optimization of the energy flow and/or economical dispatching. These new functionalities can be gathered under the generic name of smart monitoring and it is worth analyzing their impact on the architecture of smart controllers.

In complex electrical energy systems, the first task for smart controllers is to collect and aggregate the measurements coming from all the internal sub-elements. An analysis of references shows that three approaches are possible to cope with this problem. A typical approach is to use a standard serial communication like CAN bus [29] between the low-end microcontroller that is in charge of the monitoring of a given cell and the centralized SoC FPGA-based smart controller. A second solution is to use a wireless connection (WIFI or Bluetooth) [30],[31], since it offers more flexibility and scalability than a classical serial wired communication. Finally, a more radical approach is to integrate all the necessary front end analog resources needed to measure and collect the data coming from the cells in an ASIC like in [32] that integrates also the SoC FPGA-based smart controller. This solution is highly integrated but very specific and thus costly to design. However, and if, as expected, the market of the Industrial Internet of Things (IIoT) will be booming, it cannot be excluded that SoC FPGA manufacturers will propose in the future new devices including more analog capabilities than today. Such trend is the Xilinx RF-SoC device, devoted to the 5G software radio market [37].

The main tasks to be performed by the smart controller are: diagnosis [31] or [33], health monitoring [18] and energy management [23], [36]. Sometimes a higher integration is the key objective [30], where the smart controller is performing in parallel both BMS and Charger functionalities.

Depending on the time scale of these smart monitoring tasks, the controller has to apply hard real-time operating conditions (μs-ms with a full timing determinism, achieved by timer interruptions and with a

bare metal configuration of the processor in order to minimize the latency) [18], [35], or soft real-time operating conditions (from seconds to hours when timing determinism is less critical). In this case, it is of great interest to execute the smart monitoring tasks as processes of a real-time operating system like an embedded Linux [29], thus profiting from its communicating facilities. Even when hard real-time operating conditions are mandatory, it is still possible to dedicate one core processor of the SoC FPGA to run Linux OS, while the other one is bare metal and only devoted to the critical control tasks [38] ("asymmetric multiprocessing").

Regarding the nature of smart monitoring strategies, most of them are based on the simulation of a plant model [29].Some of these approaches require an optimization problem that has to be solved online [31], [36], [23]. The rest of these studies, like those integrating a digital twin [34], [35], are based on estimators or observers [30], [18]. However, whatever the smart controller has to execute, a stochastic optimization problem or an embedded DT, the computing load is high. Therefore, it is interesting to analyze how the Hw/Sw partitioning, which consists of choosing which parts of the control algorithm are implemented in a processor and which ones are implemented in a Hw accelerator, is conducted: for the EA-based optimization, the main body of the EA is implemented in Sw and the fitness function instances are implemented as Hw accelerators [31], [23]. As for estimators and observers, the Hw/Sw partitioning is usually based on the dynamics of the model to emulate: a slow temperature estimator is naturally implemented in Sw, while the battery state-of-charge estimator is done in Hw [29]. In [35], as the submodule estimators of the MMCs are prone to parallelization, they are placed in the FPGA fabric. But in [34], a full FPGA implementation is performed, which results as the only choice due to the conjoint short dynamics of the emulated power converters and the complexity of the stochastic models used.

With the progress of machine learning (ML) methods, data driven approaches are increasingly popular for the diagnosis of complex electrical energy systems. These concern classification [32] o r regression techniques [33], both requiring a complex offline training process, but the online inference process may be relatively simple. However, in many Neural Network (NN) classification or regression problems, the trained NN is fed by new incoming data from the plant. This means that conversely to [33], the local smart controller has to implement an inferred NN. An inferred NN, as a simplified version of an optimally trained deep NN, has a reduced power and latency for meeting edge applications requirements. The deep NN is trained off line; then, through pruning and quantization methods, the groups of artificial neurons that rarely or never fire are removed and the numerical precision of the weights is reduced, so that a reduced model size and a faster computation is achieved at the cost of minimal reduction in prediction accuracy [39]. Based on the parallel characteristics inherent to such algorithms, an FPGA-based or GPUbased implementation is thus highly recommended [40].

The overview above reveals that most smart monitoring applications are implemented in a SoC FPGA device since these heterogeneous computing platforms reached very good computing performance and enable their architecture's customization thanks to the FPGA fabric. With the help of a performing realtime OS like embedded Linux, these devices are easily connectable to Internet so they are good candidates to the probably biggest mutation currently experienced in digital controllers: the transformation of the "local embedded controllers" into Edge Computing Platforms (ECP). So, the "smart controllers" mentioned above are not only able to handle locally complex control functions and smart monitoring tasks, but can also be part of a larger control system that distributes some tasks to a remote Cloud Computing Platform (CCP). This transformation is directly derived from the industrial Internet of Things concept [41]. The distribution of the tasks between the ECP and the CCP can be seen as an evolution of the embedded control concept, with smart monitoring tasks processed locally. However, in [33] and [36], a different philosophy is proposed: all the prediction tasks are achieved in advance on an hourly/daily basis and the ECP just has to compare the information received from the plant with these predictions. Thus, the computing load is clearly moved remotely into a CCP and, as consequence, the ECP can remain very light like in [36], where a single DSP chip is sufficient to implement a decision maker based on simple tests.

To conclude, the fact that SoC FPGA-based ECPs are able to collect data from the cell unit controllers, use it locally to execute smart monitoring tasks and interact with a CCP where hourly/daily training of NN is achieved or where other slow supervising and storage tasks are being performed, opens new interesting lines of research. One of them is the opportunity to enlarge significantly the size of the electrical energy systems to manage [36], where the same CCP can handle the economic dispatching forecasts for several microgrids. The next step will be to integrate the possibilities of cooperation between different electrical energy systems, but reinforcing the security and the privacy of the connections between the ECP and the CCP will be of concern. Finally, sending and storing on a daily basis to a CCP relevant features computed locally by an ECP can monitor any complex electrical energy system over its entire lifespan. A lot of effort has to be dedicated to this topic part of the energy internet of things (eIoT).

V. Conclusions

The significant contributions SoC FPGA can bring to the key future developments of complex electrical energy systems, especially by referring to those ones including renewable generators and employing hydrogen technology, were discussed in the paper. Some detailed advantages and limitations were exemplified through the two specific applications presented in the case studies. One concerns a fuel cell hybrid electric system controlled through passivity-based power management associated with an ageing

prognosis algorithm. The other one reports the SoC FPGA implementation of a control system able to optimize online the dynamical configuration of a partially shadowed photovoltaic field. Besides these two case studies, authors also analyzed in detail a series of recently reported results on smart controllers for complex electrical energy systems, highlighting the importance of the increasing number of smart monitoring tasks performed by this new generation of controllers, e.g. diagnosis, prognosis, fault tolerant capabilities, optimization of the energy flow and/or economical dispatching. Despite of a certain number of limitations like the cost, that is higher than for other technologies such as SoC DSPs, a limited analog interfacing (A/D, D/A) and a designer's longer learning curve for optimal use, SoC FPGA is however one of the most promising digital technologies to implement such smart controllers. Indeed, by investigating with care the implications in terms of implementation of these new smart monitoring tasks, it was shown that SoC FPGA devices are not only able to manage complex algorithm online processing, such as an EA optimization or a Digital Twin, but they can also help to accelerate their execution by parallelizing into customized Hw accelerators several computationally demanding subtasks like fitness function calculation. Furthermore, thanks to their highly performing FPGA fabric, SoC FPGAs are also offering a high level of flexibility in terms of micro-architecture. A good illustration of this is the possibility for the designer to add one or several simple Sw core processors, thus relieving the processing system of the device from low level time-consuming tasks. Finally, another important advantage was pointed out: the ability of SoC FPGA to easily communicate both with the system to be controlled, thanks to a very large number of I/Os, and with remote cloud services, via the possibility to easily embed a Linux Operating System. This makes such a SoC FPGA-based smart controller a highly performing Edge Computing Platform, able to address the incoming challenges in terms of complexity and storage brought on by data driven approaches.

REFERENCES

[1] C. Buccella, C. Cecati, and H. Latafat, "Digital control of power converters—a survey," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 437–447, 2012.

[2] H. A. Toliyat, *DSP-Based Electromechanical Motion Control*. CRC Press, Inc., 2003.

[3] E. Monmasson, L. Idkhajine, and M. W. Naouar, "Fpga-based controllers," *IEEE Industrial Electronics Magazine*, vol. 5, no. 1, pp. 14–26, 2011.

[4] S. M. Trimberger, "Three ages of fpgas: A retrospective on the first thirty years of fpga technology," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 318–331, 2015.

[5] J. Millan, P. Godignon, X. Perpina, A. Perez-Tomas, and J. Rebollo, "A survey of wide bandgap power semiconductor devices," *IEEE Transactions on Power Electronics*, vol. 29, no. 5, pp. 2155–2163, 2014.

[6] W. Tu, G. Luo, Z. Chen, C. Liu, and L. Cui, "Fpga implementation of predictive cascaded speed and current control of pmsm drives with two-time-scale optimization," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5276–5288, 2019.

[7] J. J. Rodr´ıguez-Andina, M. D. Valdes-Pe´ na, and M. J. Moure, "Advanced features and industrial applications of fpgas- review,"˜ *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 853–864, 2015.

[8] T. Adegbija, A. Rogacs, C. Patel, and A. Gordon-Ross, "Microprocessor optimizations for the internet of things: A survey," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 7–20, 2018.

[9] "Tms320f2837xd dual-core delfino™ microcontrollers," 2013, last accessed 30 Sept. 2019. [Online]. Available: http://www.ti.com/lit/ds/sprs880k/sprs880k.pdf

[10] R. F. Molanes, J. J. Rodr´ıguez-Andina, and J. Farina, "Performance characterization and design guidelines for efficient processor–fpga˜ communication in cyclone v fpsocs," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4368–4377, 2018.

[11] [Online]. Available: https://www.arm.com/resources/designstart/designstart-fpga

[12] J. Pierson, G. Baudic, S. Caux, B. Celik, G. Da Costa, L. Grange, M. Haddad, J. Lecuivre, J.-M. Nicod, L. Philippe, V. Sonigo, R. Roche, G. Rostirolla, A. Sayah, P. Stolf, T. Minh-Thuyen, and C. Varnier, "Datazero: Datacenter with zero emission and robust management using renewable energy," *IEEE Access*, vol. 1, p. 146, jul 2019. [Online]. Available: https://publiweb.femto-st.fr/tntnet/entries/15835/documents/author/data

[13] S. Abuzant, S. Jemei, D. Hissel, L. Boulon, K. Agbossou, and F. Gustin, "A review of multi-stack pem fuel cell systems: Advantages, challenges and on-going applications in the industrial market," *14th IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2017.

[14] S. Kong, M. Bressel, M. Hilairet, and R. Roche, "Advanced passivity-based, aging-tolerant control for a fuel cell / super-capacitor hybrid system," *Control Engineering Practice*, vol. 105, no. September, p. 104636, 2020. [Online]. Available: https://doi.org/10.1016/j.conengprac.2020.104636

[15] D. Depernet, A. Narjiss, F. Gustin, D. Hissel, and M.-c. Pera, "Integration of electrochemical impedance spectroscopy functionality in proton exchange membrane fuel cell power converter," *International Journal of Hydrogen Energy*, vol. 41, no. 11, pp. 5378 – 5388, mar 2016.

[16] E. Dijoux, N. Yousfi Steiner, M. Benne, M.-C. Pera, and B. Grondin-Perez, "A review of fault tolerant control strategies applied to´ proton exchange membrane fuel cell systems," *Journal of Power Sources*, vol. 359, pp. 119 – 133, aug 2017. [Online]. Available: https://publiweb.femto-st.fr/tntnet/entries/13912/documents/author/data

[17] "White paper, real-time challenges and opportunities in socs," *ALTERA*, Mar. 2013.

[18] M. Bressel, M. Hilairet, D. Hissel, and B. Ould-Bouamama, "Model-based aging tolerant control with power loss prediction of proton exchange membrane fuel cell," *International Journal of Hydrogen Energy*, vol. 45, no. 19, 2020.

[19] G. Petrone, C. R. Paja, and G. Spagnuolo, "Photovoltaic sources modeling, 1st edition," *IEEE Wiley*, 2017.

[20] G. Petrone, G.Spagnuolo, B. Lehman, Y.Zhao, C. R. Paja, and M. O. Gutierrez, "Photovoltaic arrays dynamical reconfiguration: Fighting mismatched conditions and meeting load requests," *IEEE Industrial Electronics Magazine*, vol. 9, no. 1, pp. 62–76, 2015.

[21] M. Orozco-Gutierrez, G. Spagnuolo, J. Ramirez-Scarpetta, G. Petrone, and C. Ramos-Paja, "Optimized configuration of mismatched photovoltaic arrays," *IEEE Journal of Photovoltaics*, vol. 6, no. 5, pp. 1210–1220, Sep. 2016.

[22] P. Carotenuto, A. D. Cioppa, A. Marcelli, and G. Spagnuolo, "An evolutionary approach to the dynamical reconfiguration of photovoltaic fields," *Neurocomputing*, vol. 170, pp. 393–405, 2015.

[23] G. Petrone, F. Serra, G. Spagnuolo, and E. Monmasson, "Soc implementation of a photovoltaic reconfiguration algorithm by exploiting a hls-based architecture," *Mathematics and Computers in Simulation*, vol. 158, pp. 520–537, 2019.

[24] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, "High-level synthesis for fpgas: From prototyping to deployment," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 473–491, April 2011.

[25] "Communication on the european green deal," https://ec.europa.eu/info/publications/communication-european-green-deal en, accessed: 2020-07-08.

[26] "EU recovery plan," =https://ec.europa.eu/info/live-work-travel-eu/health/coronavirus-response/recovery-plan-europe en, accessed: 2020-07-08.

[27] "Energy system integration strategy," https://ec.europa.eu/energy/sites/ener/files/energy _system integration strategy .pdf, accessed: 2020-07-08.

[28] "Hydrogen strategy," https://ec.europa.eu/energy/sites/ener/files/hydrogen strategy.pdf, accessed: 2020-07-08.

[29] R. Morello, R. Di Rienzo, R. Roncella, R. Saletti, R. Schwarz, V. R. H. Lorentz, E. R. G. Hoedemaekers, B. Rosca, and F. Baronti, "Advances in li-ion battery management for electric vehicles," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 4949–4955.

[30] T. Gherman, M. Ricco, J. Meng, R. Teodorescu, and D. Petreus, "Smart integrated charger with wireless bms for evs," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 2151–2156.

[31] M. Garaj, K. Y. Hong, H. S.-H. Chung, J. Zhou, and A. Lo, "Photovoltaic panel health diagnostic system for solar power plants," *IEEE Applied Power Electronics Conference and Exposition (APEC), Anaheim, CA, USA*, pp. 1078–1083, 2019.

[32] Z. Li, R. Outbib, S. Giurgea, D. Hissel, A. Giraud, and P. Couderc, "Fault diagnosis for fuel cell systems: A data-driven approach using high-precise voltage sensors," *Renewable Energy*, vol. 135, pp. 1435–1444, 2019.

[33] S. Leva, M. Mussetta, and E. Ogliari, "Pv module fault diagnosis based on microconverters and day-ahead forecast," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3928–3937, 2019.

[34] M. Milton, C. D. L. O, H. L. Ginn, and A. Benigni, “Controller-embeddable probabilistic real-time digital twins for power electronic converter diagnostics,” *IEEE Transactions on Power Electronics*, vol. 35, no. 9, pp. 9850–9864, 2020.

[35] Z. Shen and V. Dinavahi, “Real-time mpsoc-based electrothermal transient simulation of fault tolerant mmc topology,” *IEEE Transactions on Power Delivery*, vol. 34, no. 1, pp. 260–270, 2019.

[36] S. Wang, X. Wang, and W. Wu, “Cloud computing and local chip-based dynamic economic dispatch for microgrids,” *IEEE Transactions on Smart Grid*, pp. 1–1, 2020.

[37] B. Farley, J. McGrath, and C. Erdmann, “An all-programmable 16-nm rfsoc for digital-rf communications,” *IEEE Micro*, vol. 38, no. 2, pp. 61–71, 2018.

[38] E. Zafra, S. Vazquez, H. Guzman Miranda, J. A. Sanchez, A. Marquez, J. I. Leon, and L. G. Franquelo, “Efficient fpsoc prototyping of fcs-mpc for three-phase voltage source inverters,” *Energies*, vol. 13, no. 5, p. 1074, Mar 2020.

[39] “Deep learning training and inference,” https://www.intel.com/content/www/us/en/artificial-intelligence/posts/deep-learning-trainingand-inference.html, accessed: 2020-11-24.

[40] X. Rancano, R. Fern˜ andez Molanes, C. Gonz´ alez-Val, J. Rodr´ ´ıguez-Andina, and J. Farina, “Performance evaluation of state-of-the-art˜ edge computing devices for dnn inference,” in *IECON 2020 - 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020.

[41] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, and X. Guan, “Industrial edge computing: Enabling embedded intelligence,” *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 48–56, 2019.