The Internet of Things

The Role of Reconfigurable Platforms

MARÍA DOLORES VALDÉS PEÑA,
JUAN J. RODRIGUEZ-ANDINA,
and MILOS MANIC

Digital Object Identifier 10.1109/MIE.2017.2724579
Date of publication: 21 September 2017

The Internet of Things (IoT) paradigm has quickly gone past various technological domains to become part of everyday life across the globe. Ubiquitous devices (things) with a certain amount of intelligence, capable of connecting to the Internet and sharing information collaboratively are now a widespread reality [1]–[3]. The general perception today indicates that the IoT may become a technological milestone, with an impact similar to that caused by the advent of the Internet itself. It will transform the Internet from a humancentric platform, where most

applications are focused on human-to-human interaction and where most of the data are provided by humans, to a machine-intensive platform.

In addition, the availability of data generated by machine-to-machine (M2M) communication and interactions among things will enable the more effective and intelligent remote control of processes or things [4]. The variety of concepts associated with the IoT paradigm is also expected to open a wide range of opportunities for technological research and development as well as new business opportunities, which will also require new regulations [5].

This article discusses different perspectives that lead to different functional and implementation models for the IoT. Reconfigurable devices, namely, field programmable gate arrays (FPGAs), are one of the leading technologies on the market for the implementation of digital systems combining software parts with hardware accelerators. The latest generations of these embedded devices include, among many other useful resources, powerful embedded hard processors supporting different operating systems, analog front ends, specialized hardware blocks for high-performance computing or cryptoacceleration, and communication interfaces compatible with the most widely used network protocols. This has given rise to the field programmable system-on-chip (FPSoC) concept [6]. The combination of the plethora of resources with standard configurable logic enables the efficient implementation of systems that perfectly fit the heterogeneous nature of IoT applications. This is because both hardware and software components can be configured according to the needs of different target applications; they are relatively low cost, low power, and compact, and their flexibility and possibility of code reuse (both hardware and software) allow the time to market to be reduced.

IoT Versus Cyberphysical Systems

The terms *IoT* and *cyberphysical system* (*CPS*) are so closely related that many professionals in the field use them interchangeably. Both technologies share core elements (such as sensors, actuators, and computing and communication resources), so it is difficult to define clear borders between them.

Different definitions for *IoT* are given in [7], coming from standardization organizations such as the European Telecommunications Standards Institute, the International Telecommunication Union, the IEEE, and the National Institute of Standards and Technology (NIST). It is worth noting that most IoT activities in the United States are considered to be part of CPSs. For instance, the NIST does not define *IoT* on its own but as associated with CPSs, and the organization uses both terms interchangeably [8].

Regardless of existing definitions, other industry and academic professionals see differences between the IoT and CPSs. Lee, for example, defines CPSs as "orchestrations of computers and physical systems" in which "embedded computers monitor and control physical processes, usually with feedback loops, where physical processes affect computations and vice versa" [9], [10]. He considers the IoT to be an implementation approach for CPSs.

Minerva et al. [7] consider CPSs as systems based on the cooperative work of sensors and/or actuators to achieve a specific goal. In this context, they consider IoT systems as vehicles that perform such cooperative work in a distributed manner. To them, contrary to IoT systems, CPSs are not necessarily connected to the Internet.

In Lin et al.'s [11] view, CPSs are intended for monitoring and control of physical components, whereas the IoT focuses on functions for sharing and managing resources and data, interfacing among different networks, massive-scale data and big data collection and storage, data mining, or data aggregation and information extraction. They describe CPSs as vertical architectures consisting of three layers, namely, the sensor/actuator, communication, and application (control) layers, and the IoT as a horizontal architecture connecting different CPSs, integrating their respective communication layers. This view of the IoT as an interconnecting infrastructure for CPSs to achieve a global objective is the most widely accepted among those who differentiate between the two concepts [12].

A similar idea is found in [13], where the IoT is defined as a complex CPS that integrates devices with different capabilities for data acquisition, identification [(ID)—note

The IoT may become a technological milestone, with an impact similar to that caused by the advent of the Internet itself.

that ID is only used in this article as a technical term for the means of identification, not for the identification itself] processing, communication, and networking. In [14], the CPS concept is associated with machines that include embedded systems capable of processing data and communicating among them to increase their autonomy, whereas the IoT is identified as a complementary technology providing the resources required for machines to communicate with things, allowing products to be identified and tracked during their whole life cycle.

The main goal of this article is to analyze the current and potential roles of reconfigurable devices, particularly FPSoCs, in an IoT context. Although, as discussed previously, different views of the IoT and CPS exist, no distinction is made here; we refer to the IoT since, from the perspective of hardware support, there are no major differences justifying a separate analysis.

Some History of the IoT

Although the term *IoT* and its associated concepts have become popular in recent years, their origin can be traced back nearly 20 years (to approximately 1999) with the establishment by Kevin Ashton of the Auto-ID Center at the Massachusetts Institute of Technology.

At the end of the 1990s, while working for a big multinational company specializing in cleaning agents, personal care, and hygiene products, Ashton realized that inventory control in retail stores needed an improvement to address the problem of running out of given products. This improvement, he understood, would have a positive impact on efficiency and profitability, as inventories could be adjusted to demand and consumer needs in real time. Ashton's idea was to provide the supply chain with some intelligence to enable it to be automatically managed, but this implied the need for a unique ID of any product or object (thing). At that point, objects were mostly identified using bar codes with Universal Product Code (UPC) encoding. Although bar-code technology remains widely used, it comes with two major limitations. First, it is a line-of-sight technology, so only one object can be identified at a time, and identification relies on the proper positioning of the product in front of the reader. Second, a UPC allows only for product categories to be identified, not the individual items within a given category.

The Auto-ID Center, a nonprofit research consortium, was created with the idea of using radio-frequency (RF) identification as the enabling technology to develop an open network capable of automatically identifying, tracking, and tracing any physical object in a global supply chain [15]. As a result, the Electronic Product Code (EPC) standard was developed, which allowed individual objects to be uniquely identified, thanks to the use of low-cost RFID tags with embedded EPC coding that could be wirelessly read. The Auto-ID Center closed in 2003, but its work was continued by EPCglobal [16], which commercializes EPC technology, and Auto-ID Labs [17], which conducts academic research in the area.

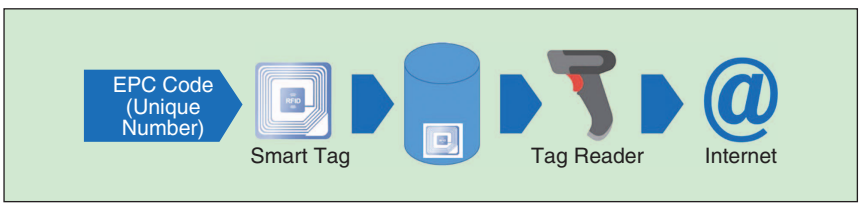

FIGURE 1 – The fundamentals of Auto-ID technology.

Auto-ID technology (Figure 1) relies on three basic actions:

1) object identification
2) data acquisition
3) information management, exchange, and analysis.

During production, each object is provided with an RFID tag with an EPC ID embedded in it. When any RFID reader (located in a production plant, storage facility, or shop) scans the tag, it gets the tag's EPC ID. This ID is sent to a host running a distributed middleware that filters data (e.g., removing repeat IDs when two readers access the same object). Since the technology targets the identification, tracking, and tracing of trillions of objects, an efficient middleware is essential to control data flows and avoid network congestion (public or private).

The middleware sends the EPC ID to an Object Name Service (ONS). ONS is a standard developed by the Auto-ID Center, based on the Internet's Domain Name Service (DNS). ONS is in charge of linking each EPC ID with a distributed network of servers on the Internet, where databases of objects are hosted. The ONS determines where to find information about any object, including an EPC ID. These servers are based on the physical markup language standard, developed by the Audio-ID Center from XML. It allows information to be organized to build complex descriptions of physical objects and their related production and commercialization processes. Eventually, these developments resulted in the first thing-oriented network, where objects can identify themselves and connect to a data network to share information with other objects, processes, and services, aiming at added-value creation, improvement of quality of service and efficiency in production processes, and cost reduction.

Nearly two decades later, the value of this revolutionary technology can be put in perspective as a means to transform various aspects of the industry. These aspects entail the way

products are fabricated, commercialized, distributed, purchased, consumed, supplied, or even recycled. This applies to numerous diverse areas, such as industry, services, retail stores, transportation and logistics, health, and food. Currently, the model based on RFID tags, EPC IDs, and ONS is still one of the most-used architectures in commercial IoT applications.

Concepts, Architectures, and Challenges of the IoT

Regardless of differences over the definition of the IoT as an open concept, all definitions retain the Auto-ID Center's seminal concept: object identification; data acquisition; and information management, exchange, and analysis. Differences mainly arise with regard to certain specifics. One is human interaction (the way we interact with other people and the environment). The other is the technology surrounding us, in which objects are typically no longer passive but are smart things, which are increasingly embedding humans into a smart environment.

The term *smart thing* also carries several meanings. In this article, we generally refer to smart things as objects capable of seeing, listening, smelling, thinking, or communicating. This includes intelligent sensors, which not only have measurement capabilities but also include processors that can run real-time computations with measured data and, usually, wireless communication resources for connection to global networks (such as the Internet) or to other sensors (to build up wireless sensor networks).

Advances in microelectronics (due to nanometer-scale technologies) and in microelectromechanical systems opened the door for the development of low-cost, low-power, small intelligent sensors. This development, in turn, enabled massive deployment of such sensors in all kinds of devices and consumer goods, which can then either be connected to data networks and located anywhere or be entirely mobile, as with smartphones and tablets. We are currently living in a world of ubiquitous sensing, which gradually approaches Mark Weiser's ubiquitous computing paradigm [18]. This is one of the multiple scenarios where the IoT is developing [19].

The model based on RFID tags, EPC IDs, and ONS is still one of the most-used architectures in commercial IoT applications.

It is clear that IoT technology, since its origins, has involved many other technologies. Depending on one's viewpoint, the IoT may be analyzed in several different ways. Three viewpoints, different though related, were identified by Atzori [1]:

1) The things-oriented perspective, referring to the technologies for object ID, sensing, and connectivity.
2) The Internet-oriented perspective, encompassing technologies for information management and exchange at the software level (middleware).
3) The semantics-oriented perspective, focusing on representing, storing, connecting, searching, and organizing the information generated by objects.

Depending on the specific viewpoint, different definitions can be given for the IoT [1], [3], [5]. Gubbi et al. [5] added another defining perspective: cloudcentric. The cloudcentric perspective views cloud computing as a unifying framework supporting different platforms for information exchange between sensors and actuators, data analysis, and representation of information.

Considering only sensing capabilities (temperature, speed, acceleration, images, voice, and heart rhythm, to name a few) and the computing power of devices such as smartphones, tablets, and wearable things, the amount of data that can be generated over even very short periods of time becomes overwhelming. The IoT will make sense only if all those data are handled securely and if the analysis and actuations coming from them are executed collaboratively to reach specific goals. This is why some researchers and developers in the context of the IoT focus on data storage and analysis, with the goal of extracting knowledge from such data. The immediate conclusion is that another emerging area, big data, is also closely associated with the IoT, and that it may give rise to a fifth defining perspective, the data-oriented perspective.

IoT Architecture

Apparently, nearly any technology can be related in one way or another to the IoT, and it is expected that new technologies will be even more tied to it. Regardless, a general IoT architecture, compatible with all the aforementioned perspectives, can be defined as consisting of a four-layer vertical structure with a bottom-up data flow [20]–[22] (Figure 2).

Physical objects are located in the bottommost (physical) layer, where they are identified and the data associated with each one of them are generated. Objects should therefore include ID resources (such as RFID tags, bar codes, or infrared sensors) and may be able to identify themselves—e.g., via an ID stored in nonvolatile memory, provided by an embedded hardware core, or via the communications infrastructure, such as a Media Access Control (MAC) address or Bluetooth ID.

The layer above the physical one is known as the *connection layer*. It links the physical objects layer with the middleware layer. It contains the interconnection resources, which (among other factors) depend on the communication system [e.g., fourth generation (4G), fifth generation (5G), Wi-Fi, Universal Mobile Telecommunications System (UMTS), Ethernet, Bluetooth, or Zigbee], the protocols, the network topology, the type of sensors used, and whether or not the object has its own network access hardware.

The middleware layer is in charge of storing object information in the databases corresponding to the service provided by each object. This layer also analyzes the data and makes decisions based on the results of this

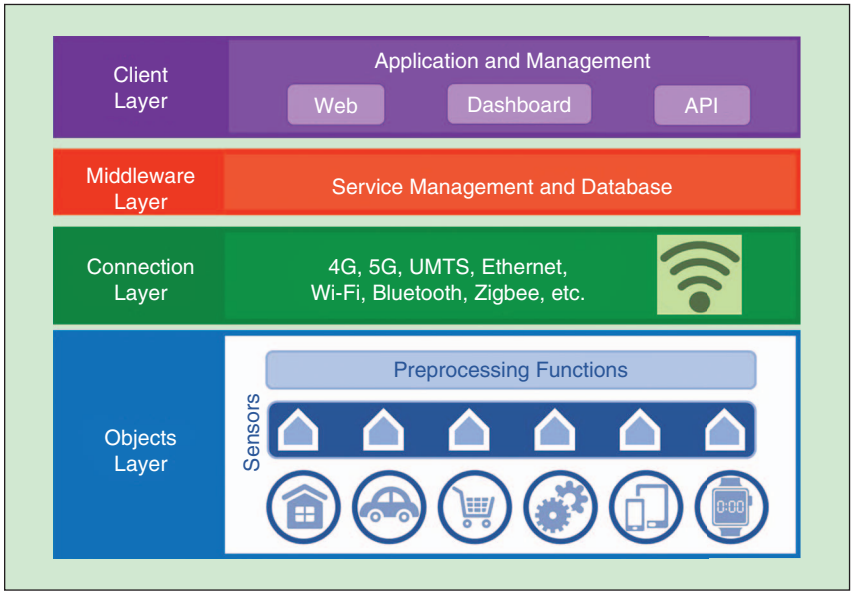

FIGURE 2 – The general architecture of the IoT.

analysis. A key aspect of the middleware is database management. The management platform varies anywhere from big data analytics (for applications requiring parallel processing of huge amounts of data) to a platform supporting complex data-processing algorithms (for applications requiring near real-time decision making).

Reducing data traffic is a key goal of any IoT application. Because of this requirement, in many current systems, objects are equipped with firmware that executes some preprocessing tasks to filter the data to be sent to the middleware through the communication resources.

The topmost layer, called the *client layer*, corresponds to the specific target application (e.g., a smart car, smart city, or smart hospital), and it is in charge of generating results and actuations from the information processed in the middleware layer. In addition, it connects the system to the environment through web-based front ends, dashboards, or application programming interface (APIs), which support M2M communications to interact with other systems.

IoT applications cover both the consumer and industrial areas [3], [22], always aiming at generating added value in terms of increased efficiency (e.g., of a production process, resource management, or energy consumption), improved health and/or safety (e.g., disease prevention or care of children and the elderly), or enhanced quality of life (e.g., smart cities). The widespread deployment of IoT technologies faces many challenges [1]–[4], [22]–[24], with standardization, scalability, security, and privacy being among the most common across all applications. In addition, current IoT systems are very heterogeneous relative to their hardware and software solutions, which, in turn, greatly complicates controls and management tasks. The availability of security, communication, and ID standards is of paramount importance for the development of more efficient, interoperable systems.

Scalability

Scalability is a problem inherent in most technologies. It may refer to the capabilities of systems to support additional requirements during their lifetime (i.e., to extend their functionality) or to the possibility of replicating all or parts of a system, e.g., to build a redundant version or to increase its computing power, throughput, or fan-out. In any case, scalability is closely related to obsolescence. Scalable solutions can easily evolve and adapt to new requirements, so they are better prepared than nonscalable ones to remain longer in the market, allowing the need for new technologies and their associated cost to be reduced.

The IoT is no exception, so scalability is one of the key factors in developing successful IoT systems [25], [26]. The number of mainly heterogeneous things connected to the Internet can reach tens of billions in a few years. Most of them will consist of both hardware and software parts, so both need to be scalable. Therefore, IoT applications must be capable of dealing with larger physical or virtual spaces; keeping pace with an increasing number of connected sensors and actuators; handling larger amounts of data to be acquired, processed, stored, transmitted, or analyzed; and using a higher number of computing nodes.

Finally, scalability is an issue that magnifies the problems associated with all of the challenges described previously: identification, access authentication, security, privacy, connectivity, data storage, management, processing, provision of services, and maintenance, to name a few.

Security, Integrity, and Privacy

The security, integrity, and privacy of the information being transmitted across data networks represent one of the key concerns of industrial IoT (IIoT) applications [2], [20], [23].

Reinforcing security to avoid attacks is one of the major challenges for the IoT. The huge number of devices connected to the network, which are easily accessible because of the IoT's inherently open nature, poses significant threats in this regard and makes the effect of any successful attack likely to be very quickly extended to many systems. Theft or loss of sensitive information, locked terminals, or devices whose operation mode is hacked or that are even physically damaged may imply large economic losses and inconvenience or serious threats to many people on a global scale. Recent examples of massive attacks affecting government systems and large companies all over the world are fresh in everyone's mind.

Privacy/confidentiality is another significant challenge. The massive deployment of the IoT implies a tremendous amount of data traveling across public networks. In many cases, these data are private or confidential, but

the open nature of the IoT makes it feasible that they can be accessed by unauthorized people or systems. IoT systems must ensure that only authorized entities can access or modify private or confidential data. In this context, the design of encryption algorithms is a highly active research area. Practical examples in the security area include

- traffic control in smart cities, where cars provide real-time information about their location and destination, which can be used to control traffic lights; a security breach in such systems could cause anything from serious traffic jams to terrible accidents
- a smart hospital where patients' data can get lost or incorrectly stored because of network errors, or worse, stolen due to network security breaches
- meteorological stations providing information to guide both citizens and emergency services in case of natural disasters.

Some of the software vulnerabilities of IoT systems affecting customer security and privacy were analyzed in [27]. Security and fragmentation (different communication protocols, some of them proprietary, being used in the same application) were identified in [28] as two key problems to be addressed in wireless communications for the IIoT. Chen elaborated on how decentralized firewalls, typically used for malicious traffic detection, do not represent a satisfactory choice in IIoT systems [4]. Instead, he proposed that security be embedded within the objects themselves. However, many of these objects are typically simple, low-cost devices, lacking the resources and capabilities required to implement strong security or cryptographic algorithms. Therefore, the design of low-cost cryptoalgorithms and hardware accelerators represents a key area for the success of the IIoT.

When it comes to the security of the IoT, there is a recognized need for specific legal regulations to be put in place. Because of the IoT's inherently interconnected nature, the need for consistent regulations across different countries becomes even more apparent. Weber [24] analyzed the legal challenges of the IoT, related to the right to access information and the use, restriction, or prohibition of some IoT mechanisms. He states the need for international regulatory approaches to be developed to enable the means to intercept attacks and ensure data authentication, access control, and user privacy. He concludes that self-regulation by companies or nations is not enough and calls for an international body supported by public entities that must, at a minimum, supervise the IoT.

Current IoT systems are very heterogeneous relative to their hardware and software solutions, which, in turn, greatly complicates controls and management tasks.

Growing Energy Needs

From a hardware viewpoint, significant challenges are posed with the need for deployment and maintenance of trillions of sensing nodes, e.g., in buildings [29], [30]. For instance, replacement of batteries in embedded sensors may become a very difficult problem in production systems. This leads to the need for low-power sensors, long-life batteries, and energy-efficient designs.

The need for combining hardware and software platforms is inherent in IoT systems. Challenges related to hardware–software integration include the definition of the system architecture, the communication protocols, the data-processing and analysis algorithms, and the hardware implementation of the middleware layer and how to access it. The current trend is the use of highly integrated, low-power embedded systems.

Another important challenge is the efficient use of bandwidth and energy in data networks when large numbers of both static and nonstationary devices are connected to them. Another problem that may occur is related to addressing objects. As the number of connected nodes increases, it eventually may exceed the addressing capabilities of current network management protocols. The need for a commonly accepted service description language and powerful service discovery methods for ease of integration of objects in a large-scale IoT infrastructure is analyzed in [27].

Standardization of communication mechanisms is another key factor to be considered. Wollschaeger et al. analyzed the evolution of industrial communications and the way they may be effected in the near future by massive IoT deployment as well as by other paradigms, such as CPSs or the tactile Internet [31]. In that article, the authors analyze the advantages and limitations of 5G networks in this context and identify management of complexity and heterogeneity as well as network organization on a logical level as the main challenges of future industrial communications.

FPSoCs and the IoT

With the continuous evolution of FPSoC architectures, it has become evident that they can now support even whole hardware/software systems. In this context, flexibility, scalability, power efficiency, security, and printed circuit board (PCB) optimization represent some of the advantageous features of FPSoC architectures.

FPGA fabrics consist of a plethora of user-configurable logic and interconnect resources, allowing any digital functionality to be implemented (limited only by the resources available in a given device) and its reconfiguration (full or partial), even during system operation. These unique features make these devices the most flexible hardware platforms on the market, capable not only of implementing many different functionalities using the same hardware

but also of adapting existing systems to new operational requirements at an affordable cost and with a short time-to-market release.

Parallelism is another significant advantage of FPGAs. The distributed nature of the logic and interconnect resources in an FPGA fabric, together with the inherent concurrency of the hardware, allows several blocks operating in parallel (with either the same or different functionalities) to be implemented on a single chip, resulting in functional operating speeds/throughput not achievable with other technologies—or at least not with the same limited cost and complexity.

Standard logic resources are complemented with a plethora of specialized hardware blocks, such as fast memories [single- or dual-port RAM, read-only memory, and first in, first out (FIFO)]. Controllers for external memory chips, frequency synthesizers (usually called *clock managers*), complex arithmetic and digital signal processing (DSP) blocks (some capable of operating in floating point), and transceivers supporting a wide range of communication protocols or serial interfaces [serial peripheral interface (SPI), I2C, and USB]. All of these combined provide highly efficient implementation platforms for applications in domains like communications or DSP, which were, until very recently, strongly coupled with other specific implementation technologies.

Notwithstanding all of this, the big leap forward happened in the last five years with the advent of FPSoC devices combining FPGA fabric with several high-performance embedded hard processors. The idea of associating embedded processors with configurable logic (typically used for hardware acceleration of time-critical tasks) is not new. Back in 1999, the same year the Auto-ID Center was created and the first initiatives toward the IoT emerged, FPGA vendors started to provide soft processor cores, i.e., processors implemented using the standard logic resources of the FPGA. However, the performance of soft processors is very limited, particularly if compared against hard processors, so their success among designers was also limited. Still, their flexibility to be customized (compared to the rigid structure of hard processors) according to performance, complexity, or cost goals makes them useful for some applications.

At the same time that microelectronics technology was continuing its evolution toward the nanometer scale, embedded systems were gaining significance in the global digital design market. The performance of embedded processors continuously increased, and embedded hard processors entered the market. The evolution of FPGA-based embedded platforms is depicted in Figure 3 [32]. There has been a fast, tremendous jump from

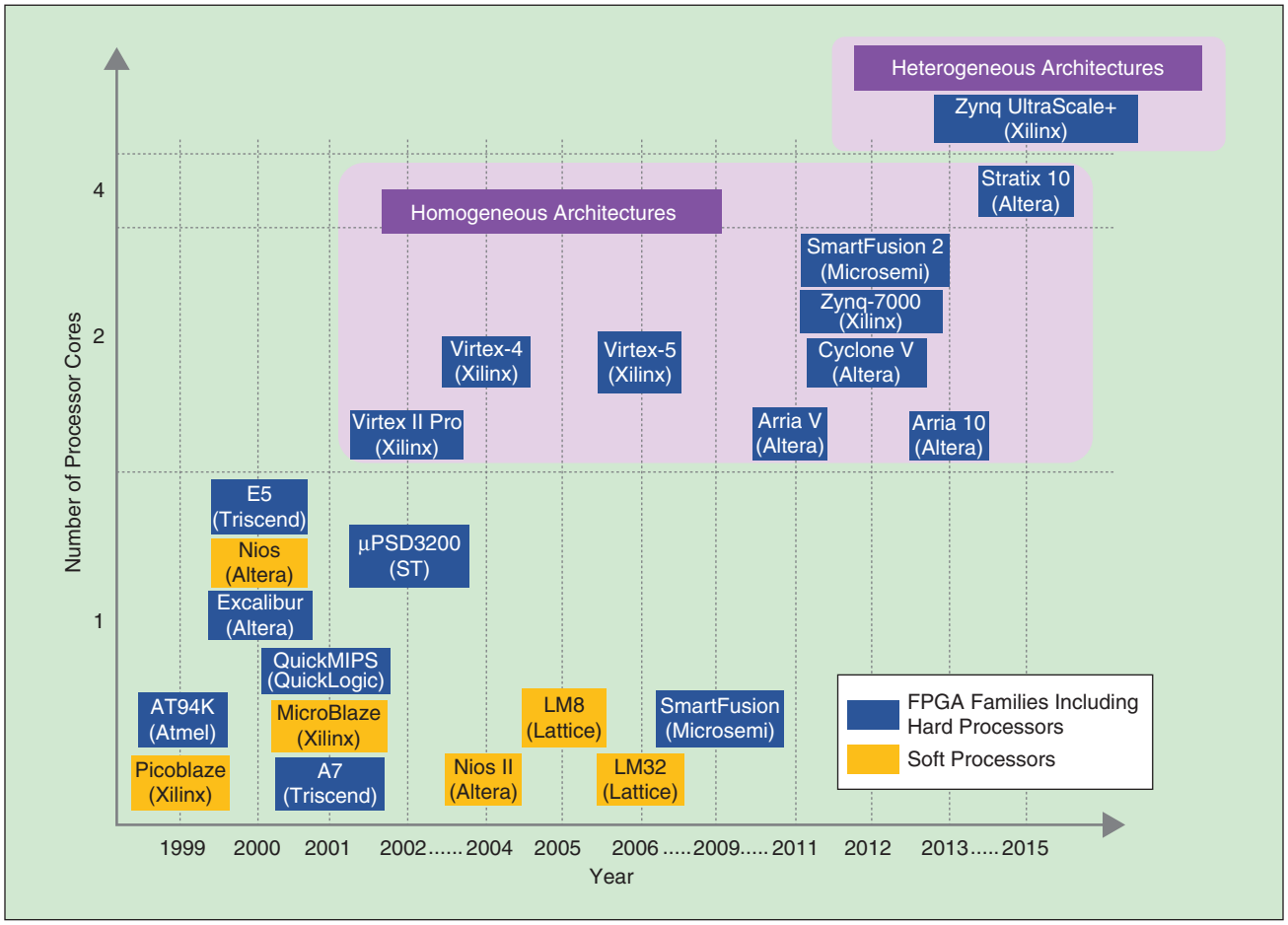

FIGURE 3 – The evolution of FPSoCs.

devices that include a single general-purpose hard microcontroller core to current ones that include more than ten powerful hard processor cores on a single FPSoC chip. That is, FPSoCs evolved from a single-core architecture to multicore homogeneous systems (i.e., including multiple identical processors) and finally to heterogeneous systems (i.e., with different processors targeting different specialized tasks and supporting different operating systems). This is a perfect fit for the inherently heterogeneous nature of IoT applications, which are becoming increasingly complex. Here, FPSoCs may be in charge of specialized tasks, many times to be executed in parallel or requiring the combination of different types of processors, e.g., microcontrollers, digital signal processors, and graphics processing units (GPUs). The advantages of having all the required resources on a single chip are evident. Most hard processors in current FPSoCs are 32-b reduced instruction set computer (RISC) cores from Advanced RISC Machine (ARM): the Cortex-M3, Cortex-A9 (dual core), and Cortex-A53 (quad core).

To give an idea of the capabilities of current state-of-the-art FPSoCs, Figure 4 shows the hard processor architecture of the Xilinx Zynq UltraScale+ MPSoC devices, which includes one ARM Cortex-A53 quad-core processor, one ARM Cortex-R5 dual-core processor, and one ARM Mali-400 MP2 GPU, resulting in a heterogeneous multicore hardware architecture. These processors can work in split (independent) or lock-step (parallel) mode, the latter intended for safety-critical applications requiring redundant systems. Of course, in addition to the processors and their hardware peripherals, the devices include an FPGA fabric with both standard logic and specialized hardware resources.

FPSoCs are particularly suitable for IIoT applications. The main requirements of the objects layer are related to resources for data acquisition and preprocessing, namely analog-to-digital converters (ADCs), memories, and computing power. Mixed-signal FPGAs exist on the market, including analog front ends. These consist of ADCs and associated circuitry and, in some cases, digital-to-analog converters (DACs) and signal conditioning circuits. Together with the usual digital resources of FPGAs, these devices provide all the elements required to interface with sensors and carry out data

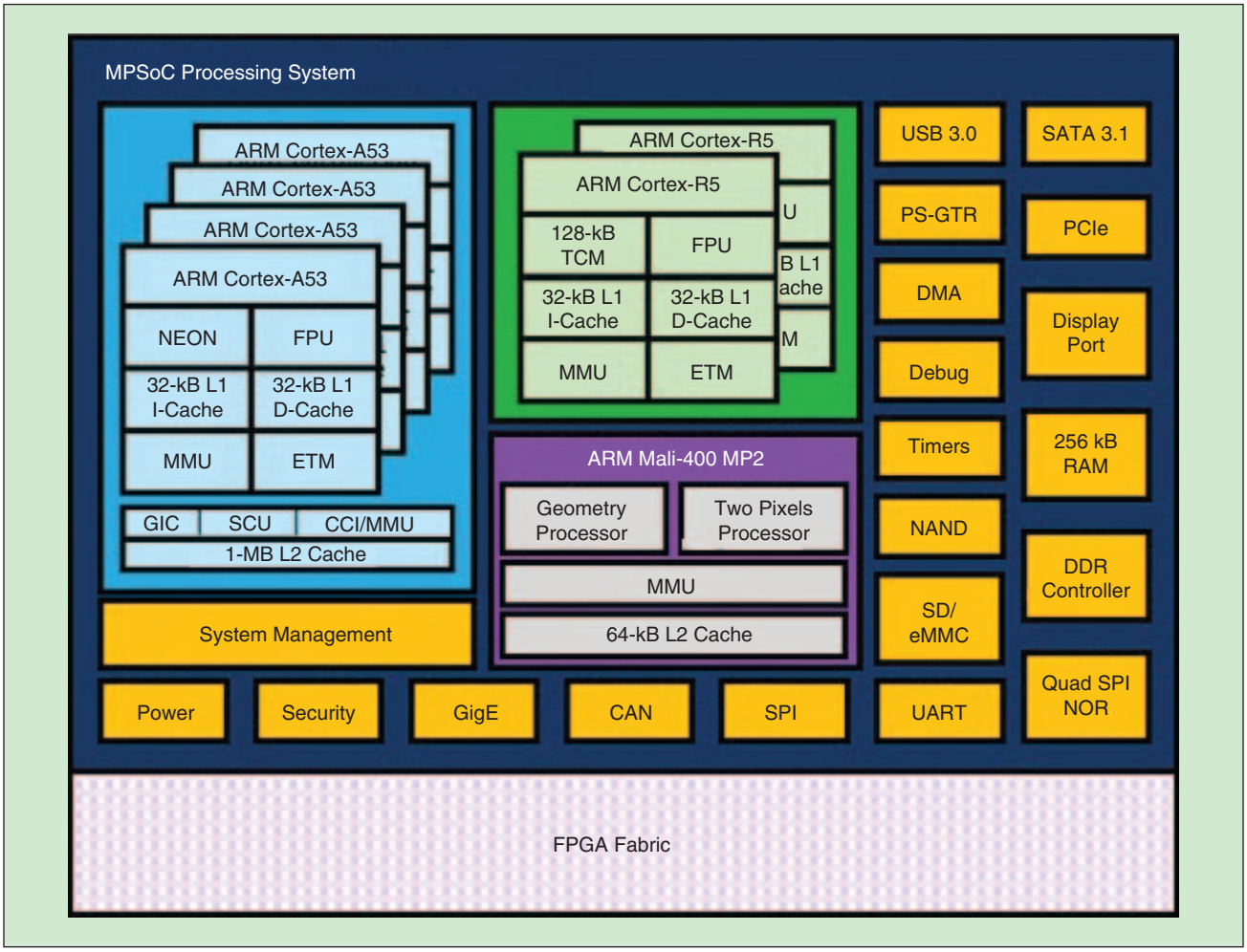

FIGURE 4 – The architecture of a Xilinx Zynq UltraScale+ MPSoC. CAN: controller area network; CCI: cache coherent interconnect; gigE: gigabit Ethernet; UART: universal asynchronous receiver/transmitter; SD/eMMC: secure digital/embedded multimedia card; DMA: direct memory access; PS-GTR: processing system transceivers; SATA: serial ATA; PCIe: PCI express; DDR: double data rate; TCM: tightly coupled memory.

A significant advantage of FPGAs with respect to pure software solutions regarding security is hardware cryptoacceleration.

acquisition and processing. Specific devices exist intended to operate as sensor hubs. These are coprocessing systems aimed at relieving a host processor from sensor management tasks, resulting in faster, more efficient, and less power-consuming processing (in the range of tens of microwatts).

Quicklogic's EOS S3 Sensor Processing SoC is intended to support a wide range of sensors in mobile devices, such as high-performance microphones or environmental, inertial, or light sensors. Its basic architecture is shown in Figure 5. It consists of a multicore processor that includes a set of specialized hardware blocks and an FPGA fabric. Control and processing tasks are executed on two processors, an ARM Cortex-M4F and a Flexible Fusion Engine (FFE), which is a proprietary DSP-like (single-cycle multiply–accumulate operation) very long instruction word (VLIW) processor. The ARM core is in charge of general-purpose processing tasks and may host the operating system in case it is necessary to use one. The FFE processor is in charge of sensor data-processing algorithms, such as voice triggering and recognition, motion-compensated heart rate monitoring, indoor navigation, pedestrian dead reckoning, or gesture detection. It supports in-system reconfiguration and includes a change detector targeting always-on context awareness applications. A third processor, the sensor manager, is in charge of initializing, calibrating, and sampling front-end sensors (accelerometer; gyroscope; magnetometer; and pressure, ambient light, proximity, gesture, temperature, humidity, and heart rate sensors) as well as data storage.

All FPGAs include full-duplex transceivers compatible with the most advanced industrial serial communication protocols, which can provide support to the connection layer of an IoT architecture. Data transfer rates of up to 56 Gb/s can be achieved by some devices, while the number of transceivers per device can be in excess of 100 (e.g., up to 144 in the Intel Stratix 10 GX family and up to 128 in Xilinx's Virtex UltraScale+ FPGAs). Some of the supported protocols are Gigabit Ethernet, PCI express (PCIe), 10GBASE-R, 10GBASE-KR, Interlaken, Open Base Station Architecture Initiative, Common Packet Radio Interface, 10-Gb Attachment Unit Interface, 10GH Small Form-factor Pluggable Plus, Optical Transport Network OTU3, and DisplayPort. In addition, multicore FPSoCs include resources to interconnect embedded hard processors among them, and they include support for standard network connections such as Ethernet.

As the two top layers of the IoT architecture are software layers; they can be easily implemented in the hard processors embedded in FPSoCs. Since these are commercial processor architectures supporting a wide variety of general-purpose and specialized operating systems (e.g., real time), it is very easy to migrate existing software applications to them from other software platforms.

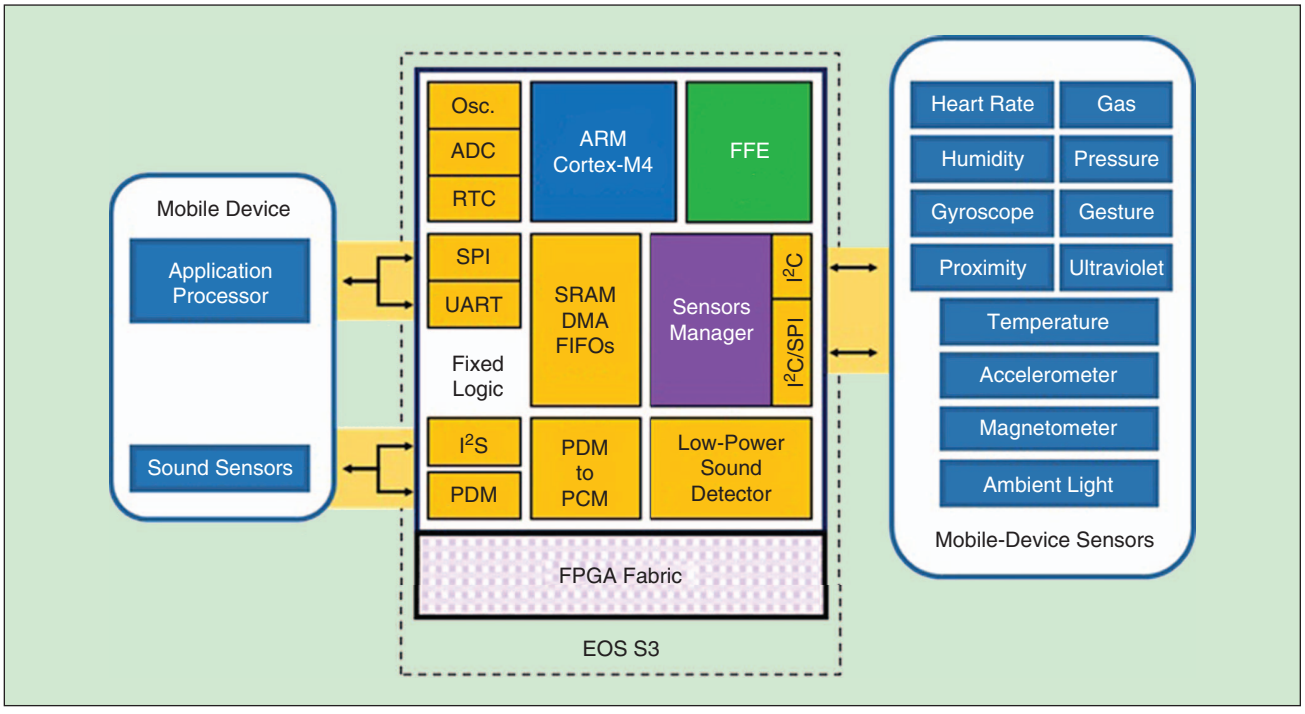

FIGURE 5 – An EOS S3 block diagram. Osc.: oscillator; I2S: integrated interchip sound; PDM: pulse-density modulation; PCM: pulse-code modulation; SRAM: static RAM; I2C: inter-integrated circuit.

As stated in the "Security, Integrity, and Privacy" section, security is one of the most critical aspects of IoT applications. When using FPSoCs, it is necessary to protect not only the software run by the processors but also the hardware modules implemented in the FPGA fabric. All FPSoCs based on ARM processors (most current devices) include ARM's TrustZone technology [33], [34], which provides algorithms for protecting data in the processor cores, their peripherals, and the buses connecting all of them. ARM processors connect with the FPGA fabric through ARM proprietary AXI4 buses, so protection is also provided at the processor–FPGA connection level.

A significant advantage of FPGAs with respect to pure software solutions regarding security is hardware crypto-acceleration. Public-key cryptography implementation in low-cost software platforms (the only ones suitable for massive deployment in an IoT context) may become unacceptably slow, particularly if hard real-time operation is required. This problem can be alleviated by executing cryptography algorithms in the FPSoCs' hardware portion.

Since reduced size and cost are significant requirements in many IoT applications, they must be carefully analyzed when considering the use of FPSoCs. Currently, there is a wide variety of device families (low-end, mid-range, and high-end FPSoCs) targeting different design requirements. Within each of these families, there are devices with different packages and input/output pin counts, as well as different amounts of embedded memory and other specialized hardware blocks. This results in different cost, power consumption, and performance. In the latest devices from Xilinx (the 7 Series and the Zynq-7000 and Zynq UltraScale+ families), sizes range from 8 mm × 8 mm to 45 mm × 45 mm [35], [36]. Therefore, they are not suitable for highly miniaturized systems. They are, however, a good alternative in the vast majority of applications where microcontrollers or digital signal processors are currently used, because of the huge amount of available embedded resources (thanks to the use of nanometer technologies, typically ranging from 28 to 14 nm).

FPGAs and, in particular, FPSoCs are very suitable devices to support the typical needs of many IoT/IIoT applications.

For example, the Xilinx Artix-7 XC7A50T device (28-nm technology) includes within a 10-mm × 10-mm, 250-pin package 52,160 logic cells, 65,200 flip-flops, 2,700 kb of RAM, 120 DSP blocks, a PCIe Gen2 block, an ADC, and up to 6.6-Gb/s transceivers. The cost as of June 2017 in an online retail store was about US$50, which would be reduced if large quantities were directly purchased from the FPSoC vendor.

Another example, in this case a high-end device, is the Zynq UltraScale+ XCZU3EG, whose 23-mm × 23-mm, 784-pin package includes (see Figure 4) a quad-core ARM Cortex-A53 superscalar processor, a dual-core ARM Cortex-R5 real-time processor, an ARM Mali-400 graphics processor, 154,000 logic cells, 141,000 flip-flops, 7.6 Mb of RAM (in addition to the processors' own memories), 360 DSP blocks, and four 6-Gb/s transceivers. The cost of a full development kit based on this device, as listed on the vendor's website in June 2017, was US$640.

As may be expected, prices vary greatly depending on the features and performance of the devices. A quick search at an online store (again in June 2017) revealed a price range between US$7 (for 500 units of a very basic device) and about US$10,000 (for a single unit of the most advanced devices).

It would be interesting to make a generic comparison between the cost of a single-chip FPSoC solution and an alternative using several interconnected discrete processor and logic chips. However, such a generic and complex comparison is very difficult to make because different applications require different bills of material (BOMs), which may include analog and mixed-signal components (e.g., amplifiers, ADCs, or DACs); sensors; power supplies; thermal management components; memory components; passive components; safety, security, and reliability components; and the PCB. According to [37], current FPSoC devices allow the overall BOM cost of a system to be reduced in several ways:

- reduction in the number of components, thanks to SoC integration
- reduction in the cost of auxiliary components, e.g., dc/dc converters, thanks to the reduction in the number of power rails required to supply the chips
- reduction in the PCB development cost, thanks to state-of-the-art packaging.

In our opinion, no single implementation platform can claim superiority over all other possible alternatives for IoT applications, but the analysis presented so far—even if brief, considering how wide the field of reconfigurable devices is—allows us to conclude that FPGAs and, in particular, FPSoCs are very suitable devices to support the typical needs of many IoT/IIoT applications, such as sensor fusion, connectivity, processing power, scalability, heterogeneity, and security. The main current drawbacks for a wider penetration of FPSoCs into the IoT market are the size (for miniaturized systems, as mentioned previously), the cost of high-end devices, and the limited performance provided by current high-level design tools, which prevents designers not experienced with low-level hardware design flows (mainly based on hardware description languages) from getting all the performance that can be achieved thanks to the advanced features of the devices.

FPSoC-Based IoT Applications

FPSoCs are becoming leading players in several IoT areas, such as secure computing, computational intelligence, cloud computing, big data, and connectivity. This section will provide a brief overview of some of the existing FPSoC-based solutions in these areas.

FPSoCs are becoming leading players in several IoT areas, such as secure computing, computational intelligence, cloud computing, big data, and connectivity.

Some of the work dealing with secure computing is reported in [38]–[40]. IBM presented in [38] the IBM 4767/Crypto Express5S, a highly programmable cryptographic coprocessor environment based on an FPGA and an application specific integrated circuit (ASIC). A hardware accelerator for the Somewhat Homomorphic Encryption scheme, based on an FPGA, is presented in [39] and is capable of carrying out data cipher operations in cloud computing systems.

An FPGA is used in [40] to implement an architecture tailored for fast signature verification in IoT applications. By using several cores in parallel, the system achieves 2,040 double scalar multiplications per second.

Physical unclonable functions (PUFs) are security circuits that calculate a unique signature ID for an integrated circuit from its physical characteristics (based on the process variations inherent in its manufacturing process). This ID enables circuits to be authenticated and traced in IoT systems. FPGA-based PUF implementations are reported in [41] and [42]. The solution in [42] takes advantage of the reconfiguration capabilities of FPGAs to adapt to the variations inherent in IoT systems.

Regarding deep learning, Intel researchers analyzed the implementation of deep neural networks (DNNs) in FPGAs [43]. DNNs feature a high level of parallelism and require floating-point matrix multiplication. Although GPUs are currently the most widely used platforms as hardware accelerators for DNNs, the new FPGAs fabricated in 14-nm technologies may become a better alternative, because they include thousands of floating-point DSP units and on-chip RAM blocks and achieve higher energy efficiency. In addition, according to Nurvitadhi et al. [43] the most recent DNN algorithms exhibit "irregular parallelism on custom data types, which are difficult for GPUs to handle but would be a great fit for FPGA's extreme customizability." The authors of that work also implemented several DNN algorithms in FPGAs and compared their performance to that of the same algorithms implemented in a high-performance Titan X Pascal GPU. The results showed that FPGAs can perform 60% faster than the GPU, while being 2.3 times better in performance per watt.

FPGA implementation of convolutional neural networks (CNNs) using OpenCL were presented in [44] and [45]. The AlexNet CNN benchmark executed in the hardware accelerator proposed in [44] processes 1,020 images/s, achieving 23 images/s/W. These results are comparable to those achieved using a Titan X GPU. The CNN hardware accelerator described in [45] achieves 866 gigaoperations per second (Gop/s) when working in floating point at 370 MHz and 1.79 teraoperations per second when working with 16-b fixed-point performance at 385 MHz.

The FPSoC implementation of Spark, one of the most widely used frameworks for data analytics, is reported in [46]. Such implementation is evaluated with a machine-learning application based on logistic regression, achieving up to 11 times the acceleration with regard to the execution time in the ARM cores. The FPSoC implementation of a recurrent neural network for speech recognition is presented in [47], achieving 282 Gop/s when working at 200 MHz, with a power dissipation of 41 W. It is 43 and three times faster than a Core i7 5930k CPU and a Pascal Titan X GPU implementation, respectively, with 40 and 11.5 times higher energy efficiency, respectively.

Examples of the use of FPGAs in cloud computing applications were described in [48]–[50]. An FPSoC is used in [48] to accelerate massive electrocardiogram signal processing, including QRS detection, feature extraction, and classification. The results exhibit a 38 times performance increase and a 142 times improvement in energy efficiency with regard to commercial servers. An FPGA-based layer between the network switches and servers to accelerate applications in hyperscale data centers is presented in [49]. The architecture where such a layer is used has been deployed at hyperscale in Microsoft's production data centers worldwide. A framework for creating network FPGA clusters in a heterogeneous cloud data center is proposed in [50]. The framework reserves computing devices, creates network connections, retrieves MAC addresses, generates the bitstreams, programs the devices, and configures them with the appropriate MAC addresses. In this way, ready-to-use network devices may be created that can interact with any other network device in the data center, including CPUs, GPUs, and IoT devices, such as sensors.

A big data application that uses FPGAs for data analysis acceleration in genome sequencing problems is presented in [51]. The experimental results demonstrate that the proposed platform could efficiently accelerate the next-generation sequencing problem with satisfactory accuracy and acceptable hardware cost.

FPGAs are increasingly used in connectivity applications. A secure IEEE 802.15.4 [59] transceiver (mainly consisting of an FPGA and an ASIC) that mitigates multiple attacks simultaneously by using a physical layer encryption approach is presented in [52]. An IEEE 802.15.4 accelerator for heterogeneous wireless sensor systems implemented in an FPSoC is proposed in [53]. In this solution, the processor supports wireless connectivity, whereas the third level of filtering specified by IEEE 802.15.4 is implemented in the FPGA fabric. A vehicle-to-vehicle IEEE 802.11p [60] communication system based on a FPGA is described in [54], which improves the packet error rate of data transmissions. Finally, the feasibility of using FPGAs for supporting environmentally aware Web services is analyzed in [55].

Prospects for the Near Future: New Generations of FPSoCs

Memory access is likely to become a performance bottleneck in IoT applications because of the large amounts of data that may be involved in them. The connection of separate processor and memory chips is inefficient in terms of both bandwidth and energy consumption per bit [56]. This is further aggravated with the increase of memory requirements, which may imply the need for several memory chips to be used. In this scenario, power consumption can reach the range of tens of watts.

These problems can be mitigated by the integration of processor and memory within the same package. This solution is implemented in the Intel Stratix 10 MX dynamic RAM (DRAM) System-in-Package FPSoC family (Figure 6). These devices combine high-performance FPGA fabric (capable of operating at up to 1 GHz) with high-bandwidth DRAM memory blocks (up to 16 GB), achieving up to a 1-TB/s bandwidth (a ten times increase with respect to solutions based on discrete chips). This comes in addition to reduced PCB complexity and power consumption at the board level. Intel Stratix 10 MX devices also include a quad-core 64-b ARM Cortex-A54 processor system and peripherals running at up to 1.5 GHz, thousands of variable-precision DSP blocks and multipliers, and up to 96 full-duplex transceivers working at up to 30 Gb/s, thereby providing a high level of concurrency and processing power.

The advent of 5G wireless technology is expected to boost the development of IoT applications because of its high-throughput, low-latency, real-time responsiveness, and reliable connectivity, providing consistent user experience anytime, anywhere [57]. To make the most of this new technology, as much of the RF signal processing as possible should be moved from the analog to the digital domain. However, this comes at the cost of increased power consumption because of data transmission between the RF and digital front ends and the need for high-sample-rate converters.

The different requirements of radio access networks in a 5G scenario imply that RF front ends must be flexible enough for hardware platforms to be easily adaptable to implement different radio solutions. A possible solution could be based on Xilinx's proposal of special FPSoC devices (called *RFSoCs*, Figure 7) combining RF-sampling data converters [12-b ADCs with rates up to 4 gigasamples per second (GSPS) and 14-b DACs with rates up to 6.4 GSPS], FPGA fabric, DSPs, general-purpose processors, and optimized RF signal-processing blocks [58].

Closing Discussion

The main thrust of this article was to discuss answers to the question "Can current reconfigurable platforms play a key role in the development and deployment of IoT technology?" For us, the answer is "Definitely, yes." The arguments supporting this statement are that reconfigurable platforms feature the following elements:

- *Versatility*: A wide variety of logic resources that are available to support the development of any kind of hardware/software embedded system as well as most of the communication protocols used in IoT/IIoT applications.
- *Flexibility*: This stems from the reconfiguration and parameterization capabilities that allow systems to be readily updated to include new features or be adapted to new operating conditions, communication protocols, and regulations. Another advantage is the ability for real-time reuse of logic resources through reconfiguration, targeting reduction of power consumption and size.
- *High performance*: By parallelizing operation, high throughput can be achieved.
- *Security*: FPSoCs are ideal platforms for cryptoacceleration.
- *Scalability*: The large amount of available logic resources, combined with reconfiguration capabilities, enables system scalability without compromising performance.

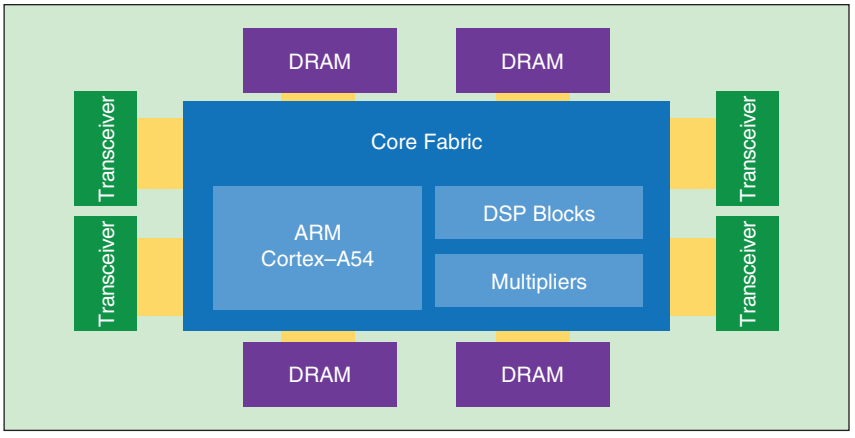

FIGURE 6 – The Intel Stratix 10 MX architecture.

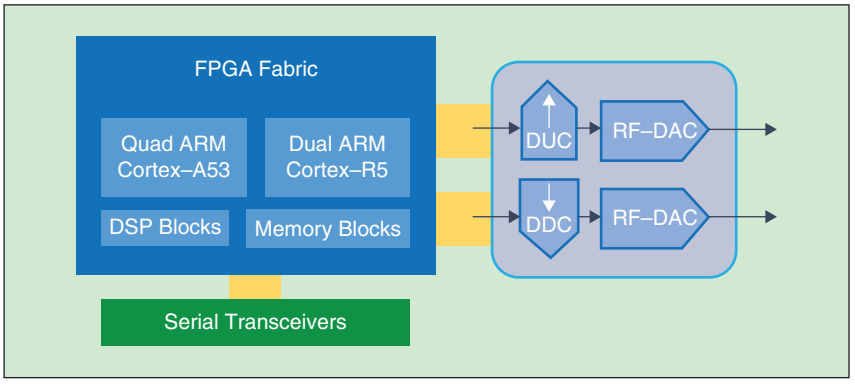

FIGURE 7 – The Xilinx's RFSoC architecture. DUC: digital upconverter; DDC: digital downconverter.

The most advanced FPSoC devices could host not only the lower layers of the IoT architecture but also the middleware and application layers.

- *Ease of design*: Vendors provide designers with an ecosystem of design and verification tools, which dramatically simplify system implementation. Until recently, designers required advanced knowledge of hardware description languages. Currently, high-level synthesis tools (using C++ or OpenCL) are available that simplify and speed up design cycles, though at the expense of some performance loss.

These features perfectly comply with the requirements of IoT systems regarding hardware support in the lower layers of the architecture (the objects and connection layers) for data acquisition, preprocessing (firmware), and communication with the upper layers (software). In fact:

- The availability of embedded memory blocks and of many multiply–accumulate units connected through minimum-delay lines enables the implementation of computing circuits with very high throughput, not achievable by conventional DSP chips.
- Integrated transceivers supporting a wide variety of communication protocols ensure seamless connection of sensing and actuating nodes to any type of network (e.g., the Internet or industrial networks).
- Reconfigurable logic allows the hardware to be tailored to the requirements of different applications, and the ability of some devices for partial reconfiguration (even at runtime) allows functionality to be changed or updated with no need for any change in the physical system itself (e.g., chips or PCBs).

The most advanced FPSoC devices could host not only the lower layers of the IoT architecture but also the middleware and application layers, thanks to the powerful hard processors embedded on them. These range from state-of-the-art general-purpose microcontrollers to real-time processors and GPUs—and current devices include several of them, either of the same or different type.

Cost is the main limiting factor for the deployment of FPSoCs in some IoT applications, but they are a very suitable solution for many others. In our opinion, reconfigurable devices will have an increasing penetration in the IoT market, particularly in IIoT applications. Use of the latest fabrication technologies that reduce area and power consumption, in addition to the newest devices that support 5G technology and include much more embedded memory than their predecessors, clearly shows that companies in the reconfigurable logic industries are aiming for a prominent position in the market.

Biographies

María Dolores Valdés Peña (mvaldes@uvigo.es) received her M.Sc. degree from the Universidad Central de Las Villas, Santa Clara, Cuba, in 1990, and her Ph.D. degree from the University of Vigo, Spain, in 1997, both in electrical engineering. She is an associate professor in the Department of Electronic Technology, University of Vigo. Her research interests include the design of reconfigurable systems based on field programmable gate arrays applied to data acquisition and conditioning systems, digital signal processing and control, wireless sensors networks, and field-programmable systems-on-chip for industrial applications. She authored more than 120 journal and conference articles. She is an IEEE Industrial Electronics Society member and a Member of the IEEE.

Juan J. Rodriguez-Andina (jjrdguez@uvigo.es) received his M.Sc. degree from the Technical University of Madrid, Spain, in 1990, and his Ph.D. degree from the University of Vigo, Spain, in 1996, both in electrical engineering. He is an associate professor in the Department of Electronic Technology, University of Vigo. His research interests include the implementation of complex control and processing algorithms and intelligent sensors in embedded platforms. He has authored more than 160 journal and conference articles and holds several Spanish, European, and U.S. patents. He is an IEEE Senior Member and an IEEE Industrial Electronics Society member.

Milos Manic (misko@ieee.org) is a professor in the Computer Science Department and director of the Modern Heuristics Research Group at Virginia Commonwealth University, Richmond. He has over 20 years of academic and industrial experience leading more than 30 research grants focusing on computational intelligence in energy, resilience, cybersecurity, and human–system interaction in buildings and critical infrastructures. He is a cofounder of the IEEE Technical Committee on Resilience and Security in Industry. He has published more than 150 refereed articles in international journals, books, and conferences and holds several U.S. patents. He built his expertise through research on a number of projects funded by industry and the U.S. Department of Energy. He is an IEEE Industrial Electronics Society member and a Senior Member of the IEEE.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] D. Miorandi, S. Sicari, F. de Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sept. 2012.

[3] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. – Comput. Inform. Sci.*, vol. 29, no. 2, pp. 1–29, Oct. 2016.

[4] Y-K Chen, "Challenges and opportunities of Internet of Things," in *Proc. 17th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2012, pp. 383–388.

[5] "Internet of Things in 2020: A Roadmap for the Future," INFSO D.4 Networked Enterprise and RFID INFSO G.2 Micro and Nanosystems in cooperation with the RFID Working Group of the European Technology Platform on Smart Systems Integration, Sept. 2008.

[6] J. J. Rodriguez-Andina, M. Valdés, and M. J. Moure, "Advanced features and industrial applications of FPGAs: A review," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 853–864, Aug. 2015.

[7] R. Minerva, A. Biru, and D. Rotondi. (May 2015). Towards a definition of the Internet of Things (IoT). *IEEE IoT Initiative*. [Online].

Available: http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf
[8] SmartAmerica Challenge. (2013, Oct. 23). [Online]. Available: https://www.nist.gov/el/smart america-challenge
[9] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed. Cambridge, MA: MIT Press, 2017.
[10] E. A. Lee, "The past, present and future of cyber-physical systems: A focus on models," *Sensors (Basel)*, vol. 15, no. 3, pp. 4837–4869, Feb. 2015. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4435108/
[11] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. (2017). A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* [Online]. Available: https://doi.org/10.1109/JIOT.2017.2683200
[12] T. Samad, "Control systems and the Internet of Things," *IEEE Control Syst. Mag.*, vol. 36, no. 1, pp. 13–16, Feb. 2016.
[13] M. Moness and A. Mahmoud Moustafa, "A survey of cyber-physical advances and challenges of wind energy conversion systems: Prospects for Internet of Energy," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 134–145, Apr. 2016.
[14] M. A. Pisching, F. Junqueira, and D. J. dos Santos Filho, "An architecture based on IoT and CPS to organize and locate services," in *Proc. 21st Int. Conf. Emerging Technologies and Factory Automation (ETFA)*, pp. 6–9. 2016.
[15] Sun Microsystems. (2003). The Auto-ID Center datasheet. [Online]. Available: https://www.fda.gov/ohrms/dockets/dailys/03/Nov03/110503/03N-0361-emc-000025-02.pdf
[16] EPCglobal. [Online]. Available: https://www.gs1.org/epcglobal
[17] Auto-ID Labs. [Online]. Available: https://autoidlabs.org/
[18] M. Weiser, R. Gold, and J. S. Brown, "The origins of ubiquitous computing research at PARC in the late 1980s," *IBM Syst. J.*, vol. 38, no. 4, pp. 693–696, 1999.
[19] M. Botterman. (2009, May 10). Internet of Things: An early reality of the future Internet. [Online]. Available: http://www.future-internet.eu/publications/view/article/internet-of-things-an-early-reality-of-the-future-internet.html
[20] R. Khan, S. U. Khan, R. Zaheer and S. Khan, "Future Internet: The Internet of Things architecture, possible applications and key challenges," in *Proc. 10th Int. Conf. Frontiers Information Technology (FIT)*, 2012, pp. 257–260.
[21] P. Fremantle. (2015, Oct. 20). A reference architecture for the Internet of Things. WSO2 White Paper. [Online]. Available: http://wso2.com/wso2_resources/wso2_whitepaper_a-reference-architecture-for-the-internet-of-things.pdf
[22] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
[23] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.
[24] R. H. Weber, "Internet of Things: New security and privacy challenges," *Comput. Law Security Rev.*, vol. 26, no. 1, pp. 23–30, Jan. 2010.
[25] E. Mezghani, E. Exposito, and K. Drira, "A model-driven methodology for the design of autonomic and cognitive IoT-based systems: Application to healthcare," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 3, pp. 224–234, June 2017.
[26] J. Venkatesh, B. Aksanli, Ch. S. Chan, A. S. Akyurek, and T. S. Rosing, "Scalable-application design for the IoT," *IEEE Softw.*, vol. 34, no. 1, pp. 62–70, Jan. 2017.
[27] B. Graham, "Addressing IoT Impact on software engineering," in *Proc. Embedded World Exhibition and Conf.*, 2017, pp. 50–52.
[28] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive Internet of Things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 28–33, Mar. 2017.
[29] M. Manic, K. Amarasinghe, J. J. Rodríguez-Andina, and C. Rieger, "Intelligent buildings of the future: Cyberaware, deep learning powered, and human interacting," *IEEE Ind. Electron. Mag.*, vol. 10, no. 4, pp. 32–49, Dec. 2016.
[30] M. Manic, D. Wijayasekara, K. Amarasinghe, and J. J. Rodríguez-Andina, "Building energy management systems: The age of intelligent and adaptive buildings," *IEEE Ind. Electron. Mag.*, vol. 10, no. 1, pp. 25–39, Mar. 2016.
[31] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
[32] J. J. Rodriguez-Andina, E. de la Torre, and M. D. Valdés, *FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics*, Boca Raton, FL: CRC, 2017.
[33] Y. Gosain and P. Palanichamy. (2014, May). TrustZone technology support in Zynq-7000 All Programmable SoCs. White Paper WP-429 (v1.0). [Online]. Available: https://www.xilinx.com/support/documentation/white_papers/wp429-trustzone-zynq.pdf
[34] ARM, Ltd. (2009, Apr.). ARM security technology: Building a secure system using TrustZone technology. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf
[35] 7 Series FPGAs Packaging and Pinout: Product Specification, Xilinx, Inc., San Jose, CA, User Guide UG475 (v1.14), Mar. 23, 2016.
[36] Zynq UltraScale+ MPSoC Packaging and Pinouts: Product Specification, Xilinx, Inc., San Jose, CA, User Guide UG1075 (v1.2), Jan. 2017.
[37] C. Murphy, E. Mohsen, and S. Kolluri, "Reducing system BOM cost with Xilinx's low-end portfolio," Xilinx, Inc., San Jose, CA, White Paper WP460 (v1.0), Mar. 2015.
[38] T. W. Arnold, M. Check, E. A. Dames, J. Dayka, S. Dragone, D. Evans, W. Santiago-Fernandez, M. D. Hocker, R. Kisley, T. E. Morris, J. Petreshock, and K. Werner, "The next generation of highly reliable and secure encryption for the IBM z13," *IBM J. Res. Develop.*, vol. 59, no. 4/5, pp. 6:1–6:13, Aug. 2015.
[39] V. Migliore, M. M. Real, V. Lapotre, A. Tisserand, C. Fontaine, and G. Gogniat. (2016). Hardware/software co-design of an accelerator for FV homomorphic encryp-tion scheme using Karatsuba algorithm. *IEEE Trans. Comput.* [Online]. Available: https://doi.org/10.1109/TC.2016.2645204
[40] Z. Liu, J. Grosschadl, Z. Hu, K. Jarvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the Internet of Things," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 773–785, May 2017.
[41] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer. (2017). Implementation and characterization of a physical unclonable function for IoT: A case study with the TERO-PUF. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* [Online]. Available: https://doi.org/10.1109/TCAD.2017.2702607
[42] A. P. Johnson, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-enabled secure architecture for FPGA-based IoT applications," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 1, no. 2, pp. 110–122, 2015.
[43] E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. G. H. Ong, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, and G. Boudoukh, "Can FPGAs beat GPUs in accelerating next-generation deep Neural Networks?" in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA '17)*, 2017, pp. 5–14.
[44] U. Aydonat, S. O'Connell, D. Capalija, A. C. Ling, and G. R. Chiu, "An OpenCL deep learning accelerator on Arria 10," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA '17)*, 2017, pp. 55–64.
[45] J. Zhang and J. Li, "Improving the performance of OpenCL-based FPGA accelerator for convolutional Neural Network," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA '17)*, 2017, pp. 25–34.
[46] E. Koromilas, I. Stamelos, C. Kachris, and D. Soudris, "Spark acceleration on FPGAs: A use case on machine learning in Pynq," in *Proc. 6th Int. Conf. Modern Circuits and Systems Technologies (MOCAST)*, 2017.
[47] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang, H. Yang, and W. J. Dally, "ESE: Efficient speech recognition engine with sparse LSTM on FPGA," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA '17)*, 2017, pp. 75–84.
[48] X. Wang, Y. Zhu, Y. Ha, M. Qiu, and T. Huang, "An FPGA-based cloud system for massive ECG data analysis," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 3, pp. 309–313, Mar. 2017.
[49] A. M. Caulfield, "Configurable clouds," *IEEE Micro*, vol. 37, no. 3, pp. 52–61, June 2017.
[50] N. Tarafdar, T. Lin, E. Fukuda, H. Bannazadeh, A. Leon-Garcia, and P. Chow, "Enabling flexible network FPGA clusters in a heterogeneous cloud data center," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA '17)*, 2017, pp. 237–246.
[51] C. Wang, et al. "Heterogeneous cloud framework for big data genome sequencing," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 1, pp. 166–178, Jan. 2015.
[52] A. K. Nain, J. Bandaru, M. A. Zubair, and R. Pachamuthu, "A secure phase-encrypted IEEE 802.15.4 transceiver design," *IEEE Trans. Comput.*, vol. 66, no. 8, pp. 1421–1427, Aug. 2017.
[53] T. Gomes, S. Pinto, F. Salgado, A. Tavares, and J. Cabral, "Building IEEE 802.15.4 accelerators for heterogeneous wireless sensor nodes," *IEEE Sensors Lett.*, vol. 1, no. 1, Feb. 2017.
[54] J. A. Fernandez, "Performance of the 802.11p physical layer in vehicle-to-vehicle environments," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 3–14, Jan. 2012.
[55] R. Brzoza-Woch and P. Nawrocki, "FPGA-based web services: Infinite potential or a road to nowhere?" *IEEE Internet Comput.*, vol. 20, no. 1, pp. 44–51, 2016.
[56] M. Deo, J. Schulz, and L. Brown. (2017). Intel Stratix 10 MX devices solve the memory bandwidth challenge. White Paper WP-01264-1.2. [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01264-stratix10mx-devices-solve-memory-bandwidth-challenge.pdf
[57] S. Ahmadi. (2016). Toward 5G: Xilinx solutions and enablers for next-generation wireless systems. White Paper WP476 (v1.0). [Online]. Available: https://www.xilinx.com/support/documentation/white_papers/wp476-toward-5g.pdf
[58] A.Collins. (2017). All programmable RF-sampling solutions. White Paper WP489 (v1.0.1). [Online]. Available: https://www.xilinx.com/support/documentation/white_papers/wp489-rfsampling-solutions.pdf
[59] *IEEE Standard for Low-Rate Wireless Networks*, IEEE 802.15.4, 2015.
[60] *IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE 802.11p, 2010.