International Journal of Security and Its Applications
Vol.7, No.5 (2013), pp.289-304
http://dx.doi.org/10.14257/ijsia.2013.7.5.27

Platforms and Applications in Hardware Security: Trends and Challenges

Edward David Moreno

DCOMP/UFS Departament of Computer Science at Federal University of Sergipe
edwdavid@gmail.com

Abstract

This paper introduces a few research studies being conducted that are using multicore embedded systems, highlighting the challenges and questions relevant to R&D (Research and Development) in computer systems architecture, focusing on design aspects and optimization of embedded systems that needs to run security solutions or cryptographical algorithms using hardware acceleration, aiming for good performance, code optimizations, and the lowest energy consumption. Another aspects are the design of dedicated processors for specific purposes, multicore systems and GPUs for security solutions and specific applications requiring high performance, with the possibility to work with FPGAs or embedded platforms with embedded multicore paradigm.

Keywords: *Specific purpose processors, Embedded Systems, Hardware Security, Performance Evaluation*

1. Introduction

The advances in integration technologies on integrated circuits are making viable integrated systems in a single chip. These systems integrate programmable processors and dedicated hardware components allowing them to be placed in a single chip in order to fulfill performance requirements for multitasking applications.

Projects with the aforementioned description are extremely complex as the decisions made in the beginning can have a strong impact on its final performance, the other reason is that integrated circuits have a rapid insertion in the Market. With new patterns, new application also arises, specially to satisfy this new applications, it is great to be able to provide **programmability**, **reconfigurability** and **scalability** in this sort of device, as great processing speeds can be achieved with small parameters adjustments in the architecture in order to adapt it to an specific purpose application [10, 34, 35].

With alternatives like these at disposal, it is easier for designers to base their chips and systems in a type of architecture that provides efficient and optimized solutions for an array of application domains, only tuning overall reconfigurable elements present in the architecture and system to work with the desired application.

The reconfigurable computing area (using programmable circuits –FPGAs) are being used as a technological alternative that can easily interface with a general purpose processor keeping the performance similar to those of dedicated hardware [17, 42].

With this speed and adaptability, reconfigurable computing has a great potential to be explored specially* in applications that need good performance with rapid prototyping (lowering time and design costs in any designing phase: development, implementation and testing) to work with real functioning parameters from the most modern applications found nowadays.

ISSN: 1738-9976 IJSIA

Most of these applications are industrial, or from the electro-electronics sector in industrialized or in technological modernization countries, with massive impact as an excellent alternative in research and development when associated with the needs of small and medium companies that need, use and make available in any way, circuits and computing digital systems.

With this last reason in mind, it is important to stress that the reconfigurable computing area can bring many academics, industrial and commercial benefits, allowing and making it easy for a larger interaction between researchers, professors, students, professionals and entrepreneurs from the most diverse sectors of economy. Besides, with the new ways of designing and developing integrated circuits brought by the use of FPGAs e SoC, it becomes desirable to use this technology in industrial applications in regions that are in the development stages.

Normally, reconfigurable systems are designed for applications which require data parallelism, high regularity and big throughput. Some examples of this sort of applications are: video compression (discreet transforms as DCT – *Discrete Cosine Transform*, and time estimative), image processing, multimedia, data encryption and transforms used in DSPs (*Digital Signal Processing*) [11, 22, 24, 33].

In this sense, the use of FPGAs in these applications is justified for the following reasons: (i) High Speed: it is possible to optimize speed tuning critical parameters to the system. (ii) Flexible Level of Security: It is relatively easy to adapt new architectures and systems to new insights and optimizations. (iii) Cost/Performance relation: It is possible to select and design functional units, taking into consideration cost and performance. Example: choosing between designing just a few and slow arithmetic units, or fast and varied ones. (iv) Lower Cost: a project that uses FPGAs are relatively low in cost.

The security in hardware area and its integration amongst different security services in embedded systems still need methods, techniques, and most importantly efficient prototypes in performance and power consumption.

For this reason, it is still possible to create, capacitate, teach, promote and develop new methodologies and expand the use of these emergent technologies in applications in fields such as education and industry, which use programmable circuits and other platforms of embedded systems. For this purpose, different R&D groups of embedded systems are focusing on research and development of security solutions (study, design, optimization of power consumption and performance of cryptography algorithm and security services) in hardware and embedded platforms, with strong emphasis in the design aspects of the project, architecture definition and design of processors for specific security applications. As a study case, it is desired to make a fully functional version of an IP of a TPM (Trusted Platform Module) [5] for VANETs networks [39, 40].

This paper presents an overview of different hardware projects and new platforms using multicore architectures focusing in hardware security, and we show and discuss some aspects which could be open for new researches.

This paper is organized in seven sections, and each one of them provides an information area and ends with questions which need to be further investigated. Section 2 presents information on security in embedded systems and specific hardware applied to aspects of information security. Section 3 approaches details of multicore and GPUs uses, highlighting subjects which can be researched in the near future. Section 4 emphasizes on techniques and research aspects of how to diminish and/or optimize the power consumption of computational applications, focusing on the need of tools which helps in the characterization of this consumption.

2. Specific Hardware for Security Solutions

The information security concept is more than wider than simply data protection in logical level. To provide real security it is required to pay attention to several internal and external details. Firstly, there is the need to characterize the system which is going to store the data to be able to identify threats, thus, being able to identify possible solutions.

According to Stallings (2008) and Paar (2011), there is the following subdivision: (i) **Isolated Systems -** The ones which are not connected to any sort of network, and, (ii) **interconnected Systems –** Nowadays, most computers are connected to a computer network, sending and receiving information from other places constantly. In this sort of system, there are potential risks.

One definition for cryptography is the study of mathematical techniques which relate information aspects, as well as confidentiality, information integrity, entrance authentication and data authentication in the origin [29, 30, 33].

On the other side, the cryptographic primitives are elemental structures used to build cryptosystems and cryptographic protocols. Menezes (2011) suggests the following criteria to evaluate such primitives:

(i) Security Level – Normally calculated based in the quantity of needed operations to reach its goal;

(ii) Functionality – The primitives need to be combined to reach information security goals. The ones which better serve these characteristics are determined by the properties of this primitive;

(iii) Operation Method – The use of these primitives depends on the possible operation modes;

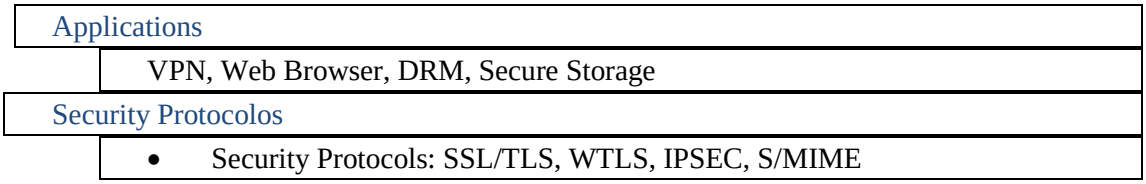
(iv) Performance – The efficacy with the primitive executes its function;

(v) Ease of Implementation – The complexity demanded by the primitive for its implementation in hardware and software.

According to Srivaths (2005), the security functionality can be seen in 3 levels: cryptographic primitives, security protocols and security applications. In Figure 1 it is possible to observe different solution examples, in each one of this levels.

There are several implementations in books and groups researching the area of “hardware security”, however, this area still lacks efficient solutions which take into consideration performance and functionality aspects, strengthening the criteria of ease of implementation. As such, it is desired to focus mainly on level 1, cryptographic primitives, and in some modules of level 2, aiming to use modern cryptographic primitives in the integration of security services, always under the optics of specific and efficient hardware design so that these solutions can be implemented and used in embedded systems, similar to which was achieved in [16].

On the other hand, it is inevitable that Ill-intentioned individuals gather to launch smarter and more efficient attacks. In the same manner, integrated security systems must be created, mainly considering the importance of integrating security services to better prevent and detect strange situations [21, 22].



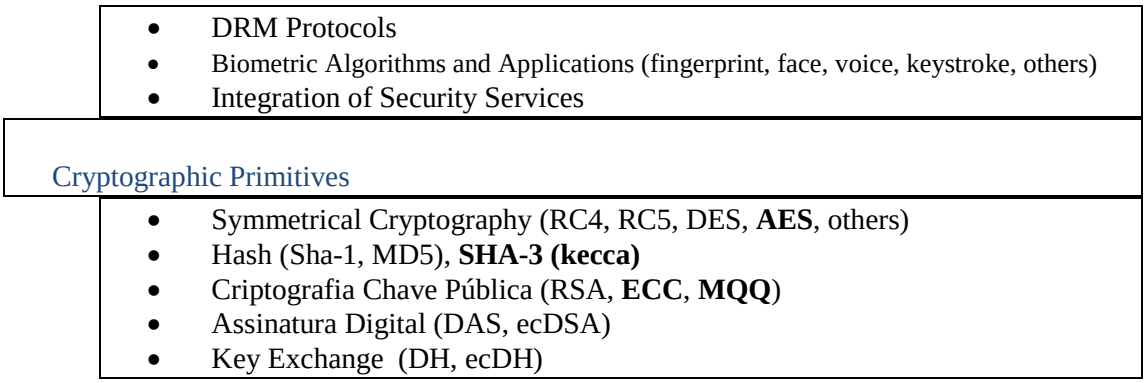


Figure 1. Classification of Security Solutions [37]

The existent integrity models of security services determine which relation between a set of specific security services in order for these to integrate information and prevent or treat anomalies in the system. However, the existent models propose solutions based on a limited set of services, ignoring or not considering the existence of others [41, 22]. Thus, the existence of difficulty in defining a strategy for the creation of an integrated security service model is verified.

2.1. Open Questions in Embedded Systems

Despite the many papers on embedded systems, there are a lot of questions yet to be answered which allows focus on research in specific architectures for hardware security applications and creating cryptographic solutions and integration of security services, evaluating its performance and power. Solutions in hardware must be efficient, with special emphasis on processors for specific applications and embedded platforms in the security area. In this process, it is also important the use of simulation techniques which can help in the investigation of architectural characteristics of the application in question, because the simulation make it possible to detect thresholds and/or critical performance points and power consumption.

Short to midterm, the following subjects were thought of:

- Study and implementation of algorithms (*e.g.*, cryptography, compression, images, among others) in hardware, specifically FPGAs and SoC, and embedded platforms such as microcontrollers, DSP, sensor networks, platforms with low computational power, when compared to traditional computational systems;
- Design processors for specific applications for hardware security in FPGAs and SoC, efficient in performance and power consumption;
- Design processors for specific applications of signals and image processing, or hardware accelerators, in FPGAs and SoC, efficient in performance and power consumption;
- Have knowledge of, using simulation techniques , execution characteristics of some of these techniques in embedded platforms; to discover critical points of performance and power consumption, which are going to help in the proposal of optimization and improvements in the description of efficient and specific architectures;
- To propose specific architectures for these applications, aiming mainly on the design of processors for specific applications, with prototyping* on FPGAs, in the

security area, image processing, and accelerators in applications of critical performance;

- To characterize the use of memory and power consumption of some of these solutions when executing in embedded systems (embedded processors, microcontrollers, FPGAs, SoC, DSP, and embedded platforms);
- To analyze the possibility to create solutions which contains partial and/or reconfiguration, making use of reconfigurable technologies;
- To design security systems (modern cryptographic algorithms, such as MQQ, ECC – Cryptography with elliptic slopes and HECC – Cryptography with Hyper Elliptic Slopes, hashing of the SHA-3 pattern, integration of security services) in hardware(FPGAs, microcontrollers, DSPs, SoC) efficient in processing, speed, use of memory and power consumption.

3. Multicore in Security Solutions

A multicore processor is made of several processors in a single physical enclosure and its interface with a motherboard. Multicore processors were introduced in several academy and market segments. To Domeika (2008), the basic motivation is performance, because the use of multicore processors can result in a faster execution, higher throughput, and lower consumption necessary in embedded applications.

A multiprocessor system is designed of several processors in a single system. The processors which compose a multiprocessor system can be single core or multicore. Figure 2 shows three different layouts of systems: a single core/simple processor; a multiprocessor system; a multiprocessor system/multicore system.

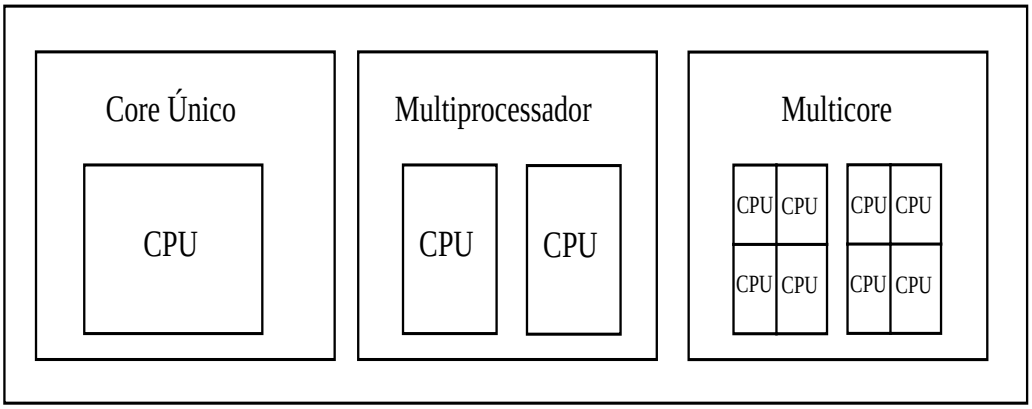


Figure 2. Three Configurations of a CPU-based System [9]

A multicore processor has two or more cores in a single chip. According to Savage (2008) a multicore differs from multiprocessors in the following aspects:: (i) communication latency - multicore usually have lower latency than a multiprocessor; (ii) broad of band – typically lower on a multiprocessor, as processors are physically closer, and; (iii) number of processors - multicore presents less processors than a multiprocessor.

A multicore processor classification is based on the sort of processor in the same chip [36]. A homogenous multicore is made of two or more processors containing the same set of instructions. A multicore heterogeneous is composed of processors with different instruction sets, with the advantage of choosing the right processor for the task in hand.

Following the suggestion made by Inoue (2009) [20], there is another classification or

multicore processors which take into account how the processors are placed in the embedded system as a whole. There are two classifications called symmetric and asymmetric, which differs in terms of access to the memory and the communication between processes.

As Meakin explains in [28], in the symmetric multiprocessing (SMP) the system processors have the same vision of the system and share the main memory. This means that values stored in the main memory can be accessed and modified by each one of the processors. In the asymmetric multiprocessing (AMP), the system view is different between the system processors and memory regions which can be separated implying that two or more processors cannot modify the same memory region.

The communication between processors in an SMP system happens by memory sharing and the operational system provide synchronization functionalities and exclusivity of access to memory regions.

In an AMP system with separated memory regions, the communication happens through messages which make it easy for servers to pack the data and send them from one SO to one that can use the data. A communication pattern that supports communication between processors in an AMP system is the Multicore Communication Application Programming Interface (MCAPI) [28]. Another technique to share values in an AMP system is to provide a memory mapping which uses specialized hardware to allow that two cores can use different address spaces.

According to Domeika (2008) [9], from the main market segments of multicore processors and their possible applications we can cite:

- Infrastructure of wireless telecommunication, in which each cellphone belongs to a processing set that receives and transmits information from another cellphones, traces routes for voice and data through the communication network and to be a link to other cellphones out of the range of covering* for the signal towers. The main benefit of multicore processors in this scenario is the throughput of data, the ability to control more calls. The main challenge in take advantage of multicore processors for many of these applications are the application themselves: they tend to be big which makes it difficult to convert them to multi-thread. Many of these applications run over proprietary operational systems which do not support SMP.
- Industrial Control permits that computers helps in the monitoring and directioning* of factories which manufactured a wide range of products used in the industrialized world. Efficient monitoring, data acquisition and distribution of information in the floor of the factory make the production more efficient. In this context, between the target industrial applications for multicore processors, important applications for test and measure instrumentation, interfaces human-machine, industrial PCs and control systems for automation.
- Military, aerial and governmental are fragments which have the two requisites before mentioned, besides the functional need of the application: firstly, the system must integrate easily with already developed systems and offer flexibility for future developments – This requirement is desirable to use commercial products from the shelves and save money, secondly, these systems need to operate in hostile environments, and still, for a long period of time.
- Data security infrastructure of companies and big corporations require a vast range of network products which can be tagged as a market segment for multicore processors. Some of these network products, like firewalls, virtual private networks, and Secure Sockets Layers (SSL), can demand high computational power in the event of an attack. Detection and defense of mass attacks require analysis of

packages which require high computational power, a scenario in which the use of multicore processors would be advantageous.

- The storage of data has its demand growing and devices capable of storing terabytes of data are becoming more common every day and needed for storing the most varied archives. Clients need to access their data everywhere and they demand security, and a safe backup method with proper data restoration in case of a failure is also required.
- The application segment for the medical and health area proposes that embedded systems deals with patient data, in handheld devices to access stored data, help with diagnosis, complex measures and systems for treatments. Multicore processors can offer high performance and low consumption for applications as systems for generation of 3D images from scanners which require big, fixed and complex machines, making them portable and faster.
- Digital surveillance uses video processing and security cameras to allow processing, analysis and storage of big volumes of data and video information to aid on criminal investigations. In these systems, automated processes can identify and trace points in faces, and also identify suspect movements. The use of multicore processors provides the ability to process high volumes of data with the most sophisticated processing algorithms available.

3.1. Multicores in FPGAs

In this section some academic works in prototyping and use in FPGA devices are presented. A FPGA provides the flexibility needed to the implementation and functional verification of several digital circuit designs [13, 14, 43].

One 4x4 multicore in a mesh network directed towards the use in FPGAs in the Altera company is presented by Minhas (2009) [31]. An interconnection system is used to connect 16 Nios II processors placed in 4 Altera Stratix II FPGA chips. In each FPGA are mapped 4 Nios II processors connect through an Address-Mapped Resource Network Interface (RNI). A hardware abstraction layer (HAL) was described, based on the message passing interface (MPI) pattern and the applications uses the HAL layer to communicate with the RNI. The RNI transfers messages with a maximum size of 512 bytes with 32 bits of data and 20 bits of header. The other authors in the paper state that MPI is a threshold in the system.

The switches are connected among them and to the interface. This interface provides the binding logic between the bridge and the nodes. The RNI communicates with Nios II processors through the Avalon BUS from Altera. This is connected in the system as an Avalon slave which allows for the RNI to assign a memory for the Nios II processor. Using memory offsets, data packages and control information can be read or written in the RNI.

To verify the functionality of the system, the authors developed an application based on lamps, which proved the communication between the system nodules. The application allows defining the destination node of the instruction which turns on or off the lamp in the destination.

In another paper, the authors Kavadias *et al.*, (2010) [23] proposed a system composed of four Xilinx Microblaze processors. The use of local memory per core allows the direct communication between cores, with lower delay and power consumption comparing to the use of communication based on cache coherence, especially with CMP architectures, make them more diffused. In this paper the authors designed an interface for caches integrated in a network, which combined the use of cache flexibility and the efficiency of local memory. This paper presents and architecture which provides local and remote access to the memories,

for individual data words as well as multiple word blocks. The system was implemented in a Xilinx Virtex5 FPGA and four Xilinx Microblaze were integrated in the developed network. Figure 3 shows the system architecture developed in [23].

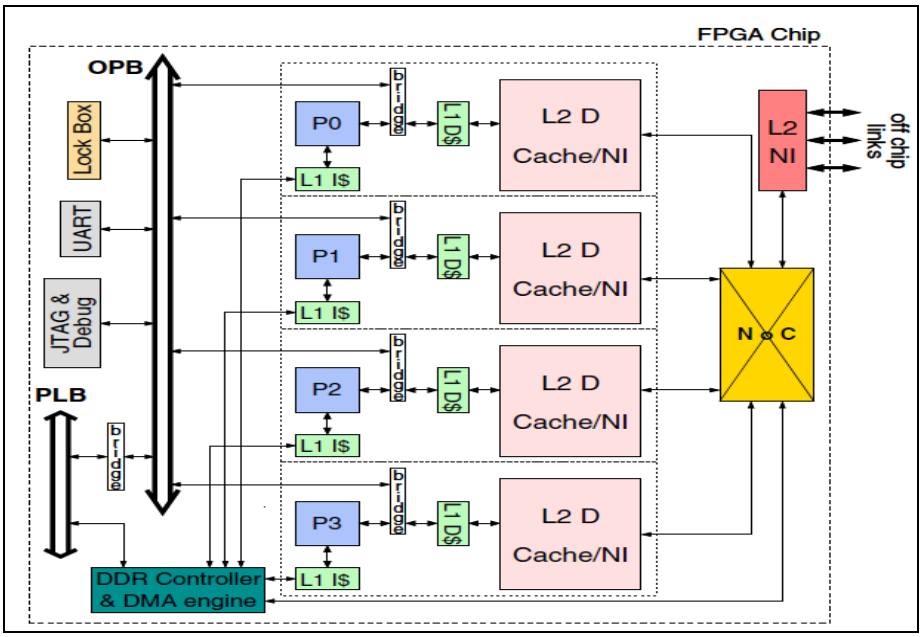


Figure 3. System Prototype with 4 Xilinx Microblaze in Network [23]

McCalpin (2011) [27] used a benchmark STREAM adaptation to evaluate the bandwidth* of the system, which is destined to verify the bandwidth in different levels of memory hierarchy. The STREAM benchmark copies three data arrays from a remote memory to a local memory, runs simple calculations over elements from the array and sends back the results from these computations. Other patterned benchmarks were also applied to find out the scalability and performance of the system.

Another paper on the subject was written by Bobda *et. al.*, (2007) [6]. The authors presented a design technique for an adaptive multiprocessor on-chip, specifically for FPGA devices. The technique consists in two steps in which the hardware infrastructure is generated, and then, the processors, the peripherals and customizable hardware components are configured. To automate the design development, obtain as much freedom from the design tools as possible and hide its complexity to the designer, an independent platform was designed. As a study case the authors implemented a problem called singular value decomposition (SVD) in a Xilinx ML310 board with up to eight processors.

The system consists of a PowerPc and two MicroBlaze processors in a Virtex II Pro 30 FPGA. The bandwidth measured for the system was of 36 Mbytes/s including the complete transaction necessary to code, send and decode a message.

Ihrig *et al.*, (2010) [19] present a tool for design flux automation called ACME which makes it easy to emulate hardware in new interconnection networks for wide multicore designs. The tools are intended to computer and network designers, which possess the digital design knowledge but do not feel comfortable with hardware description languages and synthesis fluxes. The ACME tool uses a graphical interface which allows a mixture of hardware components and software algorithms written in C, and each one of them contains user definitions in terms of delays e throughput in terms of system cycles. The ACME tool

automatically generates a hardware emulator with precise cycles, which integrates synthetized hardware with soft-cores processors which runs C code.

The paper written by Chun-Ming, *et. al.*, (2009) [7] presented a silicon prototipation methodology for a Multi-Project System-on-a-Chip (MP-SoC). A platform for multiple designs was created to integrate designs of heterogeneous SoC in a single chip, and, to show the efficiency, a MP-SoC chip was implemented using eleven designs of heterogeneous sharing a common platform. This platform is called CONCORD, which is created as a tool for verification to emulate the MP-SoC before the chip is manufactured.

To reduce risks during the SoC design flux, FPGAs are usually used to emulate the hardware before the chip is concluded. For the MP-SoC project, the authors used an ARM platform [1] in its Versatile version [2], and the CONCORD platform for hardware emulation.

The Aeroflex Gaisler company developed Leon3 [12]. Leon3 is a 32 bit processor based on the SPARC V8 architecture with multiprocessing configuration support.

The processor is totally synthesizable and p to 16 processors can be implemented in asymmetric multiprocessing (AMP) or synchronous multiprocessing. One typical configuration composed of four processors is capable of achieving 1400 DMIPS of performance. The source code for the Leon3 multiprocessor is available under the GNU GPL license for evaluation, research and educational purposes. For commercial use, a low cost license is required. Figure 4 shows Leon3 internal architecture, which follows the pattern suggested by Aeroflex Gaisler.

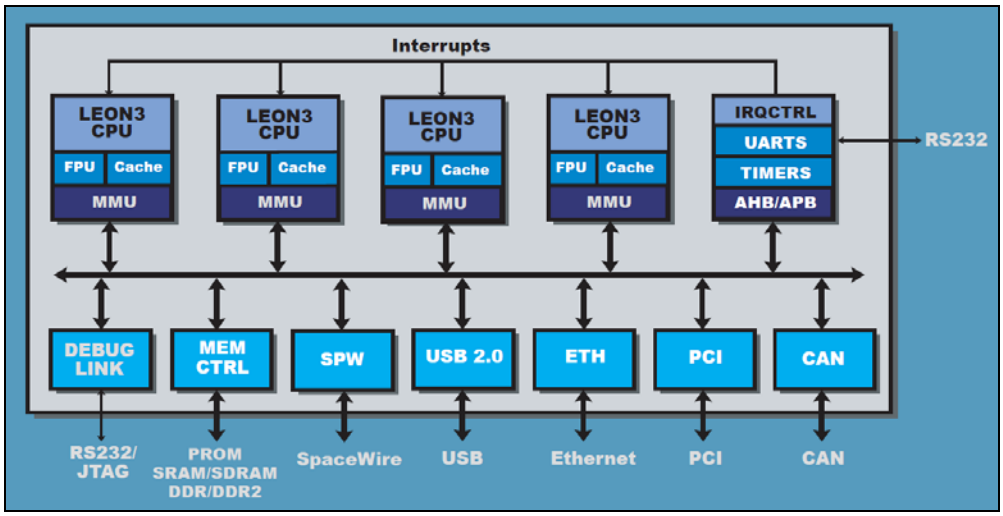


Figure 4. Leon3 Internal Architecture [12]

According to the Aeroflex Gaisler company, the Leon3 multiprocessor provides high performance with low frequency as in solutions with unique processors. This results is a significant economy in costs and power, while it maintains total compatibility with existing EDA and development fluxes.

The Leon3 can be used in both SMP and AMP configurations. The processor provides support for coherence cache, processor enumeration and interruption set. A single depuration interface enables hardware depuration in a not invasive way for both systems with one or multiple processors and provides for all internal registers and memory. A round-robin scheme for the AMBA BUS provides an honest use for all processors.

Leon3 is high configurable. The configuration of each processor in terms of cache size, FPU and MMU use can be defined individually. Asymmetric configuration like two main processors with Floating Point Unit, MMUe with two I/O processors is supported.

For SMP configurations, Leon3 supports the VxWorks, Linux 2.6 SMP and eCos operational systems. For AMP configurations, Leon3 supports the RTEMS and μCLinux operational systems.

Leon3 main characteristics are: (i) high configurability; (ii) flexible implementation using one to 16 processors; (iii) data size and cache instruction in the range of 0k to 2 Mbytes in each CPU; (iv) SPARC V8 architecture with a set instruction compatible with multiprocessor; (v) 400 MHz in an ASIC process of 0.13 μm, achieving performance of 1400 Dhrystone MIPS with four processors; (vi) cache snooping embedded for data coherence; (vii) low consumption mode where processors can be individually turned off, and; (viii) source code in VHDL or netlist.

3.2. Opened Questions in the Design of Specific Processors

Before the offering of multicore subjects, it is interesting to have dedicated solutions for specific applications, as the following list states. Later, it can be added to the same solution based on a single processor or a single processed platform, but with options of having multiple processing elements (multicore).

- Design new and modern systems in hardware security and in embedded systems, for instance, solutions for a network of sensors, solutions for automated applications, solutions for medical applications, *etc*;
- Propose new efficient architectures and specific for security systems and image processing in hardware and embedded platforms;
- Design specific hardware for modern cryptographic algorithms, as AES, SHA-3, ECC, HECC, MQQ, *etc*.
- Design specific hardware for the processing of biometric signals, such as voice, or iris recognition, fingerprint recognition, *etc*.
- Design hardware accelerators for applications/algorithms for image processing, biomedical applications, multimedia, *etc*.
- Design specific processors for solutions and integration of services and for treating information on images;
- Generate IP core for some security and image processing solutions that require hardware optimization;
- Design security systems aiming the integration of services;

3.3. Opened Questions about Multicore and GPUs

- Design efficient multicore systems in low power consumption;
- Parallelize application which require high performance and adapt them to work seamlessly in an efficient and scalable way;
- Create tools which parallelize automatically applications and execute them in an efficient way in multicore platforms;
- Design and evaluate Multicore security and image processing solutions that require high performance;
- Considering the new generation of unified hardware architecture of graphical hardware (GPUs), released by nVidia, which made possible to run generic programs,

and that applications with high level of parallelism exist, which makes clusters out of GPUs; there is still the need to study and apply GPU in several applications which require high processing power for a high data volume;
- Build tools that helps n the automated process of efficiently using he GPU in applications with high volume of information;
- Verify the use of GPU in modern security solutions: IDS and IPS of high performance, in cryptographic algorithms of critical performance, biometric algorithms, *etc*.

4. Optimization of Power Consumption

In this section, the most common methods employed for the estimative and measure of power consumption in different platforms is presented, for example: notebook, PC desktop and architectural simulation of a specific microprocessor.

The techniques used in the literature are: (i) measure of the discharge battery level for a notebook using commands inside Linux; (ii) theory estimative based on processor characteristics in a Desktop PC; (iii) architecture simulation using a simulation tool and (iv) measure of real consumption using an oscilloscope.

There are several simulators, which analyze the behavior of an algorithm following the characteristics and models of power consumption of some specific computational platform. As an example, we can cite the o Sim-Panalyzer [4], which is a simulator for power consumption based on SimpleScalar processors [3], and simulates a complete computational architecture (with CPU, cache and memory hierarchy).Based on this model it is possible to simulate real programs running on those platforms.

As a result, based on a detailed architecture simulation, the Sim-Panalyzer is able to model in detailed form dynamic power and total consumption of the target processor, and also to provide with detail the consumption of each simulated component (example: instruction cache, data cache, floating point unit, bus, I/O units, *etc*).

The parameters for profile generation in each computer component are given as input for the Sim-Panalyzer that along with SimpleScalar generates the power consumption patterns.

According Segundo Austin *et al.*, (2010) [3], the Sim-Panalyzer was originally written based on the set architecture (ISA) of the Family of processors ARM [1], obtaining outstanding results in simulations of this sort. Thus, during the execution of tests, configurations and parameters of an ARM architecture were used, to be configured and used in the Sim-Panalyer.

The Sim-Panalyzer application is made in a simple way, needing only to adjust the simulator passing, as a parameter the architecture to be simulated, along with the system characteristics, and the compiled program, along with the necessary parameters for its execution. To obtain the total consumption it is only needed to sum the energy consumption of each component simulated by the Sim-Panalyzer, obtaining the total consumption of the desired device.

There are other tools to make this measure, showed in [32]: Netsim, PowerScope, Jouletrack, SNU Energy Explorer WEB; but most of them are not available or have a restrict license use.

Real Measurements using an Oscilloscope: For the real energy consumption the current consumed by the processor executing an algorithm can be measured. To measure the current, a resistor in series with the computational platform power cable must be inserted [25, 26] then, the voltage difference from input to output is measured

from the shunt resistor is measured. With this data it is possible to calculate the current and power consumed by the processor.

The consumption calculus for the device during the algorithm execution can be extracted from the graphic* generated by the oscilloscope and multiplied by the time spent for each execution times the average power used by the algorithm, calculated using the variables: voltage and real current consumed by the platform.

For example, the graphics in Figure 5 shows more details of power consumption for a specific situation of the hashing SHA algorithm, and the different measures which can be obtained using an oscilloscope, which allows of quantifying the real power consumption of the application.

The results from the real measurements of power consumption using an oscilloscope are obtained from the graphics* generated by them,, as shown in Figure 6, where several executions can be observed, and each one of them presents the same behavior in consumption executing the same algorithm with the same parameters.

SHA-1

Parameter	Values
Time between executions	0,16 s
Time per execution	0,04 s
Voltage Variation	0,96 V
Average power during execution*	30,90 W
Consumption	1,24 J

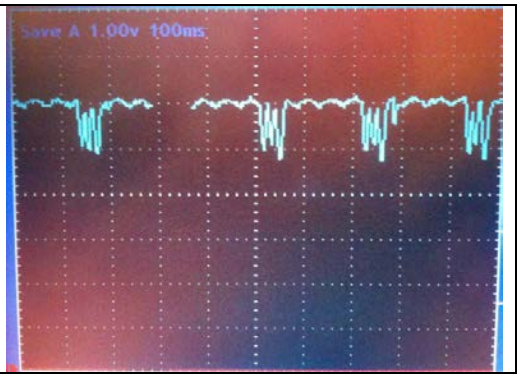


Figure 5. Voltage and Current Drained for the Execution of SHA Algorithm

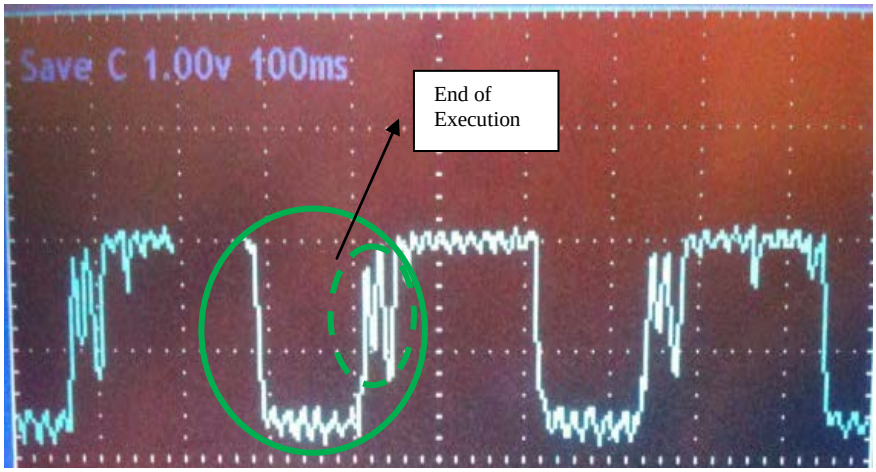


Figure 6. Details of Power Consumption in the Execution of SHA Algorithm

Figure 6 shows the specific case where the SHA program demands more power in the beginning of its execution, and it maintains constant with small variations, which corresponds to the computation of change in the constants A, B, C, D and in the algorithm, each one of them with a size of 32 bits, to compose the special word HASH looked in the input message, made of 160 bits. In the end, only an update of the last computation s updated, inner circle

inside total consumption circle) which demands less processing, thus, lower power consumption (inner circle).

Opened Questions Regarding Power Consumption

Considering that the first two methods are not efficient enough nor precise, then the community has been using many simulation techniques and experimental measures with equipments such as oscilloscopes, measuring instruments and others. Despite of that, there is still an open field to research in this area, for example:

- Create simulators for different computational platforms;
- Create tools that allows real energy consumption measure of executing algorithms and applications from MiBench's embedded systems benchmark [15], and programs from other scientific benchmark applications (Ex. o SPEC), Java (Java benchmarks), games, multimedia, among others.
- Create tools that detect thresholds in power consumption, identifying functions and/or parts of code who demand more current;
- Optimize code in order to diminish power consumption;
- Create new design and communication strategies, in hardware and software, which allows lower power consumption.

4. Conclusions

In this paper we have presented a brief discussion about modern platforms (multicore, GPUs, embedded systems, FPGA) which could be used in hardware security for new applications. We discuss the benefits about each one of them, and the open questions when they are used in security applications. We offer some ideas for doing future researches in hardware security, including new algorithms, applications, and aspects about power consumption.

Acknowledgements

The authors would like to thank FAPITEC (Fundação de Amparo à Pesquisa e à Inovação Tecnológica do Estado de Sergipe), FAPEAM, CNPq and CAPES, for financial support.

References

[1] ARM Versatile Platform, plataforma de desenvolvimento da empresa ARM, **(2010)**. Link www.arm.com/products/tools/versatile.php.
[2] ARM, An Introduction to Thumb. Advanced RISC Machines Ltd., March **(1995)**.
[3] T. Austin, "SimpleScalar: An Infrastructure for Computer System Modeling", IEEE Computer, vol. 35, **(2002)** February, pp. 59-67
[4] T. Austin, T. Mudge and D. Sim-Panalyzer Grunwald, Link: www.eecs.umich.edu/˜panalyzer/, **(2010)**.
[5] S. Bajikar, Trusted Platform Module (TPM) based Security on Notebook PCs - White Paper Mobile Platforms Group - Intel Corporation, **(2002)**.
[6] C. Bobda, "Design of adaptive multiprocessor on *chip* systems", Proceedings of the 20th Annual Conference on Integrated Circuits and Systems Design, **(2007)**, pp. 177-183.
[7] H. Chun-Ming, "Implementation and prototyping of a complex multi-project system-on-a-*chip*", ISCAS - IEEE International Symposium on Circuits and Systems, **(2009)** May, pp. 2321-2324.
[8] R. COSTA, "A. G. Desempenho e Consumo de Energia de Algoritmos Criptográficos do MiBench em Sistemas Móveis", UEA – Amazonas, **(2007)** November.
[9] M. Domeika, "Software development for embedded multi-*core* systems: a practical guide using embedded Intel architecture", 420p. Ed. Elsevier, **(2008)**.

[10] ECC - Elliptic Curve Cryptography. 10th Workshop on ECC. Fields Institute, Toronto, Canada, Link: www.cacr.math.uwaterloo.ca/.../ecc2006/announcement.html, **(2006)** September.
[11] J. Fan, X. Guo, J. DeMulder, P. Schaumont, I. Verbauwhede, "State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures", IEEE Intl. Symposium on Hardware-Oriented Security and Trust (HOST2010), **(2010)** June.
[12] J. Gaiesler, "The LEON processor user's manual", Link: www.gaisler.com/cms/index.php?option =com_content&task=section&id=5&Itemid=51, **(2013)**.
[13] M. G. O. Gericota, "Metodologias de teste para FPGAs (Field Programmable Gate Arrays) integradas em sistemas reconfiguráveis", Tese de Doutorado apresentada a Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, **(2003)**.
[14] J. Gonzalez, "Uma metodologia de projetos para circuitos com reconfiguração dinâmica de *hardware* aplicada a support vector machines", Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, **(2006)**.
[15] M. Guthaus, "MiBench: A free, commercially representative embedded suite", Workload Characterization, WWC-4. IEEE Intl. Workshop, **(2001)**.
[16] F. Hao, R. Anderson and J. Daugman, "Combining Crypto with Biometrics Effectively", IEEE Trans. on Computers, vol. 55, no. 9, **(2006)** September.
[17] T. Huffmire, B. BROTHERTON, N. Callegari, J. Valamehr, J. White, R. Kastner and T. Sherwood, "Designing Secure Systems on Reconfigurable Hardware", ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 13, no. 3, **(2008)** July.
[18] T. Huffmire, B. Brotherton, G. Wang, T. Sherwodd, R. Kastner, T. Levin, T. D. Nguyen and C. Irvine, "Security Primitives for Reconfigurable Hardware Based Systems", ACM Trans. on Reconfigurable Technology and Systems, vol. 3, no. 2, **(2010)** May.
[19] C. J. Ihrig, "Automated modeling and emulation of interconnect designs for many-*core chip* multiprocessors", Proceedings of the 47th Design Automation Conference, **(2010)**, pp. 431-436.
[20] H. Inoue, "A multi-*core* processor platform for open embedded systems. MSc Dissertation", Graduate School of Science and Technology of Keio University, Set, **(2009)**.
[21] R. Kastner, F. Fallah and A. Hosangadi, "Arithmetic Optimization Techniques for Hardware and Software Design", Cambridge University Press, ISBN-13: 9780521880992, **(2010)** May.
[22] R. Kastner, A. Kaplan and M. Sarrafzadeh, "Synthesis Techniques and Optimizations for Reconfigurable Systems", Kluwer Academic Publishers, ISBN 1402075983, **(2003)** November.
[23] S. G. Kavadias, "On-*chip* communication and synchronization mechanisms with cache-integrated network interfaces", Proceedings of the 7th ACM Intl. conference on Computing frontiers. ACM, **(2010)**, pp. 217-226.
[24] C. Kaya Koc, "Hardware Security", HTTP://*islab.oregonstate.edu/**koc**/*. **(2011)**.
[25] I. Lee, "Web-Based Energy exploration tool for embedded systems", IEEE Design & Test of Computers, **(2004)**.
[26] C. Lin, "Energy Efficiency Measurement for Multimedia Audio Decoding on Embedded Systems", Tunghai University, **(2006)**.
[27] J. McCalpin, "STREAM benchmark", Link: www.cs.virginia.edu/stream/ref.html#what, **(1995)**.
[28] B. Meakin, G. Gopalakrishnan, "Hardware Design, Synthesis, and Verification of a *Multicore* Communication", API. Univ. of Utah. TECHCON, **(2009)**.
[29] A. Menezes, "Handbook of Applied Cryptography. **(2006).** www.cacr.math.uwaterloo.ca/hac/authors/ajm.html.
[30] J. Menezes, "Guide to Elliptic Curve Cryptography", www.cacr.math.uwaterloo.ca/hac, **(2010)**.
[31] W. H. MINHAS, "Design and implementation of a plesiochronous multi-*core* 4x4 network-on-*chip* FPGA platform with MPI HAL support", Proc. of the 6th FPGA world Conference. Stockholm, Sweden: ACM **(2009)**, pp. 52-57.
[32] L. S. NETO, M. T. Chella and E. D. Moreno, "Estudo e Medição do Consumo de Energia de Algoritmos Criptográficos do MiBench", WSCAD-WIC 2010, **(2010)**, pp. 29-32.
[33] C. Paar, "Reconfigurable Hardware in Modern Cryptography", Material em www.crypto.ruhr-uni-bochum.de/en_**paar**.html, **(2011)**.
[34] C. Paar, B. Chetwynd, T. Connor, S. Y. DENG and S. Marchant, "An Algorithm-Agile Cryptographic Coprocessor Based on FPGAs", ECE Department, Worcester Polytechnic Institute, Worcester, U.S.A, The SPIE´s Symposium on Voice, and Data Communications, **(1999)** September.
[35] H. Rasheed and Y. C. R. Chow, "An Information Model for Security Integration", 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'07), **(2007)**, pp. 41-47.
[36] J. E. Savage and M. Zubair, "A unified model for *multicore* architectures", Proc. of the 1st international forum on Next-generation multicore / manycore technologies. ACM, Cairo, Egypt, **(2008)**, pp. 1-12.
[37] R. Srivaths, "SoC: Security on Chip", Proceedings of the MPSoC, www.princeton.edu/~sravi, **(2005)** July.

[38] W. Stallings, "Cryptography and network security: principles and practice", **(2008)**.
[39] I. A. Sumra, B. Hasbullah and J. L. B. Manan, "Comparative study of security hardware modules (EDR, TPD and TPM) in VANET", NIST, no. 113, **(2011)**.
[40] A. A. Wagan, B. M. Mugfah and H. Hasbullah, "VANET Security Framework for Trusted Grouping using TPM Hardware: Group Formation and Message Dissemination", CCS, **(2011)**.
[41] L. Wu, C. Weaver and T. Austin, "CryptoManiac: A Fast Flexible Architecture for Secure Communication", Advanced Computer Architecture Laboratory, Univ. of Michigan, U.S.A, In Proc. of ISCA - Intl. Conf. On Computer Architecture, Goteborg, Sweden, **(2001)**.
[42] S. Yang, K. Sakiyam and I. Verbauwhede, "A compact and efficient fingerprint verification system for secure embedded devices", Proc. of the 37th Asilomar Conference on Signal Systems, and Computers, Pacific Grove, USA, **(2003)** November, pp. 2058-2062.
[43] A. Zemva, "A rapid prototyping environment for teaching digital logic design", Education, IEEE Transactions, vol. 41, no. 4, **(1998)**, pp. 8.

Author

Edward David Moreno holds a degree in EE (University of Valle, Cali, Colombia)(1991), and MSc and PHD degrees in Electrical Engineering from University of Sao Paulo, Brazil, 1994 and 1998. The doctorate was Sandwich with University of Toronto, Canada, 1996, and Chalmers University of Technology, Goteborg, Sweden, 1997. He is currently professor at DCOMP/UFS – Federal University of Sergipe, Aracaju, Brazil. He is editorial board of four international journals: IJCNS, TCS Springer, JUCS and JCP. He has experience in the area computer science and computer engineering, with emphasis in computer systems architecture, acting on the following subjects: computer architecture, embedded systems, hardware security, power-aware computing, high performnce computing and performance evaluation.