View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Università degli Studi di Napoli Federico II Open Archive

Research works on electronic system-level design, FPGA testing, and security building blocks

Alessandro Cilardo
Department of Electrical Engineering and Information Technologies, University of Naples Federico II
via Claudio 21, 80125 Napoli, Italy, Email: acilardo@unina.it

***Abstract*—This document presents an overview of the research activity carried out by the author until the date of writing. It is also meant to report on the main results generated by a few funded project involving the author as a team member. The activity covered a range of topics involving automated generation of on-chip multiprocessor systems from high-level code, with particular emphasis on the system interconnect and the memory subsystems, design automation and test techniques for hardware-reconfigurable technologies, the design of advanced hardware blocks for cryptographic and cryptanalytical applications, the implementation and evaluation of security services in distributed environments, with special focus on time-stamping and public-key certification services, as well as the interplay between security services and hardware reconfigurability. The document presents the main highlights from the published works spawned by each of the above research threads.**

I. Overview

This document summarizes the research works on electronic system-level design, FPGA testing, cryptographic hardware, and security applications developed by the author until year 2014. The paper also includes research results achieved in the context of a few funded Research Projects, that are reported on in the corresponding sections.

One of the main research activities involved methodologies for the automated generation of on-chip multiprocessor systems from high-level code, with particular emphasis on the system interconnect and the memory subsystem. The methodologies under study were mostly targeted at reconfigurable hardware technologies, namely Field Programmable Gate Arrays (FPGAs), which provide the opportunity of customizing the system architecture driven by the application requirements. Furthermore, FPGA technologies also enable scenarios involving the interplay with high-level modelling and design tools as well as new mechanisms inherently enabled by hardware reconfigurability, such as the extension to hardware-reconfigurable systems of code mobility approaches –a traditionally software paradigm. These scenarios were also addressed by a couple of works developed by the author.

A second major line of research involved the design of advanced hardware blocks for cryptographic applications, motivated by the fact that hardware devices provide both high performance and resistance to tampering attacks, and are thus ideally suited for implementing computationally intensive cryptographic routines handling sensitive data. The activity included the design of various architectures for popular cryptographic systems, including Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC). While focused on implementation aspects, the above investigation allowed a deeper understanding of the mathematical basic building blocks in integer modular arithmetic and $GF(2^m)$ arithmetic, particularly addressing novel unified approaches for $GF(N)$ and $GF(2^m)$ finite field operations, used in the above cryptosystems, and parallel $GF(2^m)$ multiplier architectures. The work also focused on low-level arithmetic operations, particularly integer addition on long operands, recurrent in cryptographic applications, as well as acceleration of cryptanalytic applications, aimed at breaking cryptographic algorithms to demonstrate possible weaknesses.

In addition to FPGA design automation, the activity also explored new implications of hardware reconfigurability for testing, particularly techniques for online testing as well as application-dependent testing (ADT).

Because of the central focus of security applications, the author's research activity did not only focus on low-level compute technologies. In fact, a further branch of the activity covered the implementation, deployment, and evaluation of security services in distributed environments, with special focus on time-stamping and public-key certification services, as well as the interplay between security services and hardware reconfigurability in contexts where hardware cores can be distributed –pretty much like software digital contents– to FPGA mobile or in-field devices.

This paper is organized as follows. Section II describes the activities addressing the automated design and optimization of on-chip multiprocessor systems. Section III presents the main results involving cryptographic hardware as well as acceleration of cryptanalytical algorithms. Section IV reports on the activity on FPGA testing methodologies and related software tools. Section V describes the contributions on security service provisioning in distributed environments and security applications of hardware reconfigurable in-field devices. Section VI wraps up the contents of this report.

II. Electronic System-Level Design Methodologies

The activity involved methodologies for the automated generation of on-chip multiprocessor system descriptions from high-level code. A special focus was put on design automation for the system interconnect and the memory subsystem, driven by specific application requirements.

Manuscript revised December 31, 2014.

A. Automated system generation

The work presented in [1] introduced an experimental environment for electronic system-level design based on the OpenMP programming model, one of the prominent approaches to parallel programming in high-performance and scientific computing. The essential idea was the generation of multiprocessor systems on chip (MPSoCs) starting from high-level code with explicit parallelism, based on a custom compiler and a *high-level synthesis* (HLS) toolflow, allowing high-level language (HLL) programs to be automatically translated to hardware description language (HDL) modules. Blending parallel software programming paradigms with high-level synthesis, however, introduced a range of challenges at both the architectural level and the programming paradigm level, particularly involving the mismatches between the semantics of general high-level parallel code and the coding style imposed by high-level synthesis for FPGAs. The proposed environment was fully compliant with the OpenMP standard and it exhibited good scalability with respect to the number of threads and limited performance overheads. The paper also presented some comparisons with high-performance software implementations, using well-established OpenMP benchmarks, as well as with previous proposals oriented to pure hardware translation. The results indicated improved results in terms of both efficiency and scalability. The design solution was extended and described in [2], where we elaborated on the code transformations that we adopted to adapt source code to the underlying hardware synthesis process, as well as a few innovative architectural solutions for supporting OpenMP functionalities. The experimental results collected from the generated systems exhibited limited performance overheads and high application scalability, further confirming the potential impact for the automated synthesis of MPSoCs from parallel software applications.

The above work was further extended in [3], aiming at supporting automatically synthesized, heterogeneous hardware/software systems to be implemented onto FPGA devices. The work focused on a number of challenges, including the definition of a novel system-oriented Model of Computation (MoC), capturing the essential aspects of the structure of a parallel software application related to the optimization and translation process. The paper also presented an analytical optimization model based on Integer Linear Programming (ILP), as well as innovative techniques for early hardware cost prediction used to speed up the design space exploration process. The methodology was supported by a set of ad-hoc tools, including a custom OpenMP compiler and a hardware cost estimator, interacting with an existing ILP solver and a hardware high-level synthesis engine. The work also presented a case-study to show the effectiveness of the proposed methodology with a real-world application.

In this context, the work in [4] was specifically focused on the problem of estimating as soon as possible the hardware cost resulting from the translation process provided by existing tools. Such *early prediction* of hardware complexity is essential in driving hardware/software partitioning choices, where only a subset of the high-level program is implemented as a dedicated circuit. In that respect, early prediction helps estimate the hardware cost of a given high-level code segment before the (expensive) low-level logic synthesis, dramatically reducing the time required for an exhaustive exploration of different design choices. Clearly, this early estimation is inherently influenced by the specific toolchain for HLL-to-HDL translation. In the above paper, the author proposed a general framework for the systematic derivation of custom, tool-dependent metrics for early estimation. The framework was developed on top of the LLVM compiler infrastructure along with the R statistical package used to perform regression analysis. For a specific HLL-to-HDL compiler chosen for tests, we collected extensive experimental results on a large base of benchmarks, which showed interesting accuracy improvements over some related work previously presented, confirming the effectiveness of the framework in deriving a characterization of the underlying hardware compiler.

B. Automated interconnect generation

The design of the on-chip communication infrastructure in MPSoCs is of vital importance because it affects the inter-component data traffic and impacts significantly the overall system performance and cost. On the other hand, there is a very large spectrum of on-chip interconnect topologies and components that potentially meet given communication requirements, determining various trade-offs between cost and performance.

The work in [5] proposed an automated methodology for on-chip interconnect design, avoiding a manual and time consuming design space search process. The methodology took as input the description of the application communication requirements, and output an on-chip synthesizable interconnection structure satisfying given area constraints. Targeted at FPGA technologies, the approach generated an interconnection structure combining crossbars and shared buses, connected through bridges, yielding a scalable, efficient structure. It provided the first method to automatically generate FPGA-based communication architectures where heterogeneous communication elements, such as shared buses and crossbar switches, coexist in a network inherently supporting multiple communication paths.

The approach was extended in [6], addressing *joint* communication scheduling and interconnect synthesis. In fact, unlike the previous work in [5], the methodology concurrently defined the structure of the interconnect and the communication task scheduling, based again on the application communication requirements under given area constraints, but taking into account possible dependencies between tasks which could profitably be taken into account to avoid overprovisioning. This preserved the communication parallelism that can be exploited while keeping area costs low.

The above results were generalized and presented in an systematic way in a journal paper [7], centered on an automated methodology and tools to generate interconnects made of crossbars and shared buses, connected through bridges, yielding a scalable, efficient structure. The paper thoroughly described the formalisms and the methodology used to derive

such optimized heterogeneous topologies. It also discussed some case-studies emphasizing the impact of the proposed approach and highlighting the essential differences with a few other solutions presented in the technical literature.

C. Automated memory subsystem optimization

Similar to the interconnect, reconfigurable hardware platforms provide the opportunity of customizing the memory infrastructure based on application access patterns. In fact, FPGAs normally have numerous independent memory banks that can be accessed simultaneously, potentially offering a very large memory bandwidth. Adopting a suitable application-based memory partitioning strategy is thus vital to take full advantage of the memory architecture. The work in [8] addressed the problem of automated memory partitioning for FPGA-based systems generated through high-level synthesis, taking into account potentially parallel data accesses to physically independent banks. Targeted at affine static control parts (SCoPs), the technique relied on the Z-polyhedral model for program analysis and adopted a partitioning scheme based on integer lattices. The approach enabled the definition of a solution space including previous works as particular cases. The problem of minimizing the total amount of memory required across the partitioned banks, referred to as storage minimization throughout the article, was tackled by an optimal approach yielding asymptotically zero memory waste or, as an alternative, an efficient approach ensuring arbitrarily small waste. The article also presented a prototype toolchain and a detailed step-by-step case study demonstrating the impact of the proposed technique along with extensive comparisons with alternative approaches in the literature.

Additional insights along this line were given in [9], addressing area implications of memory partitioning for high-level synthesis on FPGAs. In fact, in addition to improving the potential memory bandwidth, partitioning also affects the area complexity of the generated system because the required steering logic depends on the partitioning scheme. The paper particularly considered the area implications of the lattice-based memory partitioning technique developed in the previous work. The experimental results, based on high-level synthesis tools targeted at FPGAs, showed that the proposed techniques could improve area efficiency compared to alternative approaches.

D. Further developments in hardware-reconfigurable MPSoC design

FPGA technologies enable new scenarios, involving the interplay with high-level modelling and design tools as well as new mechanisms inherently enabled by hardware reconfigurability. In particular, the work in [10] addressed the use of a popular modelling and simulation toolkit, Simulink, to automate the design of FPGA-implemented MPSoCs. The paper presented an approach to the automated identification of optimal mapping choices in a Simulink-to-MPSoC design flow. The mapping process relied on an appropriately chosen model of computation, capturing the high-level structure of the Simulink application as well as enabling formal checking of several relevant properties, such as boundedness, liveness, as well as throughput and latency formulas. The optimization approach exploited an emerging logic programming language, Answer Set Programming (ASP), for design space exploration. The proposed ASP-based solution provided a good fit for Simulink-to-MPSoC translation as a technique to automate the optimization of design choices aimed at resource utilization and execution time. A case-study and the related experimental results demonstrated the effectiveness of the proposed approach.

As mentioned above, the activity also explored the use of hardware reprogrammability for on-field dynamic mechanisms. In fact, reconfigurability enables the extension of traditionally software design paradigms to hardware systems. The work in [11], in particular, presented an attempt to extend the notion of code mobility to hardware-reconfigurable systems. We introduced an extended notion of *code mobility* –called deep mobility– which took into consideration the possibility to migrate "logical" hardware components of reconfigurable systems across a distributed infrastructure. We presented a general architecture of a mobility-aware hardware-reconfigurable system as well as a prototypical implementation which provided full support to deep code mobility while hiding the complexity of the hardware reconfiguration process. An application scenario was also described showing how the different dimensions of code mobility introduced by the work could benefit real-world distributed applications.

III. Cryptographic hardware

A major line of research involved the design of advanced hardware blocks for cryptographic applications. In fact, hardware devices provide both high performance and resistance to tampering attacks, and are thus ideally suited for implementing computationally intensive cryptographic routines which operates on sensitive data.

In [12] the author presented a hardware implementation of the RSA algorithm for public-key cryptography. Basically, the RSA algorithm involves a modular exponentiation operation on large integers, which is considerably time-consuming to implement. To this end, we adopted a novel algorithm combining the Montgomery's technique and the carry-save representation of numbers. A highly modular, bit-slice based architecture was designed for executing the algorithm in hardware. We also proposed an FPGA-based implementation of the architecture. The characteristics of the algorithm, the regularity of the architecture, and the data-flow aware placement of the FPGA resources resulted in a considerable performance improvement, as compared to other implementations presented in the literature.

Differently from the previous work, in [13] the author presented a hardware architecture and an FPGA-based implementation of the Montgomery's algorithm relying on a *digit-serial* approach, which allows the basic arithmetic operations to be broken into words and processed in a serialized fashion. As a consequence, the architecture implementation takes advantage of a short critical path and low area requirements. In fact, as compared to other solutions in the literature, the proposed

implementation of the RSA processor achieved smaller area requirements and comparable performance. The serialization factor S of the serial architecture was taken as a parameter and the final performance level was given as a function of this factor. In the work, we thoroughly explored the design trade-offs, in terms of area requirements vs. time performance, for different values of the key length and the serialization factor.

Generalizing on the previous results, the activity was aimed at exploring the design space for FPGA-based implementation of RSA. The journal paper [14] presented two alternative architectures for implementing the RSA algorithm on reconfigurable hardware. The two solutions were at the extremes of the design space, since one adopted a word serial approach, while the other had a fully parallel organization. Based on the analysis of these architectures for different values of the serialization factor, we explored the RSA implementation design space. We systematically analyzed and compared the results of the two design processes with respect to two fundamental metrics, execution time and FPGA resource usage. We emphasized pros and cons and commented on the trade-offs of the two design alternatives.

After focusing on RSA, the activity moved to other cryptosystems, particularly Elliptic Curve Cryptography (ECC). In fact, ECC has gained widespread exposure and acceptance, and has been included in many security standards. In [15] the author reviewed the essential insights behind ECC implementation, as a prominent case study of *cryptographic engineering*, a complex, interdisciplinary research field encompassing such areas as mathematics, computer science, and electrical engineering. In particular, the work showed that the requirements of efficiency and security considered at the implementation stage affect not only mere low-level, technological aspects but also, significantly, higher level choices, ranging from finite field arithmetic up to curve mathematics and protocols.

Interestingly, ECC enlarges the spectrum of the underlying mathematical operations to be supported. For its performance, ECC is critically dependent on modular multiplication, performed in one of two different algebraic structures, $GF(N)$ and $GF(2^m)$, which normally require distinct hardware solutions for speeding up execution. For both fields, Montgomery multiplication is the most widely adopted solution, as it enables efficient hardware implementations. In [16] the author presented a novel unified architecture for public-key cryptography. Based on a fully-parallel, bit-sliced unified scheme, the architecture was designed to perform integer modular multiplication/exponentiation used in $GF(N)$ as well as $GF(2^m)$ multiplication, the core operations of RSA and EC cryptography. The architecture used a radix-2 Montgomery technique for modular arithmetic, and a radix-4 most significant digit (MSD)-first approach for $GF(2^m)$ multiplication. The bit-sliced scheme was highly regular, modular, and scalable, as virtually any datapath length could be obtained at a linear cost in terms of hardware resources and no additional costs in terms of critical path. The proposed solution outperformed all similar unified architectures found at the time in the technical literature in terms of clock count and critical path. The architecture was implemented on an FPGA device. A highly compact and efficient design was obtained taking advantage of the architectural characteristics.

A further development along this direction was achieved in [17], proposing a novel unified architecture for parallel Montgomery multiplication supporting both $GF(N)$ and $GF(2^m)$ finite field operations. The hardware scheme interleaved multiplication and modulo reduction. Furthermore, it relied on a modified Booth recoding scheme for the multiplicand and a radix-4 scheme for the modulus, enabling reduced time delays even for moderately large operand widths. In addition, the work presented a pipelined architecture based on the parallel blocks previously introduced, enabling very low clock counts and high throughput levels for long operands used in cryptographic applications. Experimental results, based on 0.18 μm CMOS technology, proved the effectiveness of the proposed techniques and outperformed the best results previously presented in the technical literature.

A. Parallel multipliers

While focused on implementation aspects, the above activities allowed a deeper understanding of the mathematical basic building blocks in integer modular arithmetic and $GF(2^m)$ arithmetic, particularly addressing parallel solutions for modular and $GF(2^m)$ multiplication. This enabled the development of robust results based on formal approaches, described in a number of journal papers.

In [18], we introduced a change of representation for elements in $GF(2^m)$. The proposed representation is useful for architectures that implement unified Montgomery multiplication in finite fields $GF(2^m)$ and $GF(N)$ used for elliptic curve cryptography since it transforms a standard $GF(2^m)$ multiplication into a Montgomery multiplication and comes at virtually no cost in terms of conversion operations.

The above work applies to one of the most popular representation used in $GF(2^m)$ finite fields, i.e. polynomial representation, whose implementation properties are largely affected by the choice of the irreducible polynomial used to generate the representation. For $GF(2^m)$ field degrees where neither irreducible trinomials nor special polynomials (like Equally Spaced Polynomials, ESPs) exist, the best area/time performance has been achieved for special-type irreducible *pentanomials*. In [19] the author presented an efficient bit-parallel $GF(2^m)$ multiplier for a large class of irreducible pentanomials, relying on the newly introduced Shifted Polynomial Bases (SPBs). The theoretical part of the work derived a closed expression of the reduced SPB product for a class of polynomials $x^m + x^{k_s} + x^{k_{s-1}} + \ldots + x^{k_1} + 1$, with $k_s - k_1 \leq (m+1)/2$, and then applied the formulation to the case of pentanomials. The resulting multiplier outperformed, or was as efficient as the best proposals in the technical literature, but it was suitable for a much larger class of pentanomials than those studied previously. This property enabled the choice of pentanomials optimizing different field operations (for example, inversion), yet preserving an optimal implementation of field multiplication, as discussed and quantitatively proved in the paper.

The above results were further extended in [20], where the author of this report proposed a new representation, based

on what we called *Generalized Polynomial Bases* (GPBs), covering polynomial bases and Shifted Polynomial Bases (SPBs) as special cases. We introduced a novel formulation for polynomial basis and its variants, which is able to express concisely all implementation aspects of interest, i.e., gate count, subexpression sharing, and time delay. The methodology enabled by the new formulation was completely general and repetitive in its application, allowing the development of an ad-hoc software tool to derive proofs for area complexity and time delays automatically. As the central contribution of the paper, we introduced some new types of irreducible pentanomials and an associated GPB. Based on the above formulation, we proved that carefully chosen GPBs yield multiplier architectures matching, or in several cases outperforming, the best special-type pentanomials from both the area and time point of view. Most importantly, the proposed GPB architectures require pentanomials existing for all degrees of practical interest. Indeed, a list of suitable irreducible pentanomials for all degrees less than $1,000$ was given in the paper appendix.

A further Transactions paper explored an unconventional computational model for the implementation of parallel modular multiplication used in cryptographic applications [21]. In fact, motivated by the emerging interest in new VLSI processes and technologies, such as Resonant Tunneling Diodes (RTDs), Single-Electron Tunneling (SET), Quantum Cellular Automata (QCA), and Tunneling Phase Logic (TPL), the author investigated the application of the non-Boolean computational paradigms enabled by such new technologies. In particular, we considered Threshold Logic functions, directly implementable as primitive gates in the above-mentioned technologies, and studied their application to the domain of cryptographic computing. From a theoretical perspective, the work presented a study of the computational power of linear threshold functions related to modular reduction and multiplication. We established an optimal bound to the delay of a threshold logic circuit implementing Montgomery modular reduction and multiplication. In particular, we showed that fixed-modulus Montgomery reduction can be implemented as a polynomial-size depth-2 threshold circuit, while Montgomery multiplication can be implemented as a depth-3 circuit. The work also proposed an architecture for Montgomery modular reduction and multiplication, which ensures feasible $O(n^2)$ area requirements, preserving the properties of constant latency and a low architectural critical path independent of the input size n. We compared this result with existing polynomial-size solutions based on the Boolean computational model, showing that the presented approach had intrinsically better architectural delay and latency, both $O(1)$.

B. Speculative circuits

The activity also focused on low-level arithmetic operations, particularly integer addition on long operands, which is recurrent in cryptographic applications. In [22] the author presented a new speculative addition architecture suitable for two's complement operations. The speculative approach allows shorter combinatorial propagation delays and hence faster circuits, although these circuits might occasionally generate wrong results that need to be corrected. Existing architectures for speculative addition were all based on the assumption that operands have uniformly distributed bits, which rarely occurs in real applications. As a consequence, they were disadvantageous for real-world workloads, although in principle faster than standard adders. To address this limitation, the work introduced a new architecture based on an innovative technique for speculative global carry evaluation. The proposed architecture solved the main drawback of previous schemes and, evaluated on real-world benchmarks, it exhibited interesting performance improvements compared to both standard adders and alternative architectures for speculative addition.

In [23] the approach was extended to integer multiplication. The proposed speculative multiplier used a novel carry-save reduction tree relying on speculative $(m : 2)$ counters. A technique to automatically choose the suitable speculative counters, taking into accounts both error probability and delay, was presented. The speculative tree was completed with a fast speculative carry-propagate adder and an error correction circuit. We synthesized speculative multipliers for several operand lengths using the UMC 65 nm library. Comparisons with conventional multipliers showed that speculation is effective when high speed is required, as speculative multipliers reached a higher speed compared to their conventional counterparts and also proved to be effective in terms of power dissipation.

The speculative approach was also demonstrated in [24], where digit-level speculative addition was used for implementing modular inversion. The proposed circuit effectively addressed the problem of long carry chains caused by signed operations and allowed fast low-overhead implementations of modular inversion. The solution was particularly suitable for devices providing optimized carry-propagation logic such as field programmable gate arrays.

C. Acceleration of Cryptanalysis algorithms

Modern acceleration and heterogeneous computing technologies can have a number of inherent advantages for *cryptanalytic* applications, aimed at breaking cryptographic algorithms in order to demonstrate possible weaknesses. The author's activity was thus also aimed at exploring the adoption of advanced compute platforms, based either on software-programmable or reconfigurable hardware solutions, for cryptanalytical purposes.

The work in [25] developed a cellBE-based HPC application aimed to gain a deeper understanding of the robustness and weaknesses of the SHA-1 cryptographic hash function. In fact, in the light of previous attacks to the MD5 hash function, SHA-1 remained at the time the only function that could be used in practice, since it was the only alternative to MD5 in many security standards. The work presented a study of the vulnerabilities in the SHA family, namely the SHA-0 and SHA-1 hash functions, based on a high-performance computing application run on the MariCel cluster available at the Barcelona Supercomputing Center. The effectiveness of the different optimizations and search strategies that were

used was validated by a comprehensive set of quantitative evaluations. Most importantly, at the conclusion of our study, we were able to identify an actual collision for a 71-round version of SHA-1, the first ever at the time of writing.

Subsequently, the work in [26] moved to the exploitation of reconfigurable hardware for high-performance cryptanalysis of SHA-1. The work explored this opportunity by developing new approaches inherently based on hardware reconfigurability, enabling algorithm and architecture exploration, input-dependent system specialization, and low-level optimizations based on static/dynamic reconfiguration. As a result, the author identified a number of new techniques, at both the algorithmic and architectural level, to effectively improve the attacks against SHA-1. The work also defined the architecture of a high-performance FPGA-based cluster, achieving the highest speed/cost ratio for SHA-1 collision search available at the time. A small-scale prototype of the cluster enabled us to reach a real collision for a 72-round version of the hash function.

The above conference paper was extended in [27]. Relying on the proposed approaches inherently based on hardware reconfigurability, we designed an FPGA-based platform targeting 71-and 75-round versions of SHA-1. Under the same cost budget, the estimated times for a collision achieved by the platform turned out to be at least one order of magnitude lower than other solutions based on high-end supercomputing facilities, reaching the highest performance/cost ratio for SHA-1 collision search and providing a striking confirmation of the impact of hardware reconfigurability.

A general review of the key opportunities for cryptanalysis processing introduced by *heterogeneous computing* (generally referring to non-conventional accelerator devices such as GPUs and FPGAs) was presented in [28]. Addressing the cryptanalysis of SHA-1 as a case-study, the paper analyzed and compared three different approaches based on heterogeneous computing, namely a hybrid multi-core platform, a computing facility based on a GPU architecture, and a custom hardware-accelerated platform based on reconfigurable devices. The case-study application provided useful insights into the potential of the emerging heterogeneous computing trends, enabling unprecedented levels of computing power per used resource.

IV. FPGA TESTING

In addition to techniques for FPGA design, the activity also explored new implications of hardware reconfigurability for testing. The work in [29], developed in the framework of the PRIN project *COMMUTA: Hardware/software mutant components for distributed, dynamically reconfigurable systems*, made an attempt to blend together the concept of mobile agents and hardware reconfigurable systems to achieve self-healing properties. The mobile agent paradigm can potentially handle the complexity and heterogeneity of networked infrastructures, while runtime reconfigurable systems can provide flexible, adaptable, and high performance features. The paper presented a broad analysis of how the innovative aspects of these technologies could be exploited to implement effective test and repair strategies.

Subsequent works addressed specific testing techniques enabled by hardware reconfigurability. In fact, FPGA testing poses a number of challenges related to both the complexity of the device under test and the opportunities introduced by the support to reconfiguration. While techniques for offline testing of FPGAs, either device-oriented or application-oriented, are relatively mature, in critical applications such as avionics, space, and even numerous commercial products it is often necessary to perform *online* testing. The work in [30] presented a technique for online testing of digital designs implemented on an FPGA. The approach supported application-oriented testing, in that it covered the subset of the FPGA which is actually used for the implemented design, and considered scenarios where the FPGA component is a part of a larger embedded system. The proposed approach was in fact based on a software framework, acting as an abstraction layer for reconfigurable hardware resources. Essentially, the framework exposed to software applications a Register-Transfer Level view of the underlying hardware, allowing test procedures to be implemented as software programs. The approach proved to be especially advantageous when memory is a constraint, the case of many embedded systems. As proved by the experimental results, in fact, test procedures turned out to be very compact and much more memory-efficient than conventional approaches relying on static sets of FPGA testing configurations to be stored in system memory.

Along the same line of research, the work in [31] revisited further concepts in the so-called application-dependent testing (ADT) for FPGA devices. ADT has emerged as an effective approach ensuring reduced testing efforts and improving the manufacturing yield since it can selectively exclude a subset of faults not affecting a given design. In addition to manufacturing, ADT can be used online, providing a solution for fast runtime fault detection and diagnostics. The work in [31] introduced a set of new techniques that enabled us to overcome a few limitations in ADT techniques and effectively extend previous methodologies for ADT.

The research on ADT was further consolidated in [32]. The paper identified a number of issues in ADT techniques limiting their applicability, and proposed new approaches improving the range of covered faults, with special emphasis on feedback bridging faults, as well as new algorithms for generating ADT test configurations. Furthermore, the work introduced a software environment addressing the lack of tools, either academic or commercial, supporting ADT techniques. The architecture of the environment was highly modular and extensively based on a plug-in approach. To demonstrate the potential of the toolset, we developed a complete suite of plug-ins, based on both state-of-the-art ADT techniques and the novel approaches introduced in the paper. The experimental results presented at the end of the paper confirmed the impact of the proposed solution.

V. SECURITY SERVICES AND INFRASTRUCTURES

Recent advances in wireless technologies have enabled pervasive connectivity to Internet scale systems, including resource-constrained devices, such as mobile phones and tablets, a trend which has been referred to as ubiquitous computing over the past years. In particular, more and more

security-critical applications are being deployed in such scenarios, making it crucial to provide security services to lightweight devices. Since security functions are typically based on computationally intensive cryptographic algorithms, deploying them is particularly challenging due to the limited computing power and other constraints typically affecting the above scenarios. Hence, in addition to hardware-related topics, the activity carried out by the author of this report also covered the implementation, deployment, and evaluation of security services in distributed environments. One of those services involved a particular security application, the so-called digital time stamping. The paper [33] describes the results of a research activity, conducted cooperatively with an industrial party, involving a practical solution for the implementation of time stamping services and their exposition to the Internet for inter-enterprise integration. State-of-the-art time stamping algorithms and crucial issues related to their practical implementation were discussed. The focus was on integration problems which arise when a potentially large community of enterprises –relying on a handful of heterogeneous technologies– is willing to access remote third-party time stamping services. We proposed a practical architecture providing time stamping services, both in terms of *relative* temporal authentication using a linear linking scheme and *absolute* temporal authentication, based on publishing mechanisms and a trusted time source. The architecture was implemented using Web Services technologies. An integration experiment was conducted to evaluate the effectiveness of the proposed solution.

A similar application was addressed by [34] involving the design and implementation of an architecture for the provisioning of digital time stamping to mobile devices with limited resources. The architecture was described with respect to a case-study system and experimental results were also discussed.

A group of works addressed the interplay between security services and hardware acceleration. In particular, [35] discussed such challenges with respect to two key security services, Public-Key certification and digital time stamping, delivered to mobile devices. The work presented a multi-tier architecture combining a hardware-accelerated back-end and a Web Services based web tier for achieving interoperability while boosting performance. Further extending this research line, the work in [36] presented another solution combining hardware acceleration with a Web Services tier. The paper described the organization of the architecture, provided a detailed description of individual components, and presented the results of a thorough experimental campaign.

On the other hand, the activity described in [37] focused on a specific component, i.e. an FPGA-based key-store for improving the dependability of security services. A key-store is a facility for storing sensitive information, most typically the keys of a cryptographic application which provides a security service. In the paper, we presented a hardware implemented key-store, allowing secure storage and high performance retrieval of RSA keys. Since RSA is the most widely adopted standard for cryptographic keys, the proposed key-store can be effectively used to improve the dependability of a wide class of security services. The device was implemented on top of a Commercial Off The Shelf (COTS) programmable hardware board, namely a Celoxica RC1000 product mounting a Xilinx Virtex-E 2000 FPGA part. We described the architecture of the hardware device, illustrated the organization of the associated device driver, and evaluated the security and performance gain achieved by integrating our device in real-world applications.

Since understanding the impact of the platform architecture is a key issue for deploying efficient security-enabled applications on mobile devices, the work in [38] provided an experimental study of the influence that specific characteristics of mobile device platforms have on the final performance of security applications. The focus was on performance and resource utilization, which are key aspects when one develops applications on mobile devices. The case study was again a Web Services based solution for delivering public-key infrastructure services to mobile devices. Experiments were conducted on three different mobile terminals, spanning a large range of characteristics representative of resource-constrained devices. The results showed that: i) performance figures are not uniform in spite of similar underlying hardware characteristics, and ii) security and performance are often conflicting requirements.

In [39] we discussed the essential technical challenges behind mobile security from two seemingly separate perspectives: malware protection -the security as seen by mobile device users- and digital right management -the security as seen by operators and content providers. We suggested a unified interpretation of the technical requirements posed by these two perspectives. We also presented *TrustedSIM*, a proof-of-concept implementation demonstrating the proposed unified approach.

The journal paper [40] merged a few concepts behind trusted environments for mobile security with the potential of hardware reconfigurability. In fact, FPGAs introduced the possibility of *distributing* hardware cores pretty much like software digital contents, possibly on payment or on a subscription basis. In the work, we proposed an infrastructure for the secure distribution of such hardware digital contents (HDCs). Aimed at the practical realization of the envisioned scenario, the study analyzed the security-related features of the current FPGA devices, e.g., (partial) bitstream encryption, and took them as the underlying constraints for the definition of the infrastructure. The work clearly identified the roles involved in the secure distribution process, including a trusted third-party entity, and introduced a cryptographic protocol ensuring the confidentiality and the trustworthiness of partial bitstreams dynamically downloaded to the user's device. The study also presented a detailed case-study application scenario, namely the secure distribution of image codec components, providing a few quantitative results and demonstrating the limited overhead incurred by the proposed solution in terms of time and area costs. The study drew a few interesting conclusions and proposals for the evolution of security-related FPGA features which may enable the full realization of the secure HDC distribution concept.

As a further development of the activity, fostered by a collaboration of the author with a company providing mon-

itoring and security services, the work in [41] presented an approach to the parsing of heterogeneous data streams, addressing scenarios were enterprise business processes are geographically distributed and involve entities in loosely coupled interactions. While cooperating, these entities generate transactional data streams, such as sequences of stock-market buy/sell orders, credit-card purchase records, Web server log entries, and electronic fund transfer orders. Such streams are often a collection of events stored and processed locally, and hence they typically have ad-hoc, heterogeneous formats. On the other hand, elements in such data streams usually share a common semantics and indeed they can be profitably mined in order to obtain combined global events. The above cited work, hence, introduced a solution relying on the definition of custom grammars and automatic generation of ad-hoc parsers. The stream-dependent parsers can be obtained dynamically in a totally automatic way, provided that the appropriate grammar, written in a common format, is fed into the system. The work also presented a fully working implementation, that was successfully integrated into a telecommunication environment for real-time processing of billing information flows.

VI. Conclusions

This document reported on the research activity carried out by the author until the date of writing. The text presented the main highlights from the author's published works involving automated generation of on-chip multiprocessor systems from high-level code, design automation and test techniques for hardware-reconfigurable technologies, advanced compute solutions for cryptographic and cryptanalytical applications, the deployment of time-stamping and public-key certification services in distributed environments, as well as the interplay between security services and hardware reconfigurability. The report gave a detailed description of all the scientific publications derived from the above activities, highlighting the essential insights and the main results collected from the experimental evaluation.

Acknowledgments

The activities described in this document have been partially supported by PRIN2005 project *COMMUTA: Hardware/software mutant components for distributed, dynamically reconfigurable systems*, and POR Campania FESR 2007-2013 project *Progetto Metadistretto del Settore ICT - Sistema di comunicazione per l'integrazione delle informazioni nella distribuzione commerciale e nei punti vendita*.

Following is a complete list of publications derived from the research activities described in this document.

References

[1] A. Cilardo, L. Gallo, A. Mazzeo, and N. Mazzocca, "Efficient and scalable OpenMP-based system-level design," in *Proceedings -Design, Automation and Test in Europe, DATE*, 2013, pp. 988–991.

[2] A. Cilardo and L. Gallo, "Generating on-chip heterogeneous systems from high-level parallel code," in *Proceedings - 2014 17th Euromicro Conference on Digital System Design, DSD 2014*, 2014, pp. 161–168.

[3] A. Cilardo, L. Gallo, and N. Mazzocca, "Design space exploration for high-level synthesis of multi-threaded applications," *Journal of Systems Architecture*, vol. 59, no. 10 PART D, pp. 1171–1183, 2013.

[4] A. Cilardo, P. Durante, C. Lofiego, and A. Mazzeo, "Early prediction of hardware complexity in HLL-to-HDL translation," in *Proceedings - 2010 International Conference on Field Programmable Logic and Applications, FPL 2010*, 2010, pp. 483–488.

[5] A. Cilardo, E. Fusella, L. Gallo, and A. Mazzeo, "Automated synthesis of FPGA-based heterogeneous interconnect topologies," in *2013 23rd International Conference on Field Programmable Logic and Applications, FPL 2013 - Proceedings*, 2013.

[6] A. Cilardo, E. Fusella, L. Gallo, and A. Mazzeo, "Joint communication scheduling and interconnect synthesis for FPGA-based many-core systems," in *Proceedings -Design, Automation and Test in Europe, DATE*, 2014.

[7] A. Cilardo, E. Fusella, L. Gallo, A. Mazzeo, and N. Mazzocca, "Automated design space exploration for FPGA-based heterogeneous interconnects," *Design Automation for Embedded Systems*, vol. 18, no. 3-4, pp. 157–170, 2014.

[8] A. Cilardo and L. Gallo, "Improving multibank memory access parallelism with lattice-based partitioning," *ACM Transactions on Architecture and Code Optimization*, vol. 11, no. 4, 2014.

[9] L. Gallo, A. Cilardo, D. Thomas, S. Bayliss, and G. Constantinides, "Area implications of memory partitioning for high-level synthesis on FPGAs," in *Conference Digest - 24th International Conference on Field Programmable Logic and Applications, FPL 2014*, 2014.

[10] A. Cilardo, D. Socci, and N. Mazzocca, "ASP-based optimized mapping in a simulink-to-MPSoC design flow," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 108–118, 2014.

[11] A. Cilardo, N. Mazzocca, and P. Prinetto, "Exploring a new dimension in code mobility for ubiquitous embedded systems," in *Proceedings - IEEE 10th International Conference on Ubiquitous Intelligence and Computing, UIC 2013 and IEEE 10th International Conference on Autonomic and Trusted Computing, ATC 2013*, 2013, pp. 56–63.

[12] A. Cilardo, A. Mazzeo, L. Romano, and G. Saggese, "Carry-save Montgomery modular exponentiation on reconfigurable hardware," in *Proceedings - Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 206–211.

[13] A. Cilardo, A. Mazzeo, L. Romano, and G. Saggese, "Architecture and FPGA implementation of a digit-serial RSA processor," in *New Algorithms, Architectures and Applications for Reconfigurable Computing*. Springer US, 2005, pp. 209–218.

[14] A. Cilardo, A. Mazzeo, L. Romano, and G. Saggese, "Exploring the design-space for FPGA-based implementation of RSA," *Microprocessors and Microsystems*, vol. 28, no. 4, pp. 183–191, 2004.

[15] A. Cilardo, L. Coppolino, N. Mazzocca, and L. Romano, "Elliptic curve cryptography engineering," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 395–405, 2006.

[16] A. Cilardo, A. Mazzeo, N. Mazzocca, and L. Romano, "A novel unified architecture for public-key cryptography," in *Proceedings -Design, Automation and Test in Europe, DATE '05*, vol. 2005, 2005, pp. 52–57.

[17] A. Cilardo and N. Mazzocca, "Time efficient dual-field unit for cryptography-related processing," *IFIP Advances in Information and Communication Technology*, vol. 313, pp. 191–210, 2010.

[18] A. Cilardo, A. Mazzeo, and N. Mazzocca, "Representation of elements in F_{2^m} enabling unified field arithmetic for elliptic curve cryptography," *Electronics Letters*, vol. 41, no. 14, pp. 798–800, 2005.

[19] A. Cilardo, "Efficient bit-parallel $GF(2^m)$ multiplier for a large class of irreducible pentanomials," *IEEE Transactions on Computers*, vol. 58, no. 7, pp. 1001–1008, 2009.

[20] A. Cilardo, "Fast parallel $gf(2^m)$ polynomial multiplication for all degrees," *IEEE Transactions on Computers*, vol. 62, no. 5, pp. 929–943, 2013.

[21] A. Cilardo, "Exploring the potential of threshold logic for cryptography-related operations," *IEEE Transactions on Computers*, vol. 60, no. 4, pp. 452–462, 2011.

[22] A. Cilardo, "A new speculative addition architecture suitable for two's complement operations," in *Proceedings -Design, Automation and Test in Europe, DATE*, 2009, pp. 664–669.

[23] A. Cilardo, D. De Caro, N. Petra, F. Caserta, N. Mazzocca, E. Napoli, and A. Strollo, "High speed speculative multipliers based on speculative carry-save tree," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 12, pp. 3426–3435, 2014.

[24] A. Cilardo, "Modular inversion based on digit-level speculative addition," *Electronics Letters*, vol. 49, no. 25, pp. 1609–1610, 2013.

[25] A. Cilardo, L. Esposito, A. Veniero, A. Mazzeo, V. Beltran, and E. Ayguad, "A cellBE-based HPC application for the analysis of

vulnerabilities in cryptographic hash functions," in *Proceedings - 2010 12th IEEE International Conference on High Performance Computing and Communications, HPCC 2010*, 2010, pp. 450–457.

[26] A. Cilardo, "The potential of reconfigurable hardware for HPC cryptanalysis of SHA-1," in *Proceedings -Design, Automation and Test in Europe, DATE*, 2011, pp. 998–1003.

[27] A. Cilardo and N. Mazzocca, "Exploiting vulnerabilities in cryptographic hash functions based on reconfigurable hardware," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 810–820, 2013.

[28] A. Cilardo, "Heterogeneous computing vs. big data: The case of cryptanalytical applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8286 LNCS, no. PART 2, pp. 177–184, 2013.

[29] A. Benso, A. Cilardo, N. Mazzocca, L. Miclea, P. Prinetto, and E. Szilrd, "Reconfigurable systems self-healing using mobile hardware agents," in *Proceedings - International Test Conference*, vol. 2005, 2005, pp. 468–476.

[30] A. Cilardo, N. Mazzocca, and L. Coppolino, "Virtual scan chains far online testing of FPGA-based embedded systems," in *Proceedings - 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, DSD 2008*, vol. 2008-January, 2008, pp. 360–366.

[31] A. Cilardo, C. Lofiego, A. Mazzeo, and N. Mazzocca, "Revisiting application-dependent test for FPGA devices," in *Proceedings - 16th IEEE European Test Symposium, ETS 2011*, 2011, p. 213.

[32] A. Cilardo, "New techniques and tools for application-dependent testing of fpga-based components," *IEEE Transactions on Industrial Informatics*, 2015.

[33] A. Cilardo, A. Mazzeo, L. Romano, G. Saggese, and G. Cattaneo, "Using Web Services technology for inter-enterprise integration of digital time stamping," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2889, pp. 960–974, 2003.

[34] A. Cilardo, D. Cotroneo, C. Di Flora, A. Mazzeo, L. Romano, and S. Russo, "Design and implementation of a high performance architecture for providing digital time stamping services to mobile devices," *Computer Systems Science and Engineering*, vol. 22, no. 3, pp. 103–112, 2007.

[35] A. Cilardo, L. Coppolino, A. Mazzeo, and L. Romano, "High-performance and interoperable security services for mobile environments," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3726 LNCS, pp. 1064–1069, 2005.

[36] A. Cilardo, L. Coppolino, A. Mazzeo, and L. Romano, "Combining programmable hardware and Web Services technologies for delivering high-performance and interoperable security," in *Proceedings - 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, PDP 2007*, 2007, pp. 381–386.

[37] A. Cilardo, A. Mazzeo, L. Romano, and G. Saggese, "An FPGA-based key-store for improving the dependability of security services," in *Proceedings - International Workshop on Object-Oriented Real-Time Dependable Systems, WORDS*, 2005, pp. 389–396.

[38] A. Cilardo, L. Coppolino, A. Mazzeo, and L. Romano, "Performance evaluation of security services: An experimental approach," in *Proceedings - 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, PDP 2007*, 2007, pp. 387–394.

[39] A. Cilardo, N. Mazzocca, and L. Coppolino, "TrustedSIM: Towards unified mobile security," in *Proceedings - IEEE 10th International Conference on Ubiquitous Intelligence and Computing, UIC 2013 and IEEE 10th International Conference on Autonomic and Trusted Computing, ATC 2013*, 2013, pp. 563–568.

[40] A. Cilardo, M. Barbareschi, and A. Mazzeo, "Secure distribution infrastructure for hardware digital contents," *IET Computers and Digital Techniques*, vol. 8, no. 6, pp. 300–310, 2014.

[41] F. Campanile, A. Cilardo, L. Coppolino, and L. Romano, "Adaptable parsing of real-time data streams," in *Proceedings - 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, PDP 2007*, 2007, pp. 412–418.