PhD in Information Technology and Electrical Engineering

Università degli Studi di Napoli Federico II

PhD Student: Raffaele Martino

XXXII Cycle

Training and Research Activities Report – Second Year

Tutor: prof. Alessandro Cilardo

Table of Contents

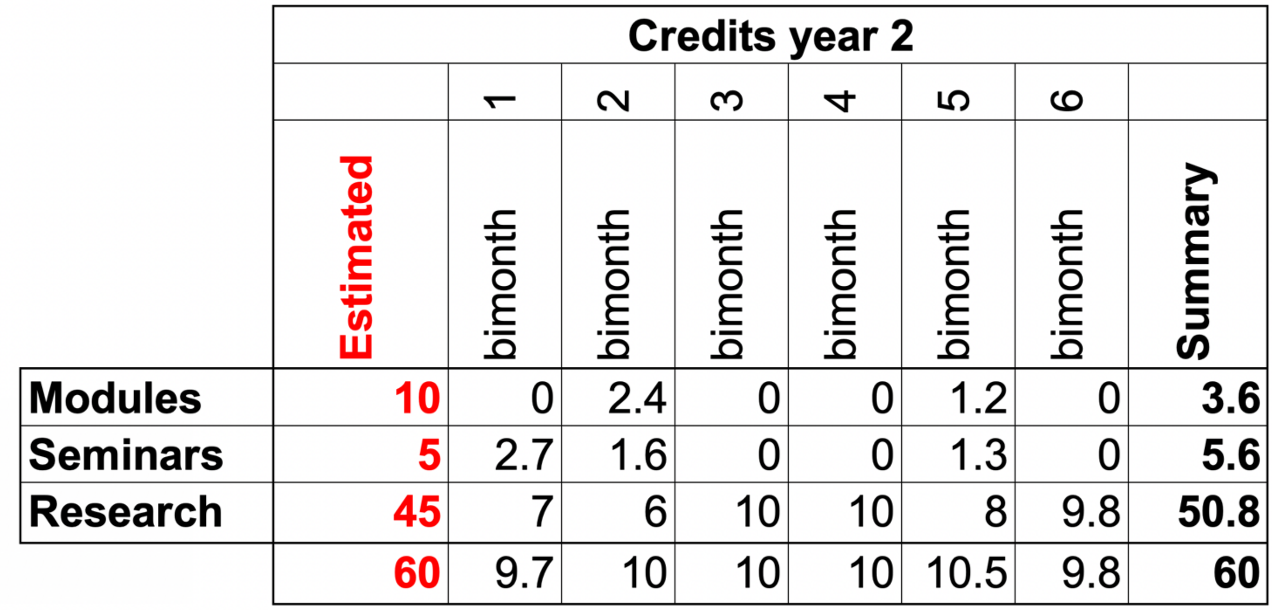
1. Information

Raffaele Martino

- Master’s Degree in Computer Engineering, magna cum laude, awarded by University of Naples Federico II in 2016
- Ph.D. student of the XXXII Cycle of the ITEE Course, at the University of Naples Federico II
- No Fellowship
- Tutor: prof. Alessandro Cilardo
- Collaboration from March 2018 with the Department of Agriculture of the University of Naples Federico II under a grant from the Regione Campania programme “URCoFi – Unità Regionale di Coordinamento Fitosanitario”

2. Study and Training Activities

Credits Summary



	Credits year 2							
		1	2	3	4	5	6	
	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary
Modules	10	0	2.4	0	0	1.2	0	**3.6**
Seminars	5	2.7	1.6	0	0	1.3	0	**5.6**
Research	45	7	6	10	10	8	9.8	**50.8**
	60	9.7	10	10	10	10.5	9.8	**60**

Courses

- “Compilers and Code Optimizations”, **ad hoc module** held by Prof. Eduardo Fusella, April – May 2018: 2.4 ECTS acquired on 24/05/2018
- “Author Seminar: How to Publish a Scientific Paper”, **improvement research skills** module held by Dr. Aliaksandr Birukou and Dr. Elisa Magistrelli, 26/11/2018: 0.4 ECTS acquired on 07/12/2018
- “Ciberconflitti: Sicurezza Informatica, Difesa, Stabilità Internazionale e Diritto Umanitario”, **improvement research skills module** held by Prof. Guglielmo Tamburrini, 28/11/2018: 0.8 ECTS acquired on 28/11/2018

Seminars

- “Etica e Intelligenza Artificiale”, **seminar** held within the seminar cycle “Tecnologie Digitali e Scienze Umane” by Prof. Remo Bodei and Prof. Guglielmo Tamburrini on 01/02/2018: 0.5 ECTS.
- “Le Nuove Frontiere della Robotica Cognitiva e l'Interazione Uomo - Robot”, **seminar** held within the seminar cycle “Tecnologie Digitali e Scienze Umane” by Prof. Alberto Finzi and Prof. Barbara Henry on 23/02/2018: 0.5 ECTS.
- “Logic – based Languages and Systems for Big Data Applications”, **seminar** held by Prof. Carlo Zaniolo on 13-15/03/2018: 0.8 ECTS.
- “Razionalità Limitata nell’Uomo e nella Macchina”, **seminar** held within the seminar cycle “Tecnologie Digitali e Scienze Umane” by Prof. Luigi Sauro and Prof. Maurizio Ferraris on 16/03/2018: 0.6 ECTS.
- “Model – based API Testing of Apache ZooKeeper”, **seminar** held by Prof. Cyrille Artho on 19/03/2018: 0.3 ECTS.
- “Lasciamo Parlare i Dati: Riflessioni sull’Apprendimento Automatico e i Big Data”, **seminar** held within the seminar cycle “Tecnologie Digitali e Scienze Umane” by Prof. Anna Corazza and Prof. Guido Roncaglia on 13/04/2018: 0.6 ECTS.
- “Internet, Intelligenza Artificiale e Tutela della Privacy”, **seminar** held within the seminar cycle “Tecnologie Digitali e Scienze Umane” by Prof. Piero Andrea Bonatti and Dr. Giovanni Buttarelli on 11/05/2018: 0.6 ECTS.
- “IBM Q: Building the First Universal Quantum Computers for Business and Science”, **seminar** held by Dr. Federico Mattei and Dr. Najla Said on 16/05/2018: 0.4 ECTS.
- “Types and Levels of Computational Explainations in AI: A Dual Process Proposal”, **seminar** held by Dr. Antonio Lieto on 14/11/2018: 0.4 ECTS.
- “Parallel and Distributed Computing with MATLAB”, **seminar** held by Eng. Stefano Marrone on 21/11/2018: 0.4 ECTS.
- “Il 5G e l’Evoluzione delle Reti Radiomobili”, **seminar** held by Eng. Francesco Mollica and Eng. Alessandro Vaccari on 26/11/2018: 0.5 ECTS.

3. Research Activity

A Flexible Evaluation Framework for Hash Designs to Meet the Needs of Innovative Applications

Hash algorithms are a fundamental building block of a number of secure applications. Traditionally, these algorithms are the basis for message integrity, which in turn constitutes the building block for Digital Signature Algorithms (DSAs) and Hash – based Message Authentication Codes (HMACs). But there are also more innovative applications which are gaining increasingly wide popularity. A paramount example is the blockchain technology, which at the very essence is an efficient way to maintain a distributed database growing with time within a peer – to – peer network. This technology was originally proposed as the enabling technology for cryptocurrencies, namely for the Bitcoin cryptocurrency, but it is clear that blockchains can be used for totally different applications; actually, to find applications of blockchain is currently a flourish research and development field.

The key idea of the blockchain is to split the database into subsequent blocks, then each block is hashed; the chain is built by including the hash value of the previous block in the hash of every block. The hash value is considered valid by the network only if it complies to a validity rule, which is characteristic of each blockchain technology. Once a new valid block is produced, it is announced to the network, so that every peer has a complete view of the whole chain. This makes impractical to alter the entries of the database without being detected, because such an operation would require to produce new valid hash values for all the subsequent blocks up to the last one, and announcing the forged chain to the network before the production of a new valid block on the original chain. Unless the majority of the nodes are compromised and agree to the forged chain, the network will agree with the original chain and the forged block will be rejected.

A fundamental part in ensuring the level of security of a blockchain is the validity rule of hash values. In the Bitcoin blockchain, the hash value of a block is valid if and only if it is lower than a specified value, called target and encoded in the block header. In order to obtain such a hash value, a nonce is included in the hash input, hence finding a valid hash value of a block implies finding a value of the nonce which makes the hash function to output a valid value. Finding a valid nonce requires the inversion of the underlying hash function, which is double SHA-256[1]; since hash functions are non – invertible by construction, the only way to find a valid nonce is brute – force. The first peer which find a valid nonce for the current block and announces it to the network is rewarded by the Bitcoin protocol with a predetermined amount of newly – minted coin: for this reason, the process of producing a new block in a blockchain is referred to as mining.

Applications put higher and higher demands on the performance of hashing. In network applications such IPSec and SSL/TLS, the performance of the hash algorithm can become the bottleneck for the whole service. Innovative applications like Bitcoin mining are even more demanding, since every little improvement in performance can translate into a significant increase of revenues, due to the competitive nature of the mining process. These requirements make hardware solutions attractive for the research activity, due to higher performance achievable and the possibility to design power – efficient implementations. These two objectives can be simultaneously optimized; alternatively, one of them can be stressed more, paying a penalty on the other metric: the hardware design provides a wider design space in which different trade – offs can be found. Apart from the optimization potential, hardware solutions provide also more security than software implementations, since they cannot be affected by cyberattacks; on the contrary they can provide a form of physical protection. For all these reasons, some security bodies like the National Security Agency (NSA) allow only hardware implementations.

The research activity focuses on both algorithm optimizations and application – specific improvements. The application currently under study is the Bitcoin mining, hence algorithm optimizations are particularly focused on the underlying SHA-256 algorithm. The first step is to study the most effective design alternatives for the circuit implementing the hash algorithm. There are several different proposals in the literature, hence a review of the different options has been performed. However, it has emerged that it is

[1] SHA-256 applied twice

not immediate to assess strengths and weaknesses of each proposal only by looking at the papers, for a number of reasons.

First, proposals are often optimised to best suit the needs of the specific application they are designed for, exploiting also hypotheses on the context of operation. For example, SHA-256 circuits designed for HMACs may assume that the message is not entirely available at the beginning of the computation and may exploit this condition to perform initialisations without paying any penalty. But this hypothesis may well not hold when using the proposed architecture for different applications, as it happens for the Bitcoin mining.

Moreover, results showed along with each proposal often are not comparable with each other due to the fact that they are not obtained with the same target technology. Put another way, it is not possible to assess the impact of different FPGA technologies on the reported result and hence it becomes impossible to evaluate the potential advantages of choosing a particular proposal on the FPGA technology chosen for the implementation, when this is different from the one used in the original paper.

In order to perform a fair and objective comparison of the different architectures proposed for SHA-256, it would be necessary to implement every SHA-256 circuit, synthesize it on the target FPGA and compare the resulting numbers. However, due to the structure of the SHA-256 algorithm, the critical path is located placed in the round core which performs the iteration function. Hence a framework has been developed to compare the different architectures of the round core without having to implement all the hashing circuit for every proposal. Instead, the framework is highly configurable in order to accommodate the needs of the different designs, and to compare various architectural solutions.

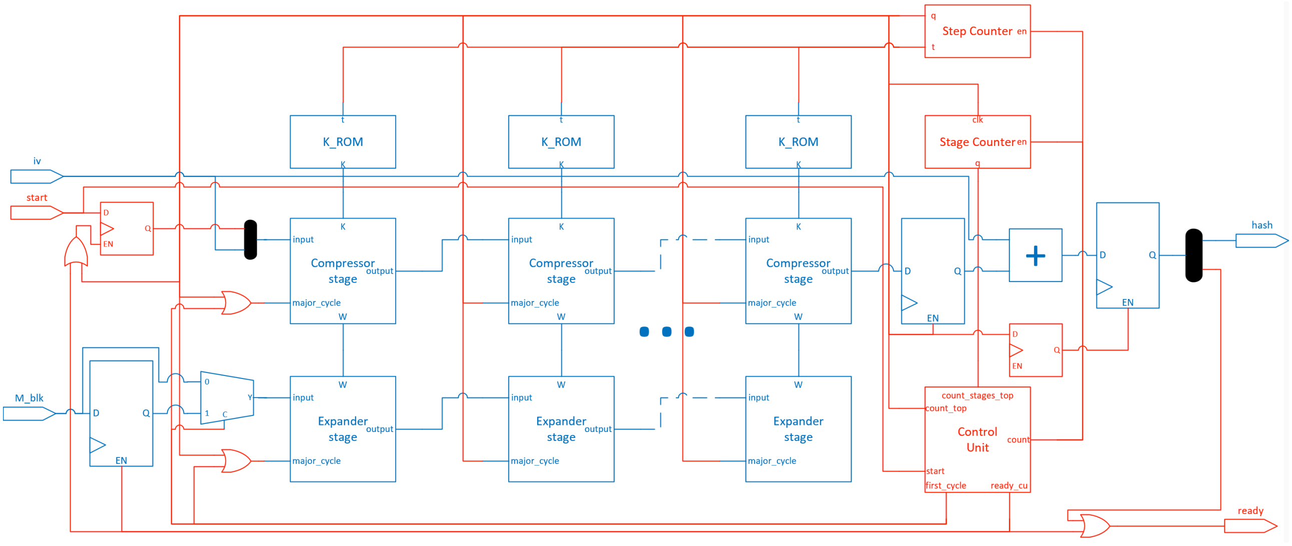


Figure 1: The developed evaluation platform. The operative part is shown in blue, the control part is displayed in red

The developed framework can also be employed as a means for quickly developing a new design for the SHA-256 transformation round core. Indeed, within the framework a designer can easily accommodate its own architecture, obtaining a fully working hash circuit without having to develop the whole architecture, especially to face all the timing issues which arise when developing the Control part of such a complex design.

The framework has been developed in VHDL for being implemented on an FPGA platform. Figure 1 shows the architecture of the developed evaluation platform. To enable the exploration of pipelined architectures, it consists of two parallel pipelines:

- One pipeline is for the Compressor, the part of the circuit which computes the main function of the SHA-256 algorithm. Each stage of this pipeline is flanked by a ROM containing the constant words required by the stage
- The other pipeline is for the Expander, the part of the circuit which derives the round keys from the input message to be hashed.

On the other hand, the Control Part is made up of an FSM and two counters:

- The main counter is the one employed to keep track of the iterations performed within each stage. The overflow of this counter hence sources the signal which drives the communication between stages. This is the most important control signal of the architecture, referred to as the *major cycle clock*.
- The other counter is ancillary to the FSM as it is used to keep count of the remaining working stages when the pipelines have to be flushed due to the exhaustion of messages which to work on.

The architecture can be configured simply by setting the following parameters

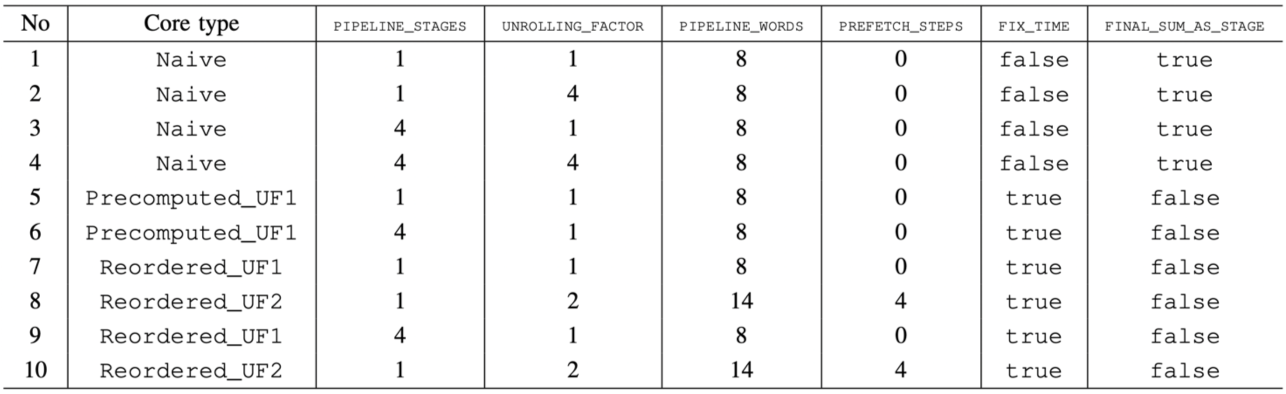
1. *Hash size*
   Due to the extreme similarity between the algorithms of the SHA-2 family, the framework has been extended to be capable of implement SHA-512 instead of SHA-256 according to the value of the hash size[2].
2. *Number of pipeline stages*
   The number of pipeline stages can be changed directly by setting this parameter. It is worth mentioning that, even if pipelining is disabled by setting this parameter to 1, the fact that the architecture of the framework is pipelined by nature implies that additional cycles set by other configurations affect only the latency of the circuit, and not its throughput.
3. *Unrolling factor*
   The unrolling factor must be set in this parameter, consistently with the unrolling factor of the transformation round core, unless the transformation round core itself is generic in the unrolling factor.
4. *Width of the Compressor pipeline stage*
   The Compressor pipeline registers, which at least contain the 8 working variables, can be enlarged to accommodate additional variables. When this happens, an initialisation unit is also added for providing the initialisation values for the additional variables. The clock cycle required by the initialisation unit adds only to the latency of the circuit, due to the underlying pipelined architecture.
5. *Prefetch shift*
   This parameter is used to anticipate the constants and the round keys when the transformation round core exploits the fact that these variables can be computed in advance of the round they are needed.

[2] Since currently only the major variants of the SHA-2 family are supported, specifying 256 univocally selects SHA-256.

6. *Tempification*
   This boolean parameter is used to modify the tempification of the circuit to accommodate the needs of the round design. Namely, it must set to true if the constants and the round keys are used before the pipeline register, otherwise it must be set to false.
7. *Final sum*
   This Boolean parameter determines whether or not the final chaining sum must be performed as a separate stage. As this stage is not counted into the number of pipeline stages, the additional clock cycle introduced affects only the latency even if the number of pipeline stages is set to 1.

The transformation round core is placed in the Compressor stage, hence different designs for the transformation round core can be used within the framework by specifying different architectures for the transformation round core. It is worth noting that a number of parameters, namely 4, 5 and 6, are fixed according to the specific transformation round design to be used in the Compressor pipeline, whereas 1, 2, 7 and up to some extent 3, are freely configurable, leading to a wide variety of different architectures. Table 1 lists the different options explored.

Table 1: Architectures explored



No	Core type	PIPELINE_STAGES	UNROLLING_FACTOR	PIPELINE_WORDS	PREFETCH_STEPS	FIX_TIME	FINAL_SUM_AS_STAGE
1	Naive	1	1	8	0	false	true
2	Naive	1	4	8	0	false	true
3	Naive	4	1	8	0	false	true
4	Naive	4	4	8	0	false	true
5	Precomputed_UF1	1	1	8	0	true	false
6	Precomputed_UF1	4	1	8	0	true	false
7	Reordered_UF1	1	1	8	0	true	false
8	Reordered_UF2	1	2	14	4	true	false
9	Reordered_UF1	4	1	8	0	true	false
10	Reordered_UF2	1	2	14	4	true	false

Preliminary results show that the most performing architecture is the one featuring a basic implementation of the SHA-256 transformation round, with both unrolling and pipelining. This alternatives outperforms the others in terms of both bare throughput and power efficiency.

Next steps

The goal of the comparison of design alternatives for SHA-256 implementation is to define an optimal design, by combining the best design techniques according to performance and/or power efficiency. It is worth noting that, due to the requirements of the applications, area occupation considerations are less relevant, hence a penalty in area occupation can be paid in order to achieve more performance (or, less likely, power efficiency), when compared with a generic hardware SHA-256 implementation.

The subsequent step will be the design of the complete Bitcoin miner. Currently available Bitcoin FPGA miners employ standard design techniques for the hashing core, and even pipelining is not always used; hence the exploitation of an optimized version of the SHA-256 circuit is expected to produce a significant improvement.

The topic of Bitcoin – related improvements does not appear frequently in the technical literature. This is because Bitcoin has been developed by a community of enthusiast developers without any involvement of academic or industrial research. Nevertheless, the fact that there is room for improvement in the Bitcoin mining algorithm has been pointed out. A large share of the input of the algorithm is fixed, hence known a priori, or "slowly" variable; here the word "slowly" is referred to the hash rate. This means that some optimizations of the hashing algorithm can be performed, not valid in general but valid only for the Bitcoin mining. Namely, some values can be hard – coded into the circuit, while other values can be computed and stored to be used multiple times, until the part of the input upon which they depend will change.

Moreover, since the Bitcoin mining is basically a challenge about the inversion of the hash algorithm, the possibility of putting in place some collision attack will be studied. The SHA-2 family of algorithm has not yet been broken by collision attacks when all the rounds are executed, but there are some successful attacks to reduced – rounds SHA-256; such an attack could be put in place were it to be possible to reduce the hash computations thanks to the fixed values.

The optimal design will also benefit from some study of the technological level. A low-level analysis of the power consumption of each element of the target platform will drive the further optimisation of the Bitcoin miner.

4. Products

In preparation

- **R.Martino**, M.Nicolazzo, G.Langella, "A full integrated system for agroclimatic and pest monitoring at farm and landscape scales in Campania Region", abstract presented at "1st Workshop on Metrology for Agriculture and Forestry (METEOAGRIFOR)", full paper for proceedings in preparation.
- Work in progress, provisional title: "A SHA-2 Flexible Framework to Explore, Evaluate and Compare Different Design"

5. Tutorship

Tutorship activities grant from the University

- Grant from the University of Naples "Federico II" to perform tutorship activities for students attending the 1st year of the B.Sc. for the academic year 2018/2019:
- Provided tutorship to students of the courses of "Fondamenti di Informatica" and "Laboratorio di Programmazione"
- 43 hours