Raffaele Martino

Tutor: prof. Alessandro Cilardo

XXXII Cycle - II year presentation

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

CONTEXT

Hash algorithms are a fundamental building block of a number of secure applications, including innovative applications like **blockchains**. Improving such applications involve both working at the components level, including improvements of the hash algorithm, and working and the system level. For the hash algorithm, a number of different optimisations have been proposed in the literature.

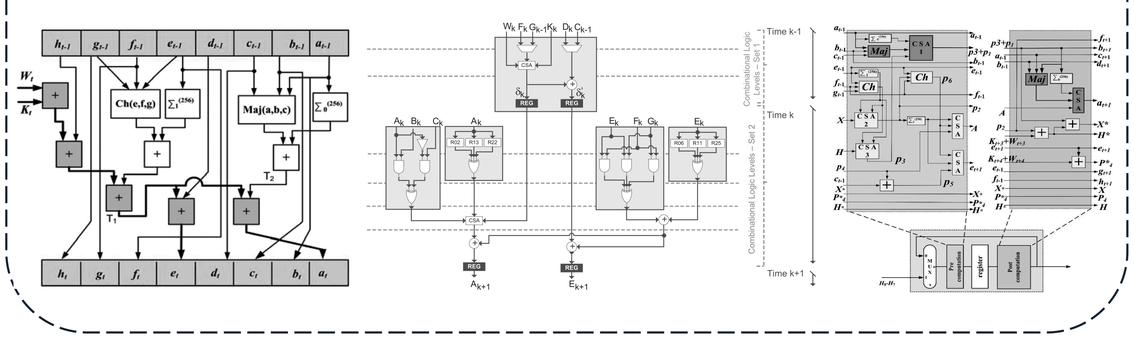
TRENDS

A blockchain is at the very essence an efficient way to maintain a distributed database growing through time within a peer – to – peer network. The database is split into subsequent blocks, then each block is hashed; the chain is built by including the hash of the previous block in the hash of the current block.
To replace a valid bloc, one should recompute the hash values for all the subsequent blocks, which requires to perform the same computational effort that was required to build the valid chain, due to the one-way property of the hash functions.

NEEDS

Determine the best hardware architecture of the SHA-2 hash algorithm according to specific design constraints.

- Abstract from the effects of different working conditions or different target architectures.



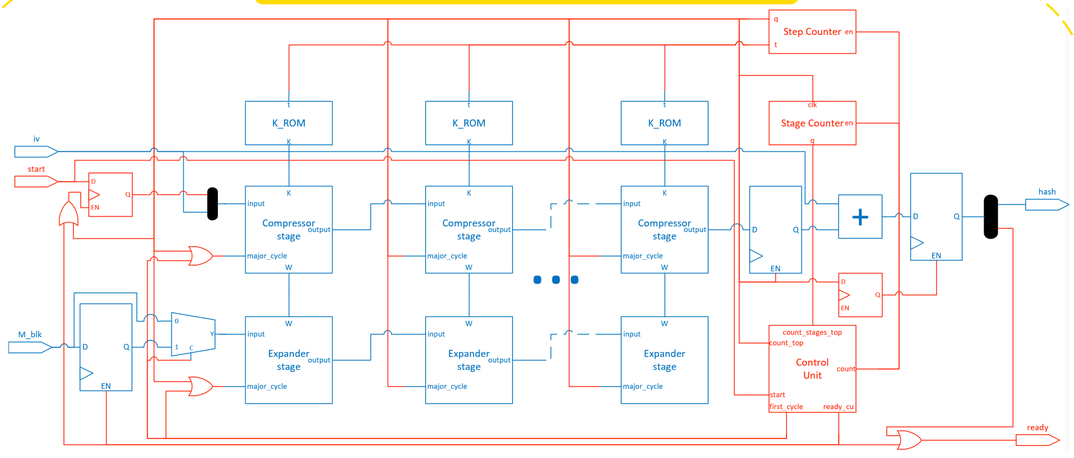
METHODOLOGY

We developed an evaluation platform to compare different architectures of the internal SHA-2 transformation round against the **same** target platform.
This allows to compare different architecture proposals fairly and under the same conditions, in order to assess their performance according to different evaluation metrics.

- This allows to make an informed decision about which architecture to pick in order to meet the specific requirements of a complex system which requires SHA-2 as a component.
- This allows also to quickly prototype a new architecture proposal for the SHA-2 transformation core, without having to develop all the circuit from scratch, instead exploiting an highly configurable architecture.

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

EVALUATION PLATFORM



The evaluation platform is constituted by:

- Input and output registers
- CP: one FSM and two counters
- OP: **two parallel pipelines**, one for the **Compressor** and one for the **Expander**

Configurable by means of VHDL `generic`s:

1) Number of pipeline stages
2) Unrolling factor
3) Width of the Compressor pipeline stages
4) Prefetch steps
5) Tempification of the parallel pipelines
6) Additional final stage

RESULTS

Evaluation has been performed considering several metrics, to take into account the different needs and constraints which can drive the selection of the most appropriate core.

- Optimization of one metric is usually paid by a loss in another metric (e.g. pay space for buying time).
- The implementation which makes use of the precomputation turned out to underperform the straightforward implementation.
- The implementation with spatial reordering outperformed the base implementation, but not in the unrolled variants. despite the deployment of even more optimisations.
- The basic implementation with pipelining and unrolling achieved the best result in terms of hash rate and power efficiency, but the most optimised spatial reordering implementation, in the pipelined variants, showed the best area efficiency.

N°	Hash rate	Area		Power	Efficiency	
	(Mhash/s)	LUT	FF	Consumption (W)	Area (Mbps/LUT)	Power (Mbps/mW)
1	6.516	1578	1875	0.573	1.057	2.911
5	5.189	1619	1608	0.559	0.821	2.376
7	6.802	1485	1640	0.582	1.173	2.992

(a) Base architectures

N°	Hash rate	Area		Power	Efficiency	
	(Mhash/s)	LUT	FF	Consumption (W)	Area (Mbps/LUT)	Power (Mbps/mW)
3	22.769	5314	4302	0.865	1.097	6.738
6	20.019	5381	4056	0.889	0.952	5.765
9	23.621	4986	4312	0.887	1.213	6.817

(b) Pipelined architectures

N°	Hash rate	Area		Power	Efficiency	
	(Mhash/s)	LUT	FF	Consumption (W)	Area (Mbps/LUT)	Power (Mbps/mW)
2	8.065	2793	1960	0.550	0.739	3.754
8	6.073	2907	2194	0.616	0.535	2.524

(c) Unrolled architectures

N°	Hash rate	Area		Power	Efficiency	
	(Mhash/s)	LUT	FF	Consumption (W)	Area (Mbps/LUT)	Power (Mbps/mW)
4	27.741	5314	4302	0.865	1.097	8.877
10	22.294	8670	5184	1.036	0.658	5.509

(d) Unrolled and pipelined architecures

CONTACTS & PROJECTS

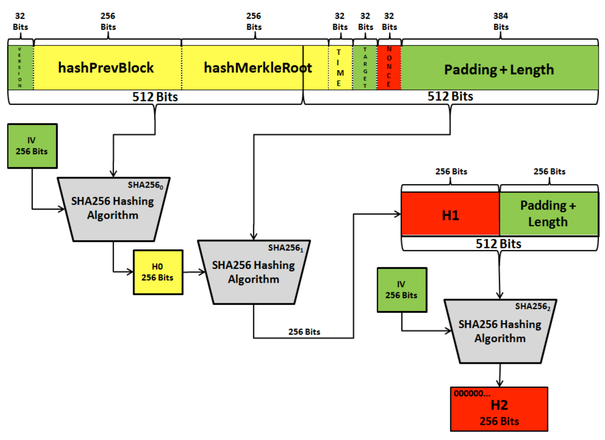
Contacts:

- raffaele.martino2@unina.it
- a.cilardo@unina.it

COLLABORATIONS

CRISP
Centro di Ricerca Interdipartimentale sulla "Earth Critical Zone" per il supporto alla Gestione del Paesaggio e dell'Agroambiente

FUTURE WORK



- The results shown call for an **analysis of the reasons** why some architectures, which were supposed to be more performant thanks to their deep optimisations, turned out to underperform the basic implementation of SHA-2.
- This activity is part of a research effort aimed to build an **optimised Bitcoin miner**, since the underlying hash function of the Bitcoin blockchain is SHA-256. The optimised Bitcoin miner that will be developed will benefit from the results of this activity, but will also feature **system-level optimisations**. The study will concentrate on:
  - Algorithm optimisations
  - Collision-based attack
  - Power-saving specific improvements
- Other applications of this work include:
  - Integration of the most appropriate SHA-2 architecture to the MANGO heterogeneous platform
  - Performing a similar analysis on the more recent SHA-3 hash algorithm