h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

RAW
Reconfigurable Architectures Workshop

27th Reconfigurable Architectures Workshop - May 2020

A FPGA-Based Post-Processing and Validation Platform for Random Number Generators

Laurent Gantel[1], Alexandre Duc[2], Lucie Steiner[2], Fabien Vannel[1], Andres Upegui[1], and Florent Gluck[1]

[1] *University of Applied Sciences and Arts Western Switzerland (HES-SO/HEPIA), Geneva, Switzerland*
Contact: firstname.name@hesge.ch

[2] *University of Applied Sciences and Arts Western Switzerland (HES-SO/HEIG-VD), Yverdon-les-Bains, Switzerland*
***Contact**:* firstname.name@heig-vd.ch

swissuniversities

Context

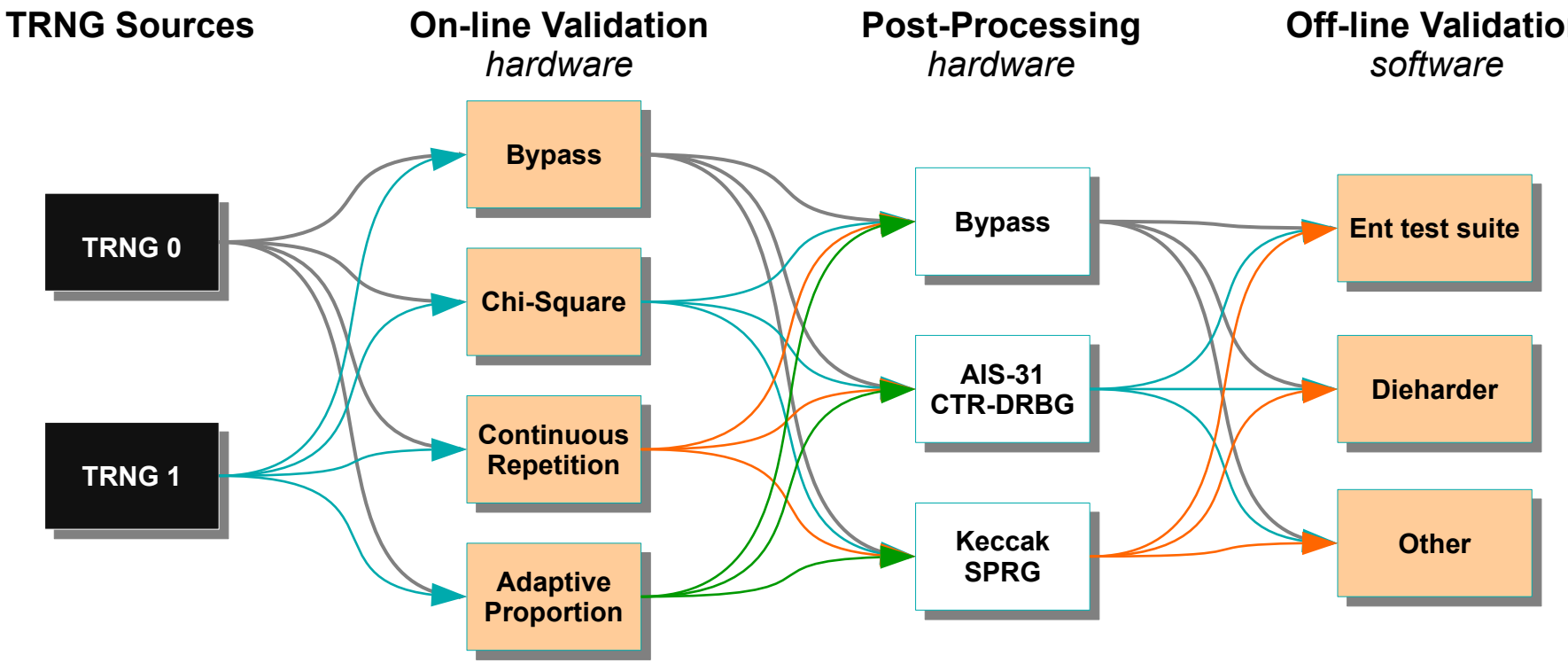
- Lots of **sensitive data** are handled by **IoT** devices
- These **low-power**, **low-performance** devices often use **poor quality pseudo-random generators** to secure their communication
- The use of **True Random Generators (TRNGs)** is a solution to provide **better quality** and **higher security**
- Issue: These TRNGs often provide **low throughput** and require **post-processing** to correct biases

Objectives

- Propose a **platform** able to **validate** and **post-process** multiple TRNG sources
- Propose a **hardware post-processing** algorithm to **improve** both **randomness** and **security**
- Propose a **flexible** platform that takes into consideration the variety of IoT devices with regards to **computation capacity** and **power consumption**

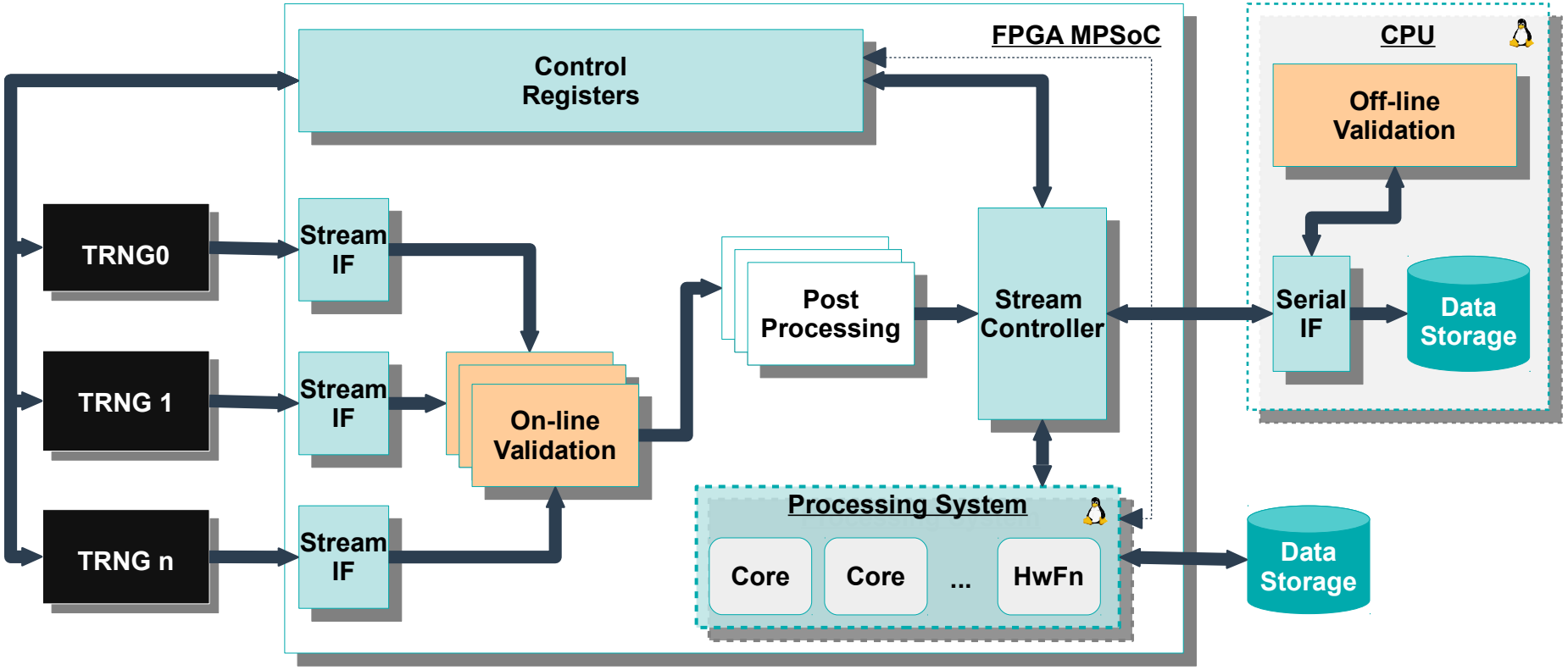
Validation Platform

- Allow to build a **custom datapath** going through **on-line validation** and **post-processing**
- Possibility to run exhaustive **off-line tests**
- The **most appropriate** TRNG source could be **selected** based on off-line test results



Flexible Platform

- Choice between **Processing System:** Computing resources for both on-line and off-line validation
- Or classic **CPU:** Comm. through **rapid serial interface** (PCIe) and **large data storage**

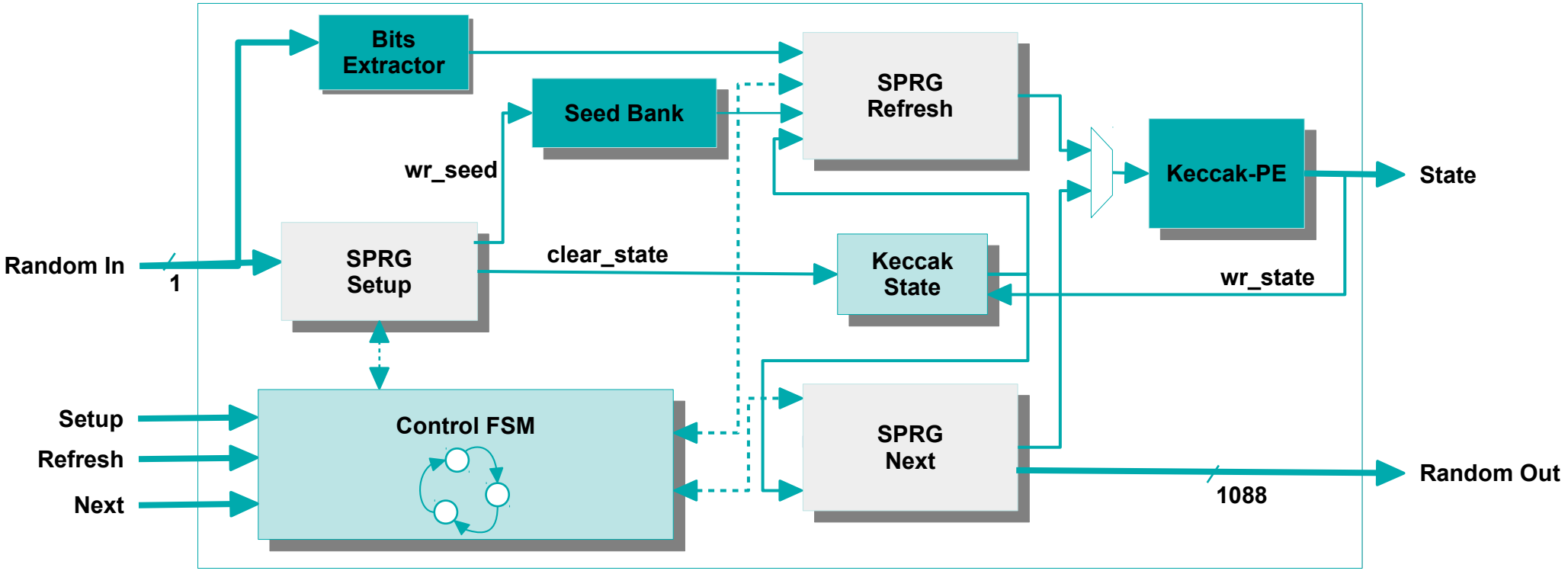


Hardware SPRG Post-Processing

- Based on **Keccak** (SHA3) cryptographic core
- **Provably secured** improvement of **sponge-based** PRNG
- Ensure **forward** and **backward secrecy**
- **Seed** support to improve **robustness**
- **State clearing** after generate to enhance **randomness**

Hardware SPRG Post-Processing Block Diagram

- **Setup:** Initialize the **seed bank**
- **Refresh:** Update the **Keccak** internal state with a seed
- **Next: Generate** random bits and clear a part of the Keccak state for security



Results

Resources

LUT	LUTRAM	FF	BRAM	DSP
16193	0	19856	15.5	0

Throughput

Output Width (Bit)	Frequency (MHz)	Throughput (Mbps)
1088	200	975.78

Speedup

Hw FPGA (ns)	Speedup / PC*	Speedup / MPSoC**
1115	x 2.97	x 42.46

* *Intel Core i7-7700*
** *ARM Cortex-A53*

DieHarder Test Suite

/dev/urandom with SPRG	***Passed tests***	**109 (97.3%)**
	Weak tests	3 (2.7%)
	Failed tests	0 (0.0%)
Quantis© RNG*	***Passed tests***	**62 (55.4%)**
	Weak tests	45 (40.2%)
	Failed tests	5 (4.5%)
Quantis© RNG* with SPRG	***Passed tests***	**108 (96.4%)**
	Weak tests	4 (3.6%)
	Failed tests	0 (0.0%)

* *TRNG device from ID Quantique SA*

Conclusion

- HwSPRG post-processing IP **improves the source randomness** and **successfully passes** the DieHarder test suite
- **Platform flexibility** makes it possible to **adapt** the validation flow to the selected TRNG source
- **On-line validation ensures** that random bits passes AIS-31 Chi-Square test or NIST SP800-90B recommended health tests *(Repetition Count and Adaptive Proportion tests)*
- **Off-line validation ensures** that the TRNG source is **always functional**

Future Work

- **Provide several versions** of the HwSPRG module

  → Offer trade-off between resource consumption and security level

- **Integrate new TRNG sources** into the platform

- **Increase flexibility** through FPGA's **dynamic partial reconfiguration**