THALES
Building a future we can all trust

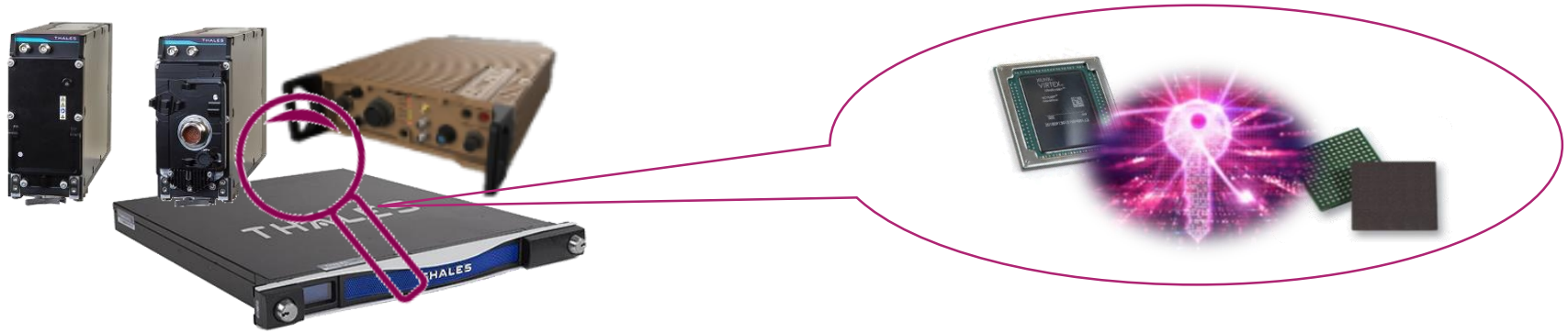
High-grade security TRNG

A practical application in industrial chip development

www.thalesgroup.com

Introduction

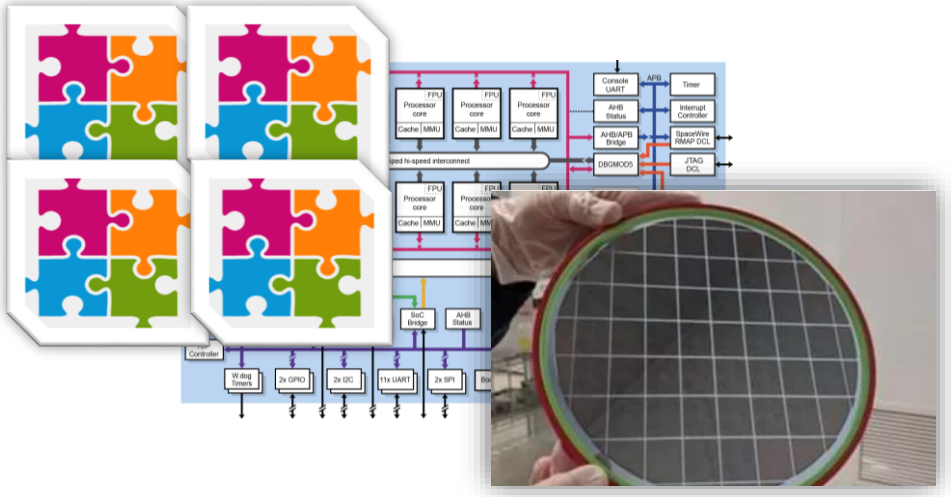
> Thales develops Cryptographic Devices, that embeds custom cryptographic chips (ASIC / FPGA)



> These chips are themselves complex systems, integrating several kinds of technologies : CPU, Interfaces, coprocessors, sensors, and also TRNG

> As chip provider

- We assemble IPs (building blocs) from partners
- We map the design on silicon
- We qualify the design (Functionality and security)
- We support evaluation & certification
- We produce and test the chips before delivery



Introduction

> Our chips fulfill external requirements

- Functionalities
  - Cryptographic services, communication interfaces, …
- Performances
  - Throughput, consumption, area, …
- Security insurance (compliancy with referential)
  - BSI AIS31
  - NIST SP800-90B
  - DGA-MI (French MOD) evaluation referential
    - "Recommendations for the Design and Validation of a Physical True Random Number Generator Integrated in an Electronic Device"
    - https://eprint.iacr.org/2024/301

Introduction

> And we also have to deal with constraints associated with industrial development and production

- Reliability / Reproducibility
  - Of the product itself
  - But also of the supply chain (including all the tools used for development, production, and testing)
- Efficiency / Competitiveness
  - Development and production efforts have to be kept coherent with a target market
- Risk management
  - No room for chance
- Dependencies management
  - Ability to change characteristics or parts of a design without restarting from scratch
  - Ability to switch to up-to-date Silicon technologies

TRNGs have several characteristics that are difficult to reconcile with theses constraints

State of the art TRNG design

> Based on a reliable source of noise

- Always present, unpredictible
- Cannot be influenced by an attacker

> Resiliant to common attacks

- Add on-chip countermeasures if neaded
- Add health test, coherent to idenfied potential failures of the design

> With respect to industrialisation constraints

- TRL level > 6
- Performance : area, throughput, alea quality
- Compatible with our constraints : integrated in the same chip, available in ASIC and FPGA…

State of the art TRNG design

> Based on a reliable source of noise

- Always present, unpredictible
- Cannot be influenced by an attacker

> Resiliant to common attacks

- Add on-chip countermeasures if neaded
- Add health test, coherent to idenfied potential failures of the design

> With respect to industrialisation constraints

- TRL level > 6
- Performance : area, throughput
- Compatible with our constraints : integrated in the same chip, available in ASIC and FPGA…

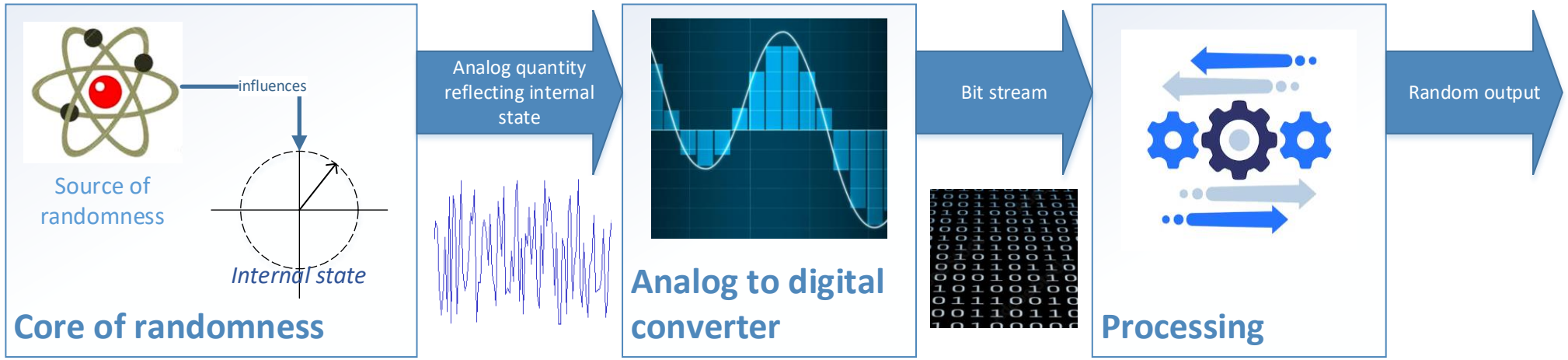
Bad candidates :

```
>cat /dev/random
```

Unix kernel random source : CPU's tick counter LSB bits at interrupt time.
Influence alea ⇔ use the keyboard

Good randomness quality. Not ASIC/FPGA integrable

State of the art stochastic model



> Close modelling of the source

- Starting with the physical noise itself
- Describing all the steps through the random generation process

> Modeling easy to explain and understand

- Keep the needed scientific background not to high
- Easy to defend the model during certification process.

> Aim of the model : lower bound to the entropy output

- Underestimation close enough to reality to limit performance drop
  - Low entropy ➔ need to compress ➔ decrease througput/area performances

Measurement methodology

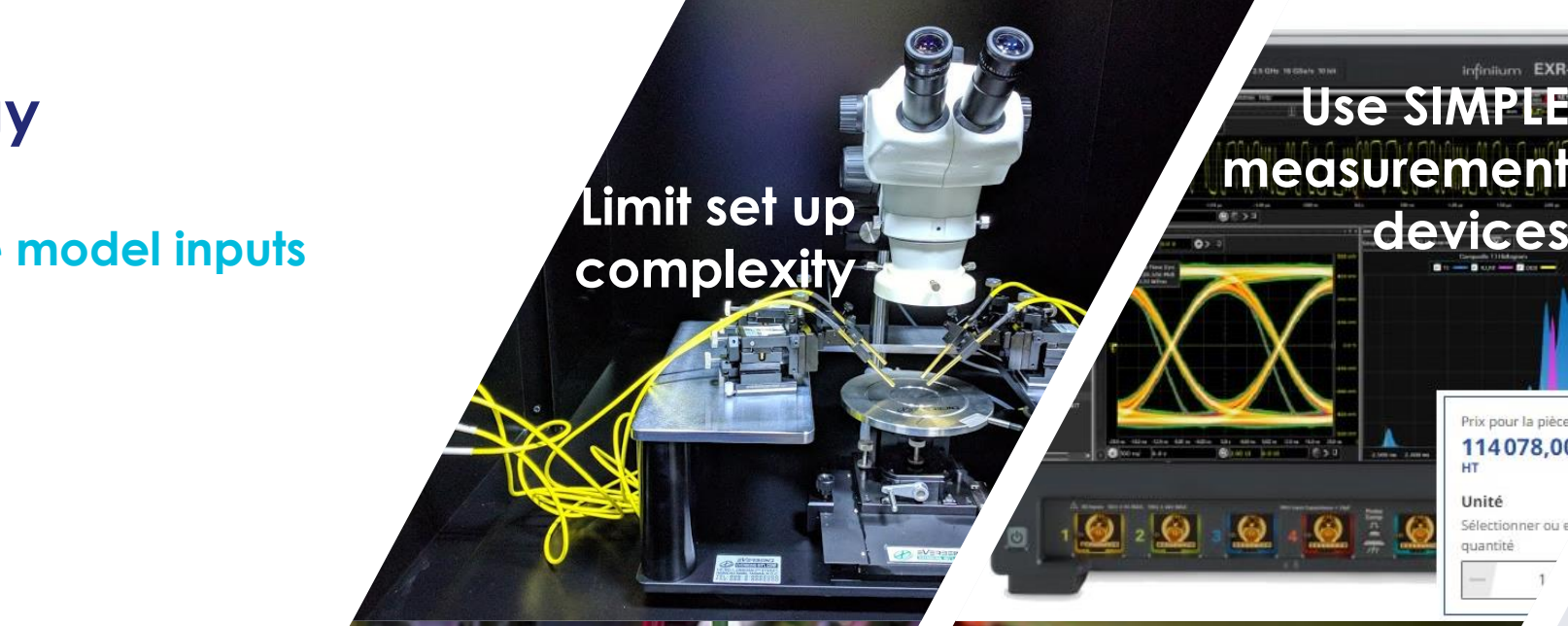
> Aim : get numerical values for the model inputs

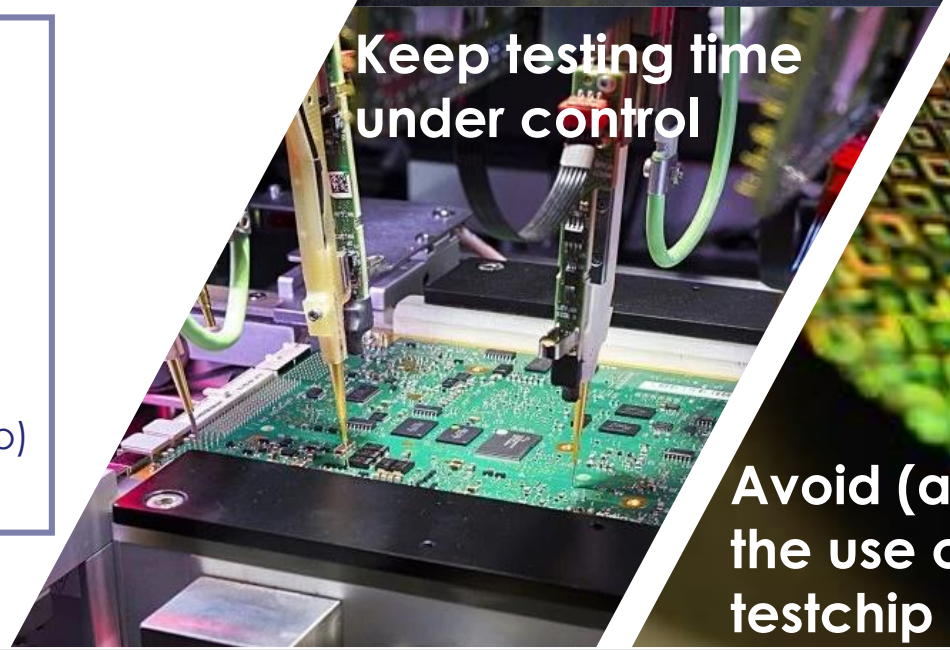
> Accurate measurement

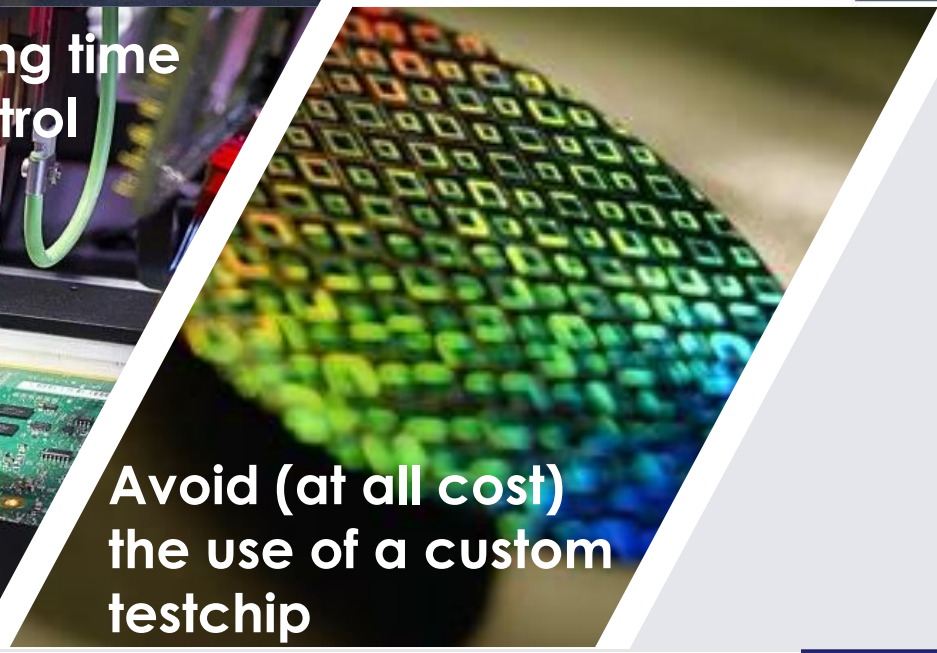
- Without affecting nominal operation
- Qualified error margin

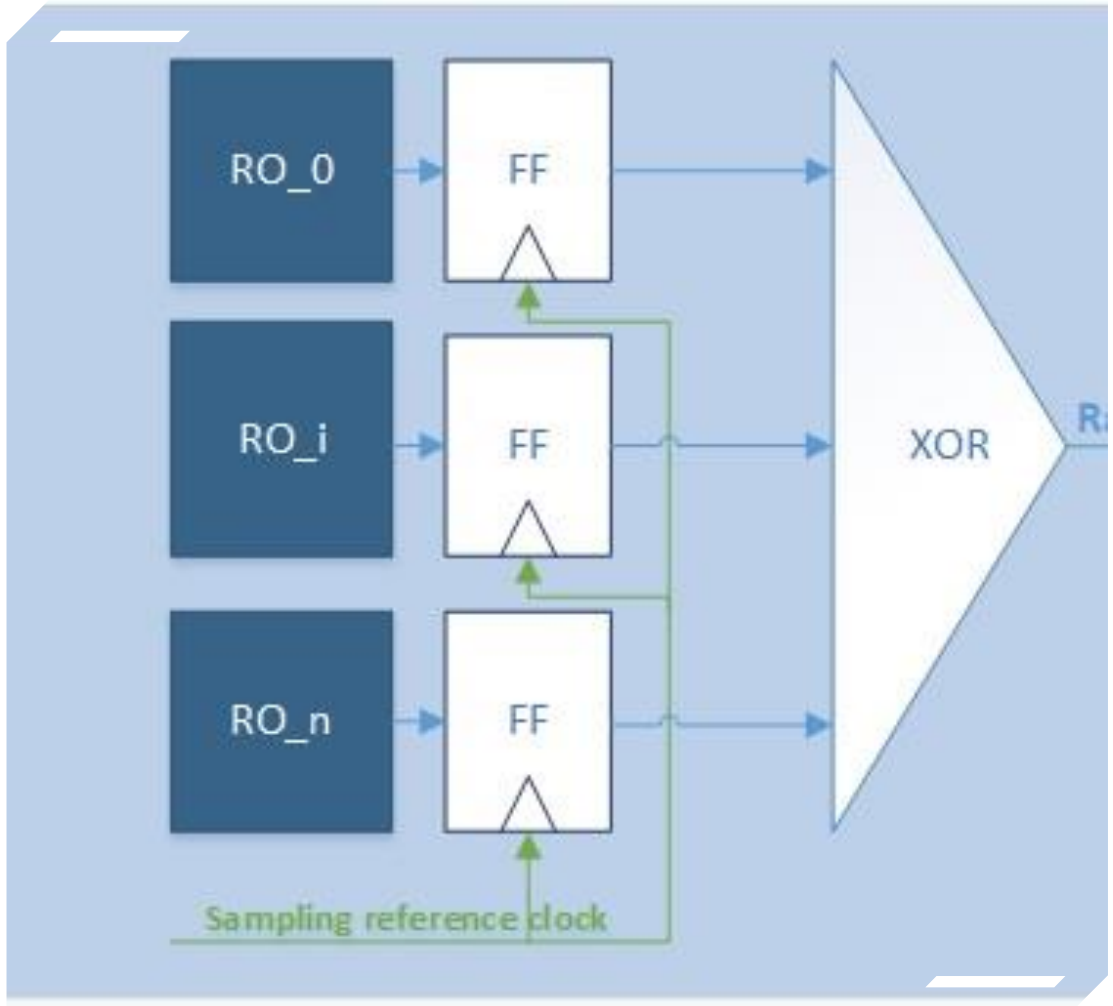
> Compatible with an industrial workflow

- Simple, repeatable, automatisable measurement process
- Afordable cost
- Dispose of early predictive value (have an idea of the result BEFORE producing the chip)









Simplified *schematic of a RO based TRNG*

Candidate solution : RO-based TRNG

> RO-based TRNG : a concrete example

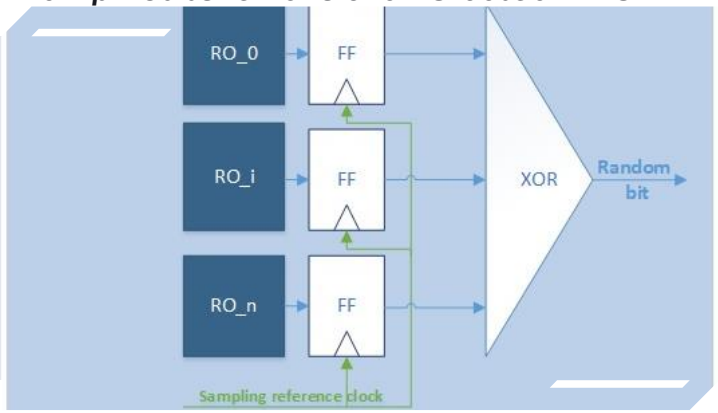
- Ring oscillators (RO) XORed together, sampled by a RO provided clock

> My point in the following slides

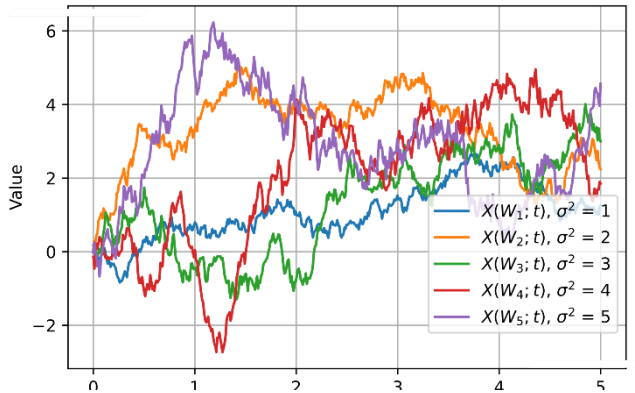
- Point out the discrepancies between "a TRNG design" and the reality of implementing that design in a real chip
- Highlight the costs of work-around measures we had to take

Candidate solution : RO-based TRNG

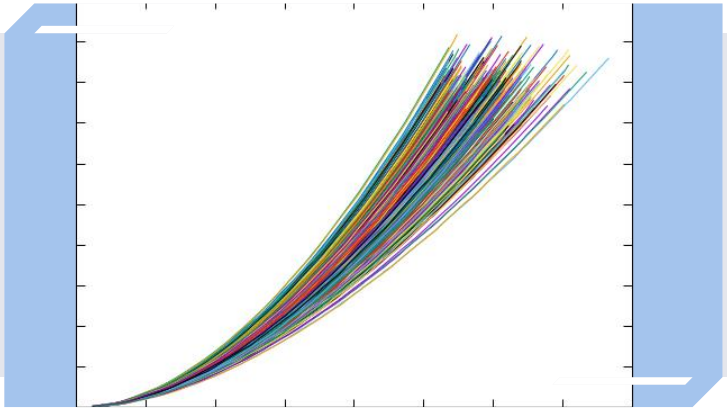
Simplified *schematic of a RO based TRNG*



Wiener process modelizing RO phase



Ouptut of the Internal Method : variance of the Wiener process as a time function



> State of the art TRNG design

- Noise : thermal noise
- State of the art compliant
- Independent from techno (ASIC, FPGAs compatible)
- Low area footprint
- Mature design

> Stochastic model

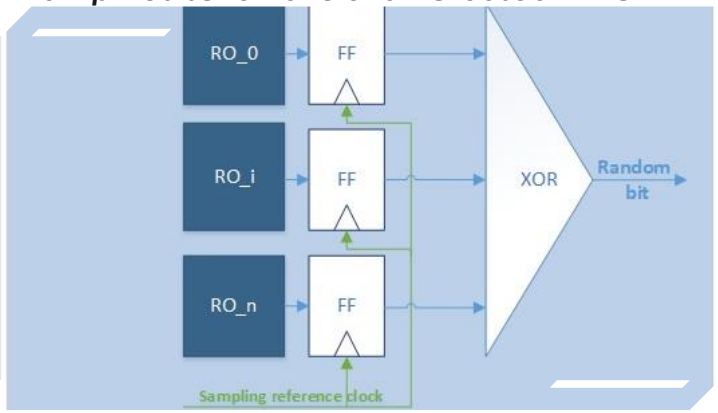
- RO phase modelization by a Wiener process
  - Model the whole process from noise to bit output
  - Based on [BLMT11]
- Conservative entropy lower bound
- "Simple enough"

> Measurement methodology

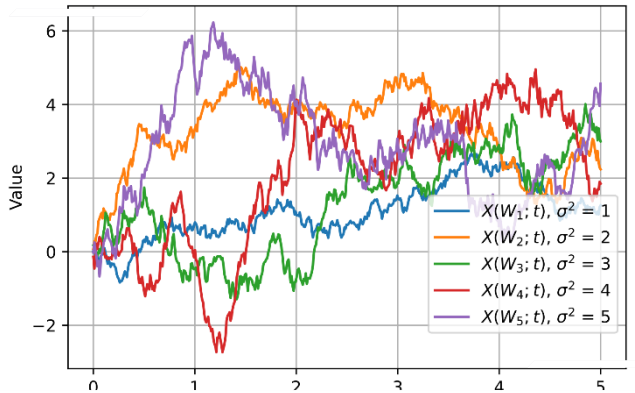
- On chip measurement
  - Based on the Internal Method from [FL14]
  - Use the operational datapath
- Needed tools :
  - logic analyser
  - Spreadsheet software

Candidate solution : RO-based TRNG

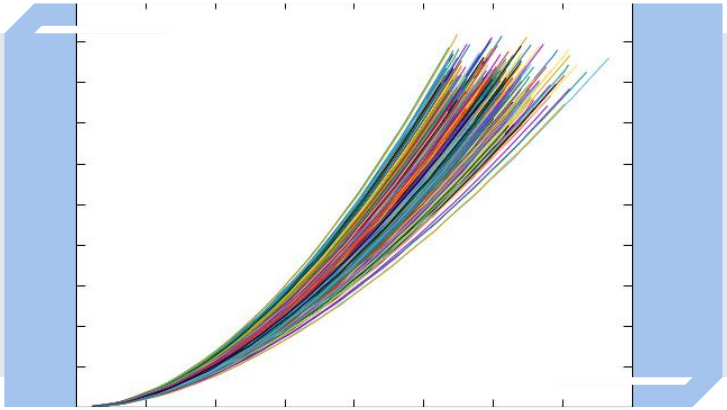
Simplified *schematic of a RO based TRNG*



Wiener process modelizing RO phase



Ouptut of the Internal Method : variance of the Wiener process as a time function



> State of the art TRNG design

- Noise : thermal noise
- State of the art compliant
- Independent from techno (ASIC, FPGAs compatible)
- Low area footprint
- Mature design

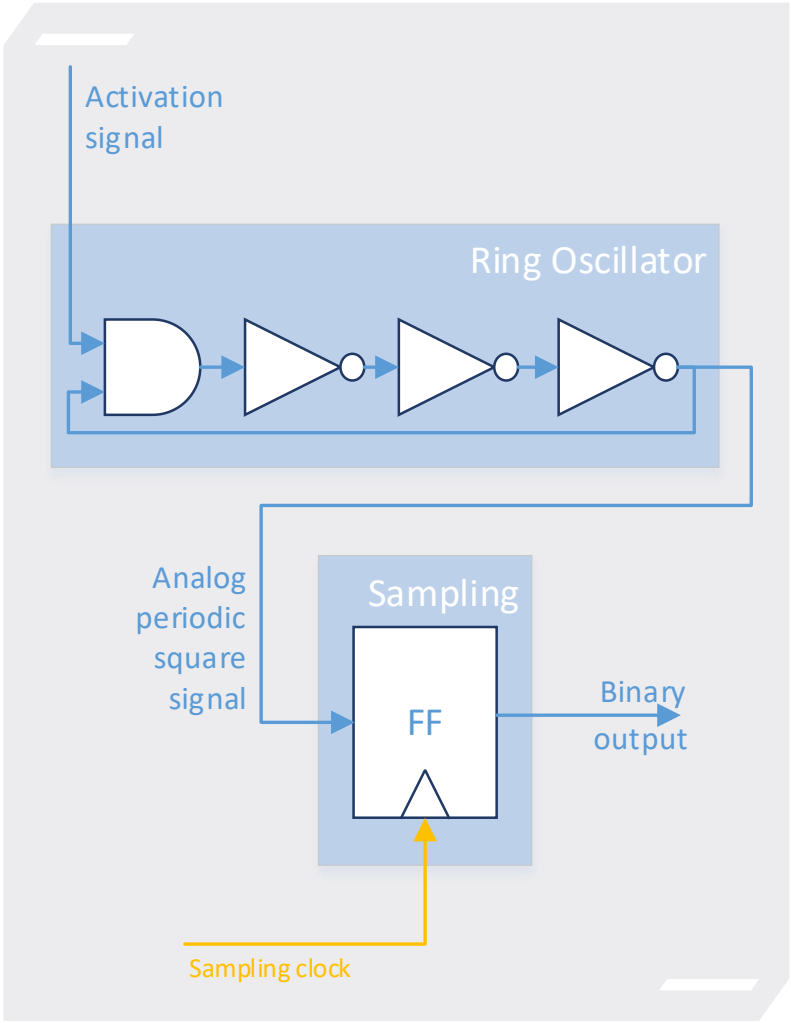
> Stochastic model

- RO phase modelization by a Wiener process
  - Model the whole process from noise to bit output
  - Based on [BLMT11]
- Conservative entropy lower bound
  - Pessimistic premises to simplify the maths

> Measurement methodology

- On chip measurement
  - Based on the Internal Method from [FL14]
  - Use the operational datapath
- Needed tools :
  - logic analyser
  - Spreadsheet software

OPEN



RO versus Design rules and P&R tools

> Combinational loops

- Forbidden (« highly not recommended ») by P&R tools
  - ➔ ***WORK AROUND*** : Turn off P&R tools for the loop (resulting in a timing and DRC blind spot)

> Asynchronous source at flip flop input

- Timing tool unable to help (flops are not intended to sample 'full random' input)
- Metastabilty unavoidable by design
  - ➔ ***WORK AROUND*** : Limit frequency will reduce metastability probability

> Mismatching objectives for RO designer versus P&R tools

- P&R tool will limit routing time as much as possible where the designer would prefer having routing evenly distributed between NOT gates
  - ➔ ***WORK AROUND*** : Strong placement constraints for the RO's gates

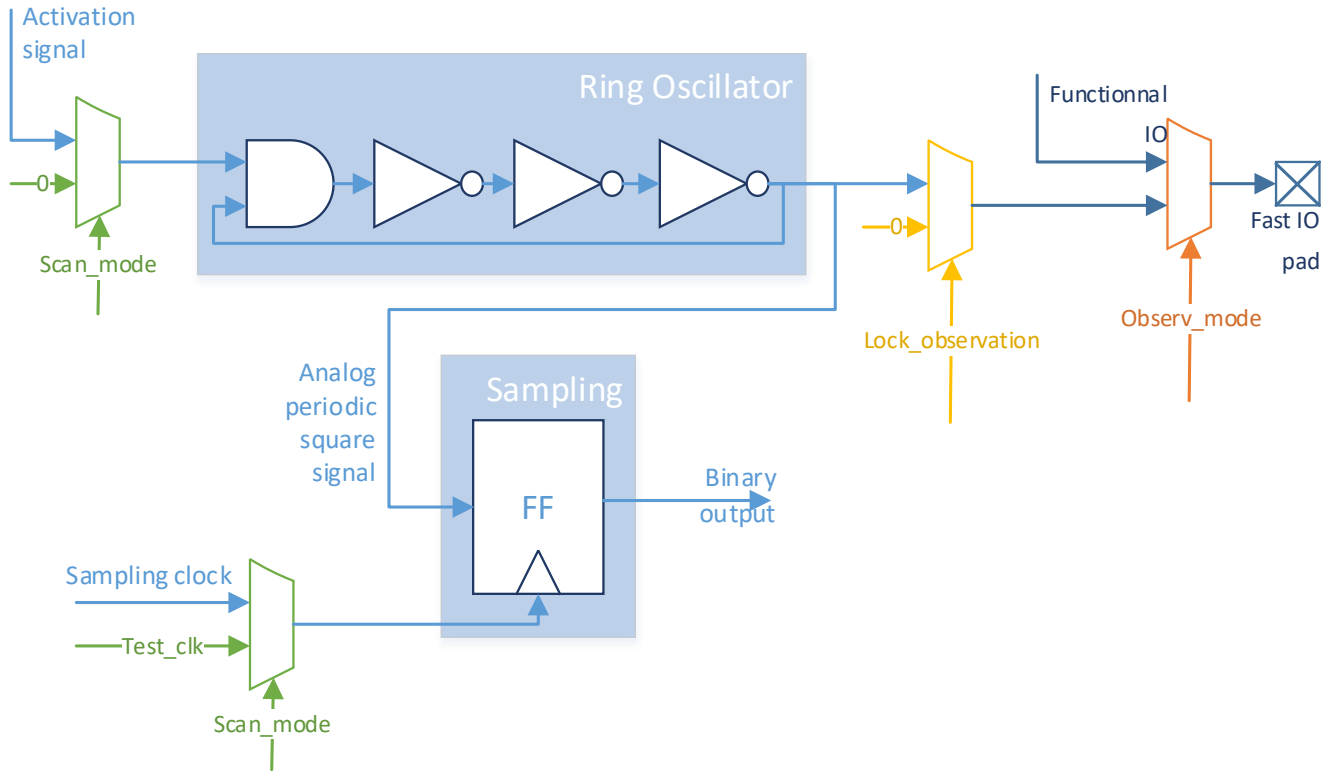
Simple RO design to full integrated DFT compliant ring

> Boundary scan insertion

- Add **logic** to disable ROs during test
- Use **custom DFT scheme** to handle ROs output flops

> Obsevability on inner TRNG signals

- Needed in the qualification process (ex get ROs frequencies…)
- Take precaution while exposing ROs off-chip
  - Potential security breach if an attacker is able to spy ROs output
  - Ex : **locks on debug paths**
- Use appropriades IO pads
  - Compliant with RO frequency
  - Consider **muxing with operational** IO, to save IO pads ➔ increase DFT complexity



OPEN

Issue : performance anticipation

> Integration constraints

- A constrained budget is allocated to the TRNG :
  - Occupied area
  - Minimal througput for a given output entropy

> Pb : entropy/througput is characterized *once* the chip have been produced

- We cannot afford yield loss due to a TRNG failing to achive its entropy/throughput constraint

How we respond

> Capitalize on previous work

- De-risking work on ‘new’ FPGA technologies
- Numbers from previous ASIC design
- In order to extrapolate entropy output for a given design

> Take margins

- Add spare ROs
- Take worst case as premises
- Margins increase occupied area (more ROs for spare, more ROs to achieve entropy output)

Issue : ensure reproductibility

> Process dispersion has a major effect on RO

- Affects RO's duty cycle and frequency
- Duty cycle : not covered by the datasheet
- Frequency range : ***very*** large, according to the datasheet

> Pb : characterization time for 1 piece with our method : 0.5 day

- Impossible to sort all the pieces

How we respond

> Design & placement precautions

- Do not use small ROs (more vulnerables to process dispersion)
- Distribute routing evenly between NOT gates
- Isolate ROs from the rest of the logic

> Caracterization process :

- Apply representatives PVT conditions (worst, best, typical…)
- Take the ***worst*** results as input for the model
- Add margins
  - Error margins related to the measurement methodology, the model…
  - Conservative margins to cover blind spots (ageing, chip activity…)

Conclusion

> **Most of the difficulties we illustrates are inherent to the fact that we want to extract randomness from physical resources, and associated tools, that are meant to produce determinism.**

> **Nevertheless, easier solutions might exists to facilitate use and integration of such IP**

> **We hope this testimony will help the community to improve TRNG and facilitate there industrial integration onchip**

- New design solution/pattern
- New characterization tools / methodology
- Or less stringent evaluation criteria for less critical use cases

Bibliography

> [LF24]

- Lubicz, D., Fischer V. *Recommendations for the Design and Validation of a Physical True Random Number Generator Integrated in an Electronic Device*, ePrint archive 2024

> [BLMT11]

- Baudet, M., Lubicz, D., Micolod, J., & Tassiaux, A. (2011). *On the Security of Oscillator-Based Random Number Generators*. Journal of Cryptology.

> [FL14]

- Fischer V., Lubicz, D. . CHES 2014. St-Malo: Springer