KU LEUVEN

COSIC

On-chip jitter measurement

for

True Random Number Generators

Bohan Yang, Vladimir Rožić, Miloš Grujić
Nele Mentens and Ingrid Verbauwhede
COSIC, KU Leuven

HECTOR

Outline

- **Random Number Generator**
- **Jitter measurement**
  - Why?
  - Why on chip?
  - Other problems
  - How?

1. Finding and calibrating a "ruler".

2. Using this "ruler" to measure.

- **Conclusion and future work**
- **Q&A**

工欲善其事，必先利其器。

Random Number Applications

Games

Cryptography

Scientific Computation

Session Keys

Signature Parameters

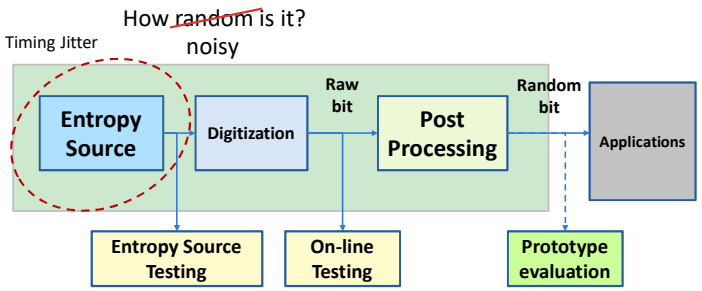
Stochastic Simulations

Challenges

Masking

Numerical Analysis

Generic TRNG Architecture



Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens
and Ingrid Verbauwhede, COSIC, KU Leuven

Design methods

Old way:

- FIPS
- NIST
- DIEHARD

Random Number Generator → 10110001010… → **Statistical Tests** → PASS/FAIL

Overestimating your entropy results in a compromised security ⇒ **Use lower bound !**

New method:

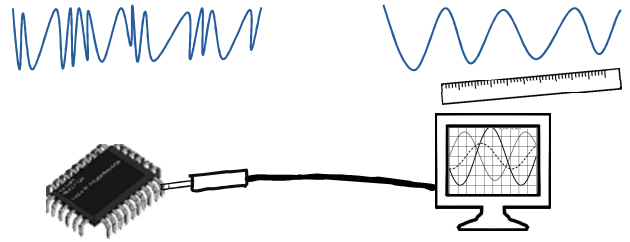
Experiments → Platform parameters → **Stochastic Model (AIS31)** DGA NIST 800-90B? → Entropy claim

Design parameters →

Assumptions → Experiments

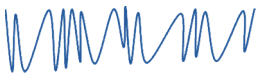
Why jitter shouldn't be measured off-chip

Off-chip jitter measurement



Jitter measurements on-chip

FPGA Xilinx Spartan6



Counter Quantization noise

LUT

~600ps

Large delay

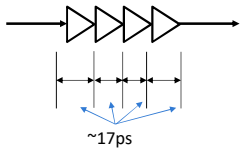
CARRY4

~17ps

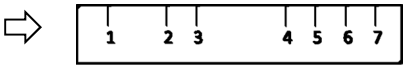
Designed to be fast
Dedicated routing

Best resolution

Jitter measurements on-chip



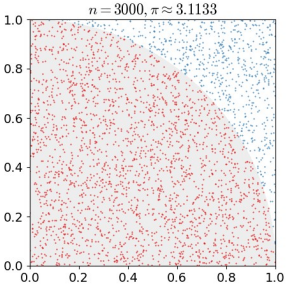
~17ps



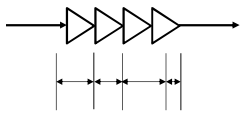
Calibration is mandatory

Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens
and Ingrid Verbauwhede, COSIC, KU Leuven

Monte Carlo Method

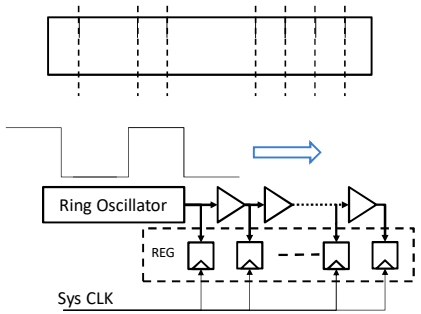


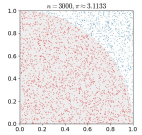
By nicoguaro - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=14609430



How to use MC to calibrate delay chain?

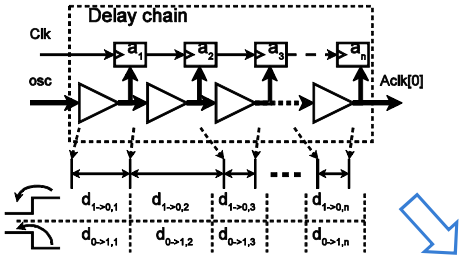
Monte Carlo Method

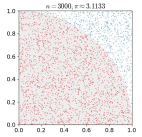




By nicoguaro - Own work, CC BY 3.0, ttps://commons.wikimedia.org/w/index.php?curid=14609430

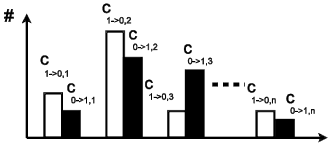
Monte Carlo Method





By nicoguaro - Own work, CC BY 3.0, ttps://commons.wikimedia.org/w/index.php?curid=14609430

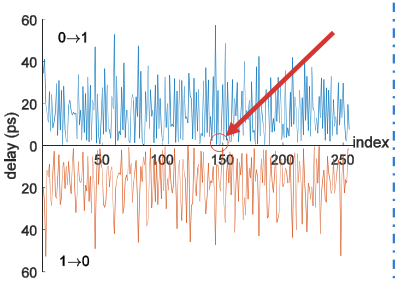
Longer delay → Higher counts



PUF?

The Monte-Carlo PUF @ FPL17

Other problems



Average: 16.8 ps

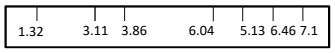
Problems: **Elements with zero delay**

Carry Look ahead adder

00000011011111111....0001111

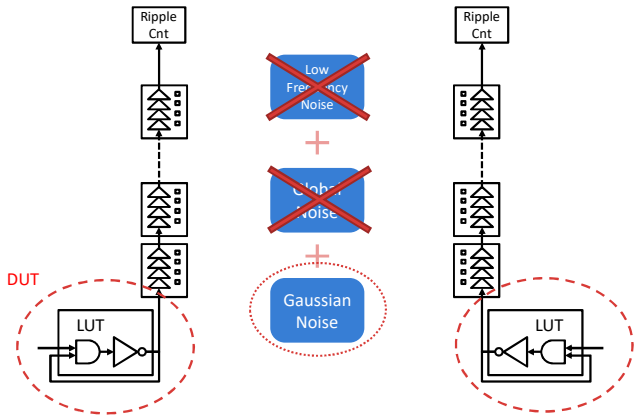
11111010000000000....1111000

Bubbles



Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens
and Ingrid Verbauwhede, COSIC, KU Leuven

Jitter measurement



Jitter measurement

Snapshot 1:
1111111100……1111111000001111111 0000000
Ripple Counter 1: 5

Snapshot 2:
1000000000……0000011111111 0000000 0111111
Ripple Counter 2: 5

t_diff

Snapshot 1:
1111111100……1111111 0000001111110000000
Ripple Counter 1: 5

Snapshot 2:
1000000000……0000011111111 0000000 0111111
Ripple Counter 2: 6

t_diff

Jitter measurement result on Spartan6

Jitter accumulated during t_m:

$$\sigma_m(t_m) = \sqrt{\frac{\sum_{i=1}^{N_{exp}} (t_{diff,i} - \bar{t}_{diff})^2}{k-1}}$$

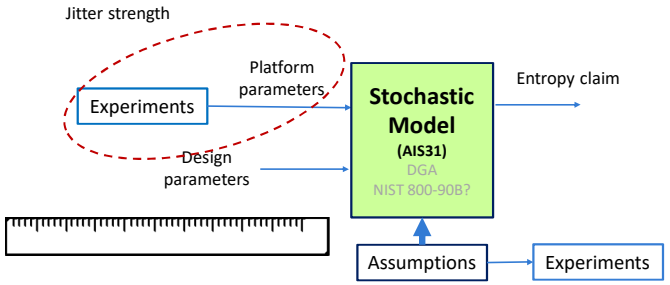
Jitter accumulation rate:

$$\frac{\sigma_m^2}{t_m}$$

Xilinx Spartan6 lx45	$\frac{\sigma_m^2}{t_m}$	$\sigma_m(10ns)$ (working @100MHz)	
With differential method	$22\ fs$	$5.26\ ps$	Gaussian noise
~~With single delay line~~	$154\ fs$	$13.9\ ps$	Gaussian noise + Global noise
~~Counter~~	$11\ ps$	$118\ ps$	Gaussian noise + Global noise + Low frequency noise

Overestimation!!

Platform parameters



Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens
and Ingrid Verbauwhede, COSIC, KU Leuven

Conclusion

Jitter measurement

- Why? To help us design and understand TRNGs.
- Why on chip? Off-chip measurement is inaccurate.
- Other problems Resolved
- How? Differential setup Short measurement time
  Collect data Apply equations ☺

Ruler is found, calibrated and used !!!!!

Future work

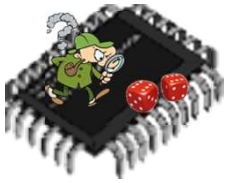
- Jitter measurement for other applications and platforms
- Implement it as the on-line testing module
- Validation of the robustness of the jitter measurement over Voltage and Temperature variations.

Q&A

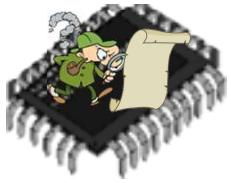
HECTOR

The project leading to this paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644052

Appendix: Random Number Generators



True Random Number Generator

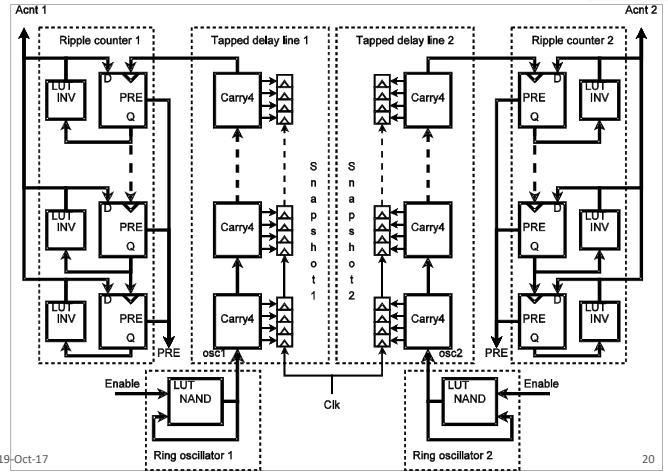


Pseudo-Random Number Generator

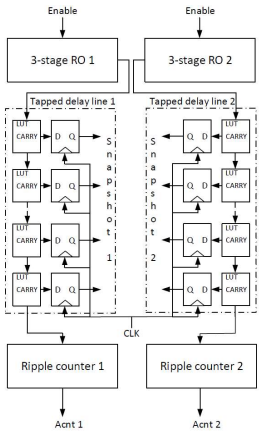
- **TRNG -> the root of a cryptographic system**
- **TRNG -> the single point of failure**
- **How good is your TRNG ?**

Bohan Yang, Vladimir Rožić, Miloš Grujić, Nele Mentens and Ingrid Verbauwhede, COSIC, KU Leuven

Appendix: Arch for Xilinx FPGAs



Appendix: Arch for Intel FPGAs



Appendix: Measurement result