- Cryptarchi 2012 -

# A NOVEL TRUE RANDOM NUMBERS GENERATOR USING SELF-TIMED RINGS

**Karim CHERKAOUI (LaHC)**
Laurent FESQUET (TIMA)
Viktor FISCHER (LaHC)
Alain AUBERT (LaHC)

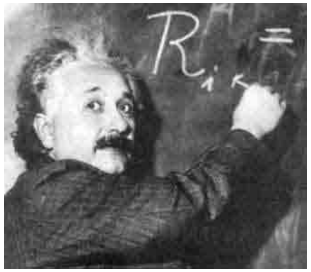
**LaHC / TIMA lab**
**Rhônes-Alpes Region PhD Fund**
**SEMBA PROJECT – ISLE Cluster**

**June 2012**

**Université Jean Monnet– St-Etienne FRANCE**

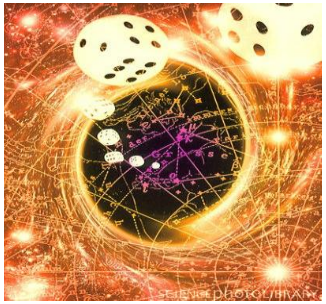
# Randomness in digital devices

- Macroscopic phenomena seem deterministic

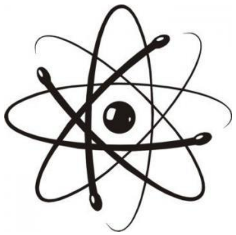


*« God does not play dice with the universe »*
*Albert Einstein*

- Standard interpretation of quantum mechanics: **microscopic phenomena are objectively random**

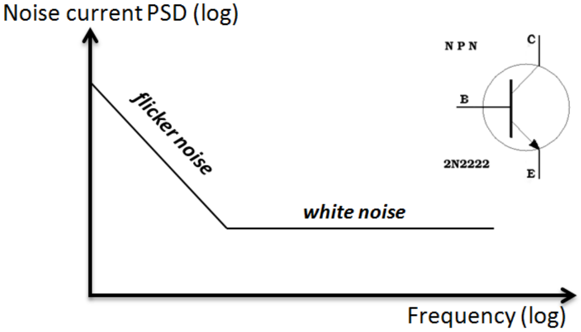


**Subatomic particles**

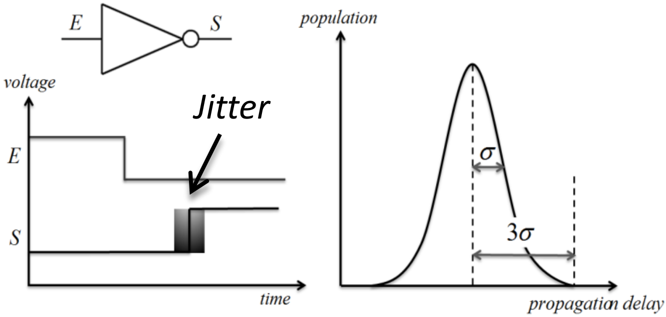


*Thermal agitation, shot noise …*

**transistors**



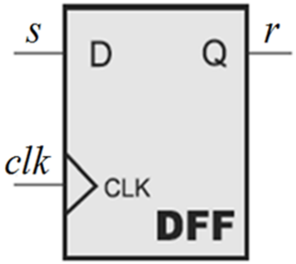
**logic cells**

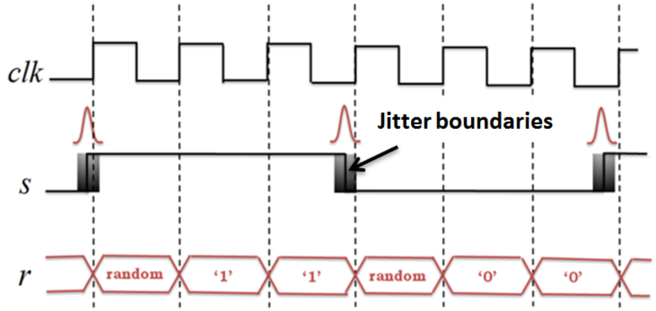


# Generating Random Numbers Using Jitter

**Jitter** Variations of a digital signal significant instants from their ideal position in time as a consequence of noise in electronic devices

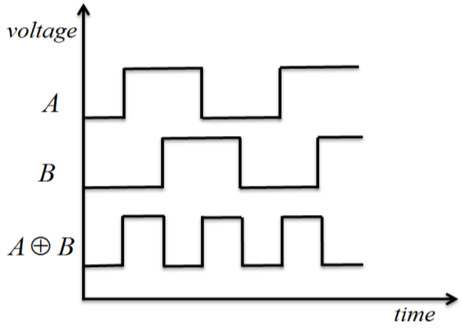
**How do we extract random numbers from jitter ?**





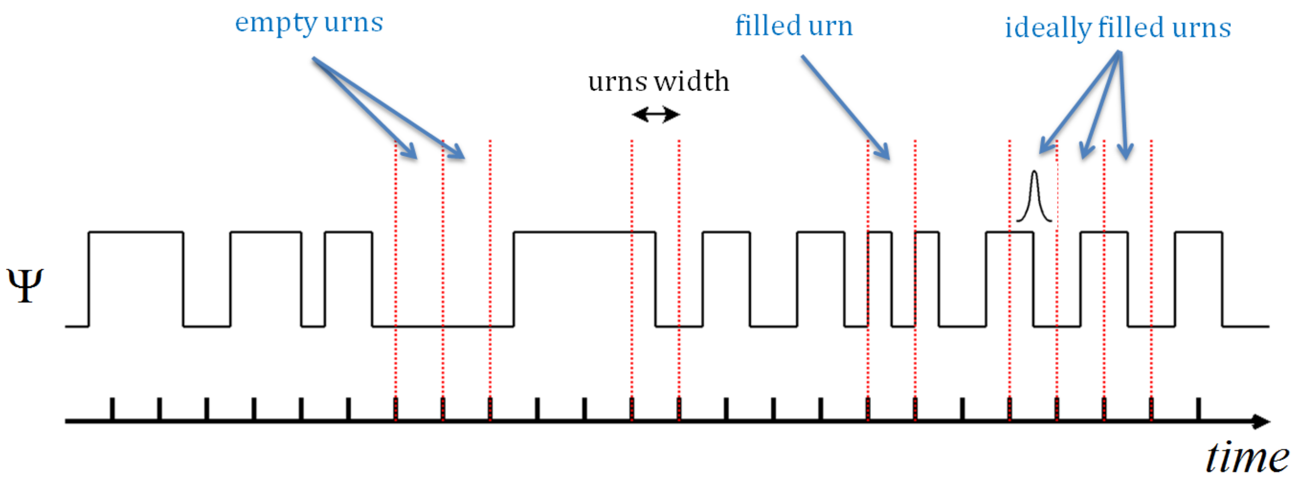
**Challenge** Jitter magnitude is very small (usually <1% of the oscillation period)

➡ **Combine several signals to reduce the time lapse between successive events**



# Combining Signals to Harvest Randomness

- The time domain is divided into equally matched intervals called « Urns »

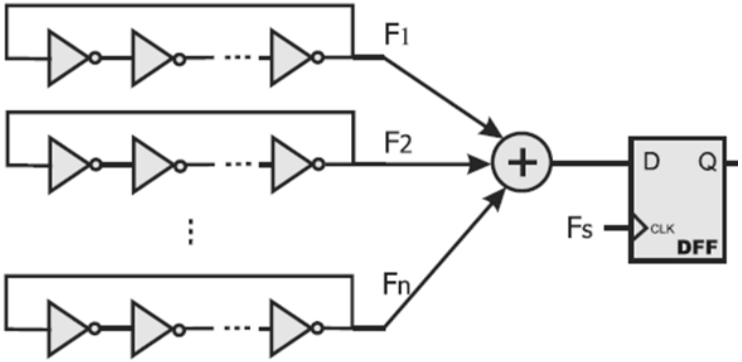


**urns width ~ jitter boundaries**

**How to fill all urns with at least one event per urn?**

# Combined RO-TRNG (1)

**Sunar's Approach** Uses several ring oscillators having the same frequency [1]



**<u>Drawback</u> No control of the relative phases of each oscillator**

- ☐ The number $L$ of uniformely random selections of $N$ urns such as each urn is selected at least once

**Coupon collector problem** →

$$L \approx N \log(N)$$

**Results** 114 rings of 13 stages + resilient function using BHC codes to achieve a satisfactory filling rate and entropy per bit

---

[1] Sunar *et. al. « A provably secure true random numbers generator with a built-in tolerance to active attacks »*

# Combined RO-TRNG (2)

**+++**

- Stochastic model
- Easy to implement

**---**

- Size / electrical consumption
- Locking and dependency [2]
- True randomness VS pseudo-randomness [2]

**Presented work** A new way to uniformely fill the time domain with events using one ring oscillator called self-timed ring

---

[2] V. Fischer *et.al. « True randomness and pseudo-randomness in RO-based TRNGs »*

# Outline

- Combined Ring Oscillators TRNG

- A Novel TRNG Architecture Using Self-timed Rings
  - TRNG Architecture and Principle
  - Self-timed Ring Architecture and Behavior

- Stochastic Model for Bias and Entropy Estimators
  - Maximal Bias and Minimal Entropy per Bit
  - Bias Correction

- Experimental Results
  - Implementation and Adequation to the Model
  - Period and Jitter Measurements
  - Statistical Evaluation

- Conclusion / Future Works

# Outline

- **Combined Ring Oscillators TRNG**

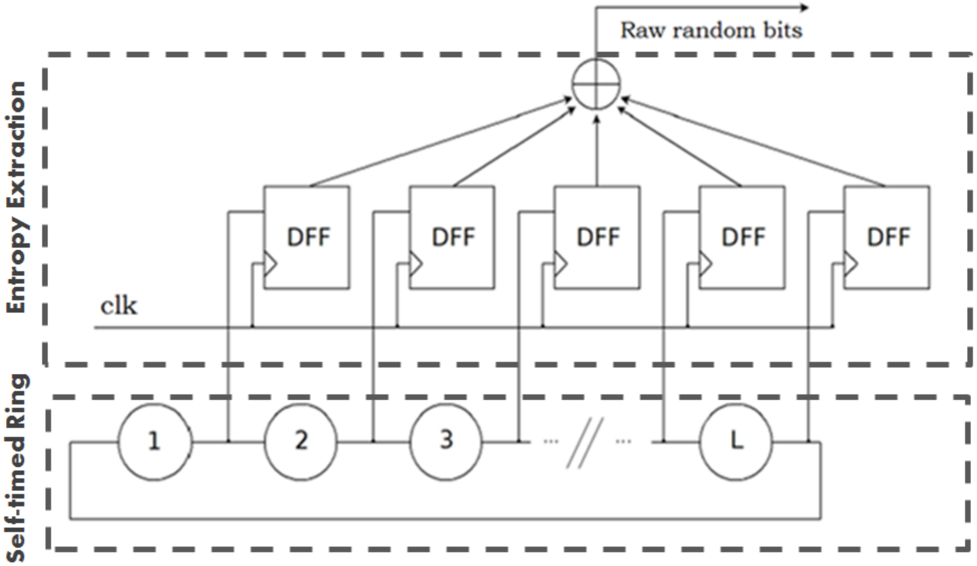
- **A Novel TRNG Architecture Using Self-timed Rings**
  - **TRNG Architecture and Principle**
  - **Self-timed Ring Architecture and Behavior**

- **Stochastic Model for Bias and Entropy Estimators**
  - **Maximal Bias and Minimal Entropy per Bit**
  - **Bias Correction**

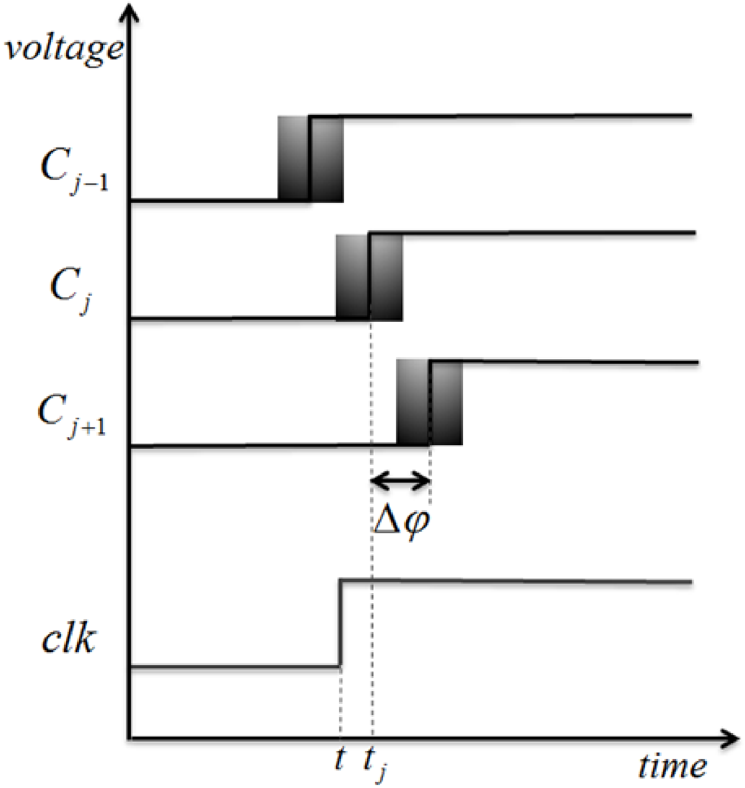
- **Experimental Results**
  - **Implementation and Adequation to the Model**
  - **Period and Jitter Measurements**
  - **Statistical Evaluation**

- **Conclusion / Future Works**

# TRNG Principle and Architecture



**Phase resolution ~ jitter boundaries**

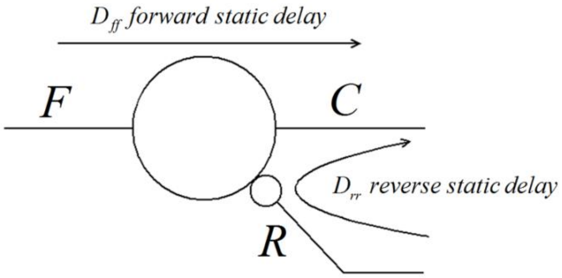


**Self-timed ring** provides equi-distantly distributed signals with a built-in control of their relative phases

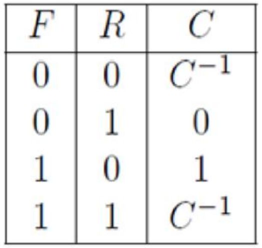
**Entropy extractor** each signal is sampled with the same global clock, a random bit is extracted using the XOR tree

# Self-timed Ring Architecture

- Asynchronous micropipeline closed to form a ring

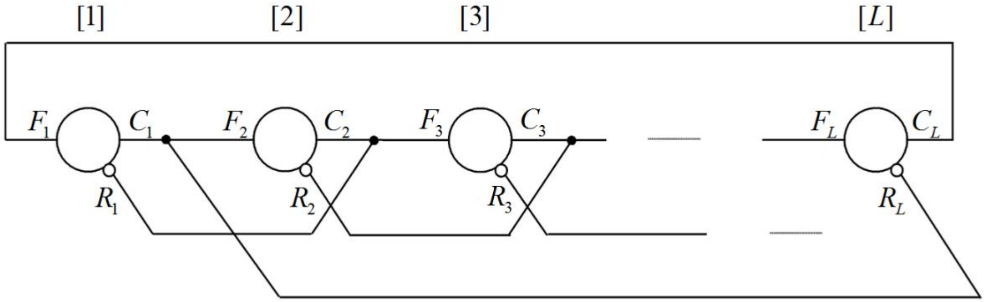


(a)



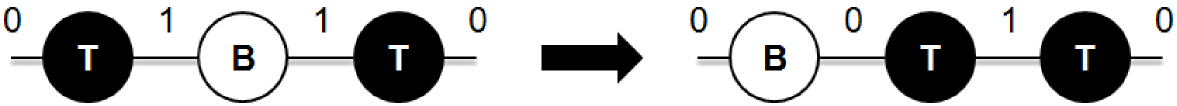
| $F$ | $R$ | $C$ |
|---|---|---|
| 0 | 0 | $C^{-1}$ |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $C^{-1}$ |

(b)



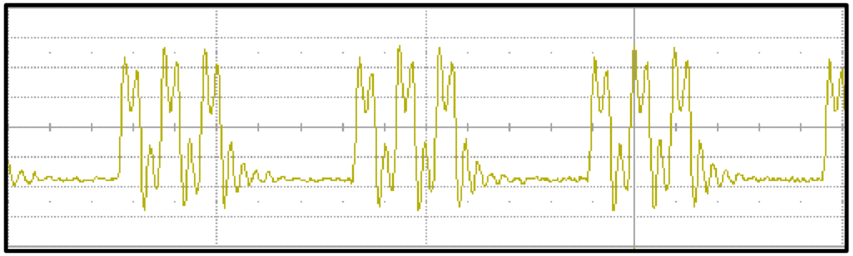
(c)

- The micropipeline stages communicate using a handshake request/acknowledge protocol

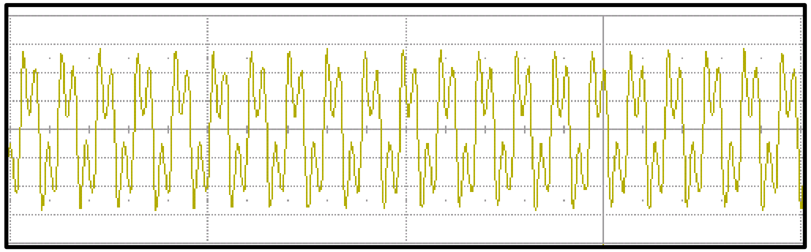


# Temporal Behavior of STRs (1)

- Two oscillation modes



**Burst mode**



**Evenly-spaced mode**

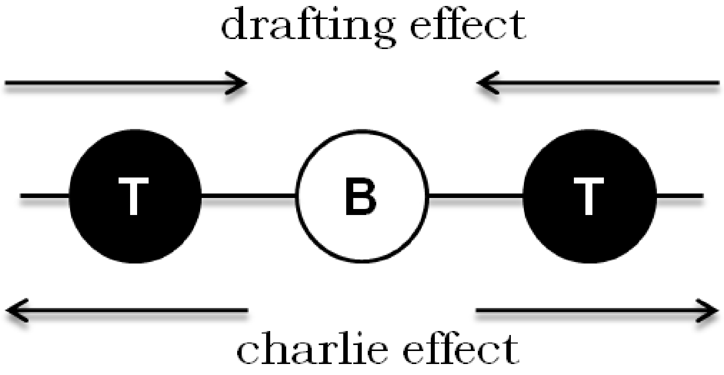
- Temporal behavior of a ring stage determined by two analog phenomena

**The Charlie effect**

The closer are the inputs events the longer is the stage propagation delay

**The Drafting effect**

The shorter is the time between two successive output commutations, the shorter is the propagation delay

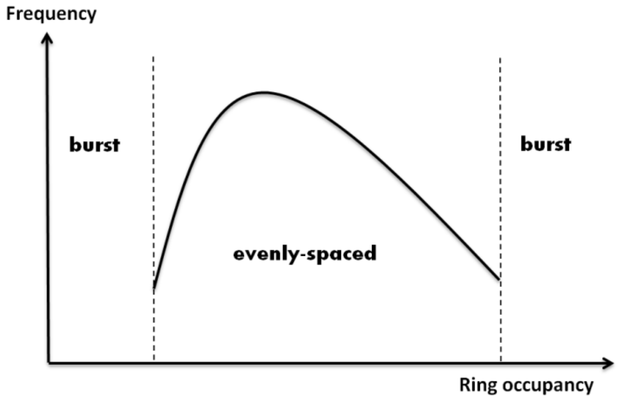


# Temporal Behavior of STRs (2)

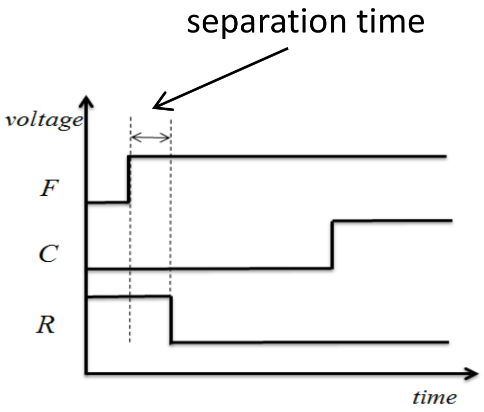
- How to set the evenly-spaced oscillation mode ?

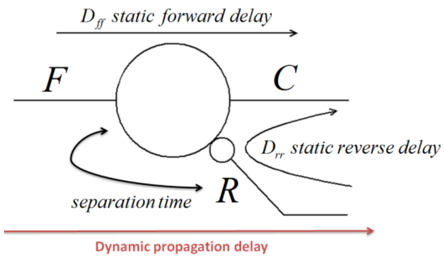
number of events → $N$
number of ring stages → $L$

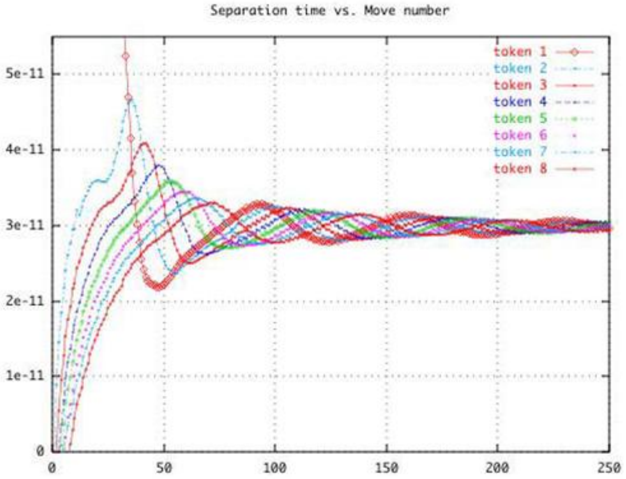
$$\frac{N}{L-N} \sim \frac{D_{ff}}{D_{rr}}$$



- The elapsed time between successive events is **auto-controlled**







[3]

---

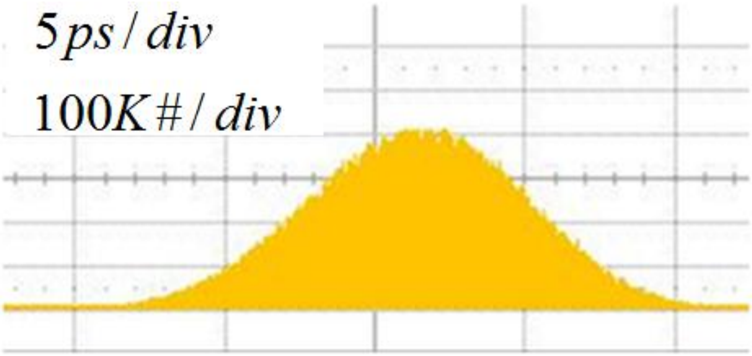
[3] S. Fairbanks and Simon Moore « *Analog micropipeline rings for high precision timing* »

# Jitter in Self-timed Rings

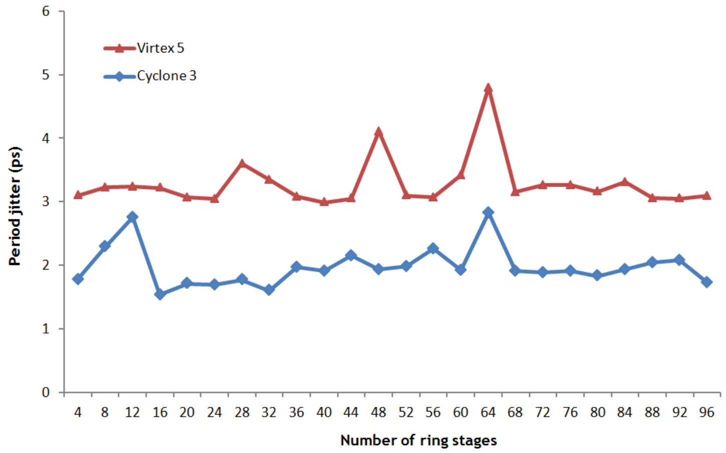
- The elapsed time between events is auto-controlled

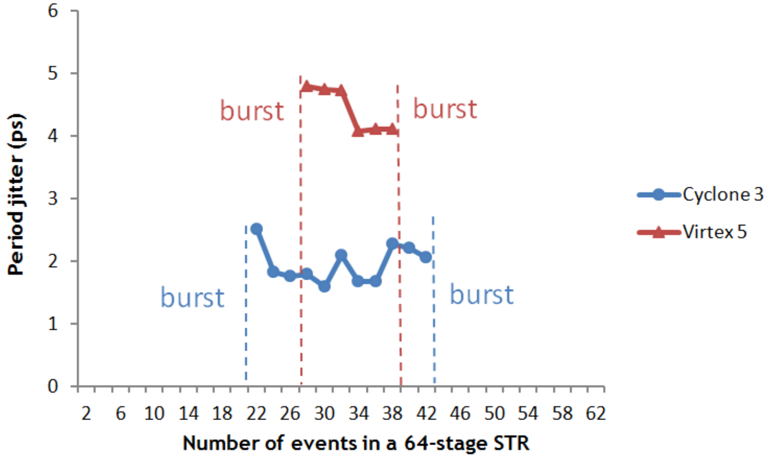
➡ **Jitter does not propagate from one stage to another**

- Measurements in Altera and Xilinx devices



**Period histogram of a 96-stage, 48 events STR in Altera Cyclone 3**





# STRs Built-in Phase Control

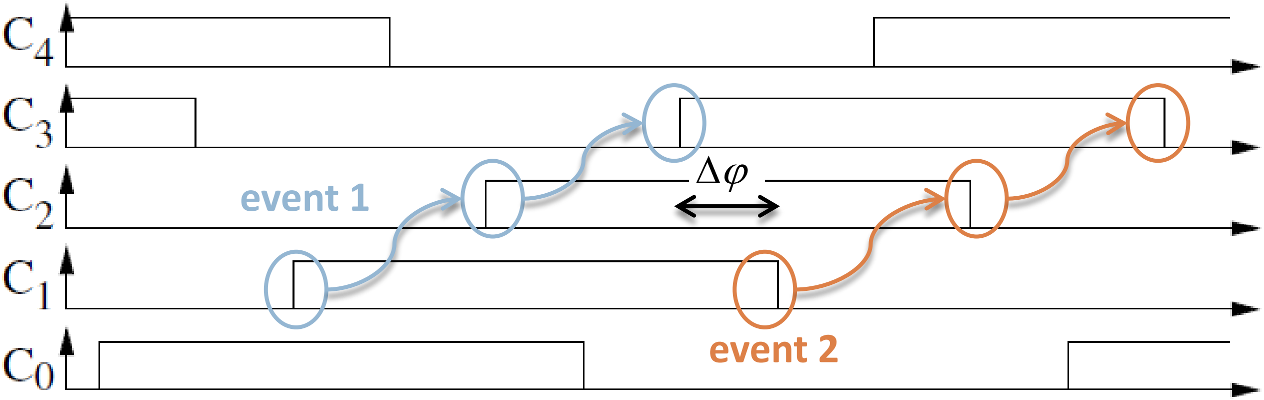
- *N* events confined in an *L*-stage STR spread around the ring
  - Phase shift between two stages separated by *n* stages

$$\varphi_n = n \times \frac{N}{L} \times 180^o$$

  - *L* and *N* **co-prime** -> ***L* equi-distant phases**

$$\Delta\varphi = \frac{T}{2L}$$

➡ **The phase resolution can be set as short as needed**

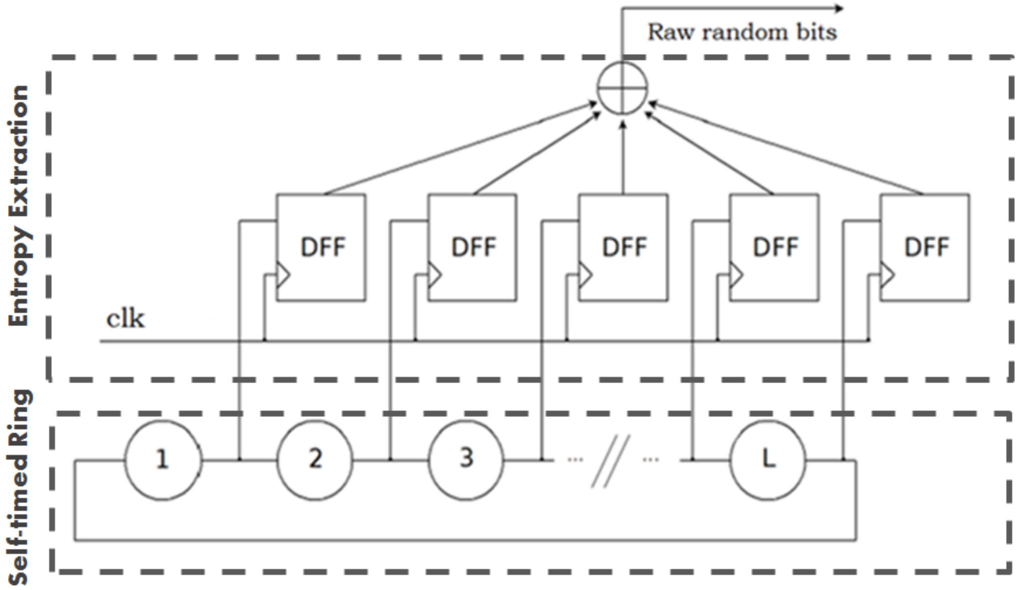


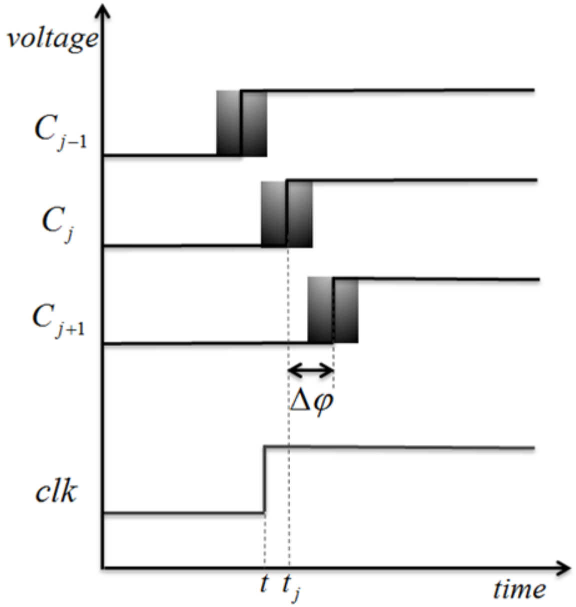
**Propagation of 2 events in a 5-stage STR**

---

[3] S. Fairbanks and Simon Moore « *Analog micropipeline rings for high precision timing* »
[4] O.Ellisati *et. al.* « *A novel high-speed multiphase oscillator using self-timed rings* »

# Summary





- Timings between the events in the STR are **auto-controlled**

- The phase resolution of the STR can be set **as short as needed**

# Outline

- **Combined Ring Oscillators TRNG**

- **A Novel TRNG Architecture Using Self-timed Rings**
  - **TRNG Architecture and Principle**
  - **Self-timed Ring Architecture and Behavior**

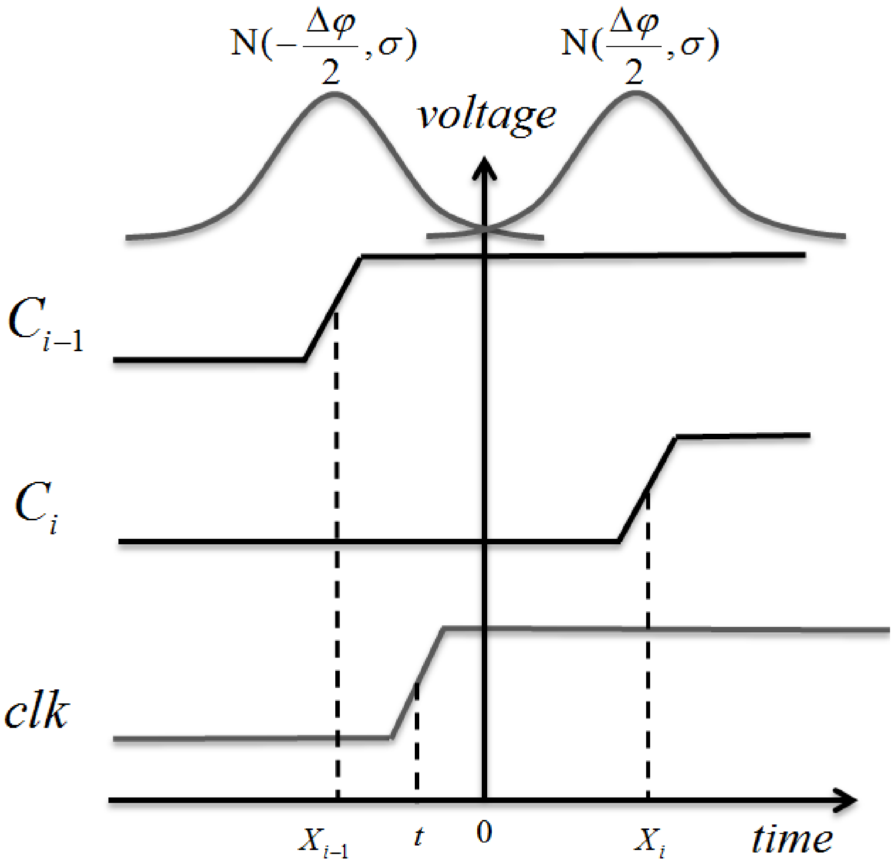
- **Stochastic Model for Bias and Entropy Estimators**
  - **Maximal Bias and Minimal Entropy per Bit**
  - **Bias Correction**

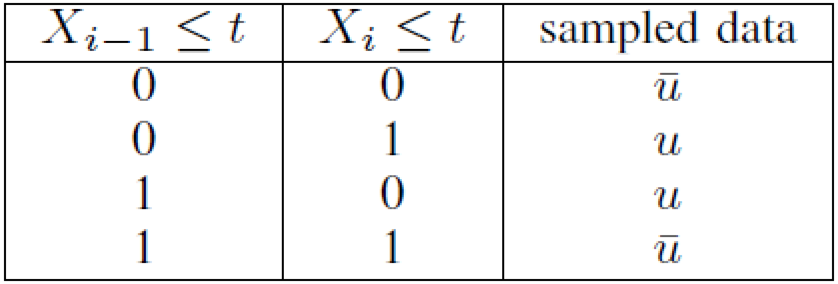
- **Experimental Results**
  - **Implementation and Adequation to the Model**
  - **Period and Jitter Measurements**
  - **Statistical Evaluation**

- **Conclusion / Future Works**

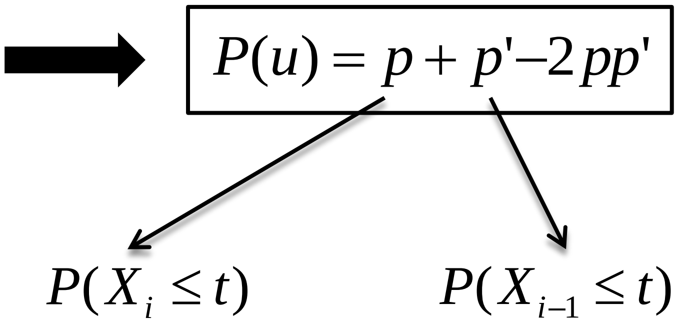
# Stochastic Model

**Objective** Compute the probability to sample a ‘0’ or a ‘1’ with respect to the sampling moment *t*





| $X_{i-1} \le t$ | $X_i \le t$ | sampled data |
|---|---|---|
| 0 | 0 | $\bar{u}$ |
| 0 | 1 | $u$ |
| 1 | 0 | $u$ |
| 1 | 1 | $\bar{u}$ |



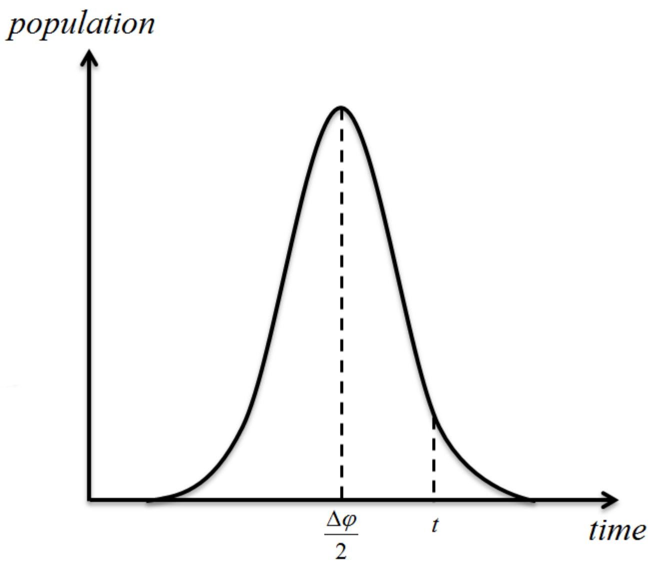
$$\boxed{P(u) = p + p' - 2pp'}$$

$P(X_i \le t)$ $P(X_{i-1} \le t)$

# Computing probabilities

- We use the cumulative density function of the normal distribution

$$\Phi(x) = \int_{-\infty}^{x} \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}} dt \quad , x \in R$$



➡ $$p = P(X_i \le t) = \Phi\left(\frac{t - \frac{\overset{= T/2L}{\Delta\varphi}}{2}}{\sigma}\right)$$

$$\boxed{P(u) = \Phi\left(\frac{t - \frac{T}{4L}}{\sigma}\right) + \Phi\left(\frac{t + \frac{T}{4L}}{\sigma}\right) - 2\Phi\left(\frac{t - \frac{T}{4L}}{\sigma}\right) \times \Phi\left(\frac{t + \frac{T}{4L}}{\sigma}\right)}$$

# Bias and Entropy Per Bit

**Definition** Entropy and absolute bias of a raw random bit at the TRNG output

$$H = -P(u)\log_2(P(u)) - (1-P(u))\log_2(1-P(u))$$

$$|B| = \left|\frac{1}{2} - P(u)\right| \longrightarrow \textit{function of } t, T, L \textit{ and } \sigma$$

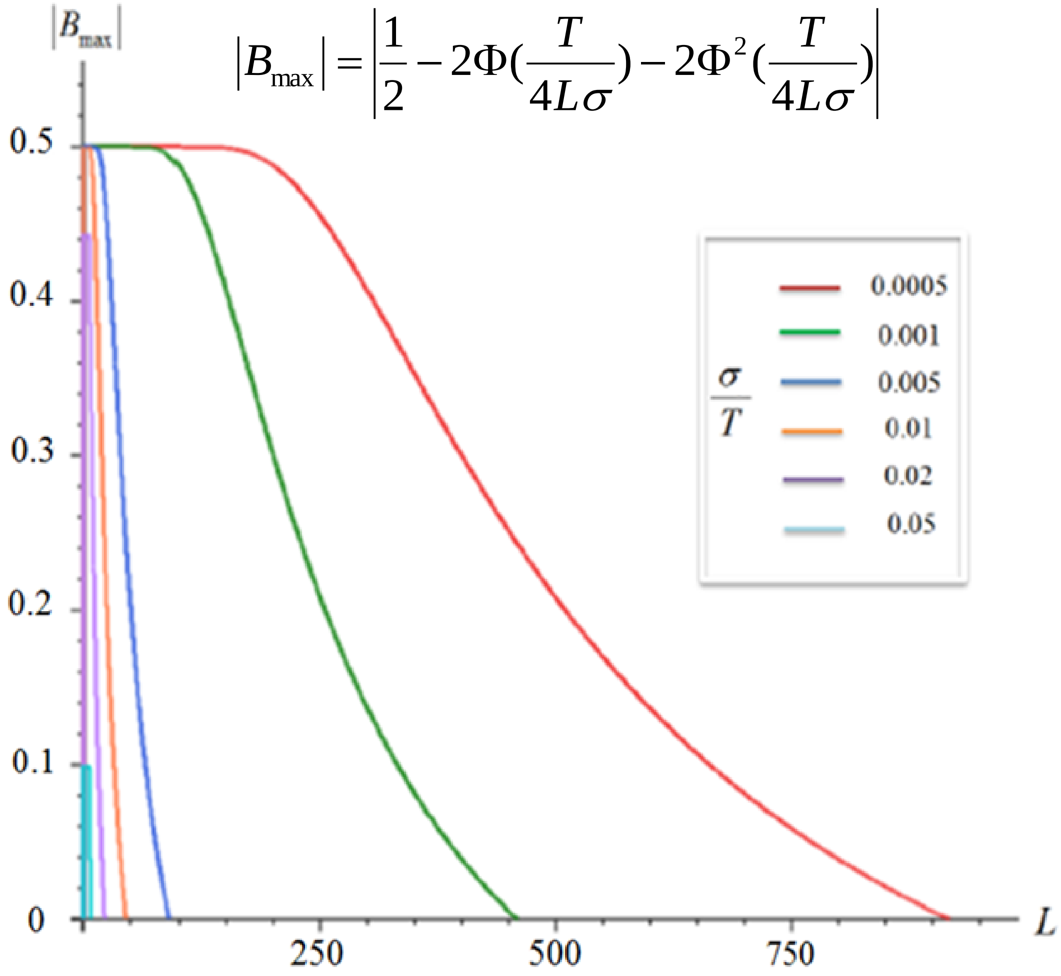
- Minimal bias and maximal entropy are obtained when *t=0*

➡ $$\boxed{H_{\min} = -P_{t=0}(u)\log_2(P_{t=0}(u)) - (1-P_{t=0}(u))\log_2(1-P_{t=0}(u))}$$

with $$P_{t=0}(u) = 1 - 2\Phi(\frac{T}{4L\sigma}) + 2\Phi^2(\frac{T}{4L\sigma})$$

➡ $$\boxed{|B_{\max}| = \left|\frac{1}{2} - 2\Phi(\frac{T}{4L\sigma}) - 2\Phi^2(\frac{T}{4L\sigma})\right|}$$

# Design Methodology



$$|B_{\max}| = \left|\frac{1}{2} - 2\Phi\left(\frac{T}{4L\sigma}\right) - 2\Phi^2\left(\frac{T}{4L\sigma}\right)\right|$$

**METHODOLOGY**

First, measure the jitter magnitude and the oscillation period, then:

**Strategy 1**

Select *L* to achieve at worst 0,01 maximal absolute bias per bit (~ 0,99 minimal entropy per bit)

**Maximize size and speed**

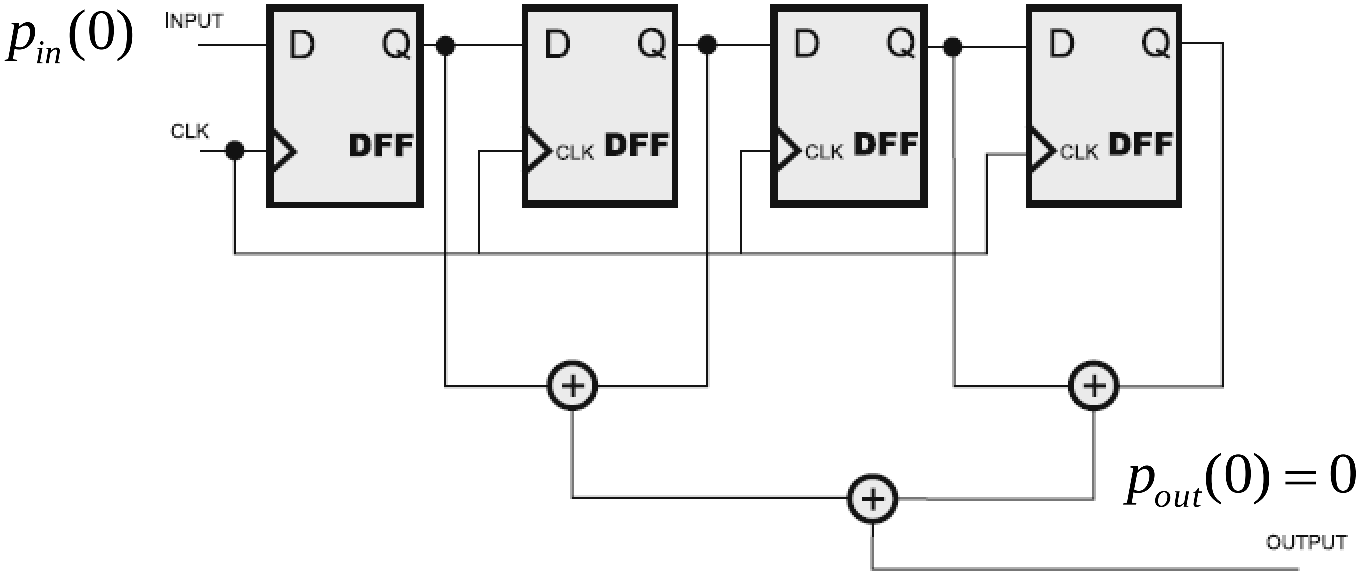
**Strategy 2**

*L* is not sufficient to achieve the required entropy per bit, successive bits are compressed using a parity filter to enhance the entropy per bit

**Minimize size and speed**

# Bias Correction

**Condition** Successive sampled bits are independent



$$p_{out}(0) = 0.5 + \overbrace{2^{n-1}(p_{in}(0) - 0.5)^n}^{\xrightarrow{n \to \infty} 0}$$

**Parity filter structure**

**Method** Compute *n* the filter order to obtain a 0.99 minimal entropy at the filter output

**Drawback** Throuput is divided by *n*

# Outline

- **Combined Ring Oscillators TRNG**

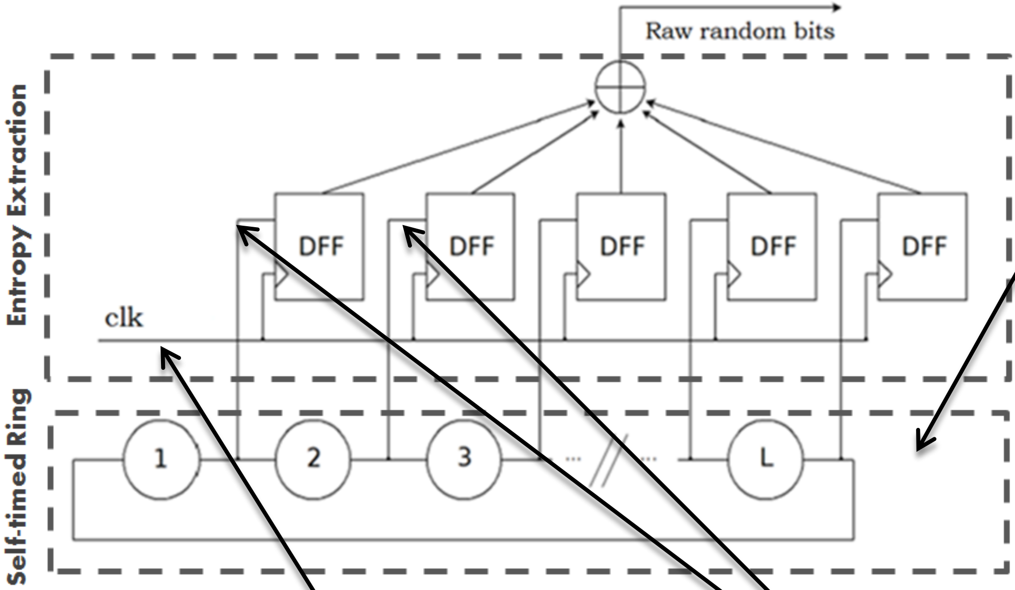
- **A Novel TRNG Architecture Using Self-timed Rings**
  - **TRNG Architecture and Principle**
  - **Self-timed Ring Architecture and Behavior**

- **Stochastic Model for Bias and Entropy Estimators**
  - **Maximal Bias and Minimal Entropy per Bit**
  - **Bias Correction**

- **Experimental Results**
  - **Implementation and Adequation to the Model**
  - **Period and Jitter Measurements**
  - **Statistical Evaluation**

- **Conclusion / Future Works**

# Implementation and Adequacy to the Model



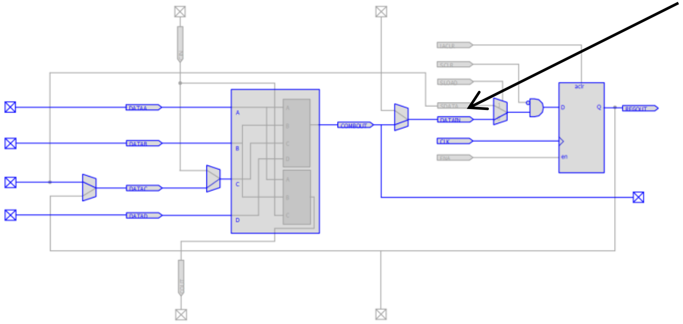
**STR implementation**

- One 4-input LUT per ring stage (available in most recent FPGAs)

$$\frac{D_{ff}}{D_{rr}} \approx 1$$

$$L = 2N - 1 \quad , \quad \frac{N}{L-N} = \frac{N}{N-1} \approx 1$$

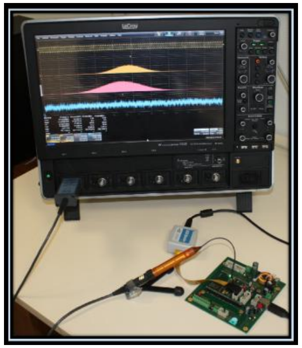
**16 MHz QUARTZ (low jitter) to evaluate the entropy of the STR**



Hardwired connection in Altera LUTs (available in most FPGAs)

# Preliminary Measurements

- Experimental setup
  - Wideband digital oscilloscope (3.5 GHz bandwidth and 40 Gsample/s) + Lecroy statistical tools
  - Low Voltage Differential Signaling (LVDS)
  - Altera Cyclone 3 FPGA



| **Number of ring stages** | **Number of events** | **Oscillation period** | **Phase resolution** | **Jitter magnitude** |
|---|---|---|---|---|
| **63** | **32** | 2.07 *ns* | 16.4 *ps* | 2.1 *ps* |
| **127** | **64** | 2.07 *ns* | 8.2 *ps* | 1.7 *ps* |
| **255** | **128** | 2.08 *ns* | 4.0 *ps* | 1.7 *ps* |
| **511** | **256** | 2.46 *ns* | 2.4 *ps* | 1.9 *ps* |
| **1023** | **512** | 2.63 *ns* | 1.3 *ps* | 1.8 *ps* |

➡ $\sigma \approx 2\,ps$

# Minimal Entropy and Maximal Bias Per Bit

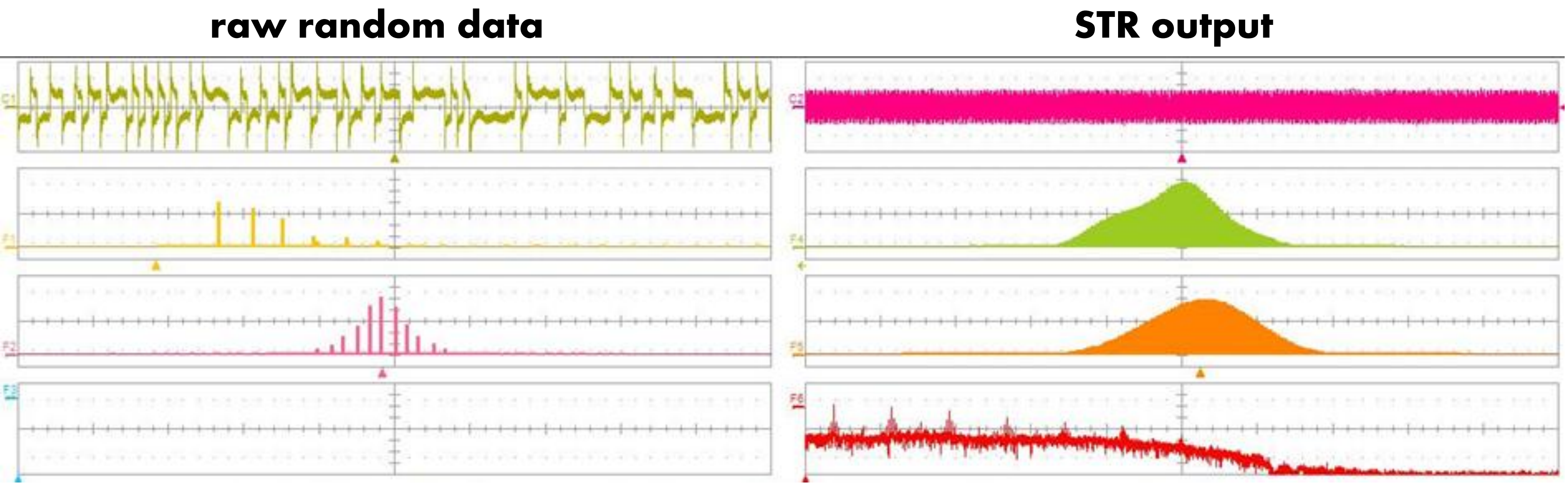
| Number of ring stages | Number of events | Maximal absolute bias per bit | Minimal entropy per bit | Minimal parity filter order to achieve 0.99 entropy |
|---|---|---|---|---|
| **63** | **32** | ~ 0.5 | ~ 0 | - |
| **127** | **64** | 0.46 | 0.26 | 38 |
| **255** | **128** | 0.42 | 0.73 | 8 |
| **511** | **256** | *0.12* | 0.96 | 3 |
| **1023** | **512** | *0.01* | 0.99 | 0 |

- Having a sufficient entropy without compression can be expensive

➡ **Seek a trade-off between size and speed**

- Two interesting configurations: ***L=255*** and ***L=511***

# Evaluation (1)



**127-stage STR with 64 events sampled at 16 Mhz**

- Statistical evaluation
  - ~ 100 MB of data using NIST SP 800-22
  - 1000 sequences of 20000 bits using FIPS 140-2 tests
  - ~ 10 MB of data using AIS-31 T0-T5 tests
- When used, the parity filter order is 8

# Evaluation (2)

| Number of ring stages | Number of events | Minimal entropy per bit | FIPS 140-1 | AIS 31 (T0-T5) | NIST SP 800-22 |
|---|---|---|---|---|---|
| 63 | 32 | ~ 0 | 55 % | FAIL | FAIL |
| 127 | 64 | 0.26 | 98 % | PASS | FAIL |
| 255 | 128 | 0.73 | 100 % | PASS | FAIL |
| 511 | 256 | 0.96 | 100 % | PASS | PASS |

Raw random data at 16 Mbits/s

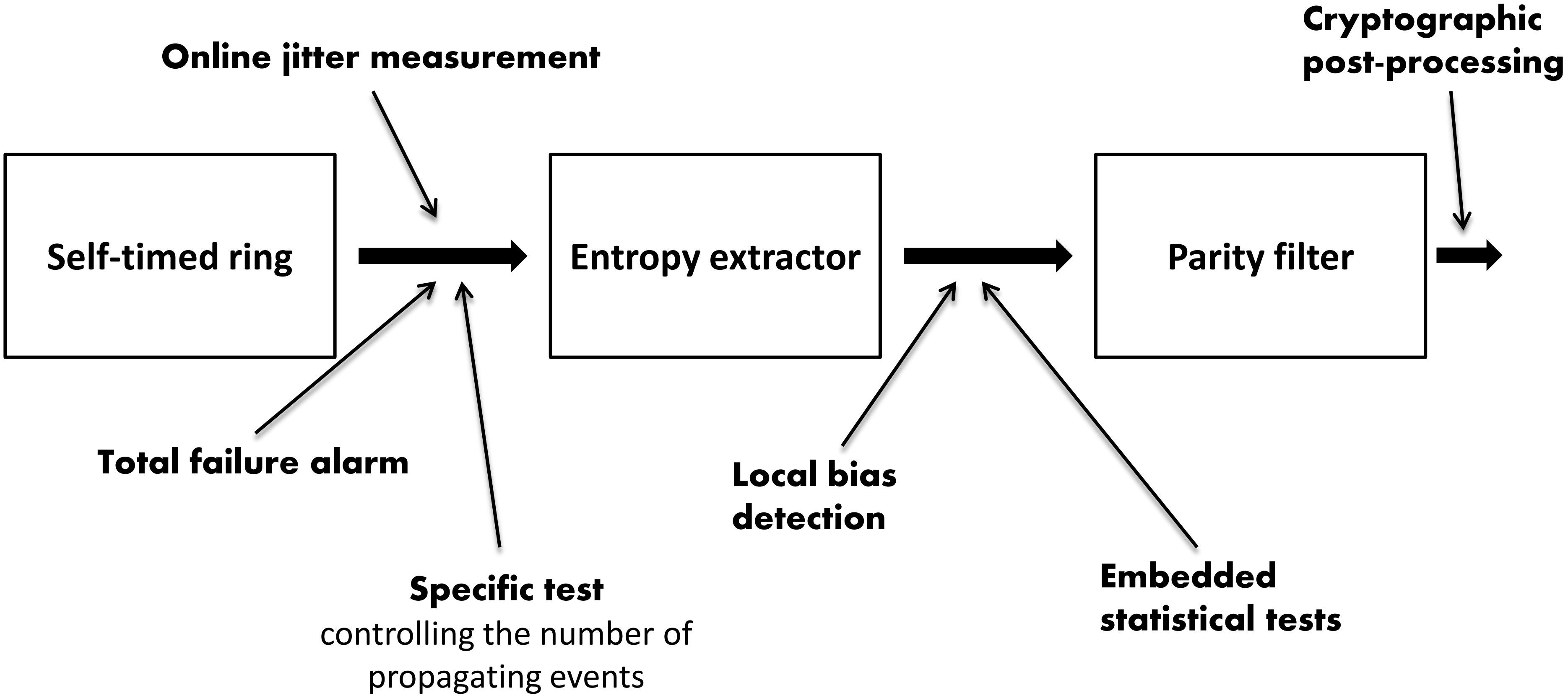
| Number of ring stages | Number of events | Minimal entropy per bit | FIPS 140-1 | AIS 31 (T0-T5) | NIST SP 800-22 |
|---|---|---|---|---|---|
| 63 | 32 | - | 100 % | PASS | FAIL |
| 127 | 64 | 0.76 | 100 % | PASS | PASS |
| 255 | 128 | 0.99 | 100 % | PASS | PASS |
| 511 | 256 | > 0.99 | 100 % | PASS | PASS |

Compressed data at 2 Mbits/s

# Conclusion

- A novel TRNG design using the jitter of propagating events in a STR
  - Based on a simple and modelable principle
  - A stochastic model for entropy and bias estimators
  - Passes statistical tests with a high throughput
- A **scalable architecture** that can be adapted to measured technological paremeters and security requirements
- AIS31 compliant ?
  - Effective online tests to achieve PTG. 2
  - Online tests + Cryptographic postprocessing for PTG. 3

# Future Works



**Online jitter measurement**

**Cryptographic post-processing**

Self-timed ring

Entropy extractor

Parity filter

**Total failure alarm**

**Local bias detection**

**Specific test** controlling the number of propagating events

**Embedded statistical tests**

Thank you !