True Random Number Generation in Reconfigurable Devices

Viktor FISCHER

Laboratoire Hubert Curien, UMR 5516 CNRS
Jean Monnet University, Member of University of Lyon
Saint-Etienne, France

fischer@univ-st-etienne.fr

December 2010

Introduction – True Random Numbers in Cryptography

- Use of Random Number Generators (RNGs) in cryptography
    - Generation of cryptographic keys with special security requirements
    - Generation of initialization vectors, nonces, padding values, ...
    - Counter-measures against side-channel attacks
- Requirements on RNGs
    - R1: Good statistical properties of the output bitstream
    - R2: Output unpredictability
    - R3: Security
        - Robustness – resistance against attacks
        - Testability of the source of randomness

LABORATOIRE Hubert CURIEN
UMR • CNRS • 5516 • ST-ETIENNE

Basic RNG Classes

- Deterministic (Pseudo-) random number generators (PRNG)
  - Algorithmic generators
  - Usually faster, with good statistical properties
  - Must be computationally secure, i.e. it should be computationally difficult to guess the next values
- Physical (True-) random number generators (TRNG)
  - Using some physical source of randomness
  - Unpredictable, having usually suboptimal statistical characteristics
  - Usually slower
- Hybrid random number generators (HRNG)
  - Deterministic RNG seeded repeatedly by a physical random number generator

TRNGs in Logic Devices

- Logic devices (ASICs or FPGAs)
  - Aimed at implementation of deterministic systems
  - Designed so that the deterministic behavior dominates
  - Some analog blocks are sometimes available (PLL, RC-oscillator, A/D and D/A converters, etc.)
- Employable physical randomness sources
  - Timing/phase instability of the clock signal – clock jitter/phase noise
  - Metastability & S/H time violation
  - Sources with limited usability (needing analog components)
    - Chaos
    - Thermal noise

TRNG Design Strategy

- Classical approach
  - Propose a TRNG principle
    - Simple – occupying small area
    - Giving (if possible) high bit-rate
    - Having small power consumption
  - Evaluate the quality by common statistical tests
    - FIPS 140-2
    - NIST 800-22
    - DIEHARD (DIEHARDER)
- Modern approach (recommendations AIS31 of the German BSI)
  - Additional requirements:
    - Detailed analysis of the origin of random behavior
    - Research of efficient generator-specific embedded tests
    - Proposition of mathematical models – entropy estimators

Motto

It is quite easy to design a "TRNG" that will
let the statistical tests pass...
☺

...but it is much more difficult to know where the "randomness" comes from and how much true randomness is there... [1]
☹

[1] knowing that only the true randomness cannot be guessed or manipulated

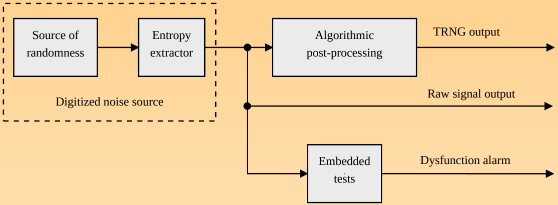
Outline

1. TRNG design
   - Randomness, its extraction and post-processing
   - Stochastic models and entropy estimators
   - TRNG implementation issues
   - Classical and new methodology of TRNG evaluation
2. Case study: RO-TRNG simulation and testing
   - RO-TRNGs of Sunar et al. & Wold and Tan
   - Experimental set-up overview
   - Comparison of TRNGs of Sunar and Wold
   - Impact of the type and size of the jitter
3. Jitter measurements
   - Jitter measurement/TRNG benchmarking requirements
   - Hardware modules for fair TRNG benchmarking
   - Characterization of hardware from jitter measurements
4. On the (in)dependence of rings in FPGAs
   - Experimental set-up
   - Results. . .
5. Conclusions

Outline

1. TRNG design
   - Randomness, its extraction and post-processing
   - Stochastic models and entropy estimators
   - TRNG implementation issues
   - Classical and new methodology of TRNG evaluation
2. Case study: RO-TRNG simulation and testing
   - RO-TRNGs of Sunar et al. & Wold and Tan
   - Experimental set-up overview
   - Comparison of TRNGs of Sunar and Wold
   - Impact of the type and size of the jitter
3. Jitter measurements
   - Jitter measurement/TRNG benchmarking requirements
   - Hardware modules for fair TRNG benchmarking
   - Characterization of hardware from jitter measurements
4. On the (in)dependence of rings in FPGAs
   - Experimental set-up
   - Results. . .
5. Conclusions

TRNGs General Structure



- Source of randomness and entropy extractor
  - Should give as much entropy per bit as possible
  - Should enable sufficient bit-rate
  - Shouldn’t be manipulable (robustness)
- Algorithmic post-processing
  - Enhances statistical properties of the output without reducing the entropy
- Embedded tests
  - Detect immediately the generator’s total failure
  - Evaluate the quality of the source of randomness in real time

Randomness and its Extraction

- Randomness sources
  - **Jitter**: short-term variation of an event from its ideal position
  - **Metastability**: ability of an unstable equilibrium electronic state to persist for an indefinite period in a digital system
  - **Chaos**: stochastic behavior of a deterministic system which exhibits sensitive dependence on initial conditions
  - **Thermal noise**: noise developed in a resistor (or passive component), even in the absence of current flow
- Randomness extraction from a jittery clock – clock sampling in regular intervals using DFFs or latches
  - **State-dependent generators**: their state evolves also without any randomness source, in this case they have a pseudo-random behavior (inner tests will not detect even the total failure)
  - **Stateless generators**: if without source of randomness, their output remain constant (inner test will detect immediately randomness insufficiency)

Post-processing

- Enhance statistical parameters of the TRNG output
  - Reduce bias
  - Increase entropy per bit (while reducing the bit-rate)
- Simple post-processing methods
  - XOR corrector (↗ entropy, ↘ constant bit-rate)
  - Von Neumann corrector (↗ entropy, ↘ variable bit-rate)
  - Linear feed-back shift registers (LFSRs) (→ entropy, → bit-rate)
- Complex post-processing methods
  - Resilient functions based on error correctors (↗ entropy, ↘ constant bit-rate)
  - Hashing, e.g. using SHA1 (↗ entropy, ↘ constant bit-rate)
  - Enciphering of generated data, e.g. using AES and generated key (→ entropy, → bit-rate)

Stochastic Models and Entropy Estimators

- Stochastic models
  - Estimate output statistical parameters (e.g. bias or entropy) depending on
    - Random input variables (source of randomness)
    - Generator principle (randomness extraction)
- Bias of the output bit-stream
  - Probability of ones on the output: $\Pr(X = 1) = 0.5 + \Delta$
  - According to AIS31, the bias (Δ) should be smaller than 0.0173
  - For uncorrelated random variables the bias can be easily reduced using post-processing
- Entropy
  - Definition for "iid" random variables from a finite set Ω:

$$H(X) = -\sum_{x \in \Omega} \Pr(X = x) log_2 \Pr(X = x)$$

  - Gives the uncertainty contained in a unit of information (bit)
  - High entropy level guarantees that the preceding or succeeding values cannot be guessed with a probability different from 0.5
  - Property of random variables and not of observed realizations
  - The entropy per bit of a good TRNG should be close to 1

Entropy-based Model 1/2

- Proposed in [1], specifies the increase of entropy per sample

Conditional entropy

$$H(R_n|R_1,\ldots,R_{n-1}) = -\sum_{r_1,\ldots,r_{n-1}\in\Omega} \Pr(R_1 = r_1,\ldots,R_{n-1} = r_{n-1})\times$$
$$\sum_{r_n\in\Omega} \Pr(R_n = r_n|R_i = r_i \text{ for } i < n) log_2 \Pr(R_n = r_n|R_i = r_i \text{ for } i < n)$$

where $r_1, r_2, \ldots \in \Omega$ ($\Omega = \{0,1\}^k$ for $k \geq 1$) are realizations of random variables $R_1, R_2, \ldots$

- If $R_1, R_2, \ldots$ form a homogeneous Markov chain, the conditional probabilities depend only on the preceding value r_{n-1}, and for sufficiently large n the probability $\Pr(R_{n-1} = r_{n-1})$ has a stationary distribution

[1] W. Schindler, W. Killmann, Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications, CHES2002.

Entropy-based Model 2/2

- Important consequences
  - Post-processing without data compression (such as LFSR and bijective transforms) does not increase the entropy – it only transforms weaknesses to others and masks them
  - For this reason, only unprocessed binary raw signal should be used to estimate the entropy of the randomness source
  - If the entropy per bit is not sufficient, only the bit-rate reduction & some data compression can increase it

Example

We suppose that a binary raw signal is biased, but the subsequent bits are independent. Inputing this bit-stream into the LFSR that is clocked with the same frequency as the sampler, will remove the bias, but the bits obtained at the output will not be independent any more.

TRNG Implementation Issues

- Resource usage (type and quantity)
- Frequency (bit-rate)
- Power consumption
- Feasibility in selected technology (available logic and routing resources)
- Design automation
  - Manual intervention (P/R) is needed for each device individually
  - Manual intervention is needed for each device package and/or family
  - Completely automated – no manual intervention is needed

Evaluation of the TRNG Using General Statistical Tests

- Classical approach: various general-purpose statistical tests are applied on the generator output
- FIPS140-2 tests [1]
  - 4 tests (Monobit, Poker, Runs, Long runs) applied on bit-streams of 20000 bits
  - These tests are not included in the last version of the standard
- NIST 800-22 tests [2]
  - 15 statistical tests with given testing strategy
  - About 1 Gbit of random data is necessary
- DIEHARD tests [3]
  - 15 statistical tests with the testing strategy similar to NIST tests
  - At least 80 millions bits are necessary to realize the tests

[1] Federal Information Processing Standard FIPS140-2: Security Requirements for Cryptographic Modules

[2] A. Rukhin et al., A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST special publication 800-22, 2001

[3] G. Marsaglia, DIEHARD: Battery of tests of randomness, 1996

AIS31 Evaluation Methodology Adapted for Physical RNG 1/4

- New approach: testing TRNG output AND randomness source
- Security requirements depend on the strength of the security mechanism
- The AIS31 TRNG evaluation methodology [1], recognizes two functionality classes
  - Class 1 applications
    - Challenge – response protocols
    - Initialization vectors transmitted in clear
    - Seeds for PRNGs class K1 and K2
  - Class 2 applications
    - Symmetric and asymmetric cryptographic keys
    - Padding bits
    - Zero-knowledge proofs
    - Seeds for PRNGs class K3 and K4

[1] W. Killmann and W. Schindler. AIS 31: Functionality classes and evaluation methodology for true (physical) random number generators, 2001

AIS31 Evaluation Methodology Adapted for Physical RNG 2/4

- Eight statistical tests are proposed for using in different phases of the TRNG evaluation
  - Tests applied to generated random numbers (TRNG output)
    - T0 – Disjointness test (subsequent random values should be different), rejection probability for an ideal random source: 10^{-17}
    - T1 – T4 – Four tests from FIPS140-1 with rejection probability limit 10^{-6} (not from FIPS140-2, where rejection limit is 10^{-4})
  - Tests applied to the raw binary signal (some weaknesses are tolerable)
    - T5 – Autocorrelation test
    - T6 – Uniform distribution test
    - T7 – Comparative test for multinomial distribution
    - T8 – Coron's entropy test [1]

[1] J.-S. Coron: On the Security of Random Sources, Gemplus, Technical Report IT02-1998.

AIS31 Evaluation Methodology Adapted for Physical RNG 3/4

- Class 1 requirements
  - Generated random vectors must pass T0 Disjointness test
  - Output random bit-streams have to pass selected statistical tests (e.g. T1 – T4)
  - The raw binary signal should be tested for the total failure of the source of randomness (fast total failure test)
  - When intended for high-end security applications, the statistical properties have to be verified in different operational conditions (temperature, power-supply)
  - On-line test(s) must check the internal random numbers on demand or in regular intervals

AIS31 Evaluation Methodology Adapted for Physical RNG 4/4

- Class 2 requirements = Class 1 requirements, plus:
  - The bias of the digitized noise (raw signal) should be ≤ 0.025
  - Tests T5 to T7 applied on the raw binary signal have to pass
  - The entropy test T8 applied on the raw binary signal must pass, too
  - The post-processing must not reduce the average entropy per bit
  - Special statistical tests have to be applied on each TRNG start (startup test)
  - For high-strength mechanisms, the statistical parameters and namely the entropy of the digitized noise signal must be tested in different operational conditions
  - For high-strength mechanisms, the TRNG must trigger on-line tests itself in regular intervals.

Outline

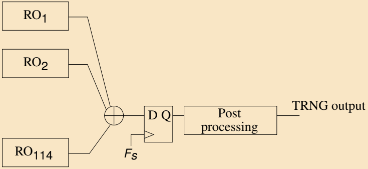
1. TRNG design
   - Randomness, its extraction and post-processing
   - Stochastic models and entropy estimators
   - TRNG implementation issues
   - Classical and new methodology of TRNG evaluation
2. Case study: RO-TRNG simulation and testing
   - RO-TRNGs of Sunar et al. & Wold and Tan
   - Experimental set-up overview
   - Comparison of TRNGs of Sunar and Wold
   - Impact of the type and size of the jitter
3. Jitter measurements
   - Jitter measurement/TRNG benchmarking requirements
   - Hardware modules for fair TRNG benchmarking
   - Characterization of hardware from jitter measurements
4. On the (in)dependence of rings in FPGAs
   - Experimental set-up
   - Results. . .
5. Conclusions

Ring Oscillator-based True Random Number Generators

Principle

- Use the RO-generated clock jitter as a source of randomness,
- In order to increase entropy per bit at the output, more rings are needed.

Sunar *et al.* [Sun2007]



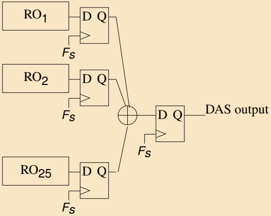
"Provably secure" RNG based on XOR-ing outputs of 114 "independent" ROs.

Sunar's Approach

- Good approach...
  1. Mathematical model (Urn model),
  2. Entropy estimators based on jitter size,
  3. Post-processing using resilient functions.
- But... unrealistic hypotheses (Dichtl & Golic, Wold & Tan, . . .):
  1. Jitter size determined by external measurements,
  2. Too many transitions in the XOR tree,
  3. Set-up and Hold time violation in the D-Flip Flop,
  4. (In)dependence between ROs (coupling).

Improvement of Sunar's Principle

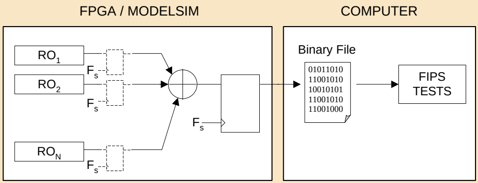
Reconfig'08: Improvement proposed by Wold and Tan



- Closer to the hypotheses of Sunar's "security proof" (XOR tree) ⇒ undeniable improvement !
- Conclusions of Wold and Tan:
  1. 114 ROs are not needed because statistical tests pass for configurations with 50 and even with only 25 ROs,
  2. Post-processing not necessary anymore,
  3. Lower cost and power consumption, because less ROs are used.

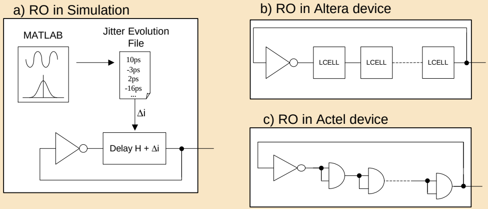
Experimental Set-up

Configuration for TRNG output testing



- VHDL description,
- 20000-bit binary file,
- FIPS 140-2 tests.

Simulation and hardware testing

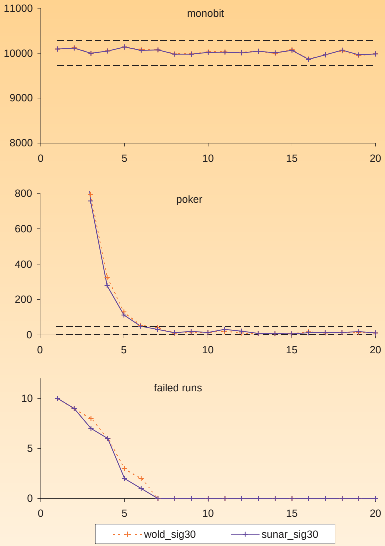


a) Matlab R2008b, Modelsim SE 6.4

b) Altera Cyclone III, Quartus II 9.0

c) Actel Fusion, Libero IDE 8.4

Test1: Sunar versus Wold in VHDL Simulations



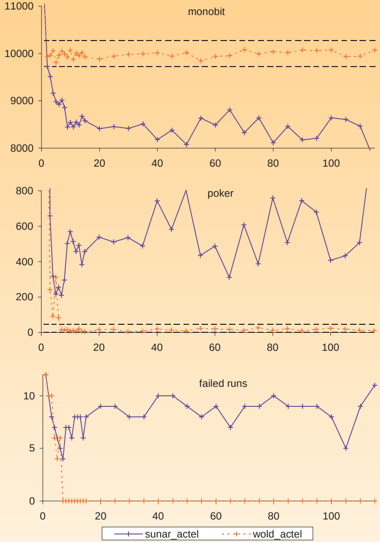
Testing conditions

- Number of ROs varies from 1 to 20,
- Each RO is composed of 9 inverters,
- Half-period: $H + \mathcal{N}(0, 30)$,
- Same random files used for generator of Sunar and Wold.

Results

- Similar (almost identical) results,
- Coherent with the mathematical analysis,
- Starting from 8 ROs, the tests always pass,
- Long run test not presented: it always pass.

Test2: Sunar versus Wold in Actel FPGA



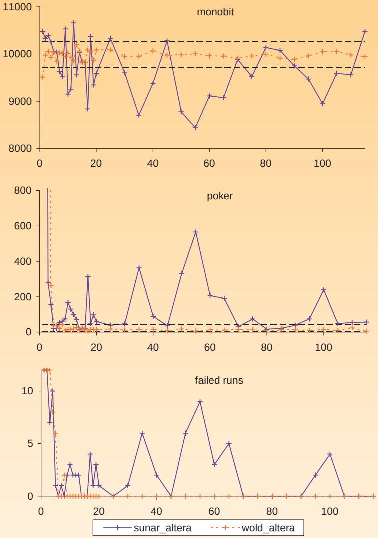
Testing conditions

- Actel Fusion FPGA,
- Number of ROs varies from:
  - 1 to 20 by increments of 1,
  - 20 to 115 by increments of 5.
- Each RO is composed of 9 inverters,
- $F_S = 25$ MHz.

Results

- Tests never (!) pass for Sunar's generator in this technology,
- Tests pass for Wold's generator starting from 8 ROs.

Test3: Sunar versus Wold in Altera FPGA



Testing conditions

- Altera Cyclone III FPGA
- Number of ROs varies from
  - 1 to 20 by increments of 1,
  - 20 to 115 by increments of 5.
- Each RO is composed of 9 inverters implemented in the same LAB,
- $F_S = 25$ MHz.

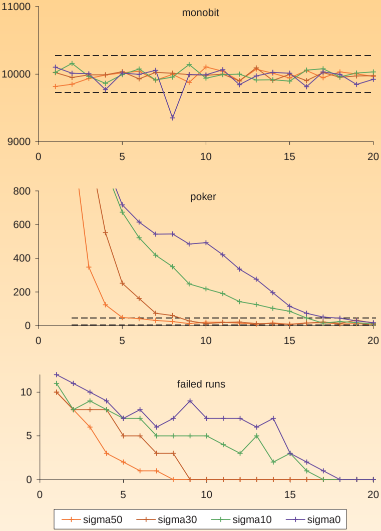
Results

- Tests pass in very few (7/39) configurations for Sunar's TRNG,
- Tests pass for Wold's TRNG starting from 8 ROs.

Conclusion of Comparison

- Claims of Wold and Tan confirmed in simulations and in both technologies (Actel, Altera),
- XOR gate output now fits Sunar's hypothesis,
- What kind of randomness makes the test pass
  - Pseudo-randomness which can be attacked,
  - True-randomness that we are searching for?
- Simulation platform helps to answer these questions:
  - Wold's and Sunar's TRNGs have the same idealized behavior,
  - In the following experiments, only Wold's design is evaluated: closer to the idealized mathematical model.

Test4: Impact of the Local Gaussian Jitter Size



Testing conditions

- Number of ROs varies from 1 to 20,
- Frequencies of RO vary between 197.5 MHz and 202 MHz,
- 9 inverters,
- Gaussian jitter: $\sigma = 0$, 10, 30 and 50 ps.

Results

1. ↗ random jitter ⇒ tests pass more easily. . . as expected,
2. Tests pass from only 18 ROs **without** any jitter . . . surprising!
3. NIST tests pass also from 18 ROs without any randomness . . .

Important Remarks before Going Further

- Wold and Tan: number of ROs reduction, from 114 downto 50 or 25 because tests passed,
- Mathematical problem: urn model of Sunar no more valid,
- Experimental result: tests passed without any randomness . . .

Remark 1

Sunar's original principle (and Wold's improvements too) produce a huge amount of pseudo-randomness that can be predicted (mathematical equation) or manipulated from outside the chip.

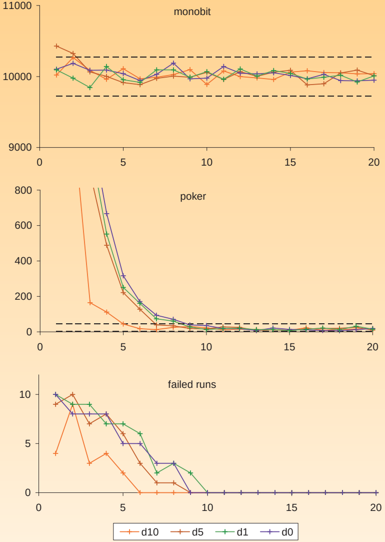
Remark 2

Analysis of statistical tests and derived conclusions of these tests to evaluate TRNG security must be done carefully.

Remark 3

Reducing the number of ROs as proposed by Wold and Tan represents a security-critical attempt for cryptographic applications and should be certainly avoided.

Test5: Influence of a Global Deterministic Jitter Component



Testing conditions

- Fixed Gaussian component $\sigma = 30$ ps,
- Deterministic component: sinusoidal signal
  - Frequency: 3 KHz,
  - Amplitude: 0 to 10 ps.

Results

- $\nearrow$ deterministic part $\Rightarrow$ tests passed more easily

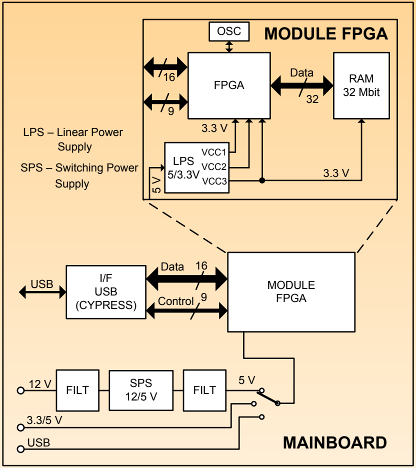
Problems

1. Results strongly dependent on the frequency of injected signal (predictability),
2. Deterministic jitter can be manipulated.

Outline

1. TRNG design
   - Randomness, its extraction and post-processing
   - Stochastic models and entropy estimators
   - TRNG implementation issues
   - Classical and new methodology of TRNG evaluation
2. Case study: RO-TRNG simulation and testing
   - RO-TRNGs of Sunar et al. & Wold and Tan
   - Experimental set-up overview
   - Comparison of TRNGs of Sunar and Wold
   - Impact of the type and size of the jitter
3. Jitter measurements
   - Jitter measurement/TRNG benchmarking requirements
   - Hardware modules for fair TRNG benchmarking
   - Characterization of hardware from jitter measurements
4. On the (in)dependence of rings in FPGAs
   - Experimental set-up
   - Results. . .
5. Conclusions

Fair TRNG Benchmarking in Different FPGA Technologies



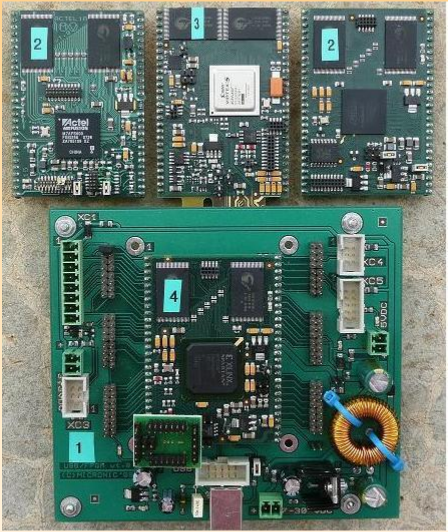
Objectives

- Hardware for fair TRNG benchmarking in various FPGA families,
- Carefully designed power supply,
- Fast random data acquisition,
- High-speed output for precise jitter measurement,
- Interfaced to PC.

Solution

- Mainboard and pluggable modules,
- Mainboard: primary power supplies, USB I/F device (Cypress),
- Module: FPGA device, oscillator, 32-MBit RAM, 2xLVDS outputs.

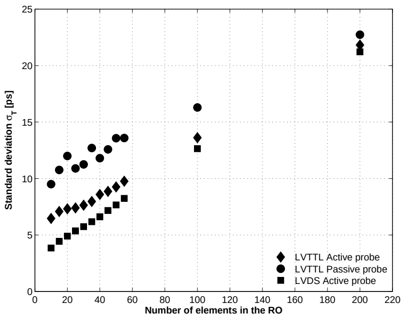
Available Hardware Modules



Five modules available

- Altera FPGA:
  - Cyclone III – EP3C25,
  - Arria II – EP2AGX45,
- Xilinx FPGA:
  - Spartan 3 – XC3S700AN,
  - Virtex 5 – XC5VLX30T,
- Actel FPGA
  - Fusion – M7AFS600.

Precise Jitter Measurement Using LVDS Outputs



Measurement setup

- Oscilloscope LeCroy WavePro 7300,
- Standard passive low-bandwidth probe,
- Differential active 3.5 GHz probe DP320-SP.

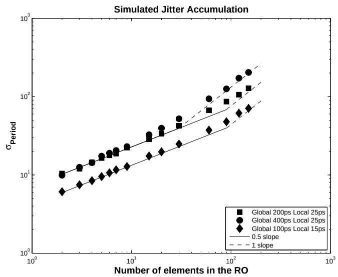
Results and conclusion

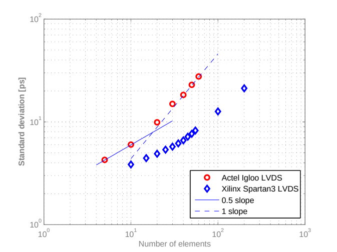
- Comments on results: LVDS outputs together with the differential probe give the smallest jitter,
- Jitter measured by Sunar (standard probe) was probably overestimated.

Question

- Is LVDS & active probe really better?

Characterization of Hardware from Jitter Accumulation





Theory

- The global jitter accumulates linearly (slope 1.0 on a log-log scale), the local jitter with square root (slope 0.5)
- The crossing point of asymptotes permits to characterize the system,
- For smaller global jitter, the asymptote 1.0 moves to the right,
- For bigger Gaussian jitter, the asymptote 0.5 moves up,
- Objective: the crossing point should be placed in North-Est direction as far as possible.

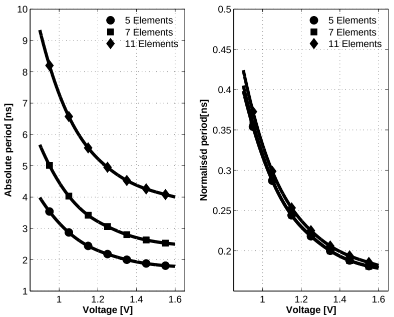
Results

- Actel: more Gaussian noise than Xilinx.

Outline

1. TRNG design
   - Randomness, its extraction and post-processing
   - Stochastic models and entropy estimators
   - TRNG implementation issues
   - Classical and new methodology of TRNG evaluation
2. Case study: RO-TRNG simulation and testing
   - RO-TRNGs of Sunar et al. & Wold and Tan
   - Experimental set-up overview
   - Comparison of TRNGs of Sunar and Wold
   - Impact of the type and size of the jitter
3. Jitter measurements
   - Jitter measurement/TRNG benchmarking requirements
   - Hardware modules for fair TRNG benchmarking
   - Characterization of hardware from jitter measurements
4. On the (in)dependence of rings in FPGAs
   - Experimental set-up
   - Results. . .
5. Conclusions

Frequency Dependence on the Power Supply



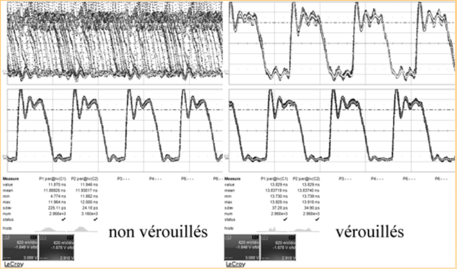
Testing conditions

- Altera Cyclone III Module,
- Nominal voltage: 1.2 V,
- Varying power supply between 0.9 and 1.6 V.

Results

- Frequency highly depends on the power supply,
- The same effect is observable for various frequencies.

Mutual Dependence of Ring Oscillator Frequencies

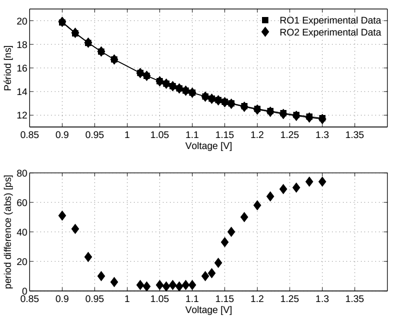


Measure	P1:per@lv(C1)	P2:per@lv(C2)	P3:- - -	P4:- - -	P5:- - -	P6:- - -
value	11.870 ns	11.946 ns				
mean	11.86926 ns	11.93017 ns				
min	4.774 ns	11.862 ns				
max	11.964 ns	12.000 ns				
sdev	225.11 ps	24.18 ps				
num	2.966e+3	3.160e+3				
status	✓	✓				
histo						

non vérouillés

Measure	P1:per@lv(C1)	P2:per@lv(C2)	P3:- - -	P4:- - -	P5:- - -	P6:- - -
value	13.829 ns	13.829 ns				
mean	13.83719 ns	13.83740 ns				
min	13.730 ns	13.739 ns				
max	13.926 ns	13.916 ns				
sdev	37.28 ps	34.90 ps				
num	2.968e+3	2.968e+3				
status	✓	✓				
histo						

vérouillés



Testing conditions

- Two similar ROs are implemented inside the FPGA,
- Frequencies are measured outside the FPGA,
- The power supply is varying between 0.9 and 1.3 V.

Results

- Frequencies are approaching and lock to the same value during a short voltage interval.

Outline

1. TRNG design
   - Randomness, its extraction and post-processing
   - Stochastic models and entropy estimators
   - TRNG implementation issues
   - Classical and new methodology of TRNG evaluation
2. Case study: RO-TRNG simulation and testing
   - RO-TRNGs of Sunar et al. & Wold and Tan
   - Experimental set-up overview
   - Comparison of TRNGs of Sunar and Wold
   - Impact of the type and size of the jitter
3. Jitter measurements
   - Jitter measurement/TRNG benchmarking requirements
   - Hardware modules for fair TRNG benchmarking
   - Characterization of hardware from jitter measurements
4. On the (in)dependence of rings in FPGAs
   - Experimental set-up
   - Results. . .
5. Conclusions

Conclusions

- Statistical tests
  - Necessary BUT far from being sufficient:
    they can pass without any randomness (null entropy),
  - Their results must be carefully analyzed,
  - Conclusion about the RNG security must be derived from the tests even more carefully.
- Entropy
  - It is NOT a property of observed random numbers... but of random variables – it cannot be measured,
  - A mathematical model is needed for evaluating the minimum entropy/bit,
  - Assumptions made in the model must be verified in hardware experiments.
- Independence of randomness sources
  - Must be thoroughly examined,
  - ROs are not only dependent, but they can be locked – RO coupling reduces drastically the entropy.

True Random Number Generation in Reconfigurable Devices

Viktor FISCHER

Laboratoire Hubert Curien, UMR 5516 CNRS
Jean Monnet University, Member of University of Lyon
Saint-Etienne, France

fischer@univ-st-etienne.fr

December 2010