TRNG based on the coherent sampling

Boyan Valtchanov, Viktor Fischer, Alain Aubert

Laboratoire Hubert Curien UMR CNRS 5516, Université Jean Monnet

Outline

- Introduction
- Principle & theoretical aspects
- Simulations
- Experimental results
- Statistical evaluation results
- Conclusions
- Perspectives

Introduction

- Use of TRNG in cryptography:
  - generation of cryptographic keys
  - nonces, IV, padding, masking etc…

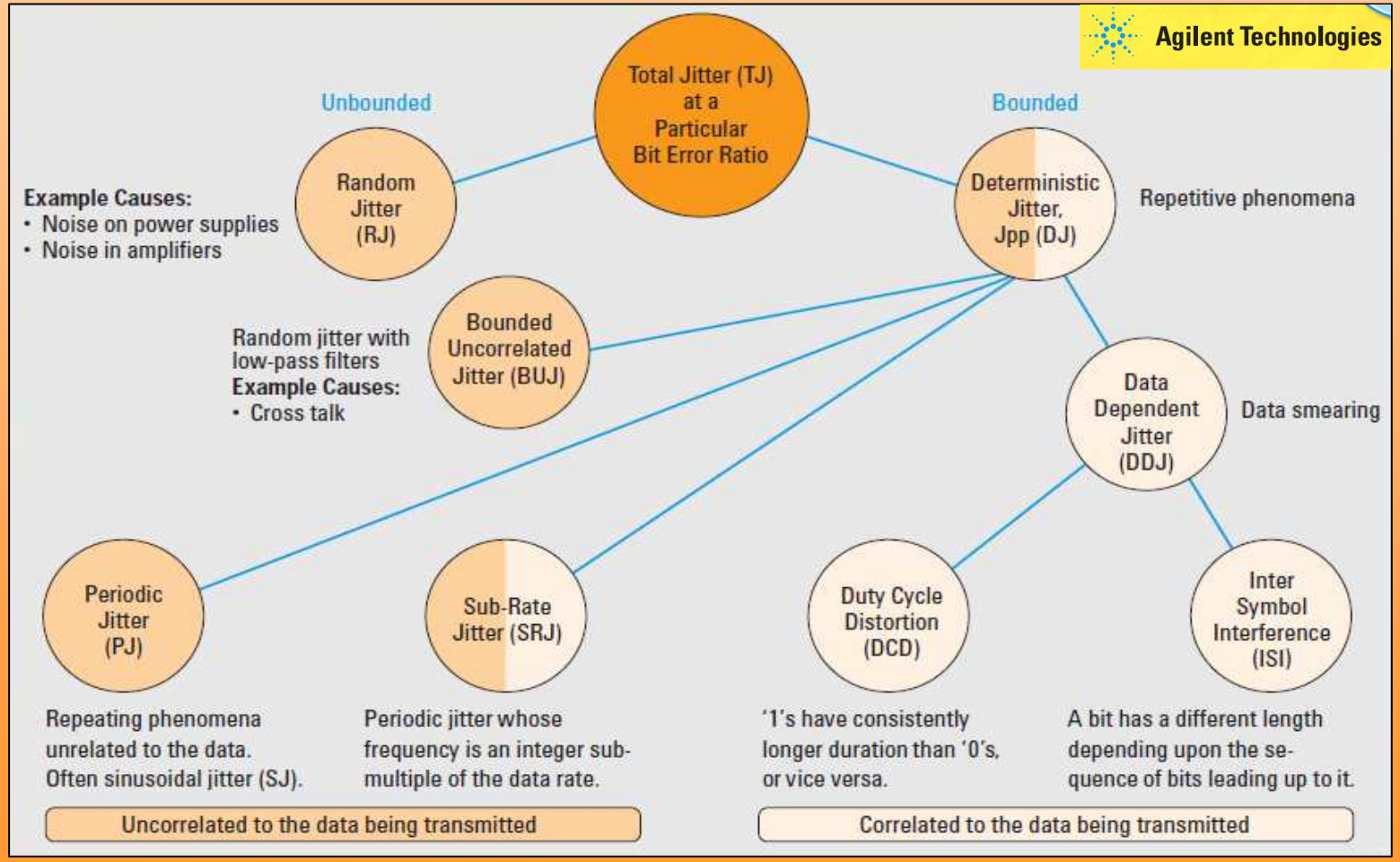
- Required characteristics of TRNG:
  - Physical entropy source (source of randomness)
  - Good statistical distribution of the random stream
  - Availability of the modem
  - Output bit-rate
  - Easy implementation - feasibility
  - Robustness against attacks etc...

Theoretical aspects

- Jitter
- Coherent sampling
- TRNG principle

Theoretical aspects (1/9)

- In communication systems **jitter is a monster** !

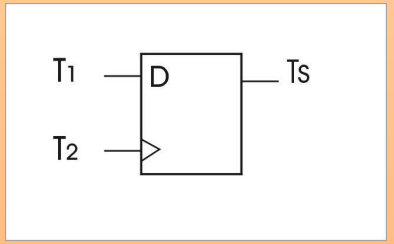


Theoretical aspects (1/9)

- For True Random Number Generation **jitter is ablessing**:
  - Clock jitter is used as source of randomness
  - Two aspects related to jitter exploitation:
    - entropy contents (source of entropy)
    - entropy extraction

Theoretical aspects (2/9)

Coherent sampling



➢ Periodic clock signals with somehow related frequencies

Two cases:

➢ T_1 and T_2 have a integer fractional relationship

They have different frequencies

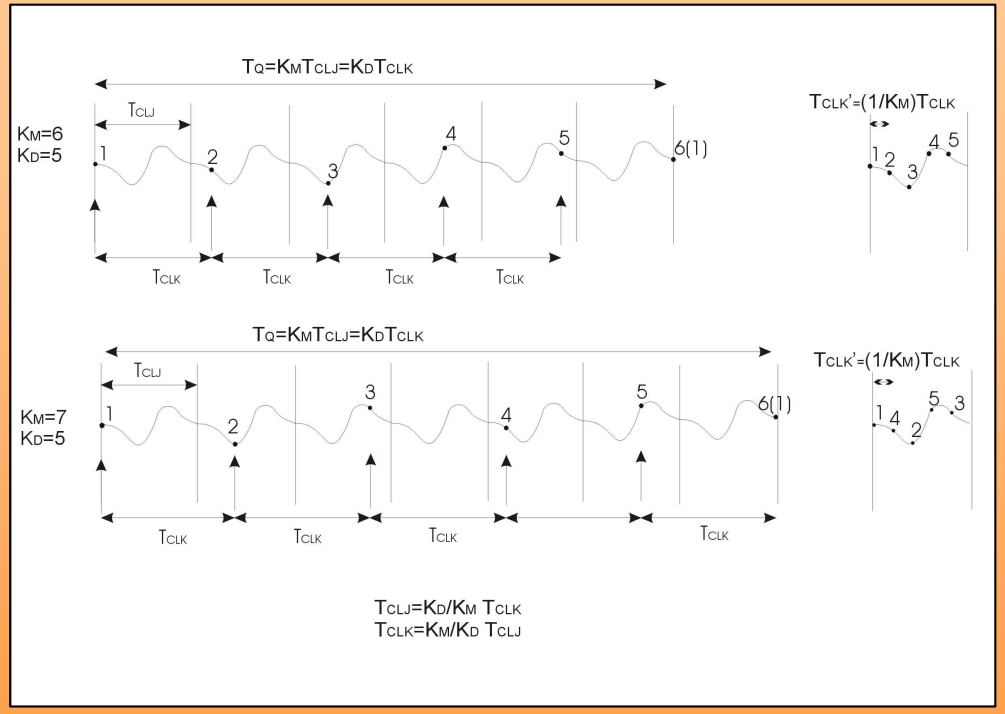
Ref: Fischer & Drutarovsky, CHES 2002

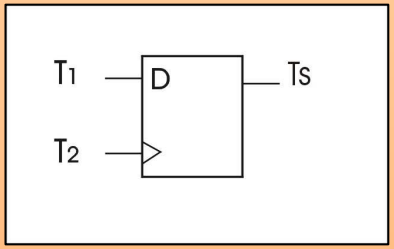
➢ $T_2 = T_1 + \Delta$

They have similar frequencies

Ref: *Kohlebrenner & Gaj, FPGA 2004*

Theoretical aspects (3/9)





$T_1=T_{clj}$, $T_2=T_{clk}$

$$\left(\frac{K_M}{K_D}T_{CLJ}\right)_{\mathrm{mod}(T_{CLJ})} \leq \frac{T_{CLJ}}{K_D}$$

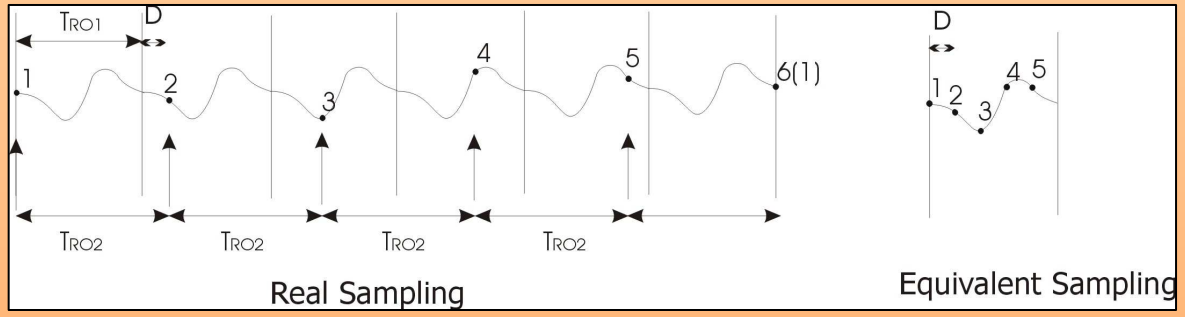
or

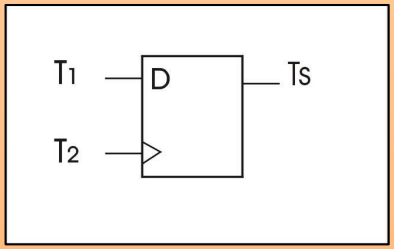
$$\Phi_2 \leq \frac{1}{K_M}T_{CLK}$$

(1)

- Depending on K_m and K_d we can have eighter:
  - **consecutive equivalent sampling** if condition (1)
  - **non consecutive equivalent sampling**

Theoretical aspects (4/8)

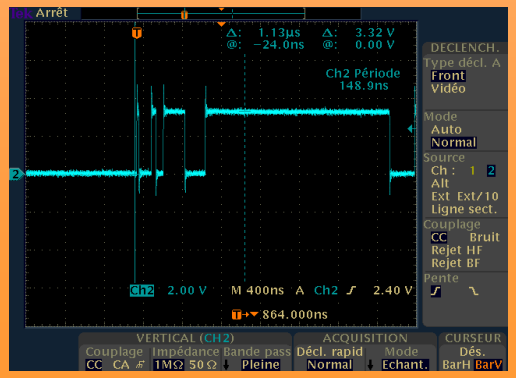




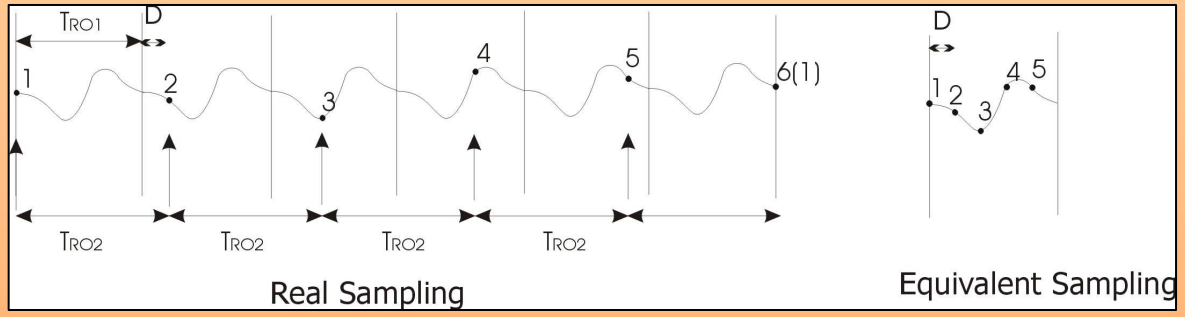
- $T_1=T_{ro1}$, $T_2=T_{ro2,}$, $\Delta=D$
- $T_2=T_1+\Delta$ (Δ small comparing to T_x)

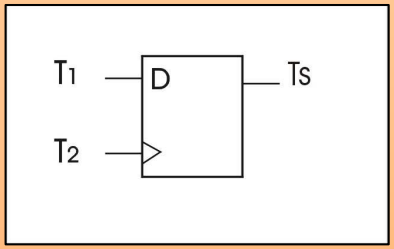
- **Warning:**

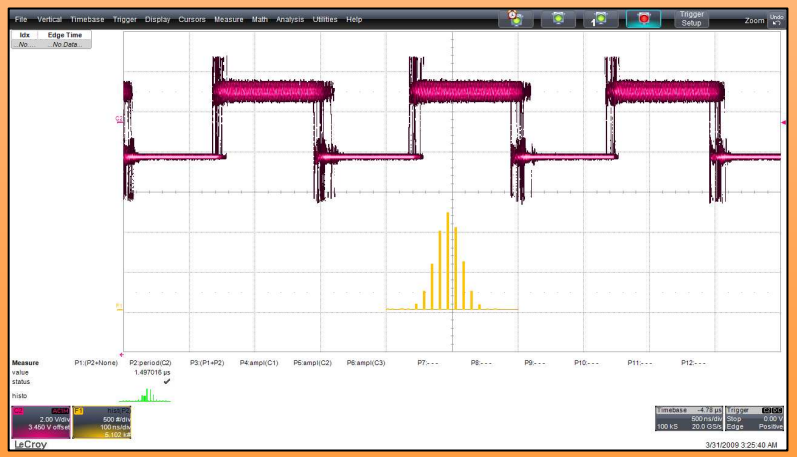
If Δ is too small we can have transients



Theoretical aspects (5/8)





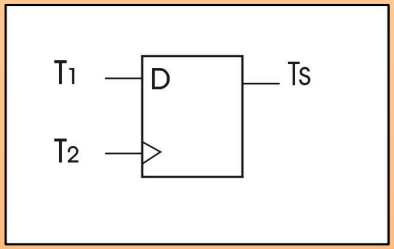


Low frequency **Beat Signal**

- We obtain a low frequency image of T_1 signal – a beat signal
- Due to the presence of jitter, its **lenght is variable** but can be expressed as an integer count of T_2

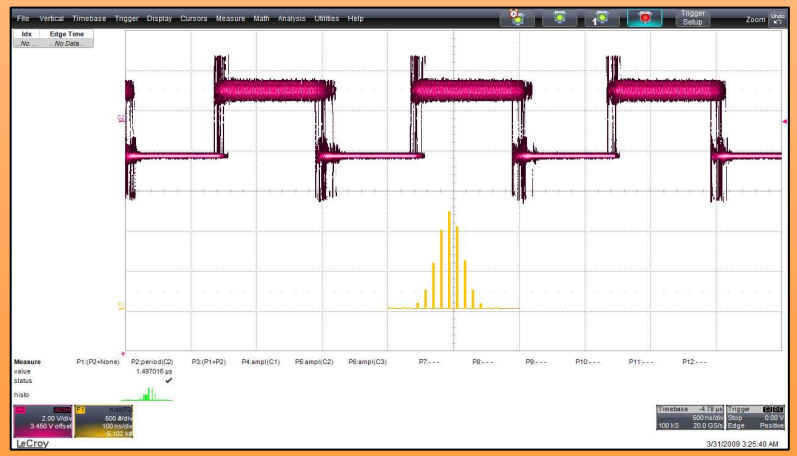
Theoretical aspects (6/9)

- If T_1 and T_2 presents only Gaussian jitter profile, we can compute the intrinsic jitter by observing the beat signal



Injected 1σ RMS jitter [ps]	Measured μ_{Tbeat}	Measured σ_{Tbeat}	Calculated 1σ RMS [ps]
10	85.83	6.37	10.21
9	85.84	5.59	8.96
8	86.02	5.00	8.02
7	85.95	4.53	7.26
6	85.91	3.81	6.11

$$\sigma = \frac{\sigma_{T_{Beat}} \Delta_{Ideal}}{\sqrt{\frac{T_{Ro_{Ideal}}}{\Delta_{Ideal}}} \sqrt{2}}$$

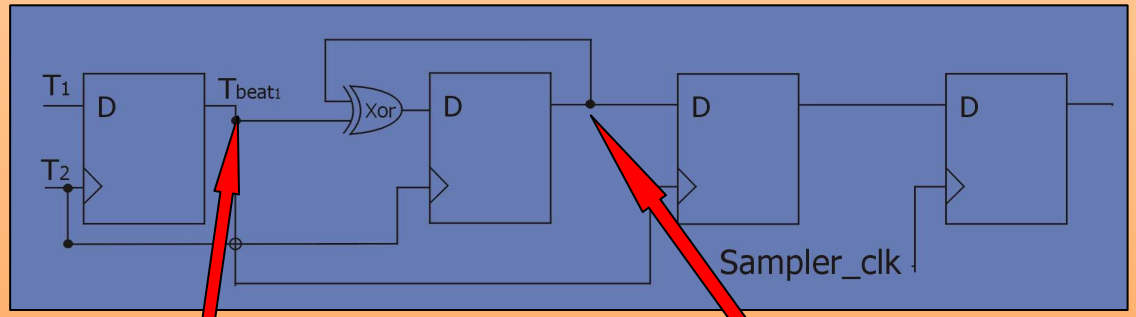


Ref: *A coherent samplingbased method for estimating the jitter used as entropy source for True Random Number Generators, SAMPTA 2009*

Theoretical aspects (7/9)

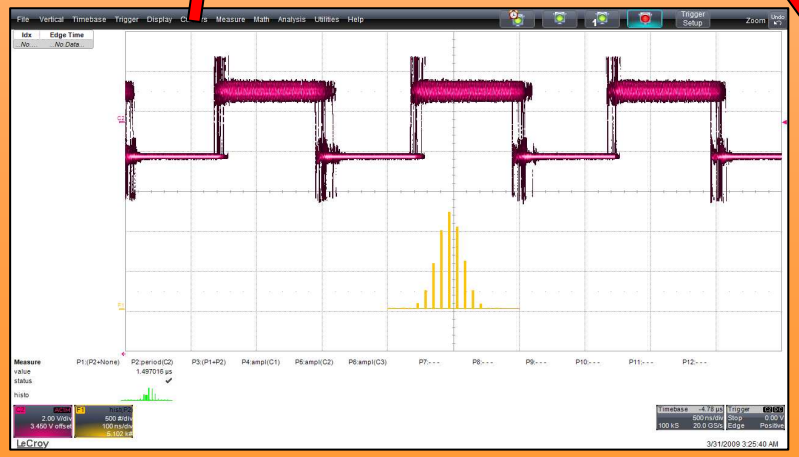
TRNG Principle

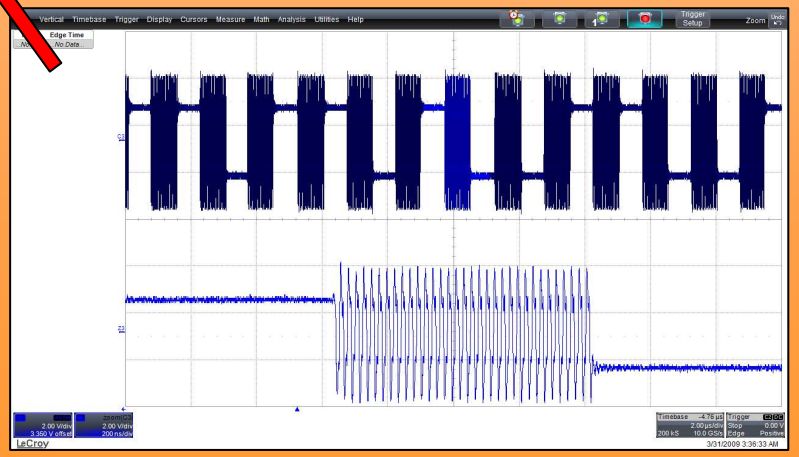
Ref: An embedded TRNG for FPGA Kohlebrenner & Gaj, 2004



$F_{Sampler\_clk} \leq F_{beat}$

simplified version





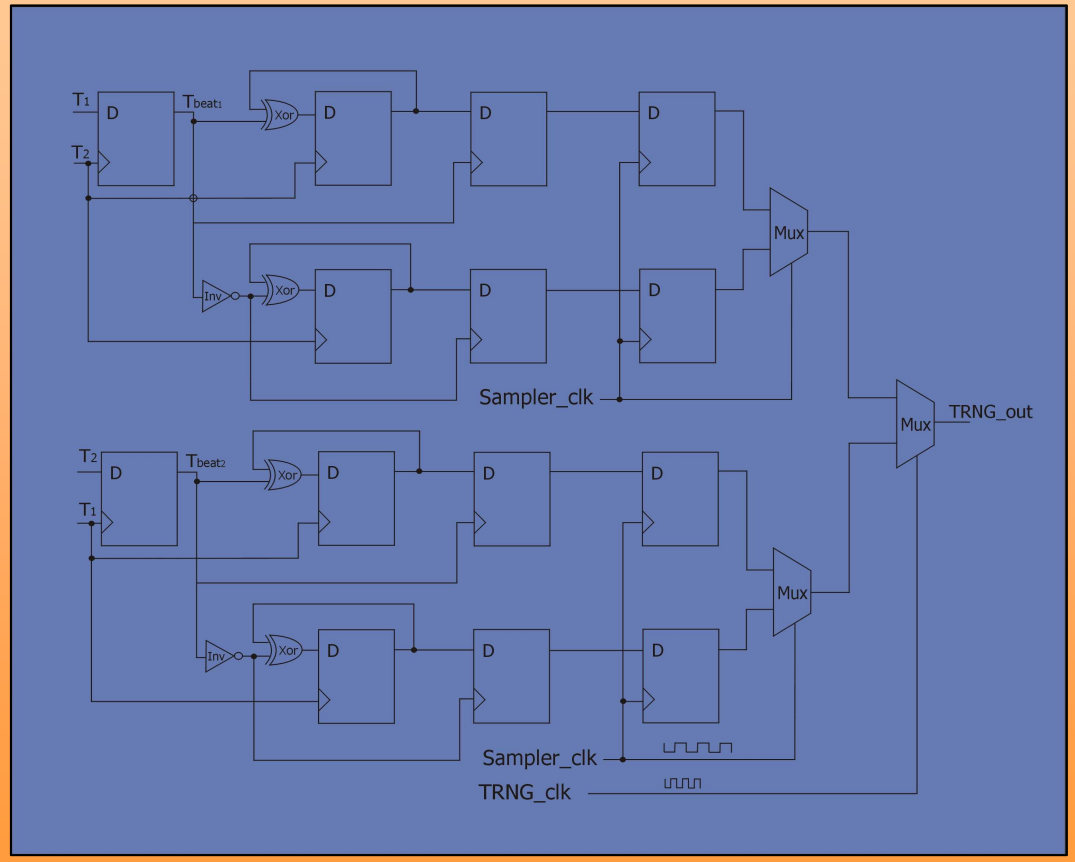
Theoretical aspects (8/9)

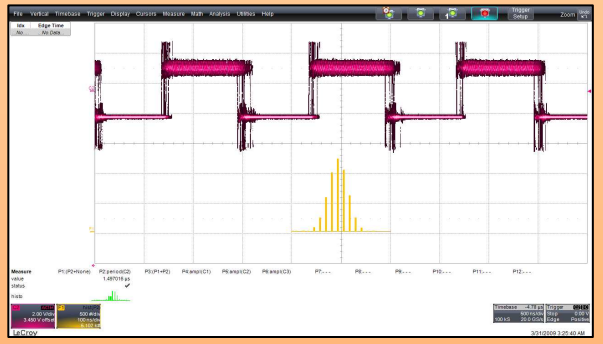
Ref: *An embedded true random number generator for FPGAs*

Kohlebrenner & Gaj, Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays

- **Advantages:**
  - no cost for PLL
- **Disadvantages**
  - device dependent (manual place & route)
  - output bitrate (~500Kbits/sec)
  - bias in the output stream

Theoretical aspects (9/9)

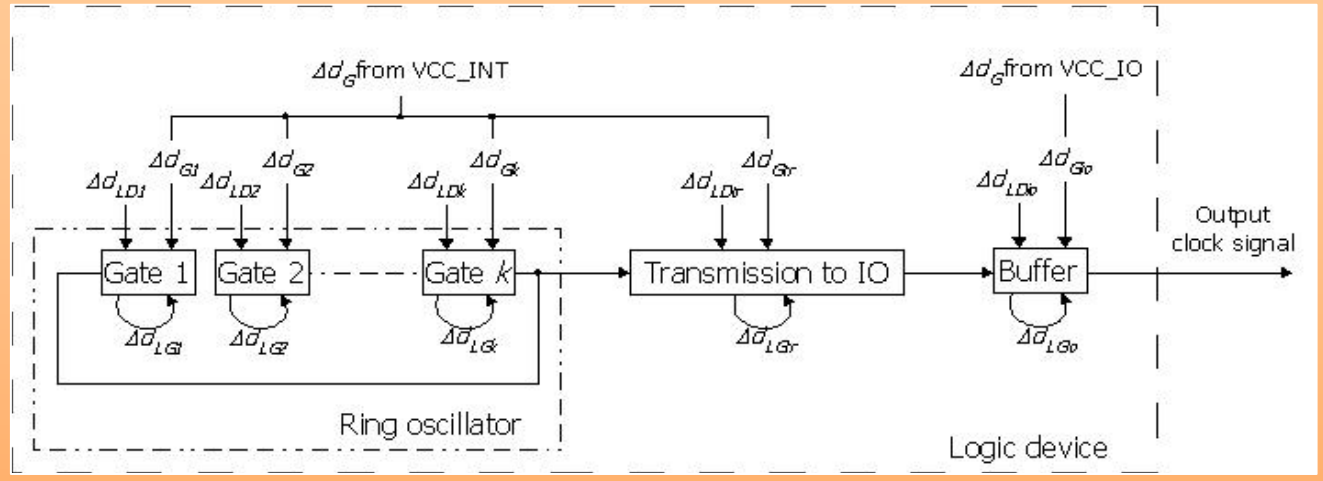




- We propose to extract 4 random bits per T_{beat} period if T_1 and T_2 are independent
- **Mutual sampling**

Simulation Results

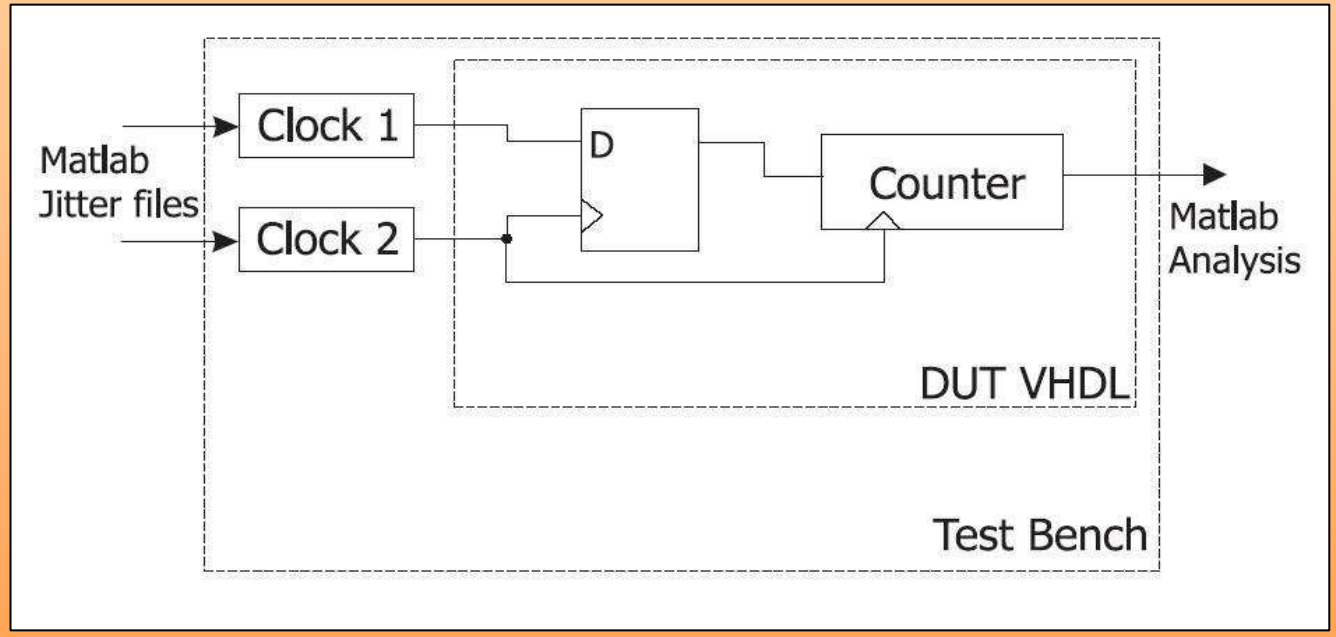
Simulation (1/6)



Ref: *Modeling and observing the jitter in ring oscillators implemented in FPGAs, DDECS'2008*

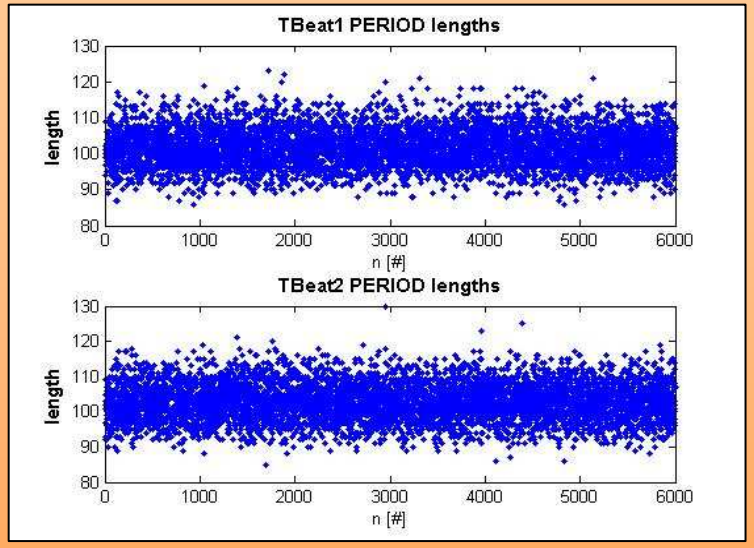
- Each element of the ring oscillatos is subject to **local** and **global** phenomena that affects its timming delay.
- The simulation permits to model the behavior of the entropy extractors for various jitter types (Gaussian & deterministic) and sizes

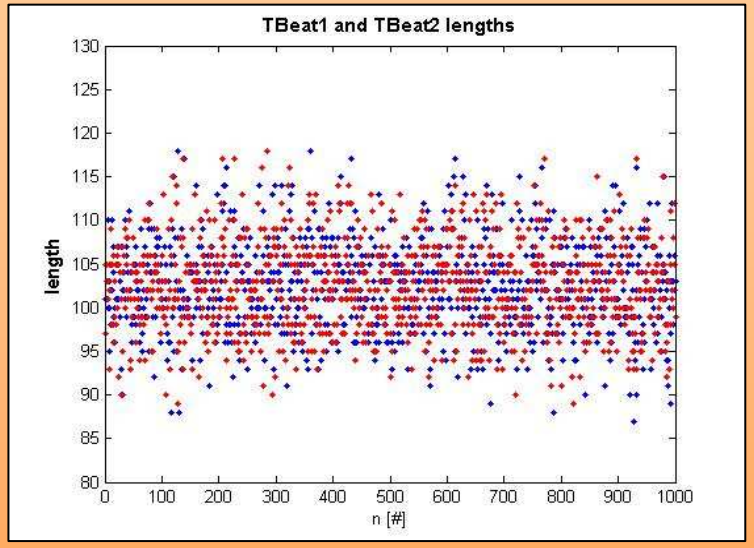
Simulation (2/6)



- We use VHDL to simulate the coherent sampling and the TRNG
- The jitter files are generated in Matlab
- Generated values determines the clock frequency

Simulation (3/6)

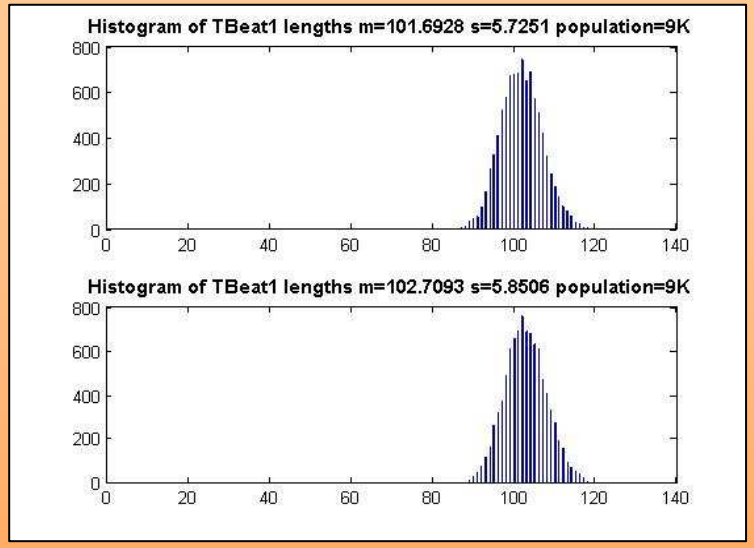


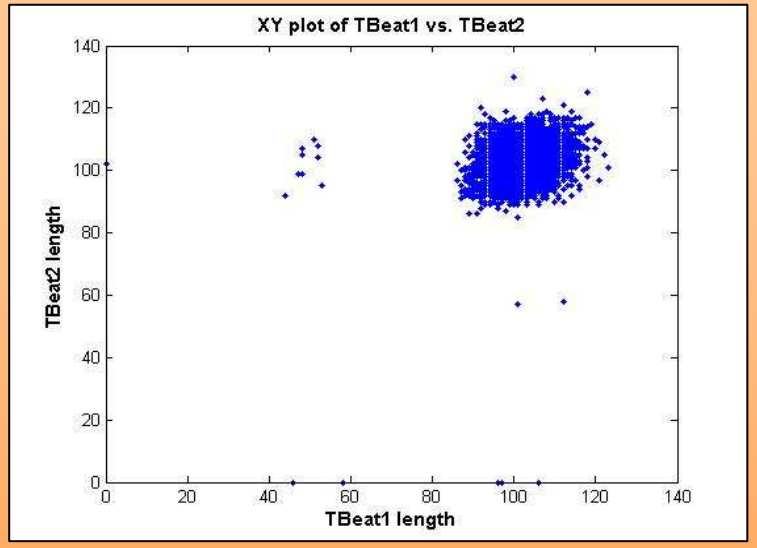


T_1 = 9,000 ns T_2=9,085ns Δ=85ps

- **Case 1:** T_1 and T_2 are independent Gaussian sources
- The obtained beat signals T_{beat1} T_{beat2} are different

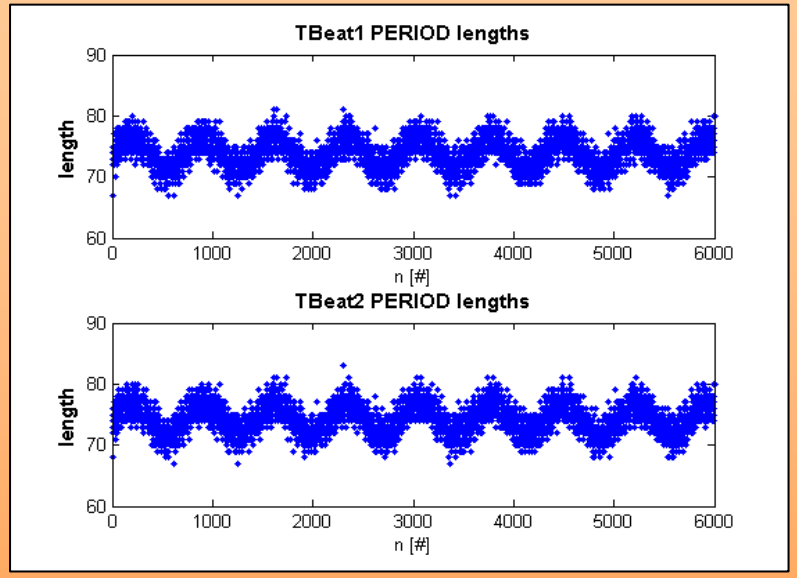
Simulation (4/6)

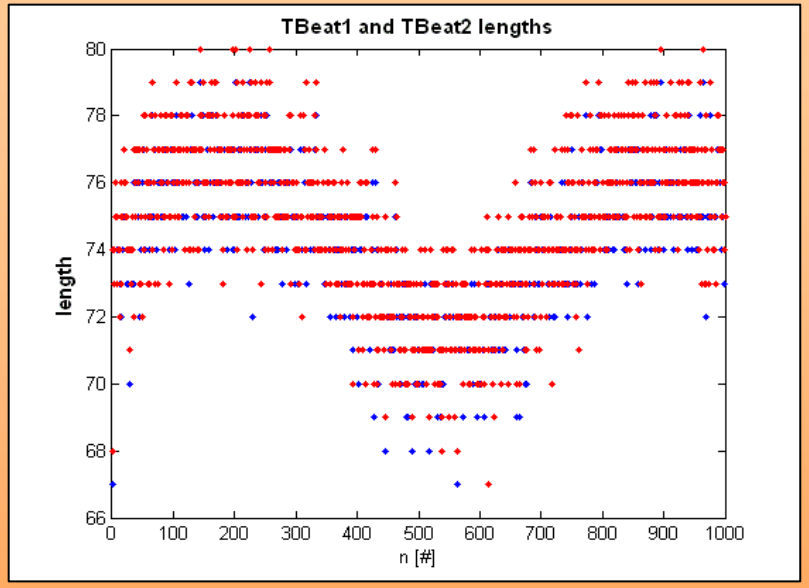




- no correlation is observed
- **XY plot:** if the two values were correlated, the plot will have a form of a uniformly filled circle

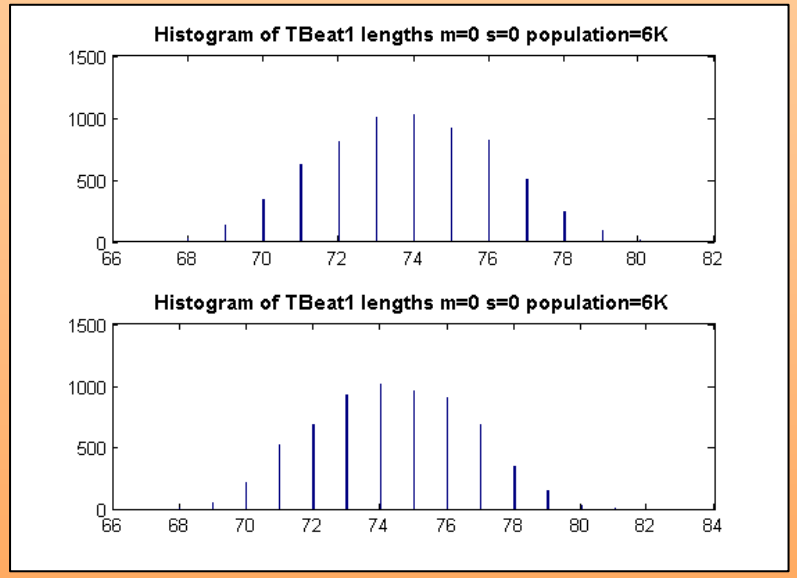
Simulation (5/6)

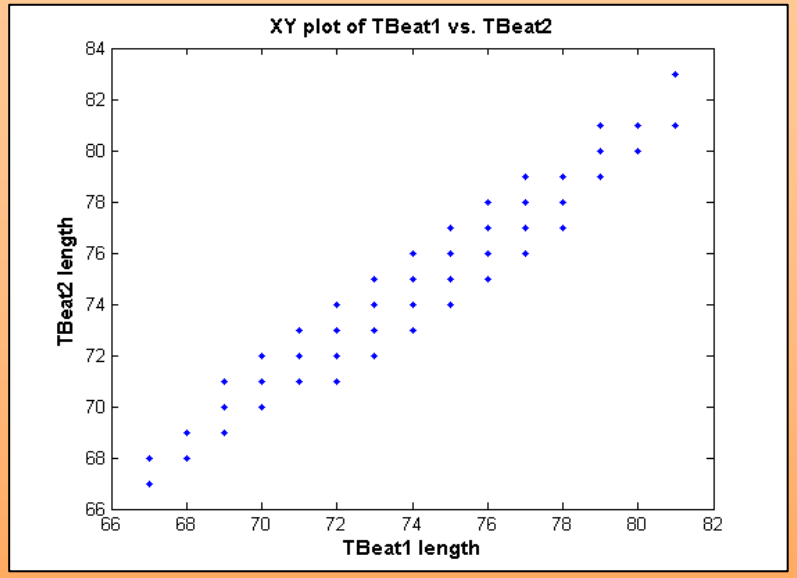




- **Case 2:** T_1 and T_2 have each a Gaussian behavior $\sigma_1 = \sigma_2 = 30ps$
- Global Deteministic component 3Khz sine is added
- T_1 =9,100ns T_2=9,160ns, Δ=60ps

Simulation (6/6)



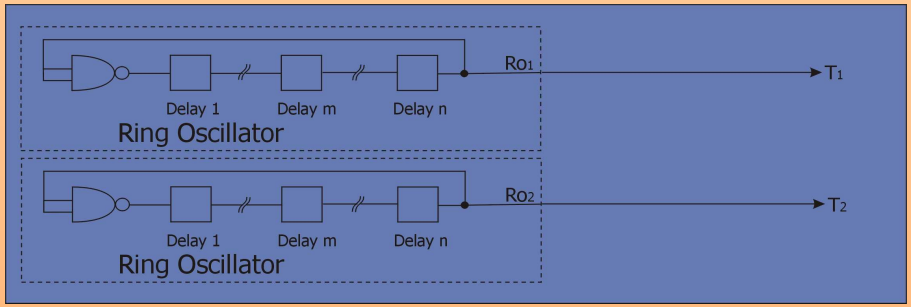


- We observe strong dependency

Practical Results

Practical Results (1/11)

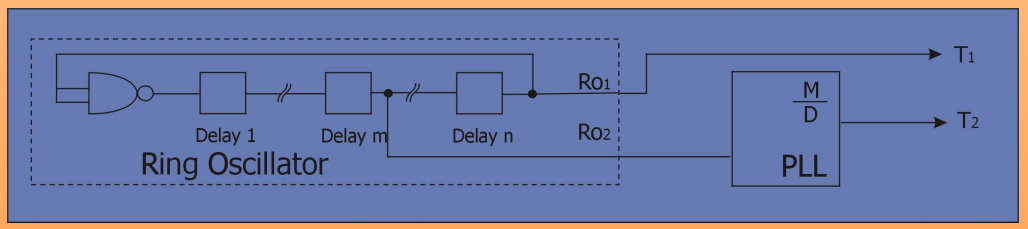
Imlementations



➢ **Case 1:**

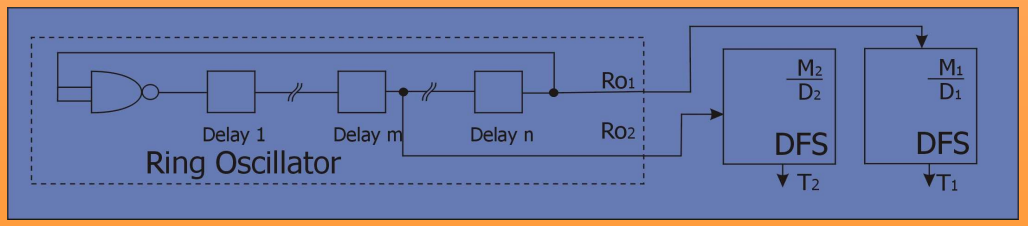
T_1 and T_2 are obtained from identical ring oscillators, they have global deterministic component

Xilinx and Actel implementations



➢ **Case 2:**

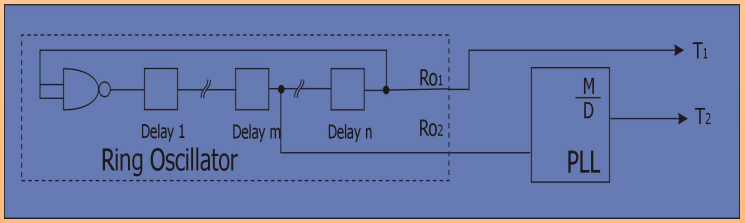
T_1 is obtained from a ring oscillator and T_2 by a PLL (Actel platform)

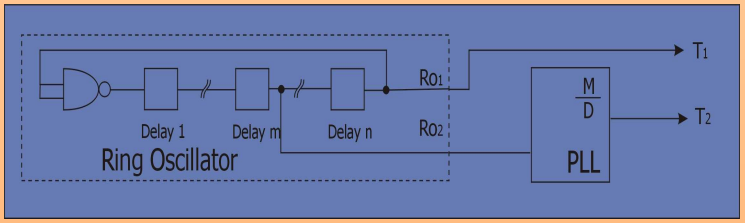


➢ **Case 3:**

T_1 and T_2 are both obtained from a DFS (Xilinx)

Practical Results (2/11)

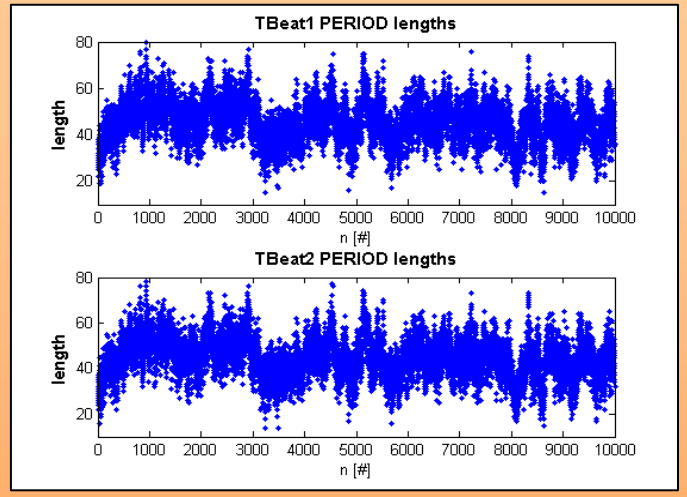


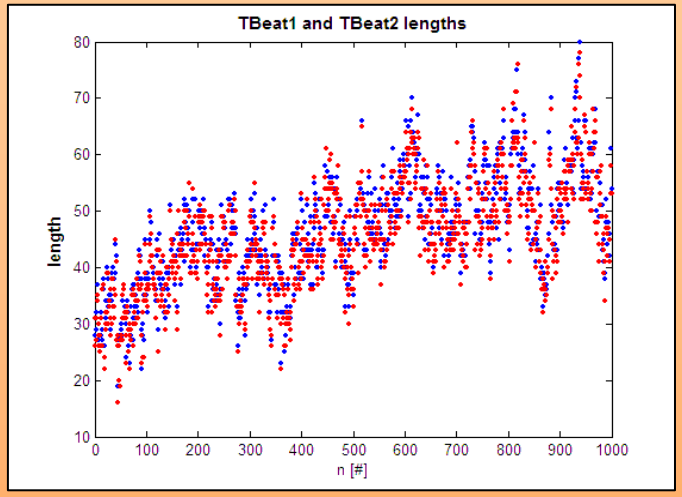


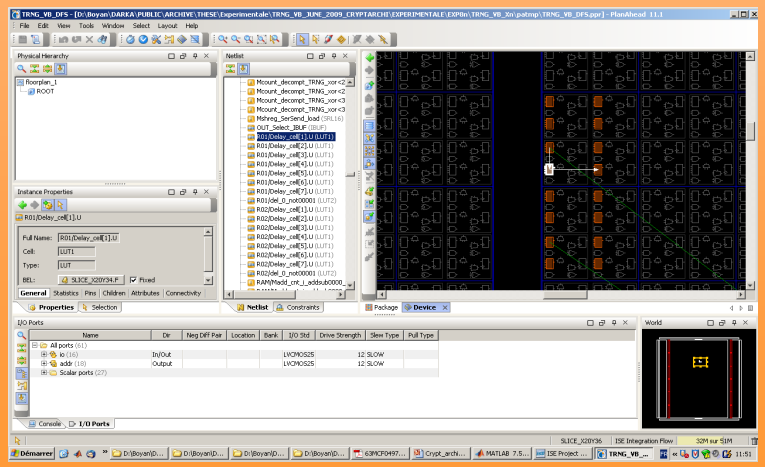
- Implementation of a ring oscillator in **Xilinx**
- Delay element is a LUT

- Implementation of a ring oscillator in **Actel**
- Delay element is an AND2 gate

Practical Results (4/11)



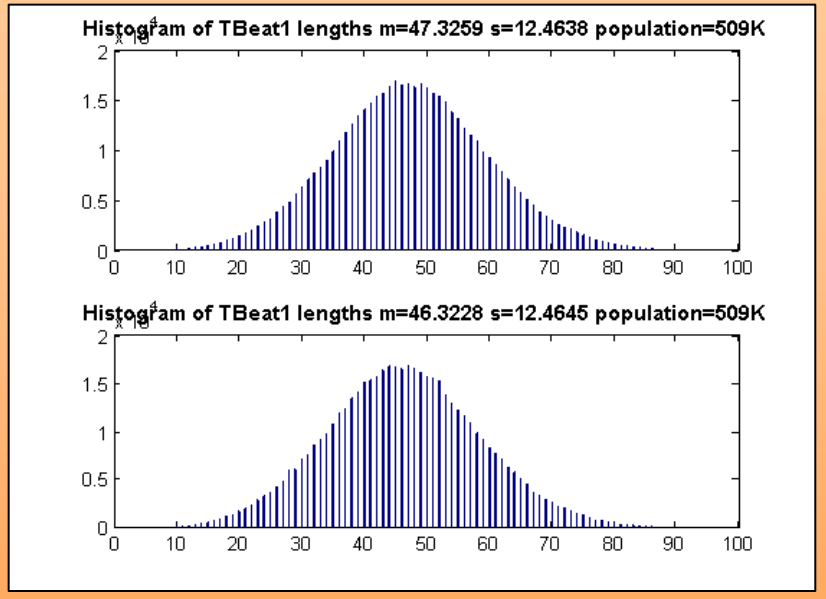


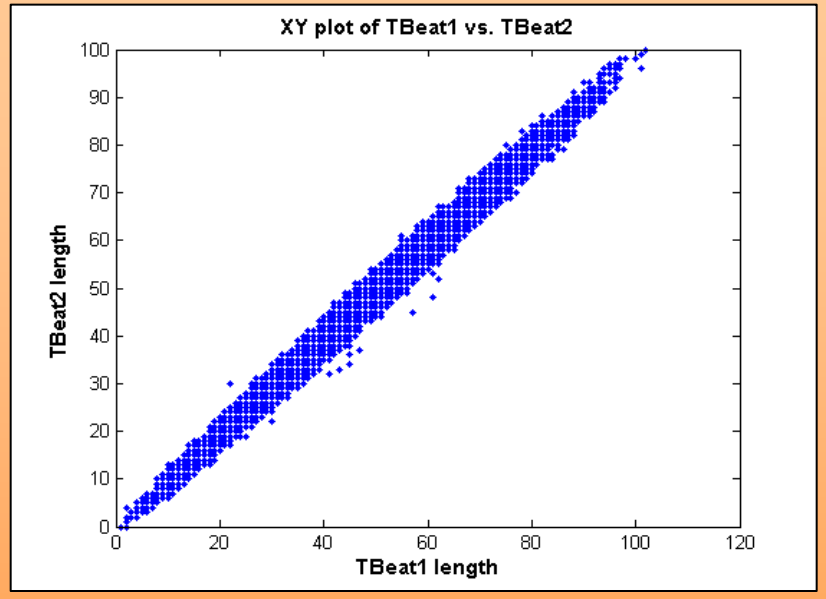


Case 1:

- Ring oscillator based implementation in the **Xilinx** Spartan 3 FPGA
- Strong dependecy is observed T_{Beat1} and T_{Beat2} differ but follow the same tendency!

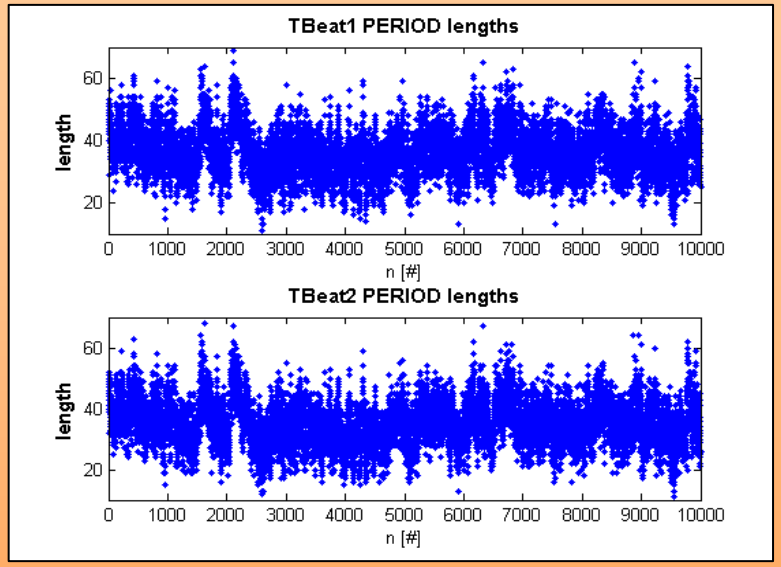
Practical Results (5/11)

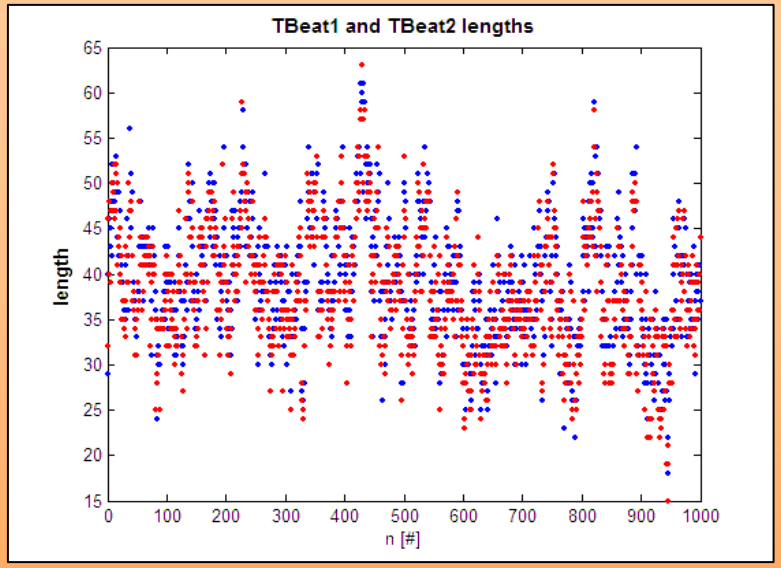


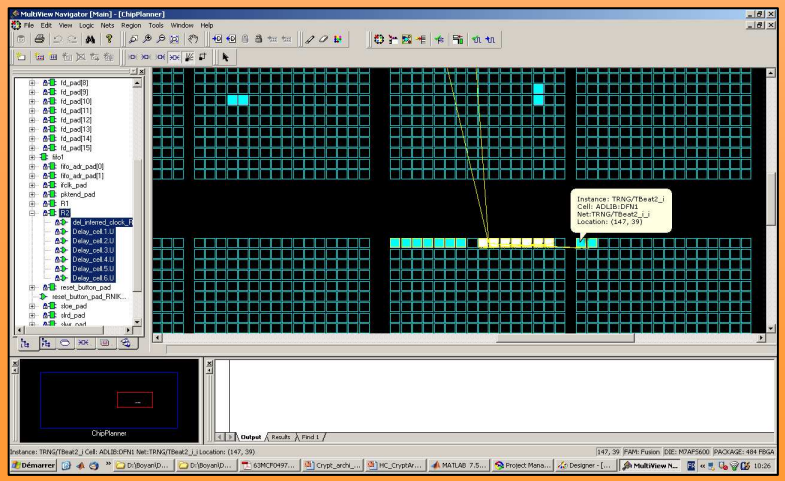


- We see that T_{Beat1} and T_{Beat2} are correlated but individually they are good candidates for randomness extraction
- Δ=30ps T_{Beat}=450Khz

Practical Results (6/11)



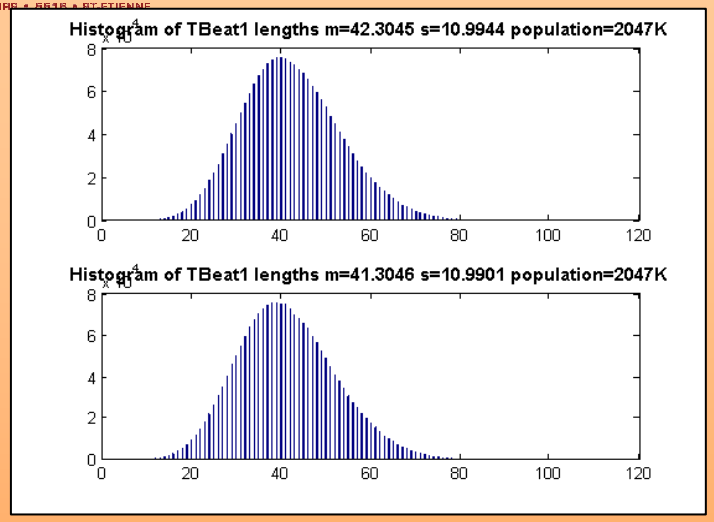


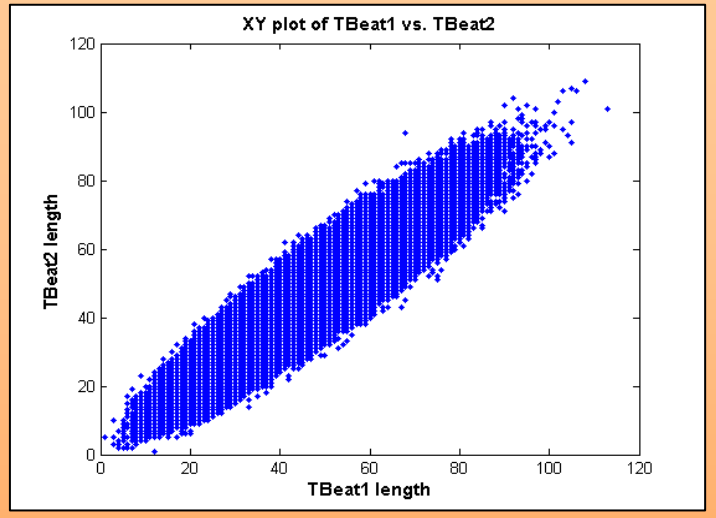


Case 2:

- **Actel** Fusion FS600 ring oscillator based implementation

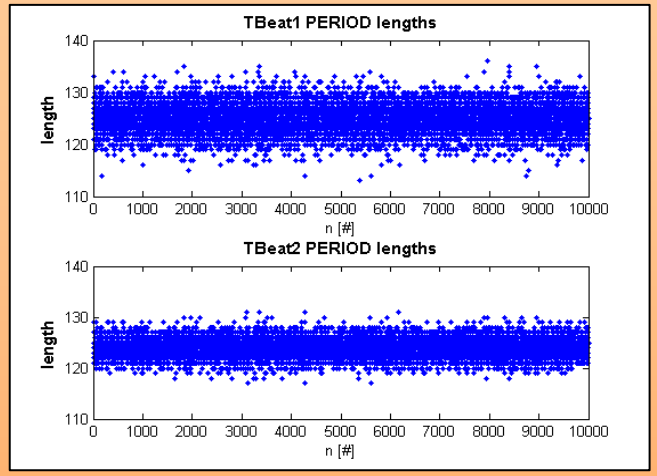
Practical Results (7/11)

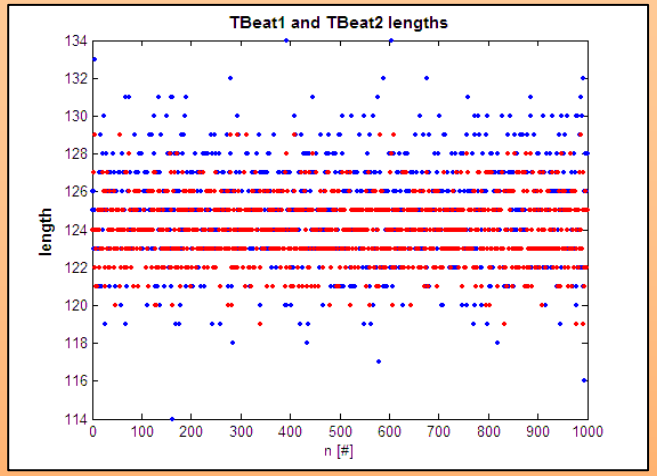




- We see that T_{Beat1} and T_{Beat2} are (less) correlated but individually they are good candidates for randomness extraction
- 7 element ring oscillator Δ=70ps T_{Beat}=900Khz

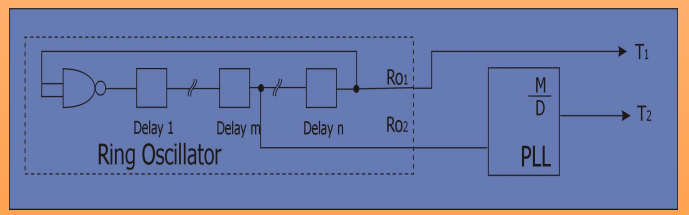
Practical Results (8/11)



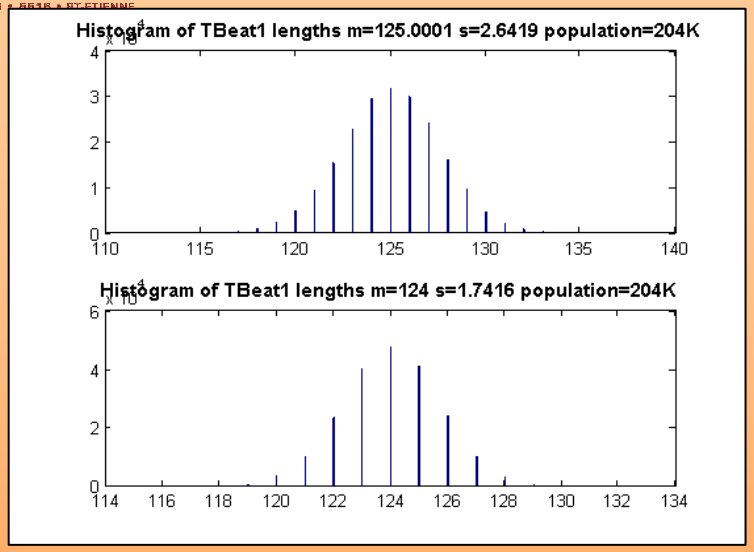


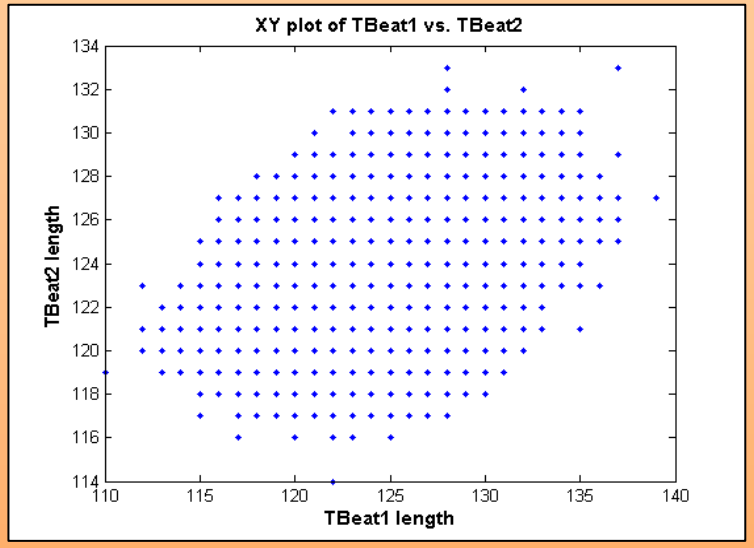
Case 3:

- PLL based implementation in **Actel** FPGA
- RO settings: 10 elements m=0 n=5
- PLL settings: M=127 D=128
- Δ=90ps



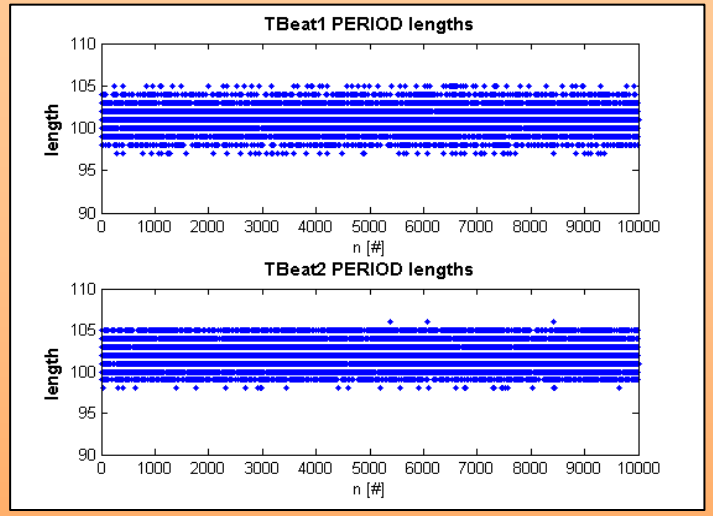
Practical Results (9/11)

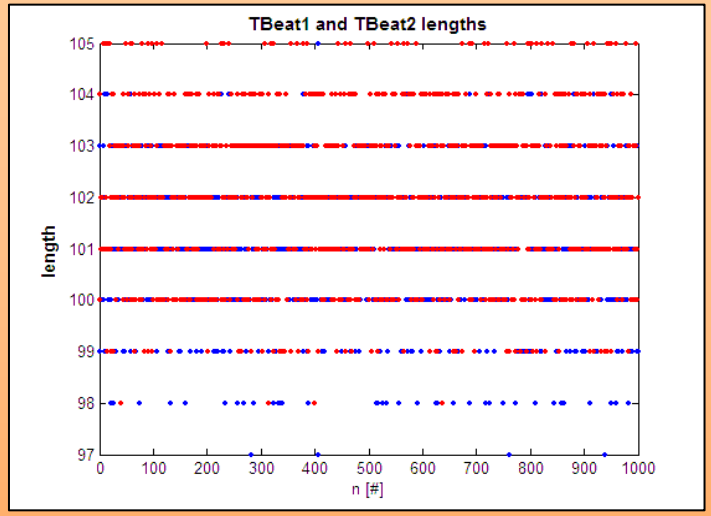




- No correlation is observed
- **Mutual sampling** may be achieved
- Note: standard deviations differs, this is due to the presence of deterministic jitter in the RO obtained signal

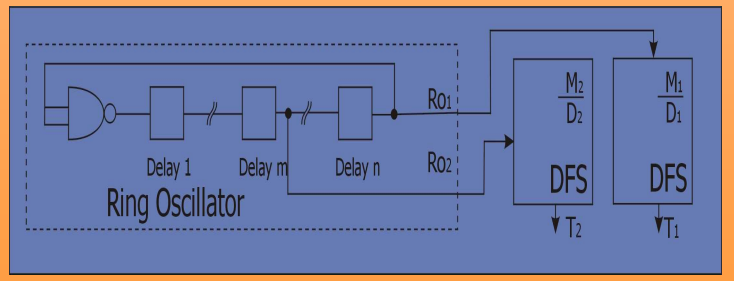
Practical Results (10/11)



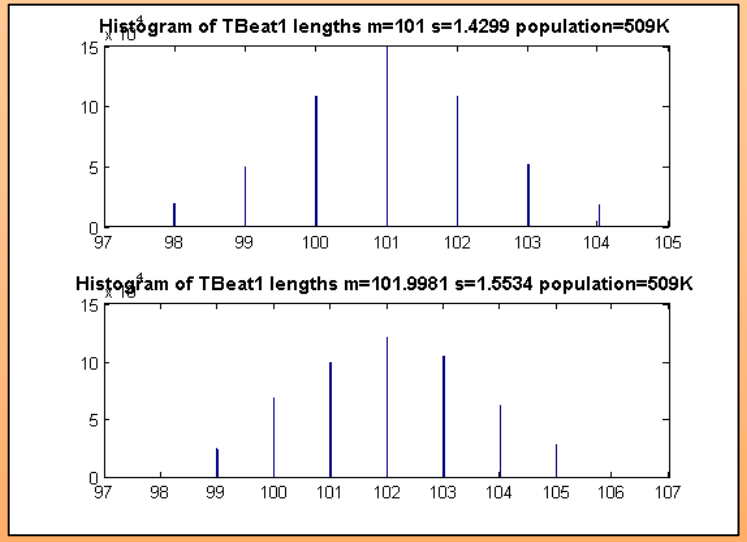


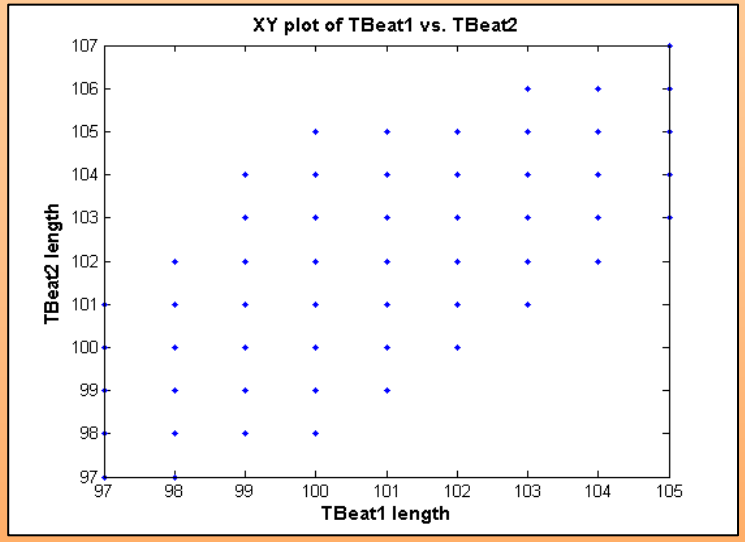
Case 4:

- DFS based implementation in **Xilinx** Spartan 3
- M_1=12 D_1=11 M_2=13 D_2=12 Δ=60ps
- 7 element ring oscillator (~ 9100ns)



Practical Results (11/11)





- Correlation is detected
- No mutual sampling is possible

Statistical Evaluation (1/2)

- **All** the proposed implementation showed **small bias** and passes the **FIPS 140-2** tests
- NIST sts2.0b software was used for statistical evaluation (Unix)
- Mutual sampling of ring oscillators **failed** the NIST tests both in Actel and Xilinx
- Xilinx DFS implementation Failed the NIST tests
- NIST strategy: 200 sequences of 1MBit where evaluated (α=0.01)

Statistical Evaluation (2/2)

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <./E>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
24	20	13	25	28	25	13	18	18	16	0.181557	0.9800	Frequency
19	18	25	27	20	20	18	21	15	17	0.749884	0.9900	BlockFrequency
28	20	15	16	24	16	26	20	22	13	0.255705	0.9850	CumulativeSums
17	12	23	16	26	16	30	19	22	19	0.171867	0.9900	Runs
19	21	26	9	21	21	28	17	16	22	0.176657	0.9850	LongestRun
23	24	24	15	16	17	19	26	16	20	0.616305	0.9950	Rank
4	11	18	18	15	29	23	25	31	26	0.000191	1.0000	FFT
24	22	20	14	17	18	17	21	26	21	0.759756	0.9750	NonOverlappingTemplate
15	27	23	14	17	22	26	20	17	19	0.446556	0.9950	OverlappingTemplate
21	21	20	23	19	22	17	16	19	22	0.985788	0.9950	Universal
32	24	33	18	22	16	19	15	13	8	0.000757	0.9700	ApproximateEntropy
18	11	12	14	10	8	14	14	16	10	0.593823	0.9764	RandomExcursions
8	12	15	10	14	11	10	15	16	16	0.680410	0.9921	RandomExcursionsVariant
27	16	23	19	27	26	20	13	15	14	0.141256	0.9900	Serial
20	22	22	18	22	15	15	23	23	20	0.897763	0.9900	LinearComplexity

The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 0.968893 for a sample size = 200 binary sequences.

The minimum pass rate for the random excursion (variant) test is approximately 0.963513 for a sample size = 127 binary sequences.

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <./L>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
9	4	9	13	8	14	8	11	11	13	0.514124	0.9900	Frequency
21	15	13	10	7	7	8	6	9	4	0.006196	0.9800	BlockFrequency
6	10	9	9	13	17	9	10	9	8	0.514124	0.9900	CumulativeSums
11	9	13	4	4	13	12	10	14	10	0.262249	1.0000	Runs
19	12	11	7	8	9	4	12	6	12	0.066882	0.9700	LongestRun
11	10	17	4	6	12	14	7	11	8	0.137282	0.9700	Rank
1	8	13	8	14	14	5	14	17	6	0.004981	1.0000	FFT
13	6	10	11	11	12	9	9	10	9	0.946308	1.0000	NonOverlappingTemplate
19	13	12	5	12	11	7	8	6	7	0.062821	0.9600	OverlappingTemplate
10	7	10	8	8	11	9	17	13	7	0.474986	0.9900	Universal
17	10	7	15	13	13	6	5	9	5	0.051942	0.9600	ApproximateEntropy
6	5	3	5	6	13	6	10	5	6	0.170294	1.0000	RandomExcursions
4	7	4	8	9	5	8	10	6	4	0.551026	0.9846	RandomExcursionsVariant
11	18	14	11	5	8	9	8	8	8	0.191687	1.0000	Serial
11	11	15	8	8	16	5	7	11	8	0.275709	0.9800	LinearComplexity

The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 0.960150 for a sample size = 100 binary sequences.

The minimum pass rate for the random excursion (variant) test is approximately 0.952976 for a sample size = 65 binary sequences.

Actel PLL implementation

2MBits/second

Mutual sampling

4 random bits / T_{Beat} period

Xilinx Ring Oscillator implementation

1.38MBits/second

No mutual sampling

2 random bits / T_{Beat} period

Conclusions

- Extension of TRNG principle was proposed
- No bias, no postprocessing!
- Extracting 2 random bits per T_{Beat} period was validated
- Extracting 4 random bits per T_{Beat} period was validated in special case
- Actel ‘push button’ implementation of TRNG was proposed
- Strong dependency of ring oscillator was observed

Perspectives

- Use an external VCO in order to avoid dependencies and be able to perform multi Δ measurements
- Implement in Virtex 5 FPGA (better DFS resolution) in order to validate the 4bits extraction (mutual sampling) in Xilinx
- Serial cascading of DFS
- Mathematical modeling for the global deterministic component