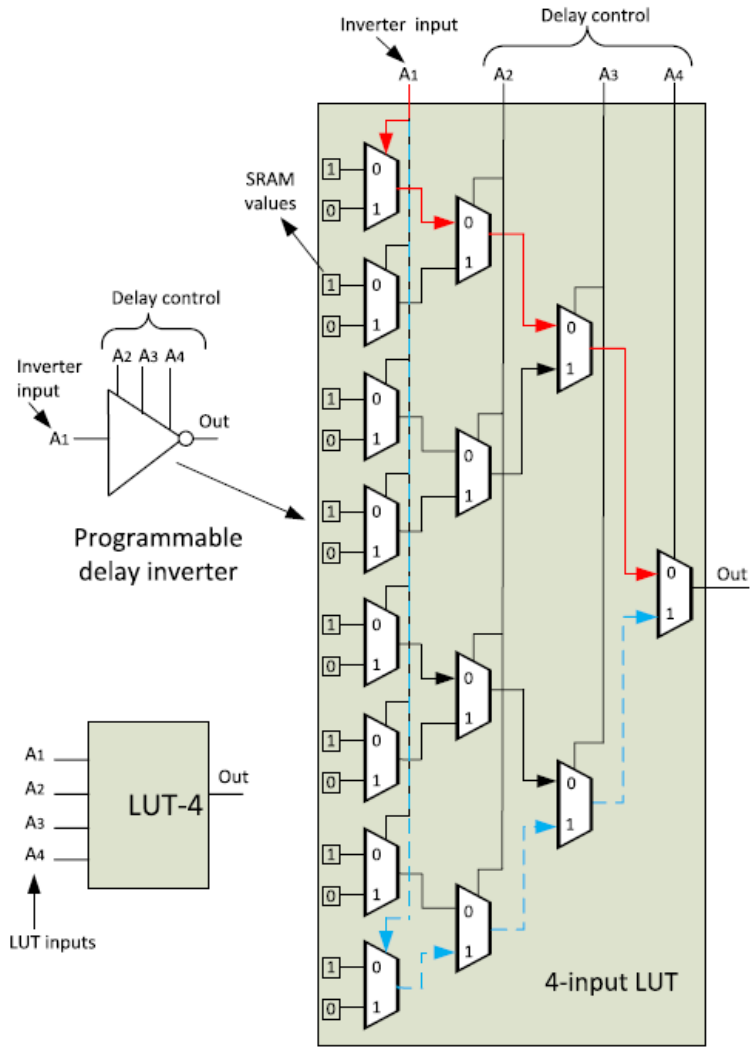
FPGA-Based True Random Number Generation Using Programmable Delays in Oscillator-Rings

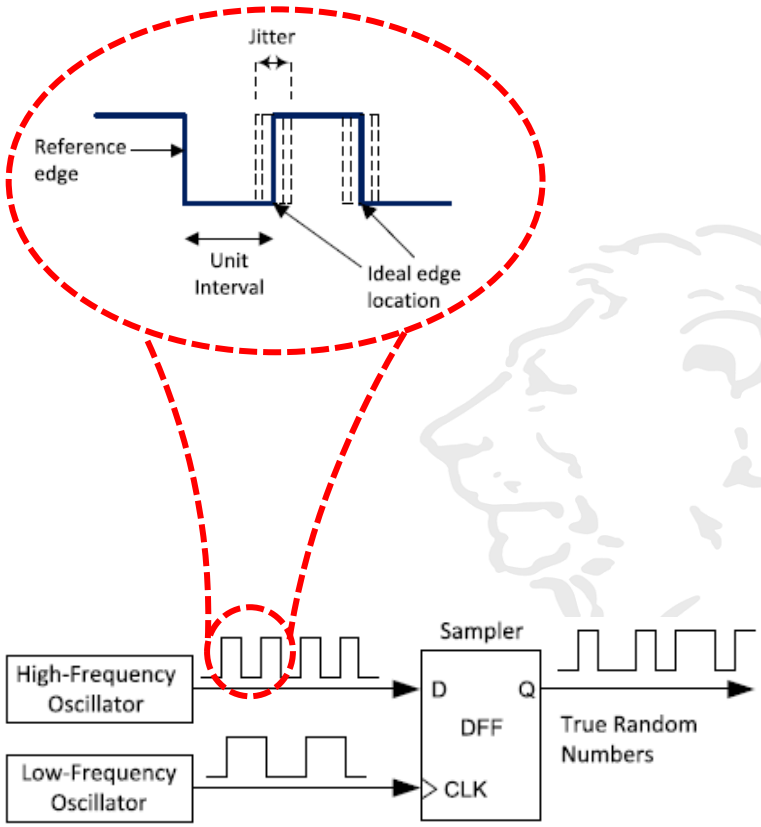
By-
Aditya Sharma (19115011)

True Random Number Generators

- True Random Number Generators (TRNGs) are widely used in cryptographic applications such as key generation, random padding bits, and generation of challenges in authentication protocols.
- The TRNGs must fulfill strict statistical requirements, be unpredictable and generate truly random numbers by making use of a physical source that is non-deterministic.
- Typical TRNGs use a single source of entropy and post-processing operation.
- FPGA-based TRNGs derive entropy from the jitter of Ring Oscillators [1].
- However, the entropy of the output bit sequence from the TRNG is drastically reduced when equal length oscillator rings configured in FPGAs are highly correlated with each other due to identical delays.
- To address this issue, we use programmable delay lines (PDLs) in the oscillator rings to create higher variation in RO oscillations from cycle to cycle and causes jitter in generated RO clocks [2].

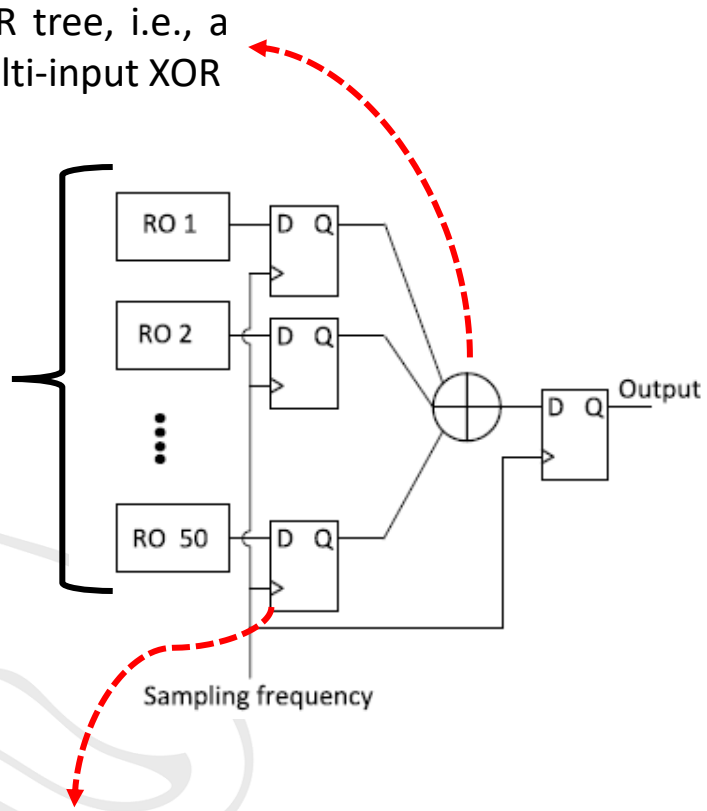


RO-based TRNG: Overview



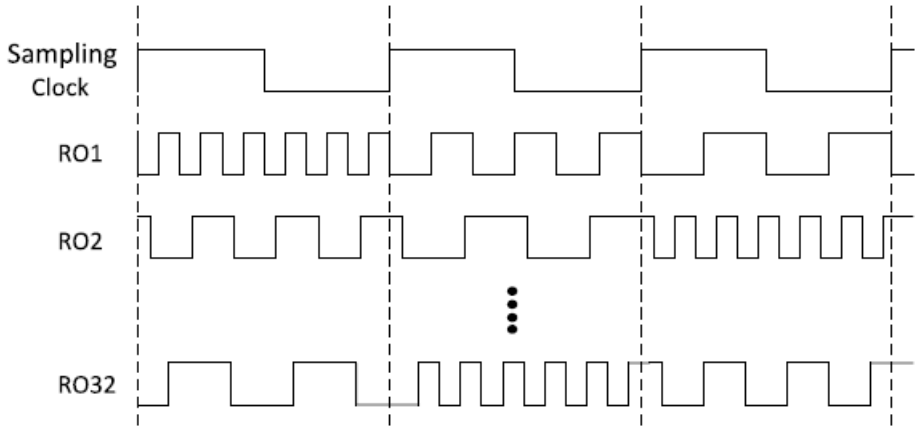
Since digital value of an oscillator's output changes periodically, its random jitter is used to generate a stream of truly random bits with DFF based sampler

The quality of generated true random bits is improved by using multiple free-running RO's



DFF sampling each RO's output protect XOR tree from high RO switching activity

RO-based TRNG: Design

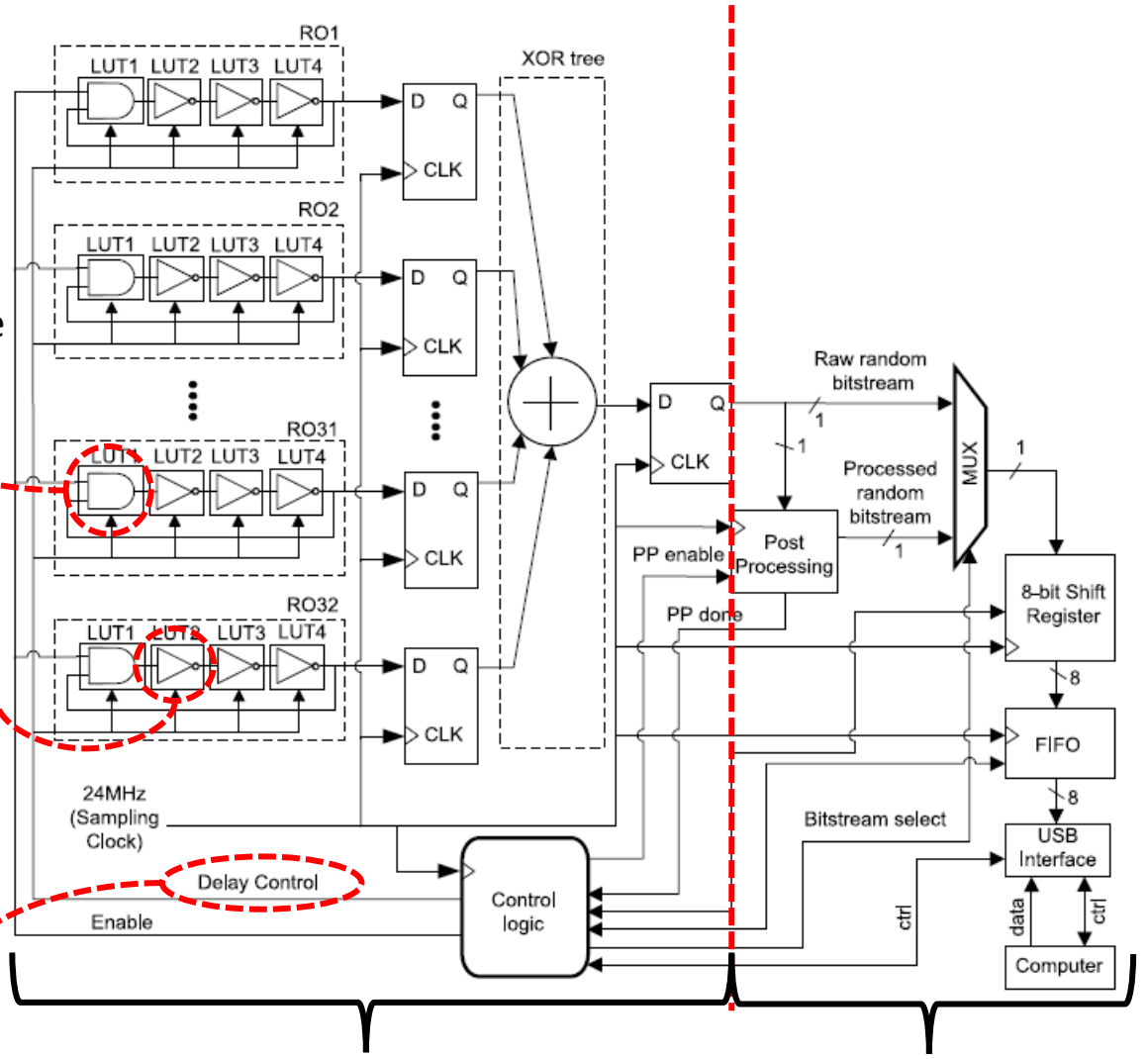


Increments inverter/PDL delay by 1 at rising edge of the 24MHz CLK

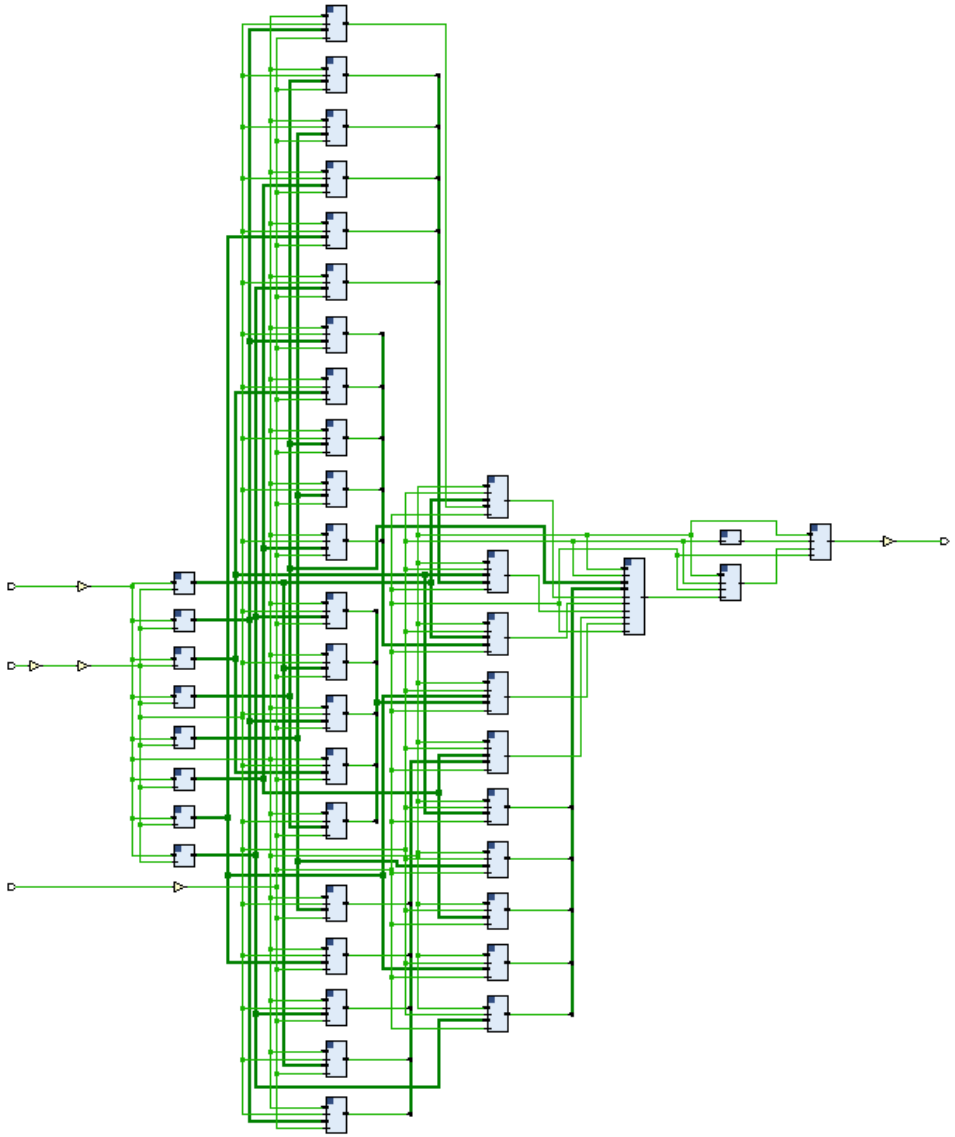
AND gates enable /disable the ROs

PDL-based LUT for inverter operation

Arbitrary 3-bit delay initialized to all PDL-based LUTs



RO-based TRNG: Constraints & Synthesis



```
create_clock -period 10.000 -name clk -waveform {0.000 5.000} -add clk

set_property PACKAGE_PIN Y9 [get_ports clk]
set_property IOSTANDARD LVCMOS18 [get_ports clk]
set_property PACKAGE_PIN F22 [get_ports clr]
set_property IOSTANDARD LVCMOS18 [get_ports clr]
set_property PACKAGE_PIN G22 [get_ports en]
set_property PACKAGE_PIN T22 [get_ports rand_num]
set_property IOSTANDARD LVCMOS18 [get_ports en]
set_property IOSTANDARD LVCMOS18 [get_ports rand_num]

set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_0/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_1/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_2/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_3/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_4/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_5/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_6/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_7/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_8/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_9/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_10/z4];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets  pdl_11/z4];
```

Since the RO is also taken up as combinatorial loop, this warning is suppressed manually for all PDLs

Tcl Console | Messages | Log | Reports | Design Runs | Package Pins | I/O Ports

Name	Direction	Board Part Pin	Board Par...	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination
All ports (4)														
Scalar ports (4)														
clk	IN				Y9	✓	13	LVCMOS18	1.800				NONE	NONE
clr	IN				F22	✓	35	LVCMOS18	1.800				NONE	NONE
en	IN				G22	✓	35	LVCMOS18	1.800				NONE	NONE
rand_num	OUT				T22	✓	33	LVCMOS18	1.800		12		NONE	FP_VTT_50

RO-based TRNG: Implementation

Future Work

- Post-processing techniques could be added to the current implementation to increase overall TRNG entropy.
- NIST statistical tests need to be employed on the generated random bitstream to verify its proposed entropy effectiveness.
- We also plan to work on RO-based TRNG designs with a much larger set of LUT instances, i.e., for increased jitter, and evaluate the relative improvement in entropy.
- As an extension of the work, we plan to employ serialized/pipelining/parallelism techniques to implement RO-based TRNGs.

References

[1] B. Sunar, W. J. Martin, and D. R. Stinson, “A provably secure true random number generator with built-in tolerance to active attacks,” IEEE Trans. Comput., vol. 56, no. 1, pp. 109–119, Jan. 2007.

[2] N. Nalla Anandakumar, S. K. Sanadhya and M. S. Hashmi, "FPGA-Based True Random Number Generation Using Programmable Delays in Oscillator-Rings," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 570-574, March 2020, doi: 10.1109/TCSII.2019.2919891.

[3] https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf

Thank You!