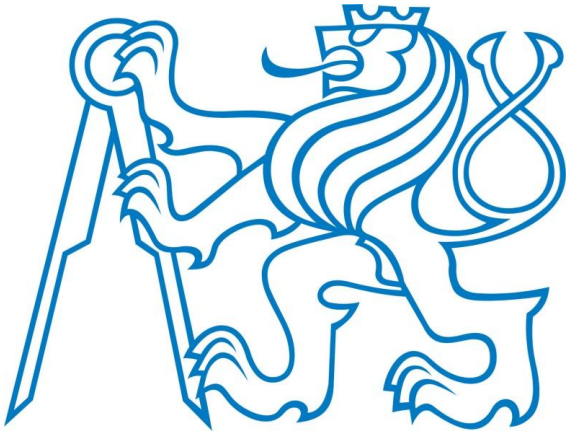
A novel ring oscillator based PUF on FPGA

Filip Kodýtek and Róbert Lórencz

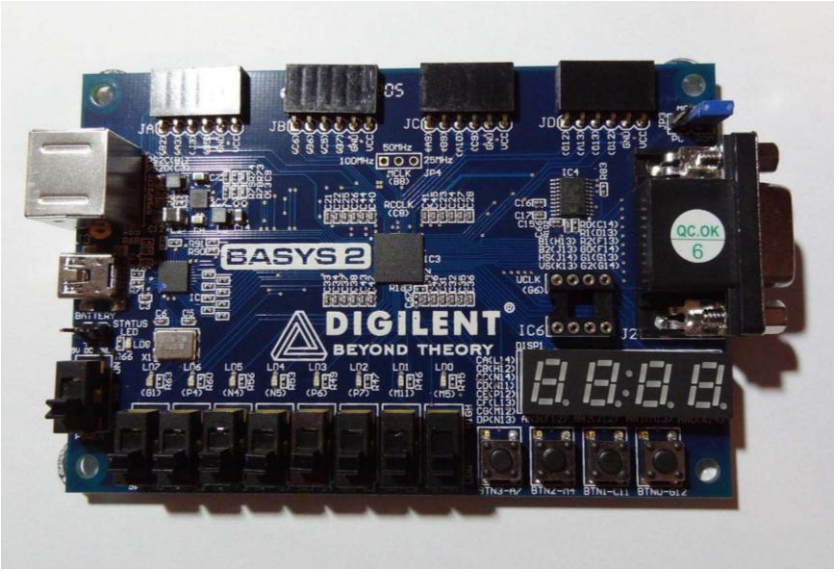
Czech Technical University in Prague
Faculty of Information Technology



CryptArchi 2014

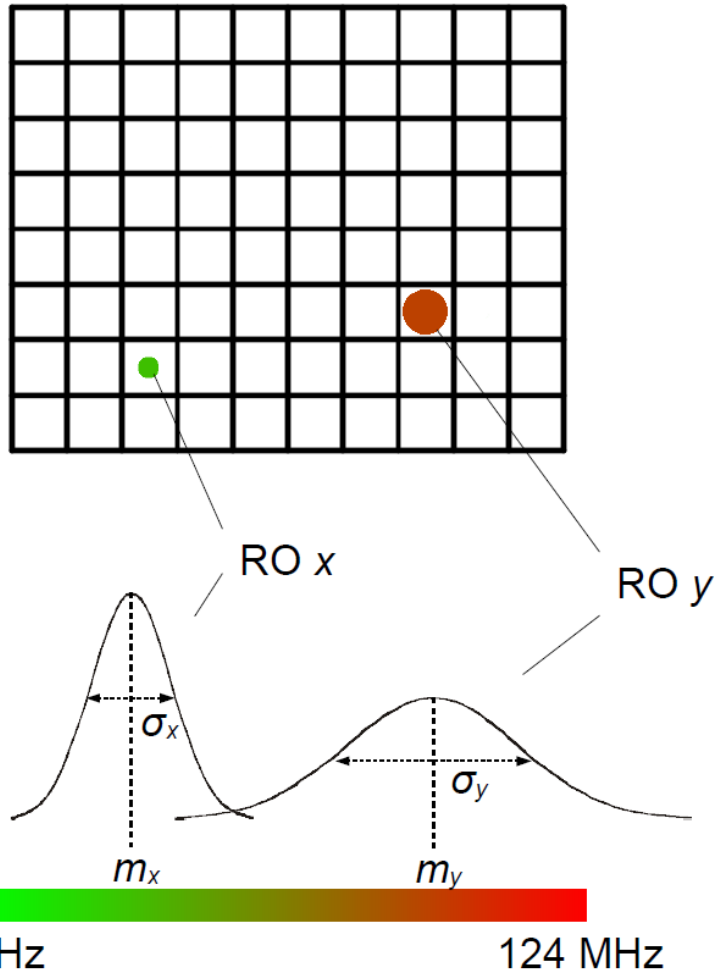
Motivation and goals

- PUF proposal for FPGA
  - easy to implement
  - area efficient
- Implementation and statistical evaluation of PUF
  - tested on 24 Digilent Basys 2 FPGA boards
- Designing key generation using the proposed PUF design

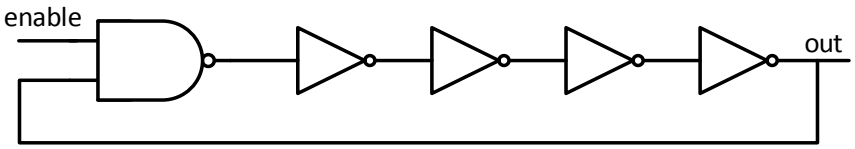


Source of the entropy

- Ring oscillator (RO) based PUF



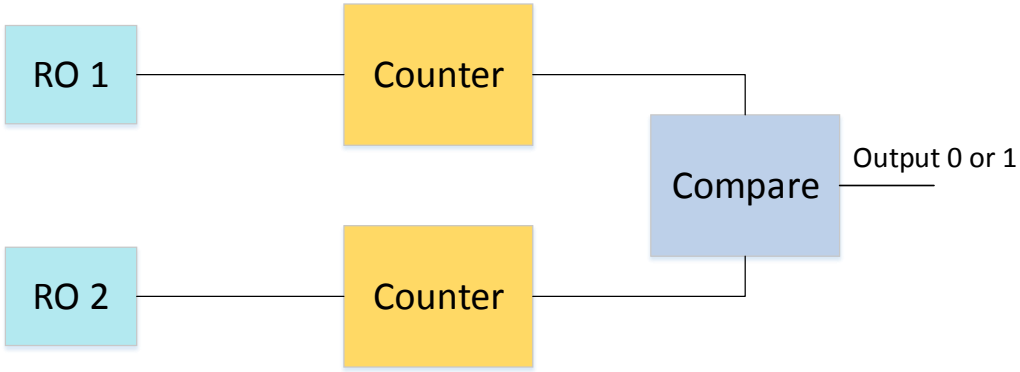
Basic building element



Feiten, L., Martin, T., Becker, B. Further Examination of Altera Cyclone IV RO-PUF Capabilities. TRUDEVICE 2013.

PUF proposal

- Classical approach
  - Frequency comparison



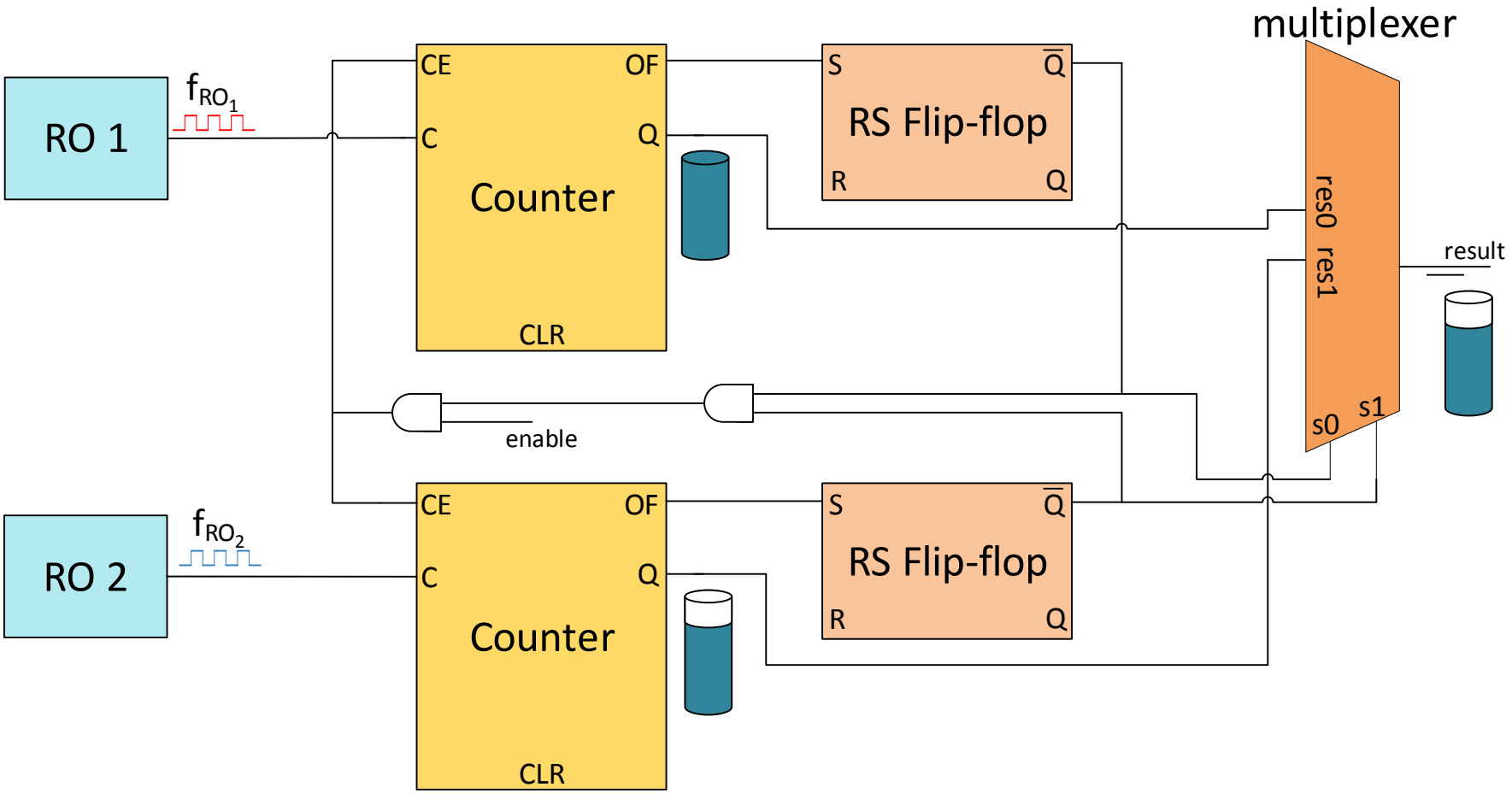
Suh, G. E., Devadas, S. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Design Automation Conference.*

- TERO PUF

Bossuet, L., Ngo, X. T., Cherif, Z., Fischer, V. A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon. In *IEEE Transactions on Emerging Topics in Computing.*

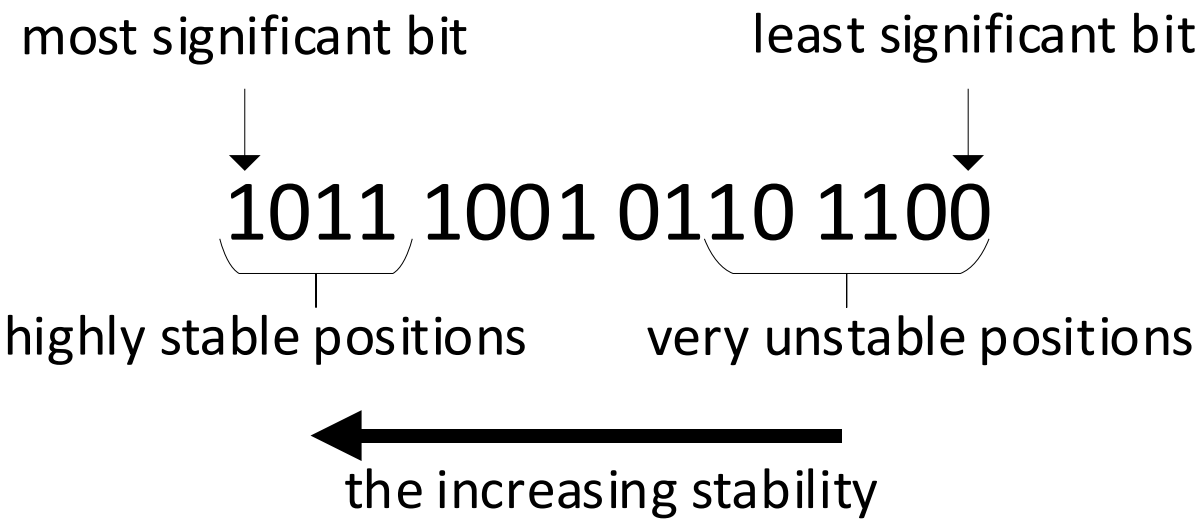
- Our design:
  - ROs do not need to be mutually symmetric
  - More output bits from one pair of ROs

Measuring the number of cycles



Processing counter values

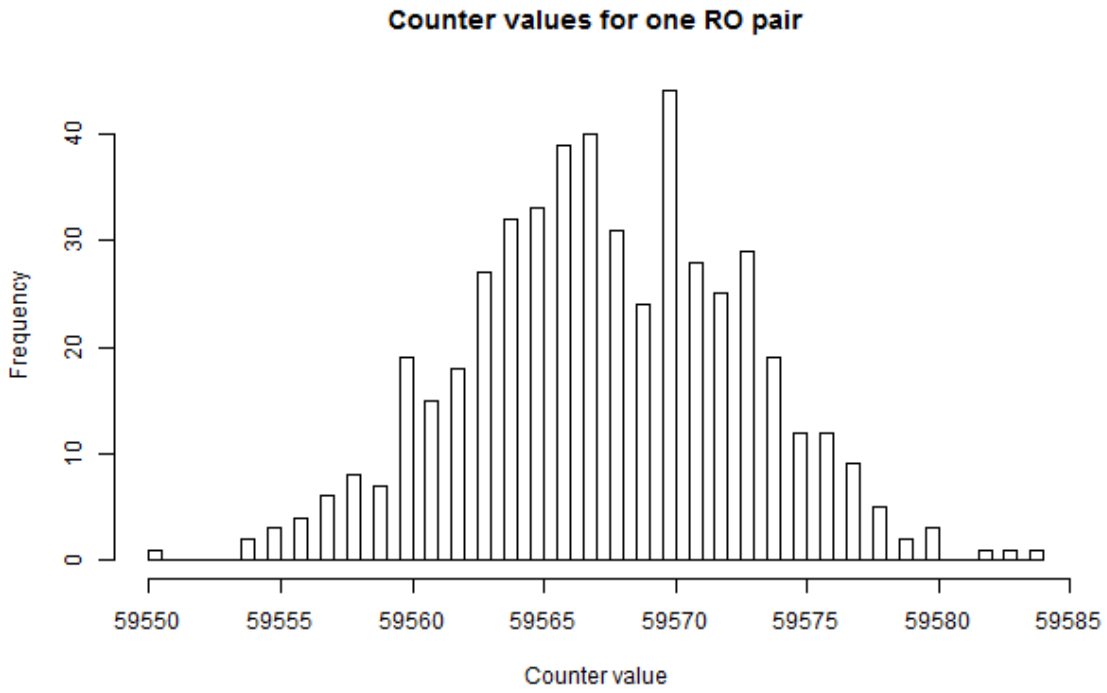
- Counter values are used directly
- Appropriately selected part of this value can be used for PUF



Stability

- Stable positions are required for PUF
- The average stability for position *i* and *n* RO pairs:

$$s_i = \frac{1}{n}\sum_{j=1}^{n} s_i(RO_j)$$

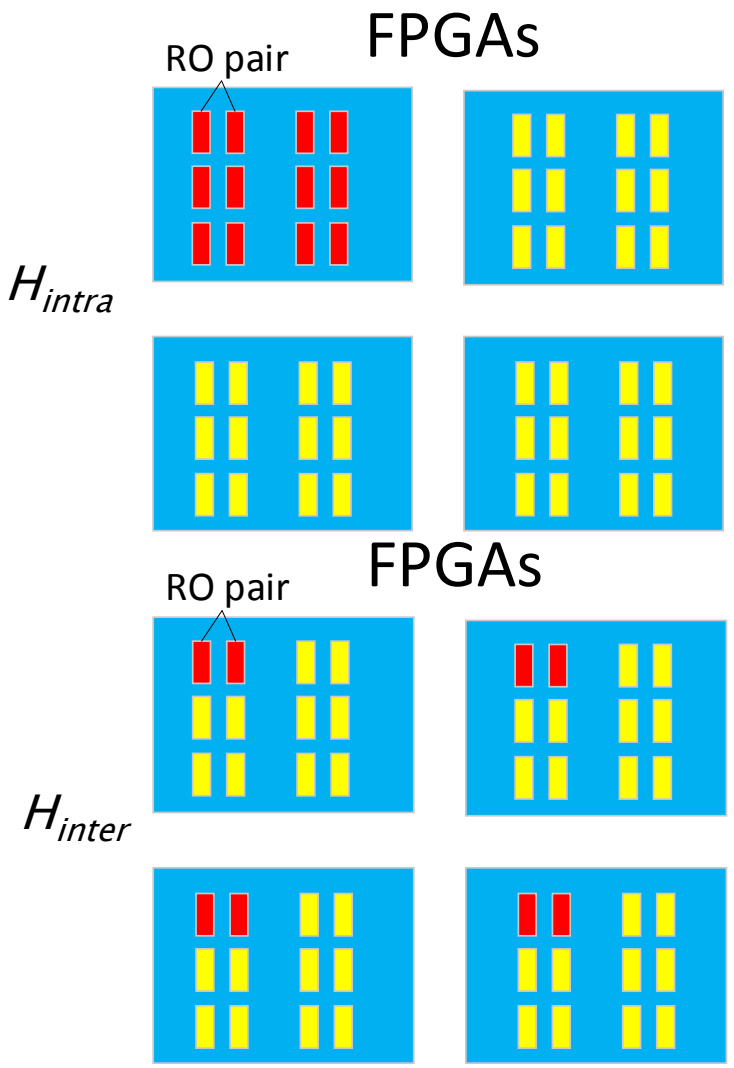


Entropy

- Selected bits have to be unique
- H_{intra} – entropy within each FPGA separately
- H_{inter} – entropy of particular RO pairs among different FPGAs

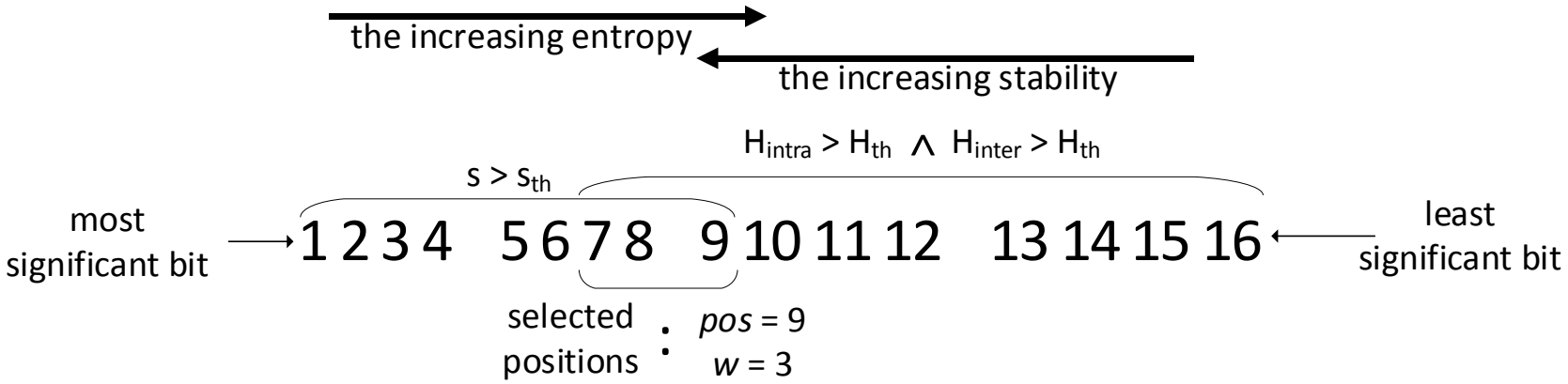
$$H_{intra}(i) = -\frac{1}{m}\sum_{j=1}^{m}\sum_{k=0}^{1} p_j(k)\log_2(p_j(k))$$

$$H_{inter}(i) = -\frac{1}{n}\sum_{l=1}^{n}\sum_{k=0}^{1} p_l(k)\log_2(p_l(k))$$



Method of selecting suitable positions

- We are looking for the intersection of high stability and high entropy
- s_{th} and H_{th} – threshold values



Position statistics

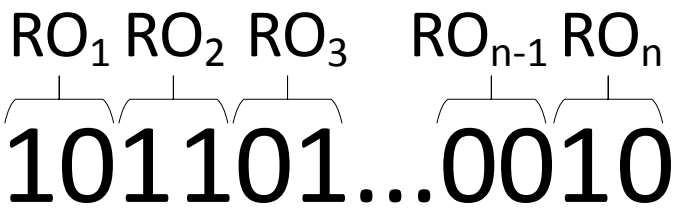
- Measurements were performed on a circuit containing 300 ROs
- 16-bit counters
- 150 RO pairs – 1000 measurements
- 450 RO pairs – 500 measurements

position(i)	s_i	H_{intra}	H_{inter}	bias
6	0.9961	0.9973	0.6585	0.5233
7	0.9920	0.9982	0.9232	0.5118
8	0.9841	0.9986	0.9682	0.4978
9	0.9677	0.9984	0.9692	0.4971

PUF design

- Selection of suitable bit positions based on the statistics
- *w* bits from each RO pair
- The PUF output is created by concatenating thus obtained bits

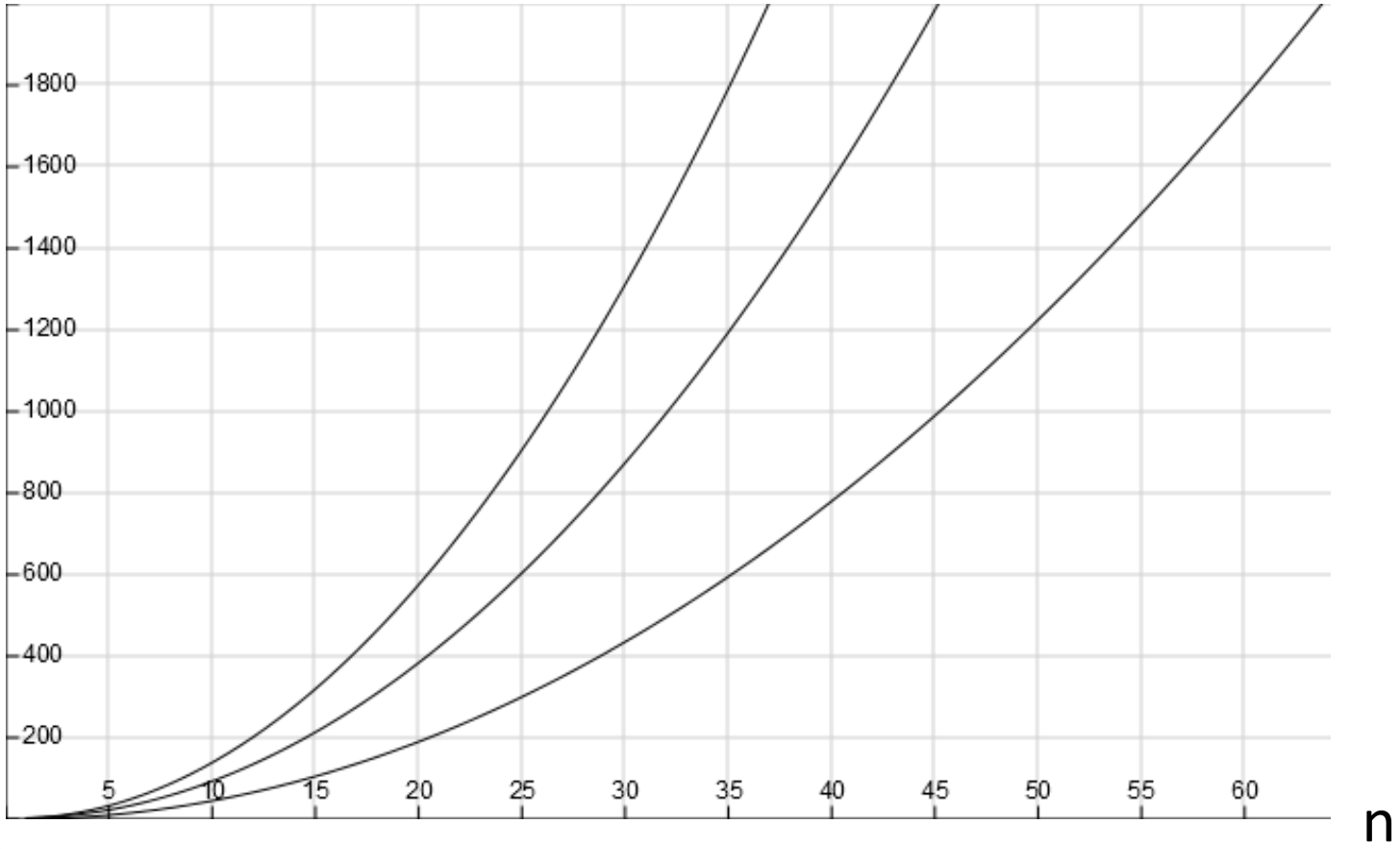
⇒ maximum amount of bits: $\binom{n}{2} * w$



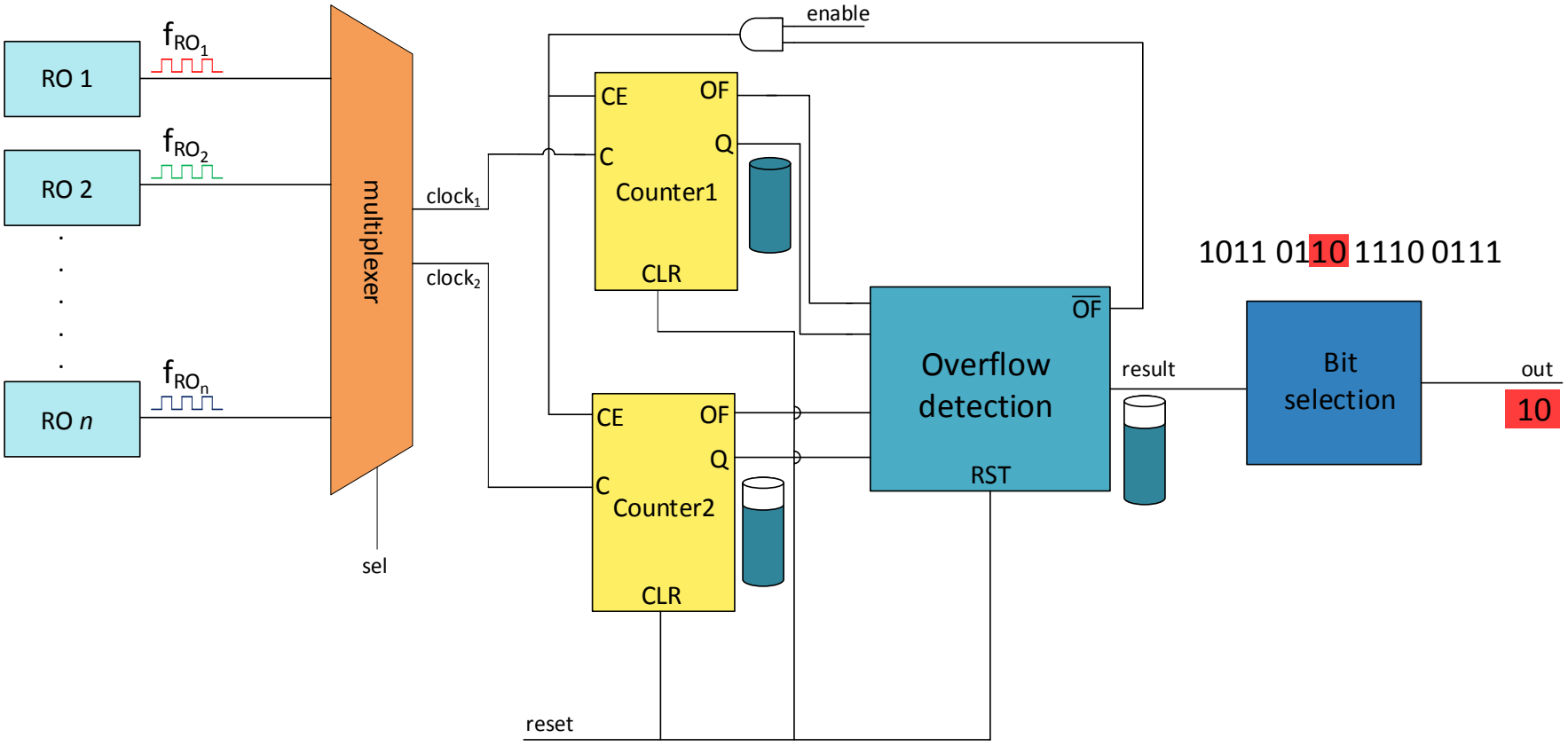
Dependency of the maximum number of bits on *w* and *n*

Max # of bits

$$\binom{n}{2} * w$$



PUF design


RO 1
RO 2
RO n
f_{RO_1}
f_{RO_2}
f_{RO_n}
multiplexer
sel
$clock_1$
$clock_2$
enable
CE
OF
C
Q
Counter1
CLR
CE
OF
C
Q
Counter2
CLR
reset
Overflow detection
$\overline{OF}$
RST
result
Bit selection
out
10
1011 0110 1110 0111

Statistical evaluation

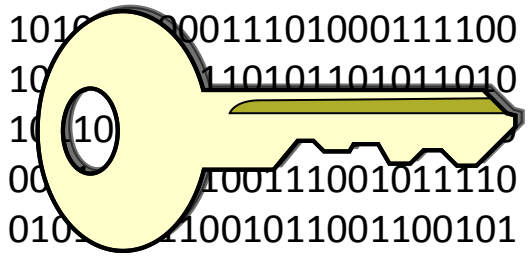
- BER and HD_{intra} should be ideally 0%
- Ideal value of HD_{inter} is 50%

positions	6–8	7–8	7–9	8–9
w	3	2	3	2
BER	0.92%	1.19%	1.87%	2.41%
HD_{intra}	1.37%	1.78%	2.79%	3.6%
HD_{intra} interval	<0%, 3.56%>	<0%, 4.56%>	<0.74%, 6.37%>	<1.11%, 8.22%>
HD_{inter}	42.69%	48.42%	48.94%	49.96%
HD_{inter} interval	<34.67%, 52.3%>	<42.33%, 56.11%>	<44.74%, 54.74%>	<45.44%, 54.67%>

- ROs are not mutually symmetric

Key generation

- Desired properties of generated keys:
  - Stability
  - Uniqueness



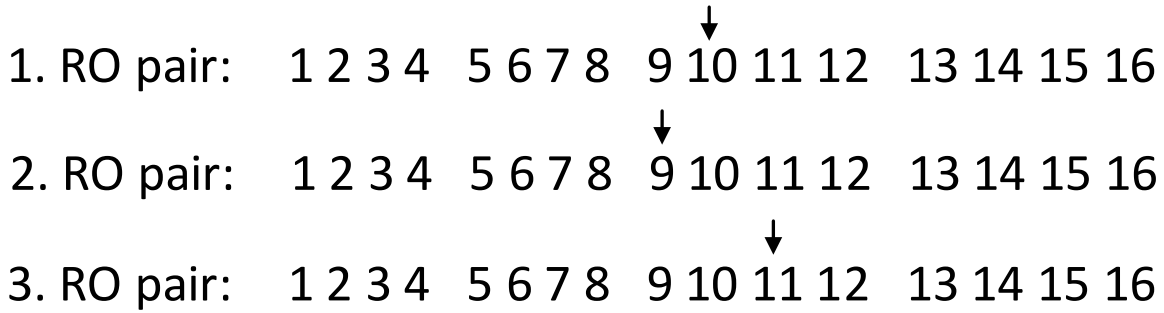
- PUF responses usually contain errors
- Additional steps to stabilize PUF responses
  - Modification of suitable bits extraction
  - Error correction code

Modification of suitable bits extraction

- So far, the suitable positions were selected globally for all RO pairs
- Each RO pair has **different suitable positions** for PUF

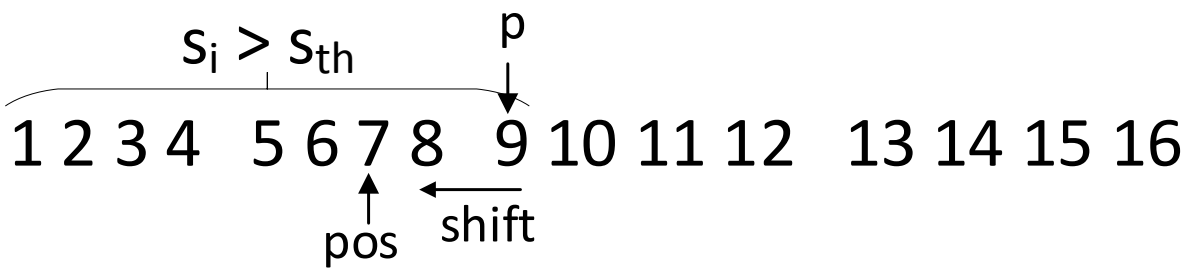
⇒ it is better to find suitable positions for each RO pair separately

  - More bits from one RO pair
  - Higher stability



Modification of suitable bits extraction

- *k* measurements for each RO pair
- Stability of each position is checked from the most to the least significant bit ($s_i > s_{th}$)
- Shift towards the most significant bit is applied ⇒ higher stability
- These positions for each RO pair have to be stored



Error correction code

- Burst errors caused by the overflow of particular counter value parts
- Repetition code R(n)

Measurement 1: 1001 1111 1111 1111
Measurement 2: 1010 0000 0000 0000

Correcting string creation

1st PUF output: 10110|01100|…|01011
⊕
1st bit of each block: 11111|00000|…|00000
Correcting string: 01001|01100|…|01011

PUF output correction

2nd PUF output: 11110|00000|…|01010
⊕
Correcting string: 01001|01100|…|01011
Result of XOR: 10111|01100|…|00001
1 0 0
Key: 10…0

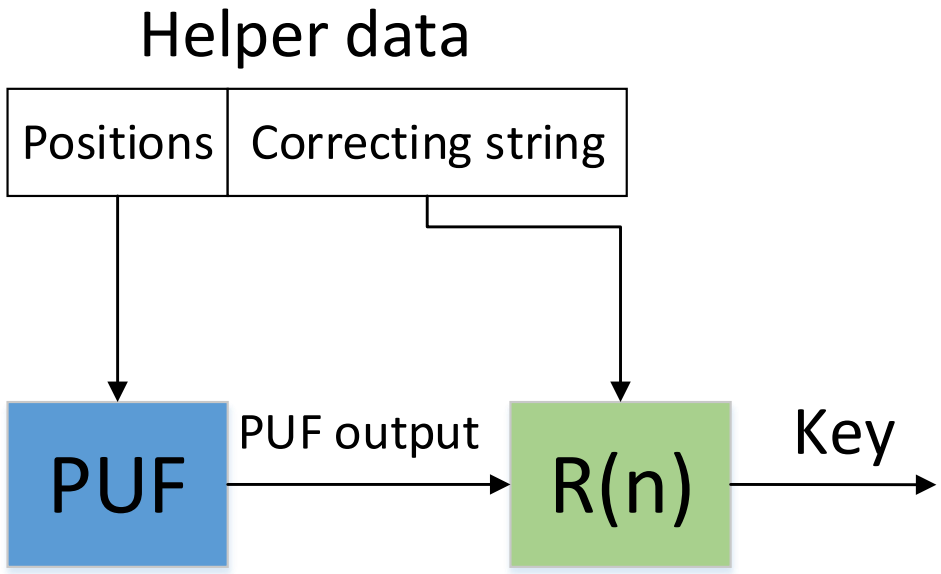
Key generation

- 2 phases:
  - Initialization phase
  - Re-generation phase

- Initialization phase
  - Determination of suitable position for each RO pair
  - Creation of correcting string for repetition code
  - Positions and correcting string represent *helper data*

Key generation

- Re-generation phase
  - Generation of PUF output based on the positions from helper data
  - PUF output correction using the previously created correcting string



Key generation – results

- 450 RO pairs
- $w = 3$
- $s_{th} = 1$

shift	0	1	2	3
min n for R(n)	15	15	12	9
HD_{inter}	48.7%	47.99%	45.97%	42.89%
Bias	50.79%	51.39%	53.5%	56.36%
Key length	90	90	112	150

Conclusion

- PUF proposal
  - More bits from one RO pair
  - ROs do not need to be symmetric
- Key generation
  - Modification of the proposed PUF design
  - Repetition code

Future work

- Testing other FPGAs
- Influence of voltage and temperature
- Testing other error correction codes
- Different placement of ring oscillators

Thank you for your attention