The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes

Daisuke Suzuki[1,2] and Koichi Shimizu[1]

1. Mitsubishi Electric Corporation
2. Yokohama National University

Outline

Background

- ✓Security chips and modules need to provide not only cryptographic functions but also tampering countermeasures.

- ✓Physical Unclonable Function (PUF) is a technique to counter tampering.

- ✓PUFs return responses to given challenges according to innate physical characteristics of artificial objects.

$$\begin{tikzcd} \text{(Challenge)} \arrow[r] & \boxed{\textbf{PUF}} \arrow[r] & \text{Response} \end{tikzcd}$$

■ PUFs known well

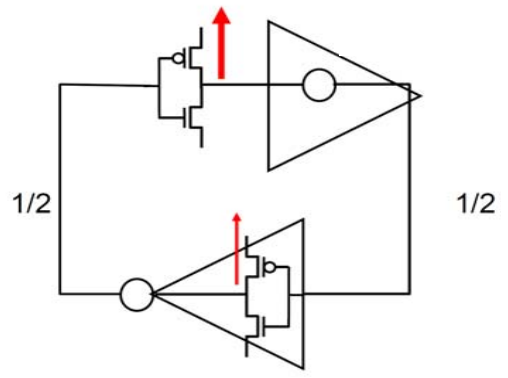
✓SRAM-PUF exploits start-up values of SRAM cells.

✓The values are a 'physical fingerprint' of a chip.

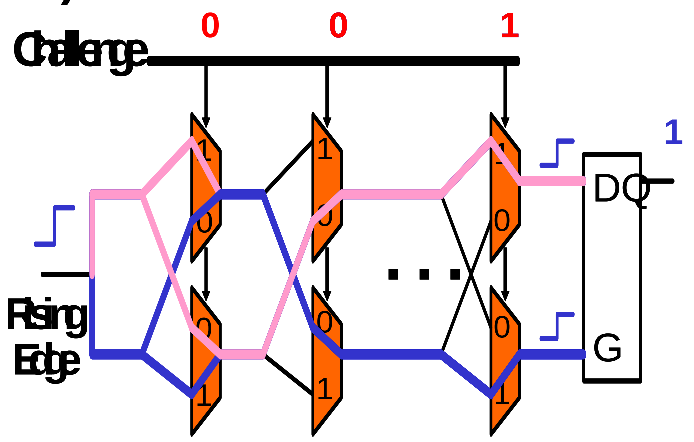
✓Delay-PUF **(e.g. Arbiter-PUF, Ring-oscillator PUF)** exploits random variation in delays of wires and gates.

✓Response data are produced by racing of gate delays.

SRAM-PUF
[CHES07]



Arbiter-PUF
[VLSI04]



■ Motivation

(M1） Need to evaluate information entropy and error rate of PUFs at design stage.

☹ It is usually difficult to obtain the physical characteristic of SRAM cells.

(M2) Need to generate unpredictable response data.

☹ Challenge-response pairs can be predicted in some delay-PUFs by machine learning attack after a decent number of pairs are collected.

■ Our contributions

(C1) We present a simple scheme to evaluate the characteristics of Delay-PUF with simulation at the design stage.

(C2) We propose a new Delay-PUF architecture: Glitch PUF

- ✓ The proposed architecture exploits glitch waveforms that behave non-linearly from delay variation between gates.
- ✓ We also present the results of the evaluation on randomness and statistical properties of Glitch PUF performed on FPGA and simulation

Outline

■ Delay variation in CMOS process

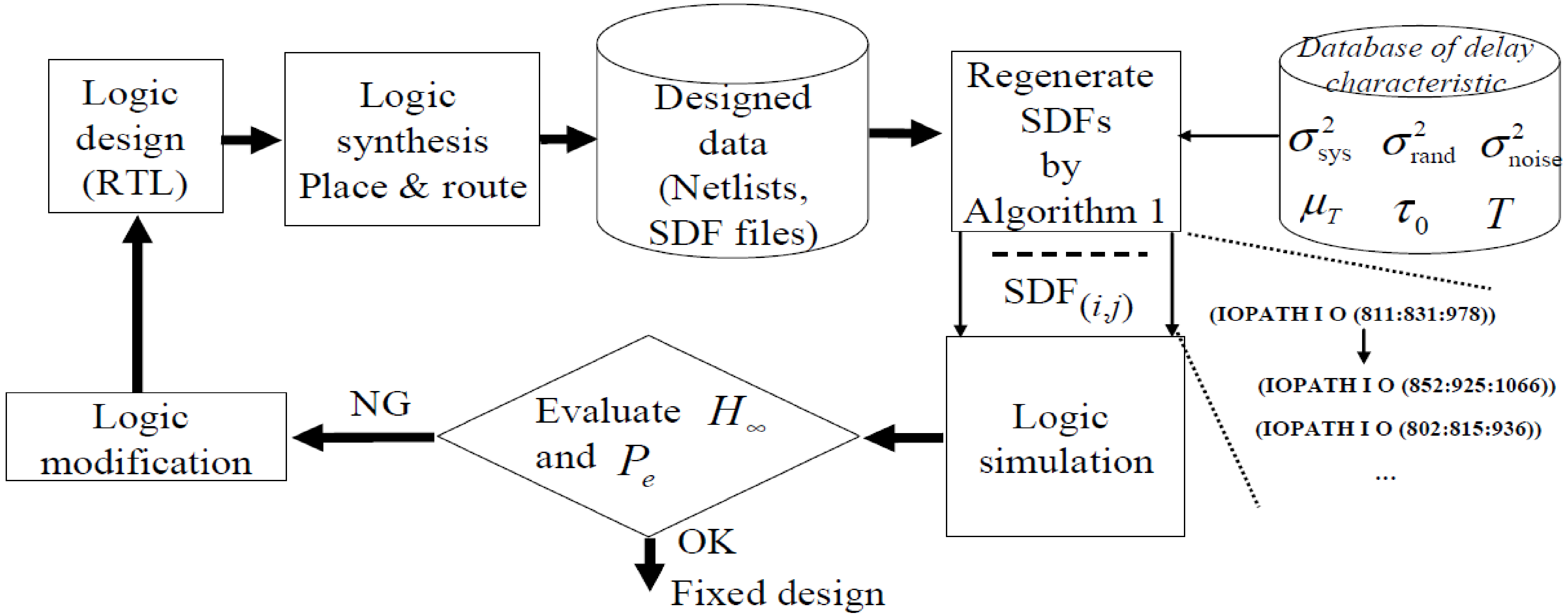
- ✓The timing analysis with delay variation, Statistical Static Timing Analysis (SSTA), has become popular over the past few years because random variation has increased with miniaturization in CMOS process.

- ✓It is anticipated from these facts that logic circuit designers will be able to access information about delay variation in a near future.

- ✓Delay-PUF is advantageous in that delay information utilized by it has affinity with logic simulation, which is performed at the design stage.

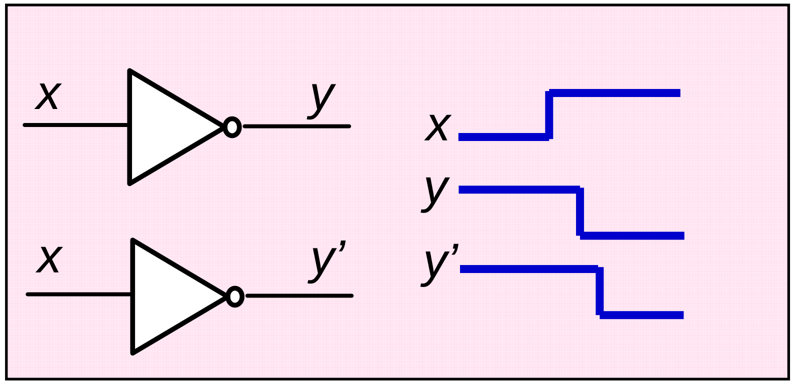
Delay-PUF simulation with variation

- Systematic delay variation between chips: σ^2_{sys}
- Random delay variation within chips: σ^2_{rand}
- Environmental random delay variation such as from dynamic IR-drop: σ^2_{noise}
- Average fraction of designed delays under 0 °C: τ_0
- Delay temperature coefficient: μ_T

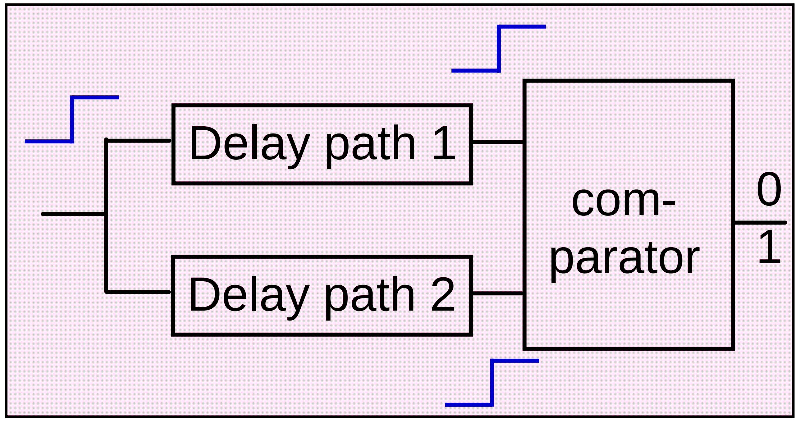

Logic design (RTL)
Logic synthesis Place & route
Designed data (Netlists, SDF files)
Regenerate SDFs by Algorithm 1
Database of delay characteristic
σ^2_{sys} σ^2_{rand} σ^2_{noise}
μ_T τ_0 T
$\text{SDF}_{(i,j)}$
(IOPATH I O (811:831:978))
(IOPATH I O (852:925:1066))
(IOPATH I O (802:815:936))
...
Logic simulation
Evaluate H_∞ and P_e
NG
Logic modification
OK
Fixed design

Outline

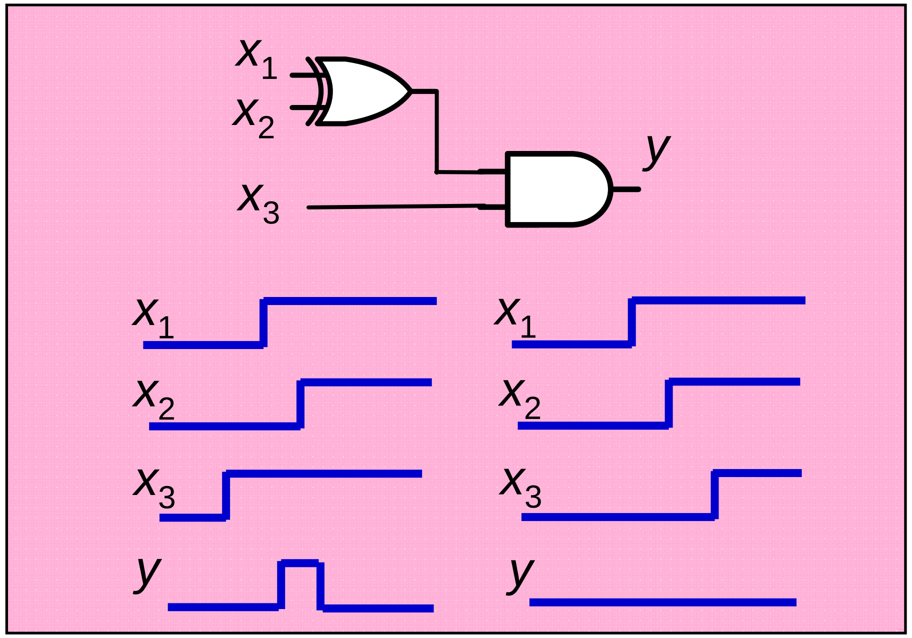
Glitch PUF: basic idea



Depending greatly on temperature and voltage



The relation between CRPs and path delays is almost linear.



The relation between glitch waveforms and path delays is non-linear.

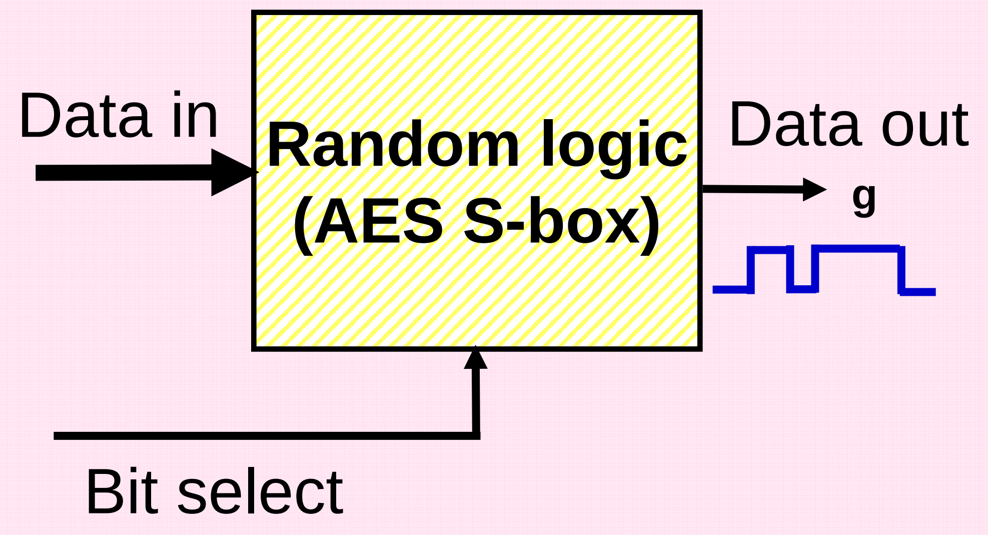
Can we use this phenomenon for PUF?

■ Glitch PUF: overall sequence

STEP 1:
Data input to a random logic

STEP 2:
Acquisition of glitch waveforms at the output

STEP 3:
Conversion of the waveforms into response bits

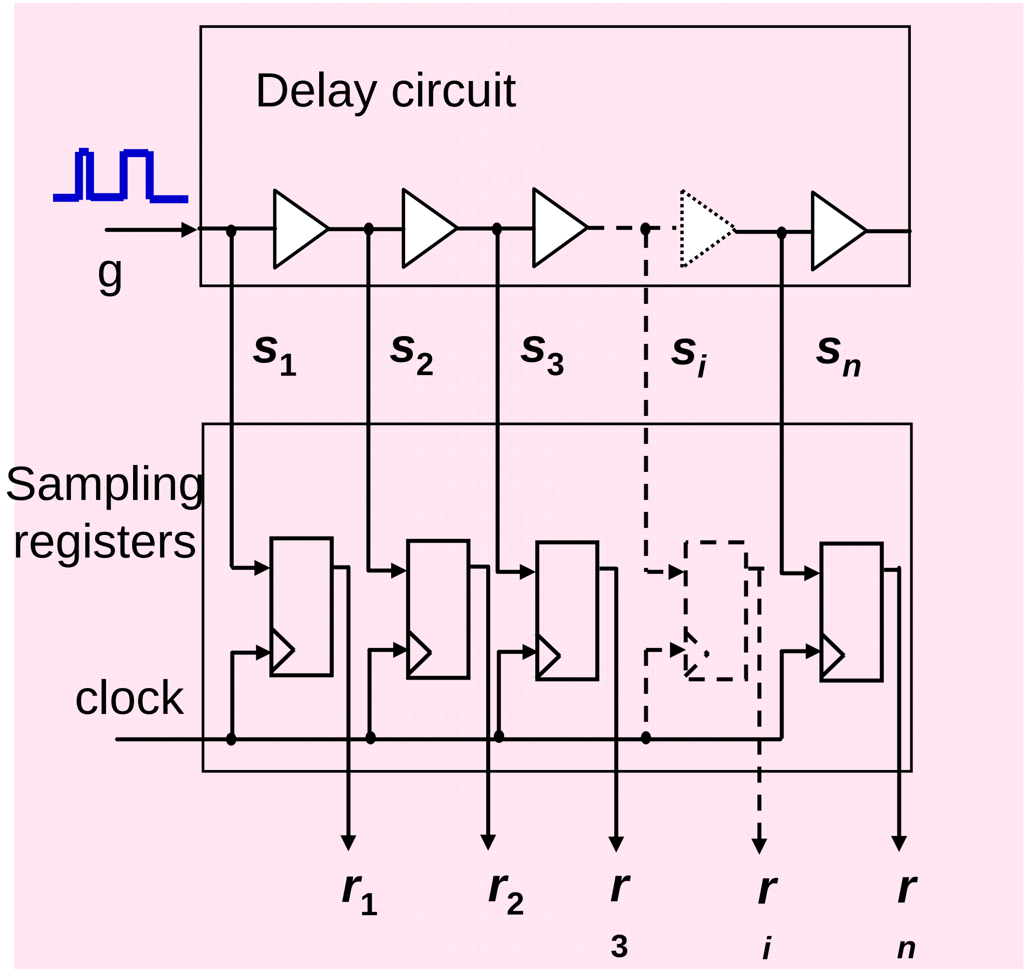


Glitch PUF: overall sequence

STEP 1:
Data input to a random logic

STEP 2:
Acquisition of glitch waveforms at the output

STEP 3:
Conversion of the waveforms into response bits

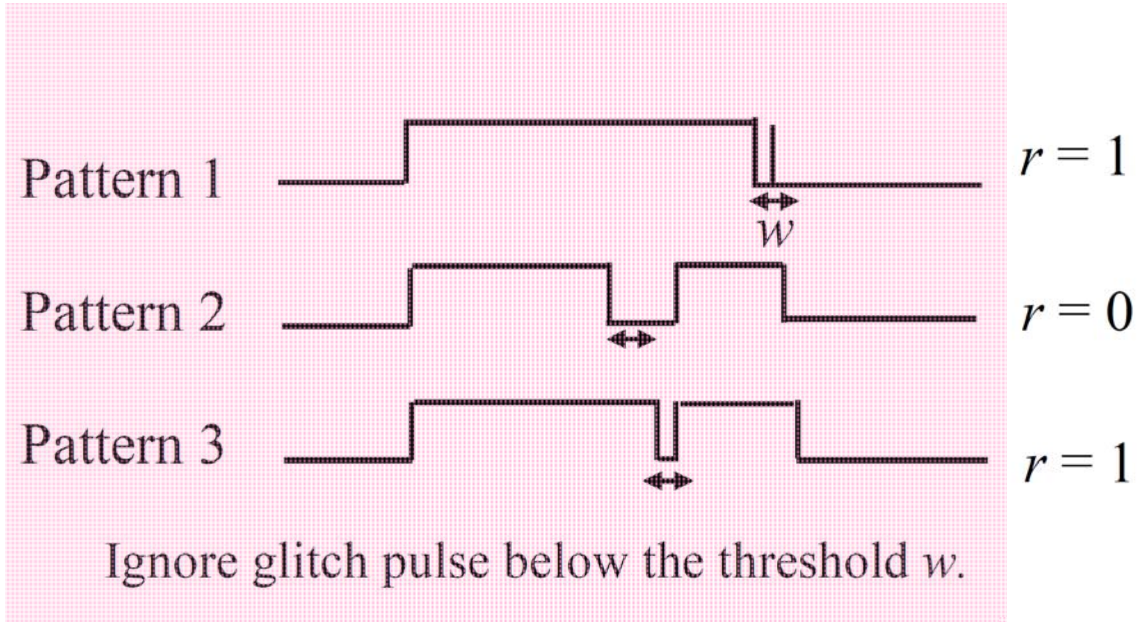


Glitch PUF: overall sequence

STEP 1:
Data input to a random logic

STEP 2:
Acquisition of glitch waveforms at the output

STEP 3:
Conversion of the waveforms into response bits

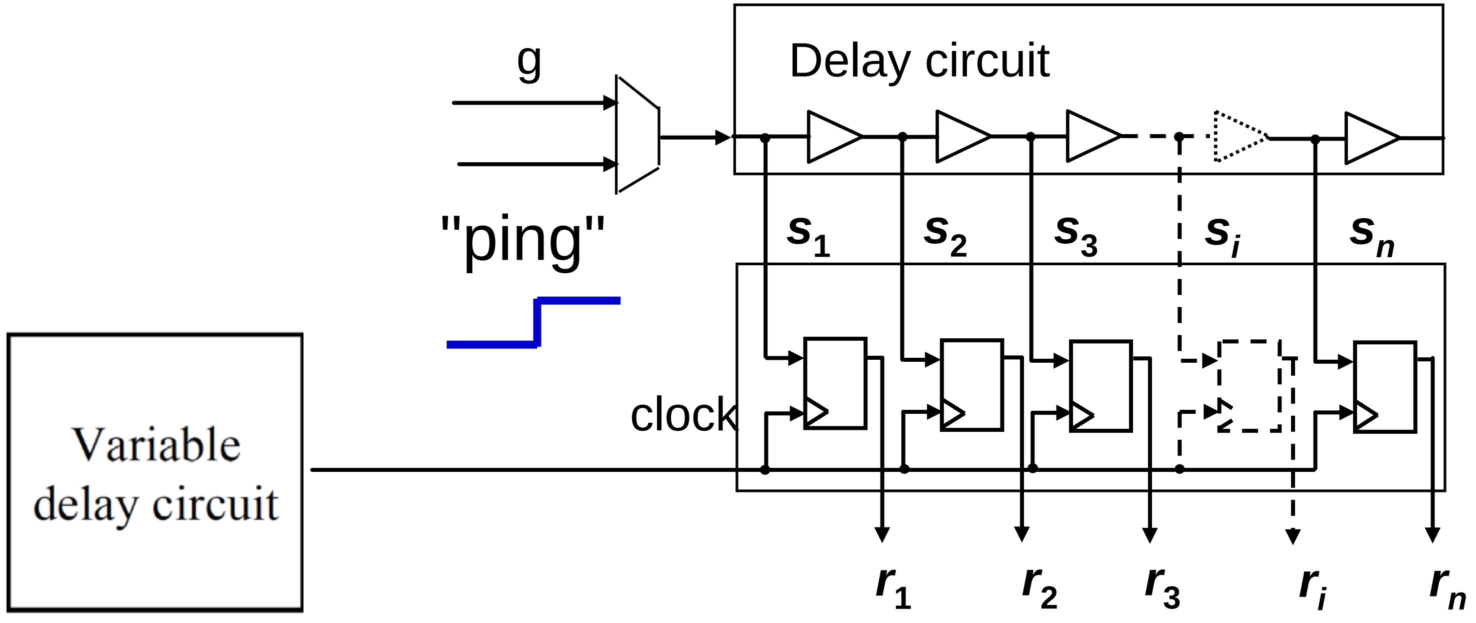


***r* = (count(posedge)) mod 2**

Glitch PUF: solution for main problems

✓How to acquire glitch waveforms correctly

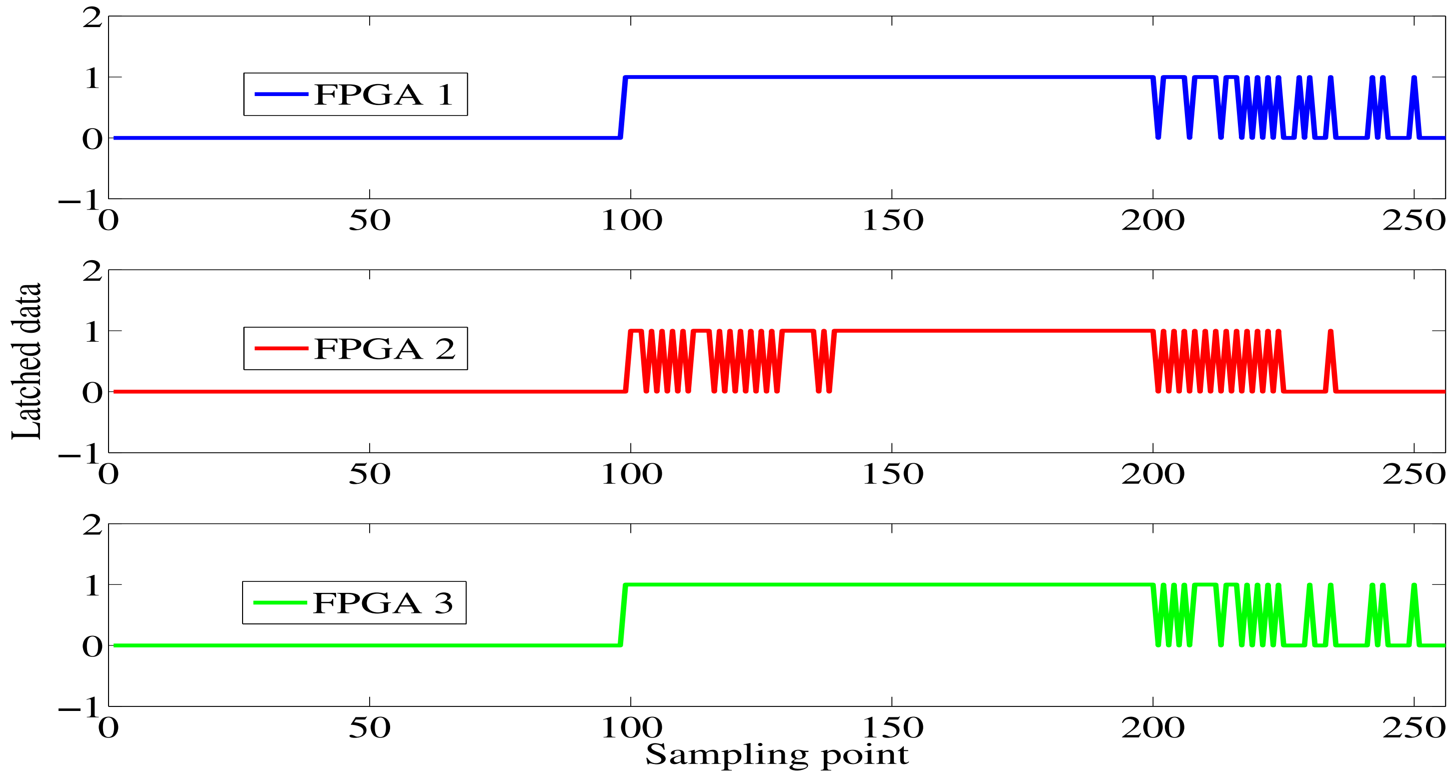
1. Re-order the results of the sampling FFs based on trial sampling of a "ping" signal



Glitch PUF: solution for main problems

✓How to acquire glitch waveforms correctly

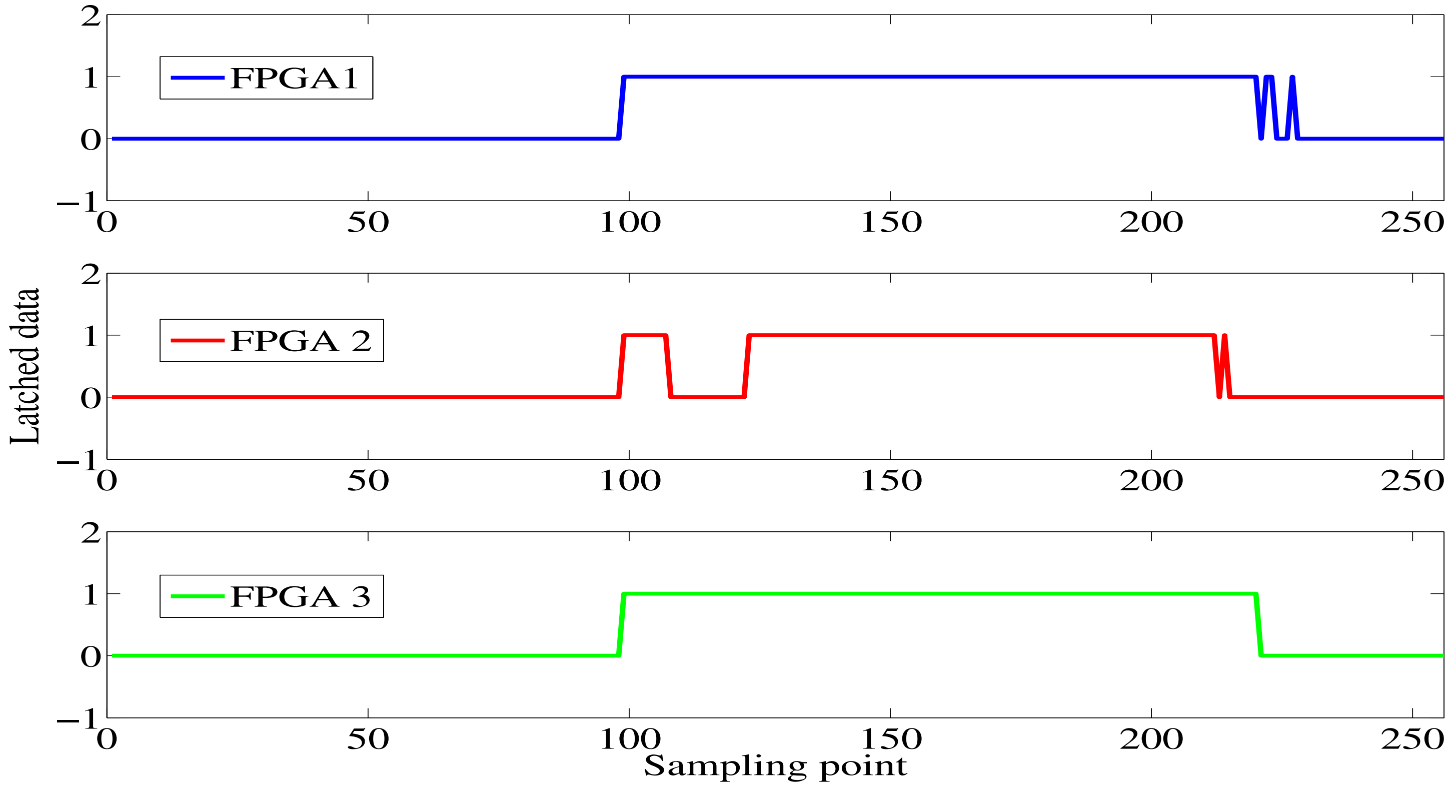
Before re-order



■ Glitch PUF: solution for main problems

✓How to acquire glitch waveforms correctly

After re-order



Glitch PUF: solution for main problems

✓How to acquire glitch waveforms correctly

1. Re-order the results of the sampling FFs based on the trial sampling of a "ping" signal

2. Eliminate transitions with very small pulse width

Glitch PUF: solution for main problems

✓How to acquire glitch waveforms correctly

1. Re-order the results of the sampling FFs based on the trial sampling of a "ping" signal
2. Eliminate transitions with very small pulse width
3. Repeat acquisition for the same input and revoke unstable CRPs (masking) in enrollment phase.

$$\begin{array}{c|cccc} R^{(0)} & 0 & 0 & 1 & 1 \\ R^{(1)} & 0 & 1 & 1 & 1 \\ R^{(2)} & 0 & 1 & 1 & 0 \end{array}$$

🔵 stable

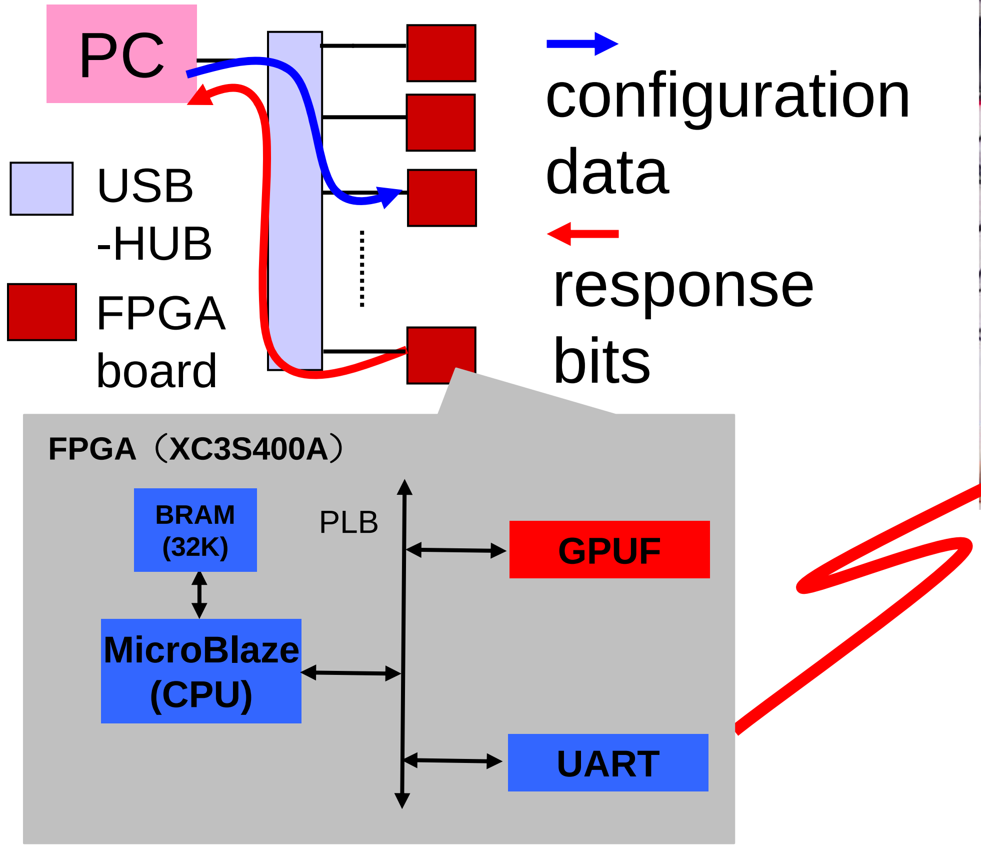
🔴 unstable

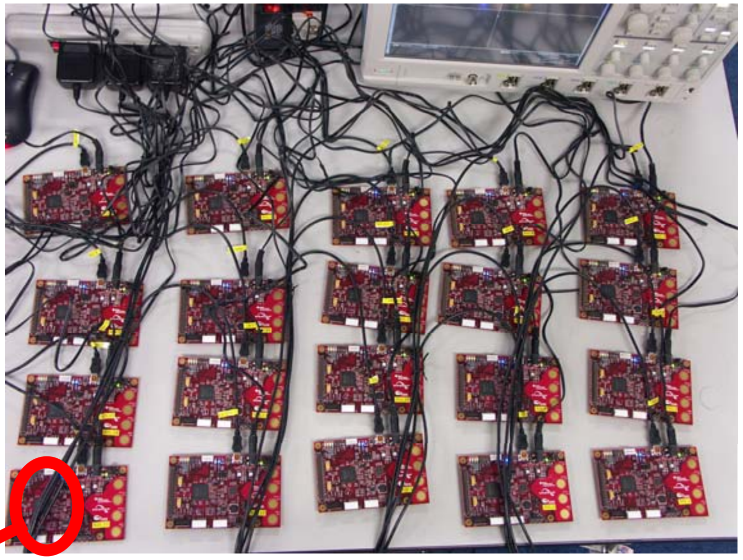
MASK 1 0 1 0

Outline

Experimental environment

✓ We evaluate delay characteristics, H_∞ and P_e using 16 FPGA boards

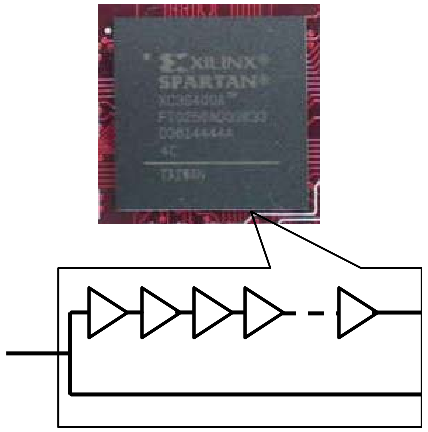




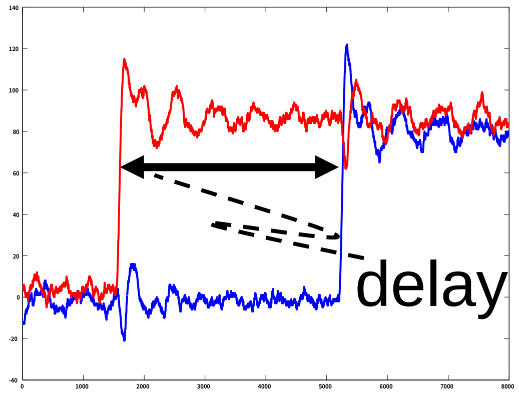
SLICEs used 891/3,584 (Whole SoC 3,186/3,584)

Process for extracting delay parameters

- ✓Delay variation is measured by using 1000 layouts for each FPGA.
- ✓Simulation parameters are calculated as fractions of corresponding worst-case delays defined in SDF generated by the EDA tool after the layout.



Buffer chain

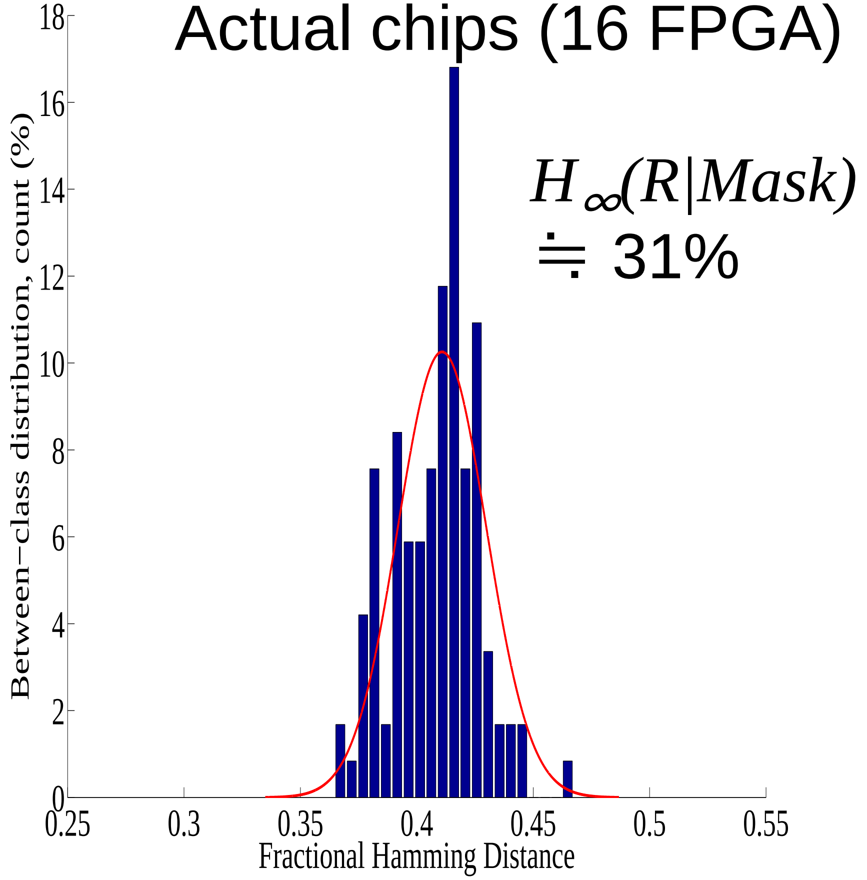


Systematic delay variation $\sigma^2_{\text{sys}}(\%^2)$	2.5037
Random delay variation $\sigma^2_{\text{rand}}(\%^2)$	5.3091
Environmental random delay variation $\sigma^2_{\text{noise}}(\%^2)$	0.0310
Average fraction of designed delays τ_0 (%)	56.9727
Delay temperature coefficient μ_T (%/°C)	0.1401

■ Histogram of inter-class

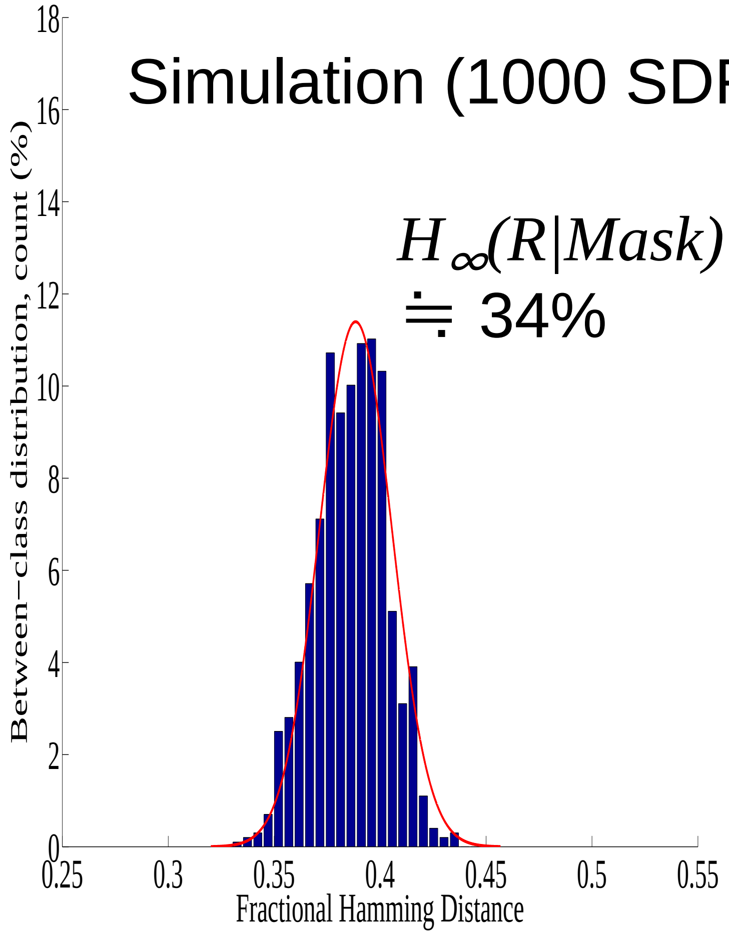
✓2048bit response bits with mask
at normal temperature and voltage (24℃, 1.20V)

Actual chips (16 FPGA)



$H_\infty(R|Mask)$
≒ 31%

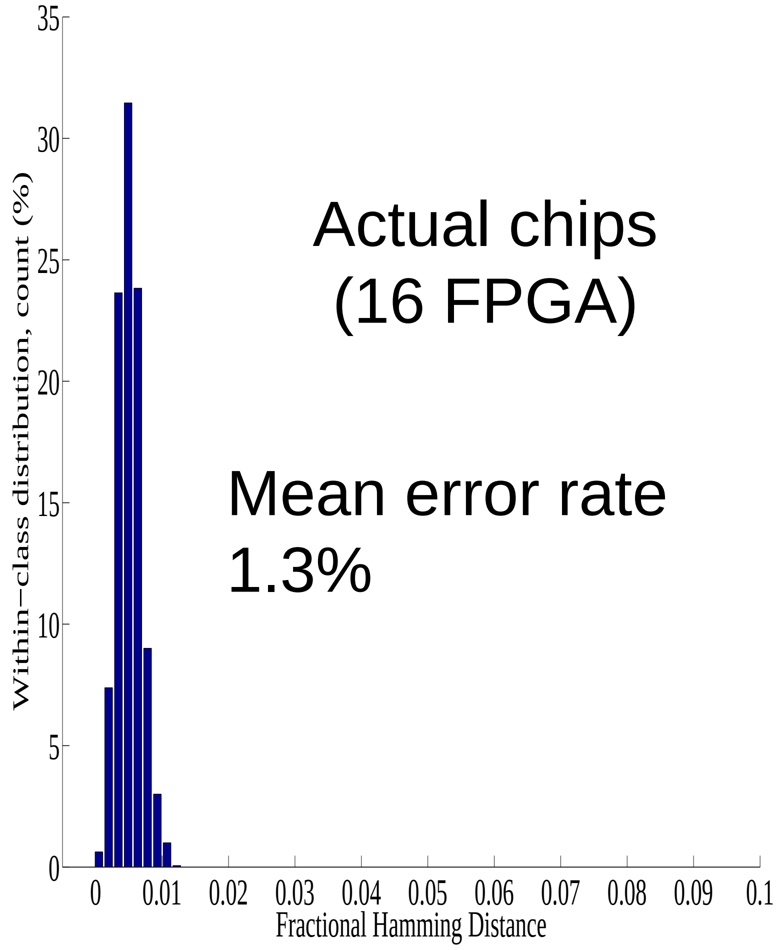
Simulation (1000 SDFs)

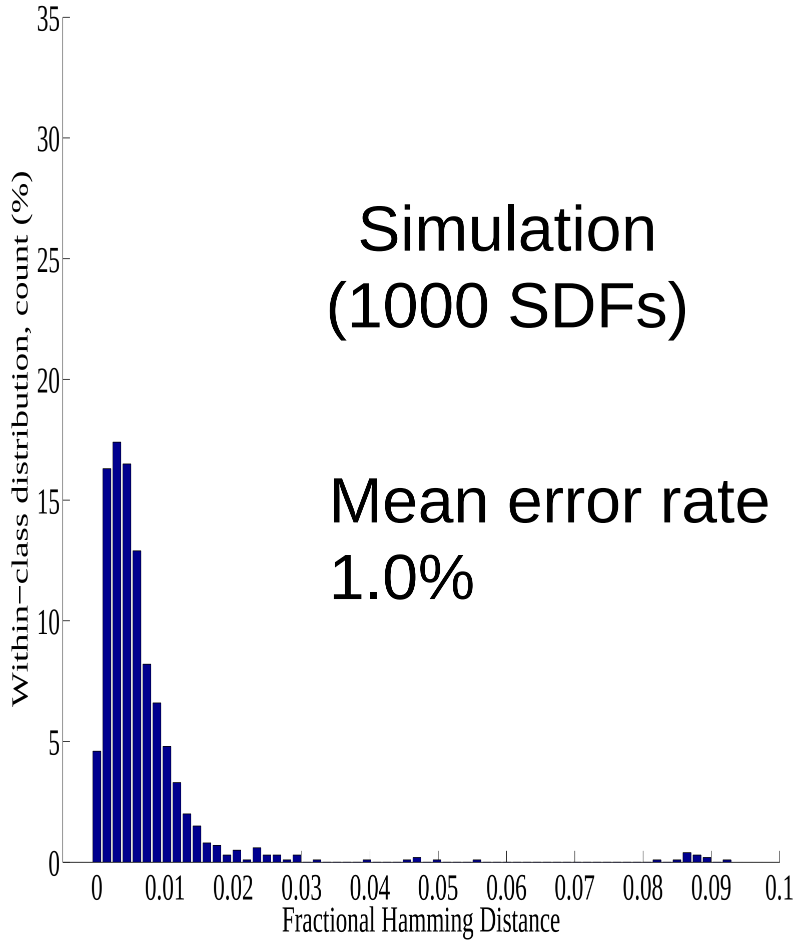


$H_\infty(R|Mask)$
≒ 34%

Histogram of intra-class

✓2048bit response bits with mask
at normal temperature and voltage (24℃, 1.20*V*)

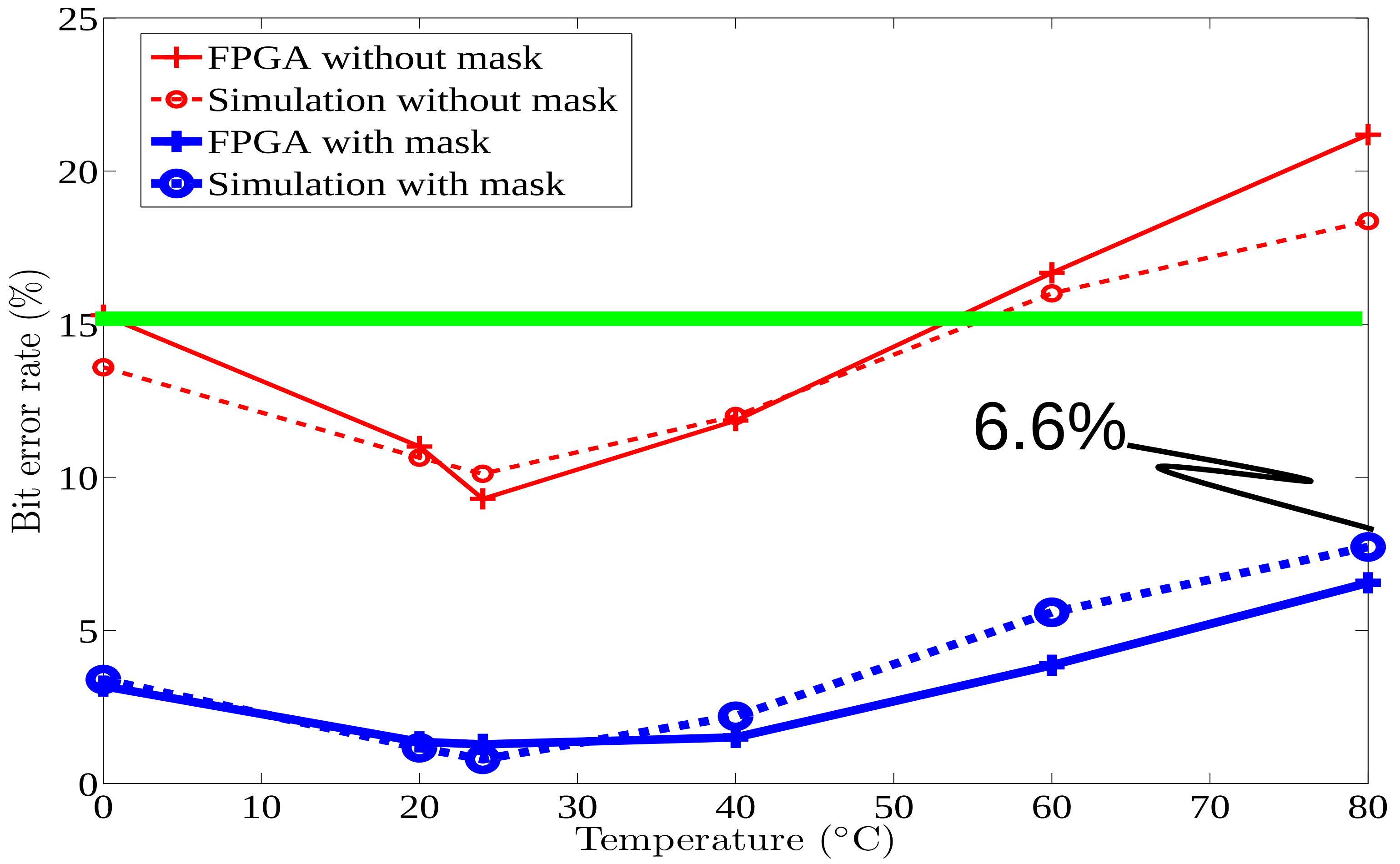




MITSUBISHI ELECTRIC

Changes for the Better

Change of error rate with change of temperature



■ Change of information amount against change of variations

✓Fixed random delay variation within chips: σ^2_{rand}

$$\left(\frac{1}{2}\sigma_{\text{sys}}\right)^2 \longrightarrow H_\infty(R|Mask) \fallingdotseq 34\%$$

$$\sigma^2_{\text{sys}} \longrightarrow H_\infty(R|Mask) \fallingdotseq 34\%$$

$$(2\cdot\sigma_{\text{sys}})^2 \longrightarrow H_\infty(R|Mask) \fallingdotseq 35\%$$

✓Minor influence by Systematic variation between chips

■ Change of information amount against change of variations

✓Fixed systematic delay variation between chips: ${\sigma^2}_{\text{sys}}$

$$(\frac{1}{2}\sigma_{\text{rand}})^2 \longrightarrow H_\infty(R|Mask) \fallingdotseq 26\%$$

$${\sigma^2}_{\text{rand}} \longrightarrow H_\infty(R|Mask) \fallingdotseq 34\%$$

$$(2\cdot\sigma_{\text{rand}})^2 \longrightarrow H_\infty(R|Mask) \fallingdotseq 40\%$$

✓Major influence by Random variation within a chip

Outline

Conclusions

✓ We proposed a concrete structure of Delay-PUF exploiting glitch waveforms.

Secrecy rate $I(R,R')$: 26% (SRAM-PUF 76% [CHES2007])

☹ This is about 1/3 that of SRAM-PUF.

☺ We can evaluate the secrecy rate by logic simulation.

In the future (many problems remain…)

- Construct a glitch generator that brings high amount of information and low error rate.
- Model machine learning attacks to Glitch PUF.
- Model logic simulation for voltage change and aging degradation through acceleration test, and evaluate them on real chips

Thanks for listening