# A PUF Design for Secure FPGA-Based Embedded Systems

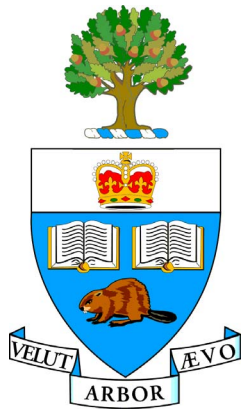
Jason H. Anderson

Dept. of Electrical and Computer Engineering
University of Toronto

IEEE/ACM Asia and South Pacific Design Automation Conference

Taipei, Taiwan

January 18-21, 2010



# Two Honda Civics

- Same year, same model, same colour, but are they truly IDENTICAL?
    - Of course not, and likewise, two chips have RANDOM variations that make their delays slightly different from one another.

# Motivation

- PUF: Physically Unclonable Function.
- Process variations make fabricated chips different from one another.
- Use the variations to generate a unique multi-bit signature for a fabricated IC.
  - Silicon “biometric”; “DNA-like”.
  - Variations uncontrollable $\rightarrow$ unclonable.
- FPGAs commonly used in embedded systems.

# Application: Chip Anti-Counterfeiting

- Fab-less semiconductor design model:
  - Chip research and design by company **A**
  - Chip manufacturing by company **B**

- Why?
  - State-of-the-art chip manufacturing facility costs several billion $.
  - Share cost across many design houses.

# Counterfeiting Risk

Company A Employee

Here's our design, please manufacture 10,000 chips!

Company B Employee (Chip Manufacturing)

Yes, sir! 10,000 on the way!

Workers! Build 10,000 for company A and build another 5,000 for us to sell!

# Detecting Counterfeiting

- How can Company A tell if some chip is one of its own?
- If each chip had it's own unique "DNA"-like signature → can do "DNA test".
- But! Don't want Company B to insert "fake" DNA into the counterfeits to make them look like the real ones.
  - Want "unclonable" DNA → PUF.

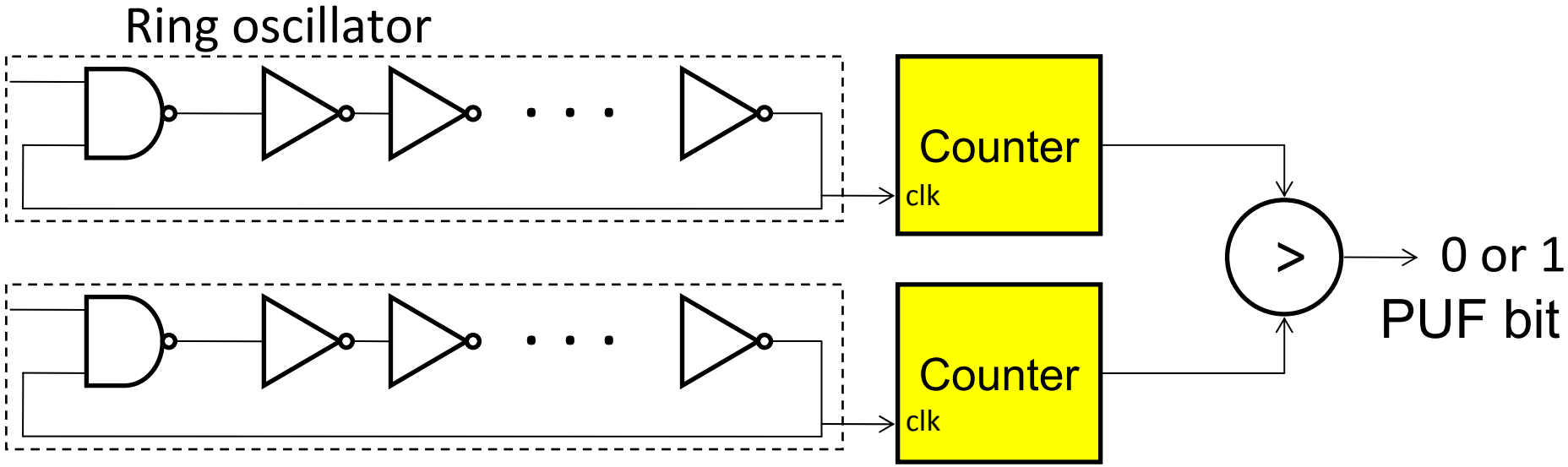
# PUF Applications

- Anti-counterfeiting
- IP protection
- Cryptography
- FPGA-specific applications
  - Chip authentication for bitstream loading

# Related Work

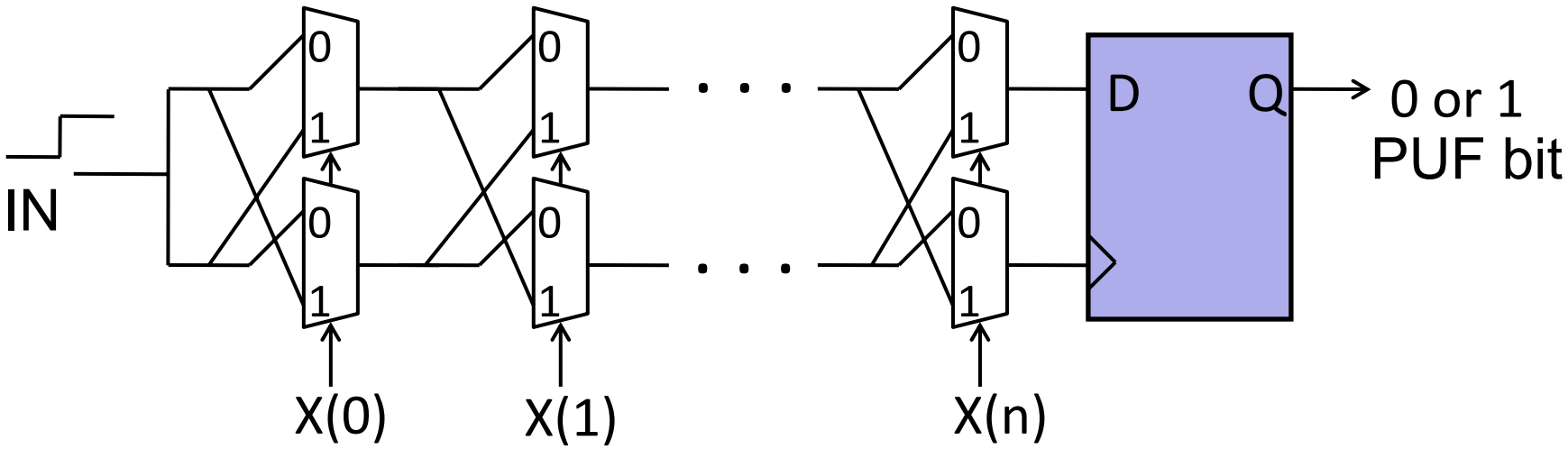
- Concept pioneered at MIT [Devadas'02].
- Several PUF designs proposed:
  - Ring oscillator PUF.
  - Butterfly PUF.
  - SRAM PUF.
- Verayo: start-up founded in 2005.

# Related Work




- Oscillator frequency depends on process variations.
- Compare oscillator frequencies to produce PUF bit.
- Ring oscillators must be **exactly matched** from logic/routing perspective.

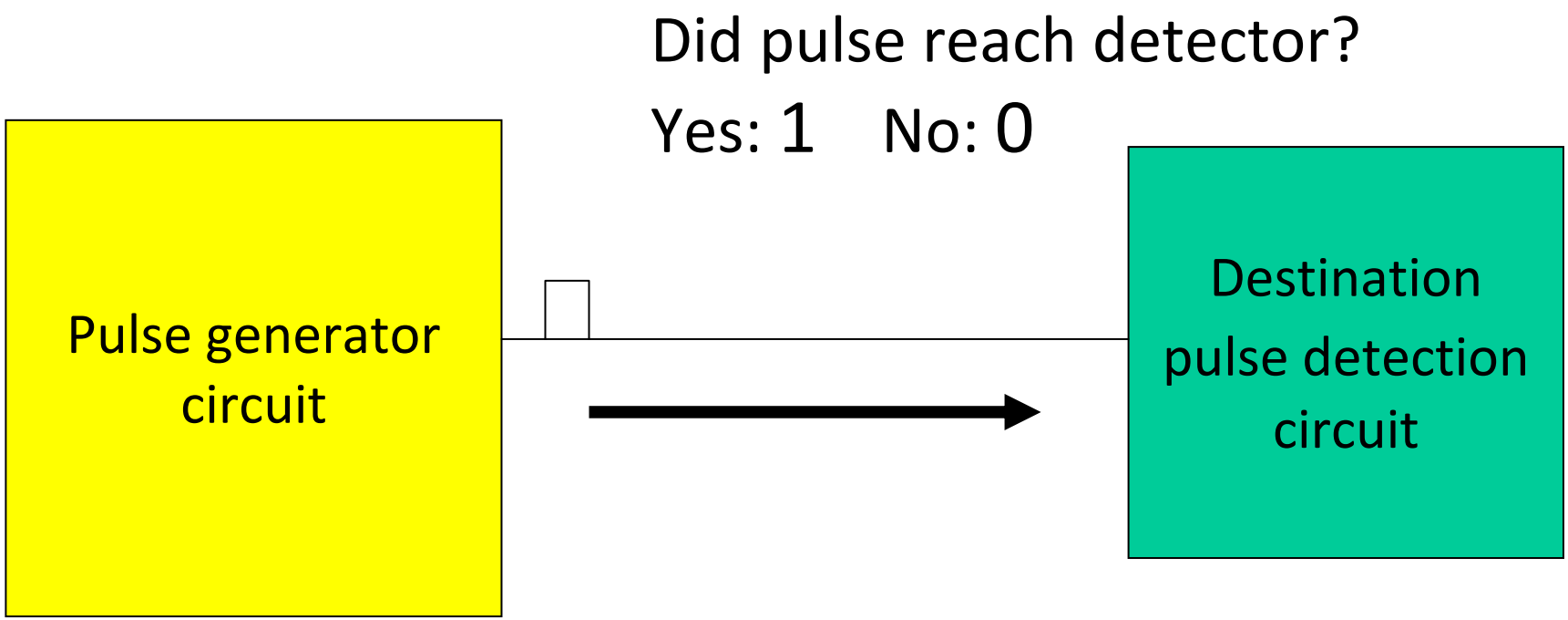
# Related Work




**Arbiter PUF**

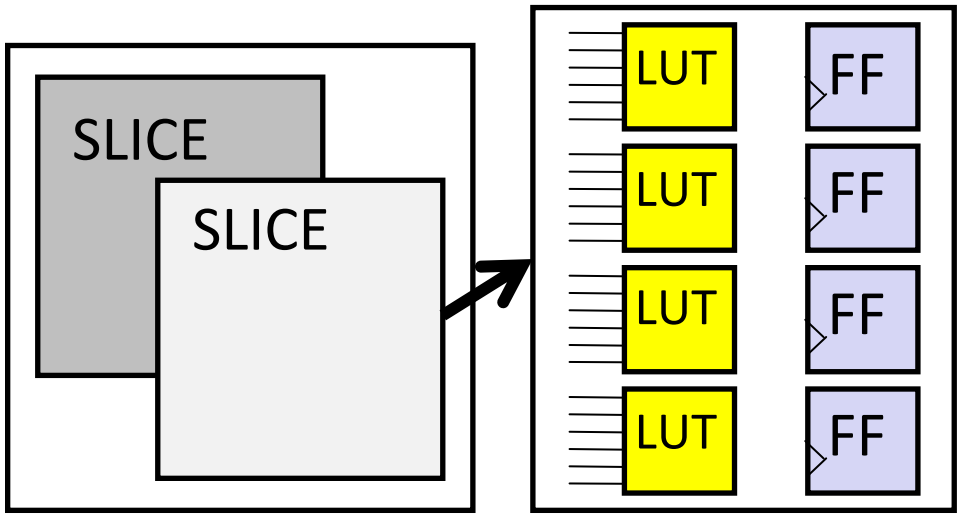
- Apply “challenge” word X(0) … X(n).
- Apply step input on IN.
- MUXes and paths must be **exactly matched**.

# Generating 1-Bit of PUF Signature



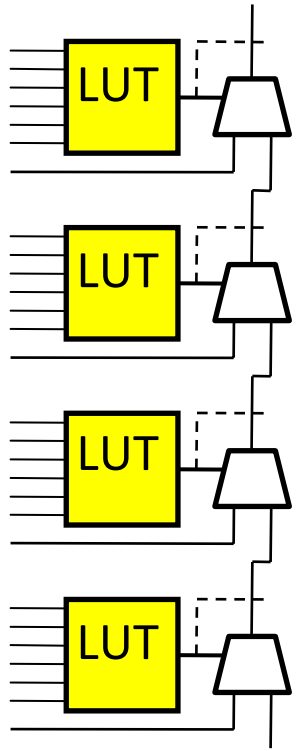

1) Generate a very short pulse (nanoseconds wide)
   -- pulse width unknown (due to manufacturing)
2) Send pulse along wire towards detection circuit

# Target: Xilinx Virtex-5 FPGA



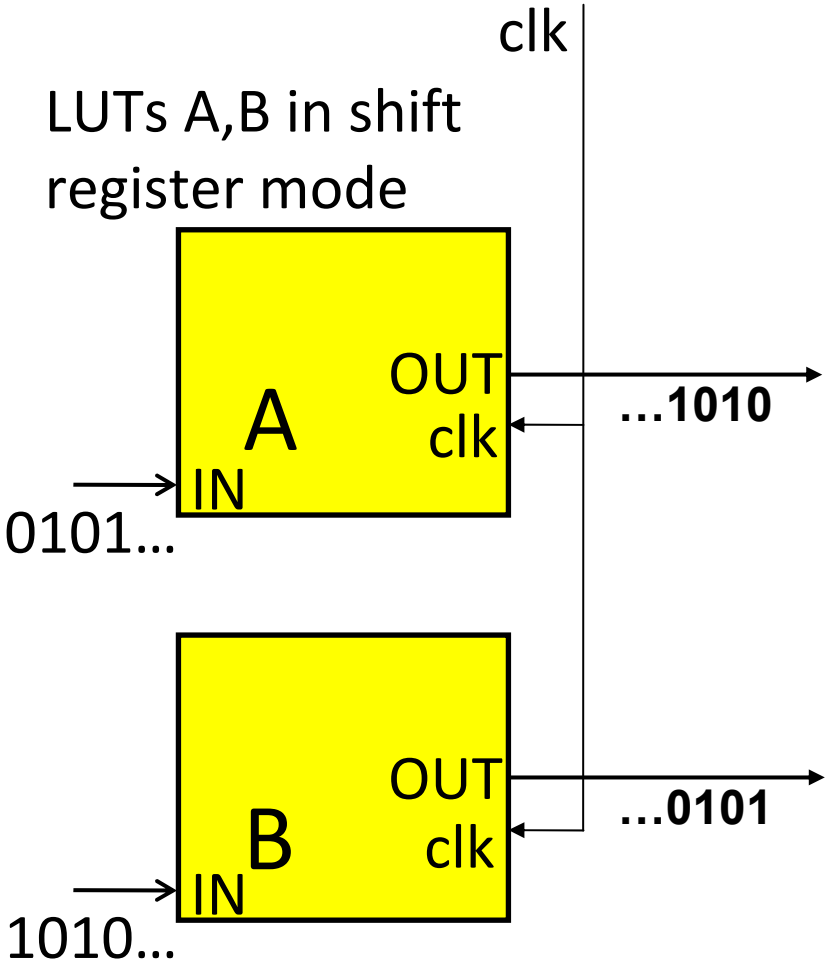

Configurable logic block (CLB)

SLICE

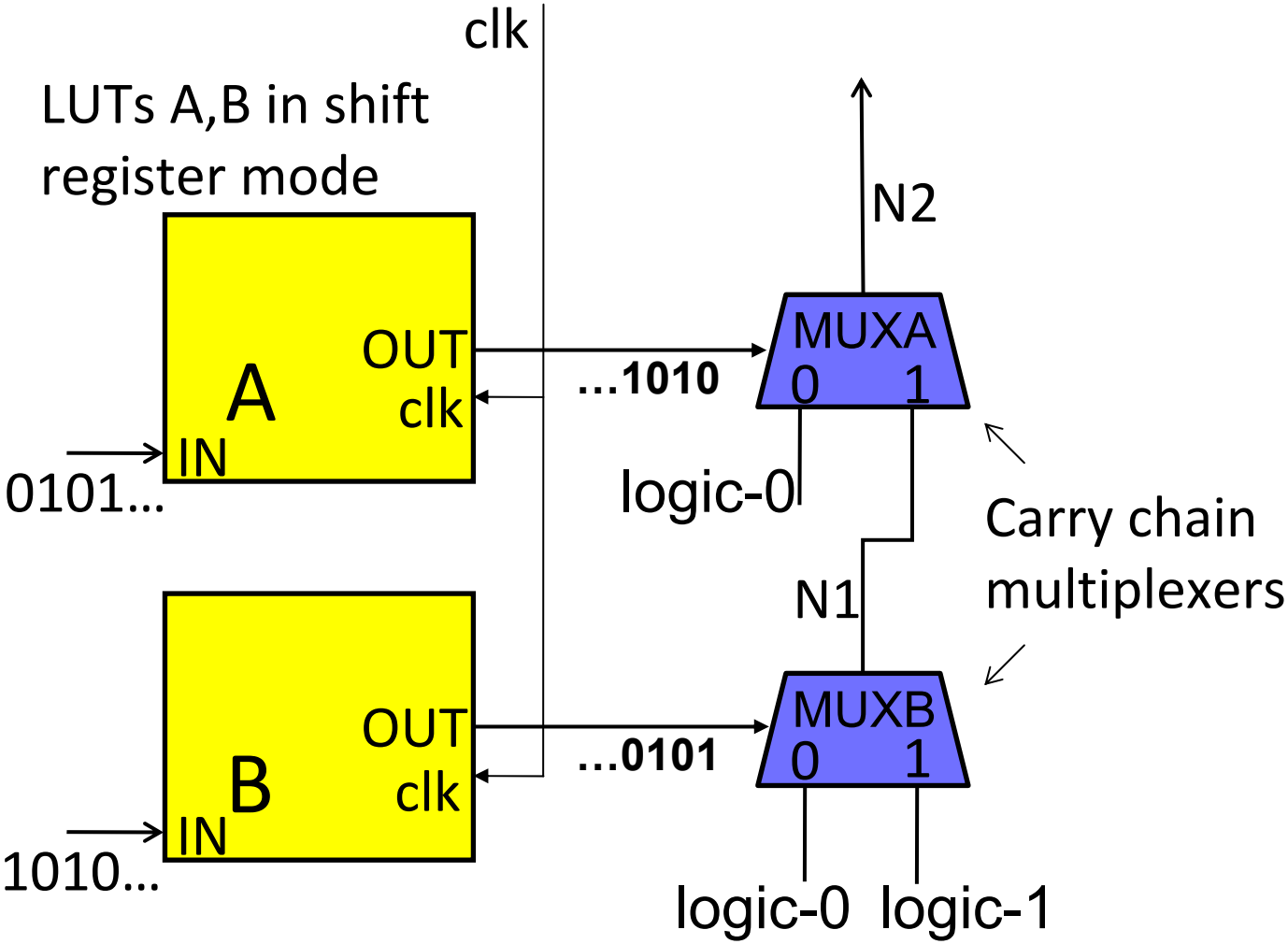



SLICE details

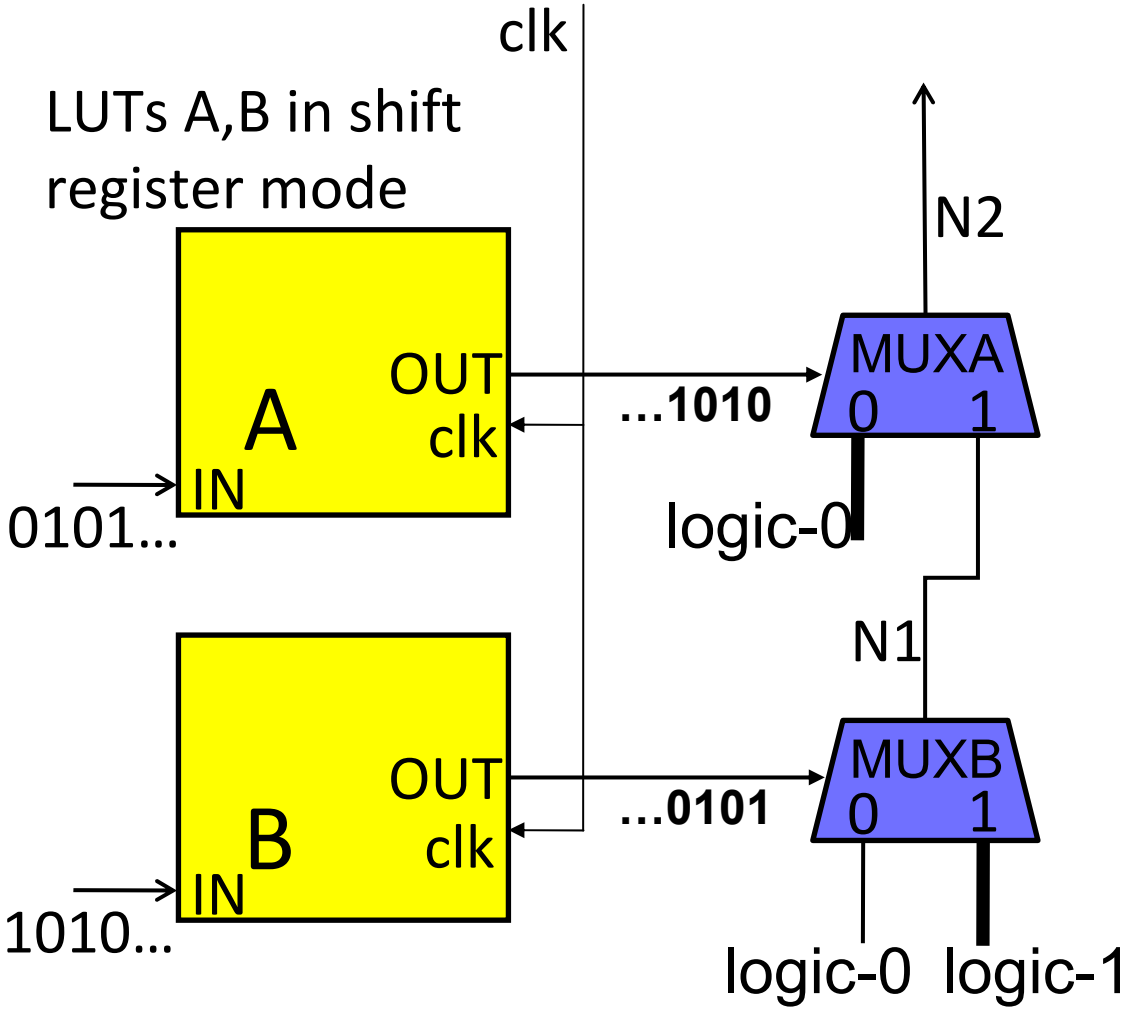
# How to Create A Short Pulse?



# How to Create A Short Pulse?

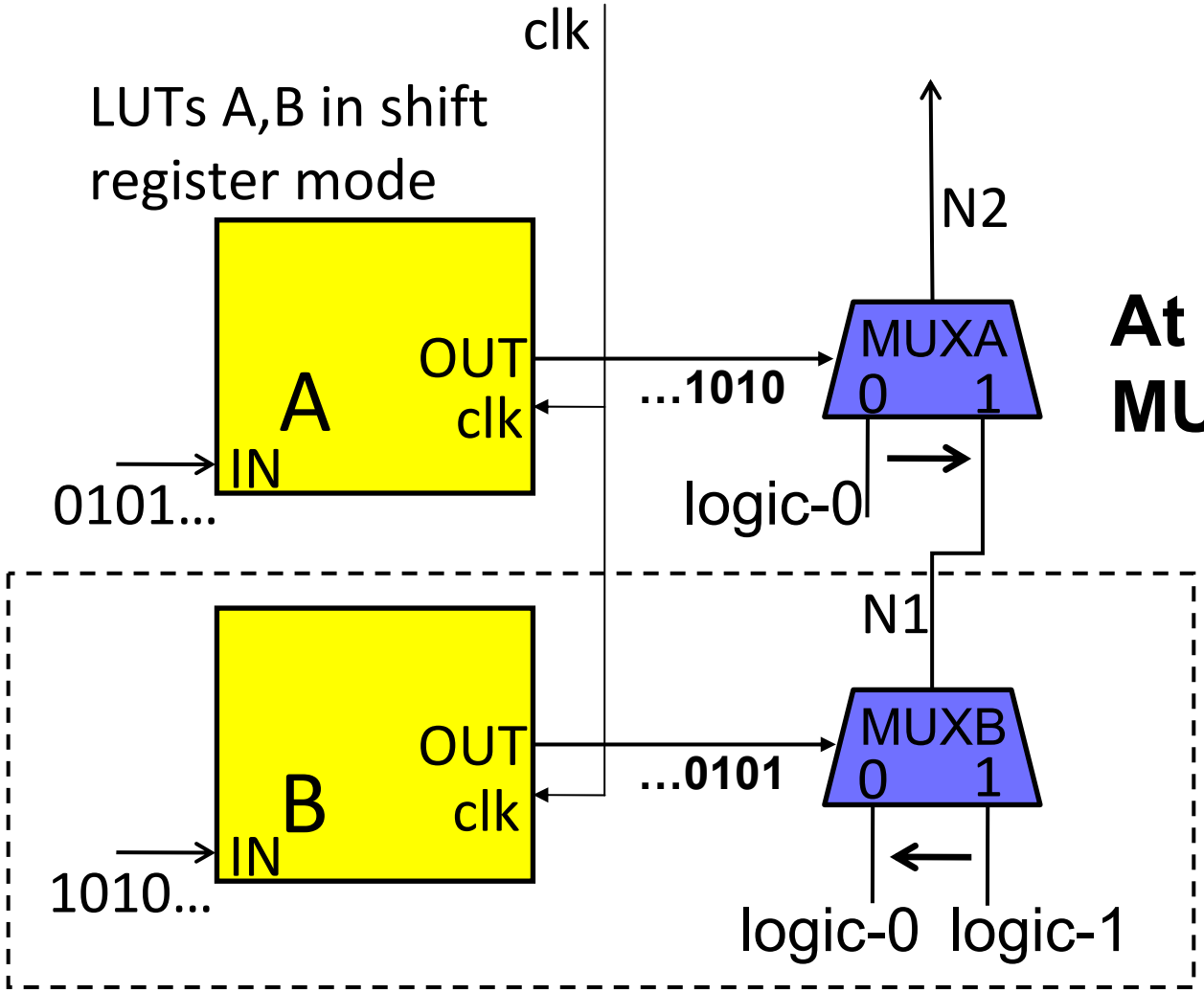


# How to Create A Short Pulse?




**Initially:**
**N2: logic-0**
**N1: logic-1**

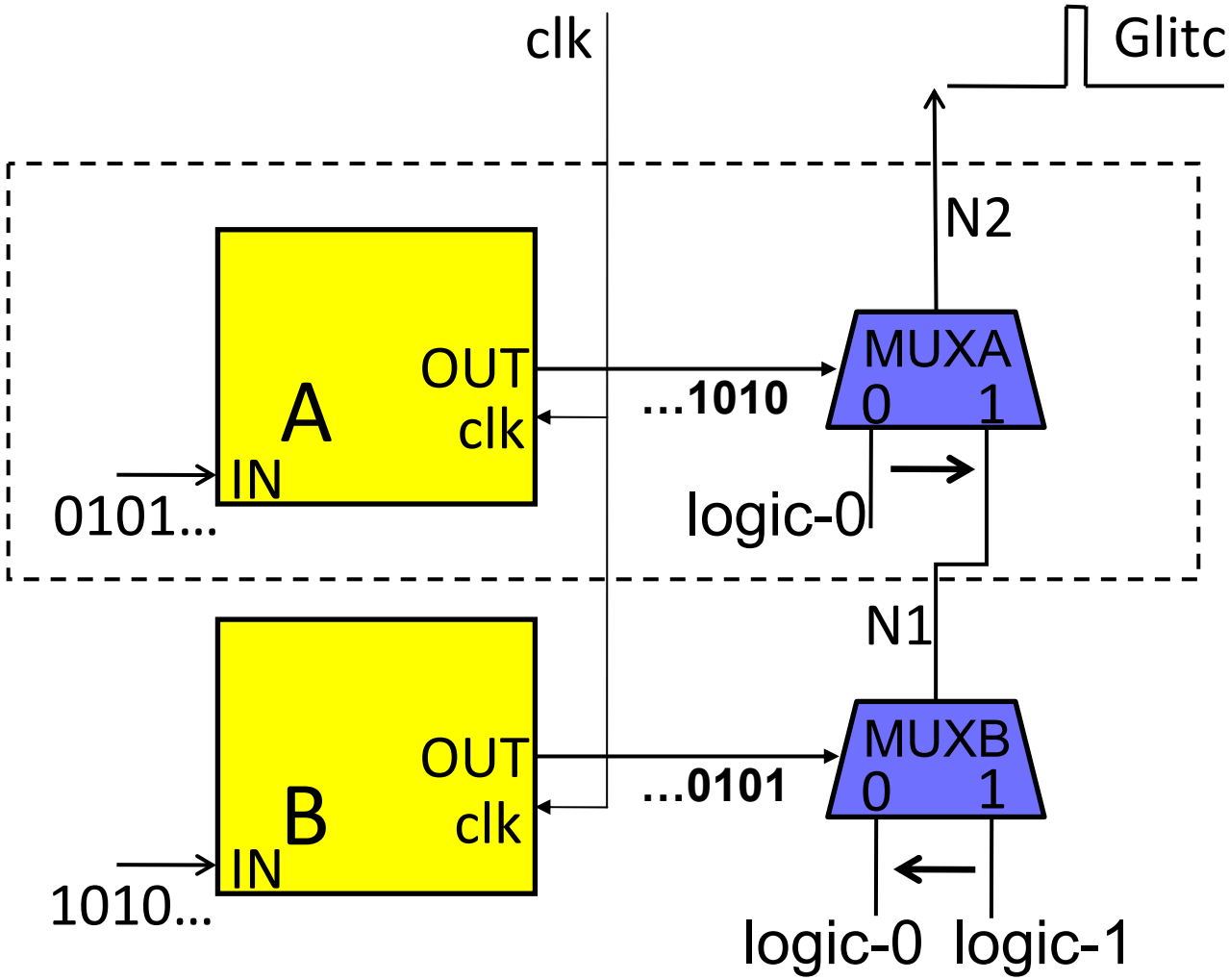
# How to Create A Short Pulse?




**At clock edge: MUX selects toggle**

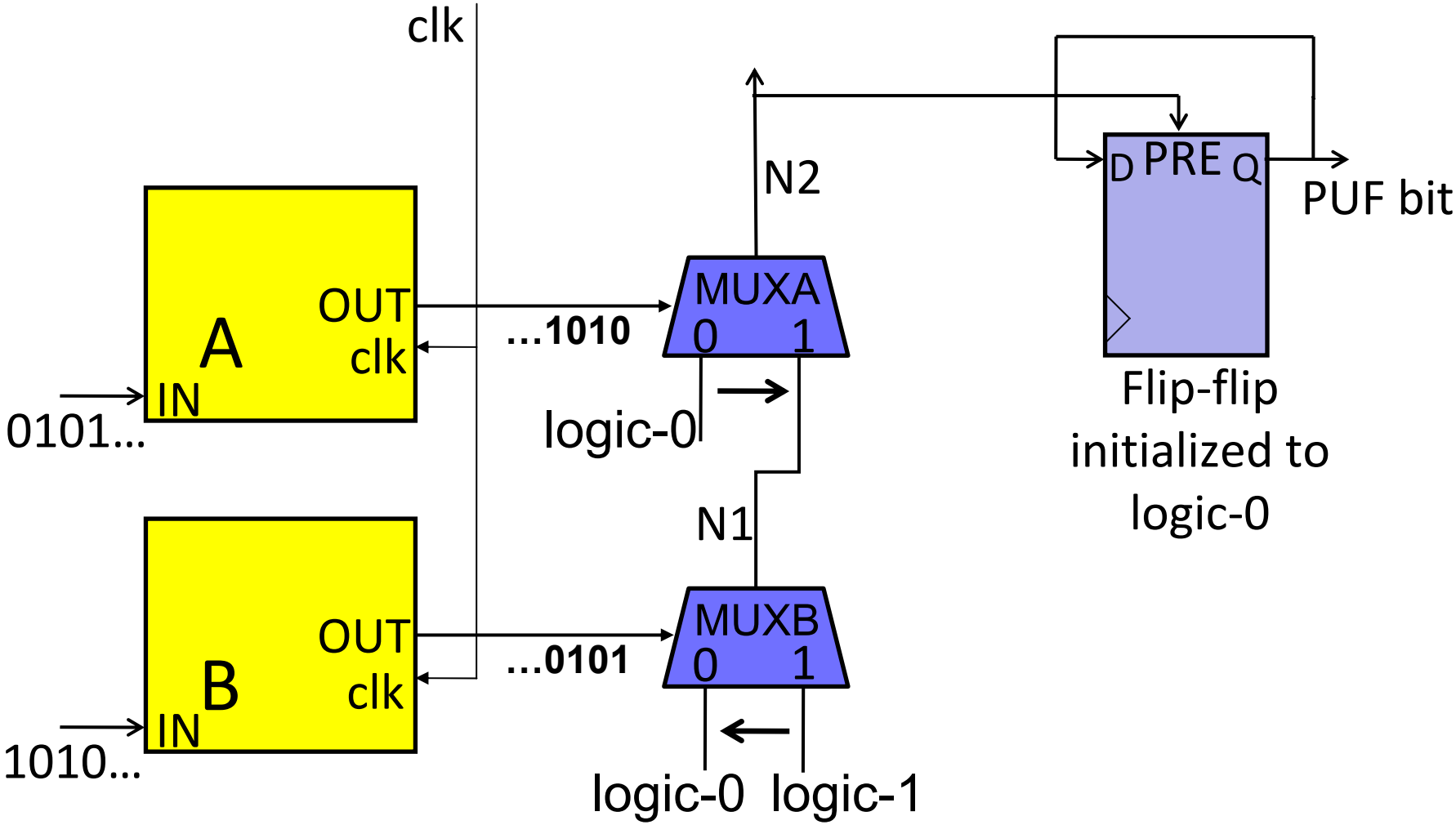
**Faster?**
**N1 → 0 before MUXA changes → N2 held @ 0**

# How to Create A Short Pulse?



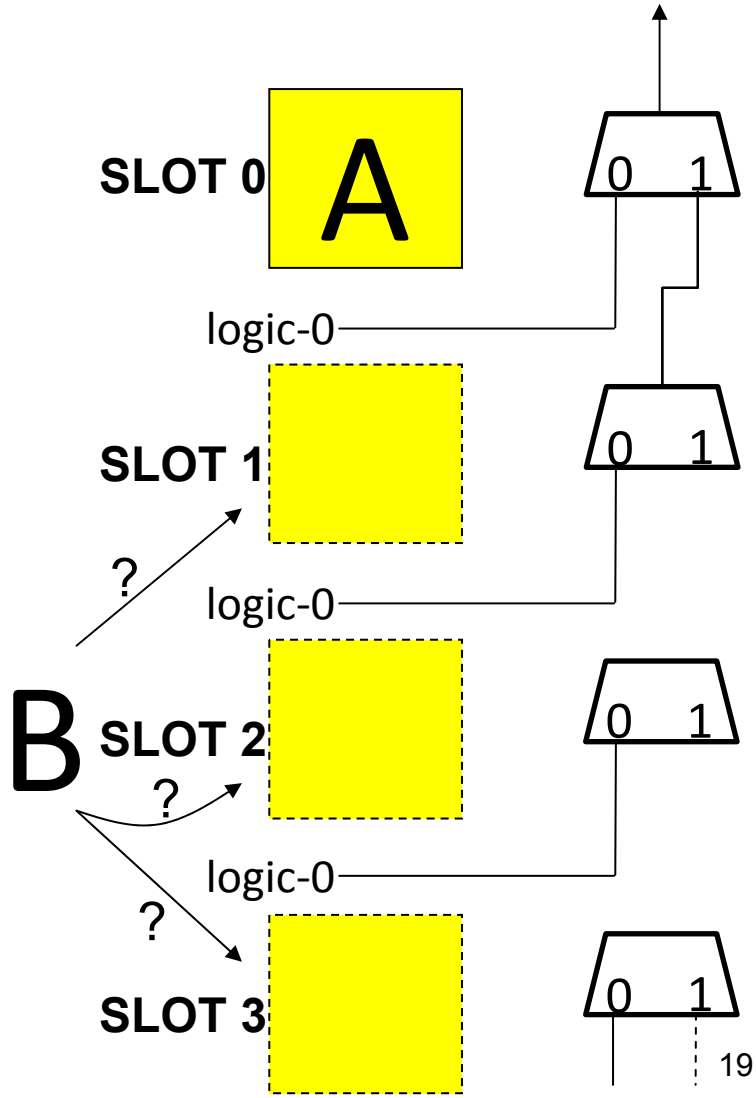

**Faster?**
**MUXA changes before N1 → pulse on N2.**

# How to Create A Short Pulse?



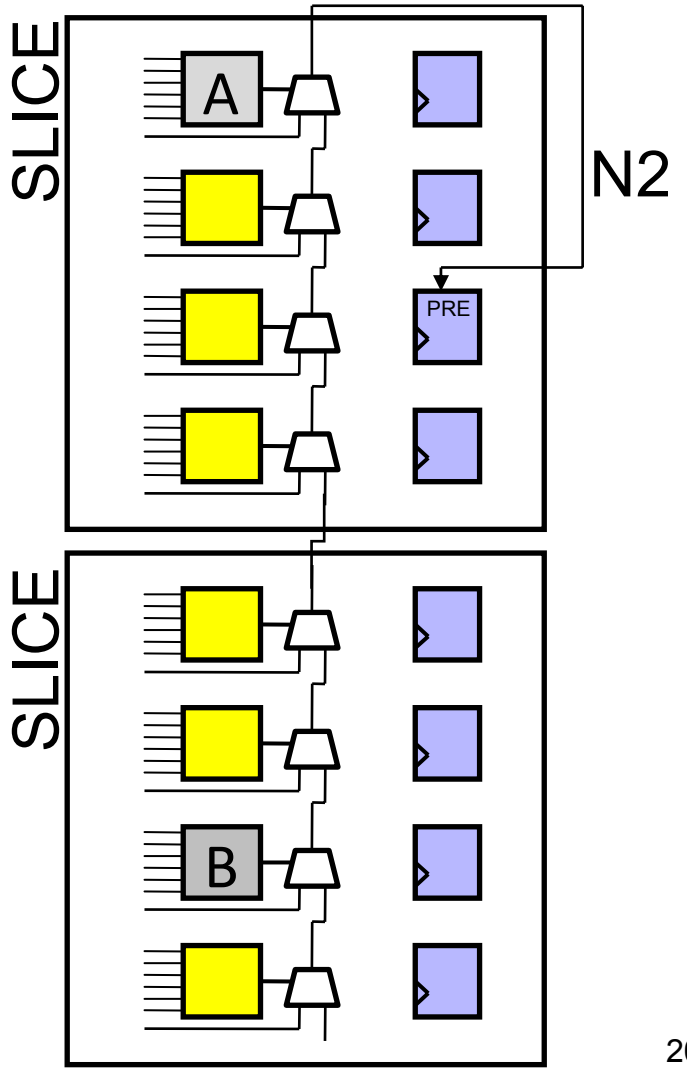
# Tuning the Pulse Width

- Ideally PUF bits 0 or 1 with equal probability.
- Problem: pulse too short and filtered out by RC nature of interconnect → many bits are 0.

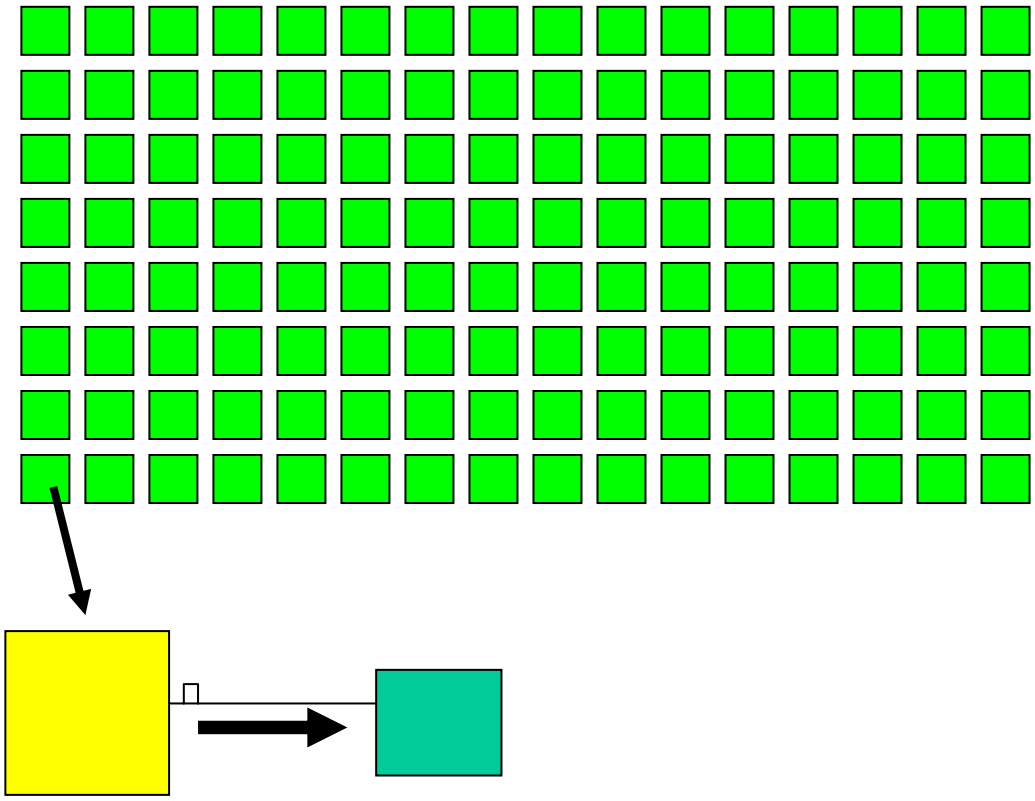


# Tuning the Pulse Width

- Carry chains continue in adjacent SLICEs.
- Attained best results by locating B in SLICE below that containing A.



# Replicate the 1-Bit Signature

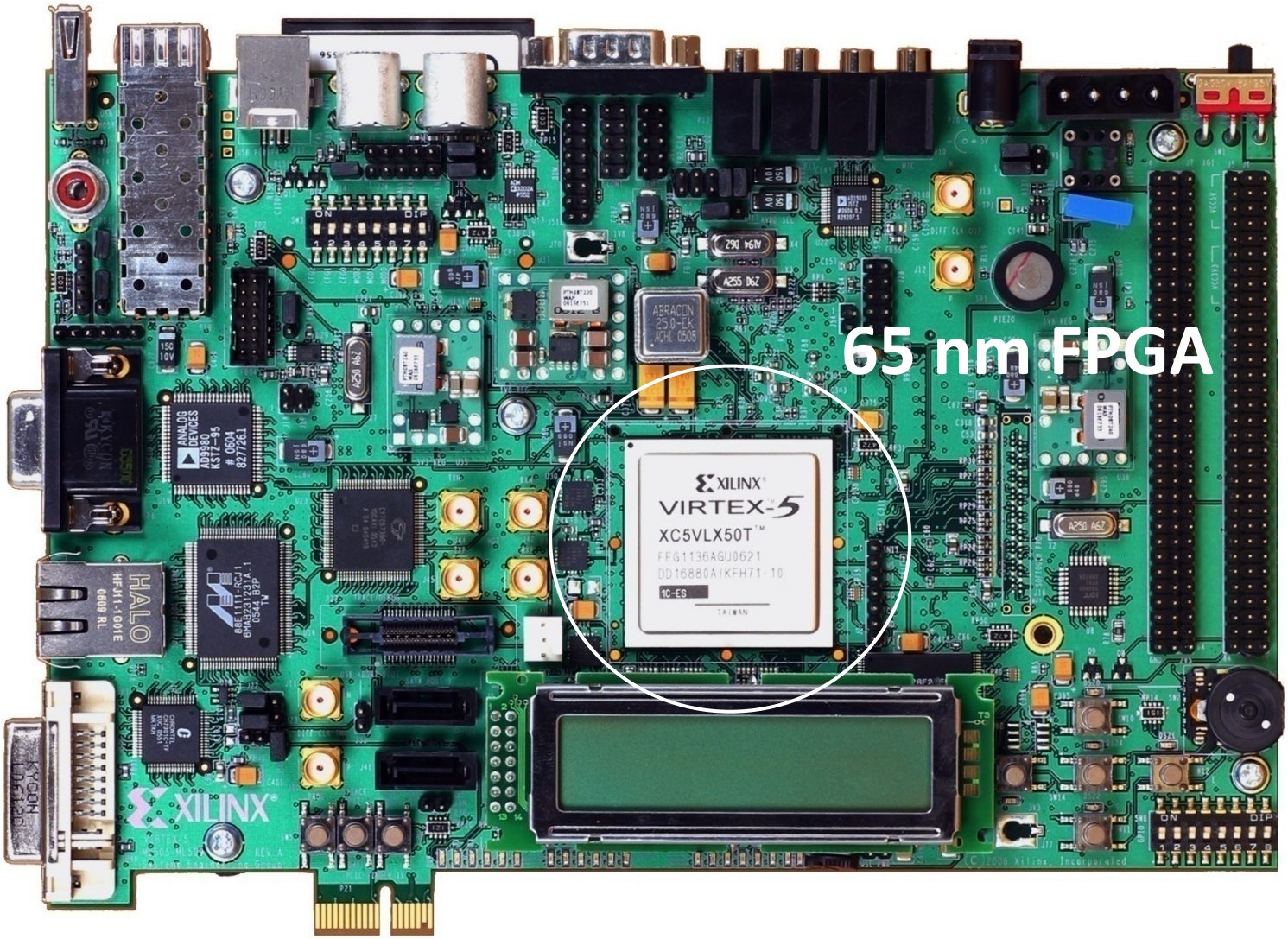


128-bit Signature

# Advantages

- PUF completely described in HDL
  - Synthesizable.
  - No matched/manual routing needed.
  - Easy incorporation into surrounding design
- Low area consumption
  - Two LUTs, carry chain, and one flip-flop per PUF bit.

# Hardware Platform



# Experimental Methodology

- Six Xilinx Virtex-5 65nm FPGAs.
- Implement PUF six times on each FPGA (in different regions).
- Total: 36 128-bit PUF implementations.
  - 630 pairs of PUFs.
- Compare “uniqueness” of pairs using Hamming distance.

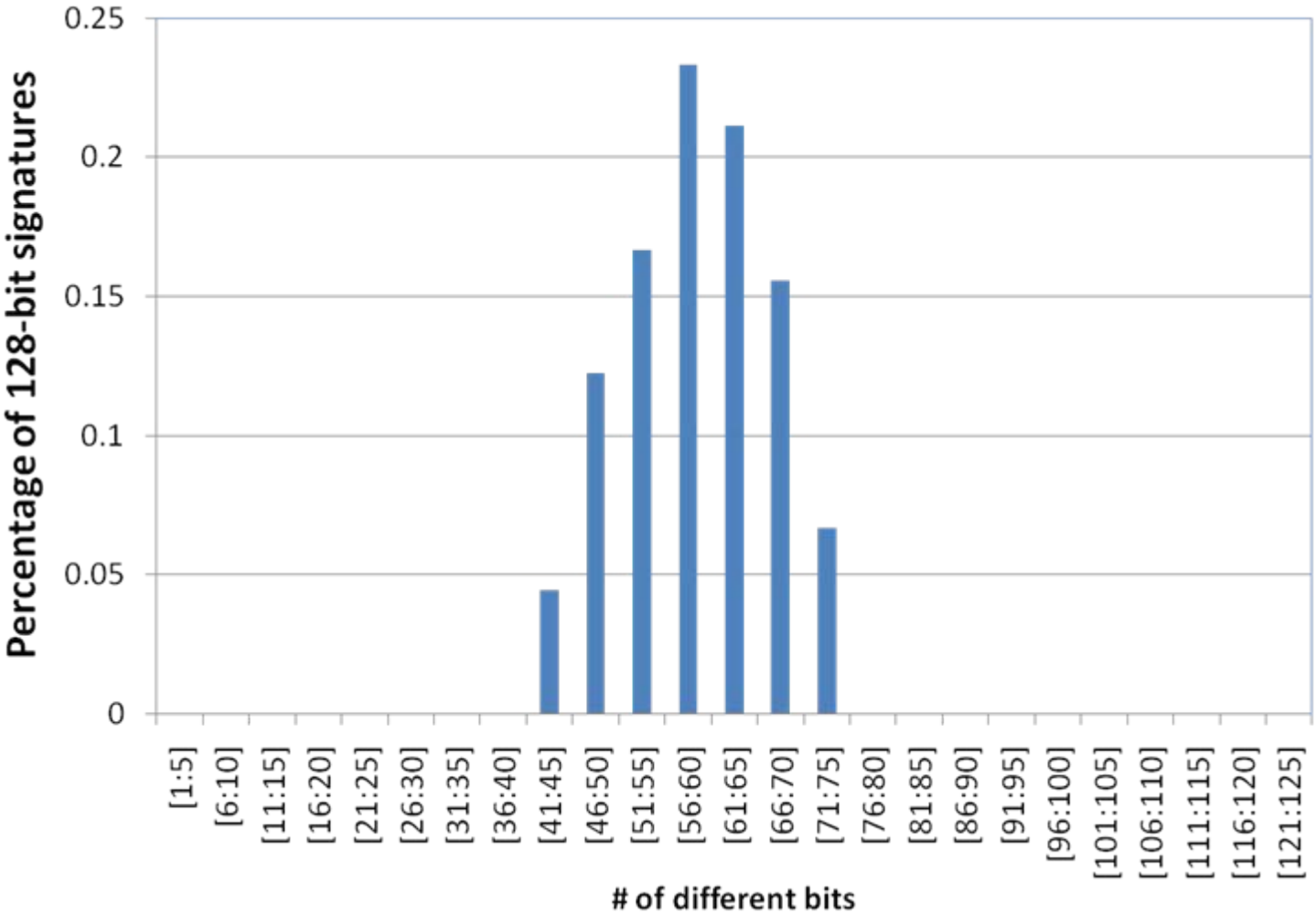
# Region Breakdown


- Six PUFs per FPGA:
  - Generate more data points.
- Can compare whether **same-die** PUFs are less unique than **cross-die** PUFs.
  - 6 boards x 15 = 90 same-die PUF pairs.

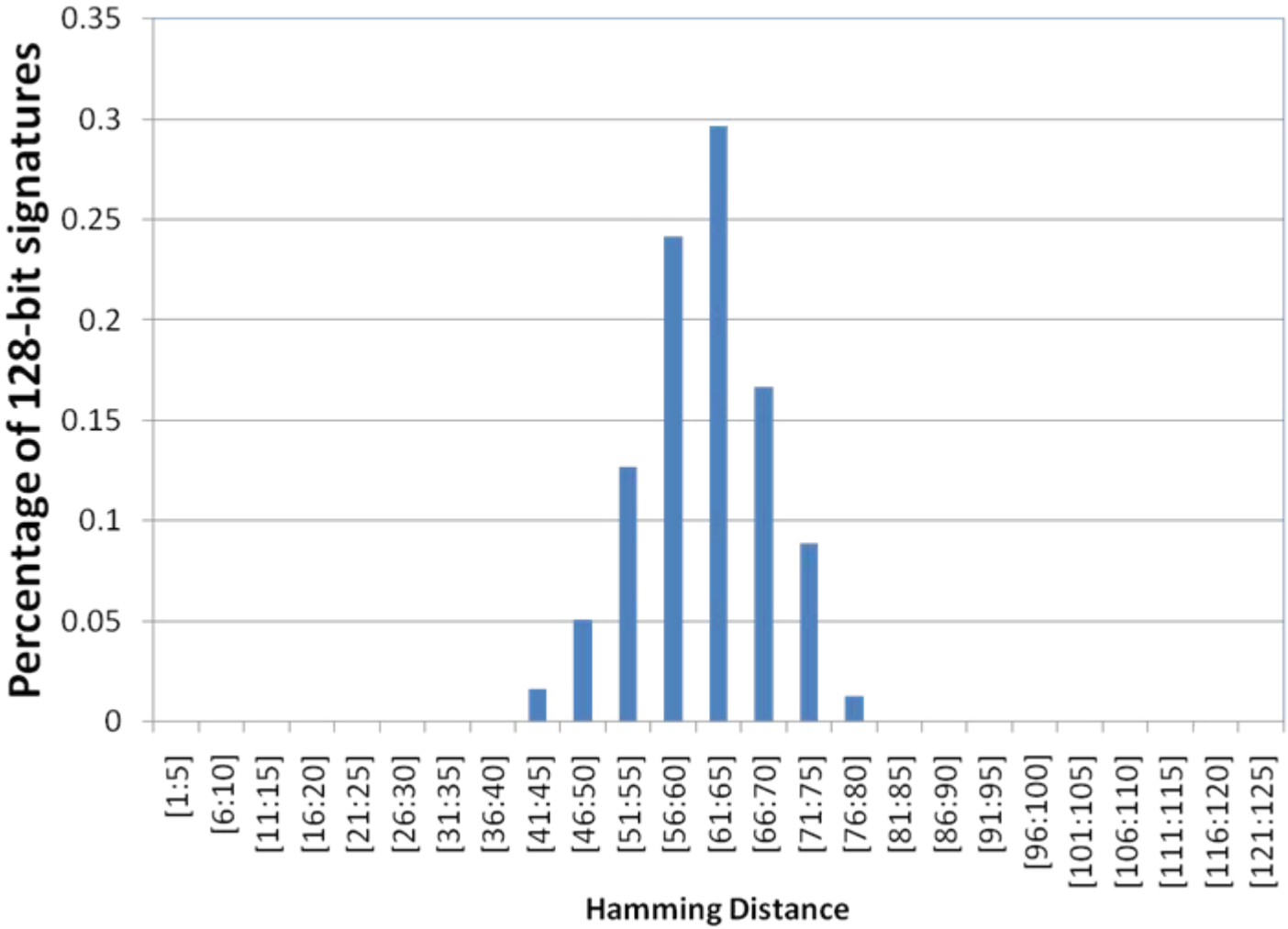
# Does It Work?

- For any two chips, each with a 128-bit PUF signature, and…
- If each bit was **equally** likely to be: a 0 or a 1 …
- Expect average Hamming distance between PUFs to be 64.

# PUF Signature Uniqueness



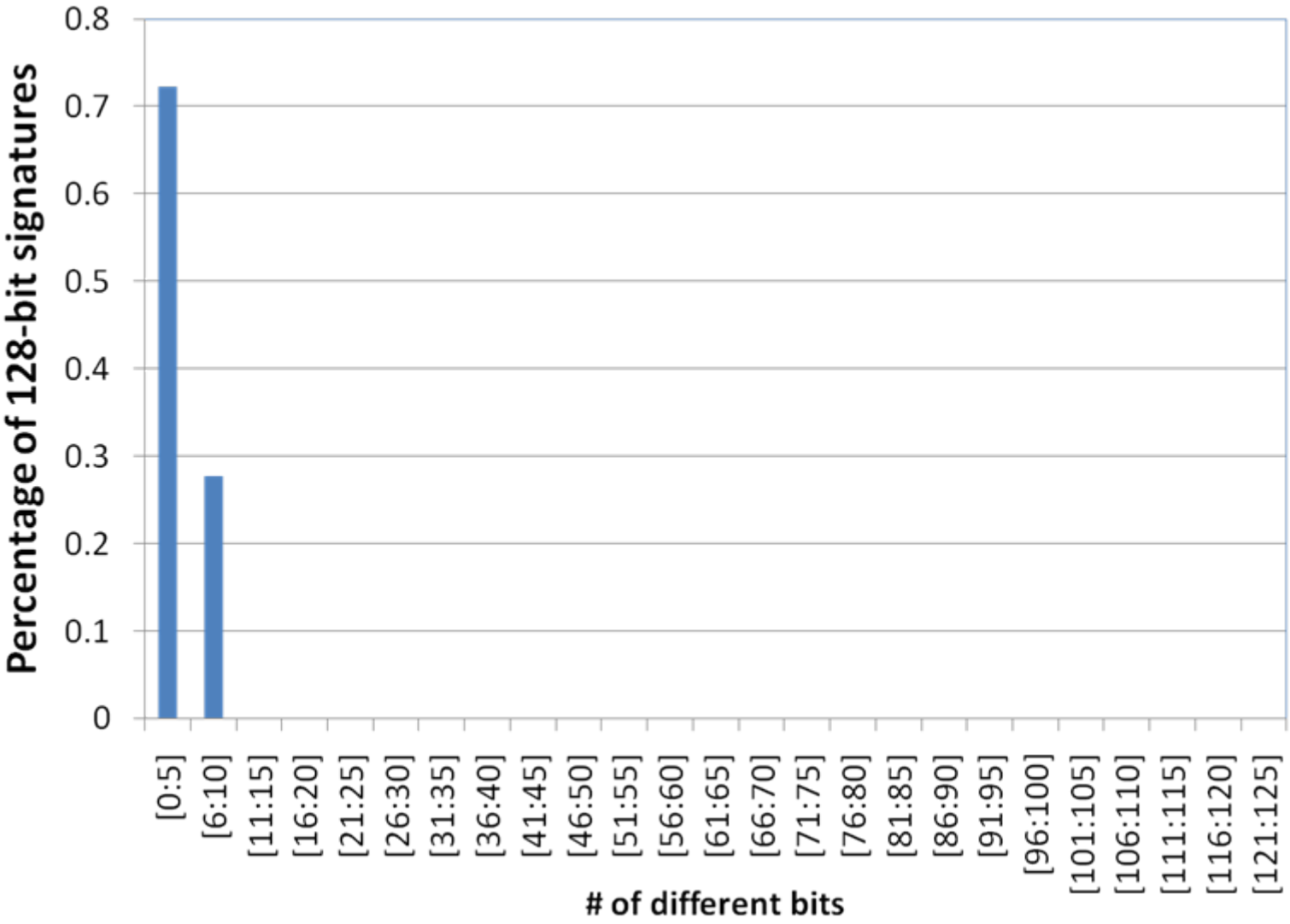
# Same-Die Signature Uniqueness



# Signature Reliability

- Chip delays change with temperature. Will signatures change at high temperature (70° C)?

# Signature Reliability



# Conclusions

- Contributions:
  - First FPGA-specific PUF.
    - Leverages FPGA architectural features.
  - Completely described in HDL
  - Small Si footprint.
- Results: reliability and uniqueness in line with prior PUFs.
- VHDL source at: www.eecg.toronto.edu/~janders

# Challenge/Response