05 Physically Unclonable Functions (PUFs)

^ not this

Engr 399/599: Hardware Security

Grant Skipper, PhD.

Indiana University

Adapted from: Mark Tehranipoor of University of Florida

Course Website

engr599.github.io

Write that down!

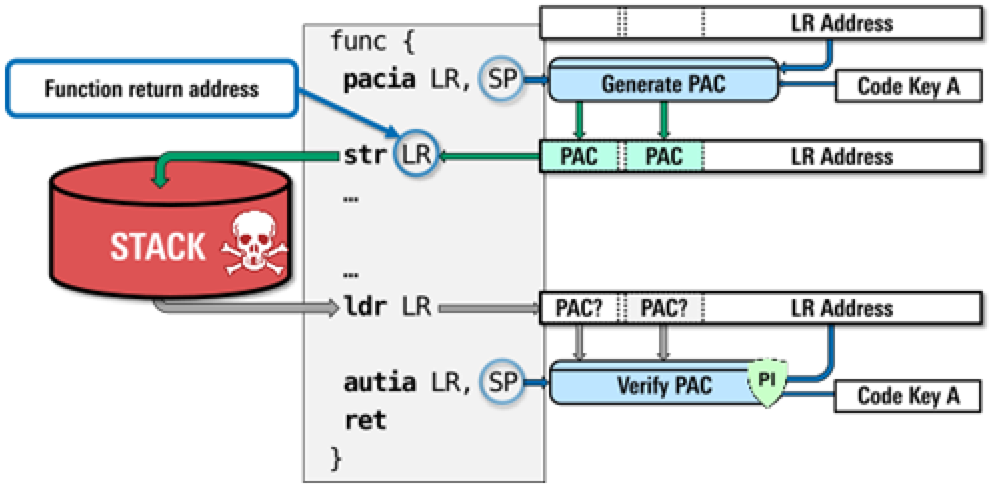
Agenda

- Review PUFs
- Finish PUFs & Start TRNG?
- Project Extension! -> This Friday! 2/21/24 Midnight!
- P2 Assigned Monday!

Side Quest!

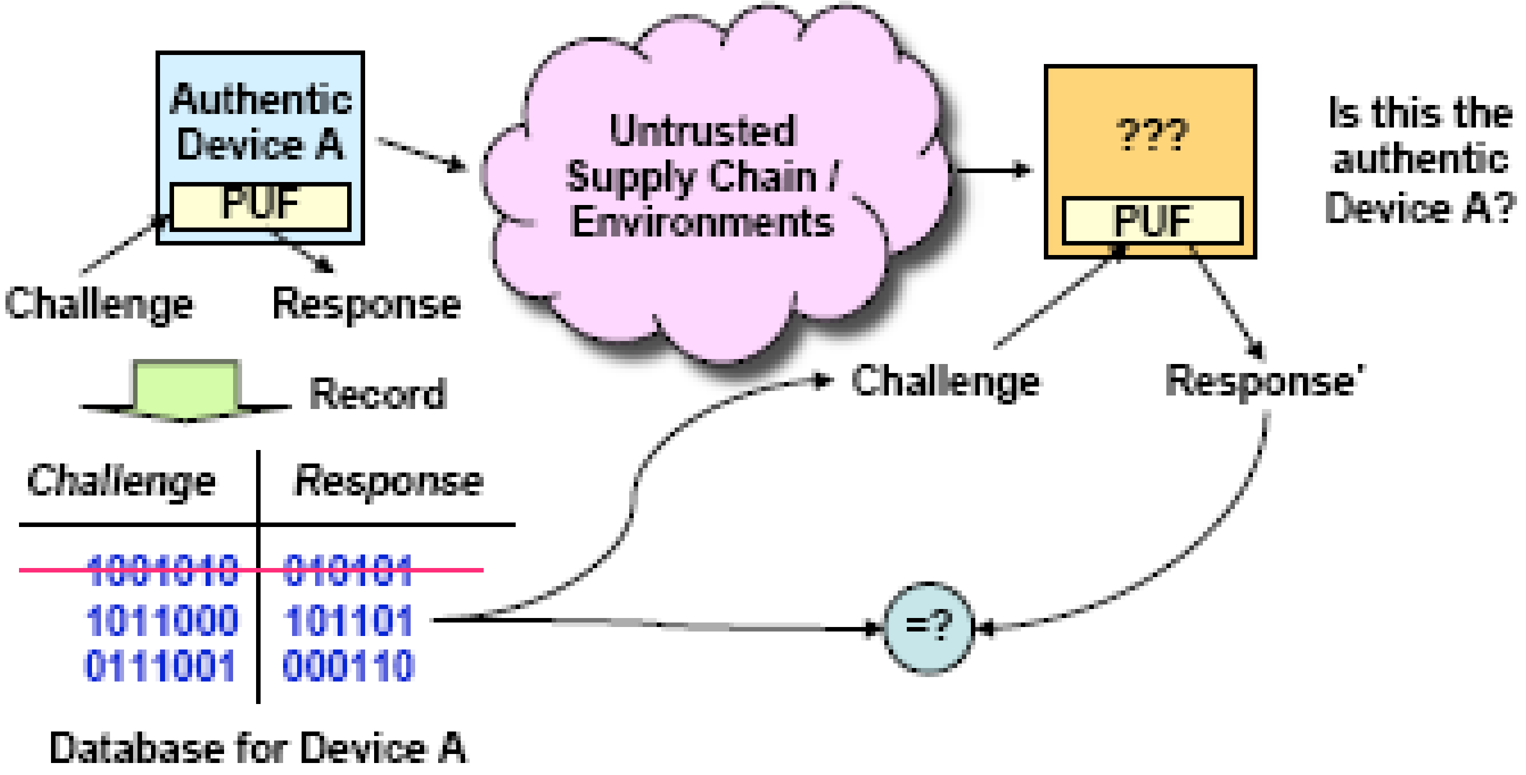
Unpatchable Hardware flaw in Apple Silicon!

PACMAN - https://thehackernews.com/2022/06/mit-researchers-discover-new-flaw-in.html



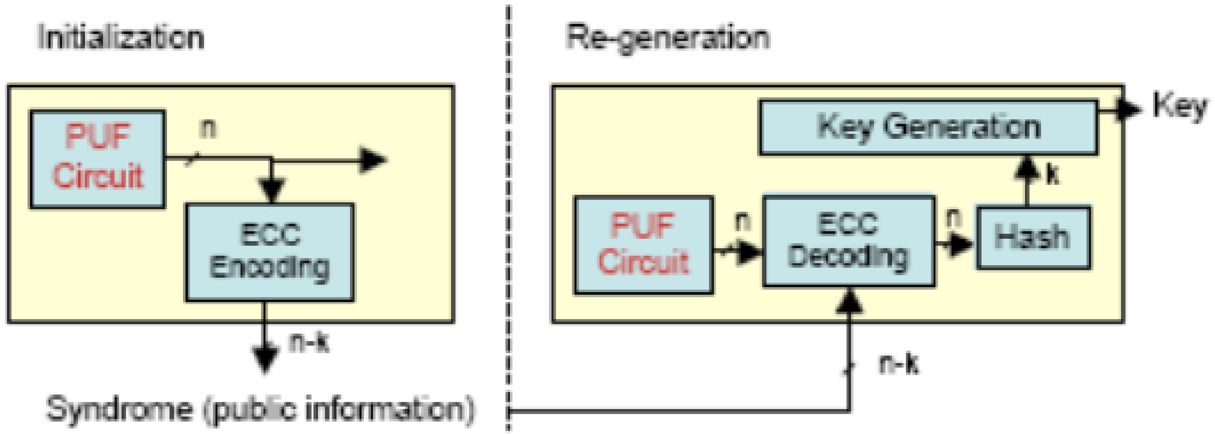
Applications – Authentication

- Same challenges should not be used to prevent the man-in-the-middle attacks



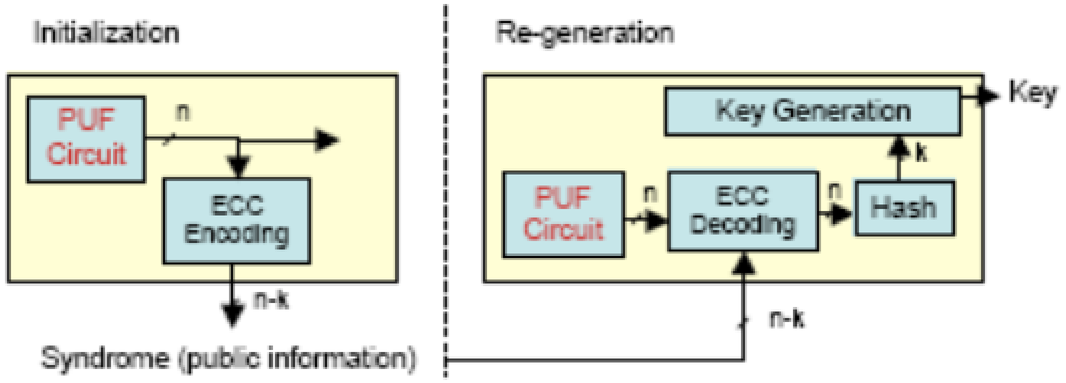
Application – Cryptographic Key Generation

- The instability is a problem
- Some crypto protocols (e.g., RSA) require specific mathematical properties that random numbers generated by PUFs do not have
- How can we use PUFs to generate crypto keys?
  - Error correction process: initialization and regeneration
  - There should be a one-way function that can generate the key from the PUF output



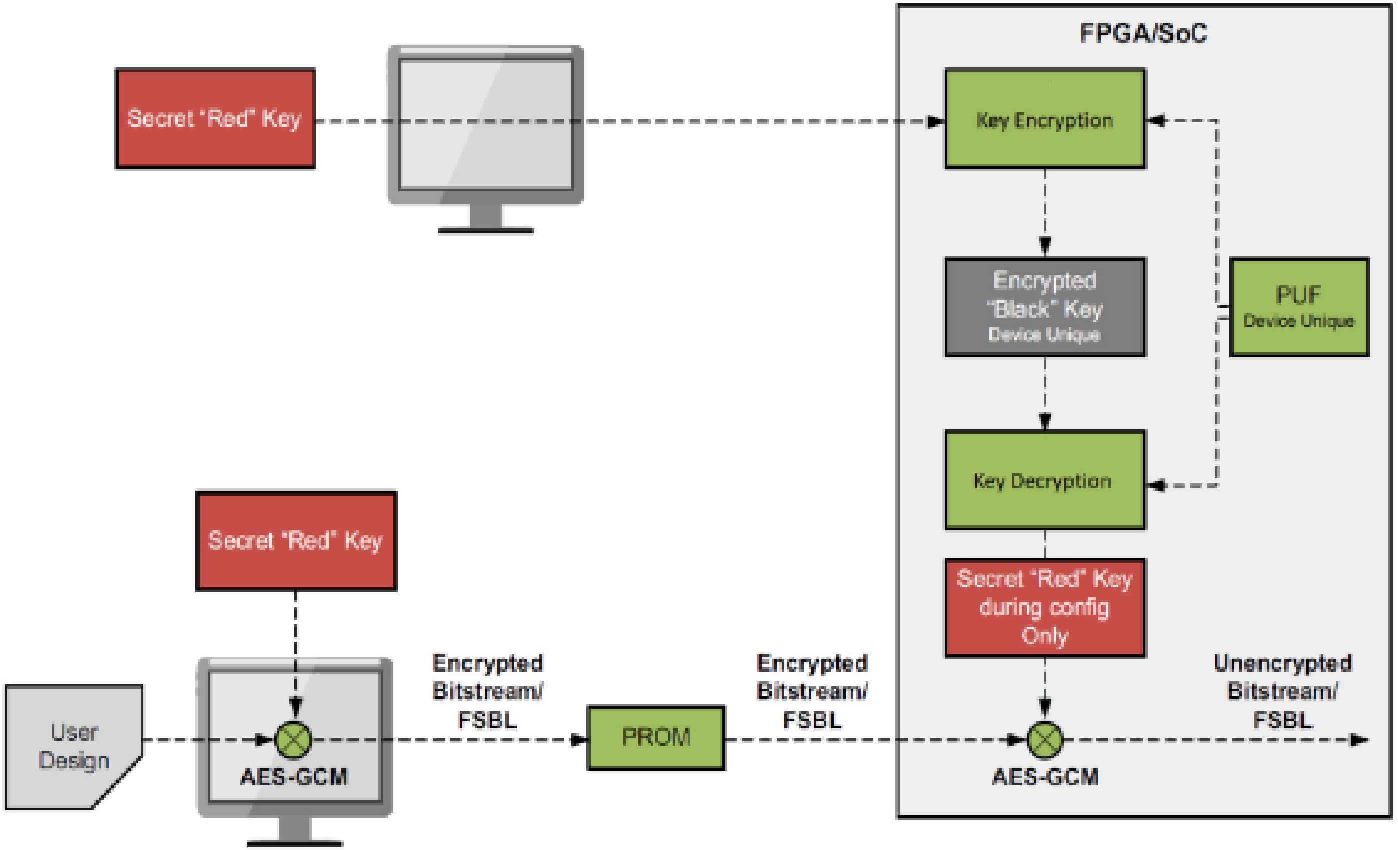
Crypto Key Generation

- Initialization: a PUF output is generated and error correcting code computes the syndrome (public info)
- Regeneration: PUF uses the syndrome from the initial phase to correct changes in the output
- Clearly, the syndrome reveals information about the circuit output and introduces vulnerabilities



Red/Black Keys

Use PUFs to hide the on-chip key!



Reliability and Security Metrics

- *Inter-chip variation:* How many PUF output bits are different between PUF A and PUF B? This is a measure of uniqueness. If the PUF produces uniformly distributed independent random bits, the inter-chip variation should be 50% on average.
- *Intra-chip (environmental) variation:* How many PUF output bits change when re-generated again from a single PUF with or without environmental changes? This indicates the *reproducibility* of the PUF outputs. Ideally, the intra-chip variation should be 0%.

Configurable Ring Oscillator

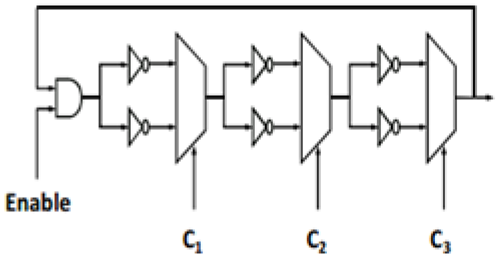


Fig. 3(a)

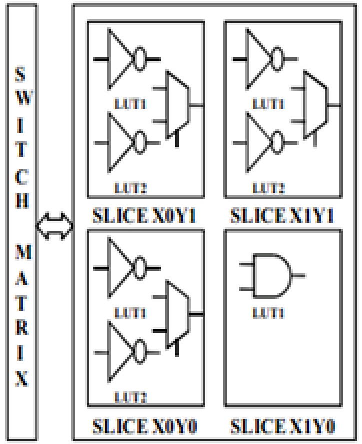


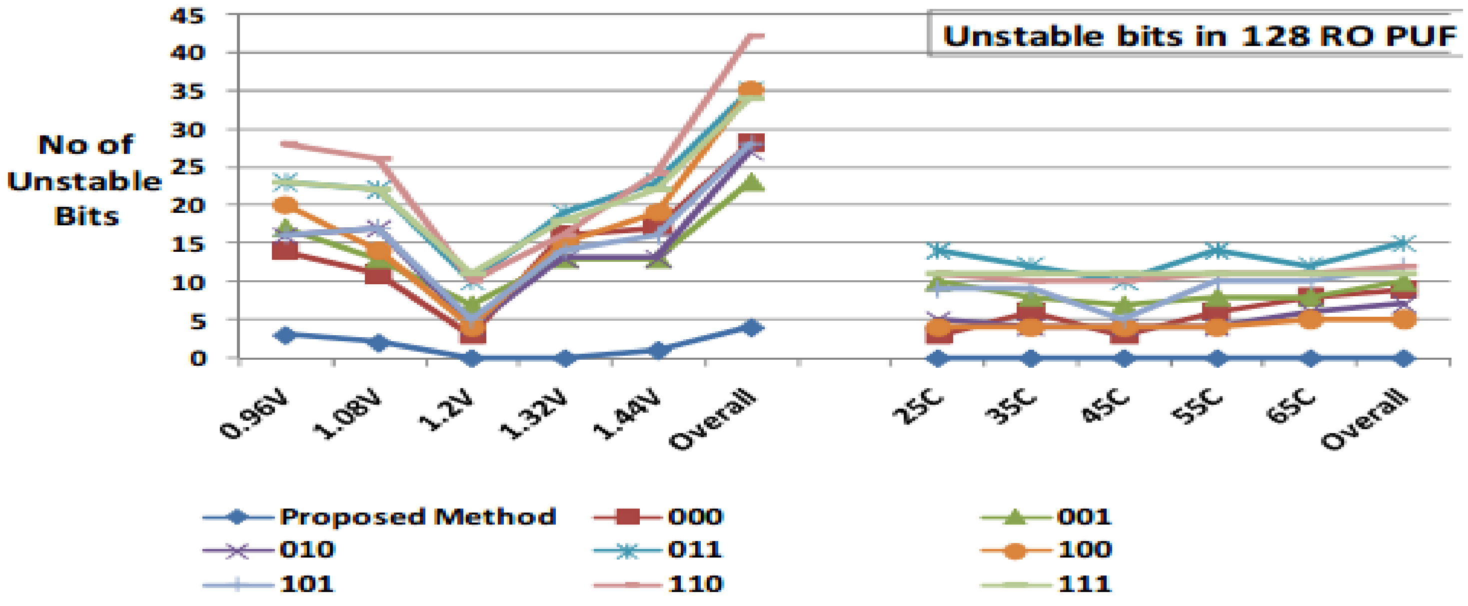
Fig. 3(b)

Table 1. Frequency differences in a configurable RO pair

$c_1c_2c_3$	Frequency of ROs in CLB i	Frequency of ROs in CLB j	Δf
000	f_0	f'_0	$\lvert f_0 - f'_0 \rvert$
001	f_1	f'_1	$\lvert f_1 - f'_1 \rvert$
010	f_2	f'_2	$\lvert f_2 - f'_2 \rvert$
011	f_3	f'_3	$\lvert f_3 - f'_3 \rvert$
100	f_4	f'_4	$\lvert f_4 - f'_4 \rvert$
101	f_5	f'_5	$\lvert f_5 - f'_5 \rvert$
110	f_6	f'_6	$\lvert f_6 - f'_6 \rvert$
111	f_7	f'_7	$\lvert f_7 - f'_7 \rvert$

- The pair which has the maximum difference in frequency in CRO is selected.

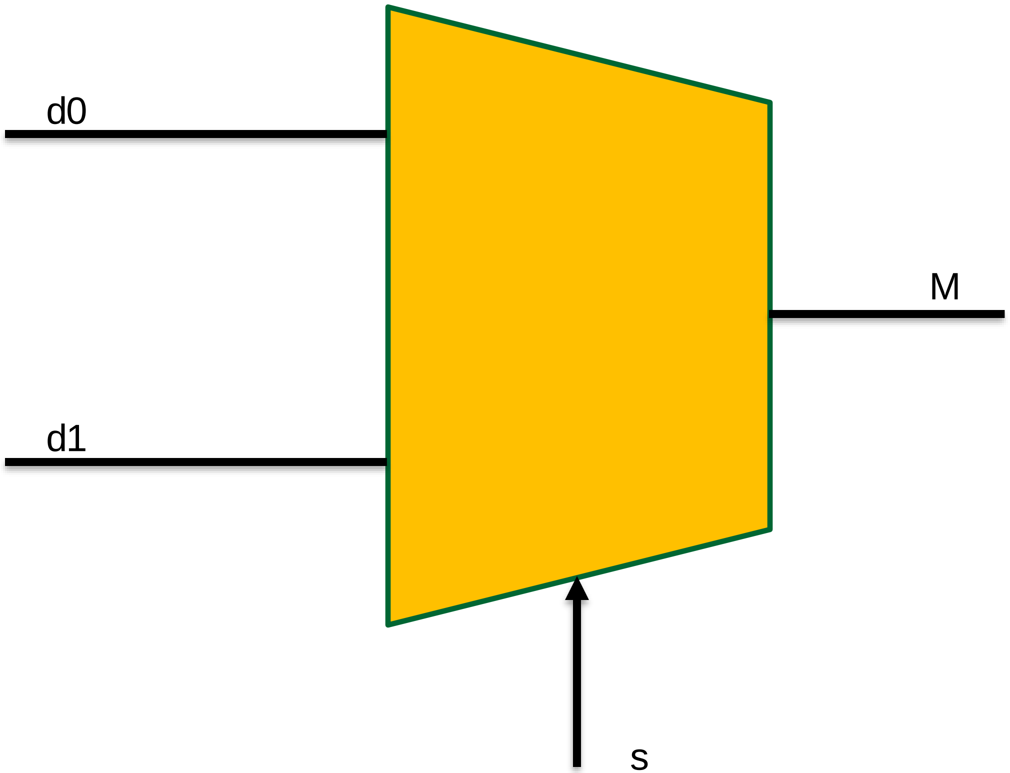
Configurable Ring Oscillator



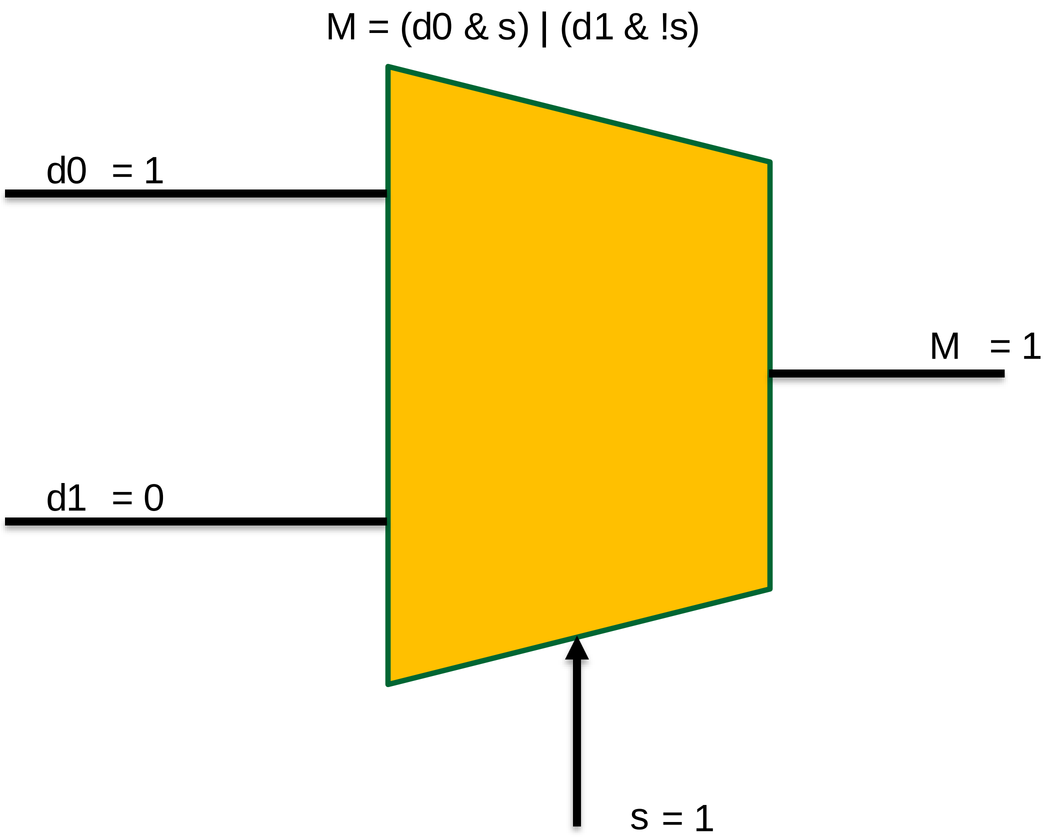
- PUF – Physically Unclonable Function
- RO – Ring Oscillator
- MUX – Multiplexer
- SRAM – Static Random Access Memory
- LUT – Look Up Table
- FF – Flip Flop

- Types of PUFs:
  - Arbiter PUF – Signal races between MUXes
  - RO PUF – Signal generated from ROs to trigger counter
  - Butterfly PUF – Exciting a cross-coupled circuit

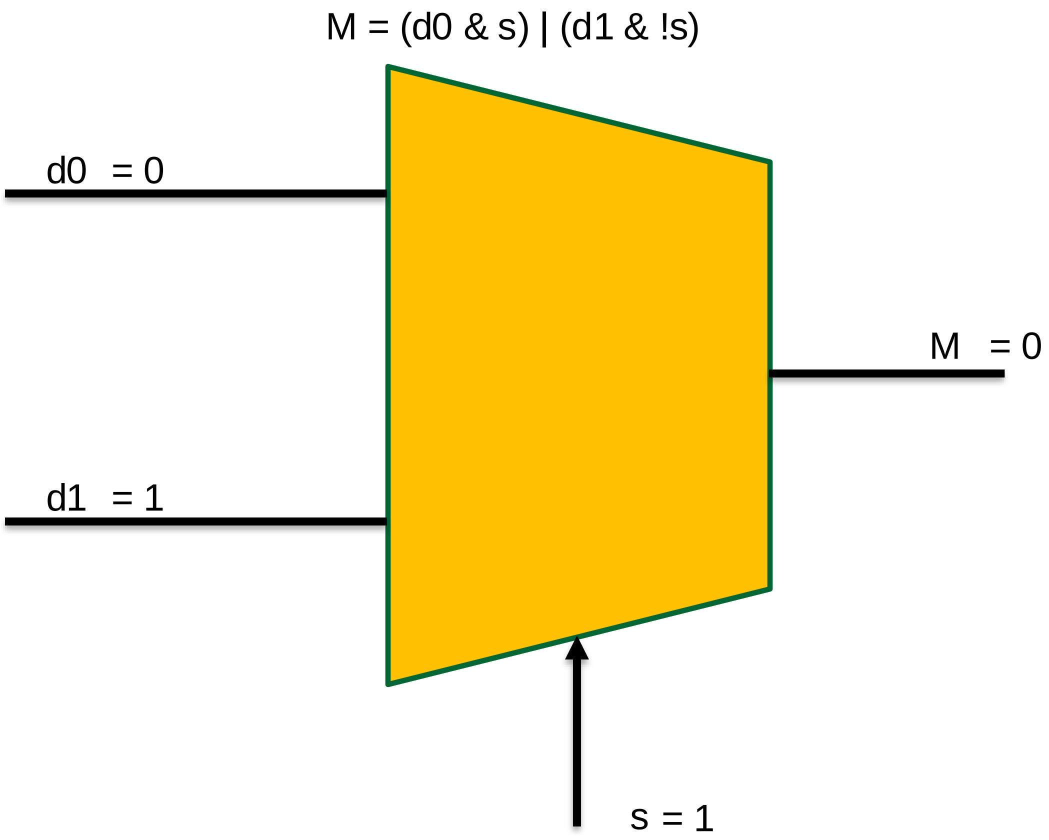
The Arbiter PUF



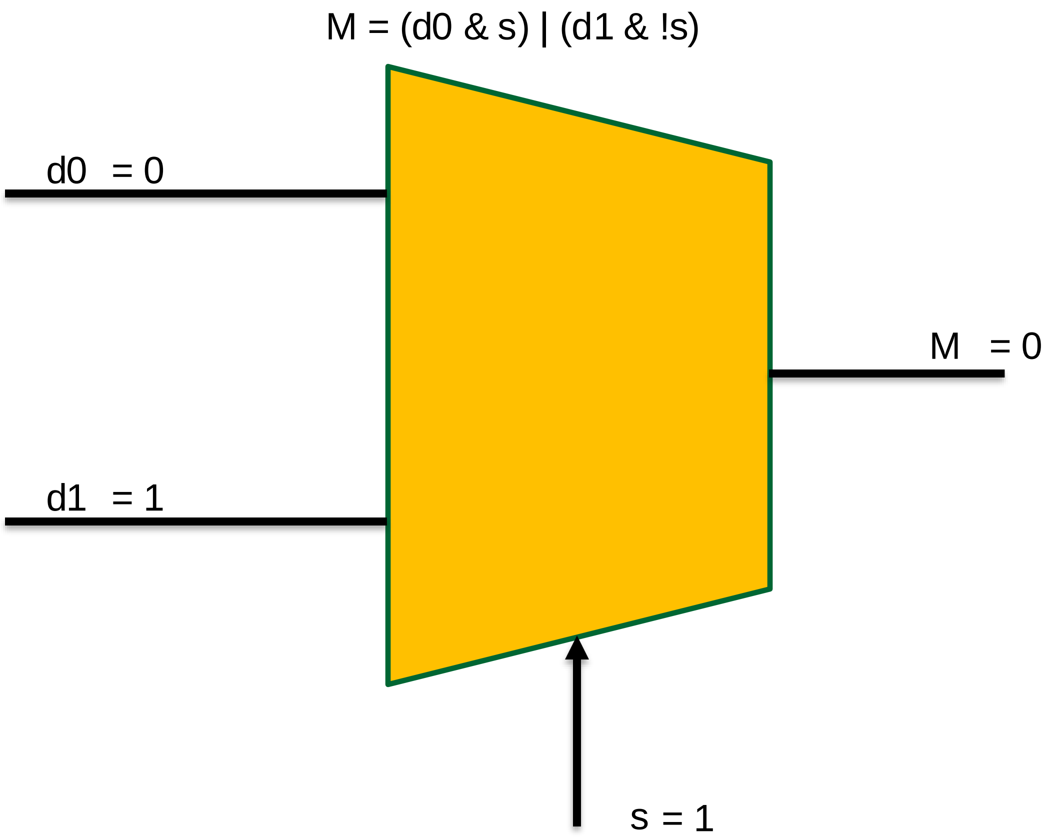
The Arbiter PUF



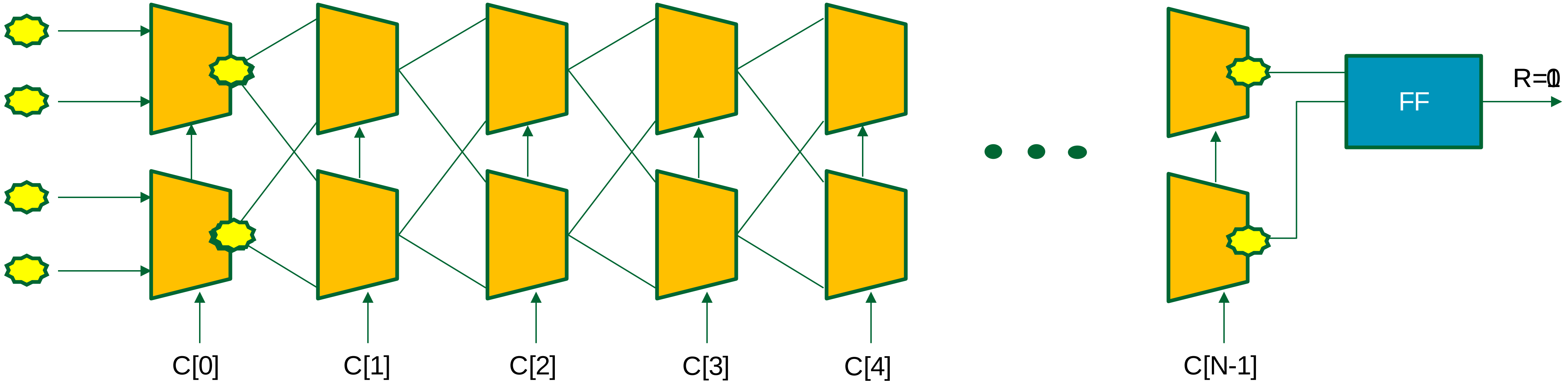
The Arbiter PUF



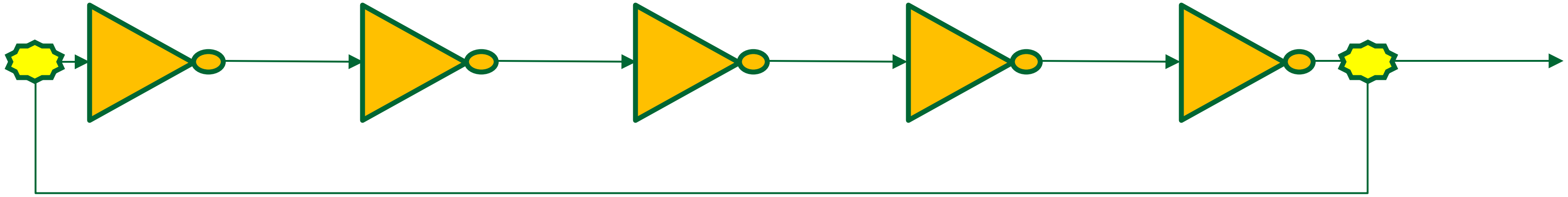
The Arbiter PUF



The Arbiter PUF

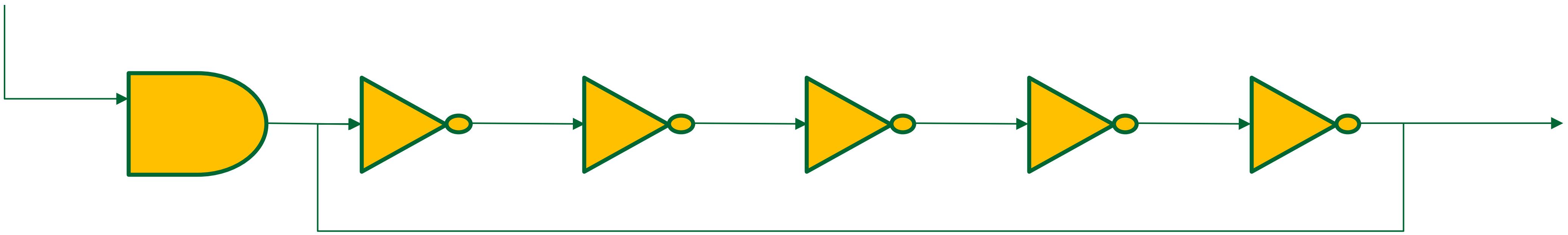


Ring Oscillator (RO) PUF

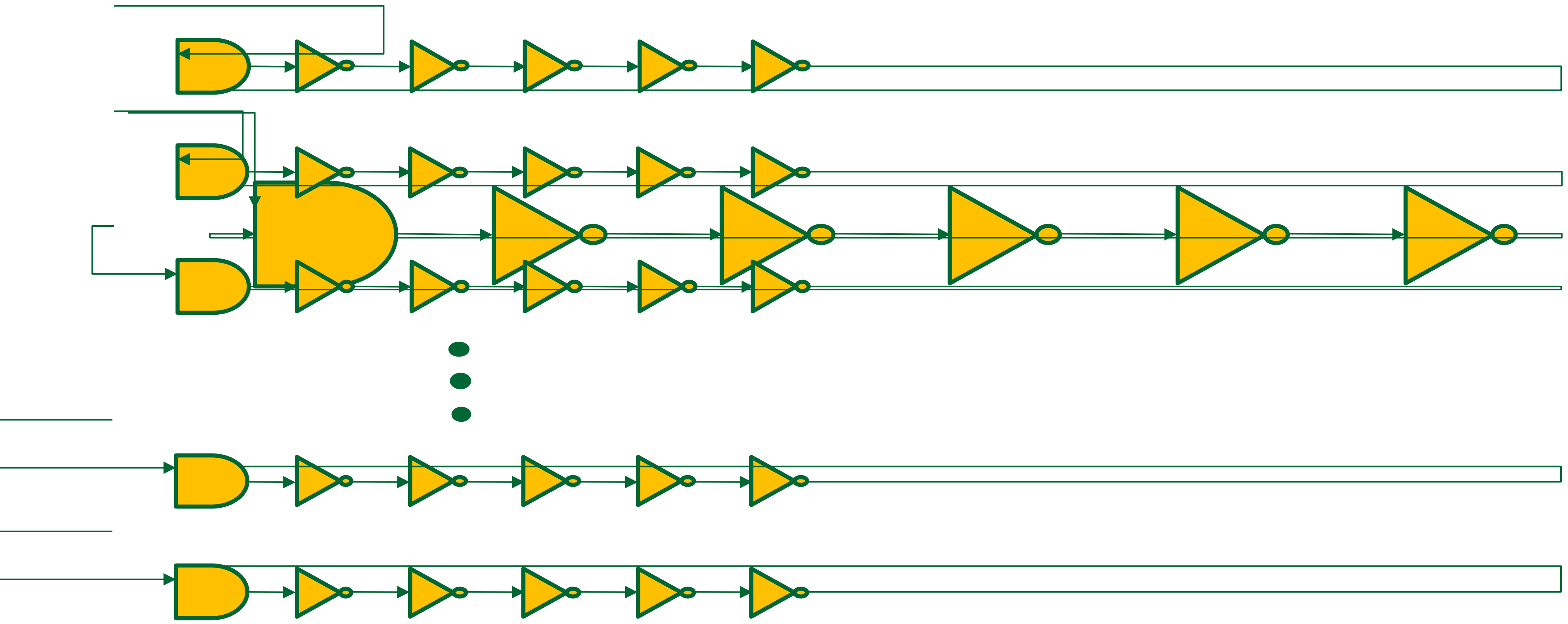


0 1

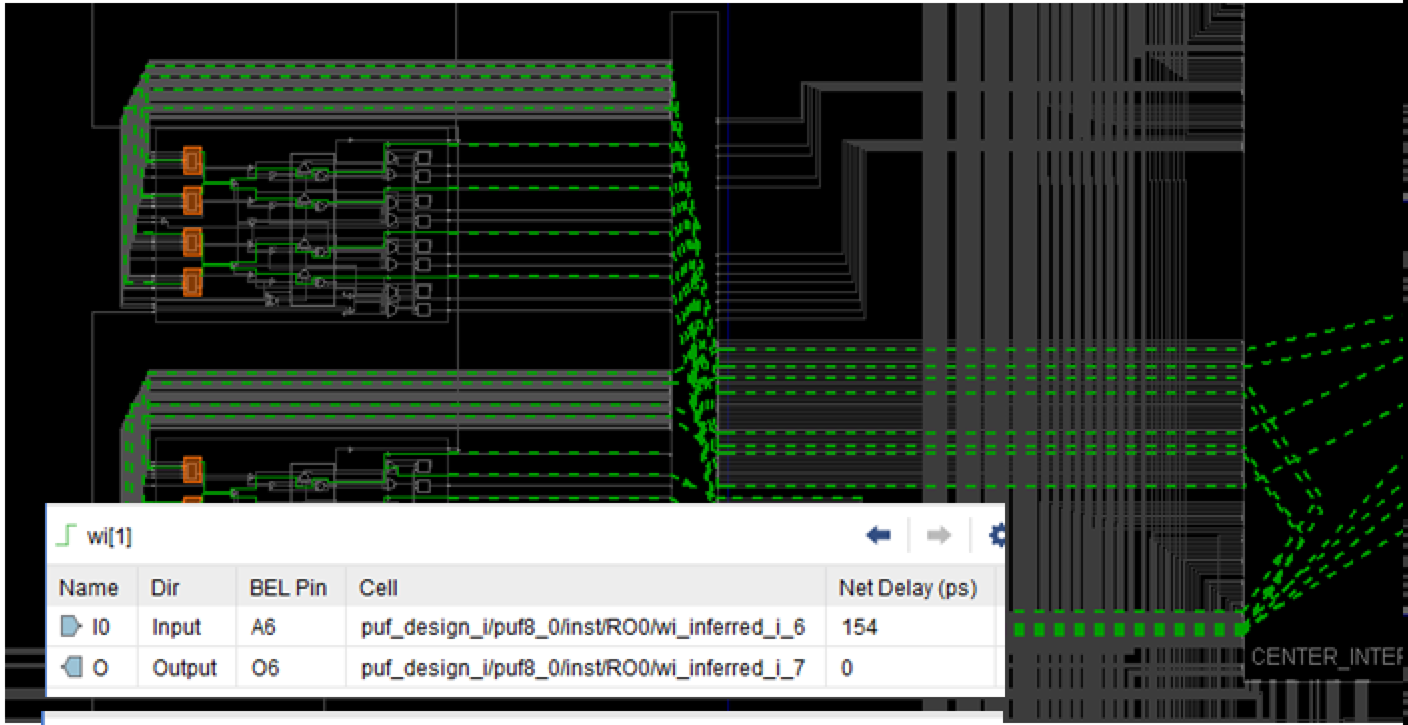
Ring Oscillator (RO) PUF

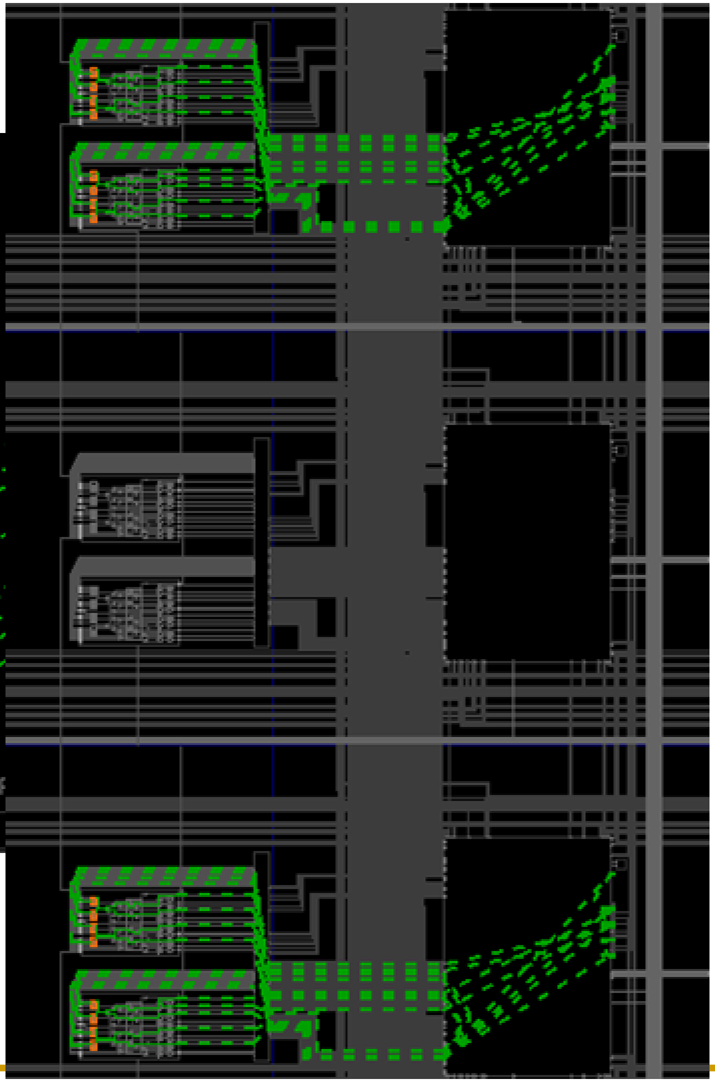


Ring Oscillator (RO) PUF



RO Place and Routing





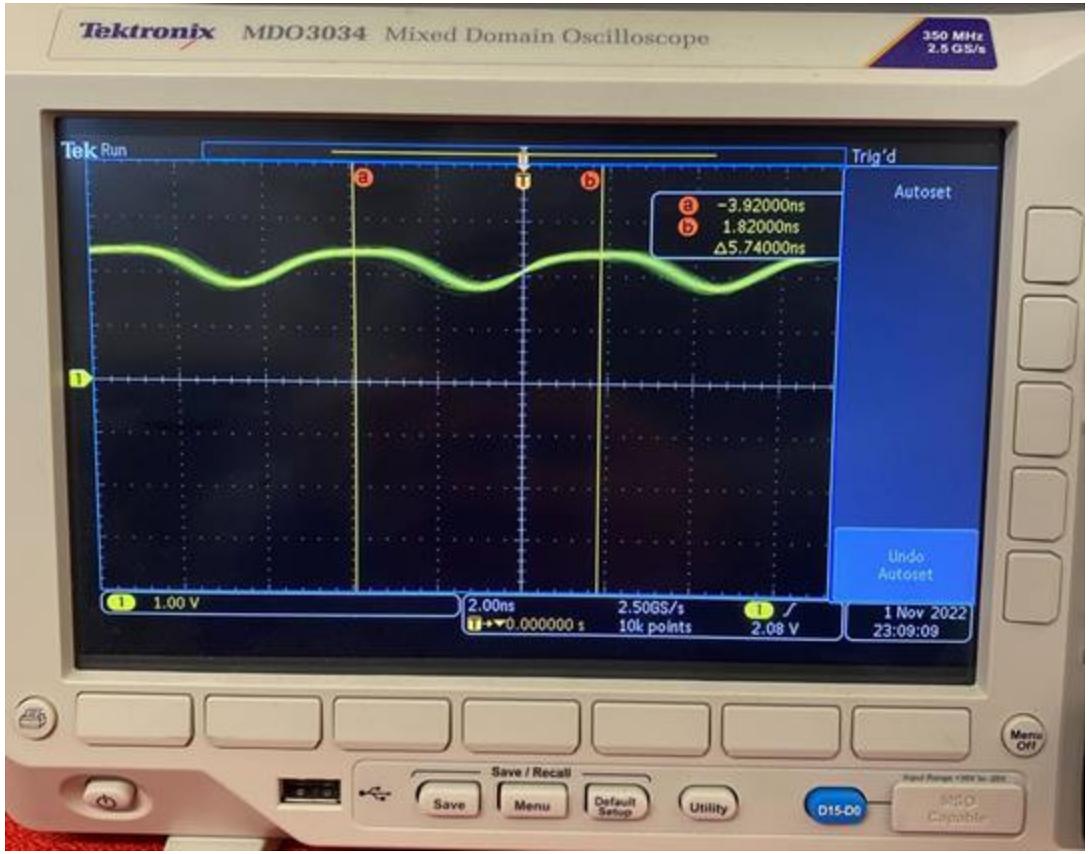
wi[1]

Name	Dir	BEL Pin	Cell	Net Delay (ps)
I0	Input	A6	puf_design_i/puf8_0/inst/RO0/wi_inferred_i_6	154
O	Output	O6	puf_design_i/puf8_0/inst/RO0/wi_inferred_i_7	0

wi[1]

Name	Dir	BEL Pin	Cell	Net Delay (ps)
I0	Input	A6	puf_design_i/puf8_0/inst/RO1/wi_inferred_i_6	151
O	Output	O6	puf_design_i/puf8_0/inst/RO1/wi_inferred_i_7	0

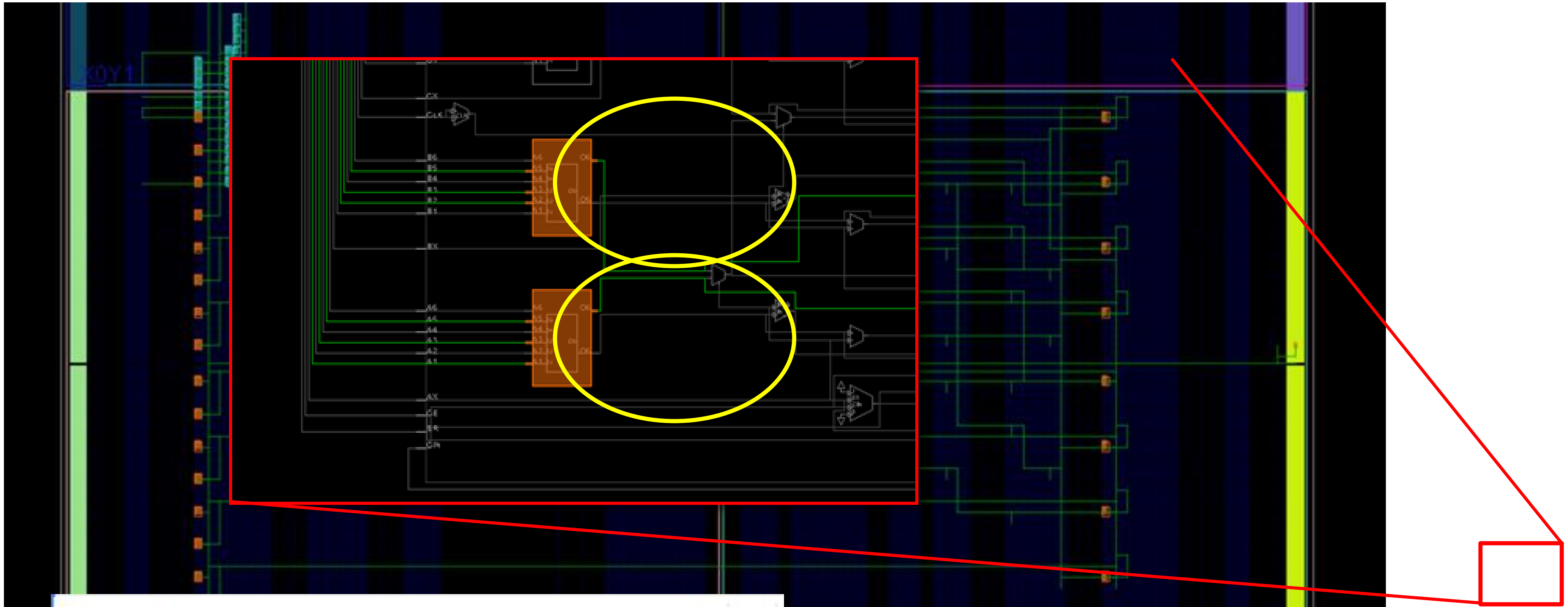
RO Frequency



14 ROs have this Waveform

2 Did not appear (Currently Troubleshooting)

RO Place and Routing



M_INST_0

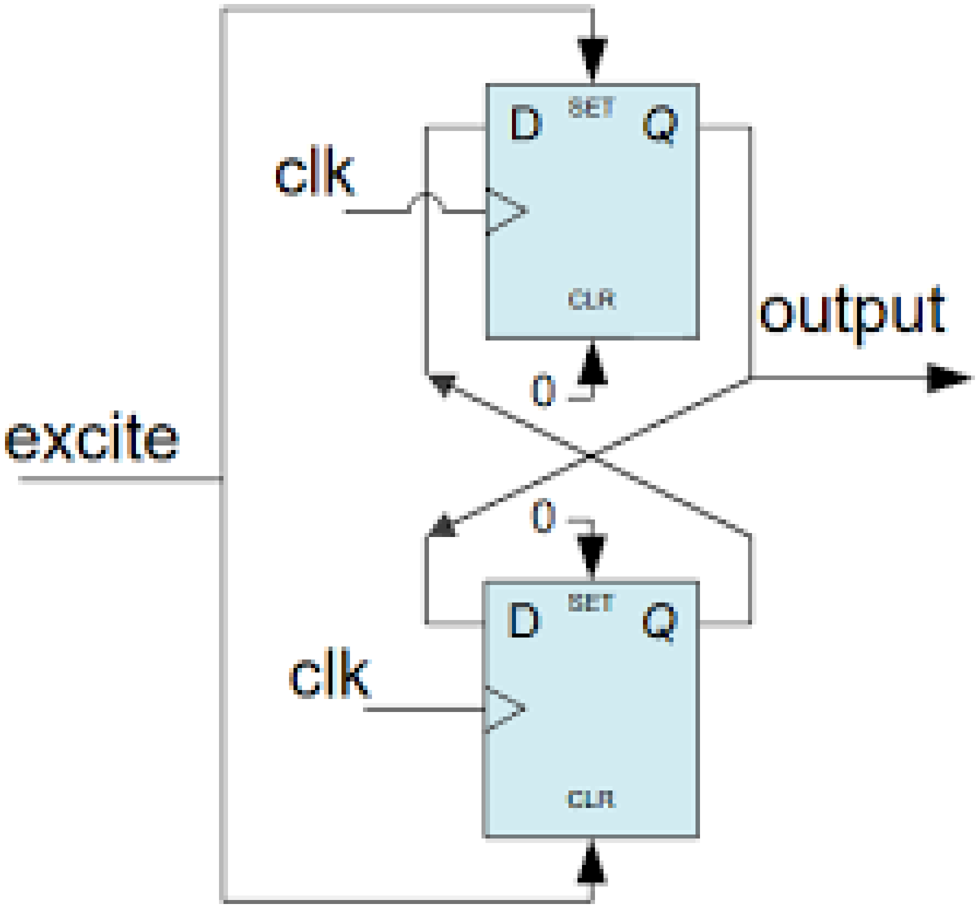
Name	Dir	BEL Pin	Net	Net Delay (ps)
I0	Input	A5	puf_design_i/puf8_0/inst/m1/m0/m0/m0/d0	1959
I1	Input	A2	puf_design_i/puf8_0/inst/m1/m0/m0/m0/s	3840
I2	Input	A3	puf_design_i/puf8_0/inst/m1/m0/m0/m0/d1	1967
O	Output	O6	puf_design_i/puf8_0/inst/m1/m0/m0/m0/M	0

Butterfly PUF

Cross Coupled DFFs.

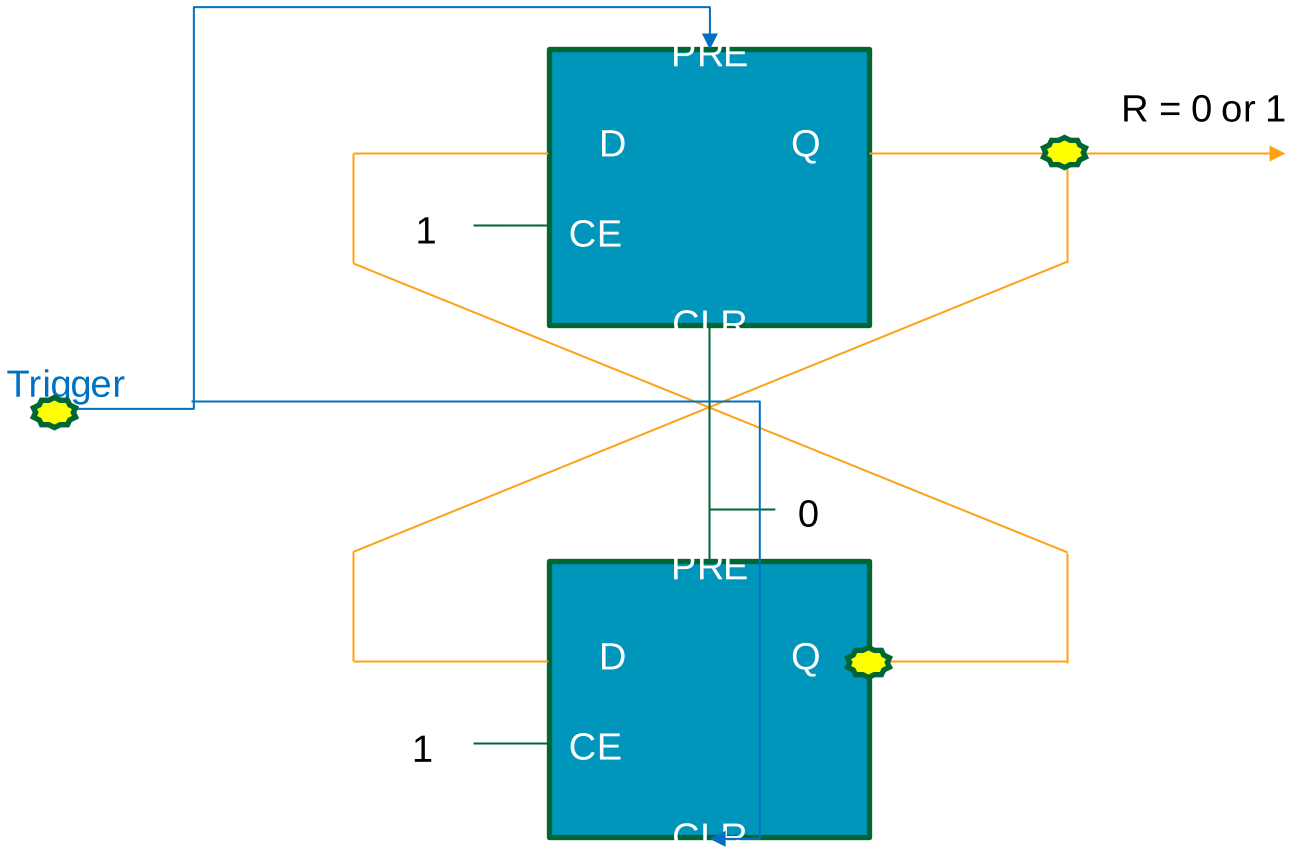
Physical 'race condition' easy to constrain compared to Arbiter.

Less machinery needed than RO PUF. (Less resources).



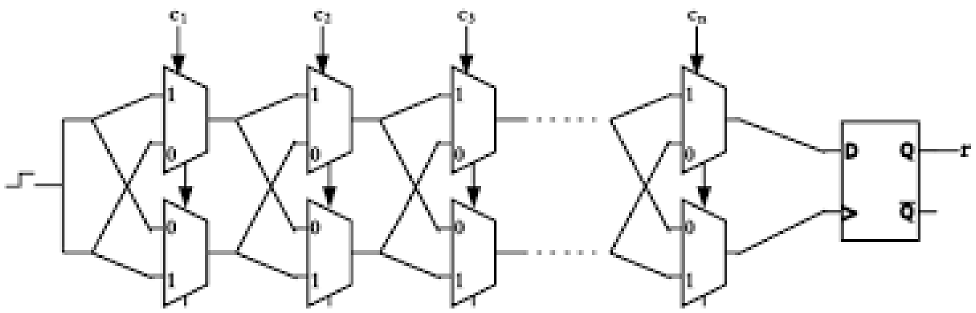
The Butterfly PUF

PRE	CLR	CE	D	Q
0	1	X	X	1
1	0	X	X	0
0	0	X	X	X
1	1	1	1	1
1	1	1	0	0
1	1	0	X	

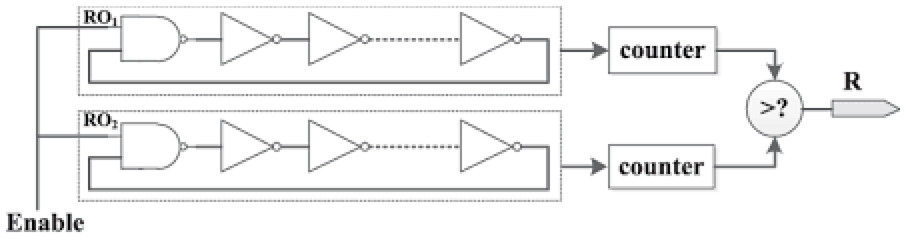


Three Primary PUF Types

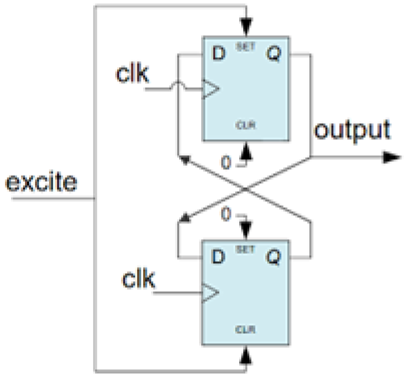
Arbiter: racing signals using MUXes (path delay)



RO: counting signals using path delay.



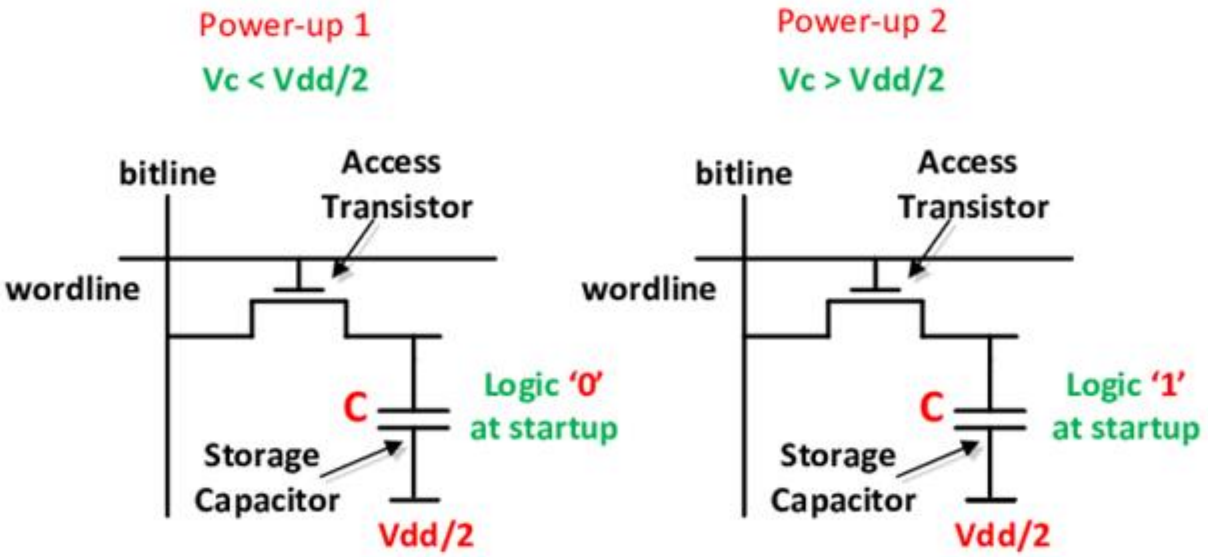
Butterfly: Activating FFs using path delay.



DRAM PUF

Relies on biases in DRAM cell startup value.

Effectively isolating the leakage characteristics of cell capacitance.



What makes a good PUF?

1. **Uniqueness!**
   a. Ideally every challenge should elicit a single unique response.
   b. High Delta between two different challenge/response pairs (eg 50% hamming distance).

2. **Difficult to model!**
   a. resilient against ML / statistical methods.

3. **Stable!**
   a. stable over lifetime/age.
   b. stable against temp/environmental stimulus.

So what are PUFs actually useful for?

1. **Authentication!**
   a. <u>If you know the unique challenge response (C/R) pairs</u>, you can interrogate the PUF to ensure it is ‘yours’.
2. **Key Entropy!**
   a. <u>If you know C/R pairs</u>, AND have a stable/reliable PUF, it can add entropy to key generation procedures.
3. **Hiding/Obfuscating data!**
   a. <u>If you know your C/R pairs</u>, a PUF can allow you to store information on chip without worrying about physically storing a key in silicon.

What Do PUFs Struggle With

1. **Communication**
   a. It is difficult to implement a PUF when the purpose is to secure a two way communication channel. More attack vectors to deal with (CPUF).
   b. Not impossible, but requires extra machinery (more attack vectors!).
2. **Key Generation**
   a. Depending on a PUF for the entire key generation process can be risky due to reliability and stability concerns.
3. **Difficult to Implement**
   a. Implementation depends on isolating physical features of the target. Requires significant expertise in the target medium.