ClkScrew

Aaron Zhang

Outline

- **Introduction to DVFS and background information.**
- What makes CLKSCREW unique?
- Challenges to CLKSCREW
- Attacks and Results
- Conclusion

Voltage

+

Frequency

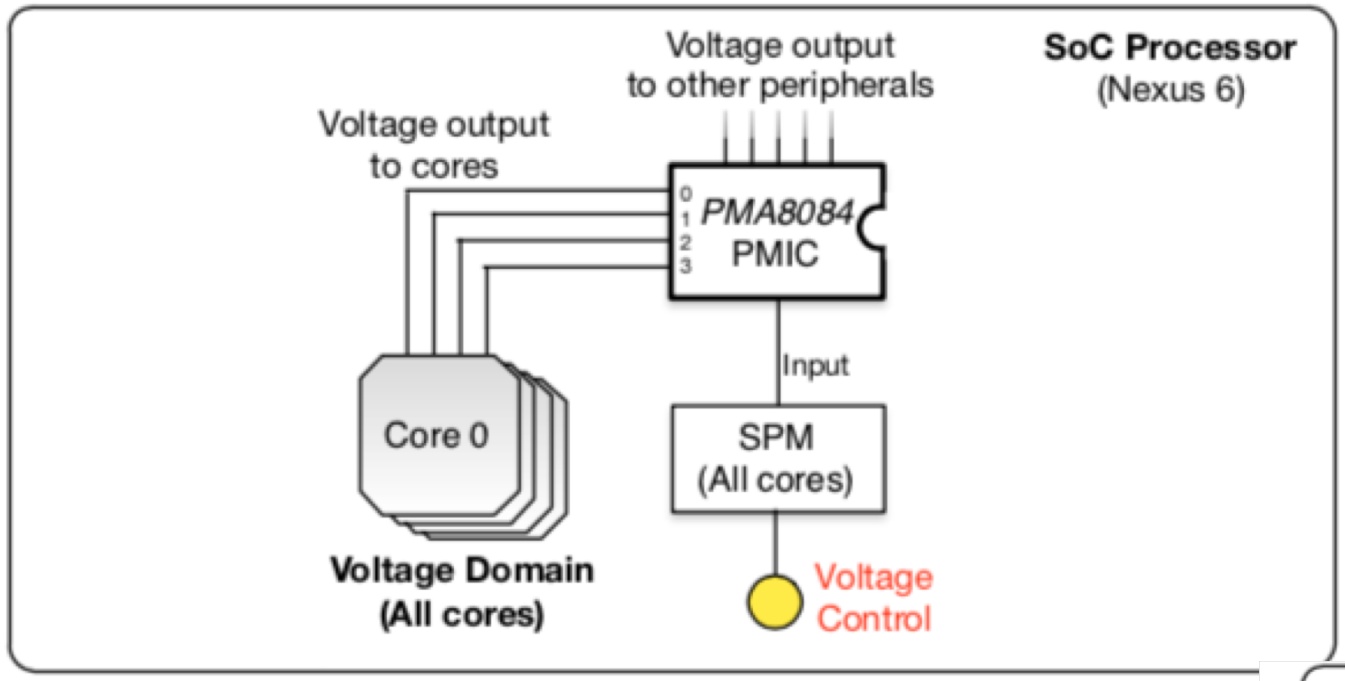
=

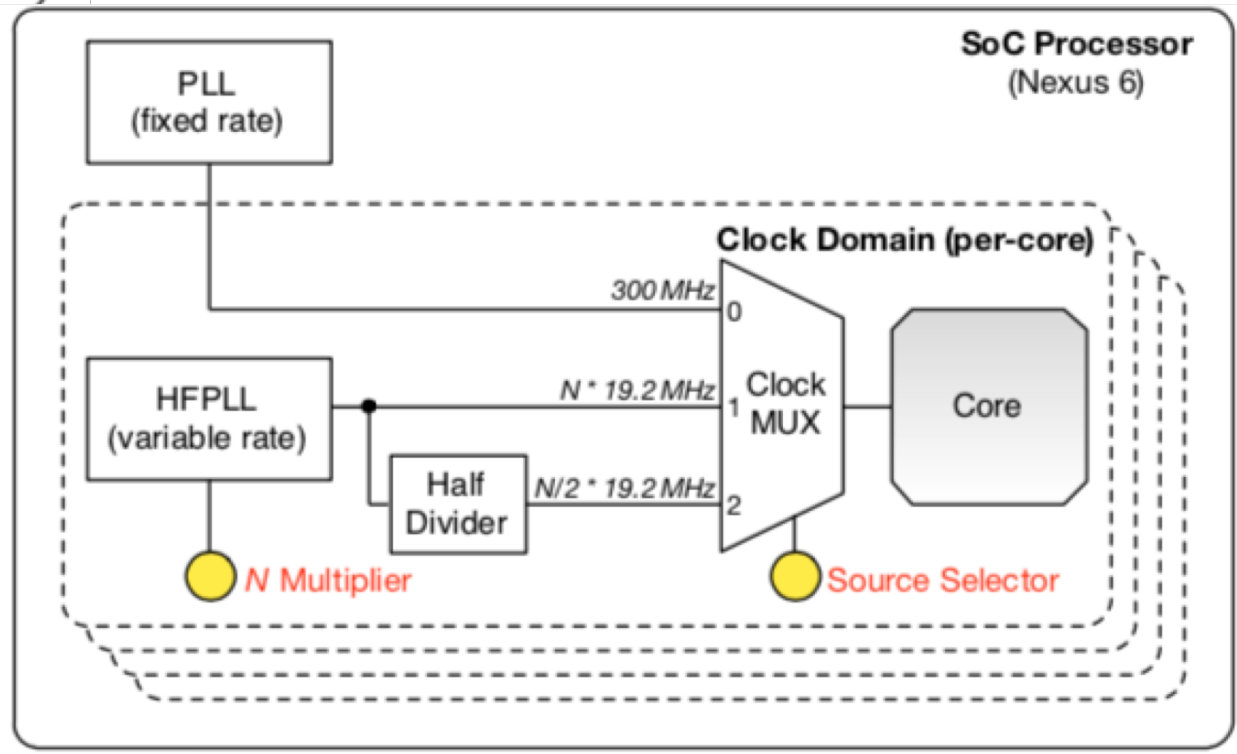
Energy Usage

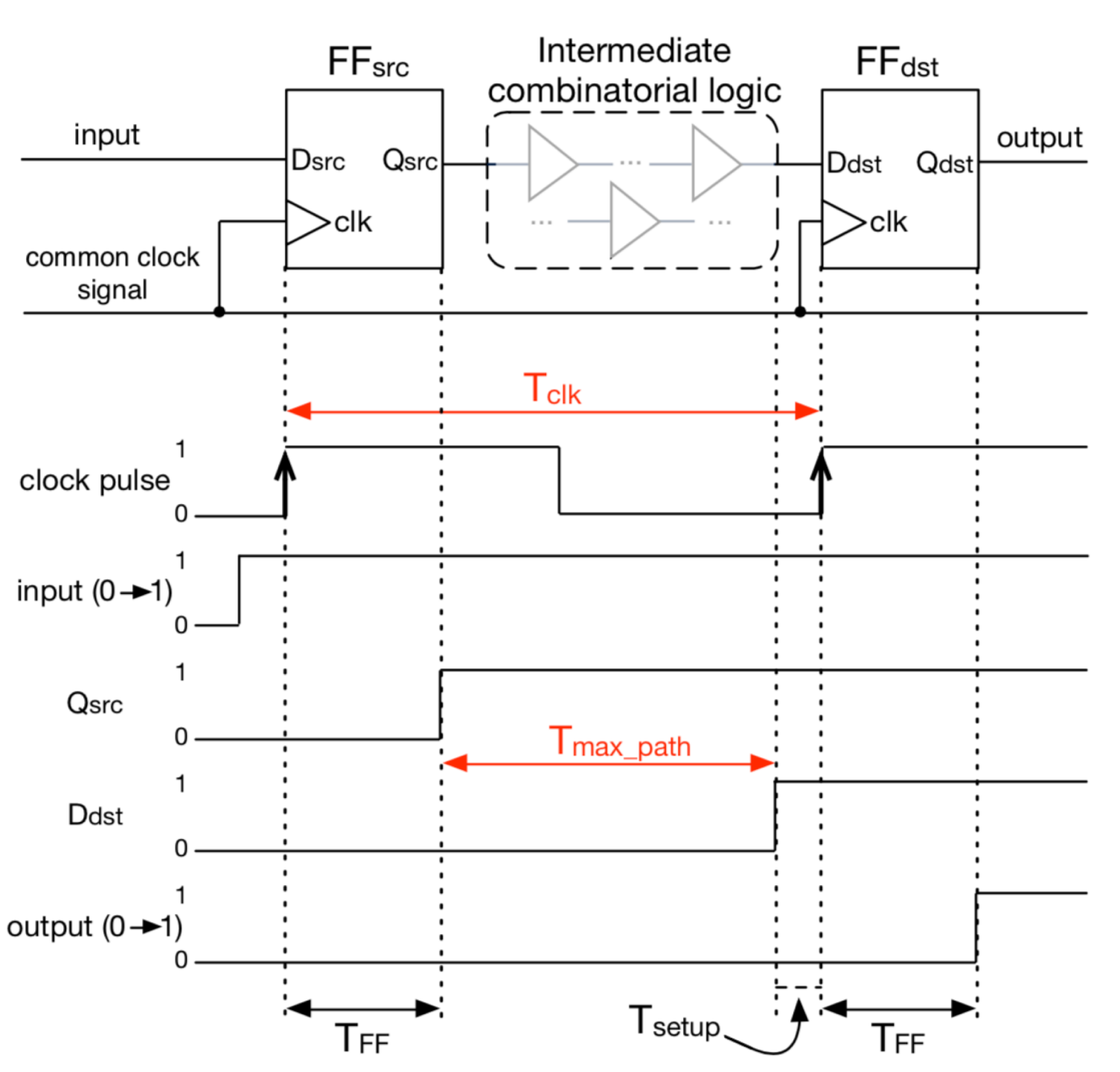
HARDWARE

DVFS
(Dynamic Voltage and Frequency Scaling)

SOFTWARE

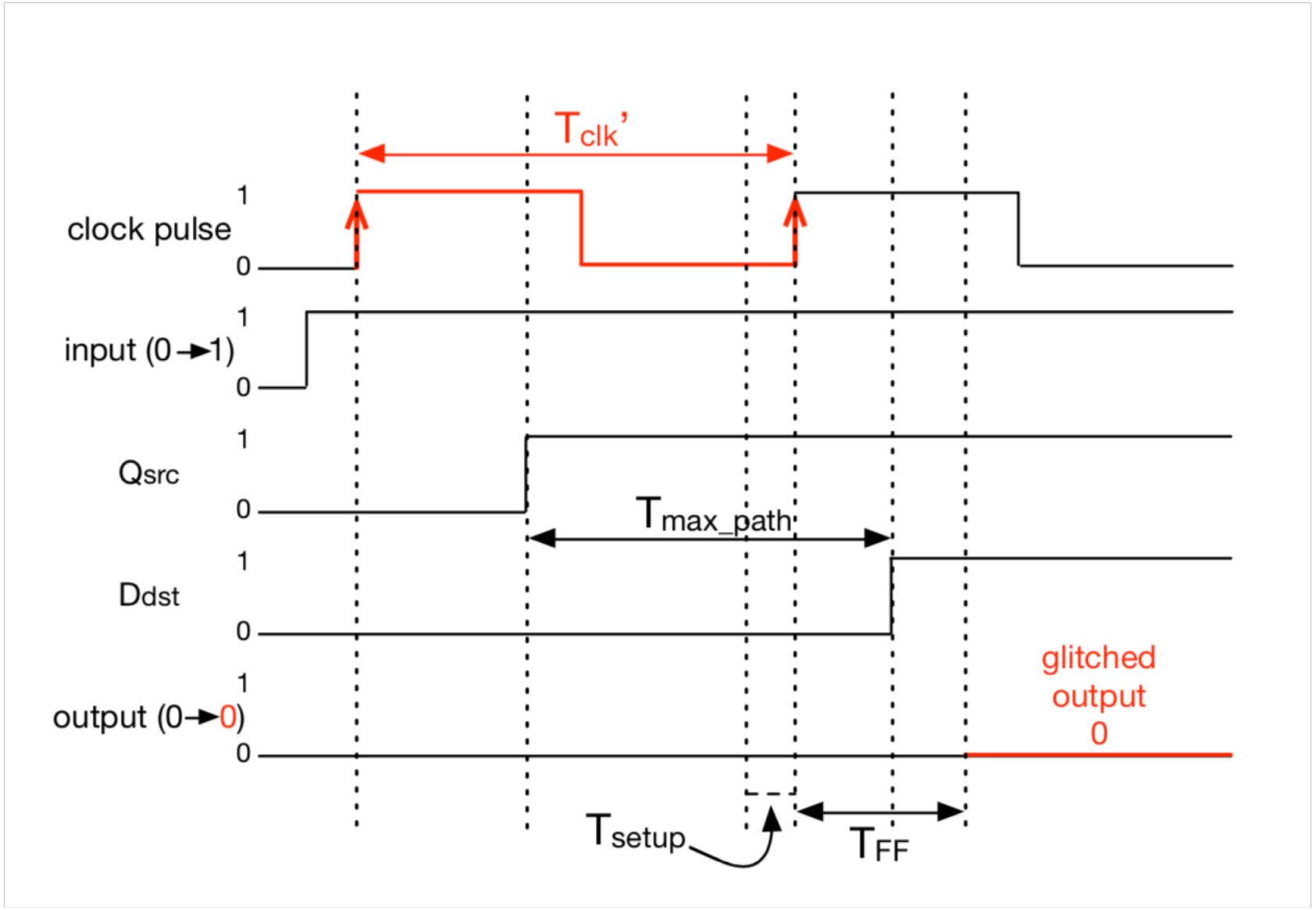

SoC Processor
(Nexus 6)
Voltage output
to other peripherals
Voltage output
to cores
0
1
2
3
PMA8084
PMIC
Input
Core 0
SPM
(All cores)
Voltage Domain
(All cores)
Voltage
Control

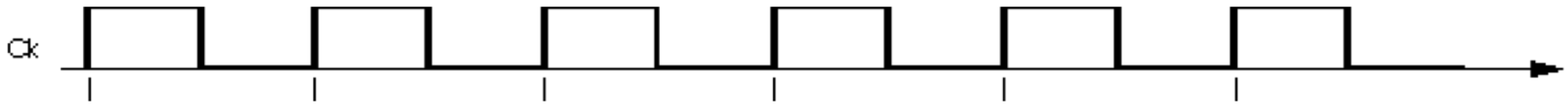

SoC Processor
(Nexus 6)
PLL
(fixed rate)
Clock Domain (per-core)
300 MHz
0
HFPLL
(variable rate)
N * 19.2 MHz
1
Clock
MUX
Core
Half
Divider
N/2 * 19.2 MHz
2
N Multiplier
Source Selector

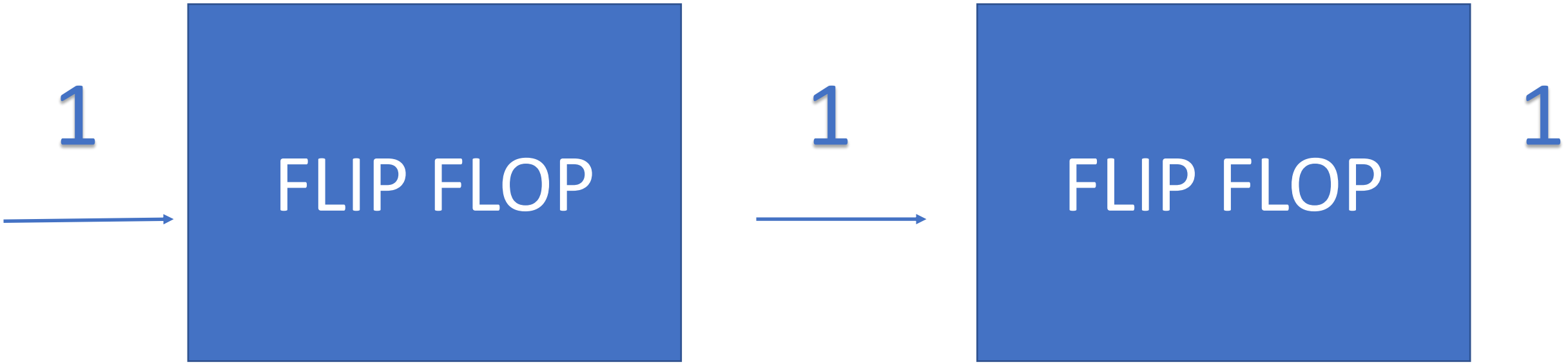

FFsrc
Intermediate combinatorial logic
FFdst
input
Dsrc
Qsrc
clk
Ddst
Qdst
clk
output
common clock signal
Tclk
clock pulse
1
0
input (0→1)
1
0
Qsrc
1
0
Tmax_path
Ddst
1
0
output (0→1)
1
0
TFF
Tsetup
TFF

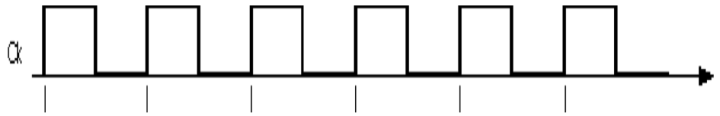
Outline

- Introduction to DVFS and background information.
- **What makes CLKSCREW unique?**
- Challenges to CLKSCREW
- Attacks and Results
- Conclusion

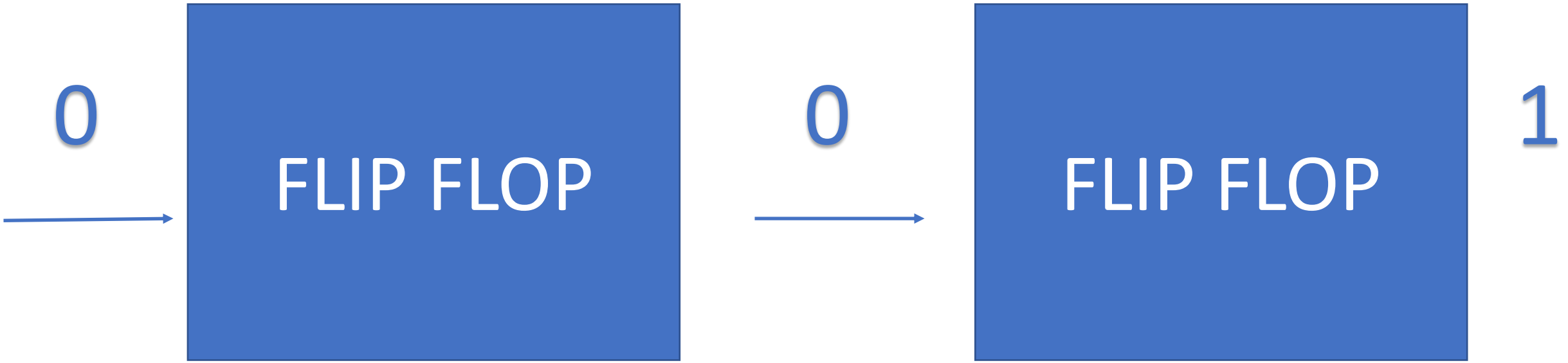

T_{clk}'
clock pulse
1
0
input (0→1)
1
0
Qsrc
1
0
T_{max_path}
Ddst
1
0
output (0→0)
1
0
glitched
output
0
T_{setup}
T_{FF}

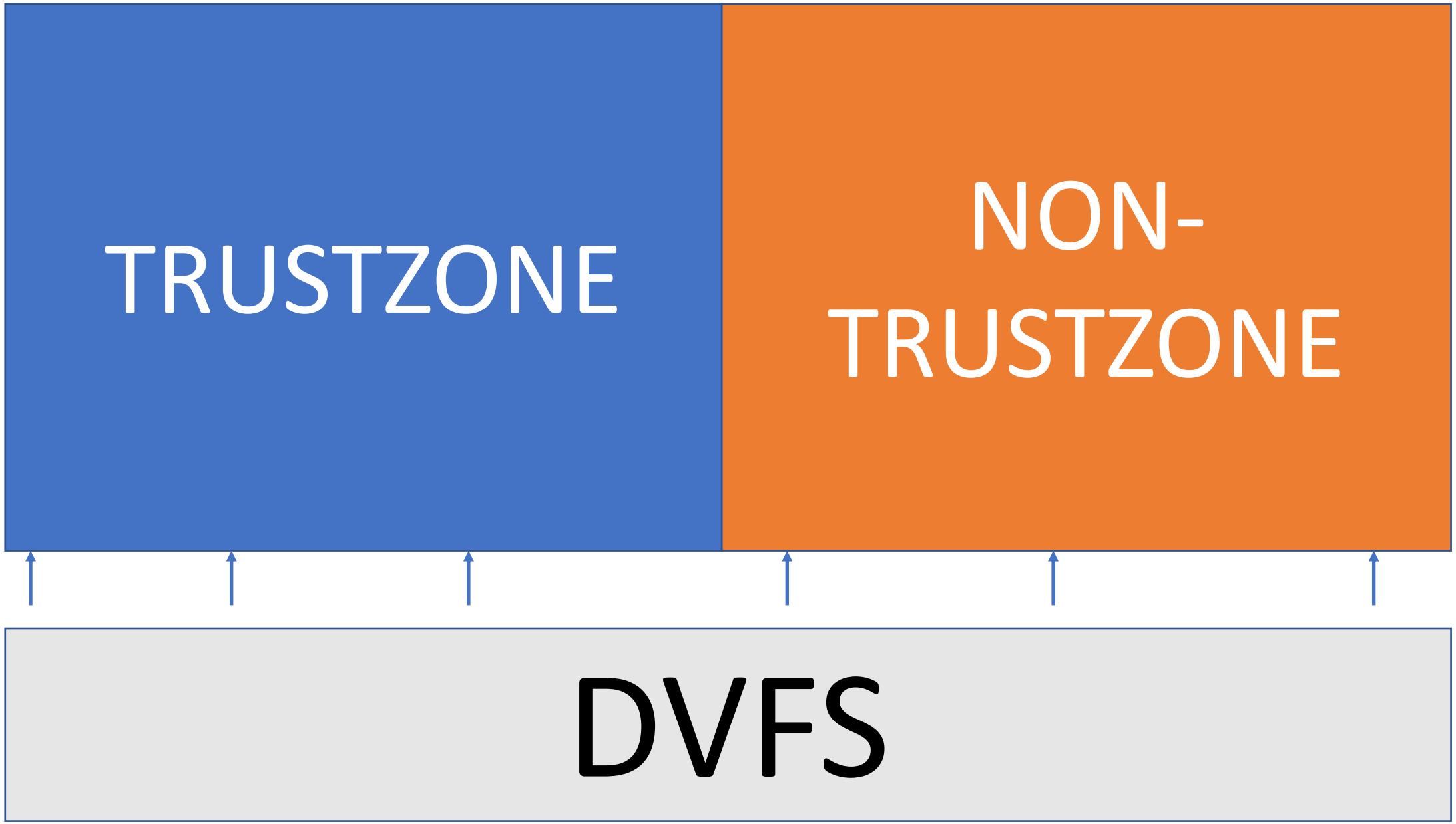

Ck


1
FLIP FLOP
1
FLIP FLOP
1



Less time for number to go through Flip-Flop




TRUSTZONE
NON-
TRUSTZONE
DVFS

Steps

1. Clear Residual States
2. Profile for Anchor
3. Pre-fault Delaying
4. Deliver the fault.

Outline

- Introduction to DVFS and background information.
- What makes CLKSCREW unique?
- **Challenges to CLKSCREW**
- Attacks and Results
- Conclusion

Do phones allow for overclocking/ under-volting?

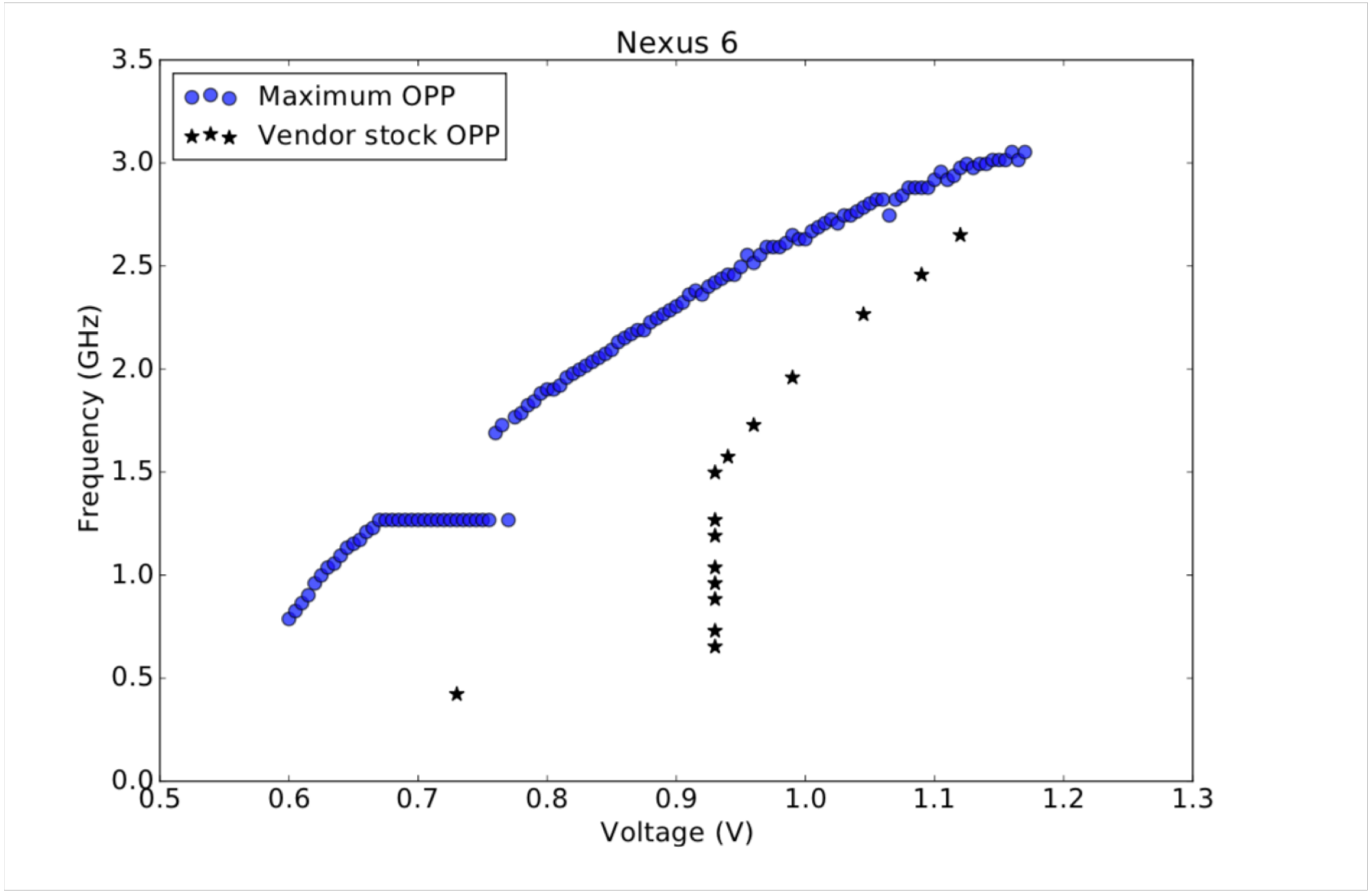
Nexus 6

nexus 6

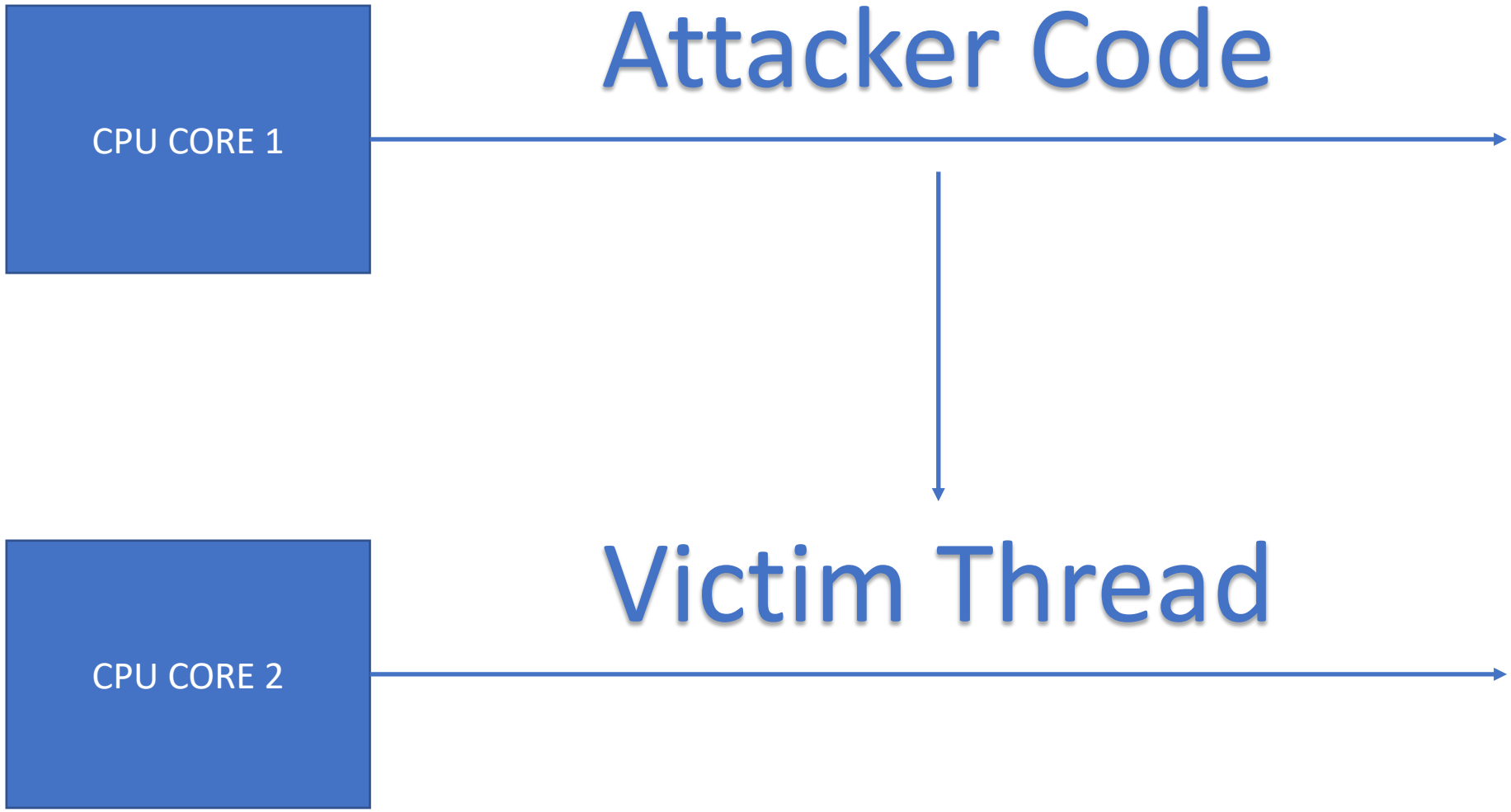
Front view of the Nexus 6

Codename	Shamu
Developer	Google and Motorola Mobility
Manufacturer	Motorola Mobility
Series	Google Nexus
Compatible networks	2G/3G/4G LTE GSM: 850/900/1800/1900 MHz **Model XT1103 (Americas):** CDMA band class: 0/1/10 WCDMA bands: 1/2/4/5/8 LTE bands:

CPU	Qualcomm 2.7 GHz quad-core Krait 450


Nexus 6
Maximum OPP
Vendor stock OPP
Frequency (GHz)
Voltage (V)
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3

How do you make sure the flip-flops do not damage the injected code?


CPU CORE 1
Attacker Code
Victim Thread
CPU CORE 2

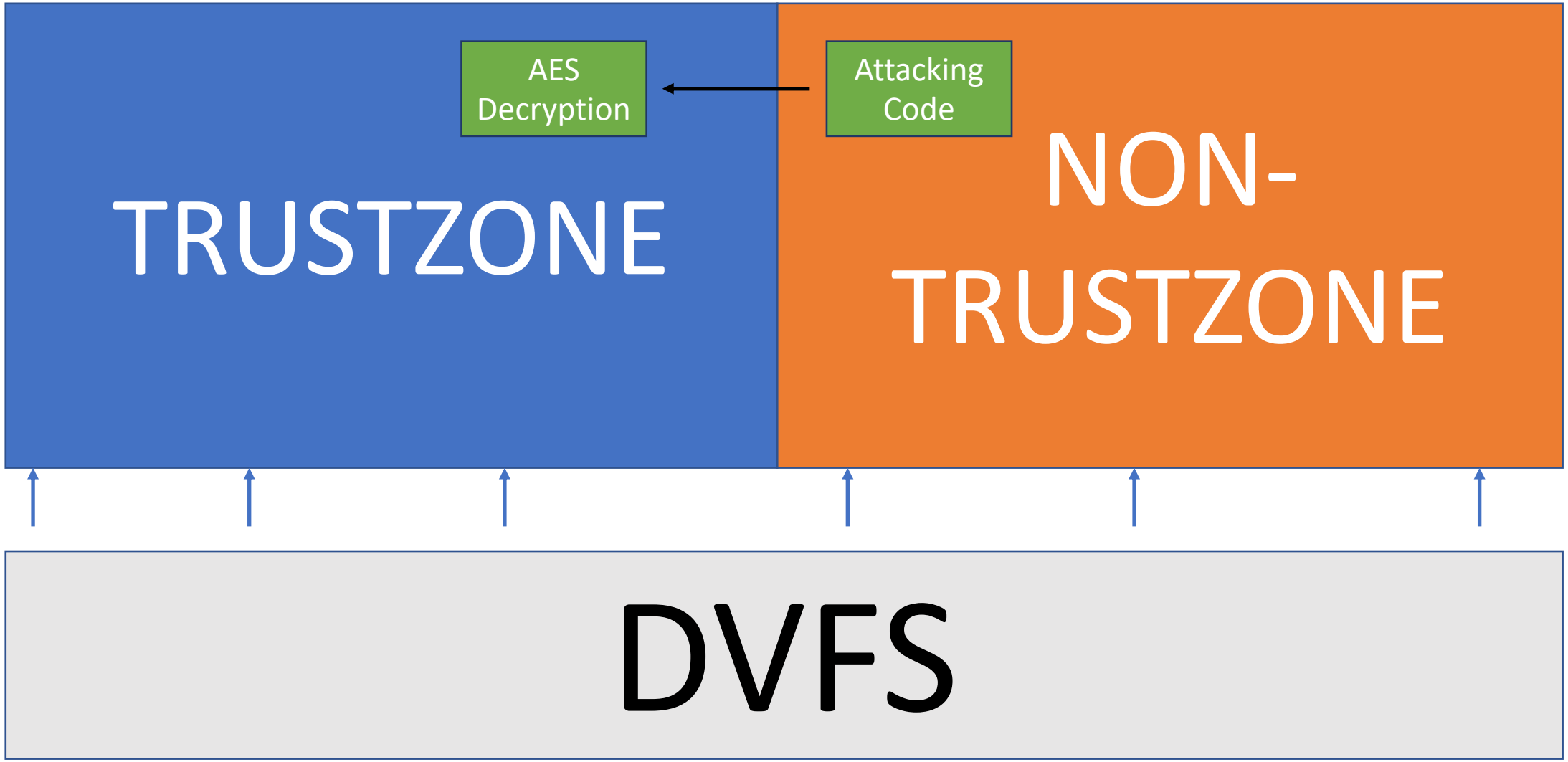
How do you get the timing precise enough?

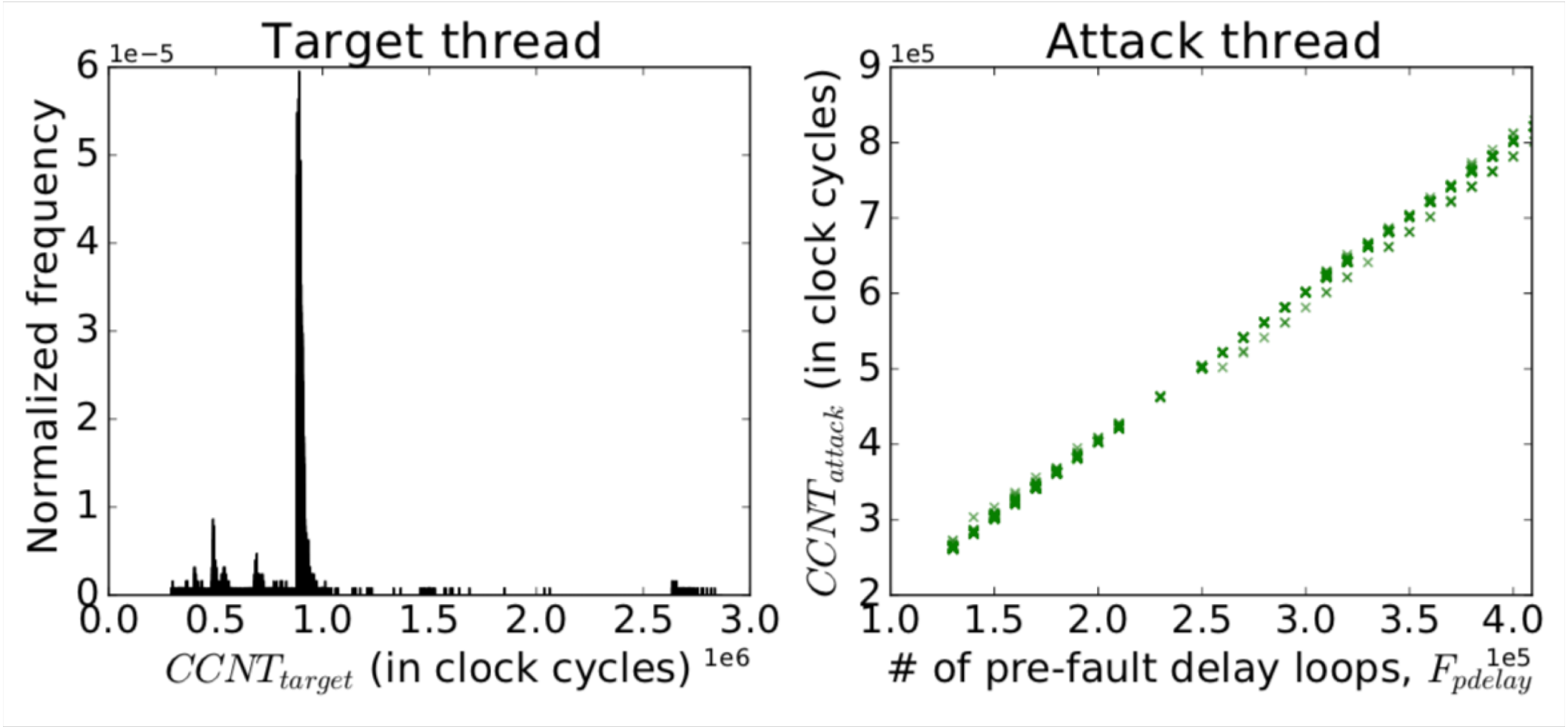
How do we make sure the attack occurs where we want it to occur?

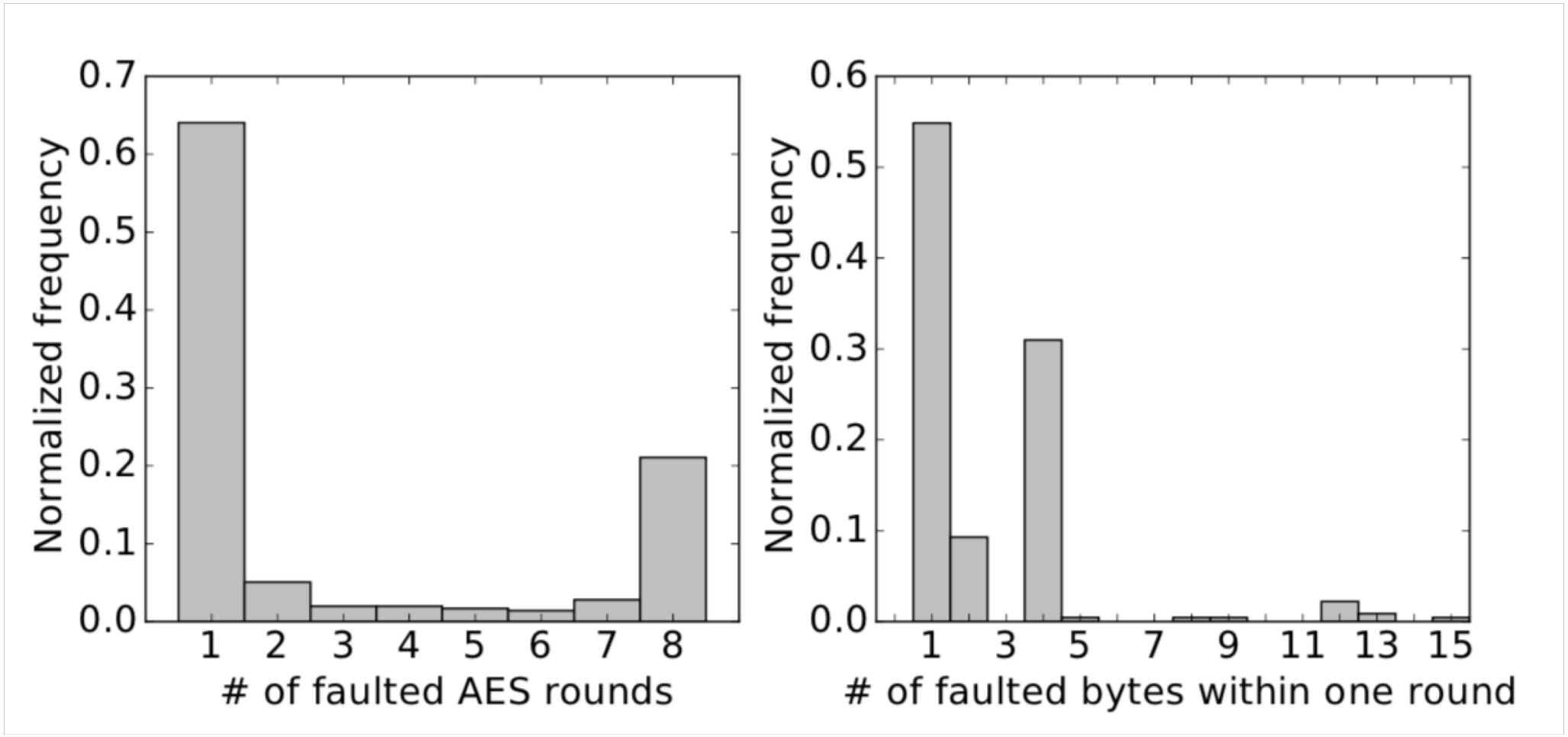
Outline

- Introduction to DVFS and background information.
- What makes CLKSCREW unique?
- Challenges to CLKSCREW
- **Attacks and Results**
- Conclusion

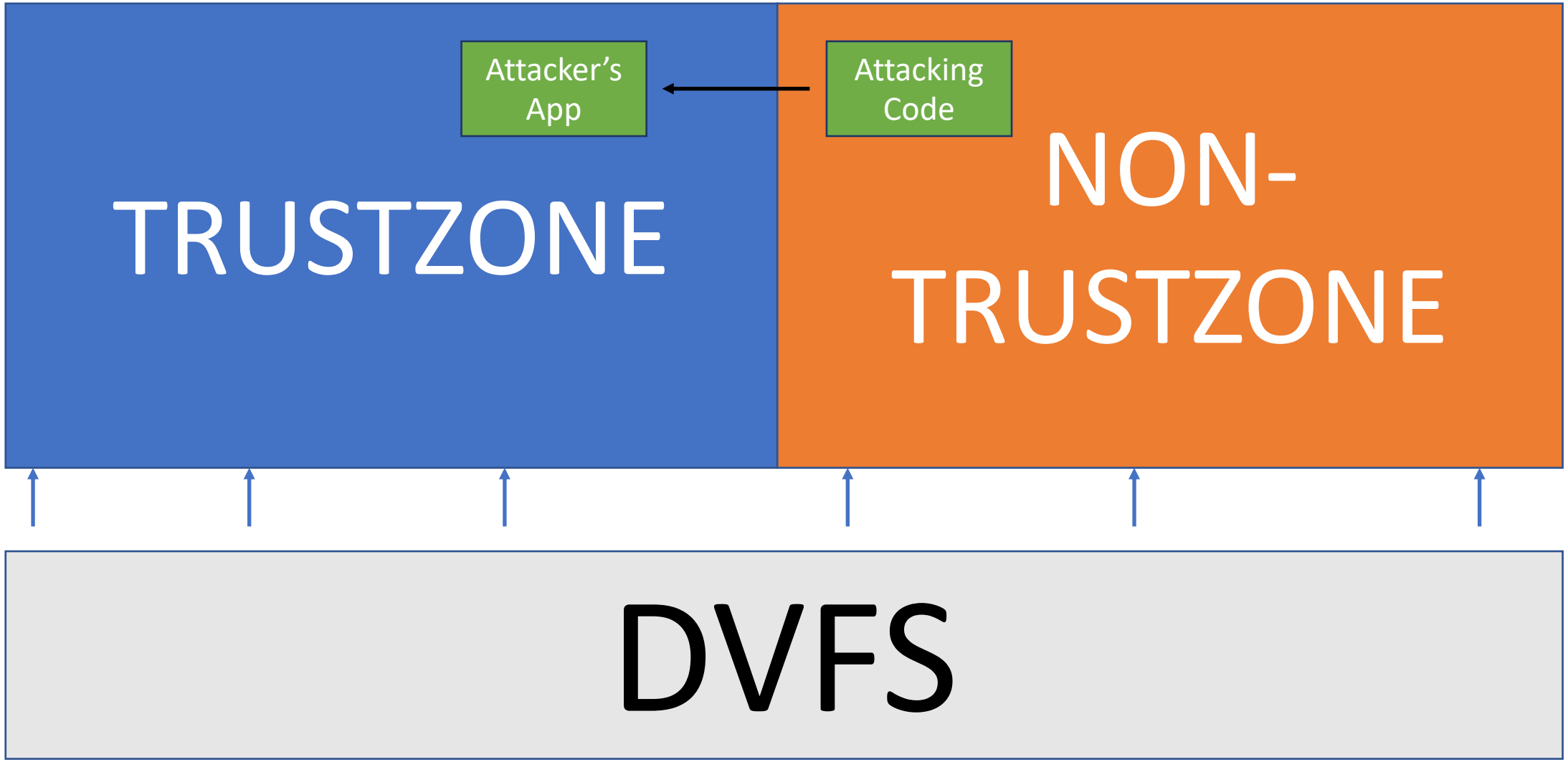
Inferring AES Keys




Target thread
1e−5
Normalized frequency
0
1
2
3
4
5
6
0.0
0.5
1.0
1.5
2.0
2.5
3.0
$CCNT_{target}$ (in clock cycles)
1e6
Attack thread
1e5
$CCNT_{attack}$ (in clock cycles)
2
3
4
5
6
7
8
9
1.0
1.5
2.0
2.5
3.0
3.5
4.0
# of pre-fault delay loops, F_{pdelay}
1e5


Normalized frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
2
3
4
5
6
7
8
# of faulted AES rounds
Normalized frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
3
5
7
9
11
13
15
# of faulted bytes within one round

Loading Apps into Trust Zone



- Each App has 4 Signatures
- One signature takes 270 Million clock cycles to validate.
- In order for CLKSCREW to corrupt data it needs to change just 65 thousand clock cycles within the entire process

65000/10800000000

=

0.0000601%

Cache Profiling

- Pick a memory address of the area of interest
- Run dummy instructions and time the amount it takes for these instructions to be removed
- Patterns for removing will tell you the pattern of the actual code.

Timing Anchor

- Track duration of consecutive cache instructions

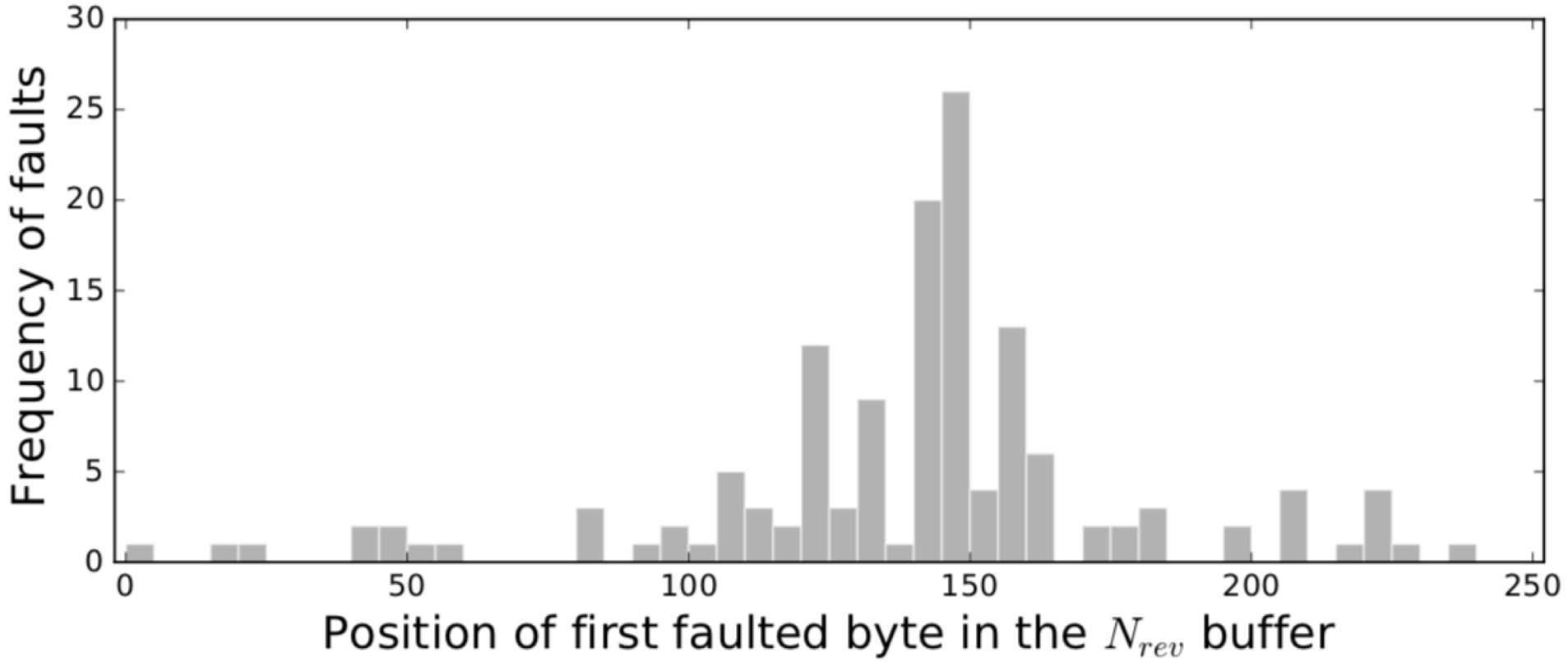


Figure 14: Histogram of observed faults and where the faults occur. The intended faulted position is 141.

One instance of Desired Fault out of 65

Outline

- Introduction to DVFS and background information.
- What makes CLKSCREW unique?
- Challenges to CLKSCREW
- Attacks and Results
- **Conclusion**

Defenses

Hardware Limits regarding Voltage and Frequency

- Make it unable for users to overclock and under-volt their phones
- Difficulties include having to remake hardware chips from scratch and having every phone and chipmaker adhere to regulation.

Separate DVFS for Trustzone

- Create a separate DVFS for Trustzone itself
- Separate DVFS' for cores on the same chip can cause massive overhead.

Randomization

- Randomize clock cycles so that attackers do not know what to expect.
- Useless when run-time time-anchors are used.

Conclusions

- CLKSCREW is a side-channel attack that utilizes voltage and frequency of devices to induce faults.

- Exploiting faults that cannot be easily changed.