Turn on, Tune in, Listen up

Maximizing Side-Channel Recovery in Time-to-Digital Converters

Colin Drewes[1], Olivia Weng[1], Keegan Ryan[1], Bill Hunter[2], Christopher McCarty[2], Ryan Kastner[1], and Dustin Richmond[3]

[1]UCSD, [2]GTRI, [3]UCSC

February 13, 2023

FPGAs are powerful...but often prohibitively expensive

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

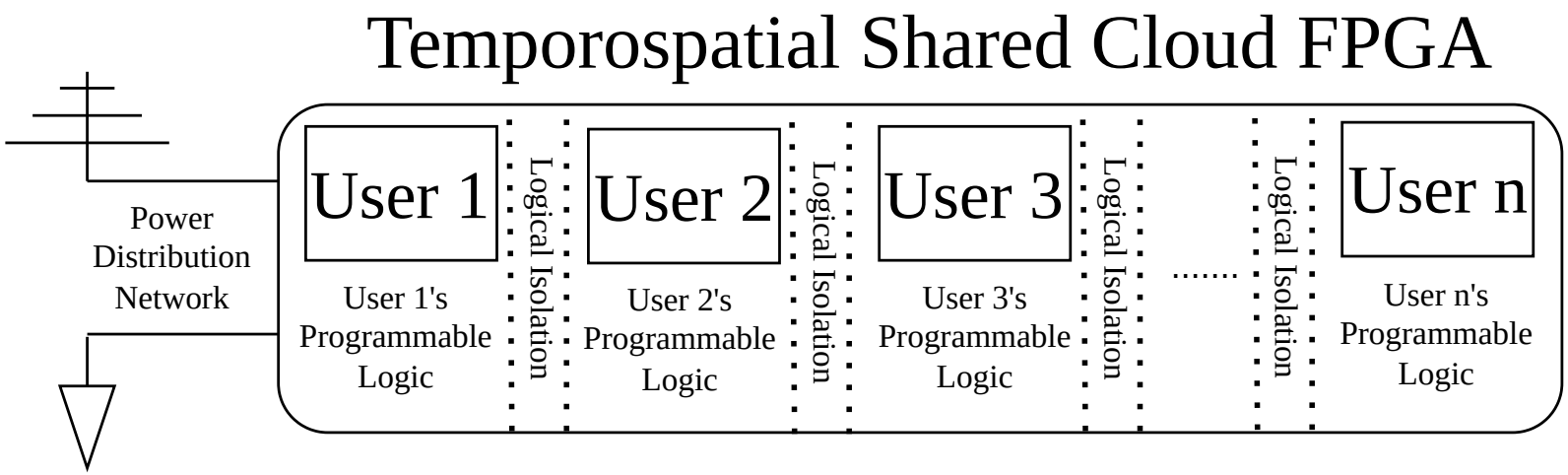
Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.



FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

While logical boundaries may be in place, they all share the same power system!

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

While logical boundaries may be in place, they all share the same power system!

If a dedicated actor were able to instantiate a voltage fluctuation sensor within their allocated logic they could...

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

While logical boundaries may be in place, they all share the same power system!

If a dedicated actor were able to instantiate a voltage fluctuation sensor within their allocated logic they could...

- Determine what core other actors are executing

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

While logical boundaries may be in place, they all share the same power system!

If a dedicated actor were able to instantiate a voltage fluctuation sensor within their allocated logic they could...

- Determine what core other actors are executing
- Determine what is running on an actor's soft-processor

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

While logical boundaries may be in place, they all share the same power system!

If a dedicated actor were able to instantiate a voltage fluctuation sensor within their allocated logic they could...

- Determine what core other actors are executing
- Determine what is running on an actor's soft-processor
- Perform a correlation power analysis attack

FPGAs are powerful...but often prohibitively expensive

This has pushed us towards...

- Cloud-FPGA leasing schemes (temporal sharing)
- Integration of proprietary IP
- Full FPGA virtualization (temporospatial sharing)

Observation: These are all multi-actor structures—potentially running hidden opaque circuitry.

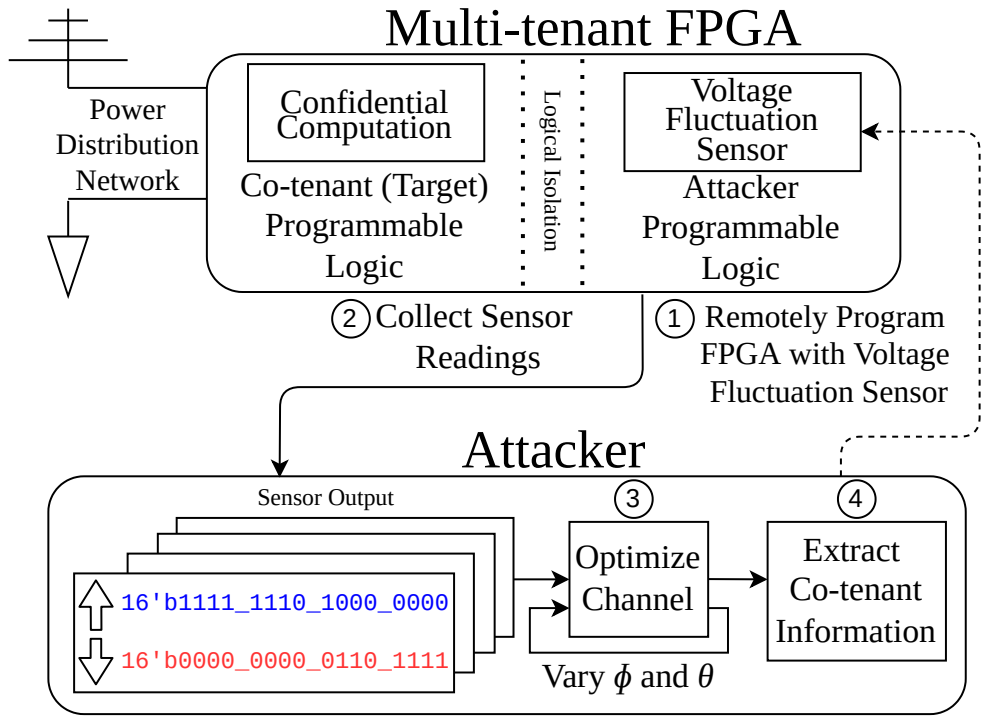
While logical boundaries may be in place, they all share the same power system!

If a dedicated actor were able to instantiate a voltage fluctuation sensor within their allocated logic they could...

- Determine what core other actors are executing
- Determine what is running on an actor's soft-processor
- Perform a correlation power analysis attack
- Detect round 7 of AES to introduce a power fault into the system

Temporospatial (Multi-Tenant) Cloud FPGAs

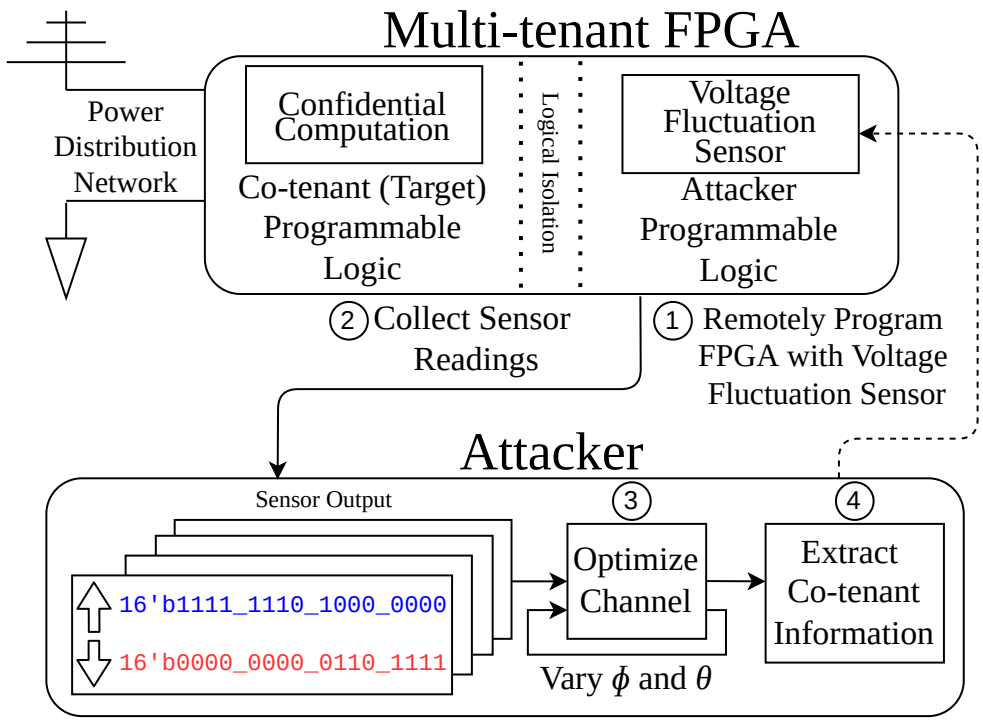
We refine our threat model around cloud multi-tenant FPGAs—but we study fundamental qualities of voltage fluctuation sensors that can apply to any multi-actor structure.



Temporospatial (Multi-Tenant) Cloud FPGAs

We refine our threat model around cloud multi-tenant FPGAs—but we study fundamental qualities of voltage fluctuation sensors that can apply to any multi-actor structure.

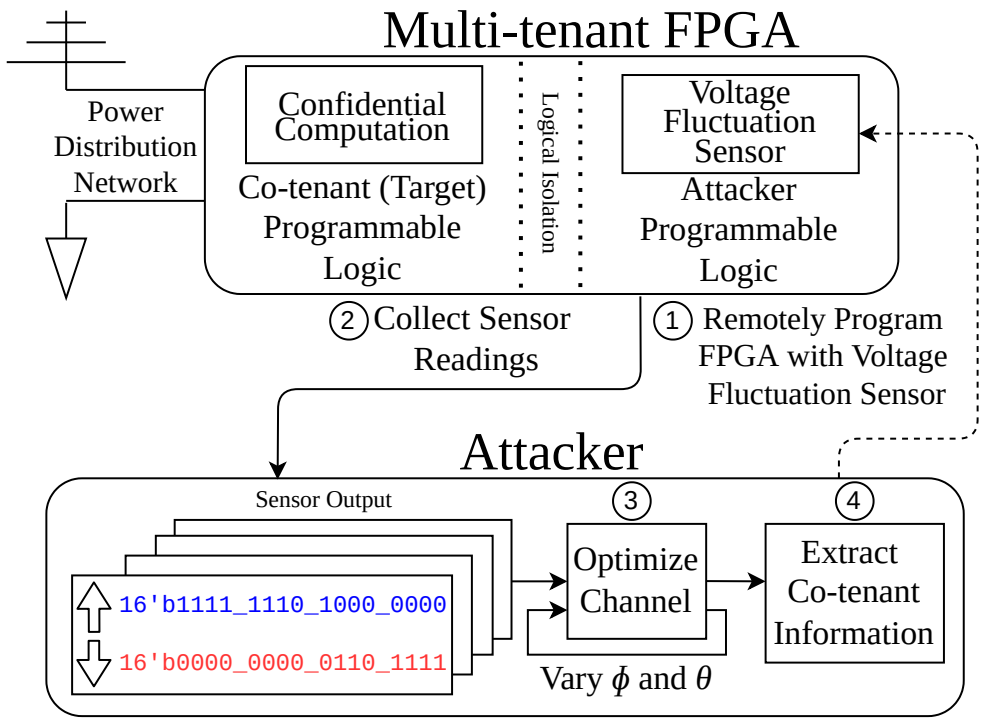
1. Attacker co-locates with another user and deploys voltage sensor



Temporospatial (Multi-Tenant) Cloud FPGAs

We refine our threat model around cloud multi-tenant FPGAs—but we study fundamental qualities of voltage fluctuation sensors that can apply to any multi-actor structure.

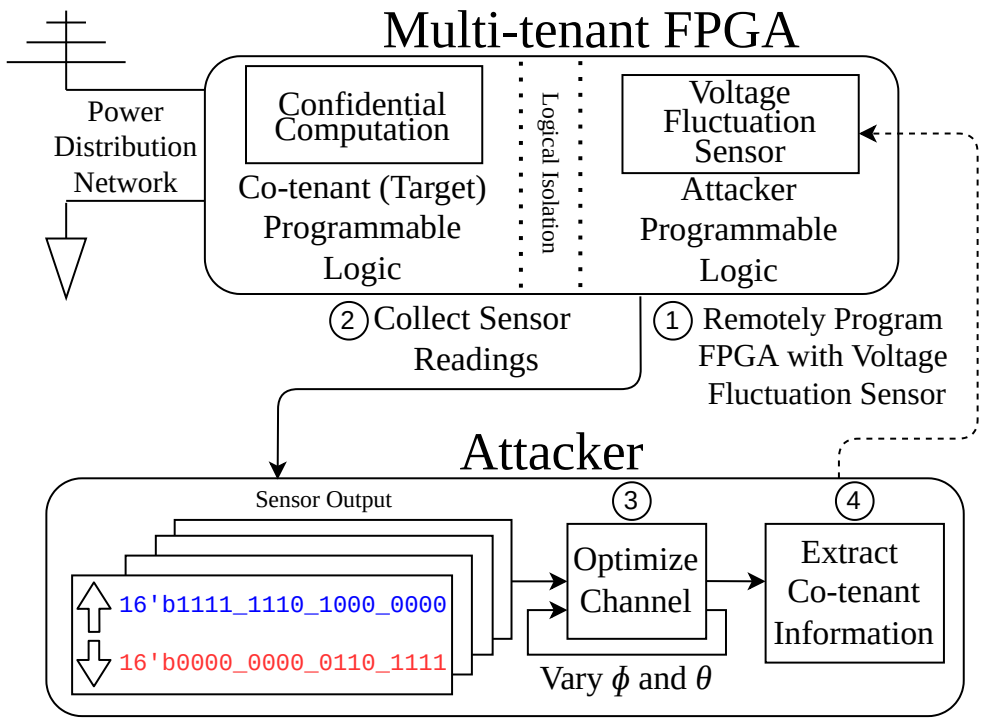
1. Attacker co-locates with another user and deploys voltage sensor
2. Collects sensor readings



Temporospatial (Multi-Tenant) Cloud FPGAs

We refine our threat model around cloud multi-tenant FPGAs—but we study fundamental qualities of voltage fluctuation sensors that can apply to any multi-actor structure.

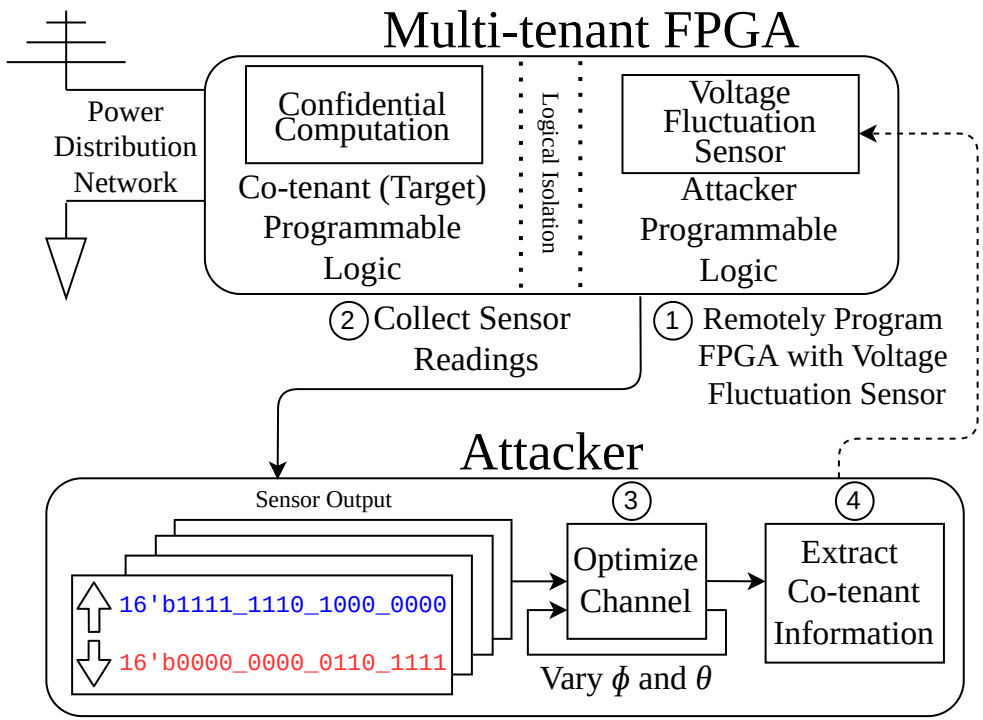
1. Attacker co-locates with another user and deploys voltage sensor
2. Collects sensor readings
3. Optimize θ and ϕ based on readings



Temporospatial (Multi-Tenant) Cloud FPGAs

We refine our threat model around cloud multi-tenant FPGAs—but we study fundamental qualities of voltage fluctuation sensors that can apply to any multi-actor structure.

1. Attacker co-locates with another user and deploys voltage sensor
2. Collects sensor readings
3. Optimize θ and ϕ based on readings
4. Extract data from sensor readings



Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network
2. Easy to disguise as they resemble carry-adders and can be designed to include no timing violations

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network
2. Easy to disguise as they resemble carry-adders and can be designed to include no timing violations

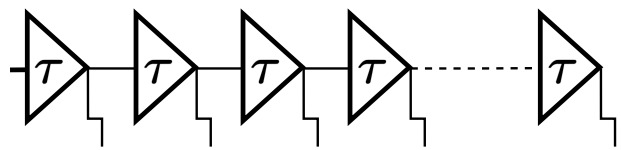
A brief overview of a canonical TDC.

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network
2. Easy to disguise as they resemble carry-adders and can be designed to include no timing violations

A brief overview of a canonical TDC.



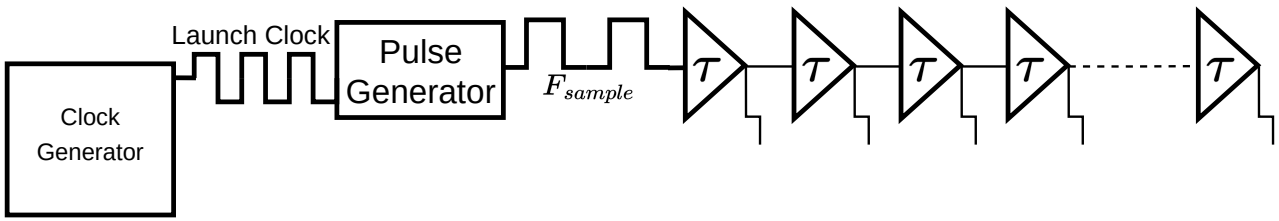
*An array of elements with uniform delay

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network
2. Easy to disguise as they resemble carry-adders and can be designed to include no timing violations

A brief overview of a canonical TDC.



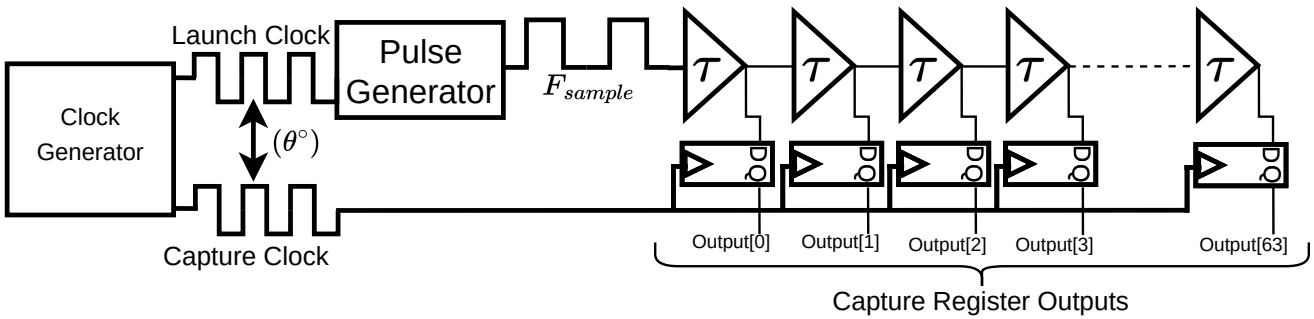
*Launch clock sends rising/falling pulses through delay elements

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network
2. Easy to disguise as they resemble carry-adders and can be designed to include no timing violations

A brief overview of a canonical TDC.



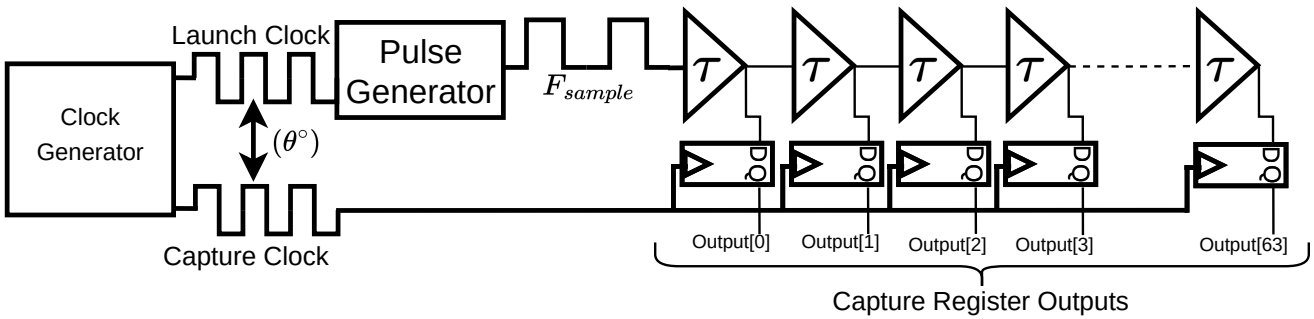
*Capture clock, which is phase offset a constant value from the launch, captures how far signals propagated through delay elements

Voltage Fluctuation Sensors

We are interested in Time-to-Digital Converters (TDC):

1. An instrument for synchronously measuring the load on the power distribution network
2. Easy to disguise as they resemble carry-adders and can be designed to include no timing violations

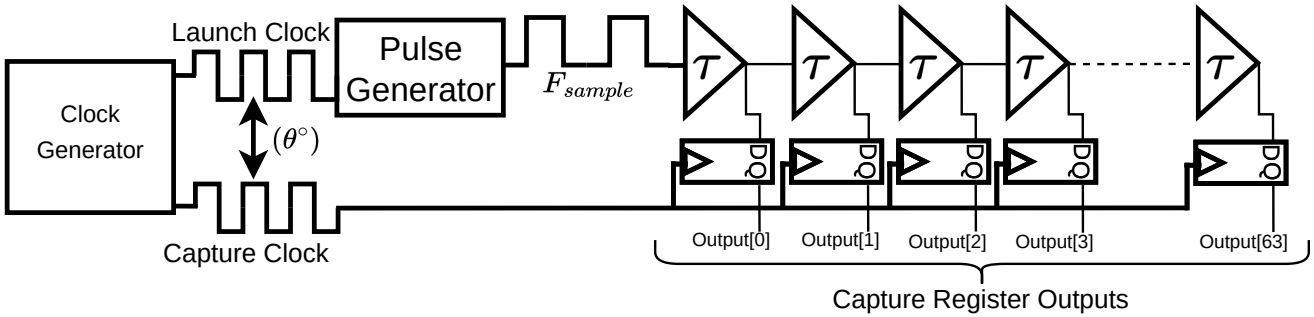
A brief overview of a canonical TDC.



Capture output distance $\propto$ state of the PDN

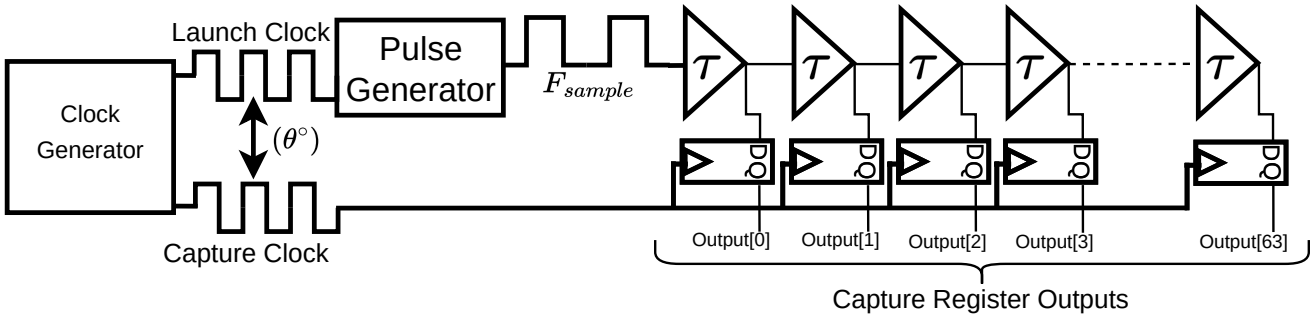
Known Problems: Architectural Irregularities

This is a nice theoretical model:



Known Problems: Architectural Irregularities

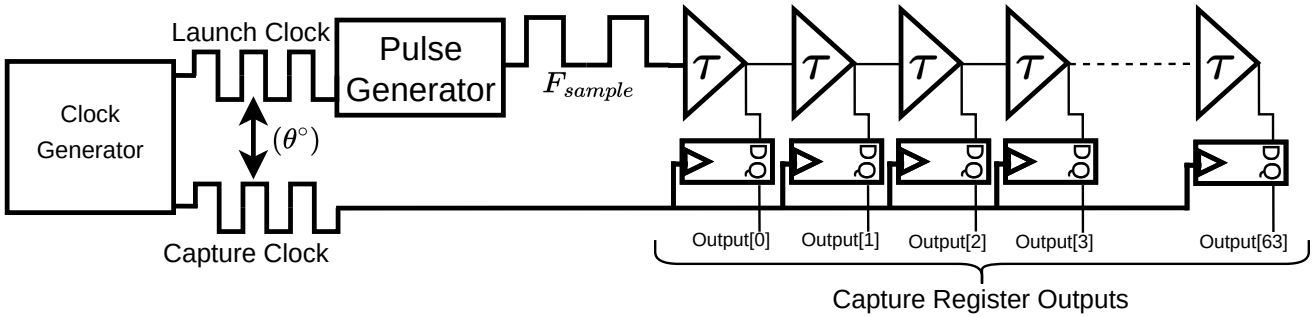
This is a nice theoretical model:



Perfect delay line doesn't exist:

Known Problems: Architectural Irregularities

This is a nice theoretical model:

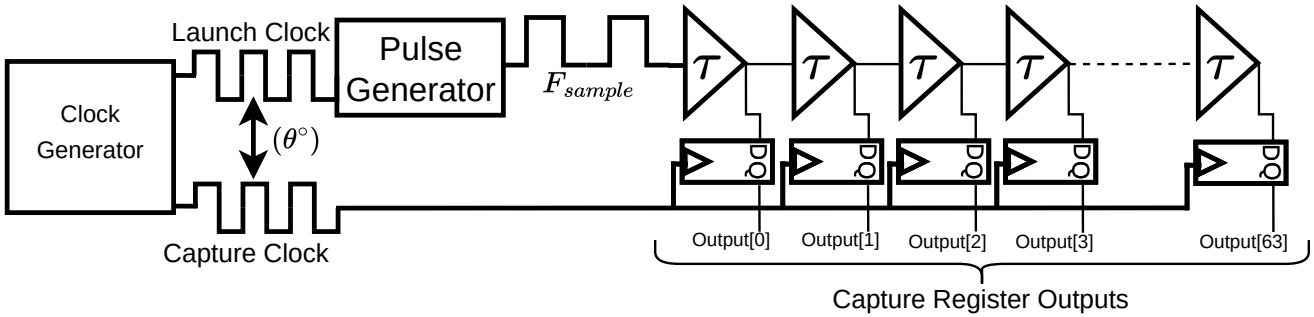


Perfect delay line doesn't exist:

- More linear propagation = more responsive to fluctuations

Known Problems: Architectural Irregularities

This is a nice theoretical model:



Perfect delay line doesn't exist:

- More linear propagation = more responsive to fluctuations
- Faster propagation = higher resolution

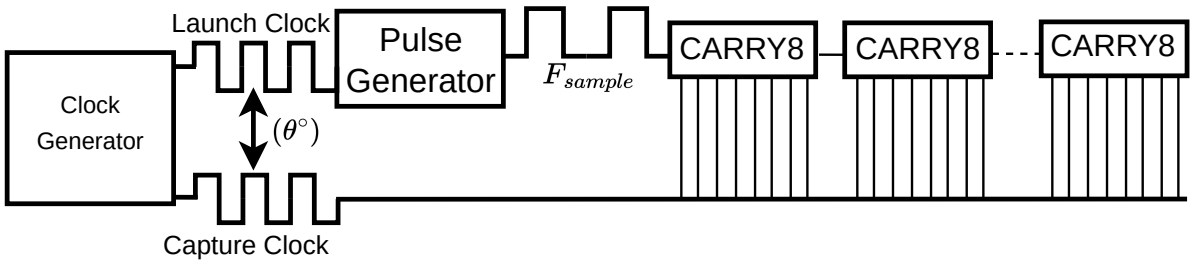
Known Problems: Architectural Irregularities

Known Problems: Architectural Irregularities

Map TDC to CARRY primitives. Fast and linear....locally.

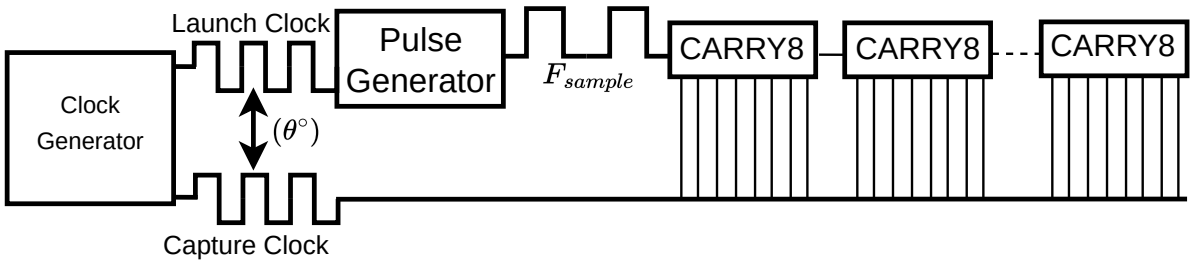
Known Problems: Architectural Irregularities

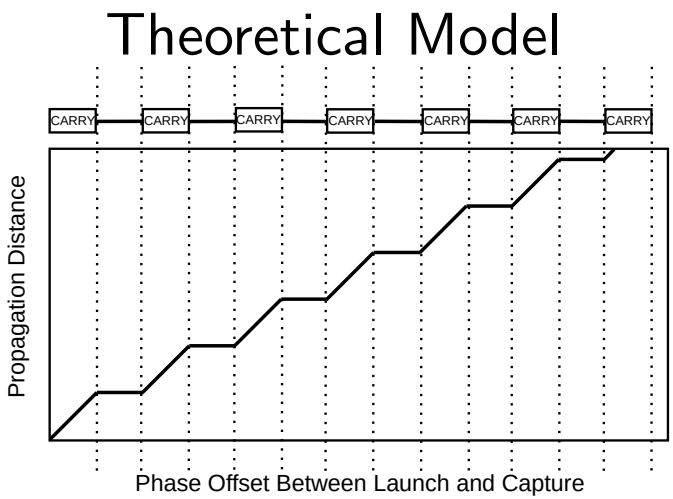
Map TDC to CARRY primitives. Fast and linear....locally.



Known Problems: Architectural Irregularities

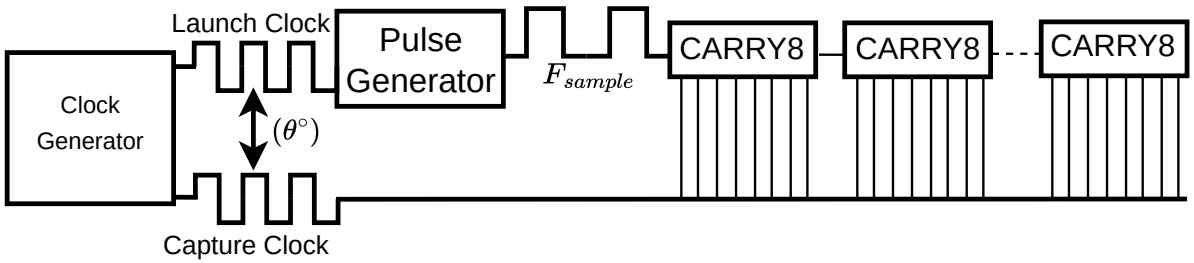
Map TDC to CARRY primitives. Fast and linear....locally.



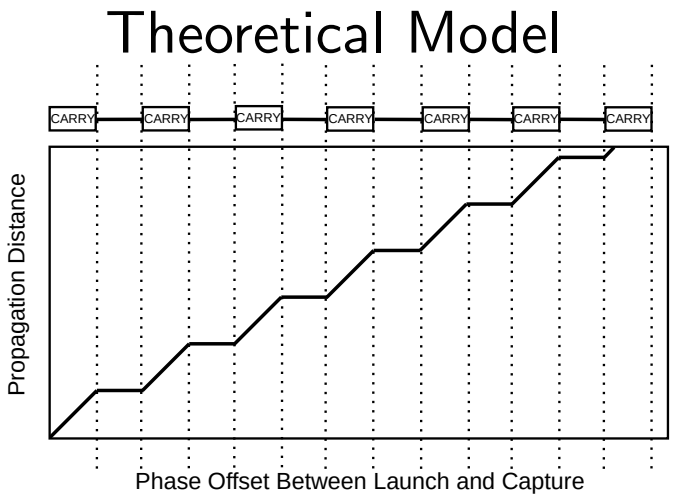


Known Problems: Architectural Irregularities

Map TDC to CARRY primitives. Fast and linear....locally.

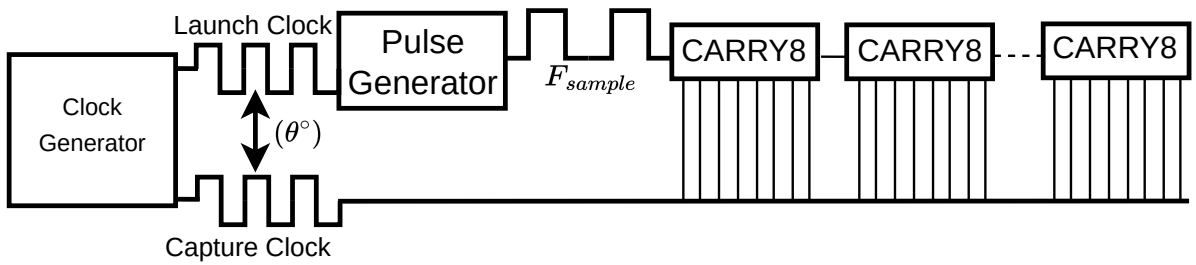


The data bears this out. Linear 4 bits on PYNQ (CARRY4s), linear 8 bits on AWS (CARRY8s).



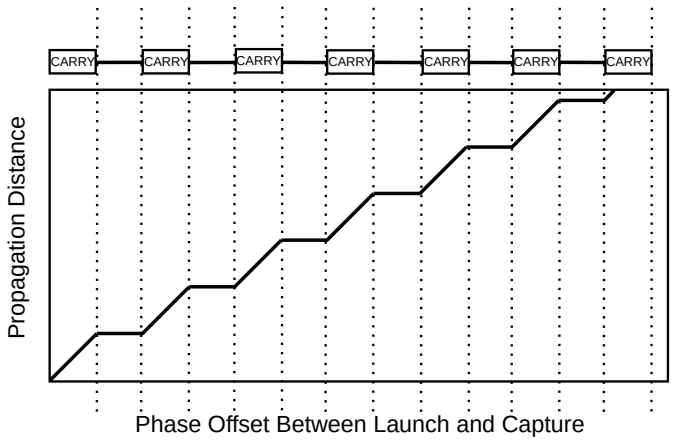
Known Problems: Architectural Irregularities

Map TDC to CARRY primitives. Fast and linear....locally.

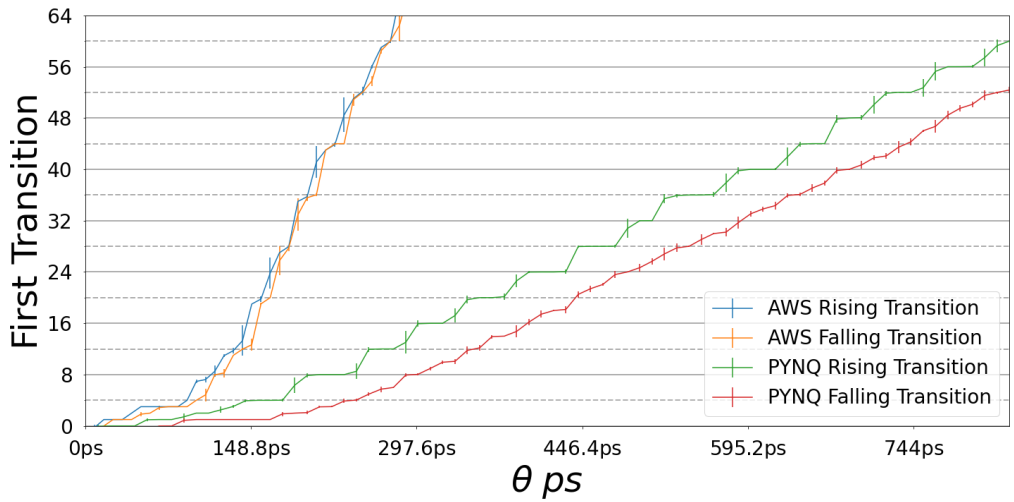


The data bears this out. Linear 4 bits on PYNQ (CARRY4s), linear 8 bits on AWS (CARRY8s).

Theoretical Model

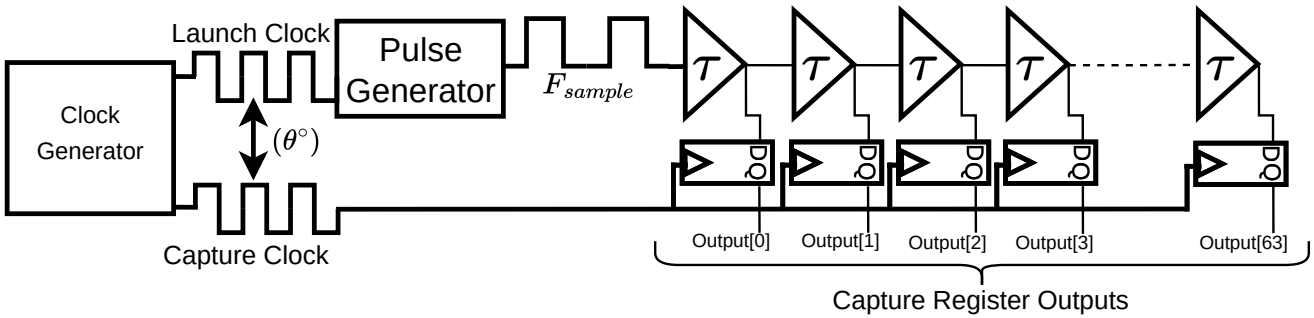


Physical Validation



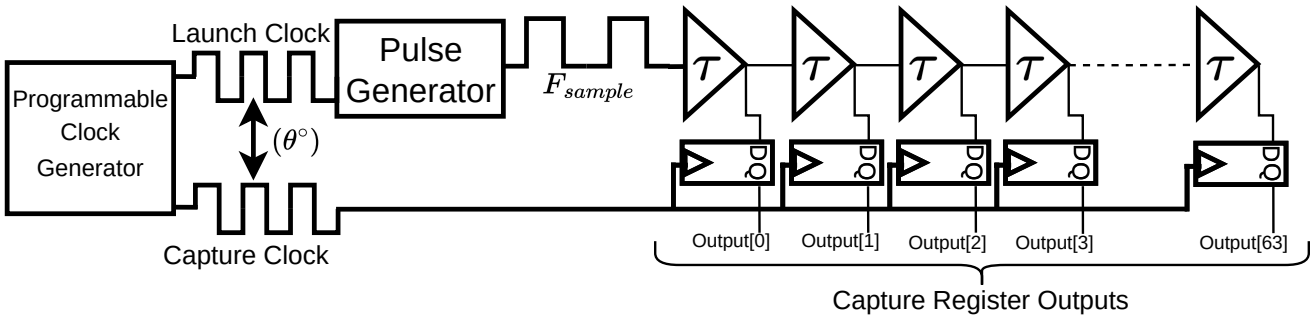
Solution: Dynamic θ Tuning

Constant phase offset.



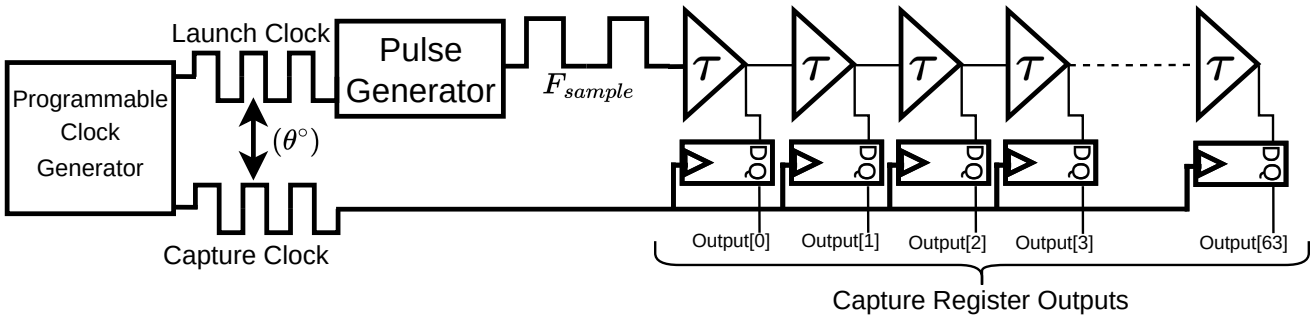
Solution: Dynamic θ Tuning

Dynamic phase offset.



Solution: Dynamic θ Tuning

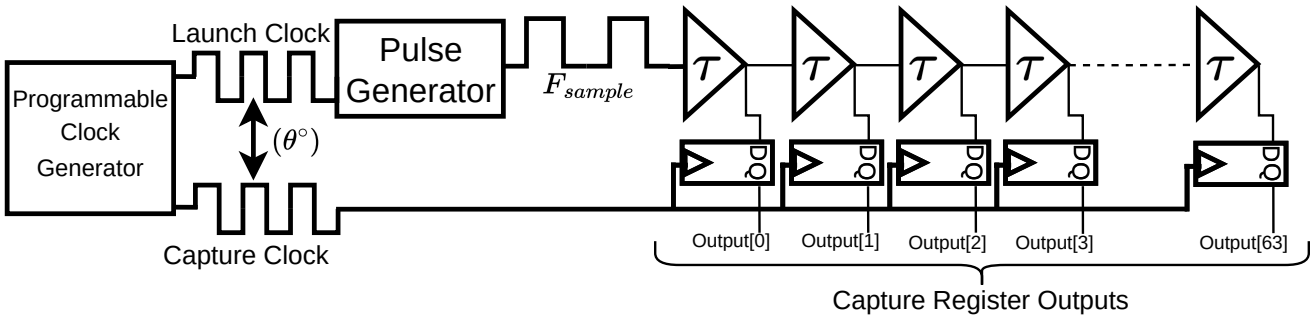
Dynamic phase offset.



- Leverage programmable clock generator

Solution: Dynamic θ Tuning

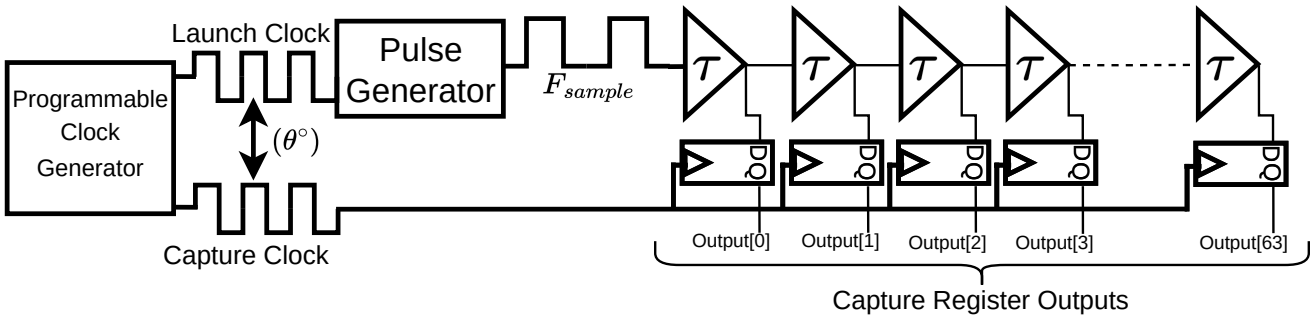
Dynamic phase offset.



- Leverage programmable clock generator
- Propagation distance is now configurable

Solution: Dynamic θ Tuning

Dynamic phase offset.



- Leverage programmable clock generator
- Propagation distance is now configurable
- Attacker can place transition at ideal point within CARRY primitive, improving sensor response to power fluctuations.

A New Problem: Sampling Phase Alignment

- θ, the phase relationship between launch and capture defines a **window**

A New Problem: Sampling Phase Alignment

- θ, the phase relationship between launch and capture defines a **window**
- Only power fluctuations within window are captured by the TDC

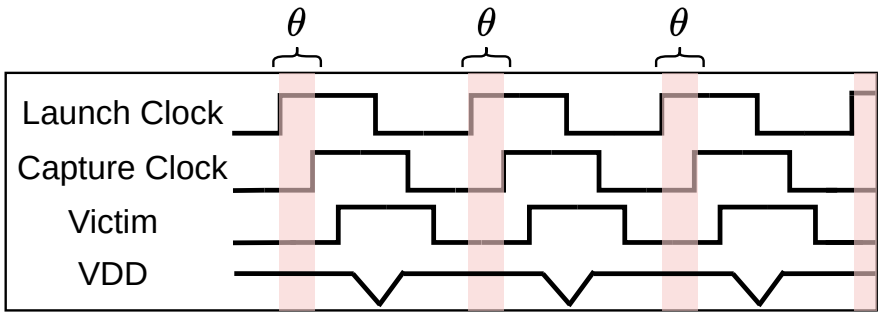
A New Problem: Sampling Phase Alignment

- θ, the phase relationship between launch and capture defines a **window**
- Only power fluctuations within window are captured by the TDC
- Victim core is also likely synchronous, and induced power fluctuation is brief

A New Problem: Sampling Phase Alignment

- θ, the phase relationship between launch and capture defines a **window**
- Only power fluctuations within window are captured by the TDC
- Victim core is also likely synchronous, and induced power fluctuation is brief

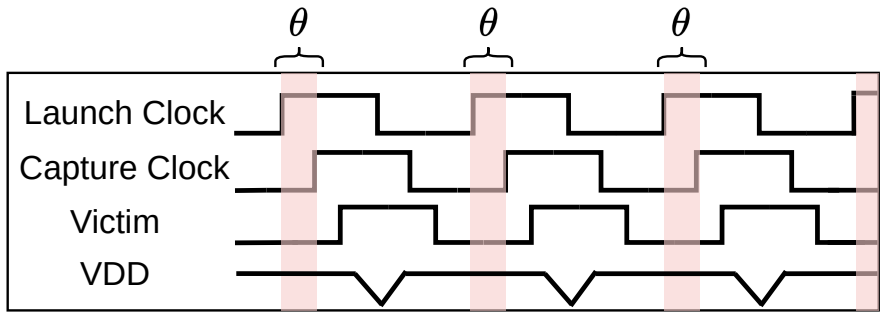
Bad Phase (ϕ) Alignment!



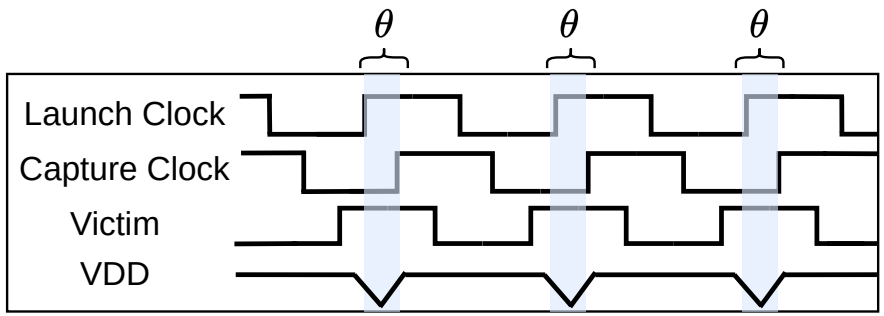
A New Problem: Sampling Phase Alignment

- θ, the phase relationship between launch and capture defines a **window**
- Only power fluctuations within window are captured by the TDC
- Victim core is also likely synchronous, and induced power fluctuation is brief

Bad Phase (ϕ) Alignment!



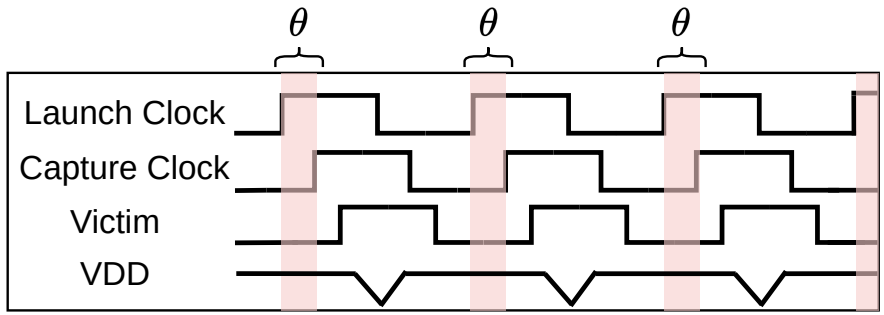
Good Phase (ϕ) Alignment!



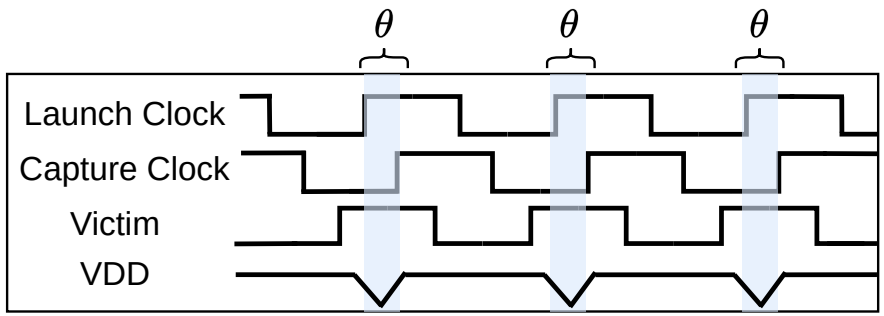
A New Problem: Sampling Phase Alignment

- θ, the phase relationship between launch and capture defines a **window**
- Only power fluctuations within window are captured by the TDC
- Victim core is also likely synchronous, and induced power fluctuation is brief
- Lets phase shift launch and capture clock together, wrt victim computation

Bad Phase (ϕ) Alignment!

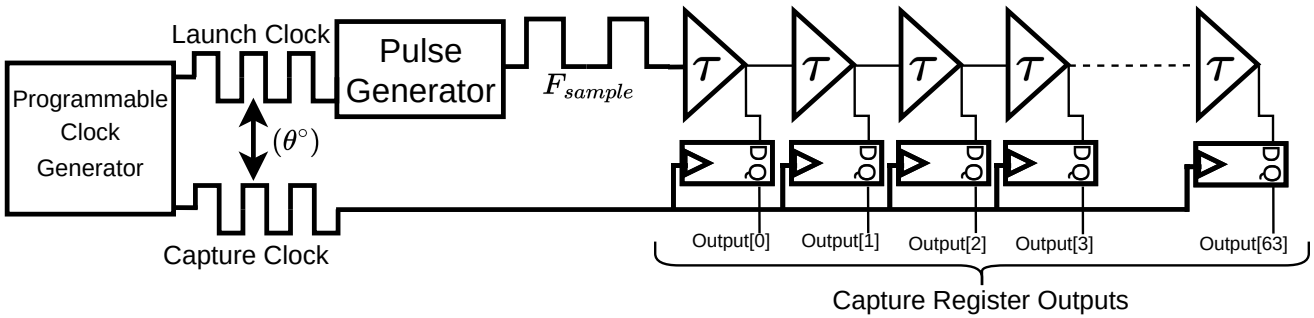


Good Phase (ϕ) Alignment!



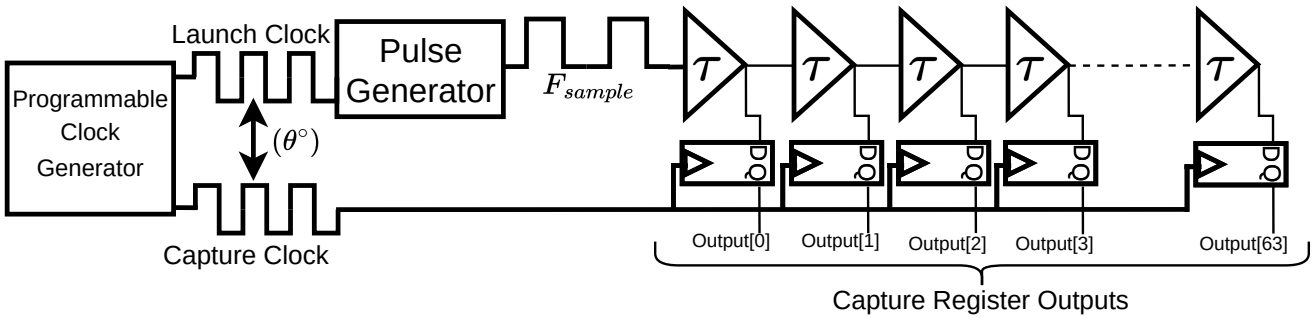
Solution: Dynamic Co-Tenant Phase Alignment

Our θ tunable Time-to-Digital Converter:

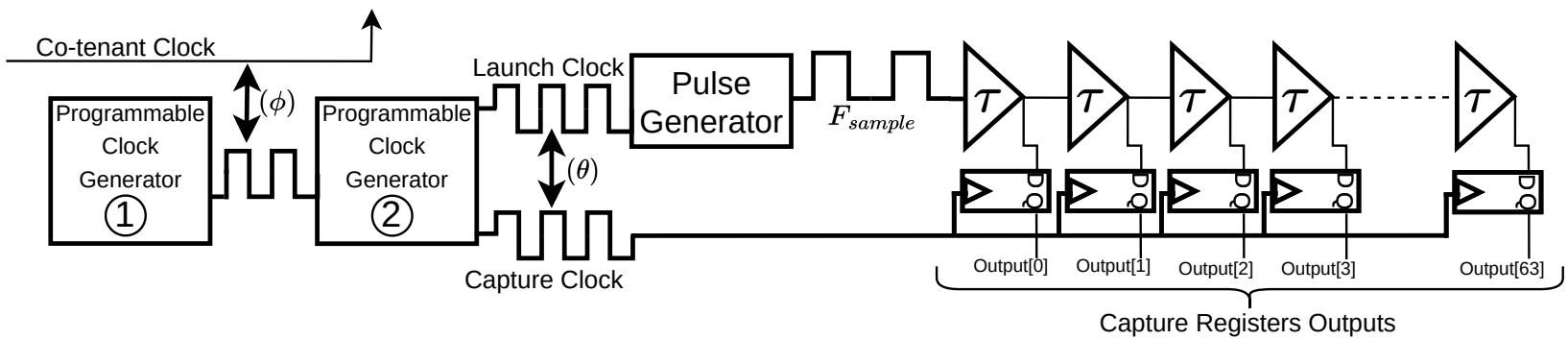


Solution: Dynamic Co-Tenant Phase Alignment

Our θ tunable Time-to-Digital Converter:



Our θ and ϕ tunable Time-to-Digital Converter:



Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.

Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.

Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.

Solution: Dynamic Co-Tenant Phase Alignment

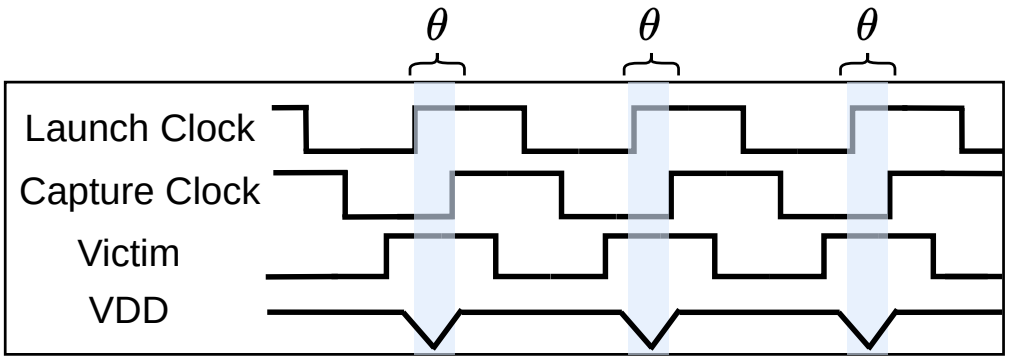
- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.

Expect to see a single optimal ϕ

Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.

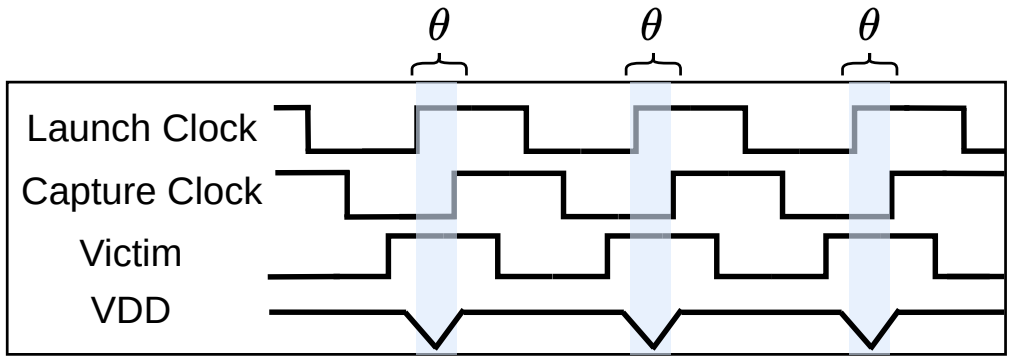
Expect to see a single optimal ϕ



Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.

Expect to see a single optimal ϕ

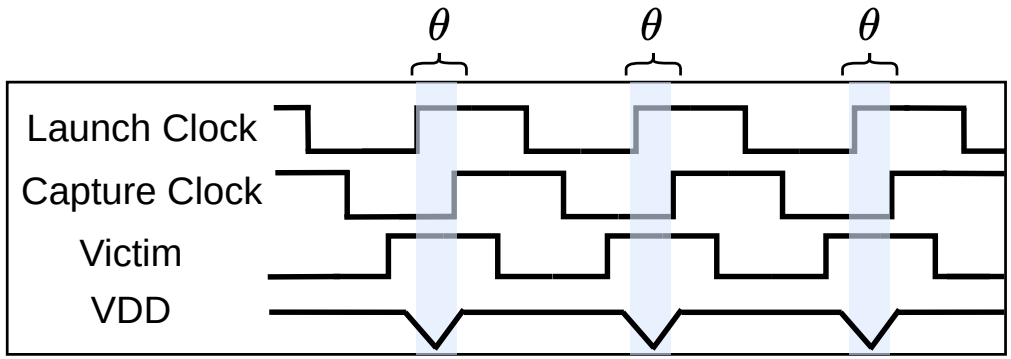


Many "optimal" ϕ

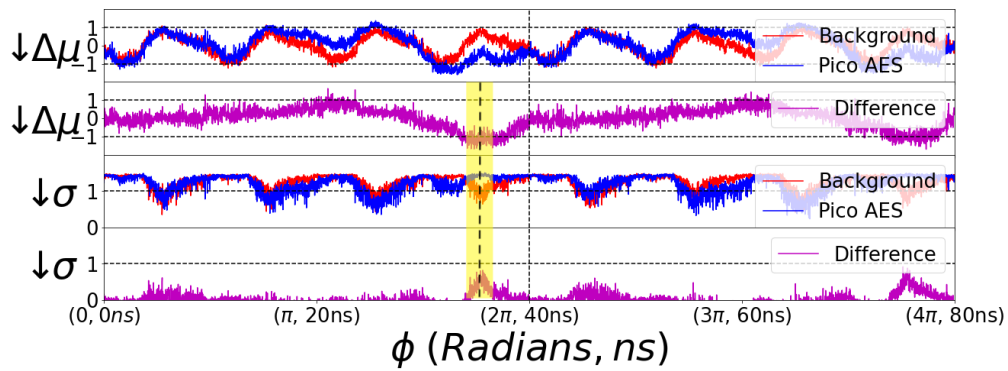
Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.

Expect to see a single optimal ϕ



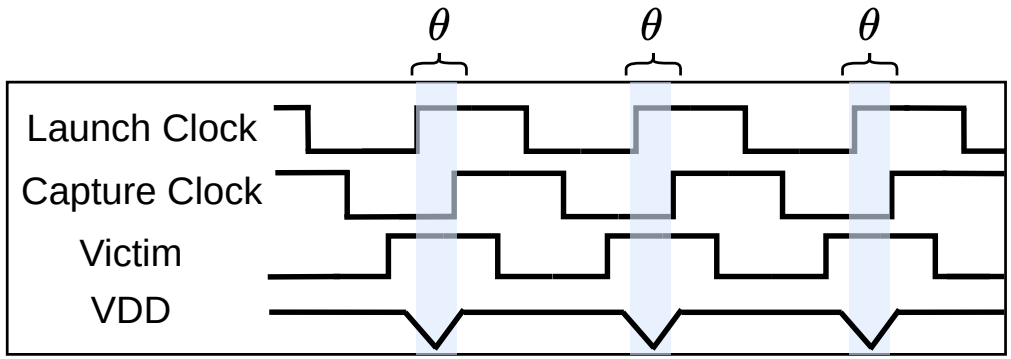
Many "optimal" ϕ



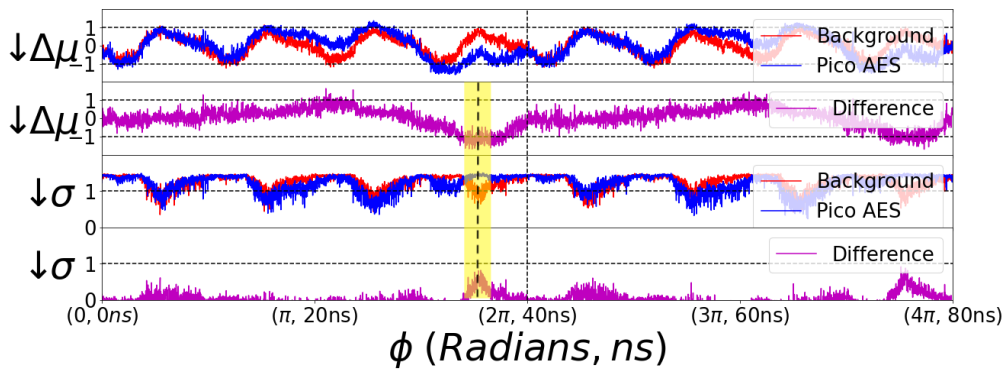
Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.
- Background sweep varies by ϕ

Expect to see a single optimal ϕ



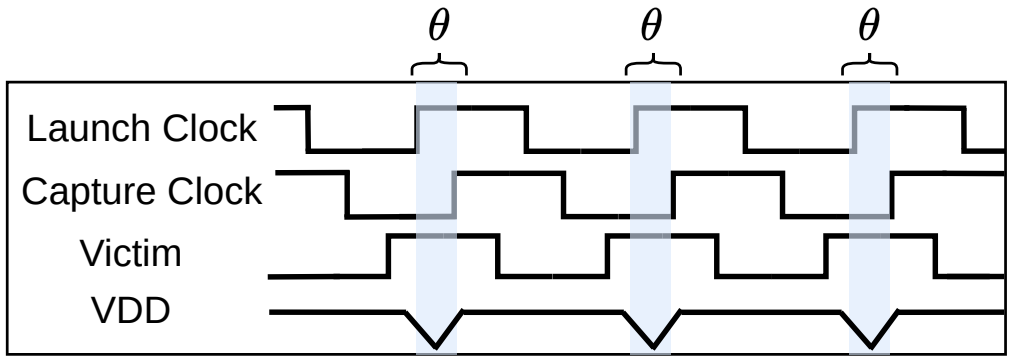
Many "optimal" ϕ



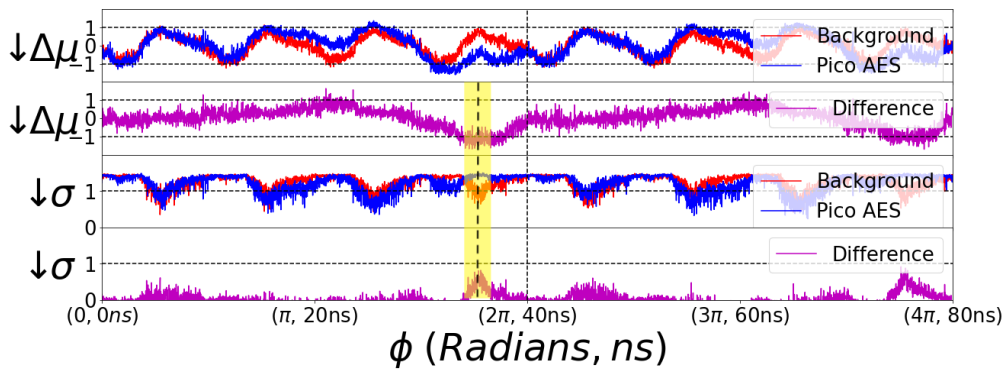
Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.
- Background sweep varies by ϕ
- Optimal ϕ blends into sensor background noise

Expect to see a single optimal ϕ



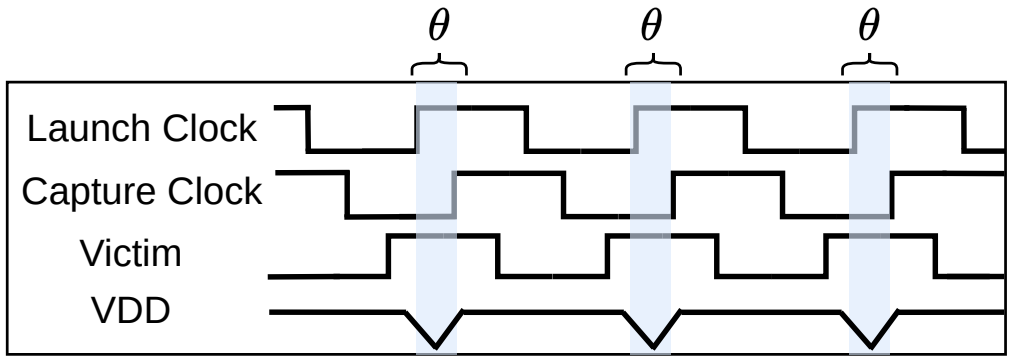
Many "optimal" ϕ



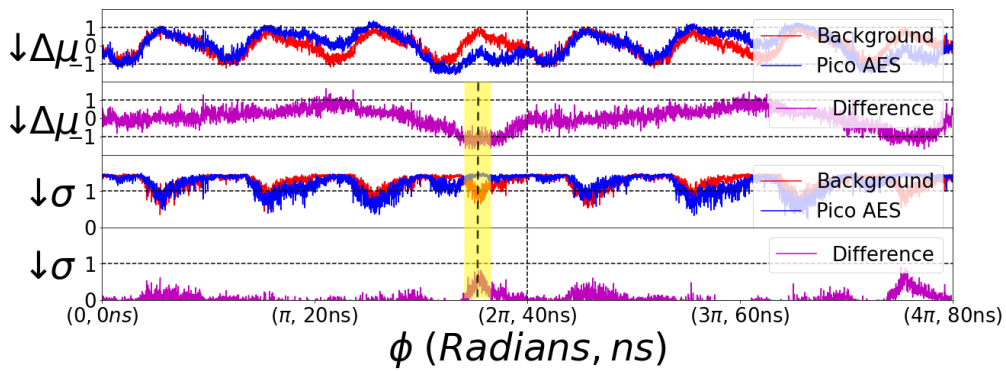
Solution: Dynamic Co-Tenant Phase Alignment

- Instantiate sensor alongside soft-processor AES. Both 25MHz.
- Rotate ϕ by two periods with processor disabled and measure.
- Rotate ϕ by two periods with processor enabled and measure.
- Background sweep varies by ϕ
- Optimal ϕ blends into sensor background noise

Expect to see a single optimal ϕ



Many "optimal" ϕ



Taking the difference from the background exposes optimal ϕ to capture power fluctuations induced by victim.

Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards

Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ

Classification Attack

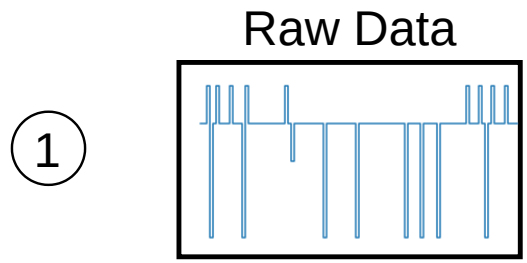
- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes

Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes
- Some number of boards are reserved for training, some for testing, representing the hypothetical divide between cloud and local FPGAs.

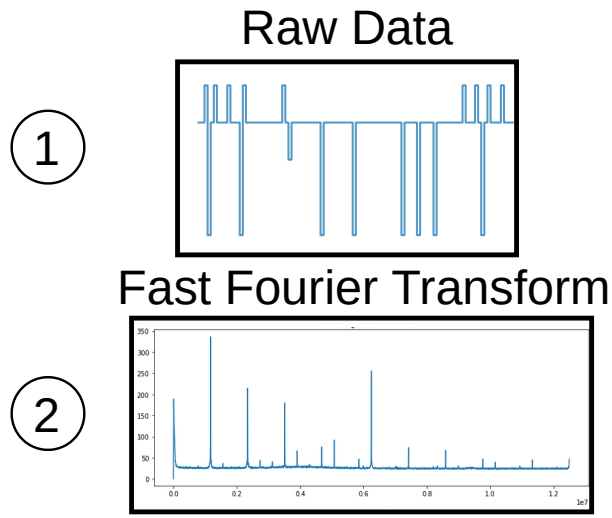
Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes
- Some number of boards are reserved for training, some for testing, representing the hypothetical divide between cloud and local FPGAs.
- The data is then processed....



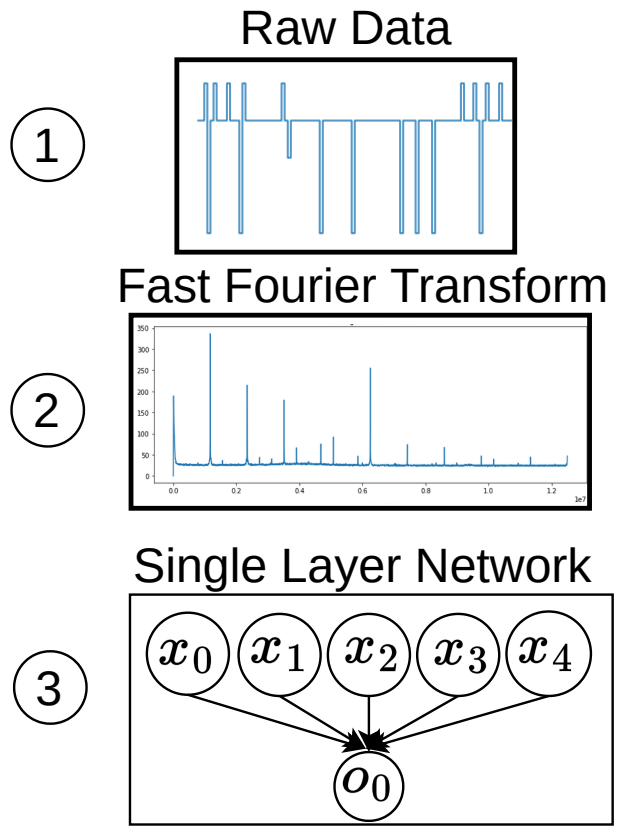
Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes
- Some number of boards are reserved for training, some for testing, representing the hypothetical divide between cloud and local FPGAs.
- The data is then processed....



Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes
- Some number of boards are reserved for training, some for testing, representing the hypothetical divide between cloud and local FPGAs.
- The data is then processed....

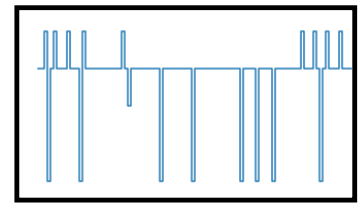


Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes
- Some number of boards are reserved for training, some for testing, representing the hypothetical divide between cloud and local FPGAs.
- The data is then processed....

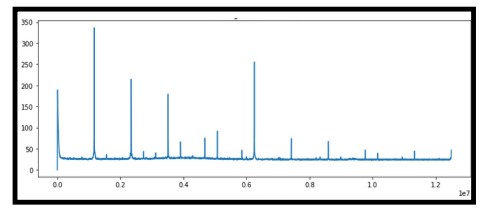
1

Raw Data



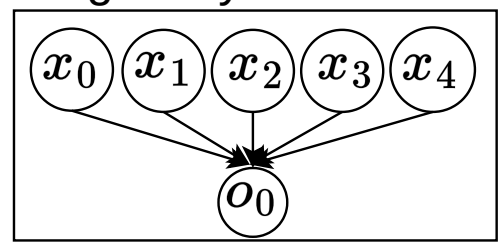
2

Fast Fourier Transform

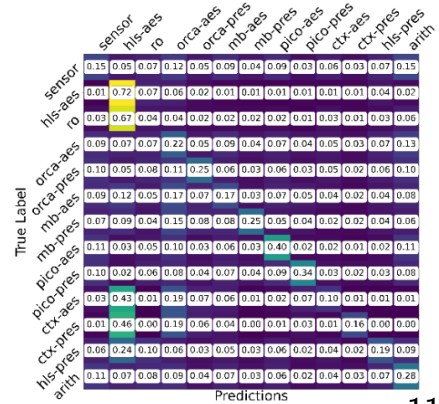


3

Single Layer Network



4

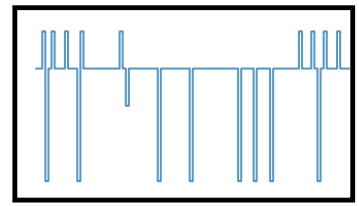


Classification Attack

- Sensor co-located with each of 13 victim computations on multiple boards
- Sensor is tuned to some value of θ and ϕ to minimize (maximize) θ and ϕ
- Sensor is sampled as victim executes
- Some number of boards are reserved for training, some for testing, representing the hypothetical divide between cloud and local FPGAs.
- The data is then processed....
- Classification accuracy across 13 applications reflects the sensors ability to capture sensitive information across the channel.

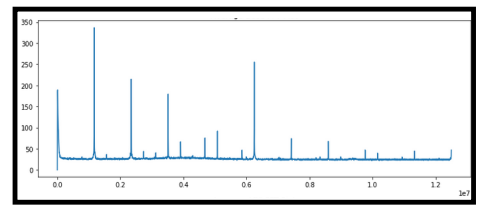
1

Raw Data



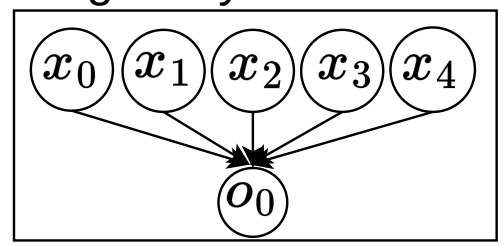
2

Fast Fourier Transform

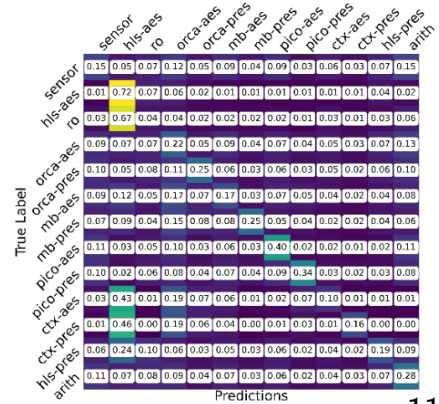


3

Single Layer Network



4

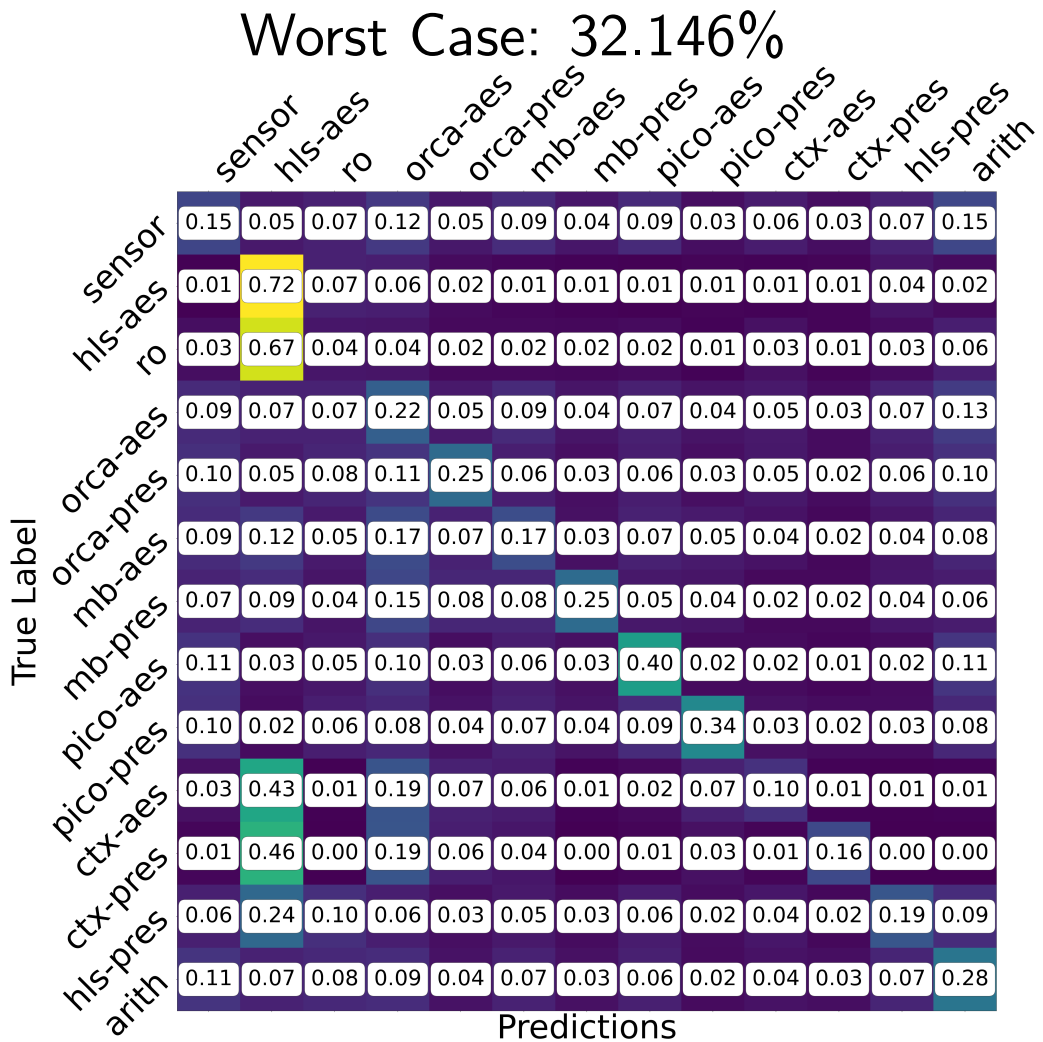


Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).

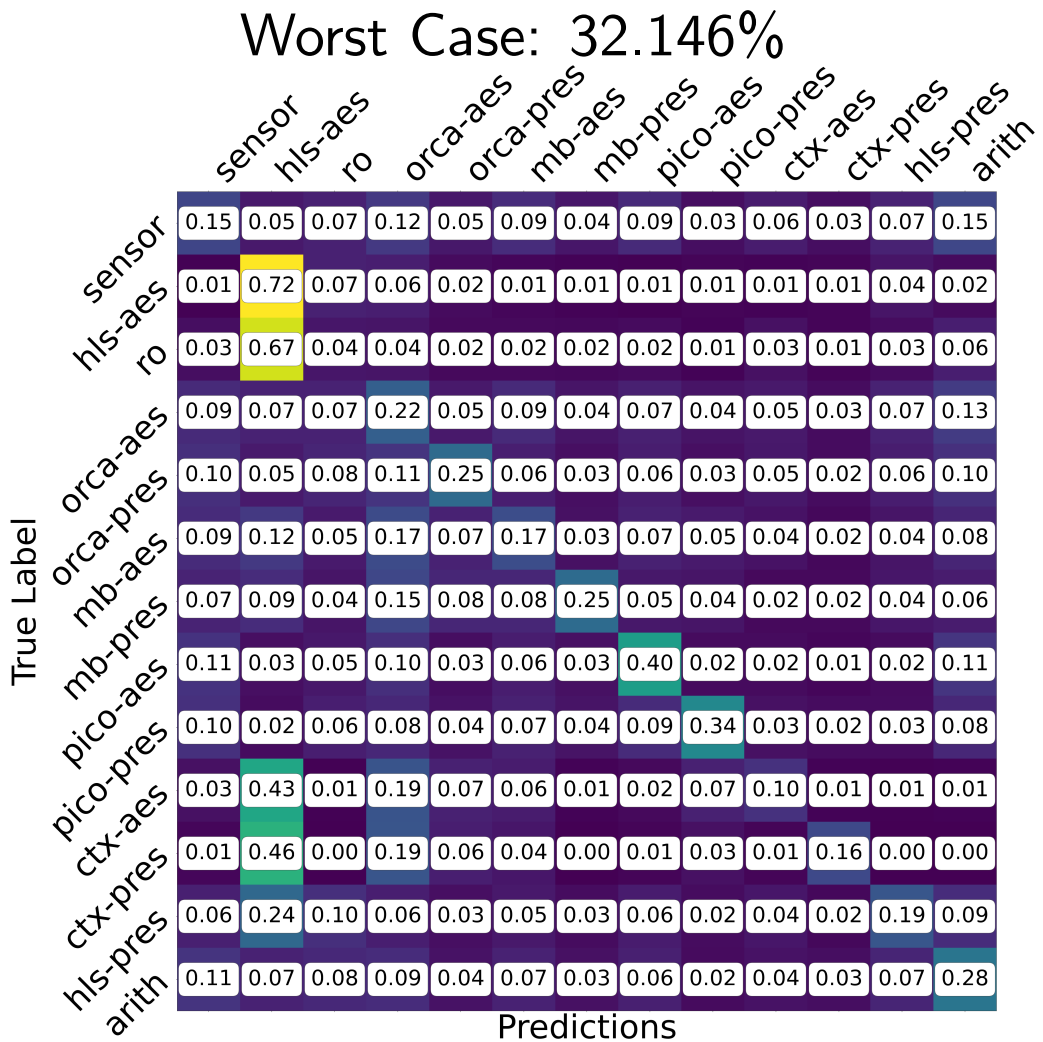
Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).



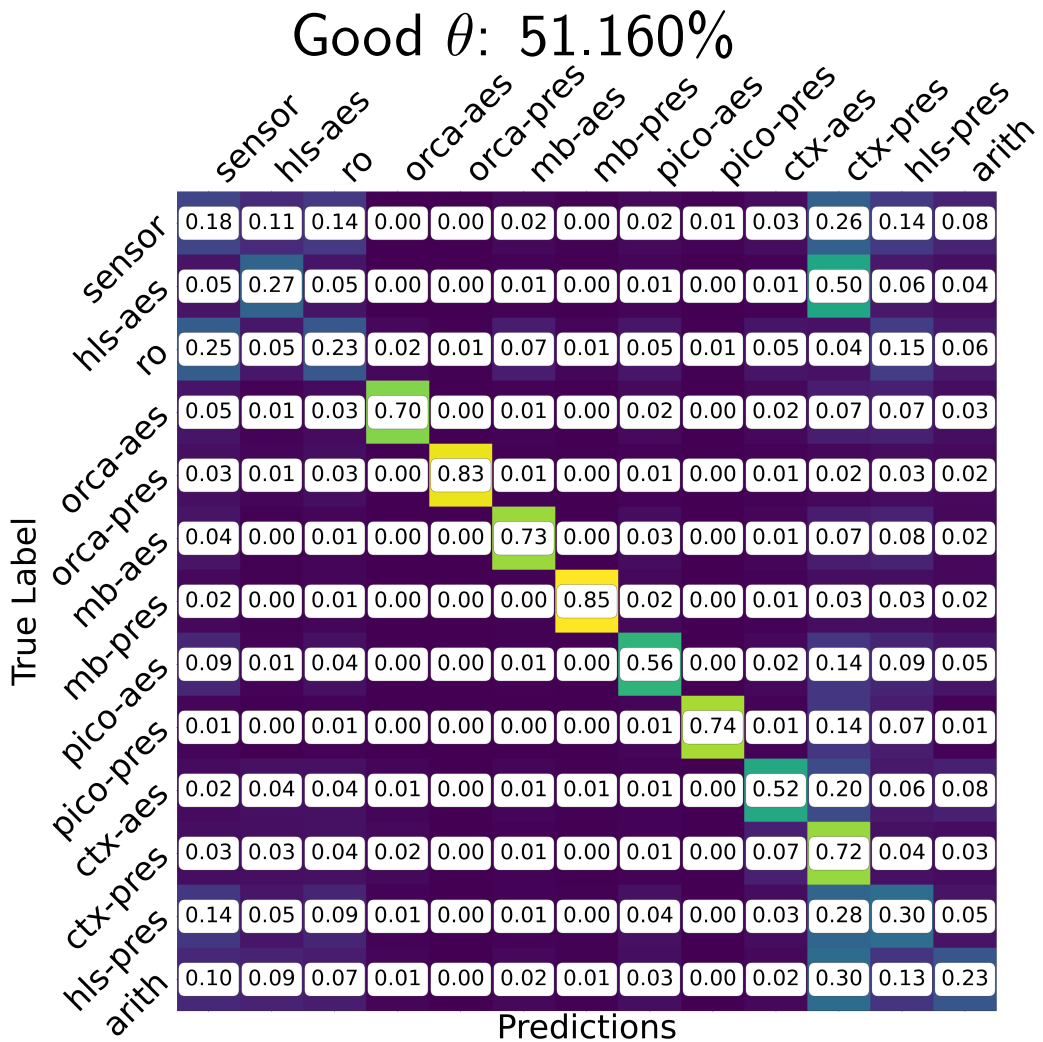
Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).
- Good θ: max standard deviation of θ, and min ϕ.



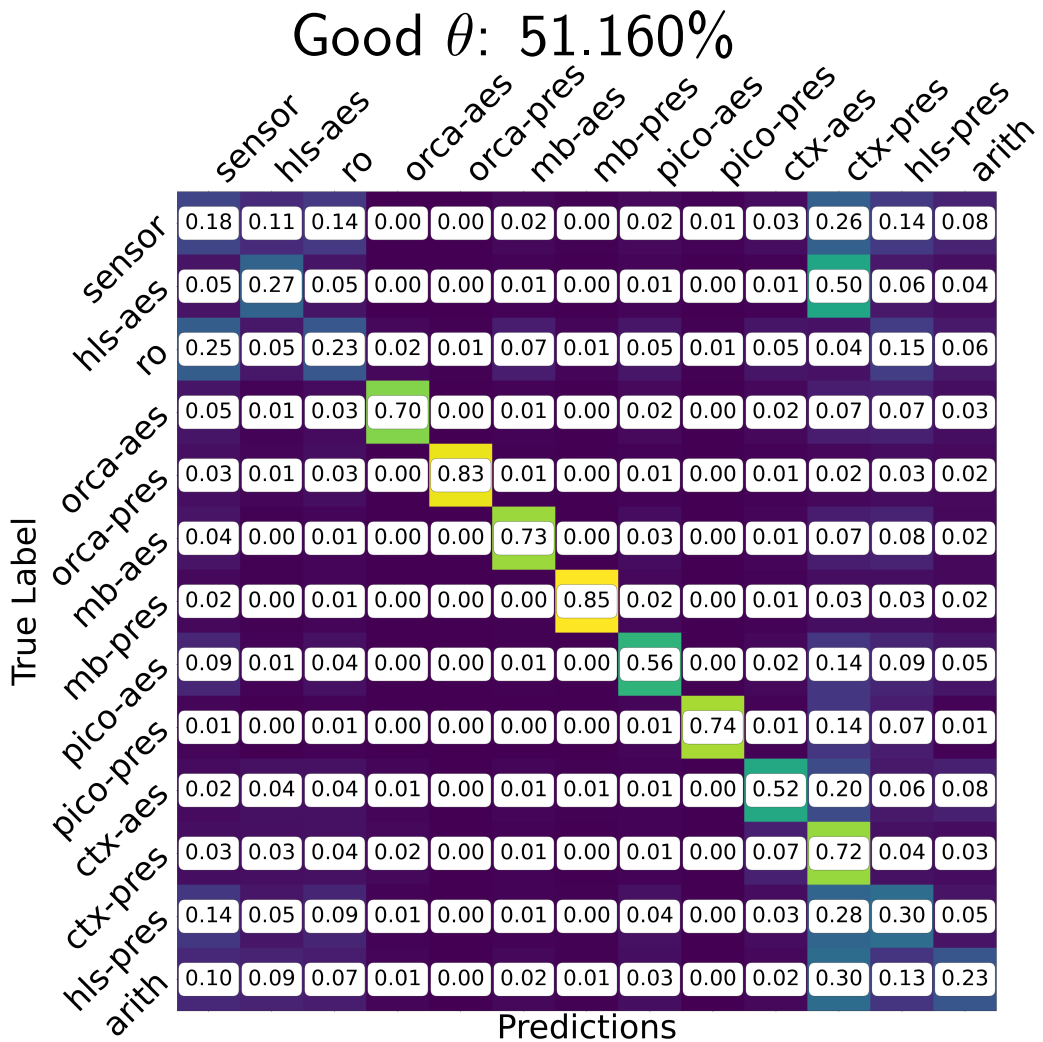
Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).
- Good θ: max standard deviation of θ, and min ϕ.



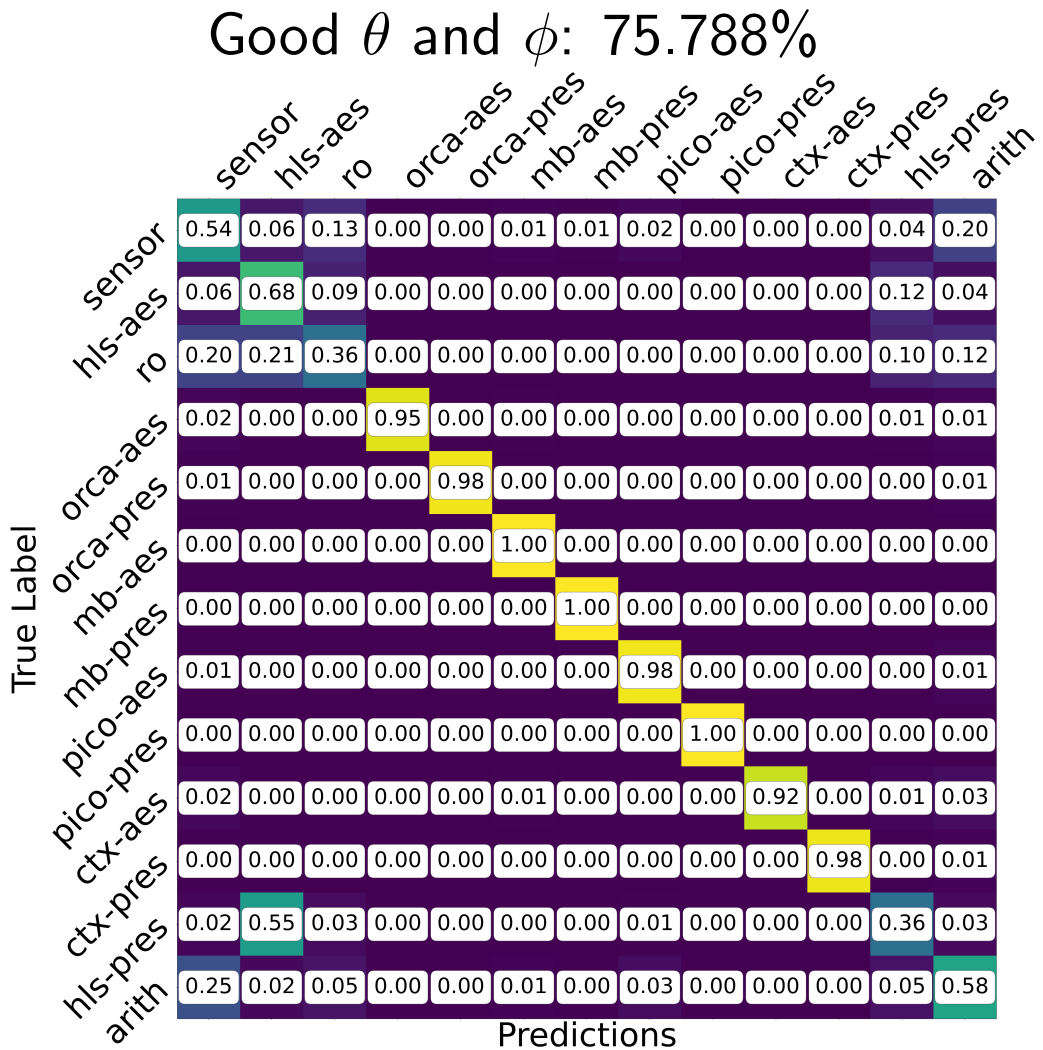
Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).
- Good θ: max standard deviation of θ, and min ϕ.
- Good θ and ϕ: max standard deviation of θ, and max ϕ.



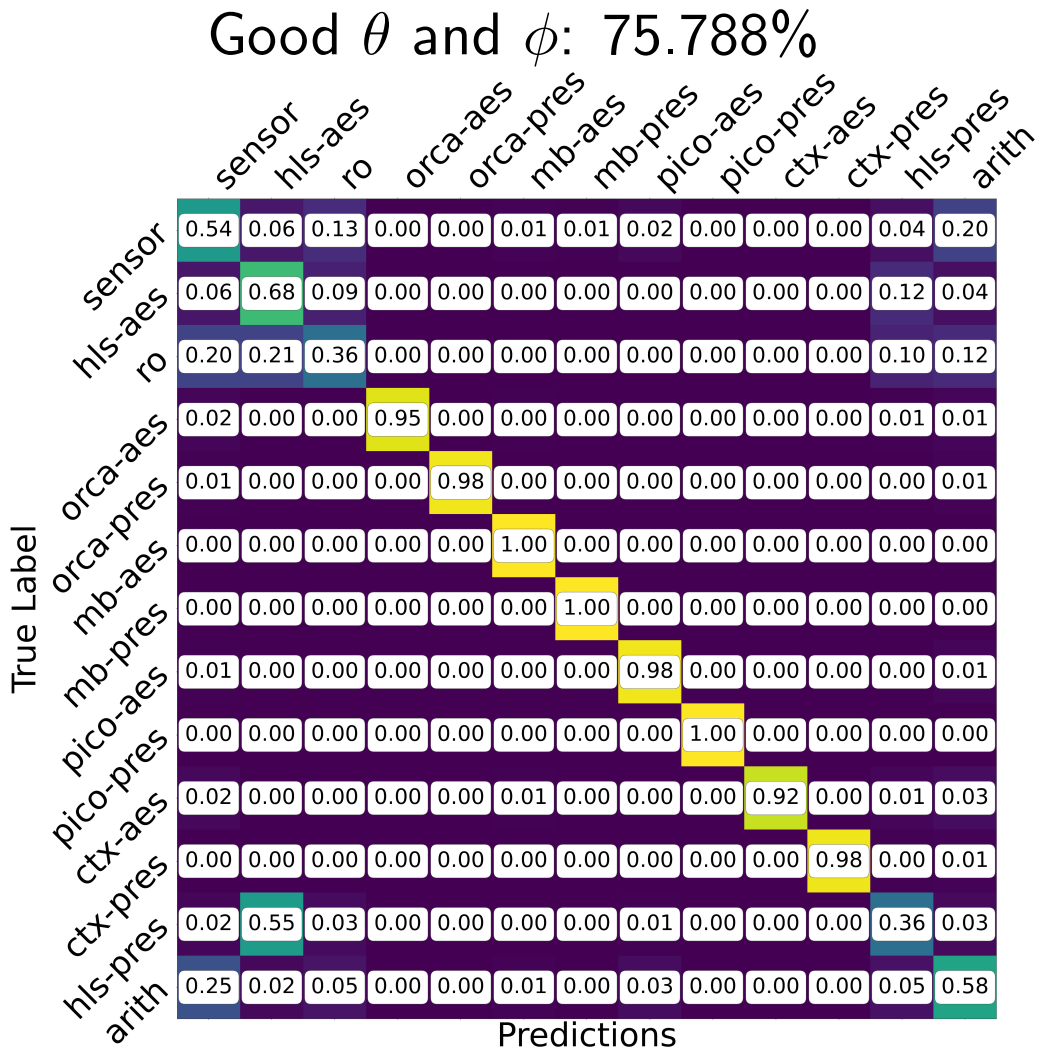
Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).
- Good θ: max standard deviation of θ, and min ϕ.
- Good θ and ϕ: max standard deviation of θ, and max ϕ.



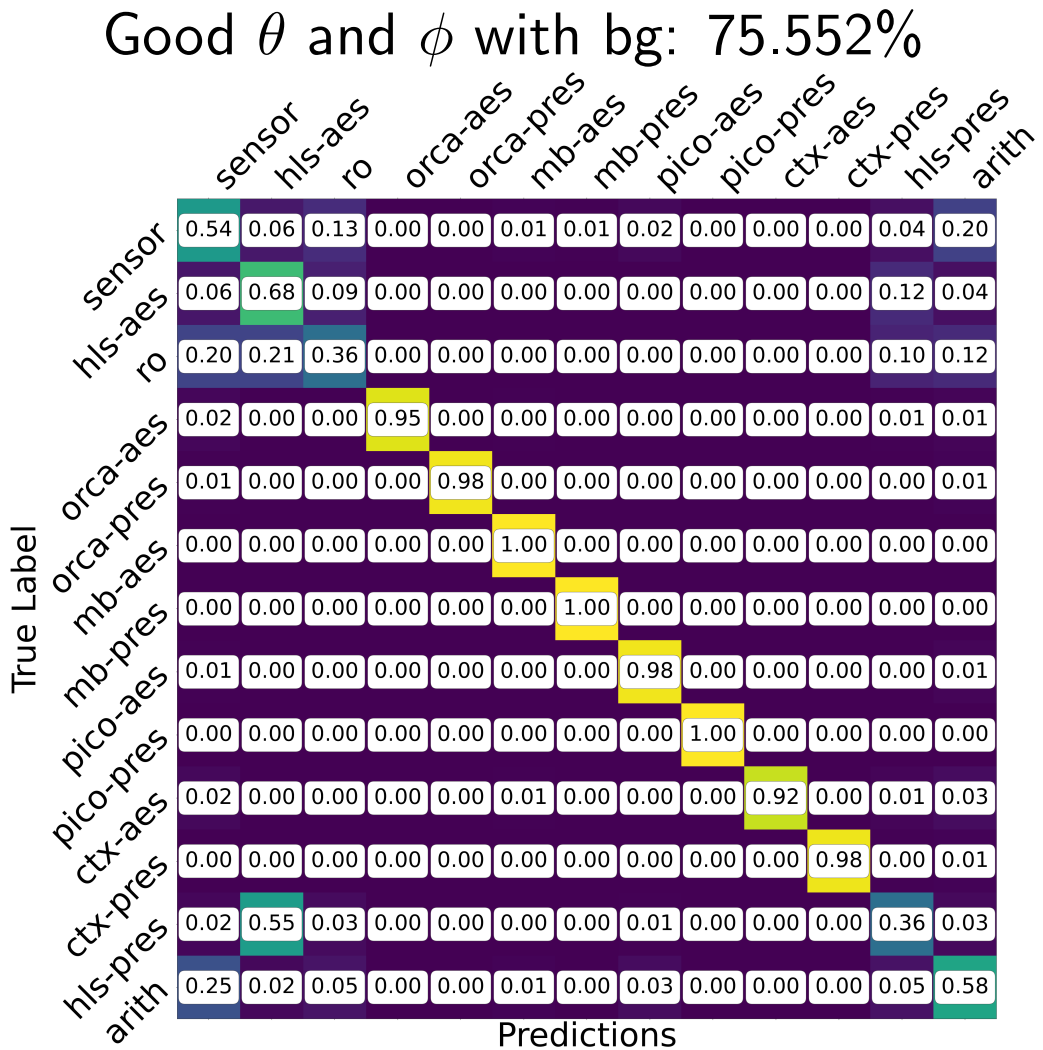
Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).
- Good θ: max standard deviation of θ, and min ϕ.
- Good θ and ϕ: max standard deviation of θ, and max ϕ.
- Good θ and ϕ: with ϕ background subtraction.



Classification Accuracy by Tuning Configuration

- Worst case: min standard deviation of θ (likely a plateau), and ϕ (likely phase misaligned).
- Good θ: max standard deviation of θ, and min ϕ.
- Good θ and ϕ: max standard deviation of θ, and max ϕ.
- Good θ and ϕ: with ϕ background subtraction.



So background subtraction is useless?

Check the cross validation performance

So background subtraction is useless?

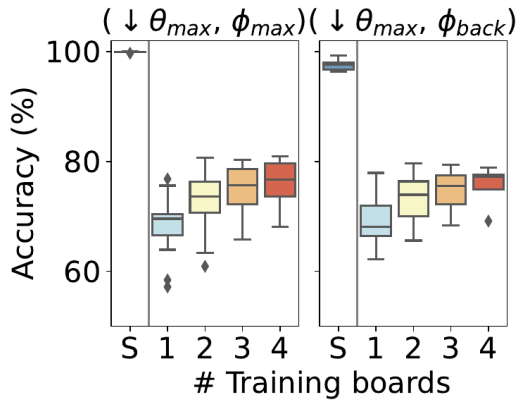
Check the cross validation performance

- Test every combination
  and number of training and
  testing board

So background subtraction is useless?

Check the cross validation performance

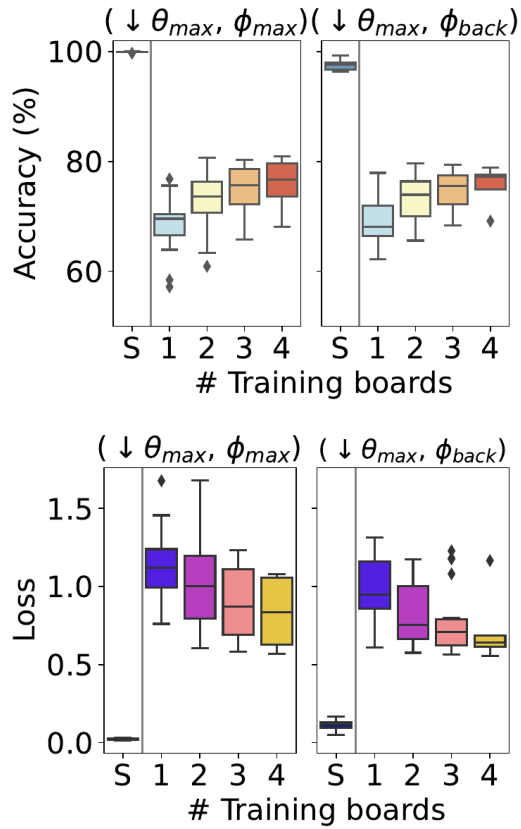
- Test every combination and number of training and testing board
- Examine accuracy vs # training boards. Reduces IQR by 2.3X.



So background subtraction is useless?

Check the cross validation performance

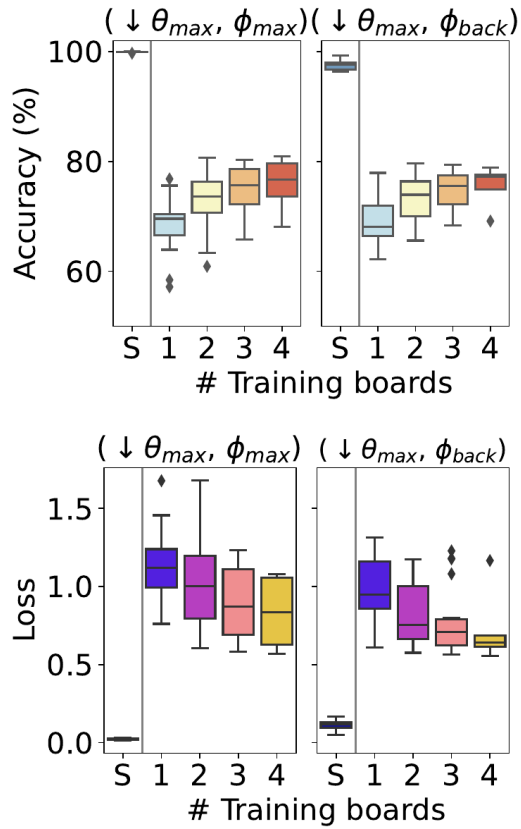
- Test every combination and number of training and testing board
- Examine accuracy vs # training boards. Reduces IQR by 2.3X.
- Examine loss vs # training boards. Reduces IQR by 5.8X.



So background subtraction is useless?

Check the cross validation performance

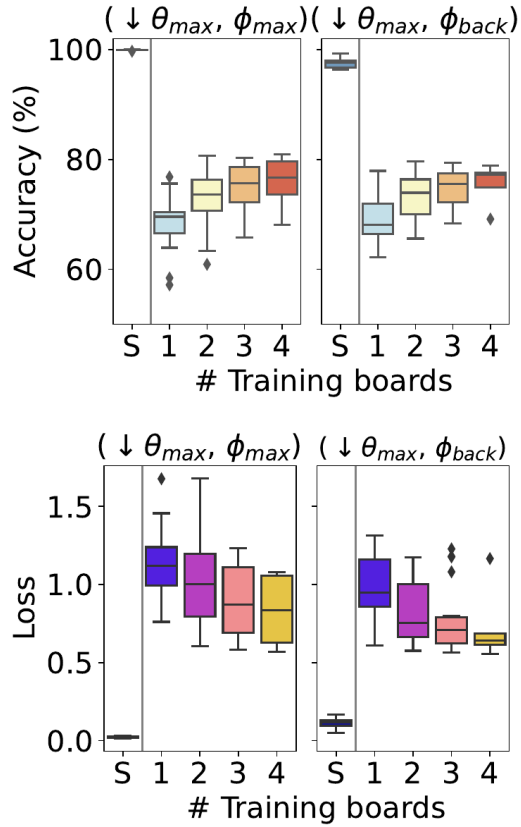
- Test every combination and number of training and testing board
- Examine accuracy vs # training boards. Reduces IQR by 2.3X.
- Examine loss vs # training boards. Reduces IQR by 5.8X.
- *Data generalizes better*



So background subtraction is useless?

Check the cross validation performance

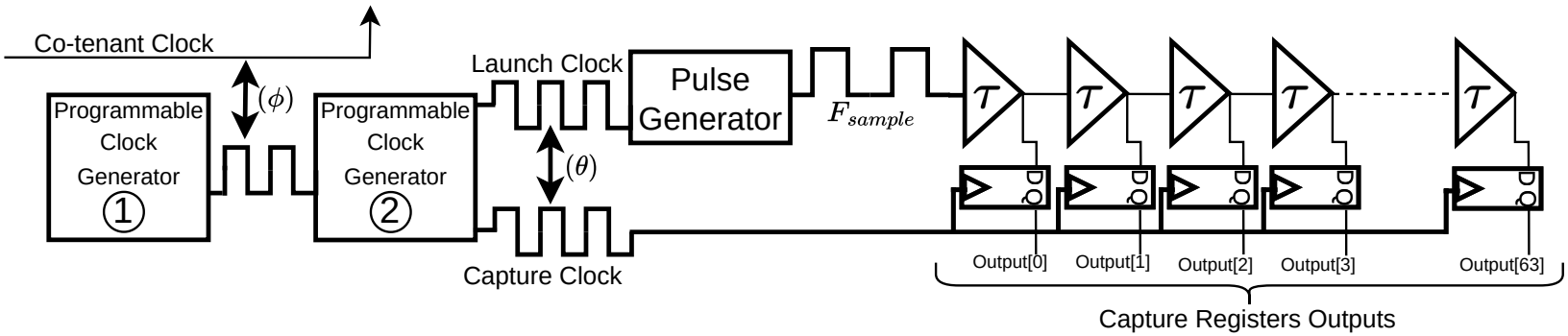
- Test every combination and number of training and testing board
- Examine accuracy vs # training boards. Reduces IQR by 2.3X.
- Examine loss vs # training boards. Reduces IQR by 5.8X.
- *Data generalizes better*

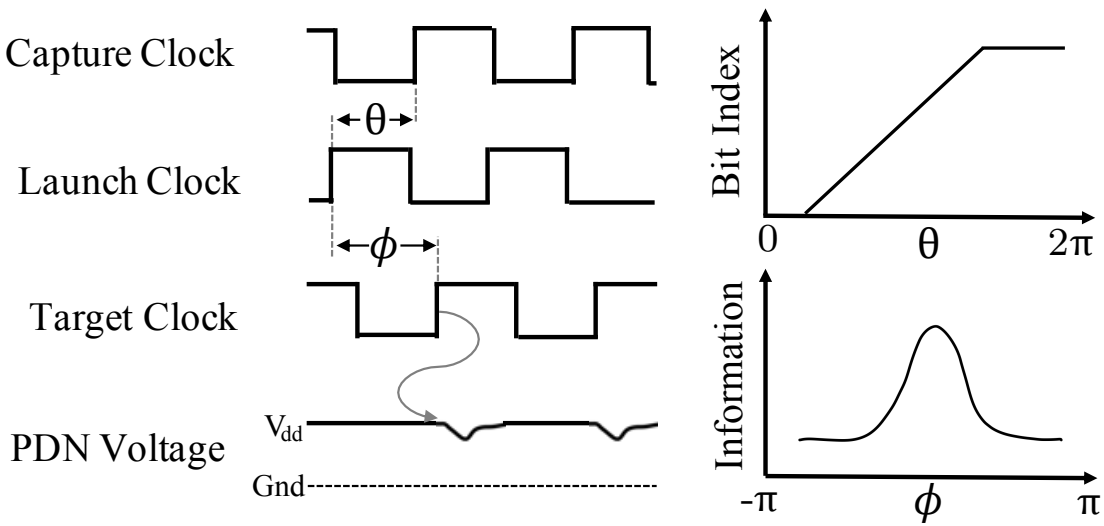


Background Subtraction is Out-of-Distribution Generalization!

Conclusion/Questions

Don't let θ and ϕ assume a random value!





Thanks!