# Capture Interfaces in the ChipWhisperer Family: OPTIMIST Hour

Jean-Pierre Thibault, NewAE Technology

January 23, 2025

# Capture Interfaces: Host-Side Python API

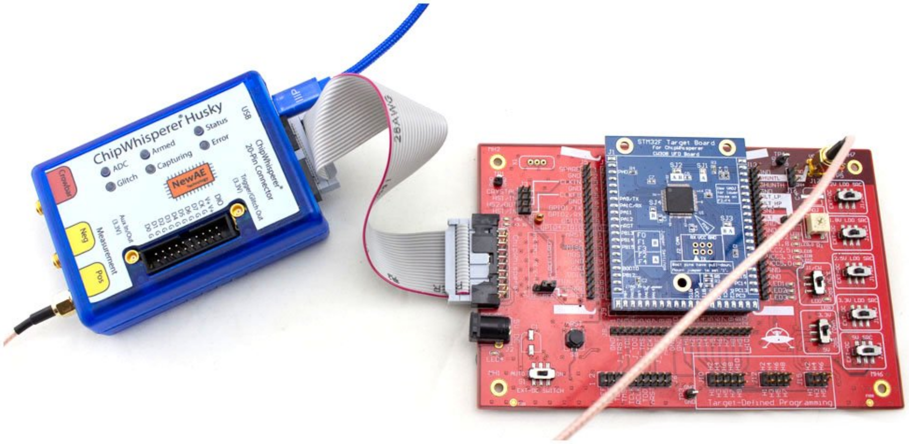
- Standard across all ChipWhisperer capture devices (Nano, Lite, Pro, Husky):

```
scope.adc.presamples = 5000
scope.adc.samples = 5000

scope.clock.clkgen_src = "system"
scope.clock.clkgen_freq = 20e6

trace = cw.capture_trace(...)
```

# Capture Interfaces: Target-Side Physical Interface

- Standard across all ChipWhisperer capture devices and targets:
  - UART communication lines
  - trigger line

- additional parallel interface (via USB) on CW305/310/340 FPGA target boards



# Relevant Issues Identified in Optimist Kick-Off Slides

1. Scalability
2. Interoperability
3. Reproducibility
4. Learning Curve
5. Need for Better Hardware

# Scalability

- Getting more traces, faster?
  - limited by USB 2.0 (cost)
  - segmented capture: up to around 1000 captures/second
  - trace format and storage also plays into this
- Cost/performance trade-off!

# Interoperability

- What can we do to facilitate interoperability?

# Reproducibility

- Primary objective since the beginning of ChipWhisperer
- What are we missing?

# Steep Learning Curve

- current learning material:
  - notebooks,
  - rtfm.newae.com (HW documentation),
  - chipwhisperer.readthedocs.io (SW documentation),
  - forum,
  - discord

- working on a new "tips and tricks" resource for:
  - common issues
  - advanced usage

NewAE Technology

# Better Hardware

- CW-Lite FPGA is full ⇒ can't add features ⇒ CW-Husky

- New Husky features:

  1. higher sampling rate
  2. faster streaming
  3. more flexible clocking
  4. more powerful SAD
  5. sequenced triggering
  6. segmented capture with lots of options
  7. built-in logic analyzer
  8. expandability/customization via USERIO header

- Trade-off between cost/capability

- What other features do you want to see?