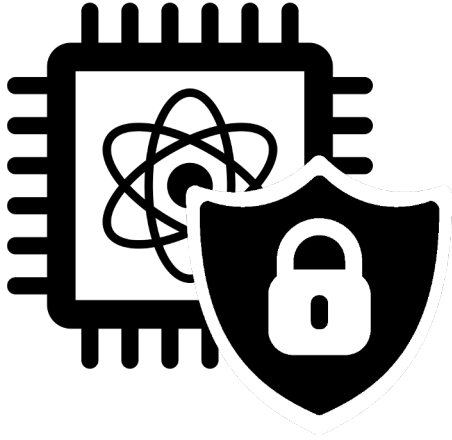




Tutorial on Security of Quantum Computing Systems

QCE Tutorial

Sept. 17, 2026

Prof. Jakub Szefer (jakub.szefer@northwestern.edu)

Electrical and Computer Engineering

Northwestern University



Computer Architecture
and Security Lab (CASLAB)



Northwestern
University

Tutorial Outline

- 1:00pm – 1:30pm: Introduction and Motivation
- 1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers
- 2:30pm – 3:00pm: Coffee Break
- 3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers
- 3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers
 - Design of QC-TEEs for Quantum Computers
- **Extra: Hardware Security for QML**
- **4:00pm: End**



Need for Security of QML Algorithms

Large numbers of QML algorithms are being developed and researched across both FTQC and NISQ domains with use-cases in industries with sensitive data or code:

Era	Primitives	Applications
NISQ	Variational Quantum Algorithms (VQAs)	VQE, QAOA, VQLS, quantum kernels, analog simulation**
Early FT	Iterative Quantum Phase Estimation (QPE)	GSEE, QAE, QMCI
FTQC	Quantum Fourier Transform (QFT), Quantum Phase Estimation (QPE), Quantum Singular Value Transformation (QSVT), Quantum Random Access Memory (QRAM)*	Shor's, Quantum Chemistry simulation, Quantum Materials simulation, HHL*, qPCA*

* QRAM is not a primitive, but needed for speedups on many QML algorithms

** quantum simulation is analog, not gate based

Security critical aspects of QML algorithm: **input data, parameters, algorithm itself, and outputs.**



QML Secrets: Input Data, Parameters, Algorithm, Outputs

- **Input data** – can be sensitive data, medical, insurance, stock market, environmental data, etc.
 - **Classical, hardcoded in the circuit**, dominant method today
 - **Quantum, read from QRAM**, made available via pre-initialized QRAM
 - **Quantum, read from sensors**, direct quantum input avoiding dealing with any classical data
 - **Quantum, read from quantum network**, inputs from other quantum devices
- **Parameters** – the model weights, encode model's intellectual property and training data
 - **Classical, gate configuration** analog control for an annealer or gates in a gate-based device
- **Algorithm** – the algorithm itself, contains intellectual property of the developer
 - **Classical, gate configuration**, classical control signals that drive qubits
- **Outputs** – the result of computation
 - **Classical, measurement outputs** read out
 - **Quantum, output to quantum network**



Aside: Limits of Data Encoding from Classical Sources

Inputs to QML can come from classical and quantum source, but encoding classical data into quantum states is expensive:

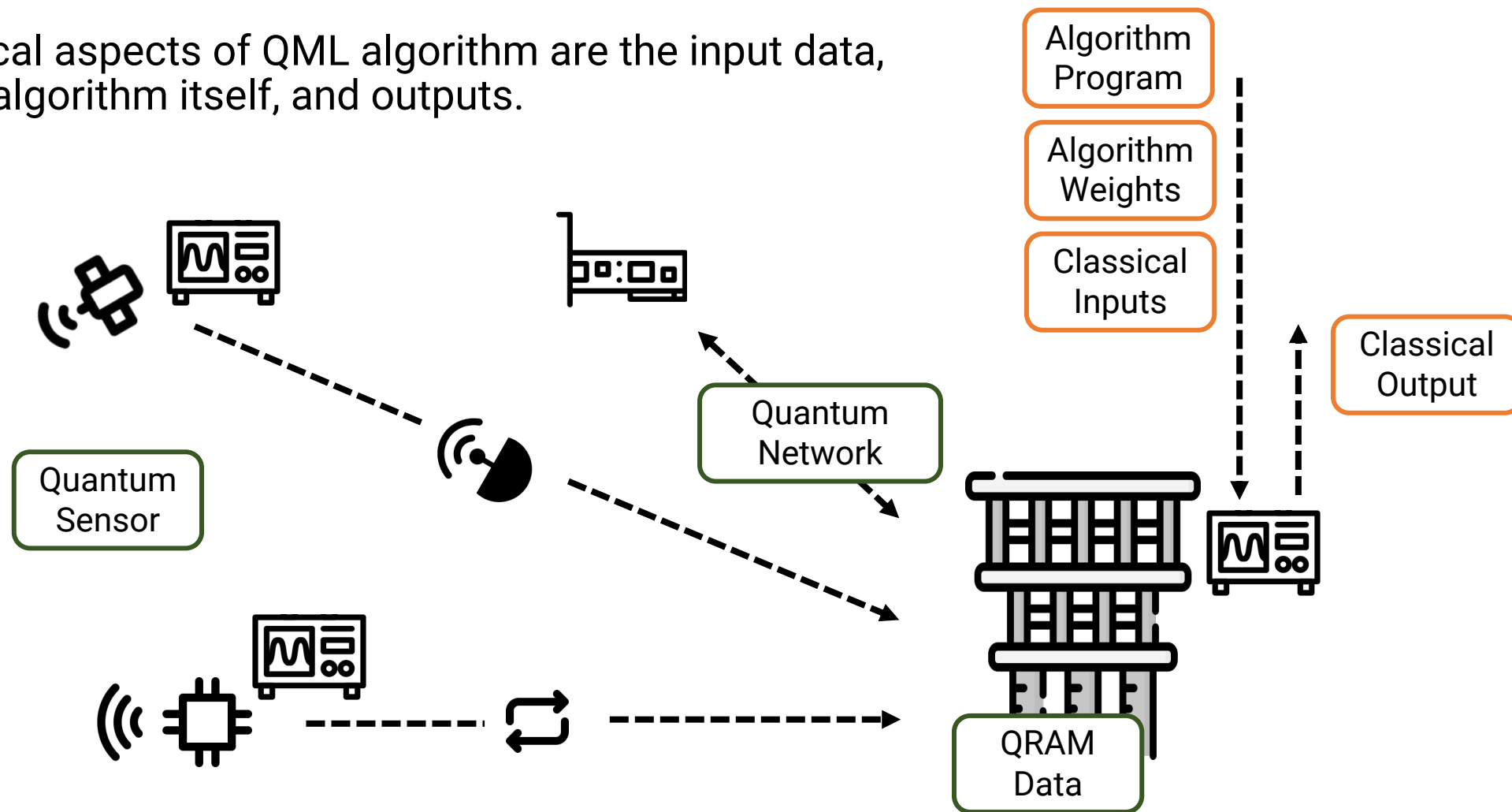
- **Amplitude encoding:** you can encode 2^n values using n qubits, but the time it takes is $O(2^n)$ gates, $O(2^n)$ depth, $O(n)$ qubits
- **Other encodings:** trade-off gates, depth, or qubits, but no encoding that gives exponential compression and cheap preparation for arbitrary data

Alternative is getting quantum data directly from quantum sensing. **Quantum sensing becomes a critical piece of QML** when considering arbitrary data. Although it has its own challenges: can't copy quantum data, currently decoherence is problem, etc.



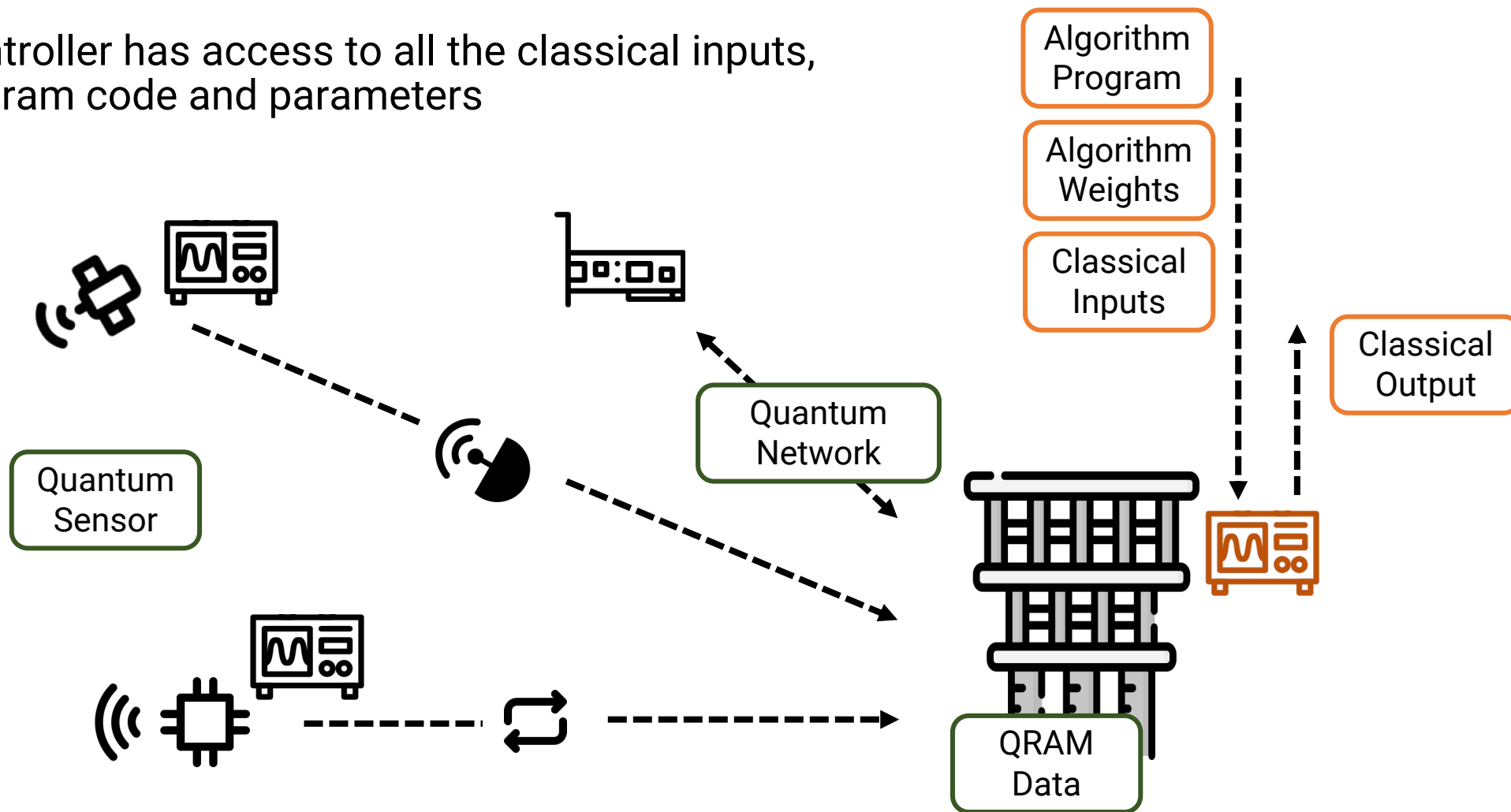
Where the Secrets Live in a Quantum System

Security critical aspects of QML algorithm are the input data, parameters, algorithm itself, and outputs.



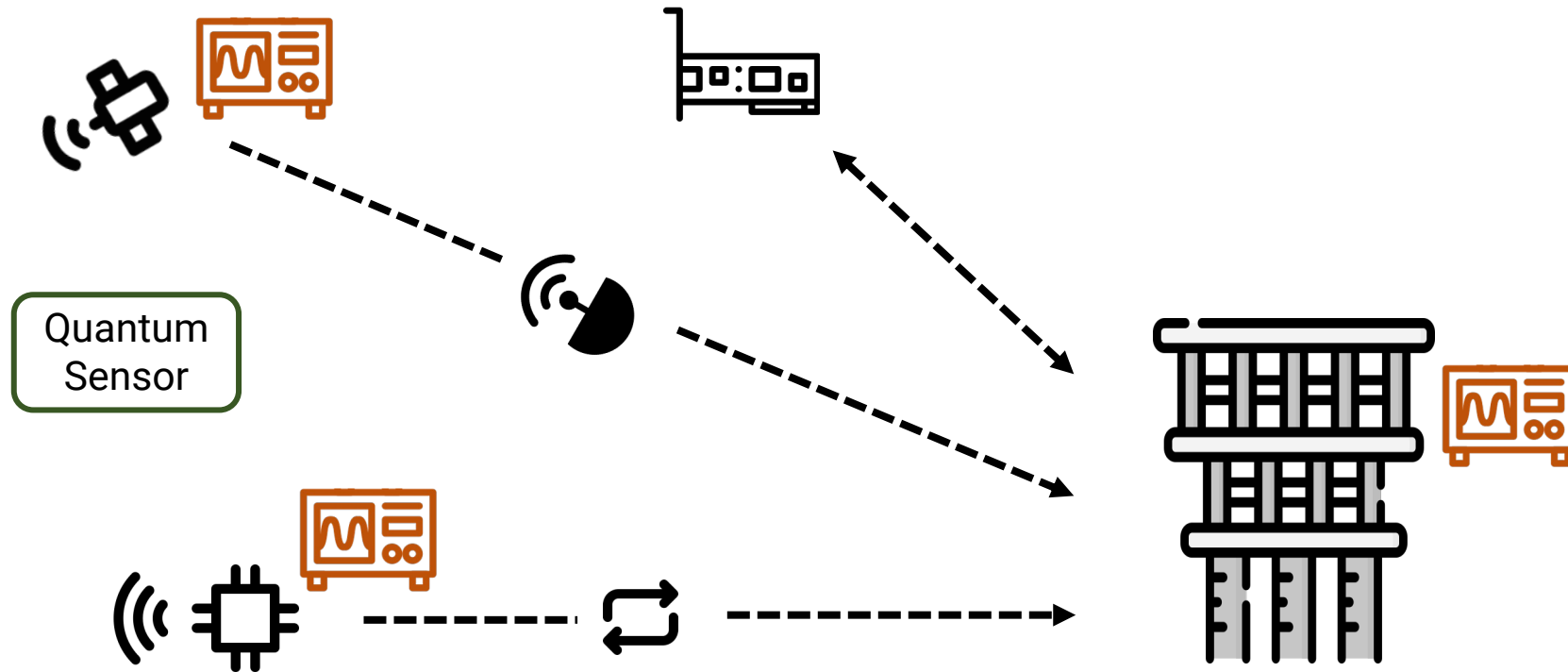
Secrets Live in Classical Controllers

Classical controller has access to all the classical inputs, outputs, program code and parameters



Secrets also Live in Quantum Sensor Controllers

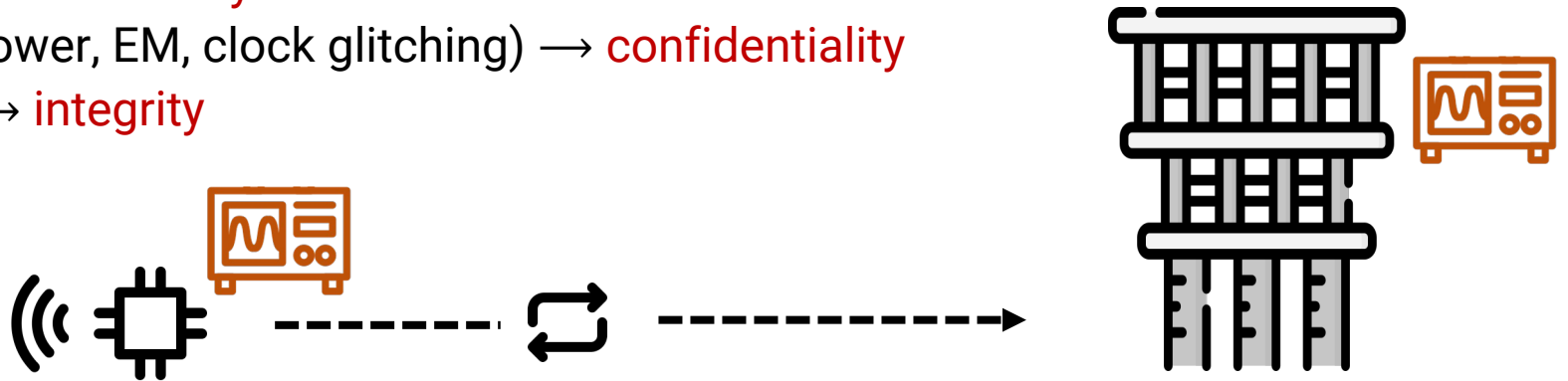
Quantum data coming from quantum sensors has no ground truth, it is whatever the sensor captures, but what sensor captures is indirectly controlled by the sensor controller



Hardware Security for QML Should Focus on Controllers

Both at the quantum computer side and quantum sensor side, it is the controllers that hold the classical secrets, but:

- **Controllers can be vulnerable to hardware attacks**
- **Controllers lack attestation features**
 - Lack of attestation of the instructions processed (quantum computer controller) → **integrity**
 - Lack of attestation of the configuration (quantum sensor and quantum computer controller) → **integrity**
- **Controllers lack defenses**
 - Non-constant time code → **confidentiality**
 - No side channel protection (power, EM, clock glitching) → **confidentiality**
 - No fault injection protection → **integrity**



Hardware Attacks: Side Channel and Fault Injection Threats

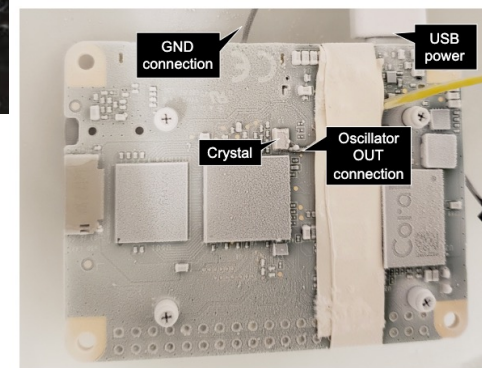
Hardware security attacks typically involve modifying the physical environment (fault injection) or physical measurement (side channels) + associated data processing

Fault Injection

Perturb the system by injecting signals or suppressing signals to alter its operation



- EM fault injection with a near-field probe
- Voltage glitching
- Clock glitching
- Laser attacks (after decapsulation)
- Thermal attacks

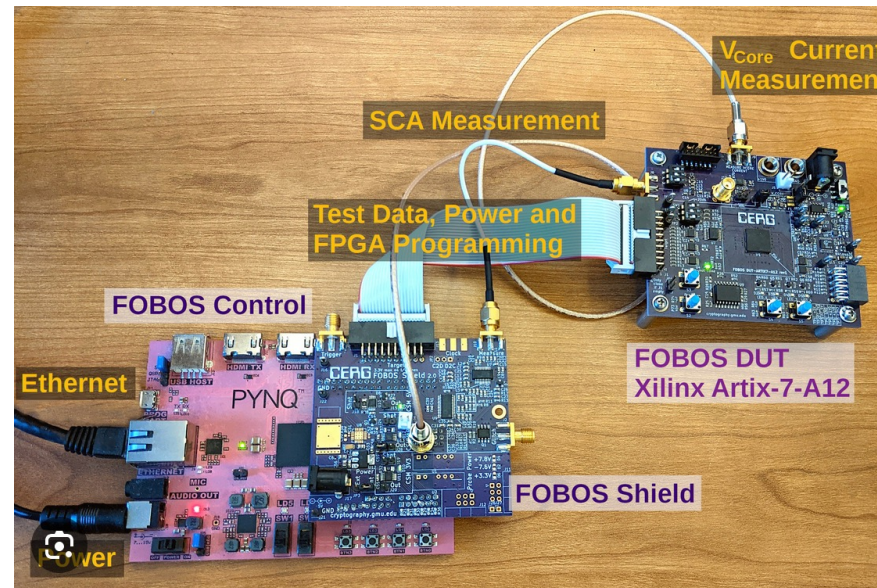


Hardware Attacks: Side Channel and Fault Injection Threats

Hardware security attacks typically involve modifying the physical environment (fault injection) or physical measurement (side channels) + associated data processing

Side Channels

Collect power, EM, or other signals as the system is operating to learn what it is doing



- Power
- EM
- Thermal
- Timing
- Optical
- Acoustic
- Etc.

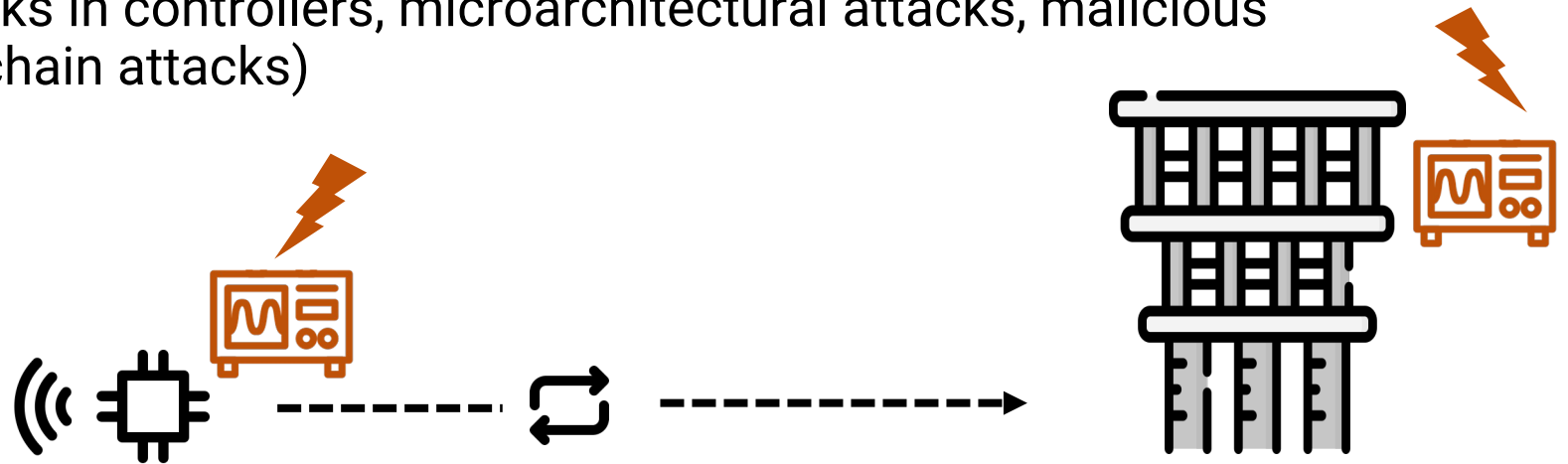


Proposed Threat Model

A threat model is a structured description of what is protected, who the adversary is (their capabilities, access, and goals), and which attacks are in- versus out-of-scope.

Different threat models are possible, but for hardware security of QML we can consider:

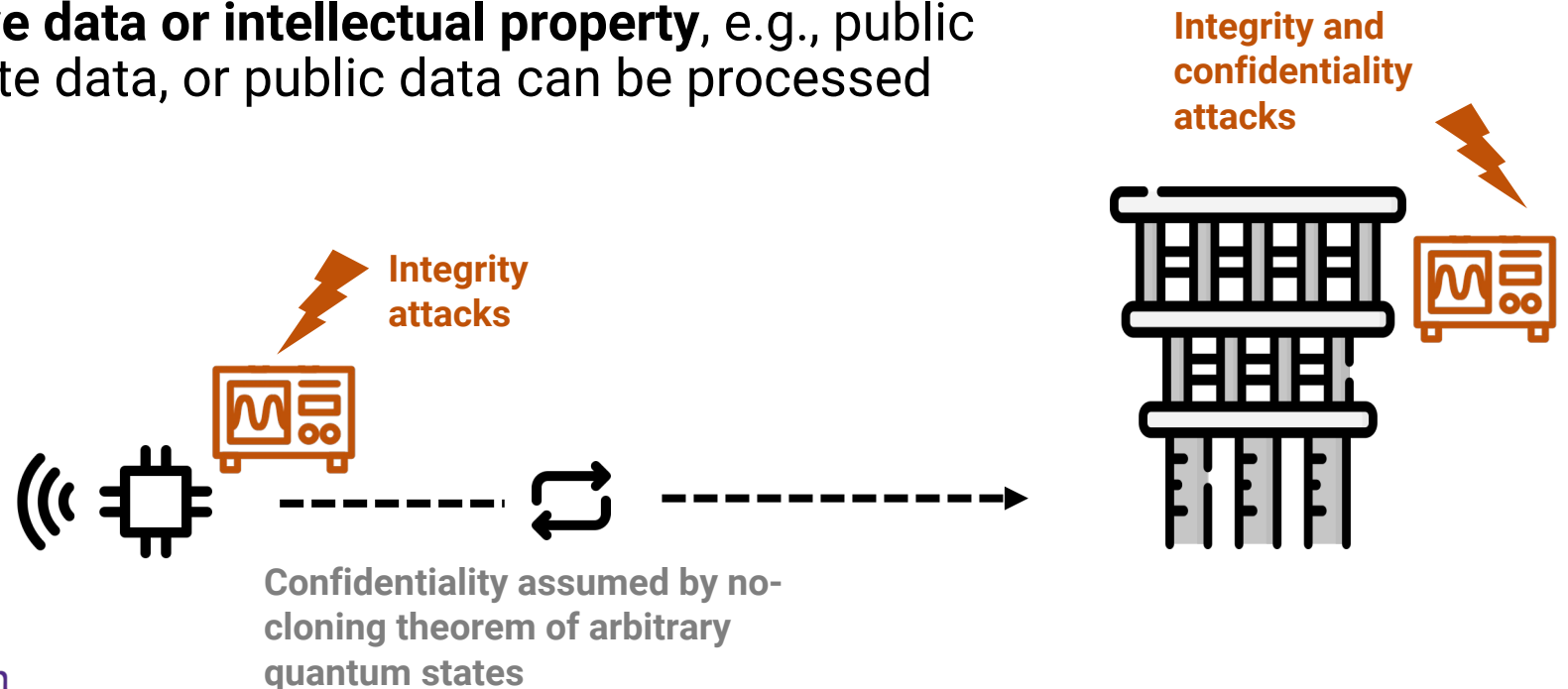
- Goal is to protect the secrets held in classical controllers
- Attackers have physical ability to monitor the operation of the controllers or perturb them
- Not considering software attacks in controllers, microarchitectural attacks, malicious firmware or hardware (supply chain attacks)



Integrity vs. Confidentiality

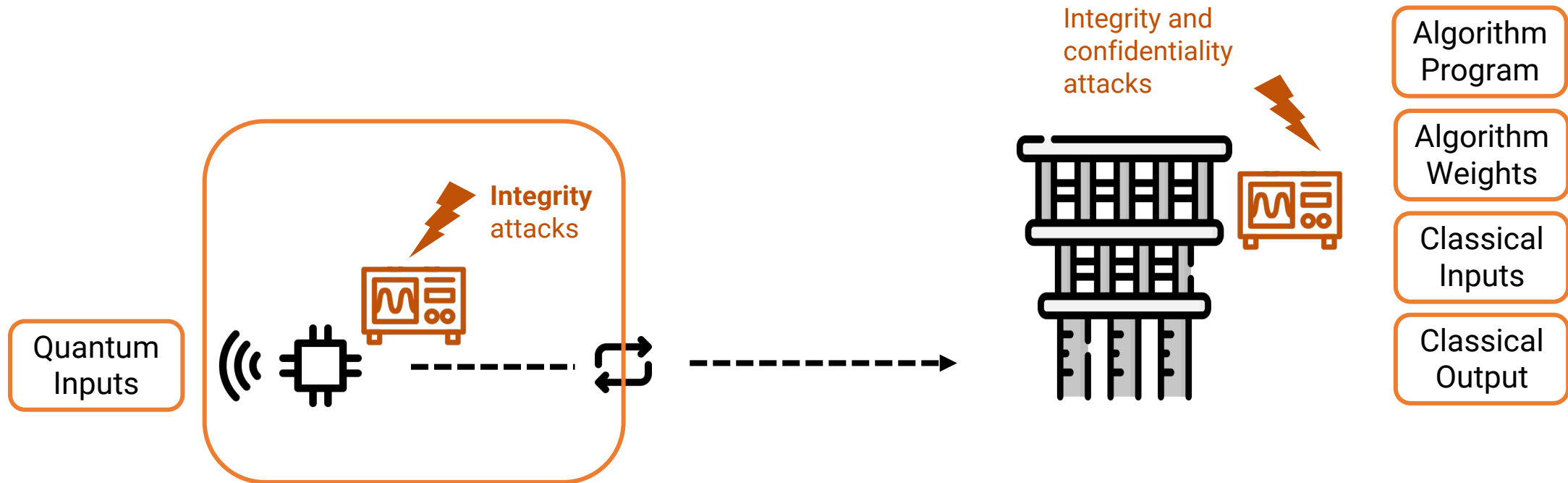
Integrity means the protected asset cannot be altered without detection, whereas confidentiality means the protected asset cannot be observed by unauthorized parties.

- **Integrity affects any data**, e.g., wrong inputs give wrong outputs even if data itself is not secret
- **Integrity is critical even if you don't care about confidentiality**, e.g., was the right data and right code executed
- **Confidentiality affects sensitive data or intellectual property**, e.g., public algorithms can work with private data, or public data can be processed by proprietary algorithm



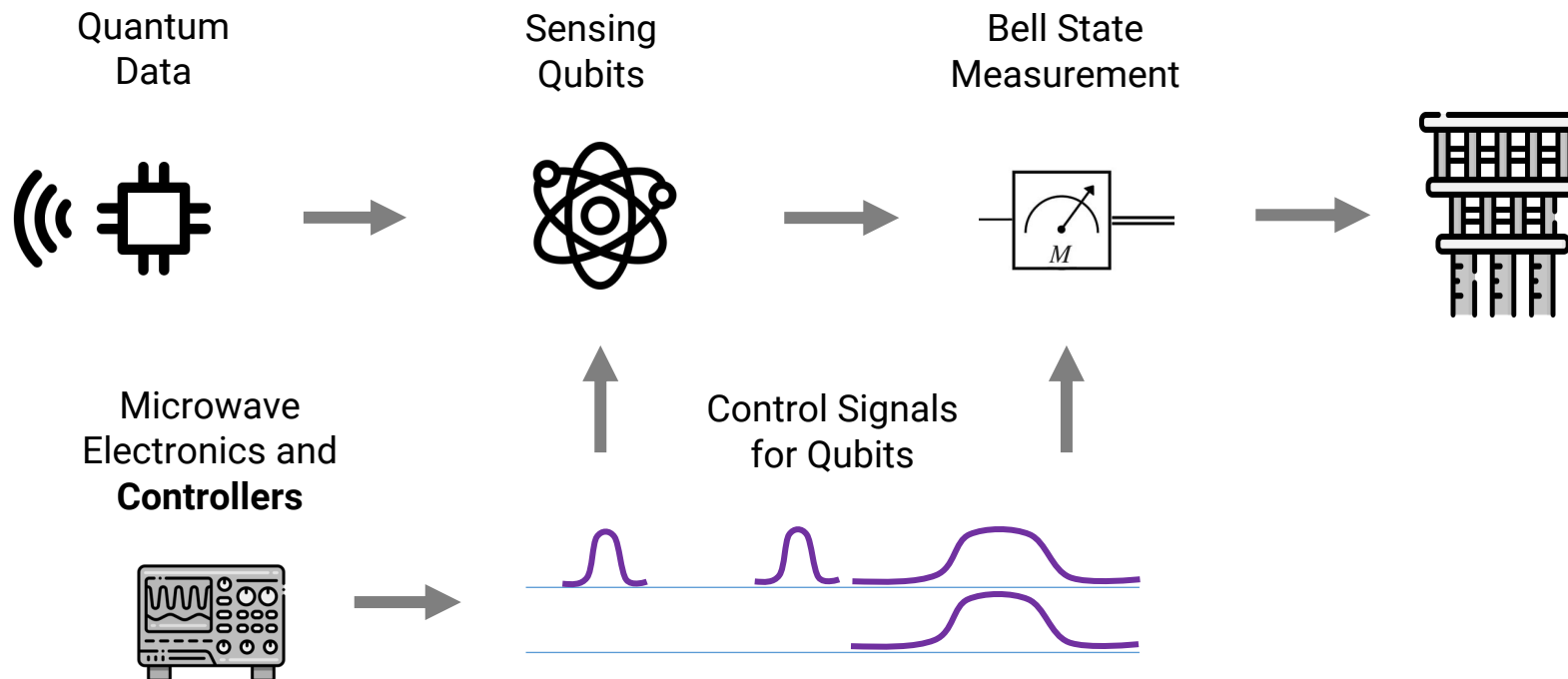
Hardware Security for Protecting Quantum Machine Learning Computations

In context of QML, hardware security must focus on:



Operation of Quantum Sensor Controller

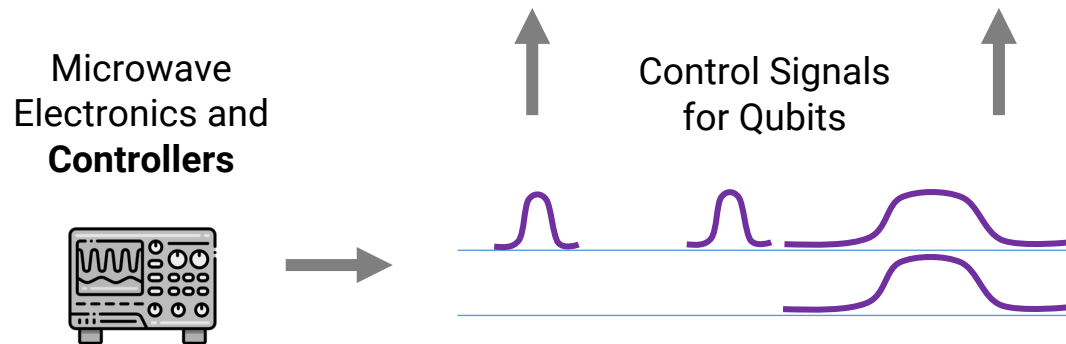
Quantum sensor controller essentially manages when sensing qubits interact with the environment to perform the measurement, and controls bell state measurement and sending data back to the quantum computer



Timing Attack on Quantum Sensor

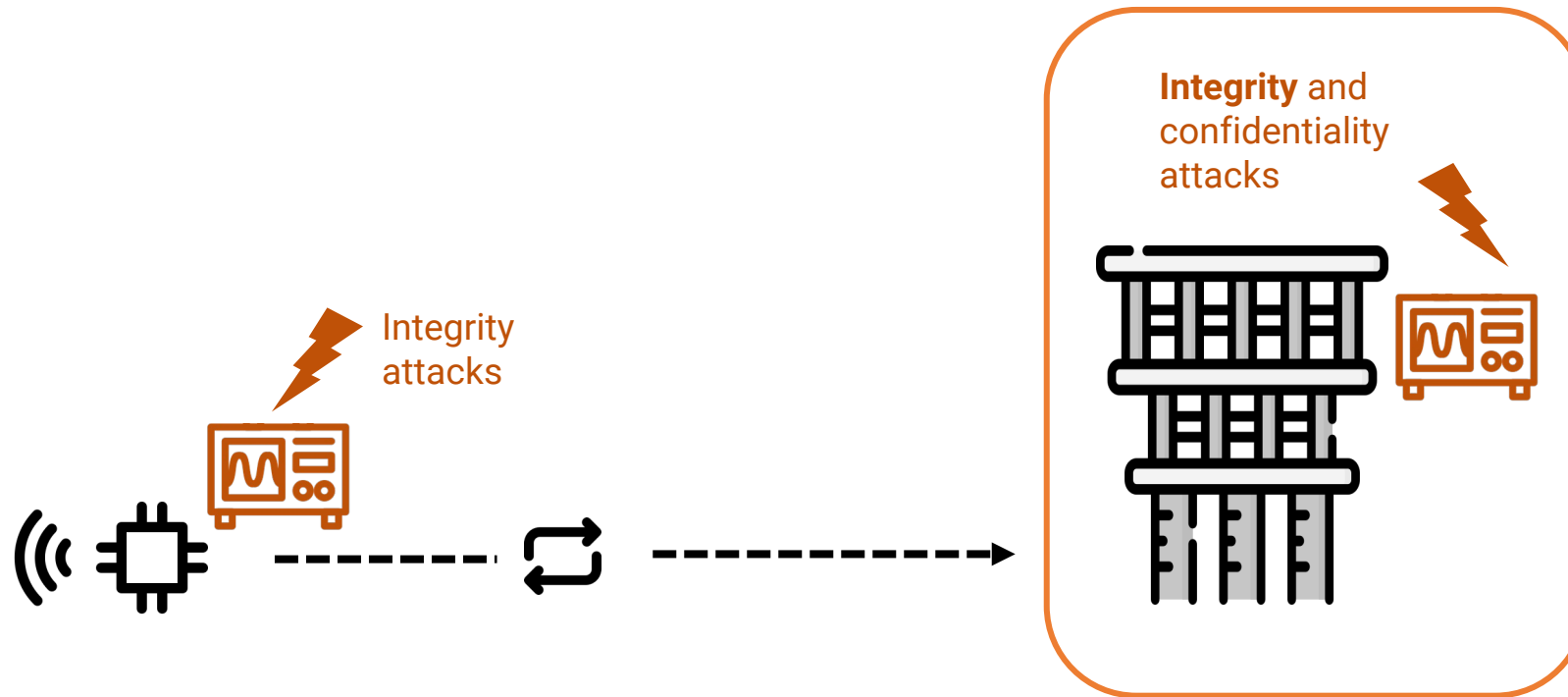
Quantum controller controls the timing, which is the most fragile part

- How long the sensing qubits interacts with the environment changes the the effective value of the data that is captured → **integrity**
- Clock glitching on the controller → effectively remove clock edges → issue control pulses slower
- Voltage glitching → skip conditional statements in microcontroller → issue control pulses faster



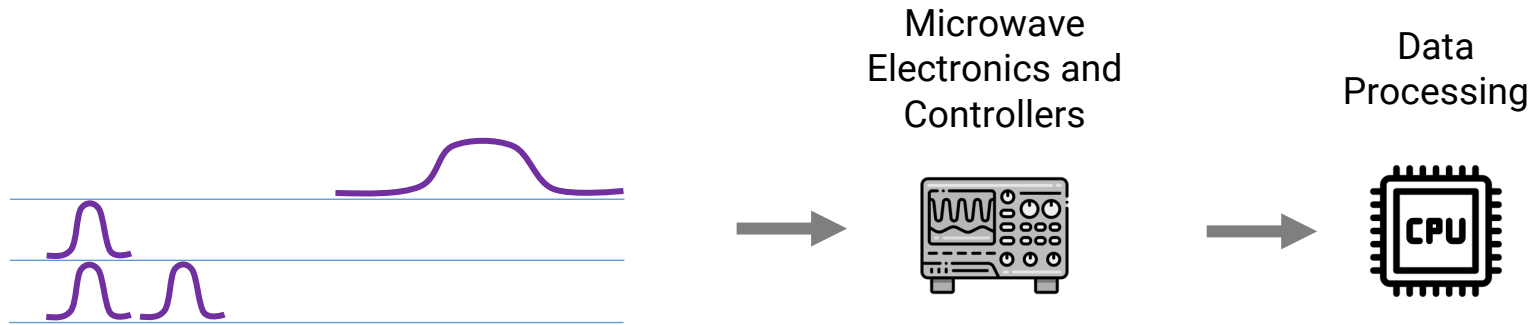
Hardware Security for Protecting Quantum Machine Learning Computations

In context of QML, hardware security must focus on:



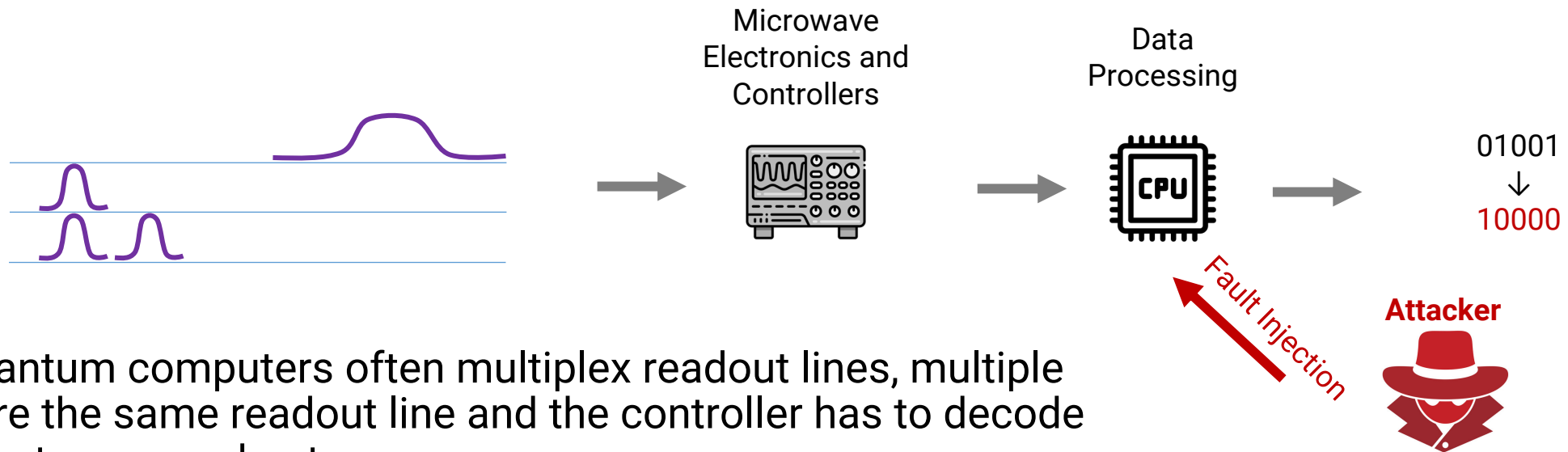
Operation of Quantum Computer Controller and Signal Decoding

The quantum computer controller manages the readout of the qubits from the system



Fault Injection Attacks on the Controller and Decoding

Fault injection occurs when attacker perturbs the operation of the controller, for example by faulting the power supply, it can affect the data read or decoded by the controller



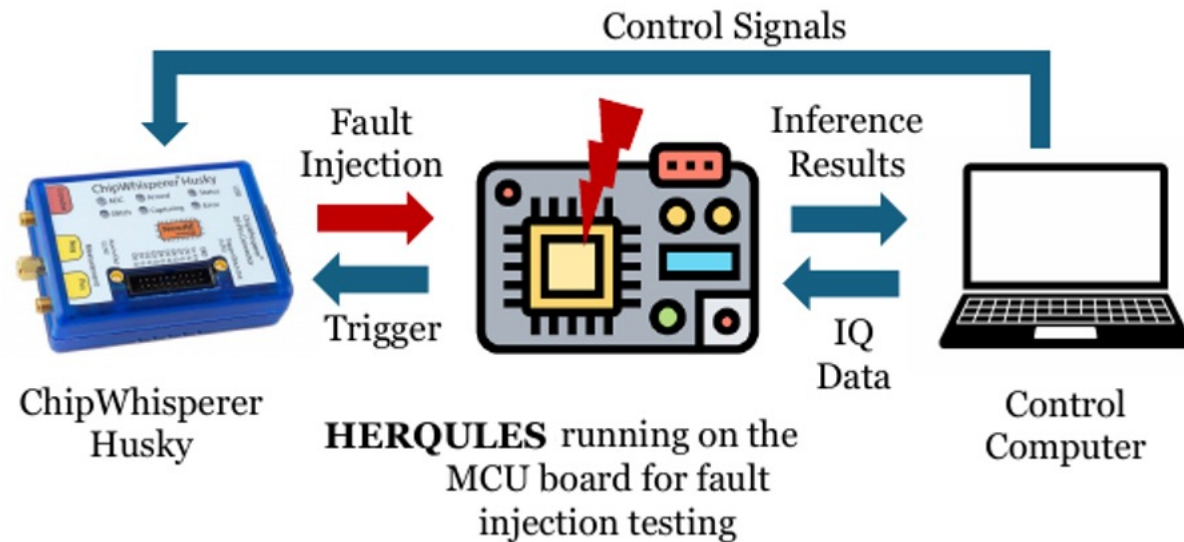
Current quantum computers often multiplex readout lines, multiple qubits share the same readout line and the controller has to decode the qubit that were read out

- HERQULES is one example of decoder that separates multiplexed IQ signals and performs readout error correction
 - **Input:** I and Q values for 5 qubits from shared readout line
 - **Output:** 5-bit string of the error-corrected readout values



Fault Injection on the Readout Discriminator

Faulting the readout discriminator can generate wrong outputs, setup requires injecting voltage glitches during the discriminator's computation:

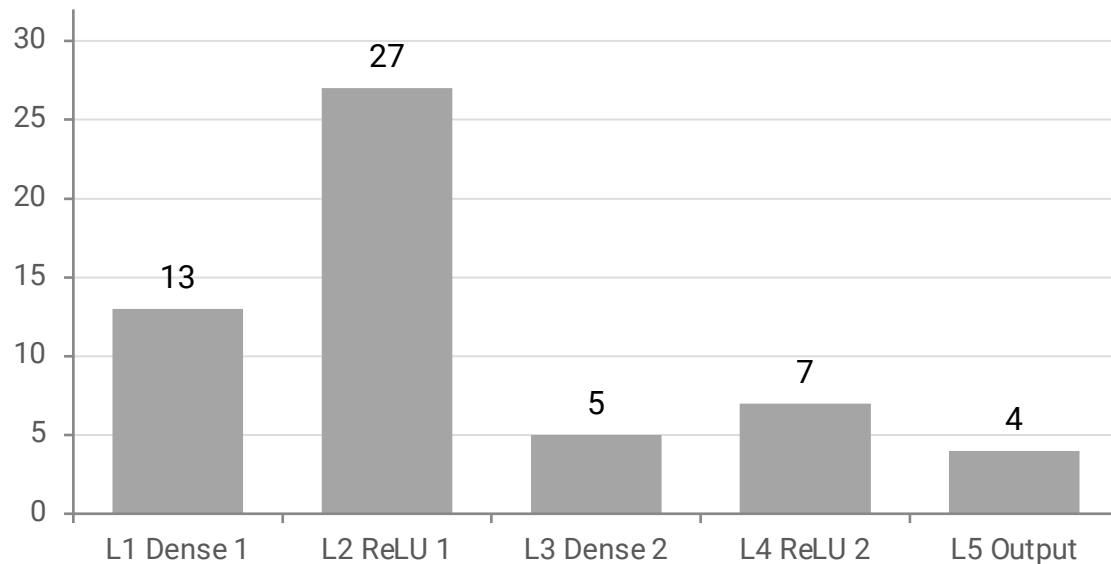


Fault Injection on the Readout Discriminator

HERQULES uses a 5 layer neural network: Dense 1 → ReLU 1 → Dense 2 → ReLU 2 → Output

Fault injection in any layer can trigger faulty outputs:

Best-case successful faults per layer

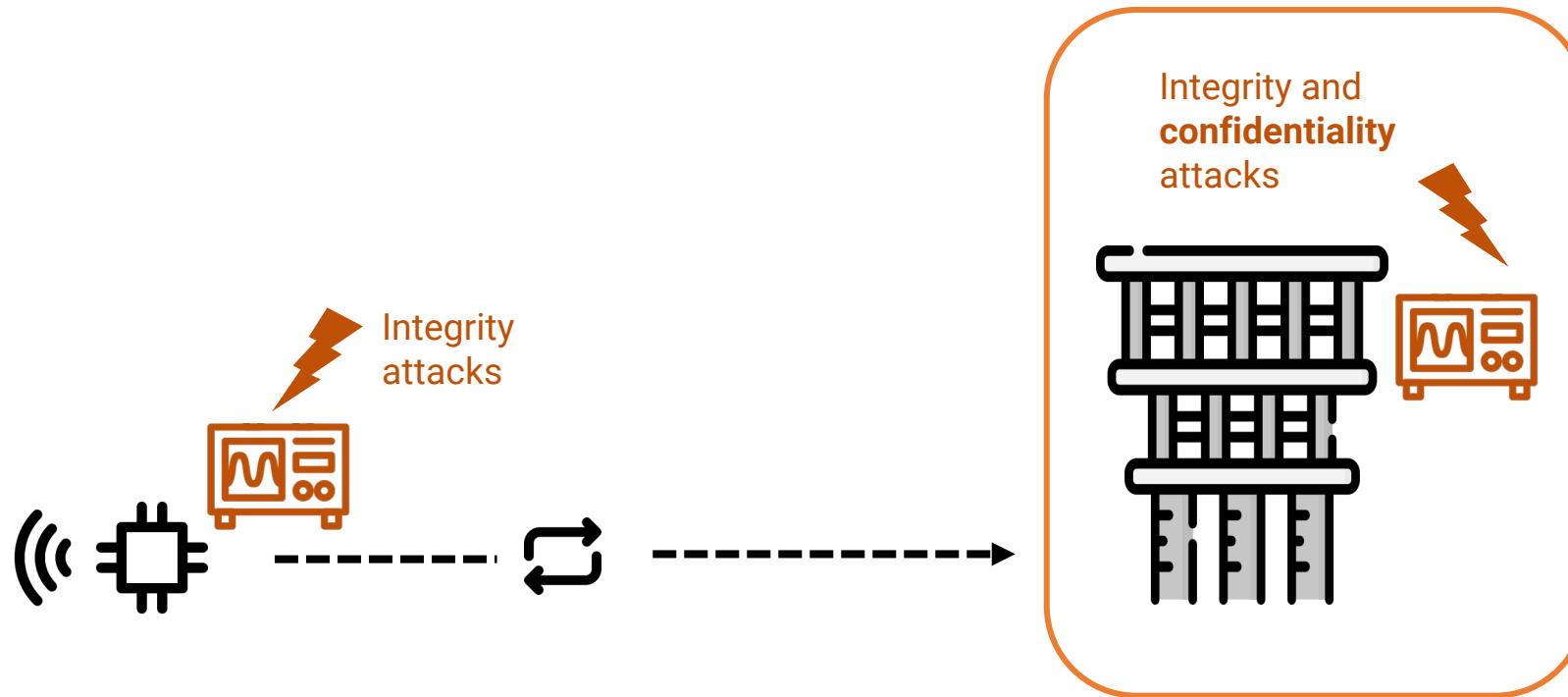


Layer	Width	Offset	Ext. offset	Rpt	Faults
Dense 1	2400	2400	10026	2	13
ReLU 1	2700	2600	14208	5	27
Dense 2	2600	2800	36170	5	5
ReLU 2	2500	2400	39175	5	7
Output	2500	2400	117065	5	4



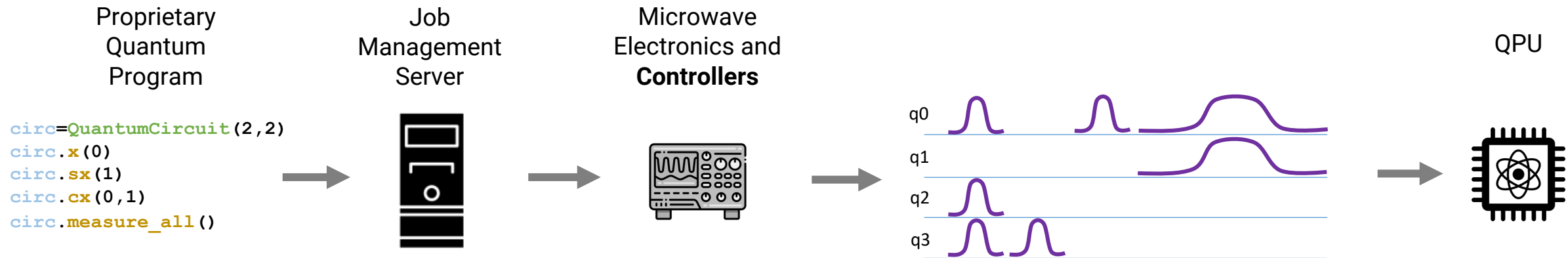
Hardware Security for Protecting Quantum Machine Learning Computations

In context of QML, hardware security must focus on:



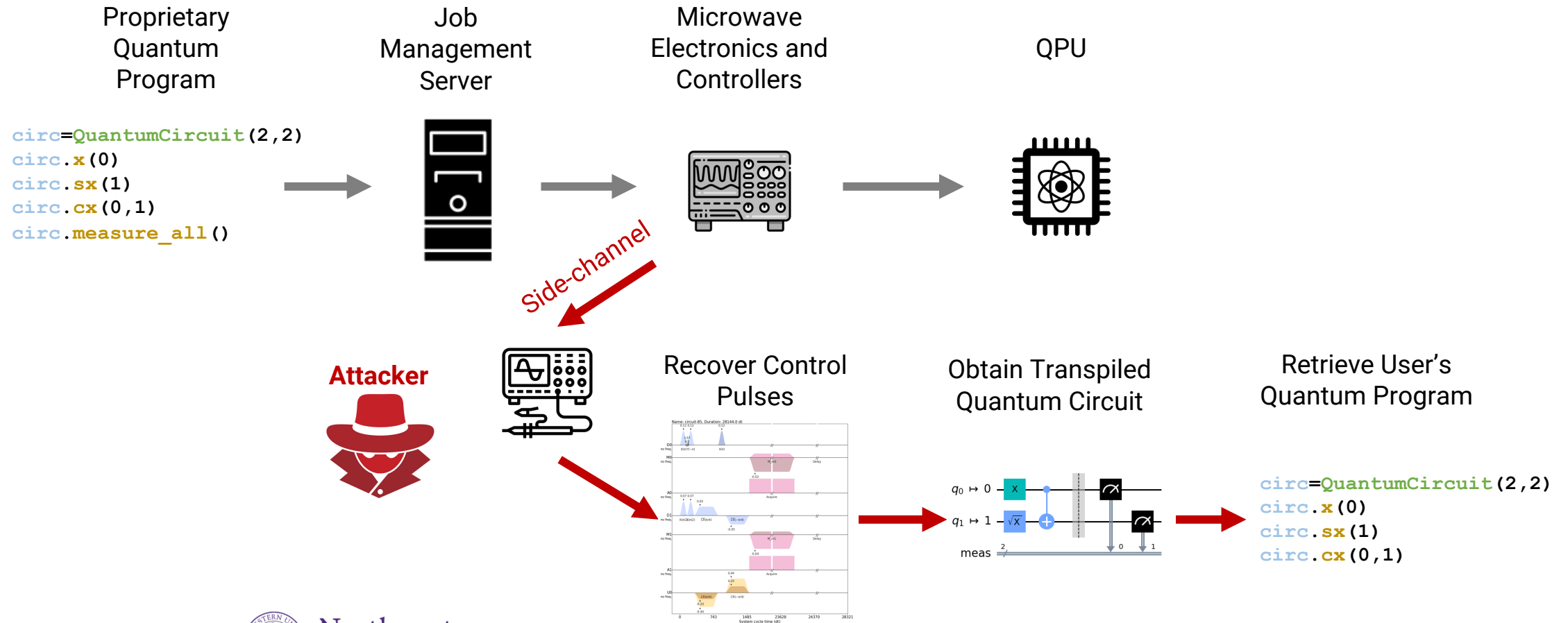
Operation of Quantum Computer Controller

Quantum computer controller essentially converts digital representation of the quantum gates into set of analog RF control pluses, it uses CPU or FPGA + DACs and ADCs



Power Side Channels through Quantum Controller

Power side channels can help attacker discover the control pulse shapes and sizes, leaking information about the potential quantum gates that were executed



Detailed Signal Generation by the Controller

- Arbitrary waveform generators (AWGs) generate the control signals to actuate qubits:

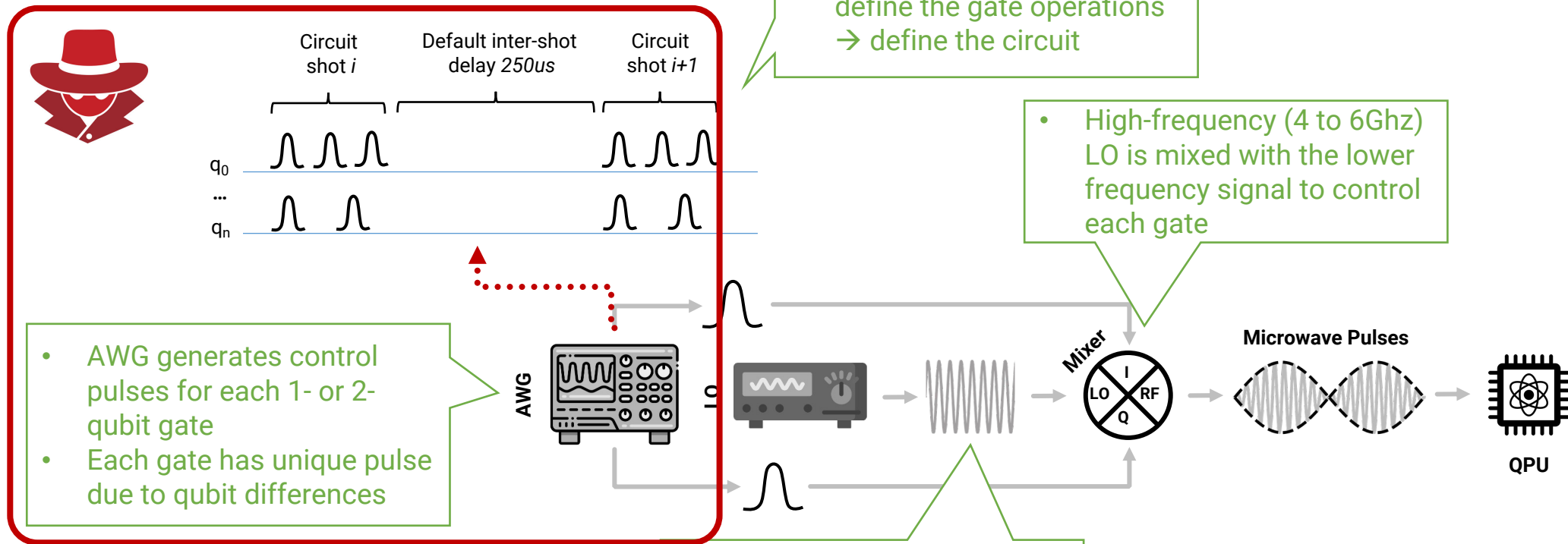
- Shots of circuit are separate by some delay (e.g. reset gate) to reset qubits

- Set of pulses is used to define the gate operations
→ define the circuit

- High-frequency (4 to 6GHz) LO is mixed with the lower frequency signal to control each gate

- AWG generates control pulses for each 1- or 2-qubit gate
- Each gate has unique pulse due to qubit differences

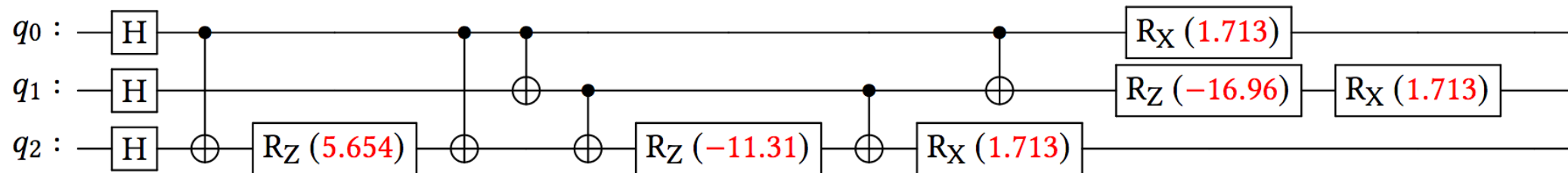
- Local Oscillator (LO) oscillates at the frequency of the target qubit



Example Circuit Ansatz Identification

Optimization circuits often involve an ansatz, which is a parametrized circuit

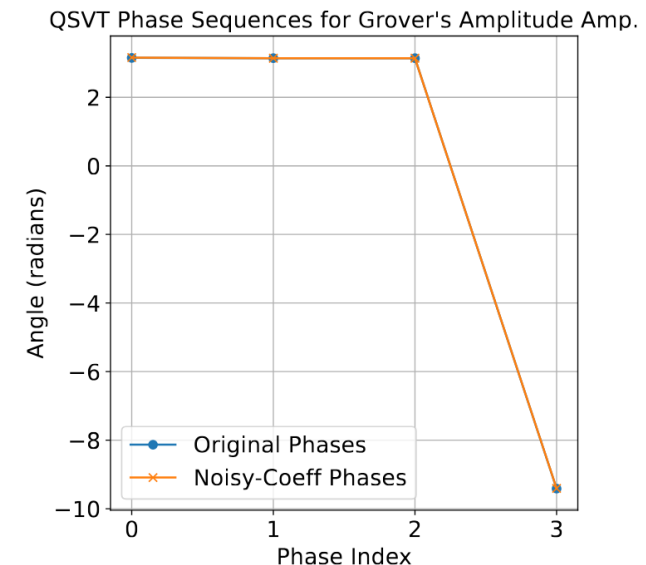
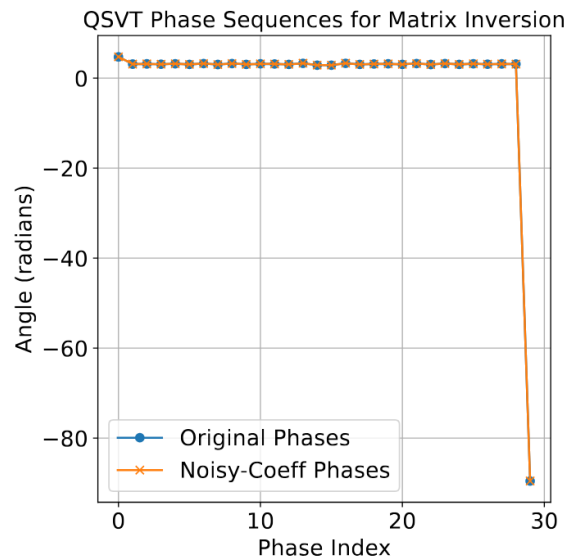
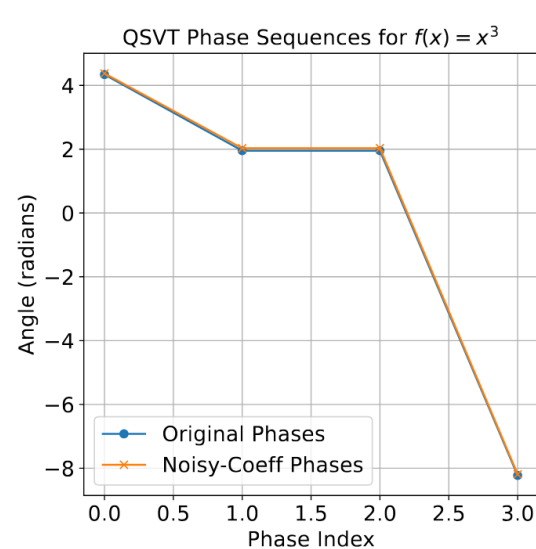
- Attackers **can** identify which ansatz was used, but the parameters of the ansatz **cannot** be easily recovered by the attackers, frequent use of virtual RZ gates in the ansatz makes them naturally less vulnerable to attacks
- Example 3-qubit quantum approximate optimization algorithm ansatz:



Attacks in Context of QML

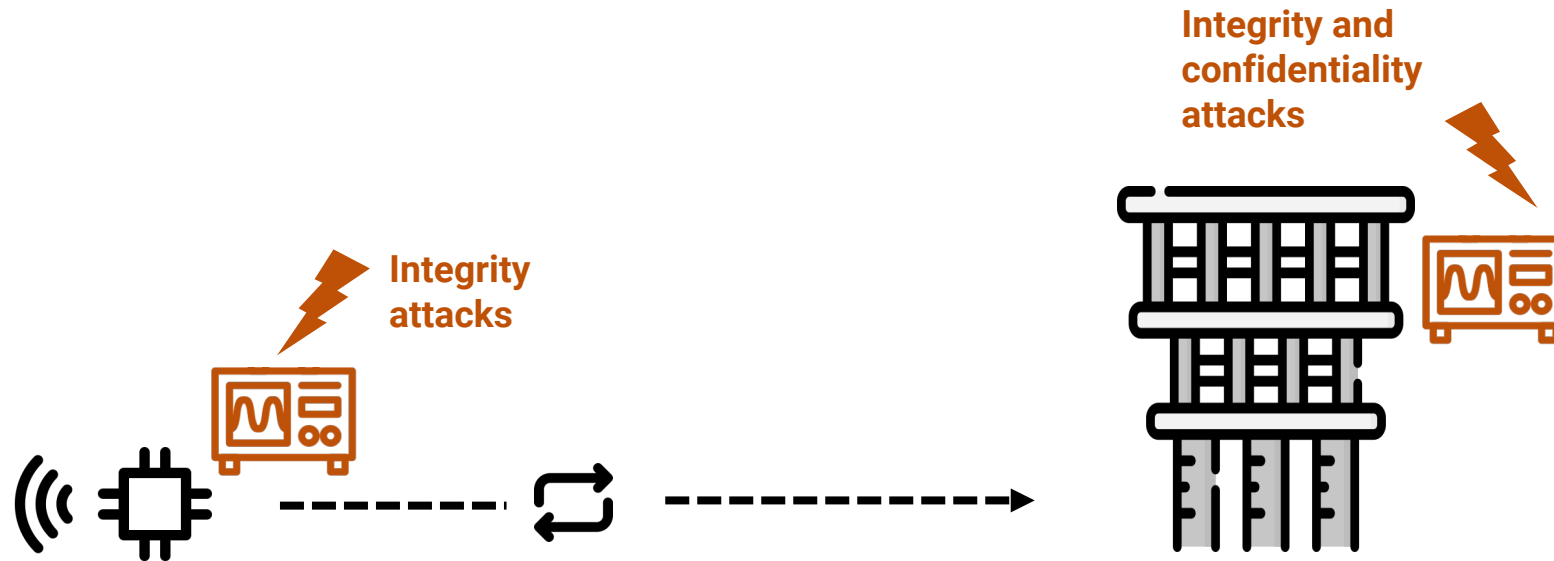
Quantum Singular Value Transformation (QSVT) is a powerful framework that can be applied across a wide range of quantum applications, including solving linear systems of equations, phase estimation, or amplitude amplification as employed in Grover's search algorithm.

QSVT is in effect a meta-algorithm that can be used to realize various other functionalities or applications, but its configuration leaks the type of computation:



Hardware Security for Protecting Quantum Machine Learning Computations

In summary, in context of QML, hardware security must focus on:



Hardware Security for Protecting Quantum Machine Learning Computations

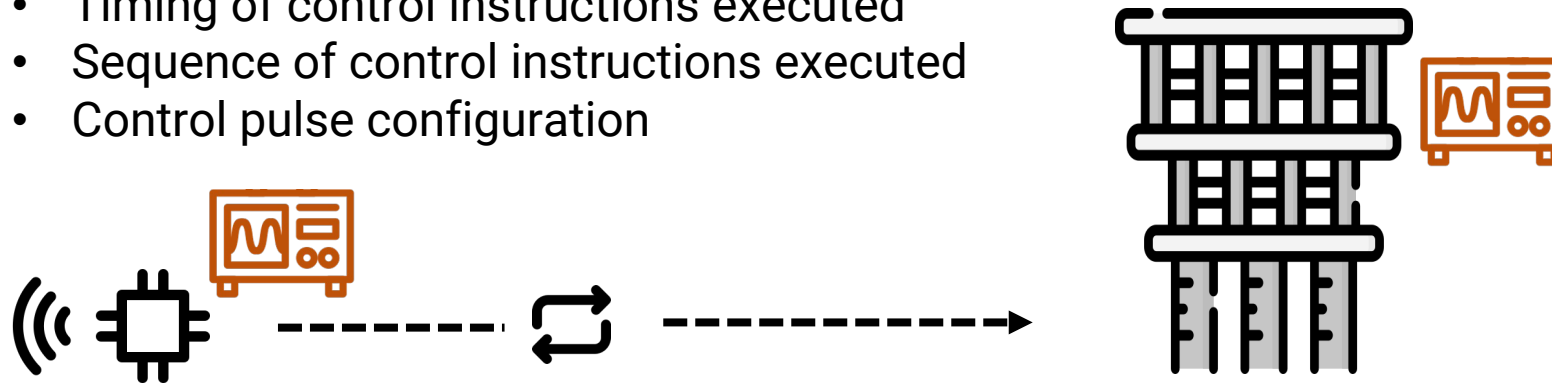
In summary, in context of QML, hardware security must focus on:

Need integrity measurement:

- Timing of control instructions executed
- Sequence of control instructions executed
- Control pulse configuration

Need integrity measurement:

- Timing of instructions executed
- Sequence of instructions executed
- Configuration of the pulses



Need confidentiality protection:

- Masking or obfuscation of the control pulses
- Masking or obfuscation of the controller's code
- Fault detection sensors, or redundancy



Tutorial Outline

- **1:00pm – 1:30pm: Introduction and Motivation**
- **1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers**
- **2:30pm – 3:00pm: Coffee Break**
- **3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers**
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
- **Extra: Hardware Security for QML**
- **4:00pm: End**



Summary Many Attacks and Defenses Exist, But Many Questions Remain

Secure Quantum Computing is an emerging field that aims to address security and privacy concerns arising from the delegation of quantum programs to untrusted remote servers.

DEFENSES

 **Quantum
Computer Antivirus**
HOST'23

 **Reset Gate
Defenses**
SEED'24

 **CASQUE**
HOST'24

 **QC-TEE**
CAL'23

Software
Level

System / Controller
Level

Hardware / Qubit
Level

ATTACKS

 **Pulse-Level
Trojans**
S&P'25

 **Attacks On HHL**
QUILLS'24

 **Power Side-
Channel Attacks
on Controllers**
TCHES'24

 **Higher-Energy
Attacks on QRAM**
ICCAD'24

 **Higher-Energy
Attacks**
CCS'23

 **Reset Gate
Attacks**
CCS'22

 **Long-Range
Crosstalk**
arXiv'25



Resource: Quantum Security Tutorials Web Page

CASLAB Home Our Group Publications Books Tutorials

Next Tutorial in:
no upcoming tutorial is currently scheduled

Next:

[Tutorial on Security of Quantum Computing Systems](#)

September 17, 2026, co-located with QCE 2026.

Future:

[Winter School on Attacks, Defenses, and Secure Design of Quantum Computing Systems](#)

November 11, 2026, co-located with QCCS 2026.

Past:

[Tutorial on Quantum Computer Architecture Security](#)

June 27, 2026, co-located with ISCA 2026.

[Tutorial on Security of Quantum Computing Systems](#)

August 31, 2025, co-located with QCE 2025.

[Tutorial on Quantum Computer Architecture Security](#)

June 21, 2025, co-located with ISCA 2025.

© 2026 Jakub Szefer

Tutorials on Classical and Quantum Computer Architecture and Hardware Security

About: A series of tutorials on classical and quantum computer architecture and hardware security, presented by Prof. Jakub Szefer of the Computer Architecture and Security Lab (CASLAB) at Northwestern University.

Tutorials on Classical and Quantum Computer Architecture and Hardware Security

Prof. Jakub Szefer (jakub.szefer@northwestern.edu)
Electrical and Computer Engineering
Northwestern University

Computer Architecture and Security Lab (CASLAB) Northwestern University



Prior tutorials include: ISCA 2026, ISCA 2025, HOST 2025, ICCAD 2024, QCE 2023, HOST 2021, 2019, ACACES 2019, HOST 2019, HiPEAC 2019, and HOST 2018.



Computer Architecture
and Security Lab (CASLAB)



Northwestern
University

Resource: Quantum Computer Hardware Cybersecurity BibTeX

qc-hardware-cybersecurity-bibtexPublic

PinUnwatch4Fork7Star15

main1 Branch0 Tags

Go to fileAdd fileCode

caslab-codeUpdating research paper list with recent and older papers, mainly 202...14a4309 · now34 Commits

README.md	Updating email address in README file.	last year
refs-qc-hardware-cybersecurity-papers.bib	Updating research paper list with recent and older papers,...	now
refs-qc-hardware-cybersecurity-surveys.bib	Update refs-qc-hardware-cybersecurity-surveys.bib	11 months ago

README

About

This repository contains a BibTeX bibliography file with references to research papers pertaining to security of quantum computers. Anybody is encouraged to contact the author or make pull request to add papers they would like to be considered for inclusion in the bibliography. Papers on certain topics such as post-quantum cryptography or quantum key distribution are excluded from this bibliography so that the focus can remain only on research papers dealing with attacking and defending quantum computer systems, architectures, and hardware.

Contact

To have papers considered for addition to this list, please create a pull request or or e-mail the author at: jakub.szefer@northwestern.edu

About

List of references for quantum computer hardware cybersecurity papers.

ReadmeActivity15 stars4 watching7 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first pack](#)

Contributors4

caslab-code

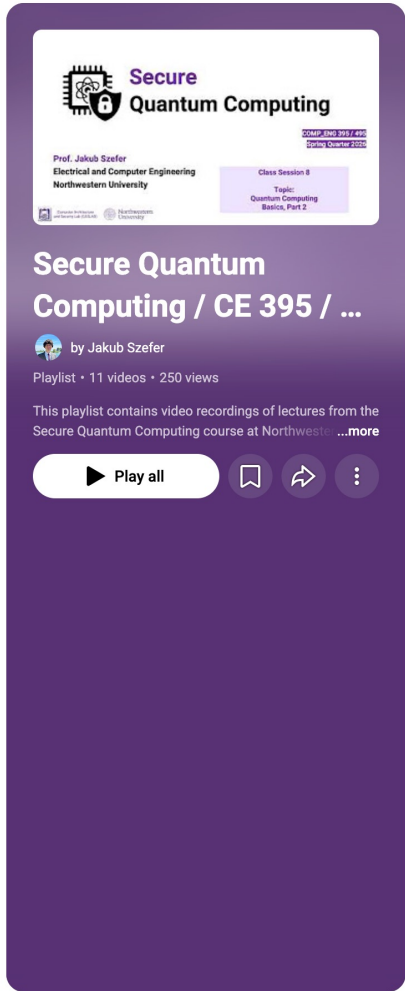
cnpapado Kc

chuanqixu C

yizhuo1007



Resource: YouTube Lectures



 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 8 Topic: Quantum Computing Basics, Part 2 Duration: 1:07:23	Secure Quantum Computing / Spring 2026 / Class 8 / Quantum Computing Basics, Part 2 Jakub Szefer 18 views • 2 weeks ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 7 Topic: Quantum Computing Basics, Part 1 Duration: 53:19	Secure Quantum Computing / Spring 2026 / Class 7 / Quantum Computing Basics, Part 1 Jakub Szefer 43 views • 3 weeks ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 6 Topic: Threat Categories for Quantum Computing Duration: 59:45	Secure Quantum Computing / Spring 2026 / Class 6 / Threat Categories for Quantum Comp. Sys., Part 2 Jakub Szefer 173 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 4 Topic: Threat Categories for Quantum Computing Duration: 54:28	Secure Quantum Computing / Spring 2026 / Class 4 / Threat Categories for Quantum Comp. Sys., Part 1 Jakub Szefer 90 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 3 Topic: Computer Security Basics, Part 2 Duration: 1:00:04	Secure Quantum Computing / Spring 2026 / Class 3 / Computer Security Basics, Part 2 Jakub Szefer 134 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 2 Topic: Computer Security Basics, Part 1 Duration: 1:12:52	Secure Quantum Computing / Spring 2026 / Class 2 / Computer Security Basics, Part 1 Jakub Szefer 268 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 1 Topic: Introduction to Secure Quantum Computing Duration: 1:14:13	Secure Quantum Computing / Spring 2026 / Class 1 / Introduction to Secure Quantum Computing Jakub Szefer 371 views • 1 month ago	⋮



Resource: Upcoming QCCS Symposium 2026

[QCCS 2026](#) [Program](#) [Venue](#) [Directions](#) [Contact](#) [Prior QCCS 2025](#) [Prior QCCS 2024](#) [Prior QCCS 2023](#)

QCCS 2026 **4th Annual Quantum** **Computer Cybersecurity** **Symposium**

To be held November 11 and 12,
2026 at The Guild Lounge in
Scott Hall at Northwestern
University

