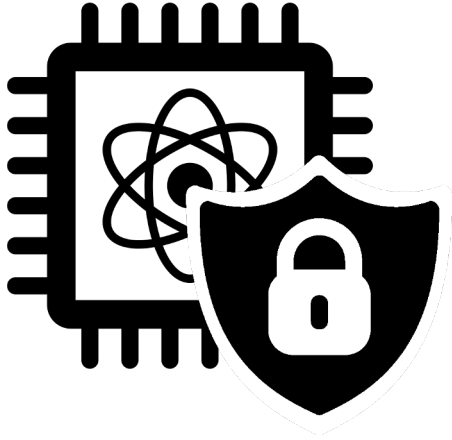




Tutorial on Security of Quantum Computing Systems

QCE Tutorial

Sept. 17, 2026

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

Electrical and Computer Engineering

Northwestern University



Computer Architecture
and Security Lab (CASLAB)

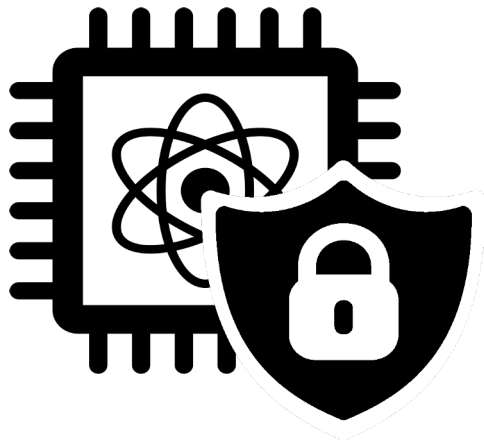


Northwestern
University

Tutorial Outline

- 1:00pm – 1:30pm: Introduction and Motivation
- 1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers
 - Crosstalk and Fault Injection Among Qubits
 - Fault Injection in Reset Gates
- 2:30pm – 3:00pm: Coffee Break
- 3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers
 - Side Channels in NISQ Controllers
 - Side Channels in FTQC
- 3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers
- 4:00pm: End

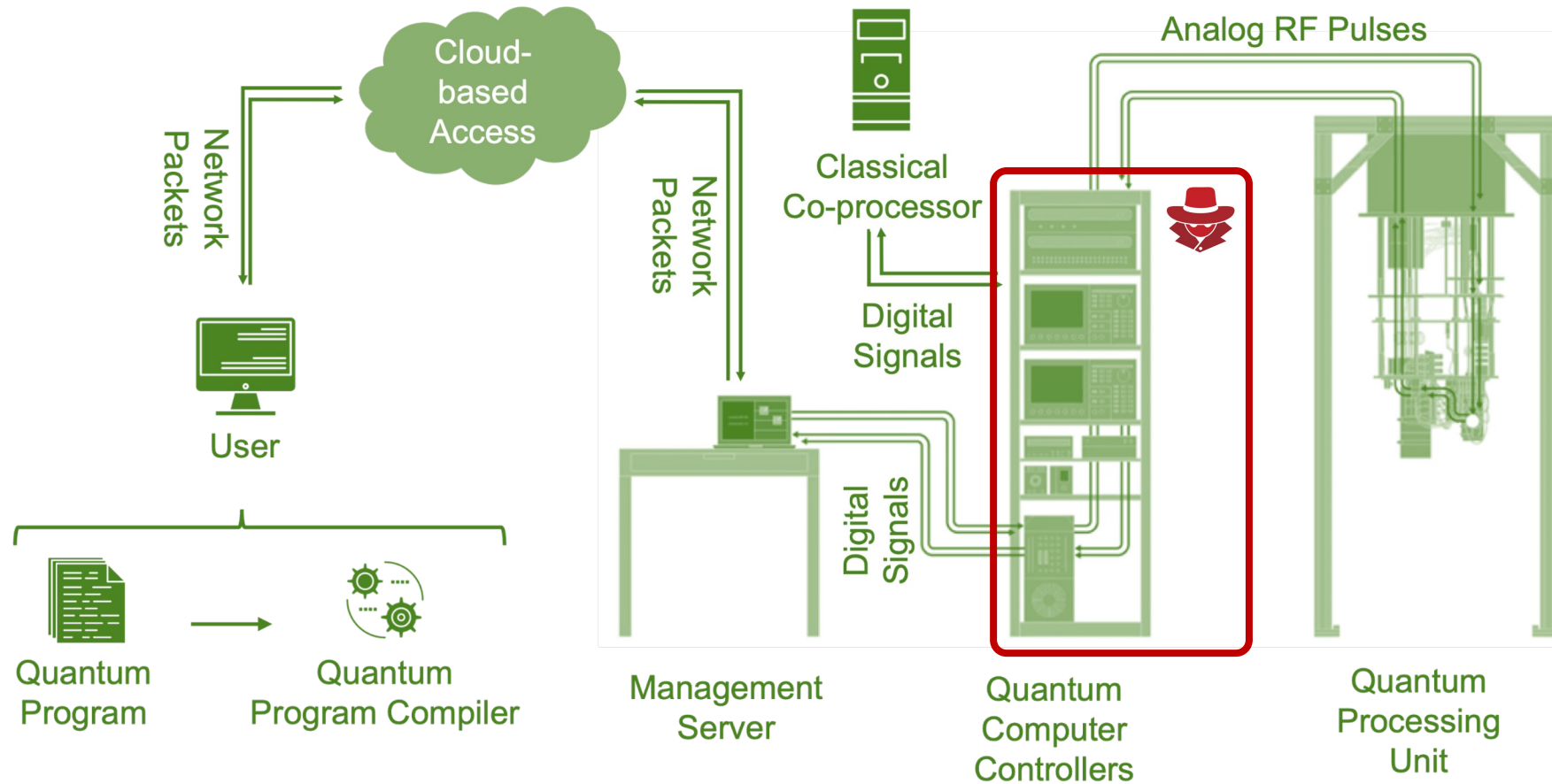




Side Channels in NISQ Quantum Computer Controllers



Side Channels in Quantum Controllers

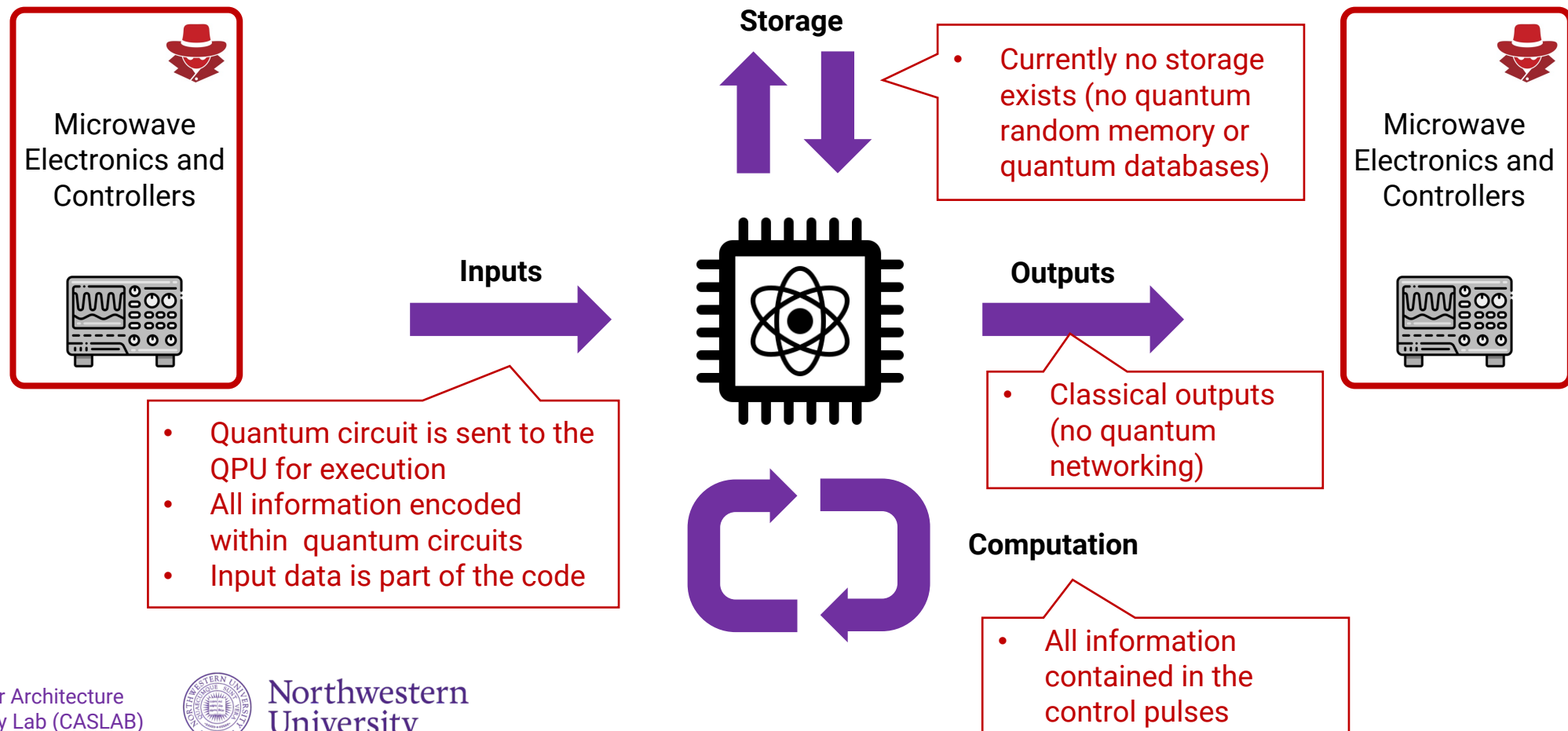


S(ide channel) – **P**(hysical) – **C**(lassical)



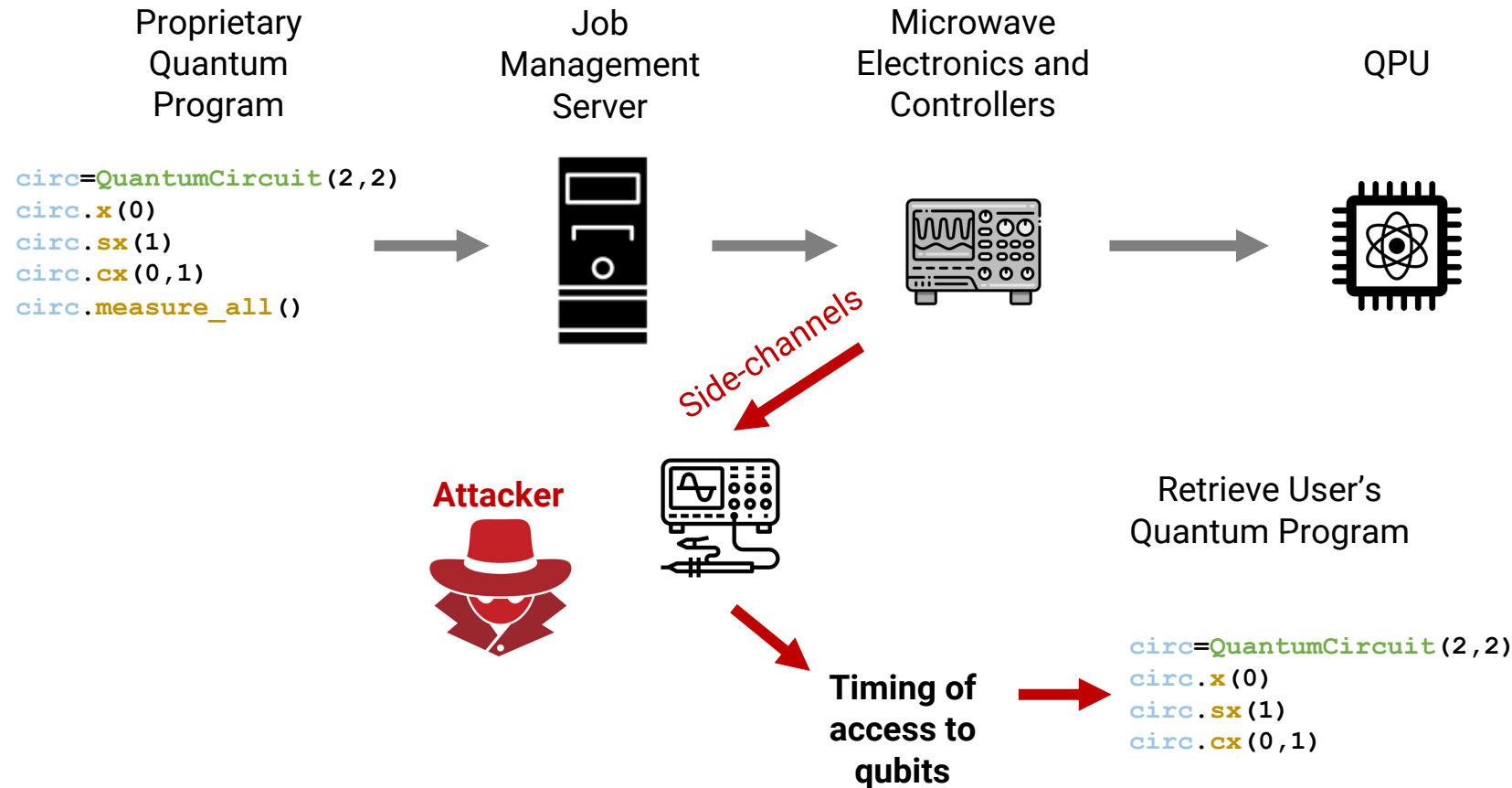
Controllers are the Interface to the QPU

- Controllers are the interface to the QPU, all inputs and outputs pass through controller:



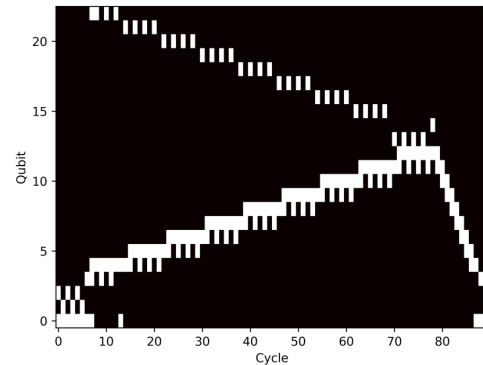
Timing Side Channels through Quantum Controller

- Collecting side-channel information from the controller electronics:

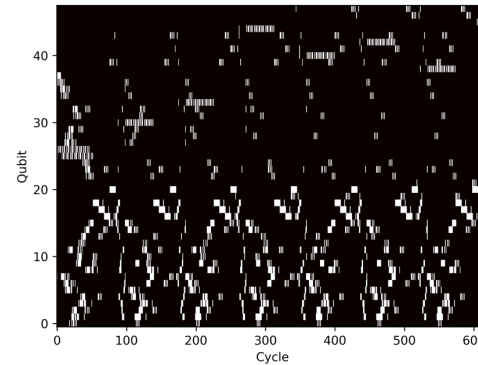


Memory Access Pattern Analysis

The pattern of memory access references timestamps can reveal insights about program structure:

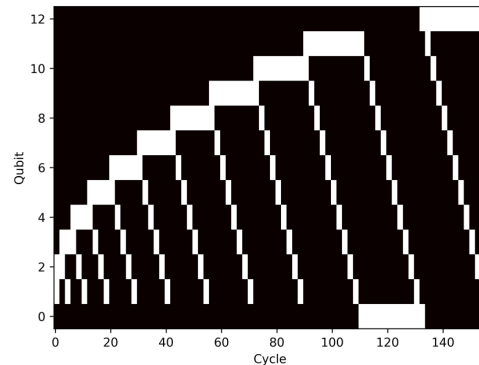


(a) Adder

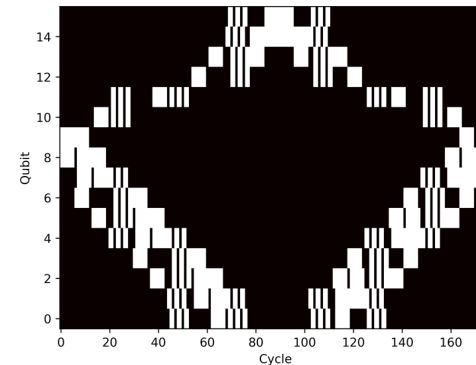


(b) Product

Memory access pattern means which qubits are accessed at which time



(c) QFT

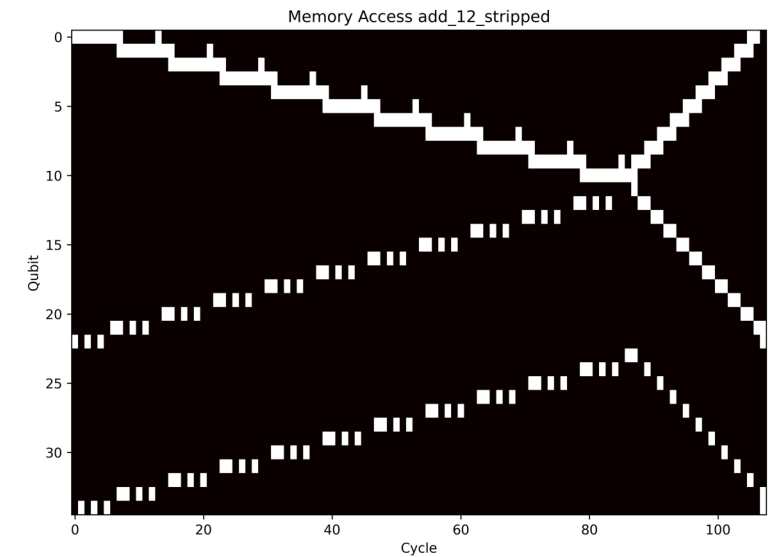
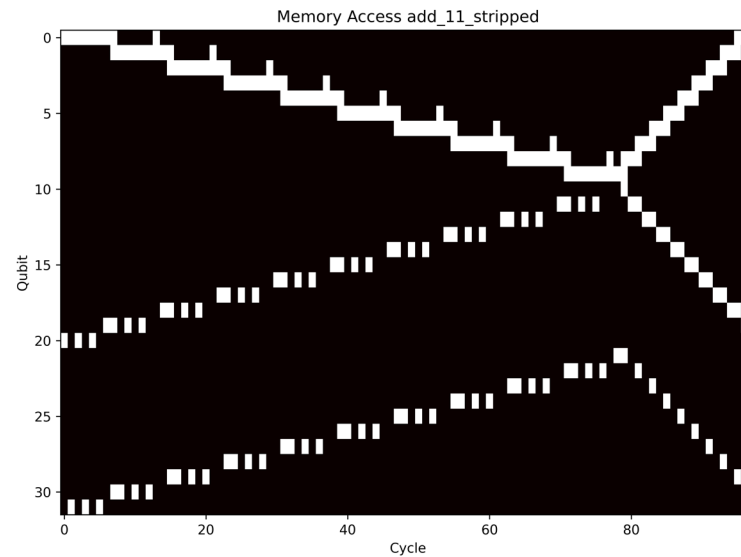
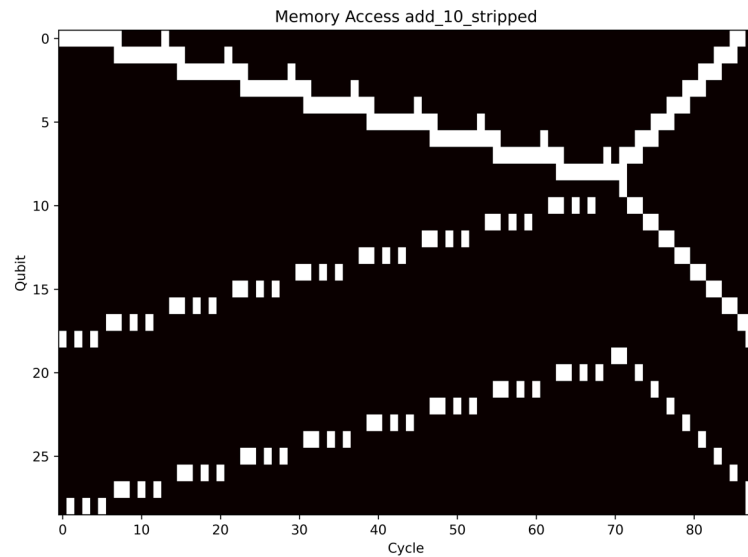


(d) Trotter



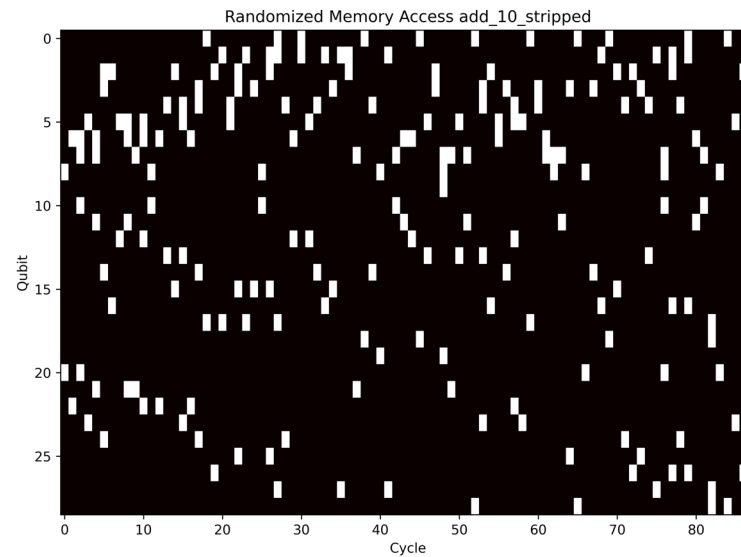
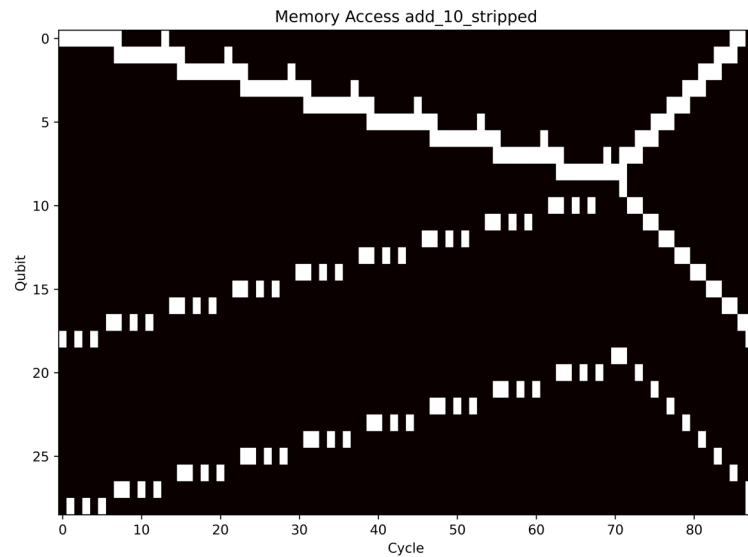
Memory Access Patterns

- For a fixed benchmark, there are repeating access patterns for **different benchmark sizes**:



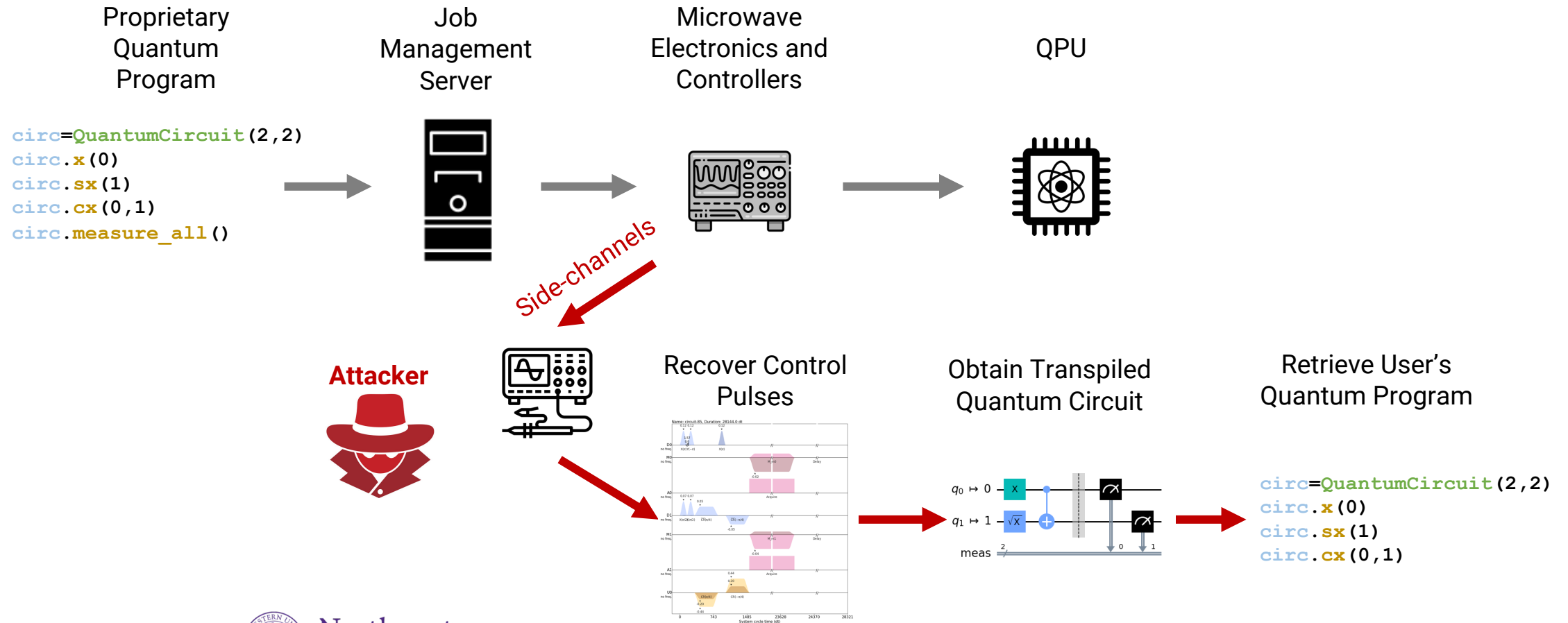
Memory Access Patterns

- For a fixed benchmark, **visual patterns disappear with randomization** of qubit allocation (bit can still be found through access pattern analysis):



Power Side Channels through Quantum Controller

- Collecting side-channel information from the controller electronics:



Detailed Signal Generation by the Controller

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

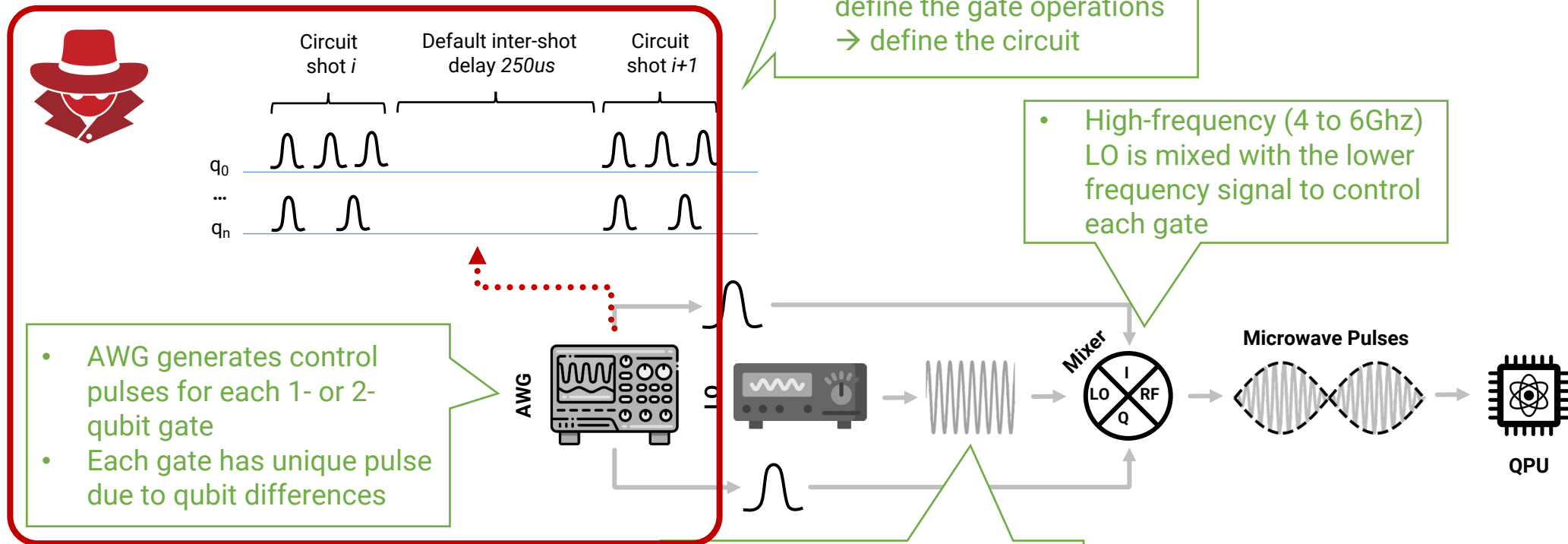
- Shots of circuit are separate by some delay (e.g. reset gate) to reset qubits

- 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



Attacker Goals for Information Leak

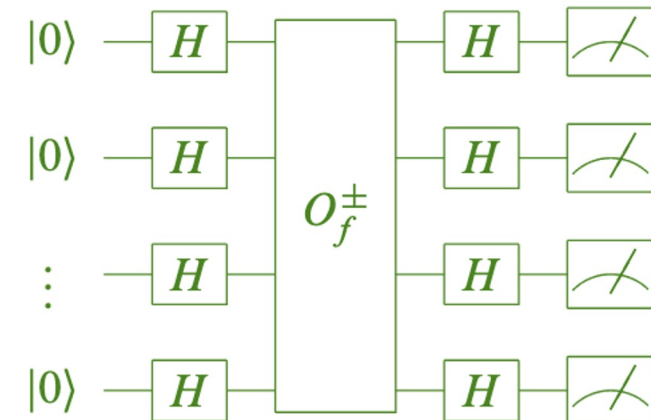
- **(UC) User Circuit Identification** – Given knowledge about the set of possible circuits executed on the quantum computer, find which circuits the user actually executed
- **(CO) Circuit Oracle Identification** – Given a known circuit, such as Bernstein-Vazirani, but an unknown oracle, find the configuration of the oracle used in that circuit
- **(CA) Circuit Ansatz Identification** – Given a known circuit, such as a variational circuit used in machine learning applications, but an unknown ansatz, find the configuration of the ansatz used in that circuit



Oracle Oracle (CO) Example: Bernstein-Vazirani Algorithm

The Bernstein-Vazirani algorithm is a quantum algorithm that solves a specific hidden-string problem using only one query to a black-box function. It is one of the simplest examples showing how quantum parallelism can extract global information efficiently.

- **Problem statement:** we have an oracle implementing a function $f: \{0,1\}^n \rightarrow \{0,1\}$ which is promised to be implement a hidden string function $f(x) = s \cdot x = \sum_{i=1}^n s_i \cdot x_i \mod 2$,
- **Classical solution:** in the classical algorithm, we need to query n times (try sequentially inputs where you set each bit of input to 1 to learn corresponding bit of s)
- **Quantum solution:** in the quantum Bernstein-Vazirani algorithm, we need 1 query



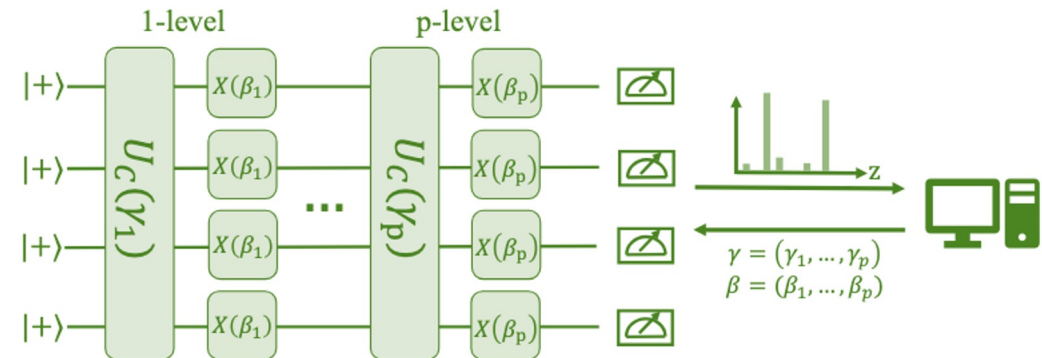
Reference: Xu et al. (CCS 2023)



Oracle Ansatz (CA) Example: QAOA

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical algorithm designed to solve combinatorial optimization problems by preparing quantum states with a parameterized ansatz.

- **Problem statement:** QAOA seeks to approximate the optimal solution by preparing a quantum state using a parameterized circuit (ansatz) and then measuring bitstrings that are likely to have high-cost values.
- **Classical solution:** in the classical algorithm, solutions are often exponential a^n
- **Quantum solution:** NISQ hardware is too small to be able to test large circuits, but solution can perhaps be polynomial



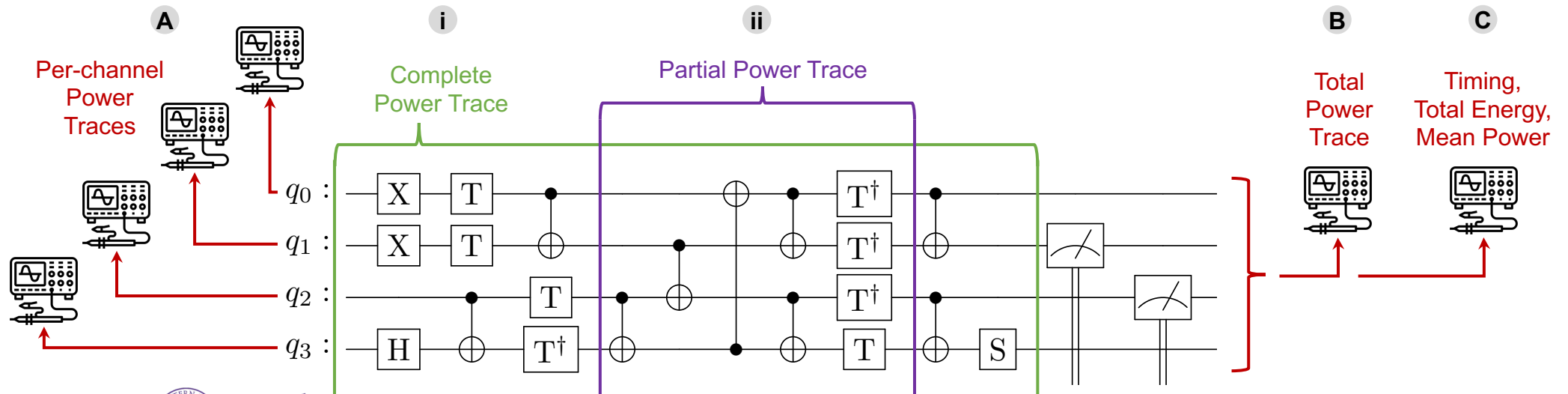
Attacker Goals for Information Leak

- **(QM) Qubit Mapping Identification** – Given a known circuit, identify the placement of which physical qubits were used
- **(QP) Quantum Proc. Identification** – Given knowledge about the pulses for quantum processors and a circuit, find the quantum processor on which the circuit was executed
- **(CR) Circuit Reconstruction** – Given knowledge about the pulses for quantum computer basis gates, reconstruct the complete, unknown circuit from the power traces



Side Channel Taxonomy for Controller Attacks

- We can consider different capabilities of the attackers who can (A) measure per-channel traces, (B) measure total power trace, and (C) measure individual quantities such time, energy, power
- Attacker can target different parts of (i) the complete circuit or (ii) portion of the circuit



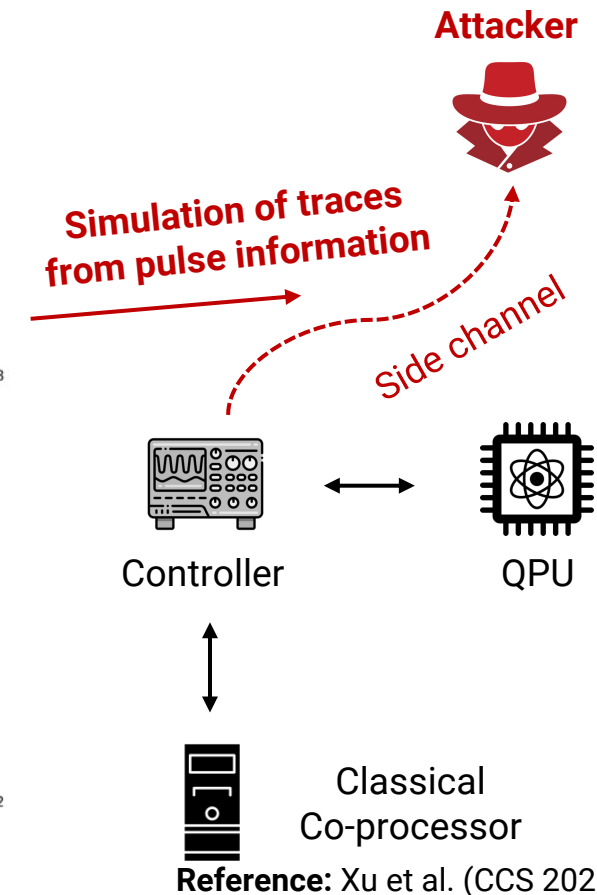
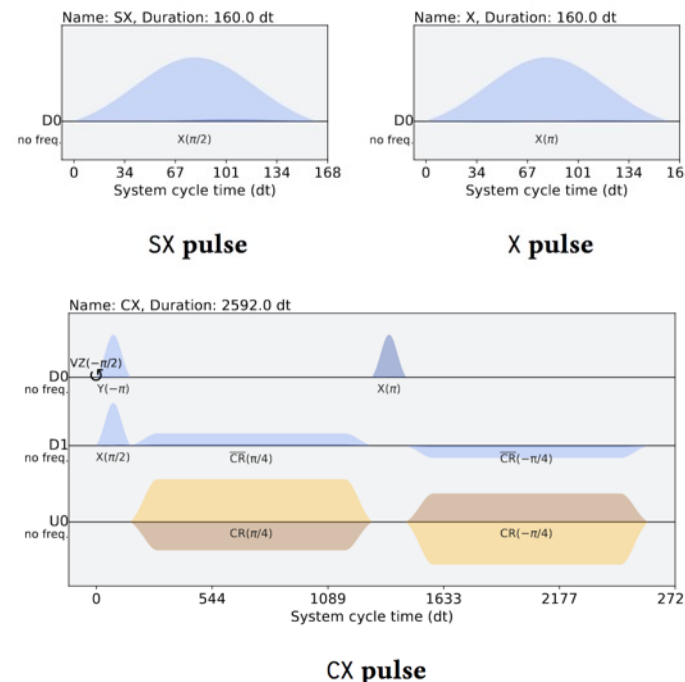
Side Channel Taxonomy for Controller Attacks

- C** {
 - Timing Attack** – Measure execution time of one shot of a circuit to recover user circuits (**UC**)
 - Total Energy Attack** – Measure total energy consumption of the control equipment over execution of one shot of a circuit to recover users' circuits (**UC**) as well
 - Mean Power Attack** – Measure mean power consumption of the control equipment over execution of one shot of a circuit to recover users' circuits (**UC**) as well
- B** {
 - Total Power Trace Attack** – Measure a trace of the total power consumption of the control equipment over execution of one shot of a circuit of all the channels to recover user circuits (**UC**), circuit oracle (**CO**), circuit ansatz (**CA**), qubit mapping (**QM**), and quantum processor (**QP**) with some accuracy
- A** {
 - Per-Channel Power Trace Attack** – Measure a traces of the power consumption of the control equipment over execution of one shot of a circuit of each channels to perform circuit reconstruction (**CR**), thus recovering user circuits



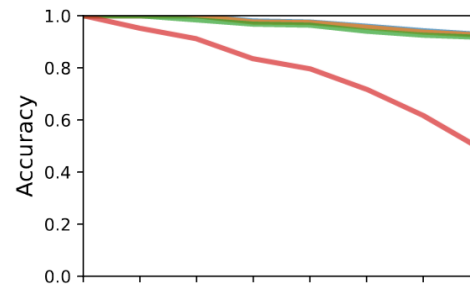
Simulation of Control Pulses

- Control pulse information (amplitude and duration) can be obtained from quantum computer provider or vendor
 - First order assumption power consumption proportional to the pulse amplitude and duration
 - Constant pulse:
 - Energy: $E \propto A^2 \times T$
 - Power: $P \propto \frac{E}{T} \rightarrow P \propto A^2$
 - Gaussian pulse:
 - Energy: $E = \int_{-\infty}^{\infty} |A(t)|^2 dt$
 - Power: $P(t) \propto |A(t)|^2$

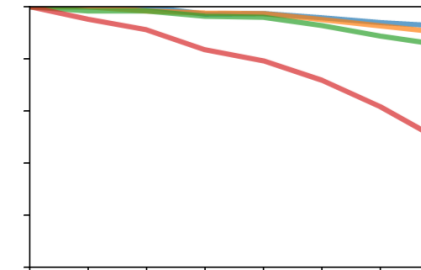


Example User Circuit (UC) Identification Evaluation

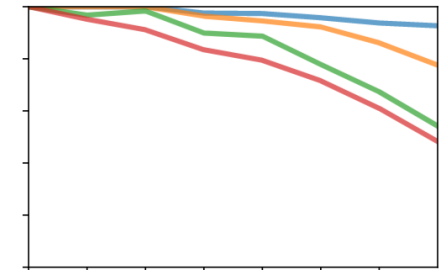
- **UC** attacks can be carried out by collecting energy, mean power or duration information, as well as by power time series collection
 - Testing done using QASMBench ver. 1.4
- Goal to identify 1 out of N benchmark circuits:



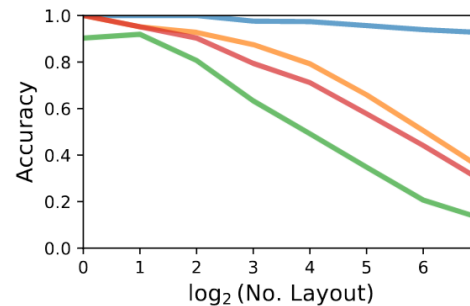
(a) Error Rate: 0.0



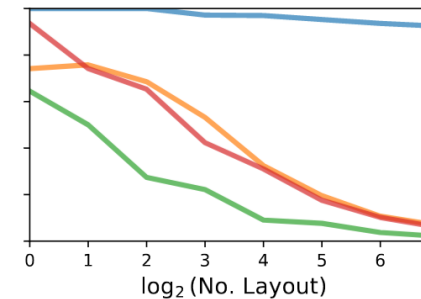
(b) Error Rate: 0.00001



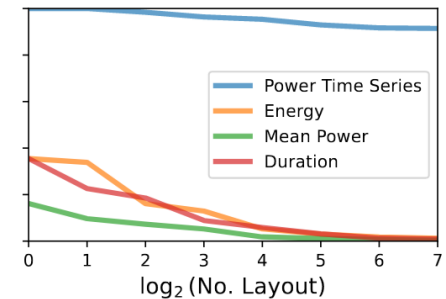
(c) Error Rate: 0.0001



(d) Error Rate: 0.001



(e) Error Rate: 0.01

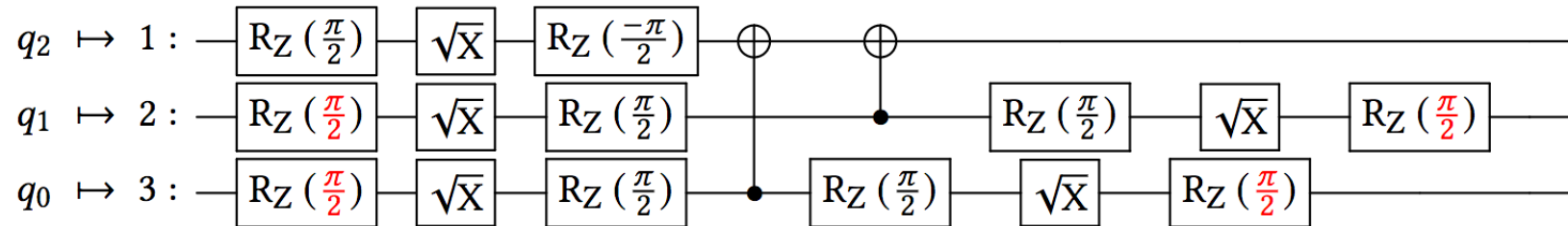


(f) Error Rate: 0.1



Example Circuit Oracle (CO) Identification

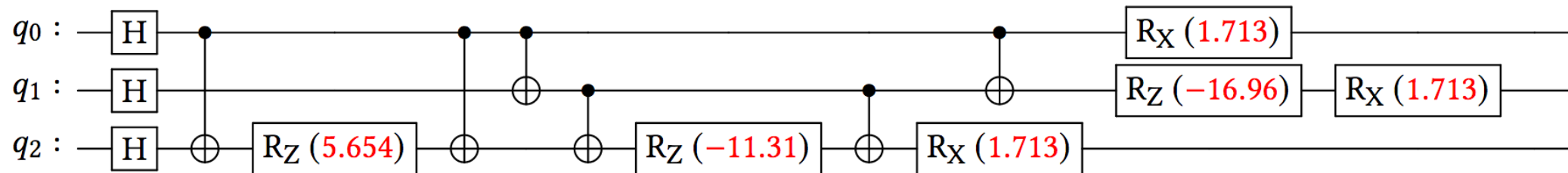
- Many quantum algorithms consist of oracles, which act like black boxes that return desired quantum states based on the input
- For BV, since the oracles are quite different from each other, the minimum circuit distance is not 0, which means the oracles **can** be distinguished from each other
- For DJ, the circuits for different oracles can be the same, and the only changes are the angles of the rotation gates, such as RZ gate, which means the oracles **cannot** be distinguished



Example Circuit Ansatz (CA) 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:



Example Qubit Mapping (QM) and Quantum Proc. (QP) Ident.

- Because of the unique per qubit control pulses, the power traces also encode the information of the physical qubits to which the quantum gates are applied to, i.e. QM
- Because of the unique per qubit control pulses, the power traces also encode the information of the quantum processor on which the circuit executes, i.e. QP
- The minimum normalized circuit distance is used to evaluate the results for QM and QP, the larger value means it is simpler to distinguish the circuits:

QASMBench Benchmark	Parameters			Attacks		CR
	Qubit	Gate	CX	QM	QP	
deutsch	2	10	1	0.025	0.116	✓
dnn	2	306	42	0.039	0.116	✓
grover	2	15	2	0.143	0.116	✓
iswap	2	14	2	0.143	0.116	✓
quantumwalks	2	38	3	0.125	0.117	✓
basis_change	3	85	10	0.673	0.068	✓
fredkin	3	31	17	0.800	0.411	✓
linearsolver	3	26	4	0.735	0.080	✓
qaoa	3	35	9	0.546	0.570	✓
teleportation	3	12	2	0.473	0.075	✓
toffoli	3	24	9	0.096	0.573	✓
wstate	3	47	21	0.789	0.101	✓
adder	4	33	16	0.727	0.201	✓
basis_trotter	4	2353	582	0.895	0.220	✓
bell	4	53	7	0.781	0.196	✓
cat_state	4	6	3	0.744	0.241	✓
hs4	4	28	4	0.545	0.327	✓
inverseqft	4	30	0	0.000	0.001	✓
qft	4	50	18	0.817	0.287	✓
qrng	4	12	0	0.000	0.001	✓
variational	4	58	16	0.792	0.239	✓
vqe	4	73	9	0.660	0.194	✓
vqe_uccsd	4	238	88	0.858	0.241	✓
error_c3	5	249	61	0.855	0.220	✓
lpn	5	17	2	0.576	0.194	✓
pea	5	126	57	0.874	0.210	✓
qec_en	5	52	16	0.746	0.250	✓
qec_sm	5	8	4	0.573	0.266	✓
qaoa	6	408	84	0.869	0.283	✓
simon	6	65	23	0.796	0.605	✓
vqe_uccsd	6	2289	1199	0.906	0.278	✓
hhl	7	1092	298	0.873	0.317	✓



Example Circuit Reconstruction (CR)

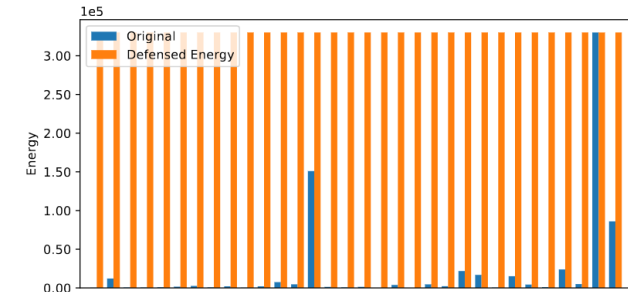
- The most powerful attacker we analyze is one who has access to per-channel power traces
- We implement an algorithm to reconstruct the circuit by identifying each gate in the pulse trace
 - For CR, the checkmark shows the non-virtual gates in the original circuit are correctly reconstructed given the per-channel power traces.

QASMBench Benchmark	Parameters			Attacks		
	Qubit	Gate	CX	QM	QP	CR
deutsch	2	10	1	0.025	0.116	✓
dnn	2	306	42	0.039	0.116	✓
grover	2	15	2	0.143	0.116	✓
iswap	2	14	2	0.143	0.116	✓
quantumwalks	2	38	3	0.125	0.117	✓
basis_change	3	85	10	0.673	0.068	✓
fredkin	3	31	17	0.800	0.411	✓
linearsolver	3	26	4	0.735	0.080	✓
qaoa	3	35	9	0.546	0.570	✓
teleportation	3	12	2	0.473	0.075	✓
toffoli	3	24	9	0.096	0.573	✓
wstate	3	47	21	0.789	0.101	✓
adder	4	33	16	0.727	0.201	✓
basis_trotter	4	2353	582	0.895	0.220	✓
bell	4	53	7	0.781	0.196	✓
cat_state	4	6	3	0.744	0.241	✓
hs4	4	28	4	0.545	0.327	✓
inverseqft	4	30	0	0.000	0.001	✓
qft	4	50	18	0.817	0.287	✓
qrng	4	12	0	0.000	0.001	✓
variational	4	58	16	0.792	0.239	✓
vqe	4	73	9	0.660	0.194	✓
vqe_uccsd	4	238	88	0.858	0.241	✓
error_c3	5	249	61	0.855	0.220	✓
lpn	5	17	2	0.576	0.194	✓
pea	5	126	57	0.874	0.210	✓
qec_en	5	52	16	0.746	0.250	✓
qec_sm	5	8	4	0.573	0.266	✓
qaoa	6	408	84	0.869	0.283	✓
simon	6	65	23	0.796	0.605	✓
vqe_uccsd	6	2289	1199	0.906	0.278	✓
hhl	7	1092	298	0.873	0.317	✓

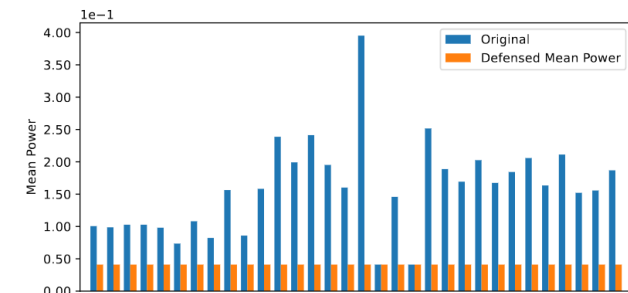


Software Protections

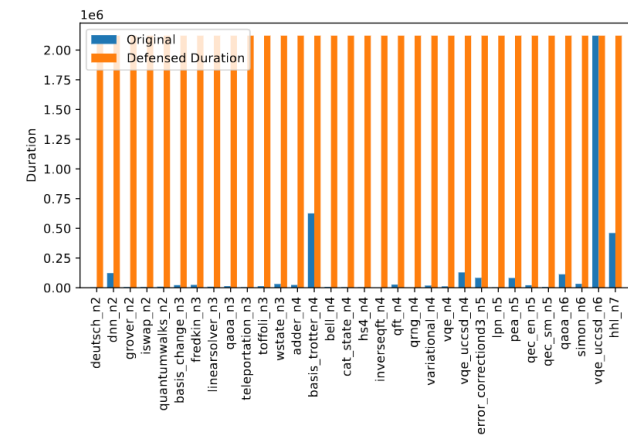
- **Software defense** to protect from these attacks, the insight is to add additional gates to the circuit so that the metric values of all circuits in the list can be made similar to each other:
 - a) **Add pairs of X gates** (defend energy measure attack)
 - find the largest energy in the circuit list and then add two X gates units on different qubits to reach the largest energy
 - b) **Add delay (ID) gates** (defend mean power measure attack)
 - find the circuit with smallest mean power among the set of circuits, and then add delays to the other circuits so each can reach the smallest mean power
 - c) **Add delay (ID) gates** (defend timing attack)
 - find the largest duration in the circuit list, and then add delay gates for all other circuits to make the duration the same as the largest duration



(a) Energy of the benchmark.



(b) Mean power of the benchmark.

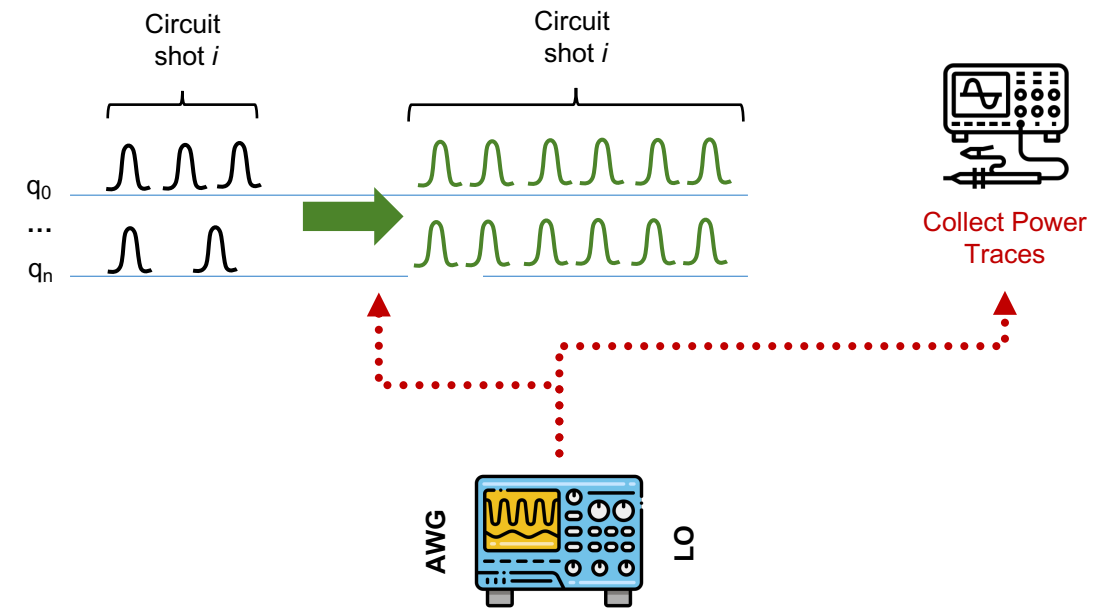


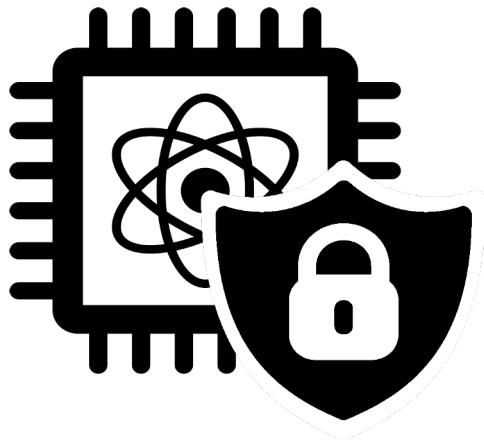
(c) Duration of the benchmark.



Software Protections

- **Software defense:** may be possible to leverage virtual RZ gates to confuse attackers
 - RZ gates are virtual, they effectively shift the phase of the RF pulses, but do not themselves generate RF pulses
 - Assume this phase shift is not detected through power side channel, compared to power consumption generated by the non-virtual pulses
- Any SX gate can be made into two SX:
 - $SX = RZ \cdot SX \cdot RZ \cdot SX \cdot RZ$
- Any X gates can be made into two SX:
 - $X = SX \cdot SX$





Side Channels in FTQC



Comparing NISQ and FTQC Threats

- **NISQ** (Noisy Intermediate-Scale Quantum), approx. now–2030, features 50–1000s of noisy qubits
- **FTQC** (Fault-Tolerant Quantum Computing), approx. from 2030, features 1000s and more of physical qubits to create logical qubits, 1000:1 to 100:1 ratio

NISQ Security Threats:

- Noise-related, inject noise or monitor noise
- Imperfect operations
- Low-level pulse access abuse

FTQC Security Threats:

- Flaws in error correction, correlated errors
- Error correction decoding attacks
- Code-switching, downgrade codes or monitor code changes

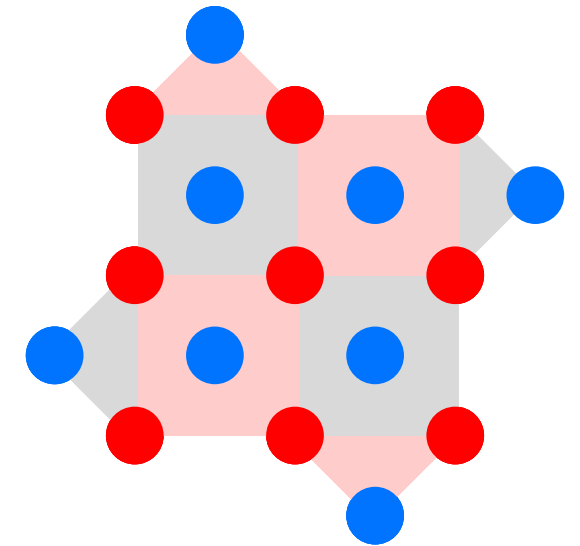
Both:

- Monitoring execution, **can't clone qubits, but can clone instructions**
- Controller logic and hardware vulnerabilities



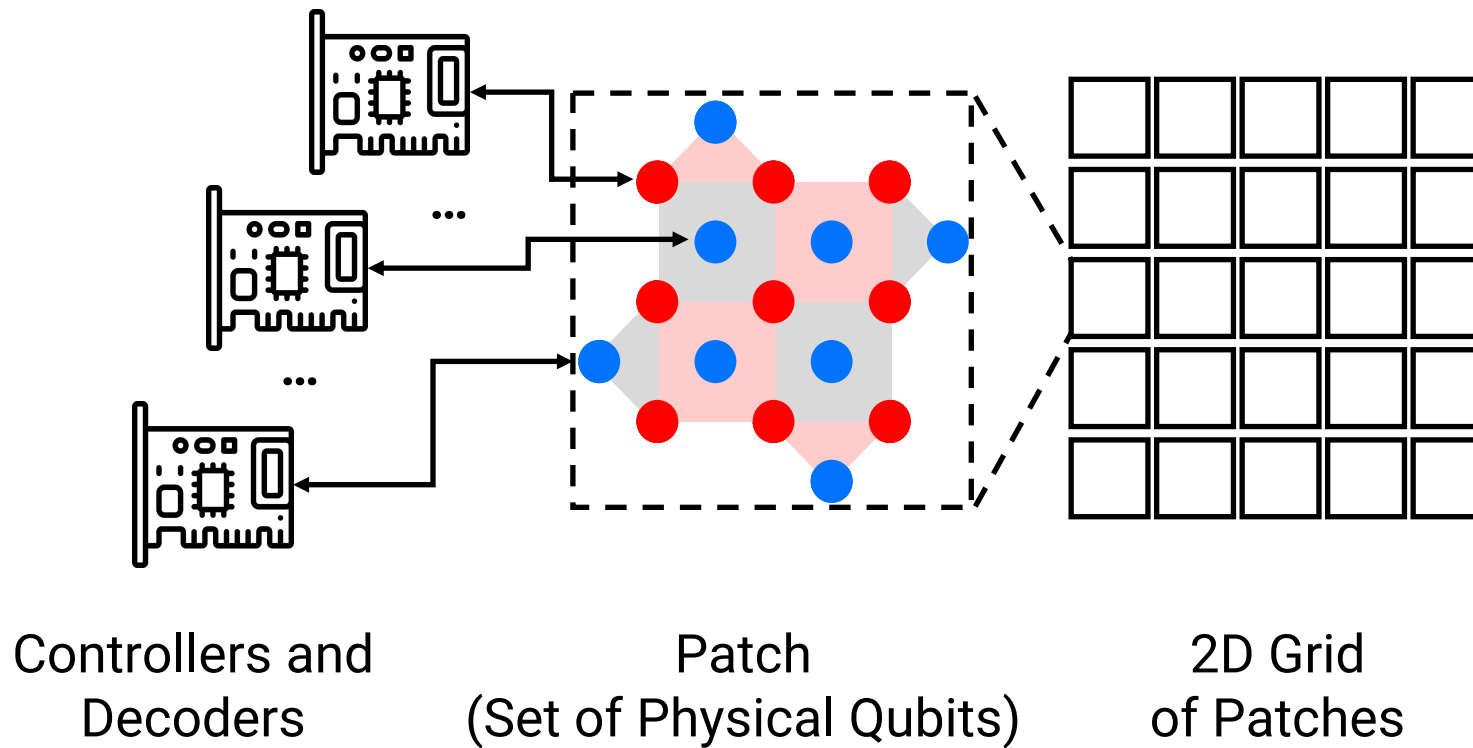
Fault-Tolerant Quantum Computing

- Emergent Fault-Tolerant Quantum Computing (FTQC) aims to mitigate noise and errors in quantum computers through error correction
- FTQC produces logical qubits with reduced error rates by using additional physical qubits to enable redundancies to faults
- Many error correction techniques exist, one very promising is surface code, that uses data and stabilizer qubits:
 - Data qubits: hold the actual quantum information (the logical quantum states)
 - Ancilla (or stabilizer) qubits: used to measure error syndromes without disturbing the data qubits

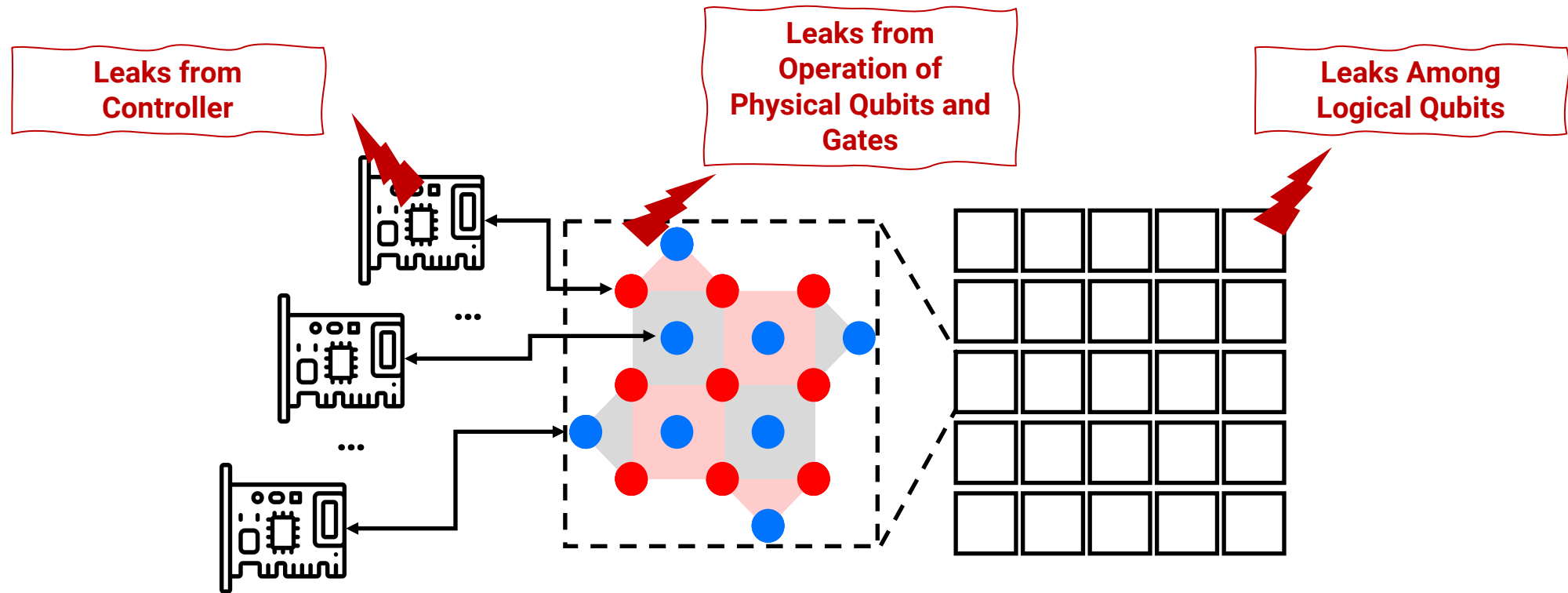


FTQC Control Infrastructure

- Controllers, containing Quantum Error Correction decoders and other logic, interface to the physical qubits, each group of physical qubits makes up a patch or a logical qubit

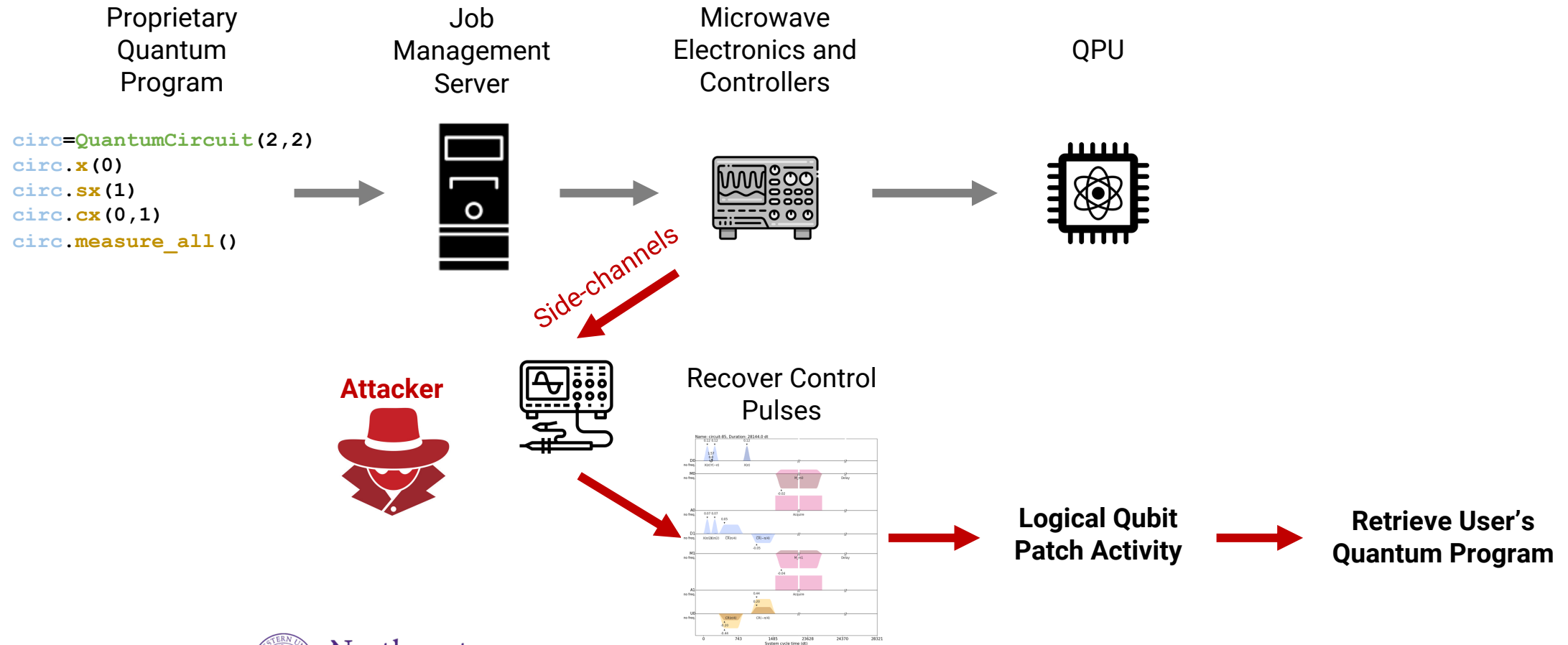


Side Channel Threats In FTQC



Power Side Channels through Quantum Controller (FTQC)

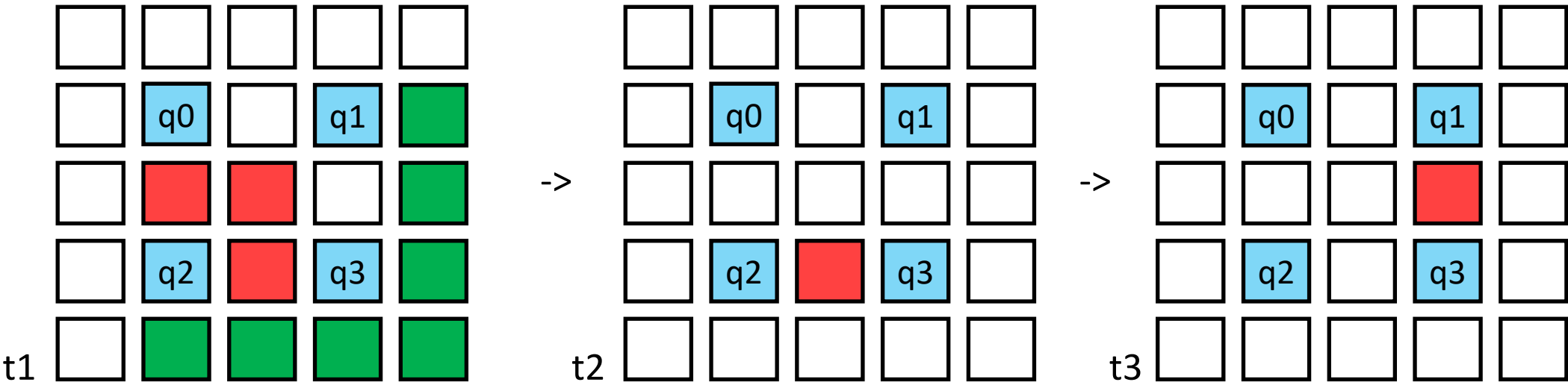
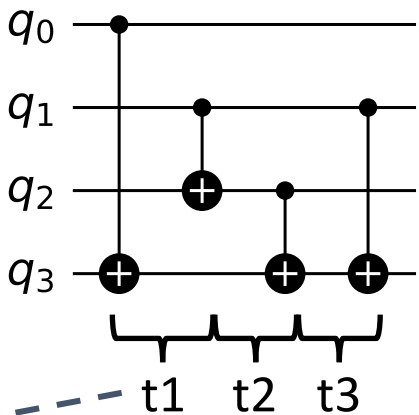
- Collecting side-channel information from the controller electronics:



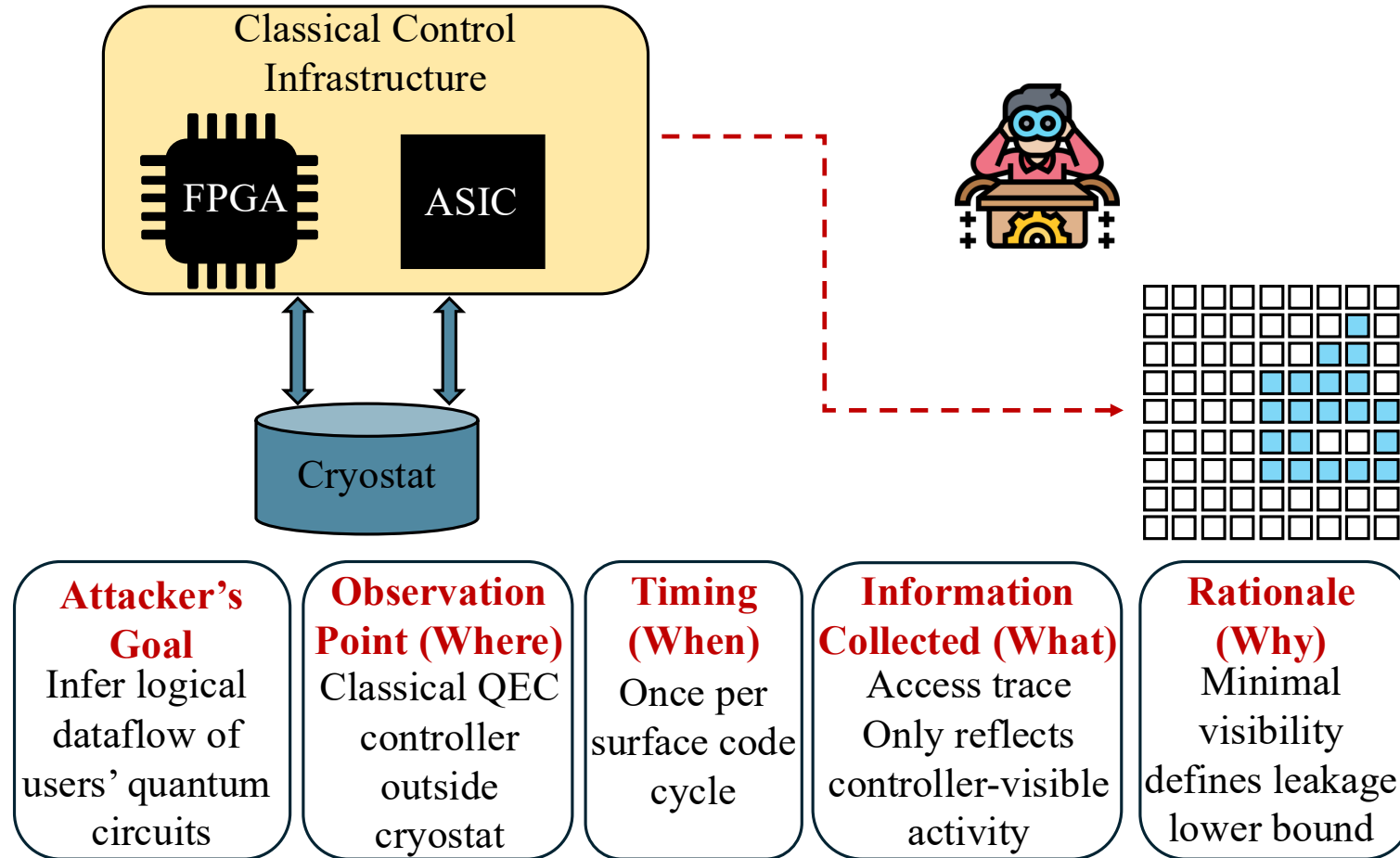
Tracing Attacks on FTQC and Lattice Surgery

Insight: Tracing merging and splitting operations can reveal two qubit operations

Example: activity on quantum chip corresponding to logical circuit



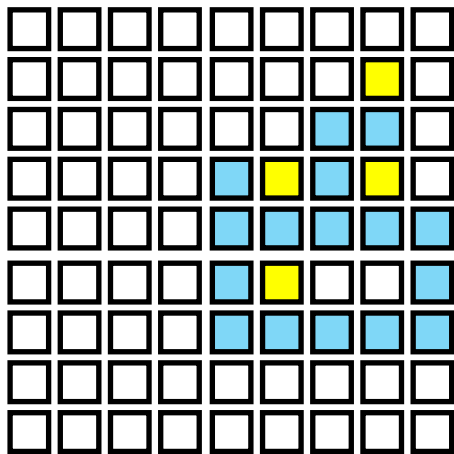
Attack Model



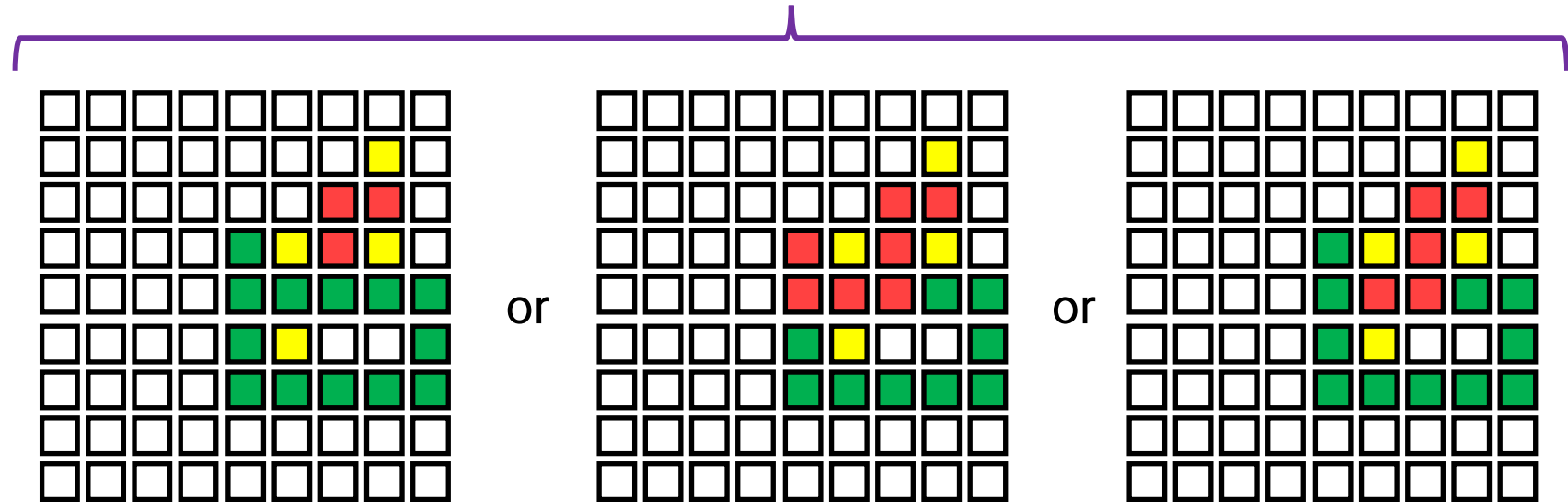
Tracing FTQC Chip Activity

Challenge: There can be ambiguity in the traces

Activity observation



Possible traces:



FTQC Tracing Attack Process and Results

Observation: Tracing FTQC activity can reveal connections between qubits as they change over time

Attacks process:

1. Each merge and split is a two-qubit operation – save observations from traces
2. Build Directed Acyclic Graph (DAG) of the circuit from the traces – DAG where vertices represent gates and directed edges represent qubit dependencies, have excess vertices due to ambiguities in the traces
3. Use a databases of common quantum functions saved as DAGs
4. Database DAGs can be used as “signatures”, use subgraph matching to see which known functions match the DAG that was built

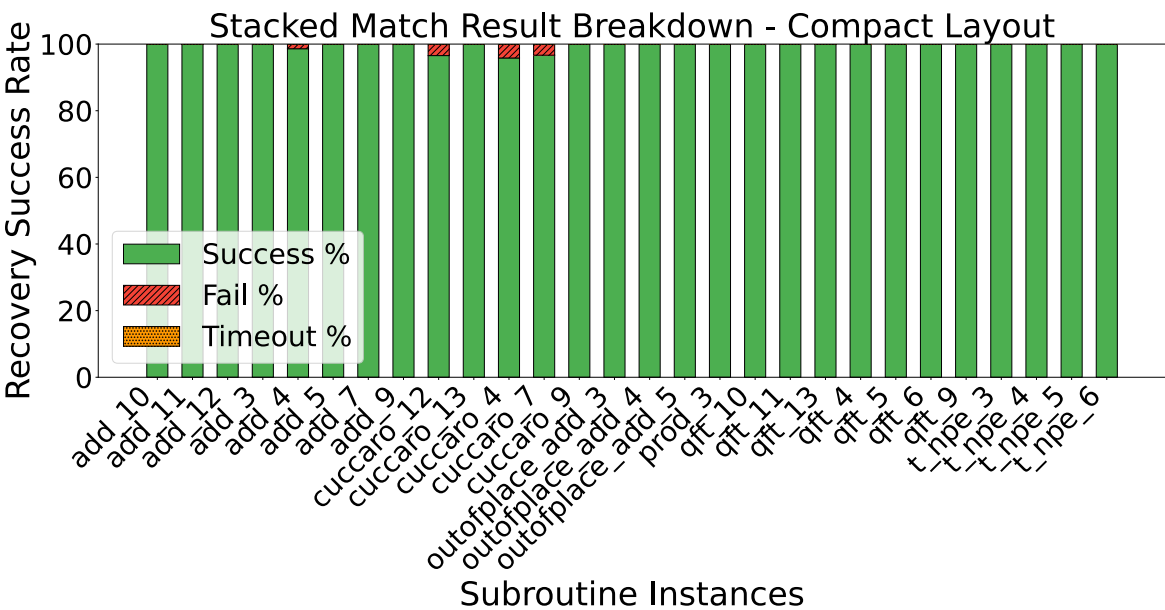


FTQC Tracing Attack Process and Results

Synthetic Benchmarks

Synthetic	Instances	# Qubits	Avg. DAG Nodes		
			Compact	Sparse	Interm.
random_mix_1	qft_5 + t_npe_4 + add_3	29	207	376	206
random_mix_2	qft_6 + t_npe_3 + add_4	26	153	205	151
random_mix_3	qft_4 + t_npe_4 + add_5	34	225	395	236
random_mix_4	qft_11 + t_npe_5 + add_7	56	439	1106	446
random_mix_5	t_npe_3 + add_9	35	173	202	173
random_mix_6	qft_13 + add_10	42	265	269	263
random_mix_7	qft_13 + t_npe_5	38	413	985	423
random_mix_8	add_11 + t_npe_6	68	496	1492	519
random_mix_9	add_3 + qft_4	12	36	36	34
random_mix_10	add_4 + qft_4	15	48	48	47
random_mix_11	prod_3 + outofplace_add_3 + qft_4	32	173	205	173
random_mix_12	prod_3 + cuccaro_4 + qft_4	32	152	176	152
random_mix_13	t_npe_3 + cuccaro_7	25	112	204	113
random_mix_14	outofplace_add_5 + qft_10 + add_5	40	199	297	195
random_mix_15	add_4 + outofplace_add_4 + cuccaro_4	34	105	230	102
random_mix_16	t_npe_3 + qft_5 + cuccaro_9	34	142	311	147
random_mix_17	cuccaro_12 + qft_4 + t_npe_3	39	152	424	157
random_mix_18	add_12 + t_npe_3 + qft_6	50	238	331	243
random_mix_19	t_npe_4 + add_12 + qft_5	56	307	469	312
random_mix_20	qft_9 + cuccaro_13 + adder_12	72	270	482	270

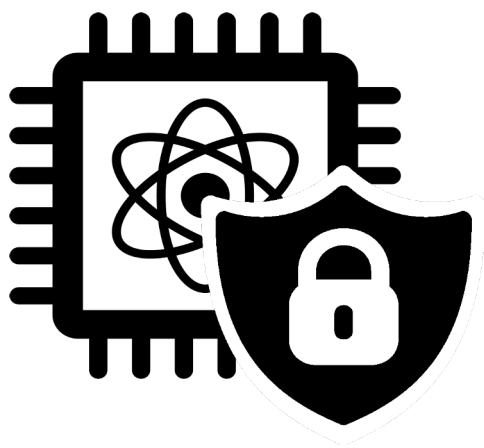
Sample Results



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**
 - Side Channels in NISQ Controllers
 - Side Channels in FTQC
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
- **4:00pm: End**



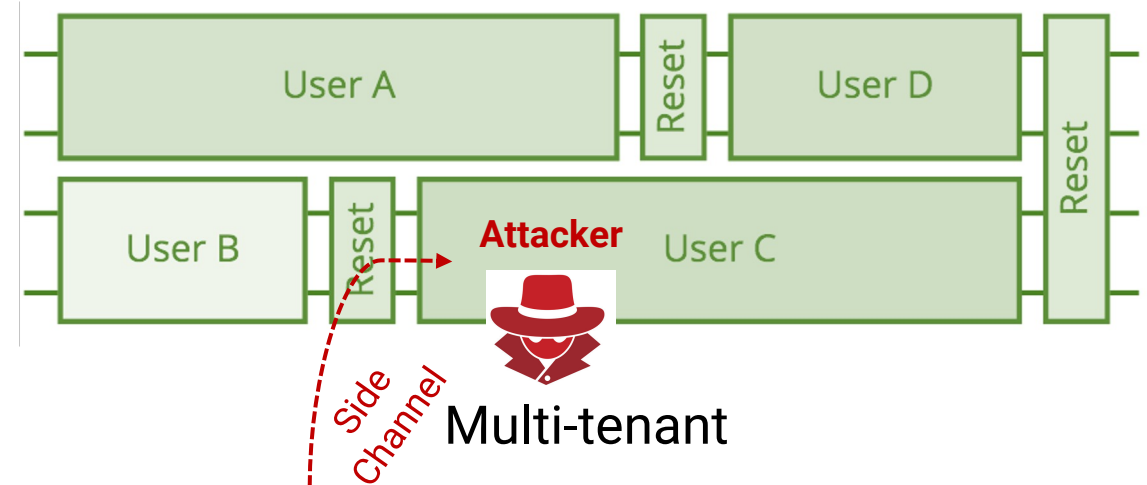
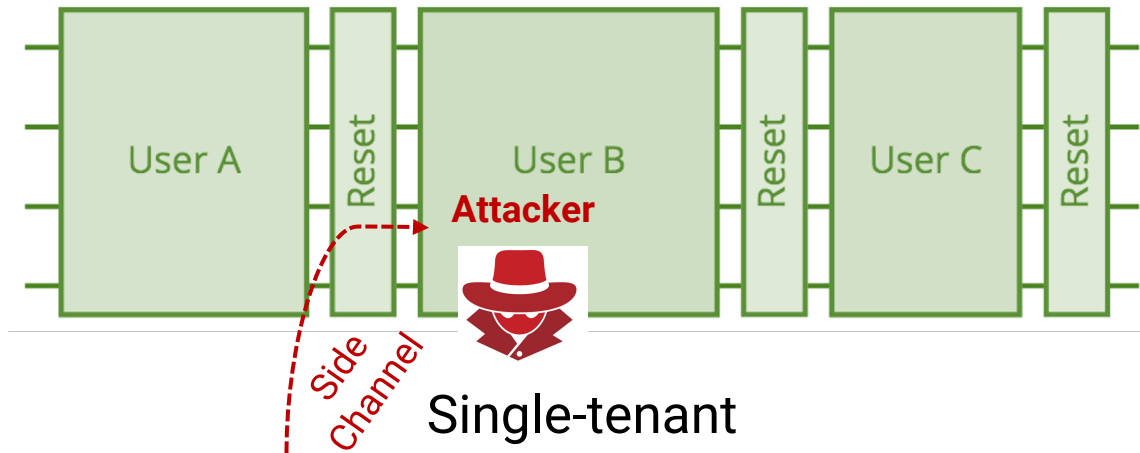


Side Channels Across Reset Gates in NISQ



Affecting Qubit Outputs When Reset Gates are Imperfect

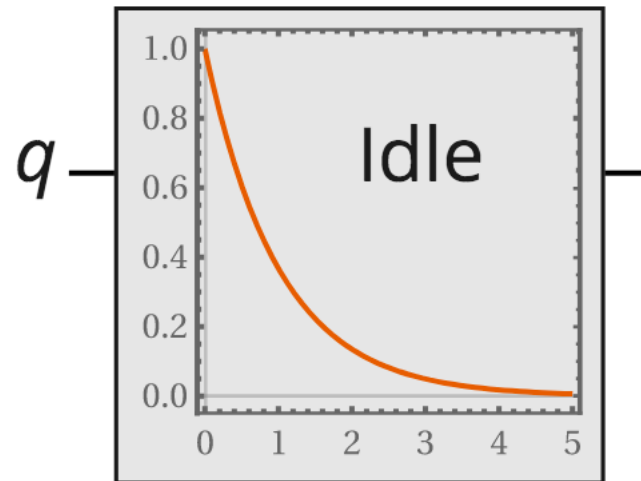
- It is possible to share the quantum computing hardware in single- or multi-tenant ways:



Example Means to Reset Qubits in Superconducting Computers

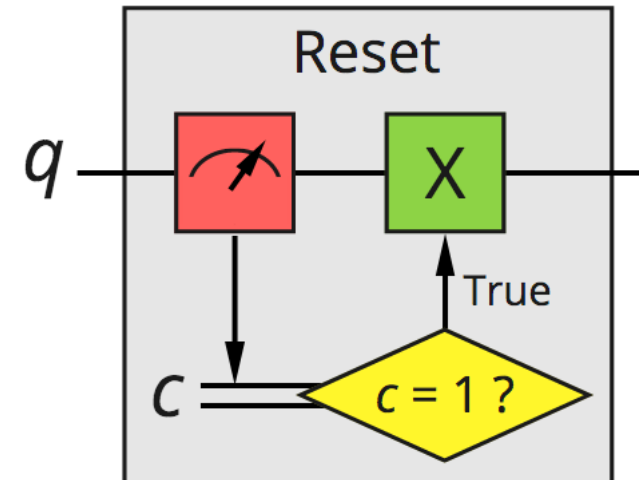
Thermalization ($250\mu\text{s}$ - 1ms)

- Good: Low state retention
- Bad: Poor scaling with T_1



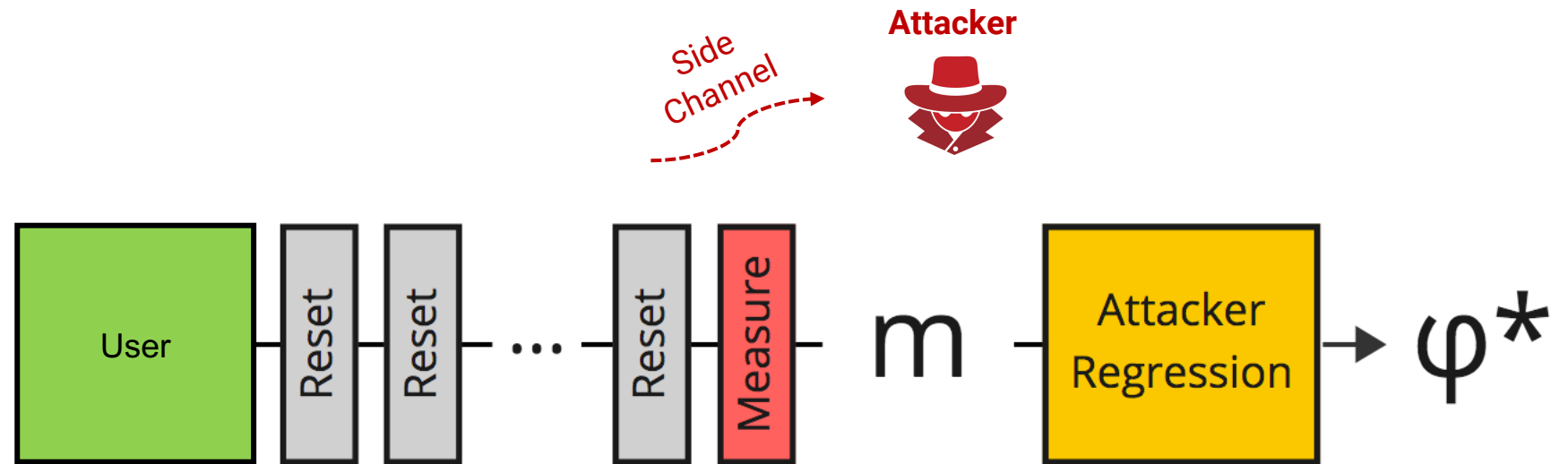
Active reset ($1\mu\text{s}$)

- Good: Fast and scales well
- Bad: Detectable state leakage



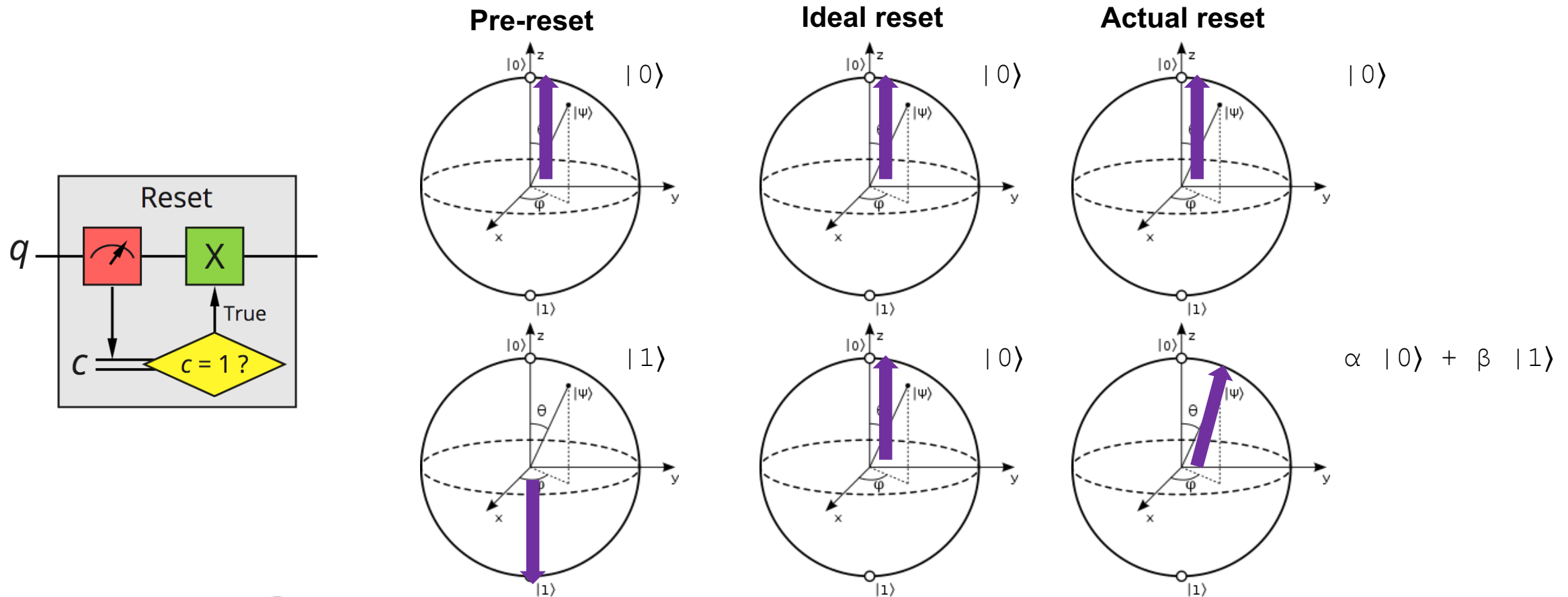
Recovering Information Across Reset Operation

- Attacker scheduled after victim, shot by shot, on the same qubit
- Aim to correlate qubit state $|\Psi\rangle$ with post-reset measurement m
- Build model that infers $|\Psi\rangle$ given m and number of resets r



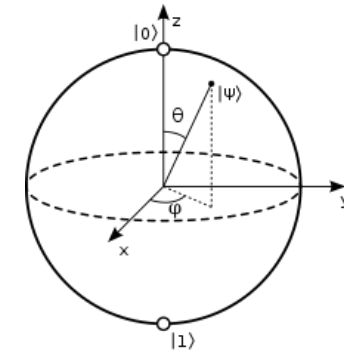
Observed Behavior of Reset Gates

- Observed reset gate behavior on superconducting qubit quantum computers:



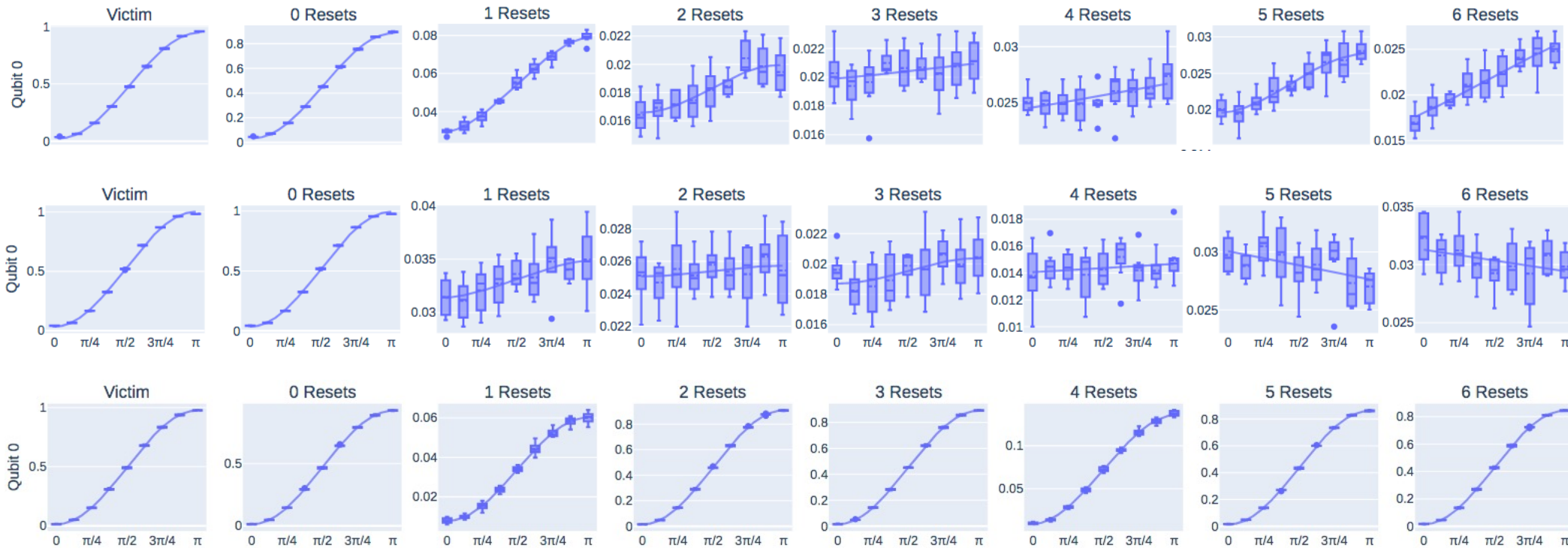
Measurements of Reset Behavior

- Experimentation demonstrated that after even multiple resets, the qubit state is not ideal $|0\rangle$
- Probability of measuring $|1\rangle$ post reset for different θ angles:



$$\alpha = \cos \frac{\theta}{2}$$

$$\beta = e^{i\varphi} \sin \frac{\theta}{2}$$



IBM Jakarta
backend

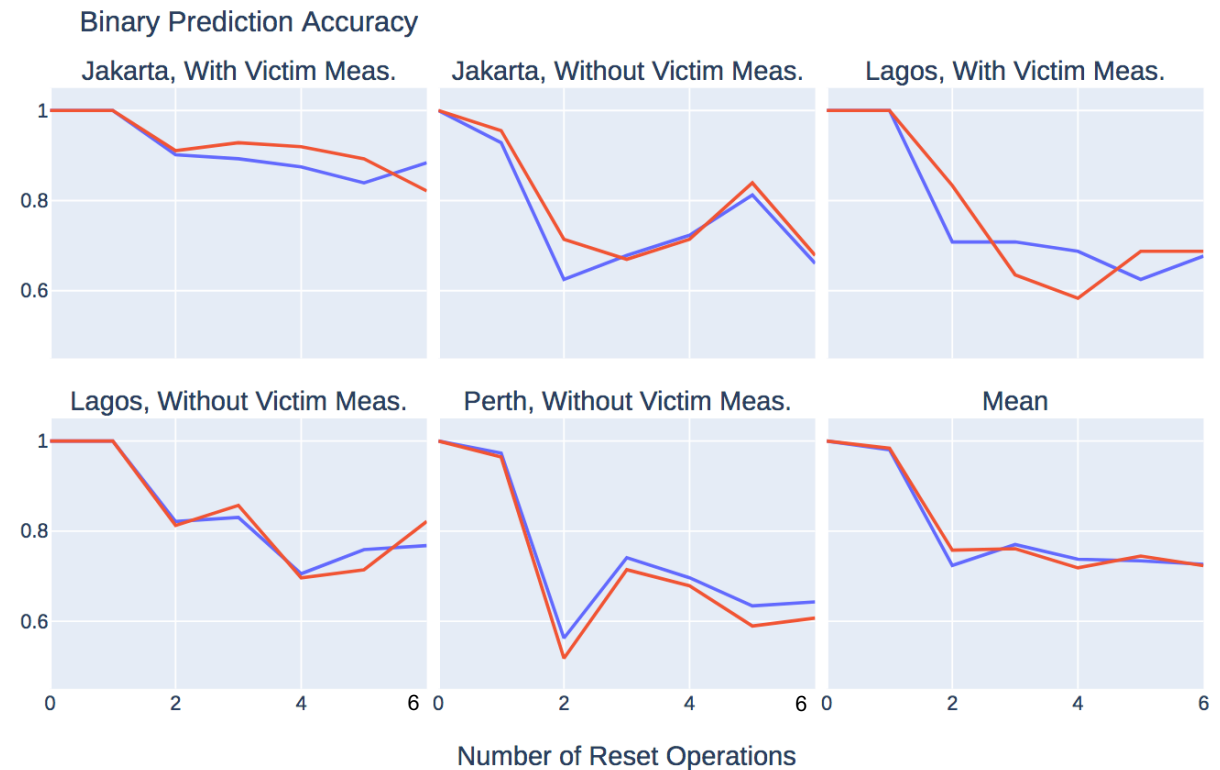
IBM Lagos
backend

IBM Perth
backend



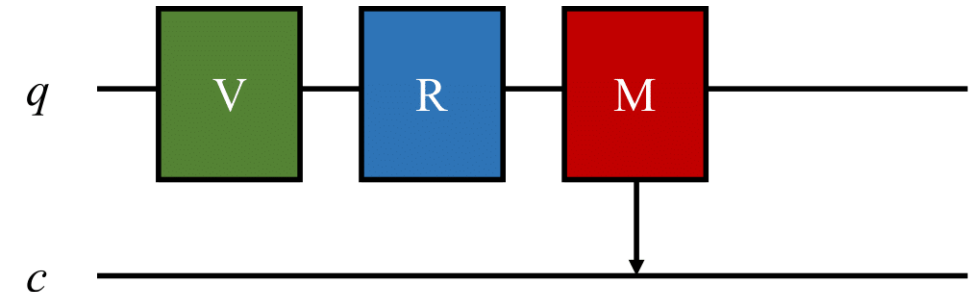
Automated Qubit State Prediction

- Build model that infers $|\Psi\rangle$ given measurement m and number of resets r which have occurred
- Prediction accuracy is restricted to testing data of $\theta \in \{0, \pi\}$, i.e. qubit states that approximate $|0\rangle$ or $|1\rangle$
- We acquire reconstruction θ^* from the channel and compare proximity of θ^* to 0 and π , and choose the reconstructed qubit state to be $|0\rangle$ or $|1\rangle$ correspondingly

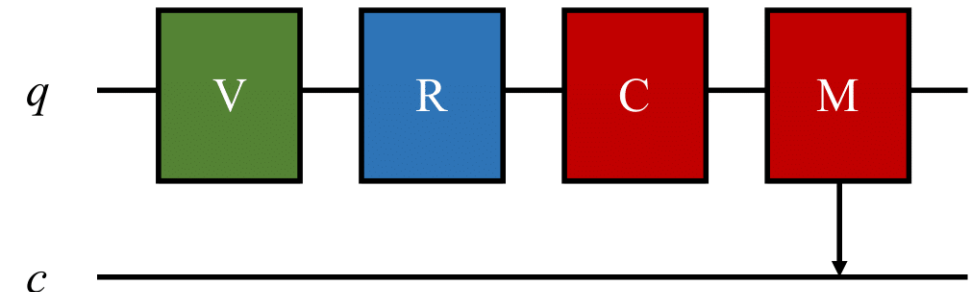


Stealthier Reset Gate Attacks

- Reset gate attack using simple measurement can be easily detected: scan for user circuits that begin with a measurement operation
- Attackers can develop stealthier reset gate attacks to evade defenses
 - Need to explore and evaluate different strategies attackers could take, then develop defenses
- Attackers can add a masking circuit (C) before measurement to make their circuit harder to classify as a malicious circuit



Basic Attack

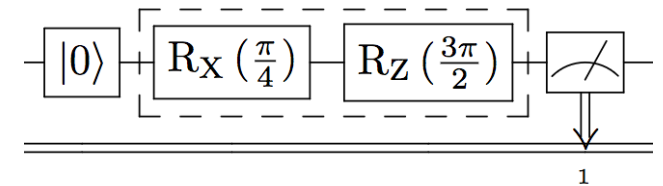
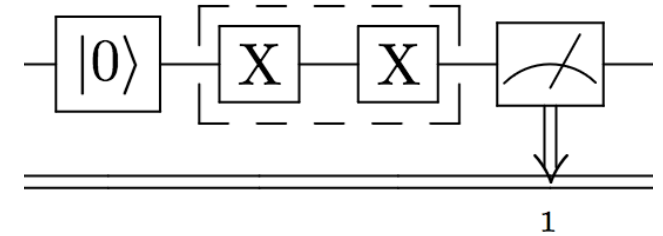


Stealthier Attack



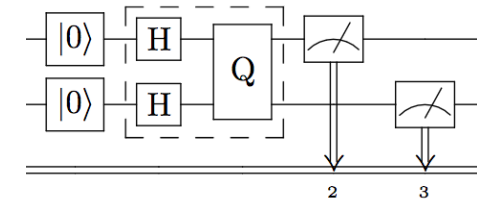
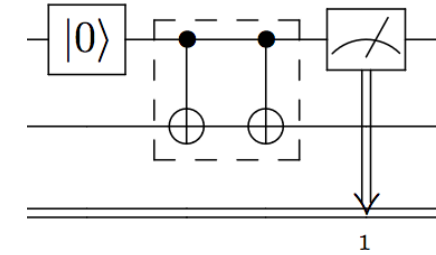
Examples of Simple Masking Circuits

- **Identity Circuits** – Since effectively there is no rotation, the attacker's measurement should return the same values as it would be right after the reset operation
 - For example, circuits consisting of an even number of single-qubit **X** gates on each qubit, such that the total effective angle of rotation θ is 0
- **RX and RZ Gate Circuits** – Because the rotation angle is known, the attacker can infer the qubit 1-output probabilities as they would be right after the reset gate
 - For example, circuits consisting of single-qubit gates with effective θ (**RX** gate) rotation and φ (**RZ** gate) rotation



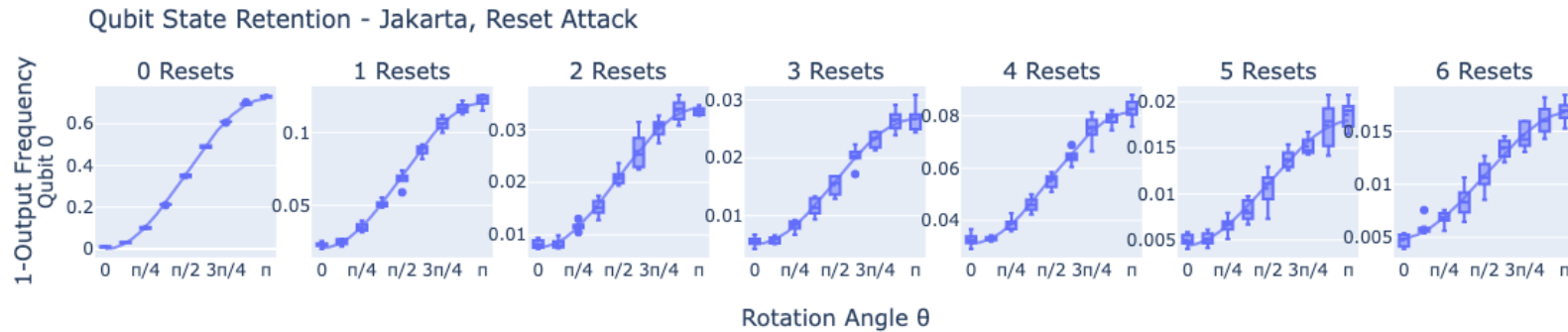
Examples of Simple Masking Circuits

- **cx Gate Circuits** – The control qubits of **cx** gate experience delay (due to duration **cx** gate) but otherwise can be leveraged by an attacker since they do not experience any rotations
 - For example, circuits consisting of multiple **cx** gates
- **QASM Benchmark Circuits** – Realistic circuits would be ideal masking circuits, but will make inferring the victim's state most difficult if not impossible
 - For example, circuits from the QASM benchmark suite

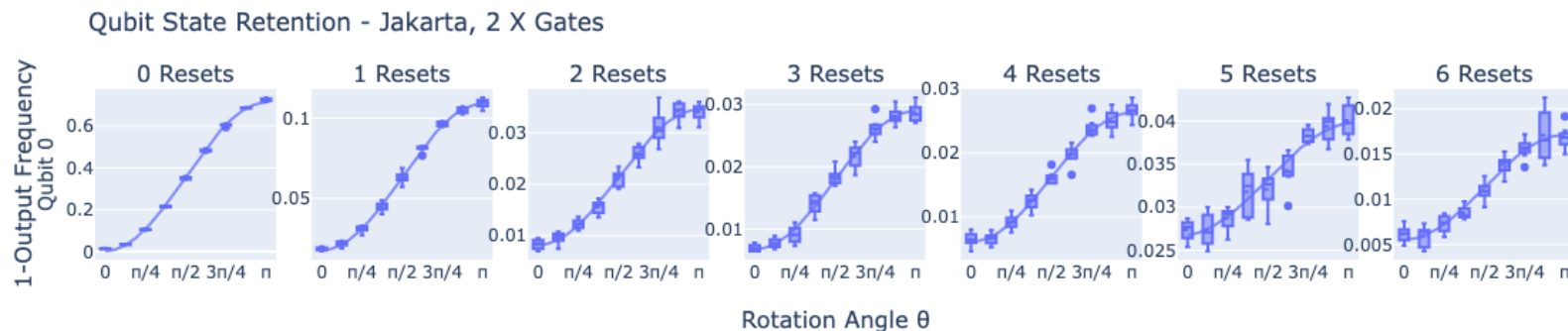


Recovering Victim's State when Masking Circuit is Used

- Comparison to baseline when no masking circuit is used:



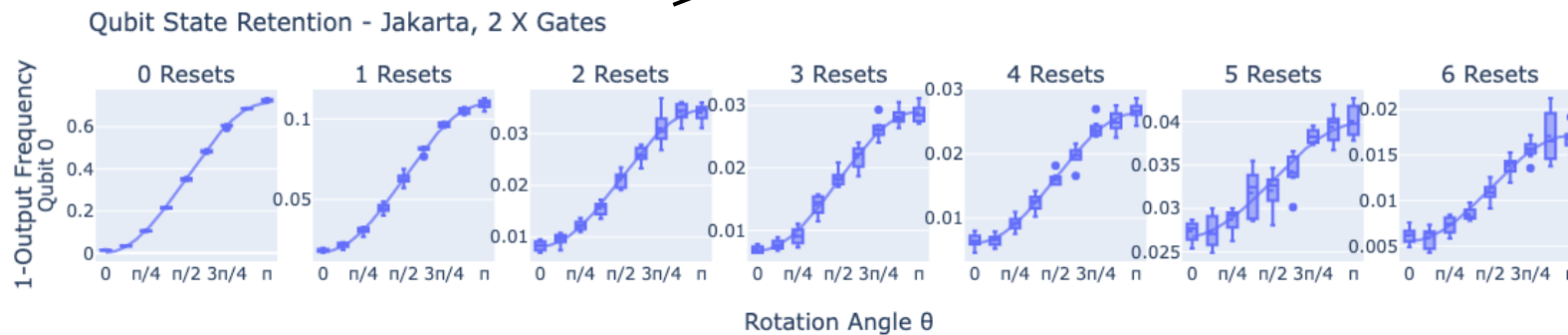
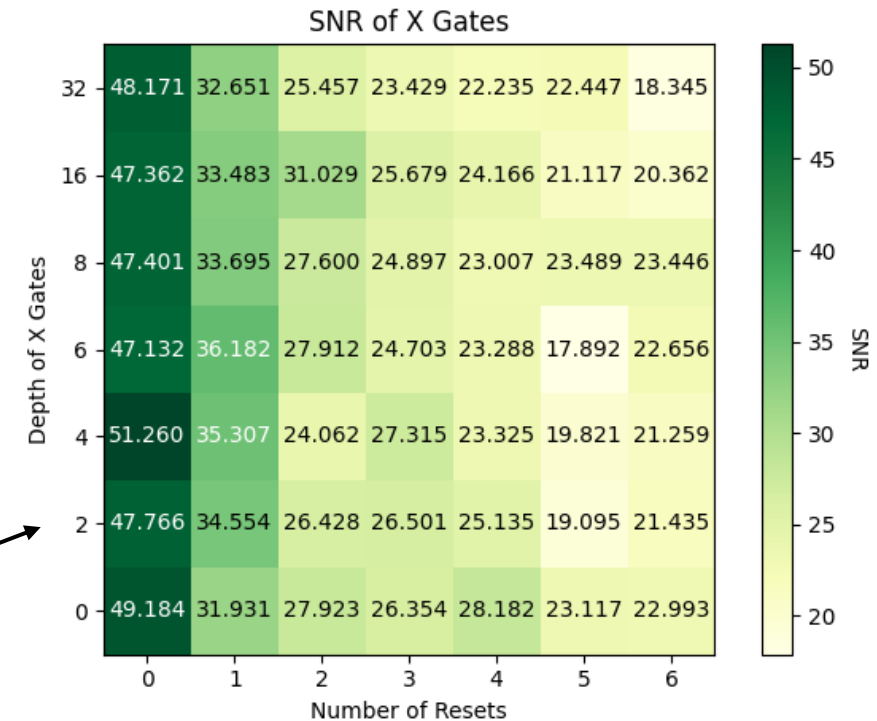
- Probability of measuring $|1\rangle$ post reset for different θ angles when **identity circuit** is used:



Analysis of Masking Circuit Effectiveness with SNR

- The signal-to-noise (SNR) ratio is defined as $\text{SNR} = a/\sigma$
 - a is the error channel characterization parameter, which represents the amplitude of the sigmoid fit
 - compute the standard deviation in 1-output frequency for each fixed θ as ϕ varies and then compute the average standard deviations over all input θ values which gives σ

- Identity circuit example:**



Side Channels in Reset Gates

Many attacks have been demonstrated:

- Side channel attack on reset gates (2022)

Various defenses have been proposed in parallel:

- Randomize circuit outputs with Quantum one-time pad

Future directions in this research:

- Continued investigation of qubit reset mechanisms

