PNW PLSE

# Programming Abstractions for Quantum Computing

Jennifer Paykin

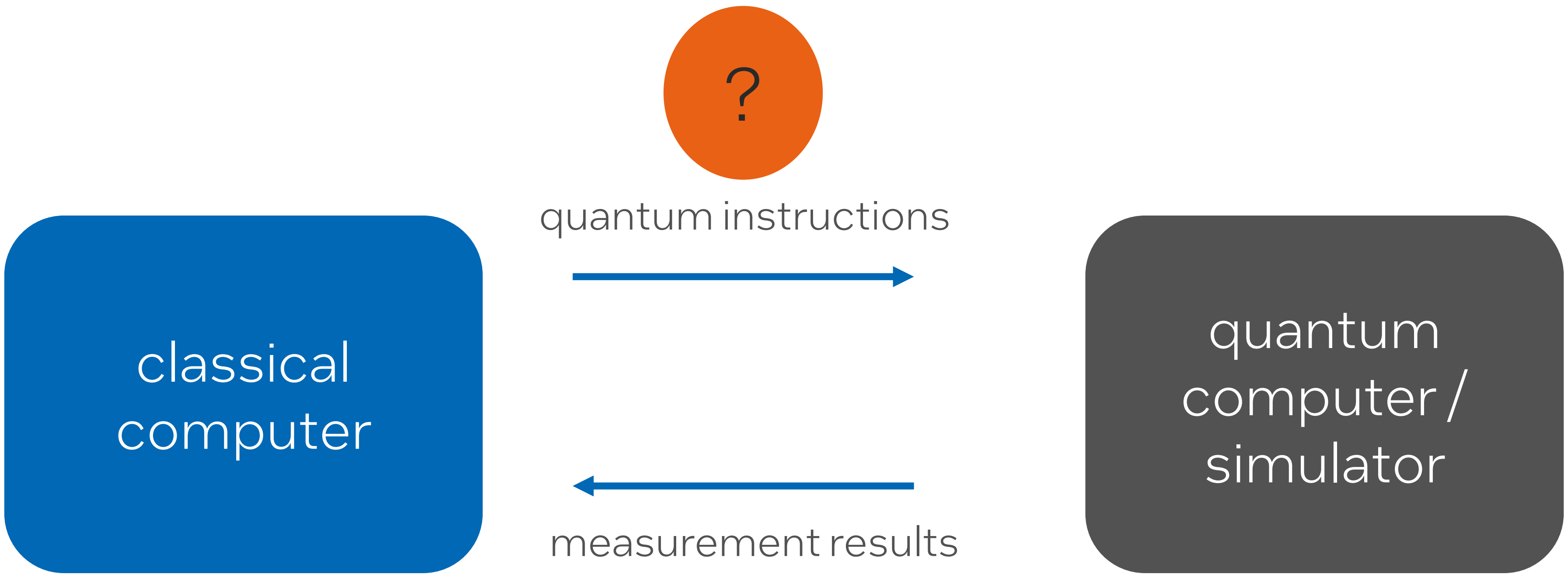
jennifer.paykin@intel.com

Research Scientist, Intel Labs

May 9, 2023

intel.

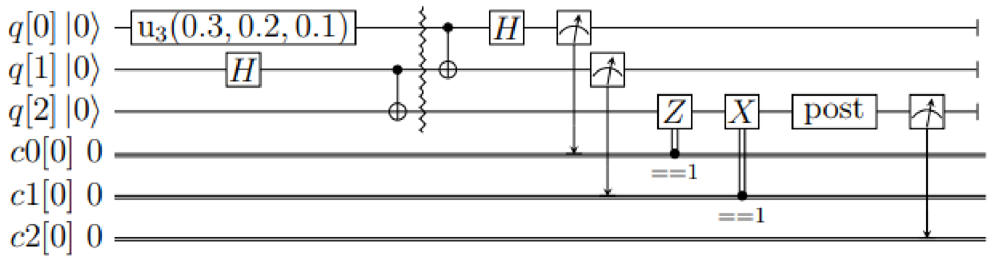
# Quantum Computing as a Co-processor



# Quantum Circuits

circuit description language (e.g. python)

```
// quantum teleportation example
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
creg c0[1];
creg c1[1];
creg c2[1];
// optional post-rotation for state tomography
gate post q { }
u3(0.3,0.2,0.1) q[0];
h q[1];
cx q[1],q[2];
barrier q;
cx q[0],q[1];
h q[0];
measure q[0] -> c0[0];
measure q[1] -> c1[0];
if(c0==1) z q[2];
if(c1==1) x q[2];
post q[2];
measure q[2] -> c2[0];
```




quantum instructions

classical computer

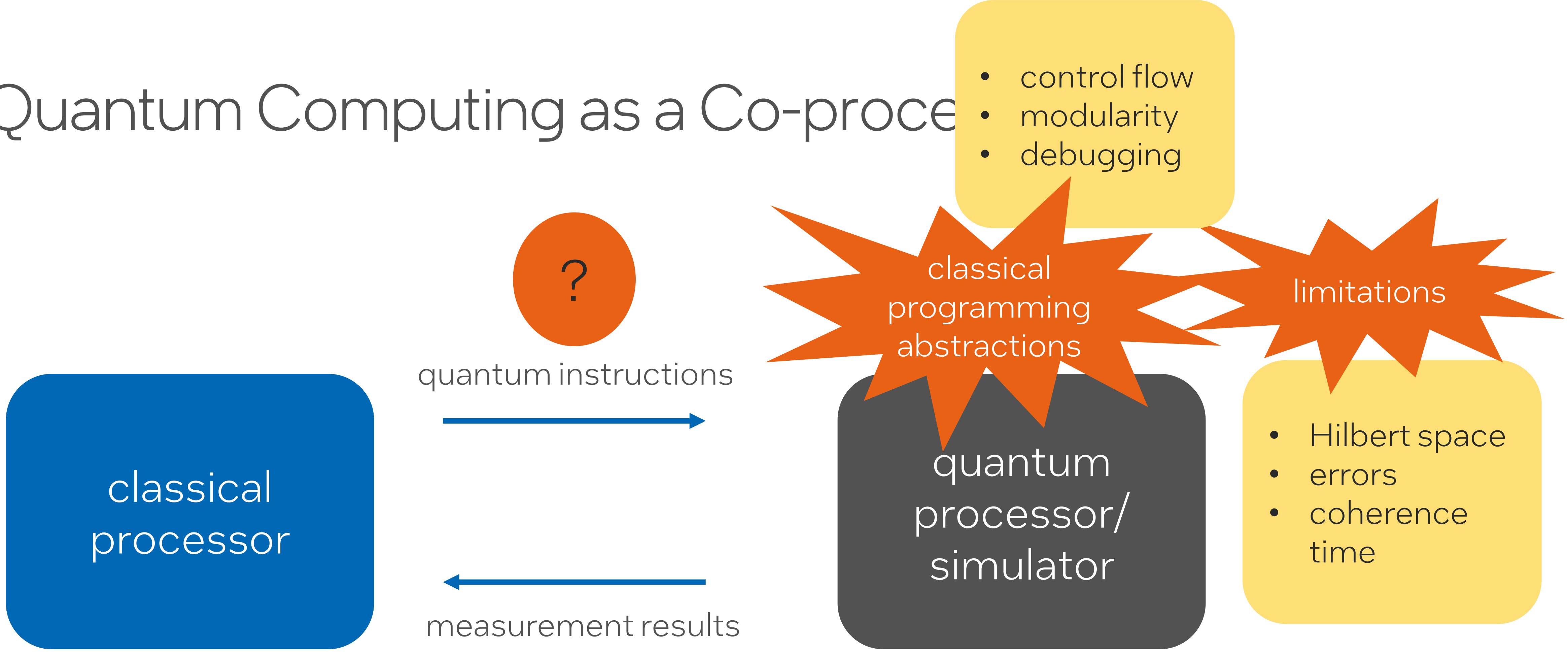
quantum computer

measurement results



Cross, Andrew W., et al. "Open quantum assembly language." *arXiv preprint arXiv:1707.03429* (2017).

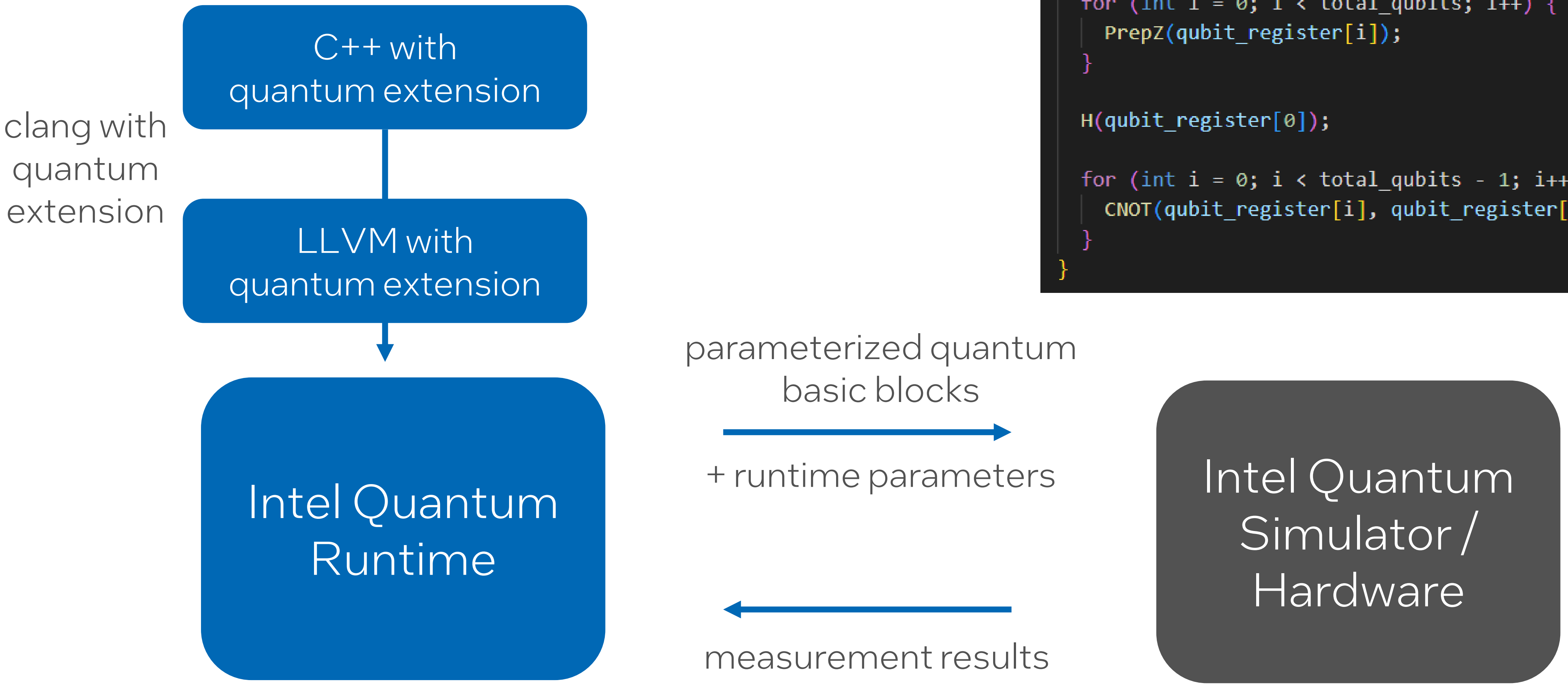
# Quantum Computing as a Co-proce



# Intel Quantum SDK

developer.intel.com/quantumsdk

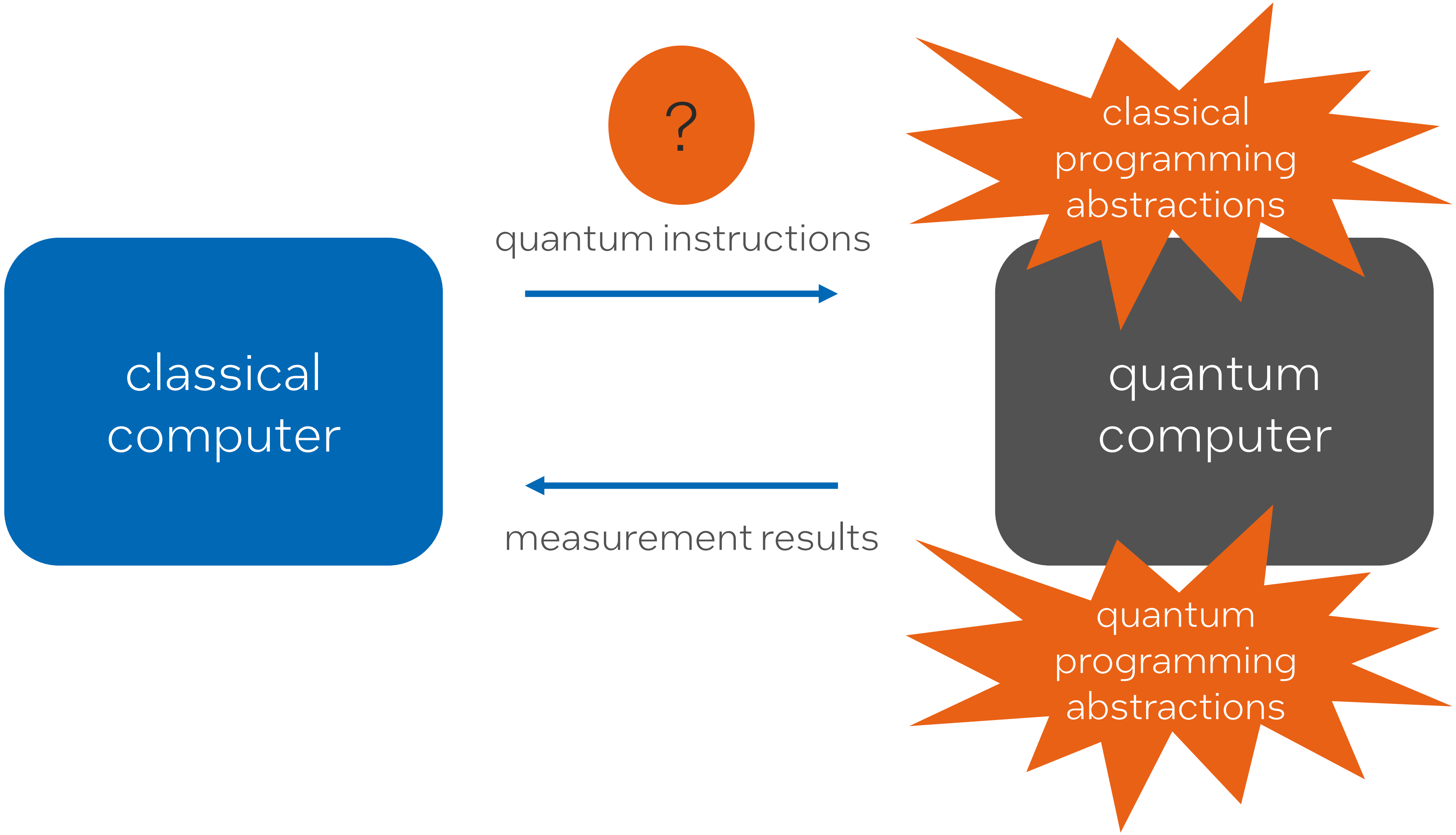
```
quantum_kernel void ghz_total_qubits() {
  for (int i = 0; i < total_qubits; i++) {
    PrepZ(qubit_register[i]);
  }

  H(qubit_register[0]);

  for (int i = 0; i < total_qubits - 1; i++) {
    CNOT(qubit_register[i], qubit_register[i + 1]);
  }
}
```



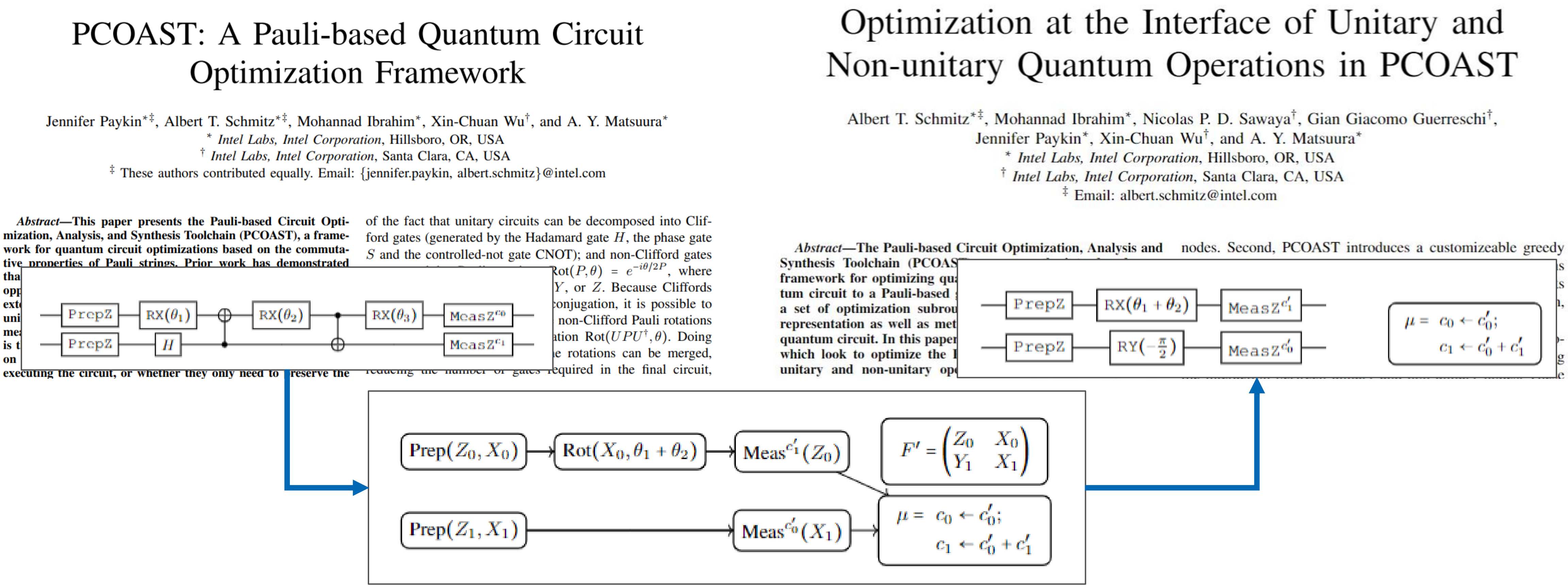



Khalate, Pradnya, et al. "An LLVM-based C++ Compiler Toolchain for Variational Hybrid Quantum-Classical Algorithms and Quantum Accelerators." *arXiv preprint arXiv:2202.11142* (2022).

# Quantum Computing as a Co-processor



# PCOAST: A Pauli-based IR




In submission

# PCOAST: A Pauli-based IR

## Hamiltonian Simulation

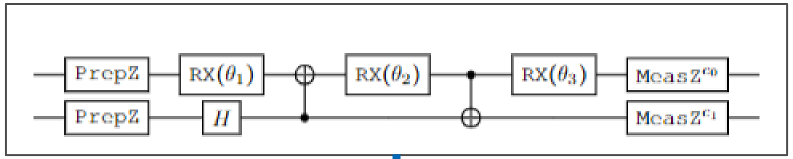
$$H = \theta_0 Z_0 Z_1 + \theta_{12} Z_1 Z_2 + \theta_{23} Z_2 Z_3 + \theta_{03} Z_0 Z_3 + \theta_{all} Z_0 Z_1 Z_2 Z_3$$

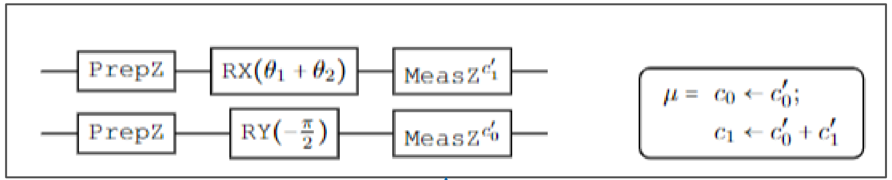
Schmitz, Albert T., et al. "Graph optimization perspective for low-depth Trotter-Suzuki decomposition." *arXiv preprint arXiv:2103.08602* (2021).

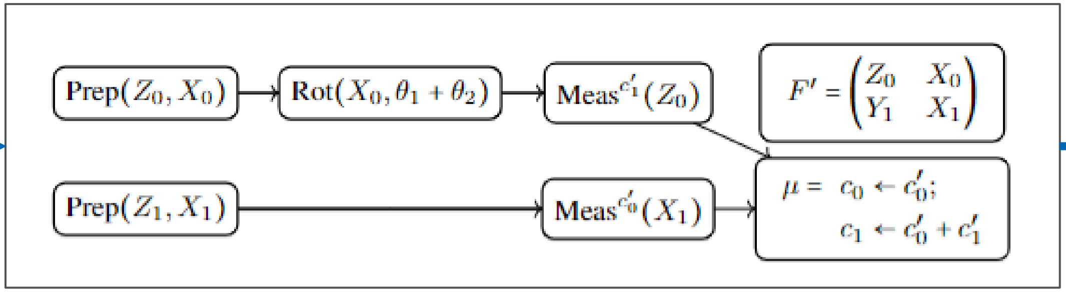
## Quantum Kernel Expressions

```
QExpr multiCtrlU(QExpr U, qbit* qs, cbit* xs) {

    return control(qs, "|111>", U))
          + map(MeasZ, qs, xs);
}
```

Matsuura, Anne, Albert Schmitz, and Jennifer Paykin. "A Functional Approach to the Modular Construction of Quantum Logic." *Bulletin of the American Physical Society* (2023).




In submission

PNW PLSE

# Programming Abstractions for Quantum Computing

Jennifer Paykin

jennifer.paykin@intel.com

Research Scientist, Intel Labs

May 9, 2023

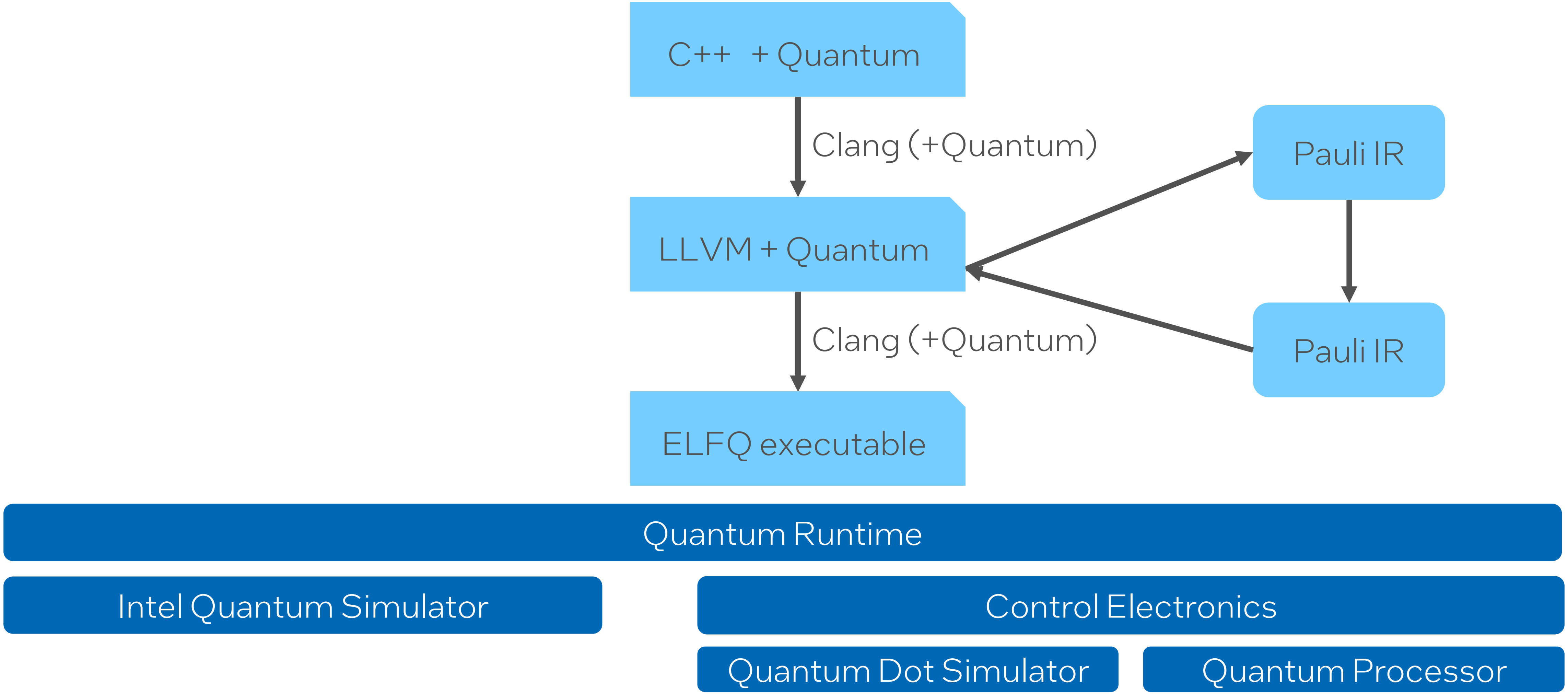
Intel Quantum SDK 1.0
developer.intel.com/quantumsdk

intel®

# Quantum Computing 101

- Qubits: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Superposition: $\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
  - $\alpha^2 + \beta^2 = 1$
- Unitary transformations $U \in \mathbb{C}^{2^n} \times \mathbb{C}^{2^n}$
  - $|\varphi\rangle \mapsto |U \cdot \varphi\rangle$
- Measurement:
  - meas$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{cases} |0\rangle & \text{w/ probability } \alpha^2 \\ |1\rangle & \text{w/ probability } \beta^2 \end{cases}$

# Intel Quantum SDK Stack



intel®