QUANTUM MACHINES

QOP

Quantum Orchestration Platform

Unlocking a New Era in Quantum Computing

The Quantum Orchestration Platform is a first of its kind full-stack solution that allows you to run even the most complex quantum algorithms, from complex multi-qubit calibrations to quantum error correction, right out of the box.

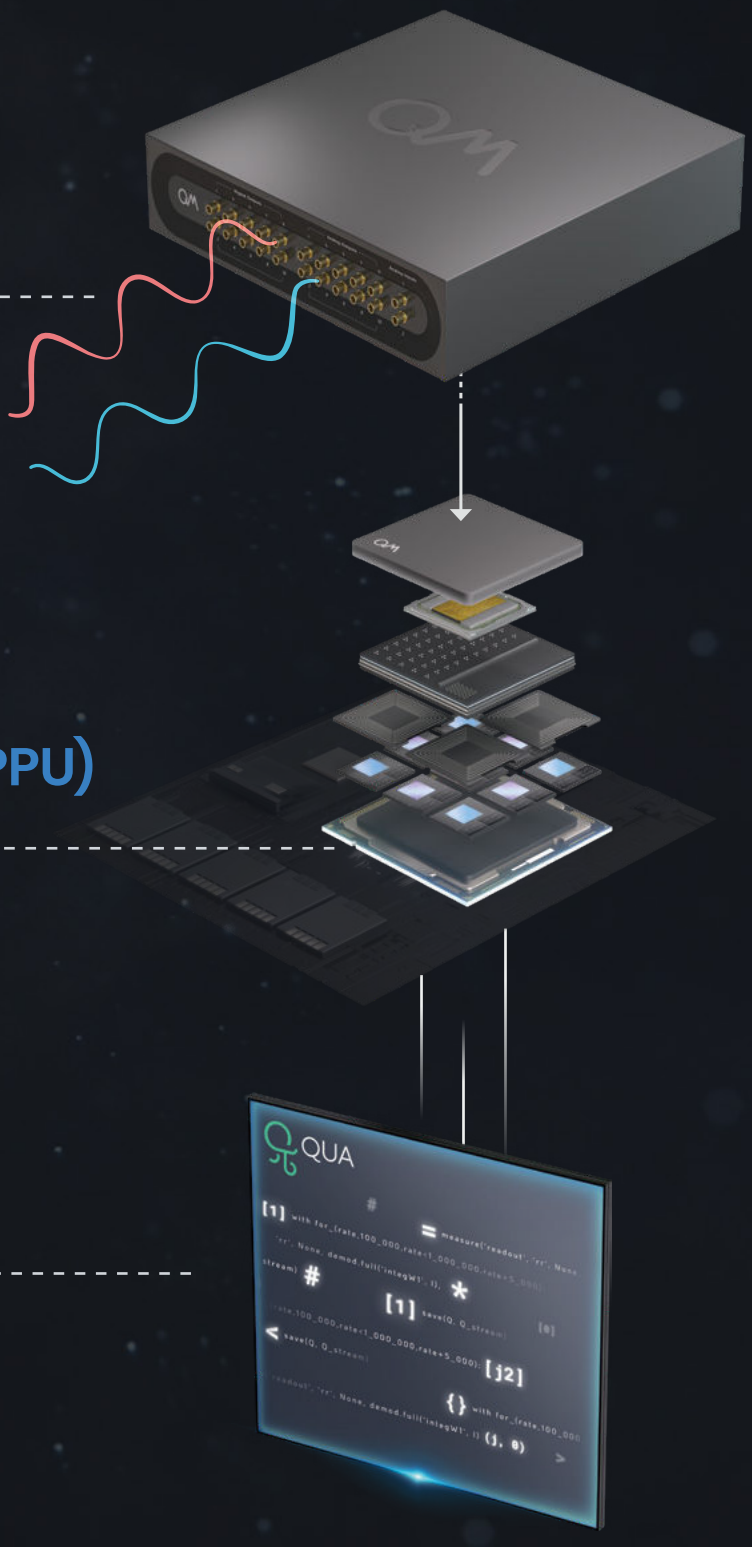
As quantum technologies have made incredible leaps in recent years, we realized there has to be a platform that addresses the the community's vast span of growing needs.

A platform that would allow quantum experimentalists, like ourselves, to run even the most challenging experiments of their wildest dreams.

Welcome to the Era of Quantum Orchestration

Quantum Orchestration Platform

The Quantum Orchestration Platform (QOP) represents a fundamentally new approach to quantum computing. The platform accelerates the quantum research and development of today and enables the quantum breakthroughs of tomorrow.



OPX+

Run State of the Art Experiments with Ease

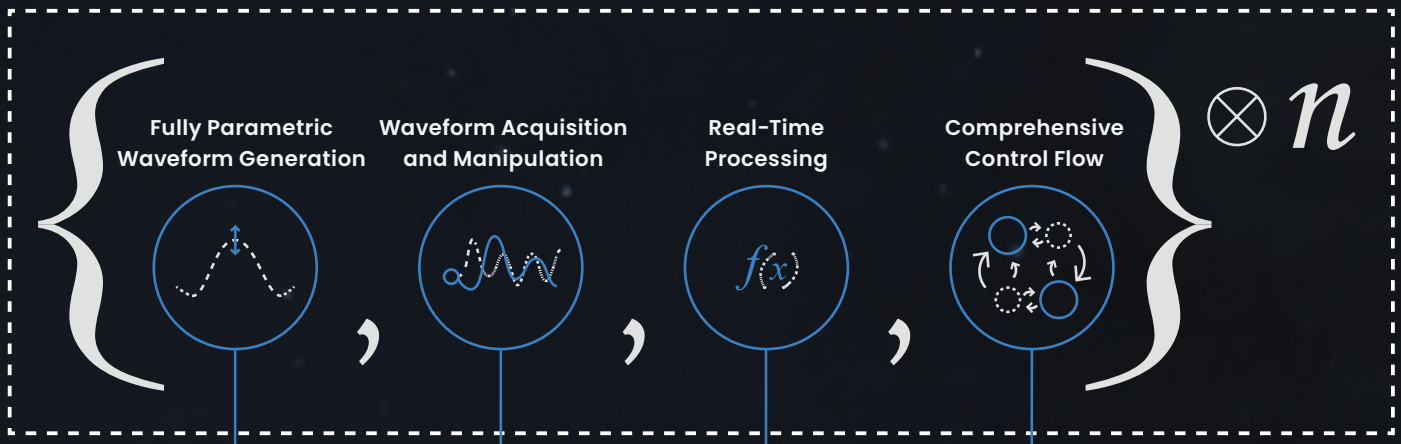
Pulse processing unit (PPU)

Achieve the Fastest Time to Results

QUA

Code Quantum Programs Seamlessly

Your Protocols Live in This Phase Space



Real-Time Waveform Generation

- Fully parametrized: length, frequency, phase (relative & absolute), amplitude, bandwidth, chirp
- Compensations: Crosstalk matrix, FIR and IIR filters
- And many more

Real-Time Measurements

- High fidelity analog to digital conversion
- General integration & demodulation
- Weighted Integrations accumulated integrations, sliced integrations, etc.
- Time Tagging, TTL counting

Real-Time Processing

Turing complete:
- Basic arithmetics
- Evaluation of trigonometric functions
- Vector operations
- Casting of variable types
- And more (Turing complete)

Real-Time Control Flow

- If/else
- For loops
- While loops
- Switch case

Real-Time 'Quantum' Estimations

- State estimations
- Error estimations
- Bayesian estimations
- Correlation functions
- Neural nets based estimations

Real-Time Multi-Qubit Feedback

- Qubit stabilization & tracking
- Quantum error correction
- And many more

The Quantum Orchestration Platform Covers This Space!

- Easily express quantum algorithms and experimental protocols that comprise all of the above.
- Seamlessly sync measurements, real-time calculations, and pulses of different quantum elements.
- Loop over a wide range of parameters in real-time, including intermediate frequencies, amplitudes, phases, delays, integration parameters, measurement axes, etc.
- Use if/else and switch-case statements to condition operations in real time (real time feedback).
- Define procedures (macros) to be reused in the code and access an extensive family of libraries.

OPX+

Run State of the Art Experiments with Ease

Designed from the ground up for quantum control, the OPX+ lets you run the quantum experiments of your dreams right from the installation. Run the most complex quantum algorithms and experiments in a fraction of the development time.

The OPX+ is designed to meet the extremely demanding requirements of quantum control protocols, including complexity, timing, precision, and latency.

OPX+ Is Simply Unmatched. Here's Why:

- **Easily run complex control protocols**
- **Run a wide range of protocols without compromising on specs**
- **Scale up quantum computers and run error correction codes from day one**
- **Consult our quantum team at any time**

Modular System

The OPX+ is completely modular. Adapt it to your needs. Order any number of channels you require for your experiments. 4 or 10? You decide!

Digital Outputs

10 output channels seamlessly synced with any control protocol

Analog Outputs

10 output channels with advanced pulse shaping and frequency multiplexing capabilities

Analog Inputs

2 input channels with easily programmable frequency demultiplexing and pulse analysis capabilities

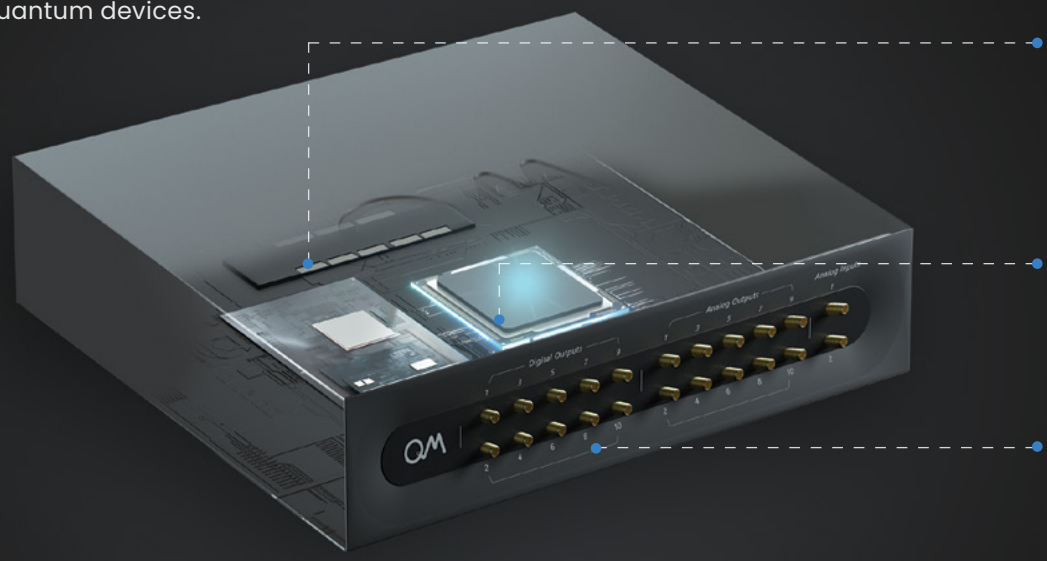
Advanced Use-Cases

- Ultra-low latency (<200 ns) feedback protocols, including quantum error correction
- Real-time state estimation for on-the-fly decision making
- Fast calibration: including pulse parameters, qubit frequency, IQ-mixer imbalances, etc.

An Architecture Made for Quantum

At the core of the OPX+ is the pulse processing unit (PPU), an FPGA-based processing unit comprised of multiple waveform generators, digitizers, and processing units, all integrated within a unique and scalable architecture. The PPU was built to run quantum experiments in the most efficient way and with the lowest latencies possible.

In addition to the PPU, the OPX+'s RF Frontend, which includes digital-to-analog converters (DACs) and analog-to-digital converters (ADCs), was designed to address the demanding requirements for controlling and operating quantum devices.



OPX+ Specs

Analog Outputs Characteristi	
Number of channels	10
Waveform Sampling rate	1 GSa/s
Waveform Vertical Resolution	16 bits
Output Frequency (With QM optional Octave®)	DC to 400 MHz (-6dB) (2 to 18 GHz)
Output Voltage (50 Ω load)	-0.5 to 0.5 V
Rise/Fall Times (10% to 90%)	1 ns
Jitter	< 1ps
Skew between channels	typical: 30-70ps
Noise Floor (0.5V_{PP} , 50 Ω load)	7 nV/√Hz
Worst harmonic component (100MHz)	-45 dBc
Phase Noise floor (100MHz, >200kHz)	-150 dBc
Total Harmonic Distortion (100MHz)	52 dBc
IP3 10MHz between carriers (150MHz)	24 dBm
SFDR (100MHz)	64 dB
Pulse Processor Unit (PPU) - Hadamard™	
Processor cores	18
Simultaneous Demodulation operations / output tones	18
Tones to output ports connectivity	All-to-all
Embedded FPGA	XCVU13P+ UltraScale X
Stream processor	ARM (A72-Cortex 1)
General & Digital Characteristics	
Digital output channels	10
Digital output high	3.3V (LVTTL)
Ext. input triggers	1
External clock	10, 100, 1000 MHz

Analog Inputs Characteristics	
Number of channels	2
Sampling rate	1 GSa/s
Sampling resolution	12 bits
Input Frequency (With QM optional Octave®)	DC to 400 MHz (2 to 18 GHz)
Input Voltage (zero gain)	-0.5 to 0.5 V
Controllable Input Gain	-12 to +20 dB
Time Tagging Resolution	<50 ps
Noise Floor (0.5V_{PP} , 50 Ω load)	typical: 10 nV/√Hz
Total Harmonic Distortion (100MHz)	61 dB
Real-time Capabilities	
Minimal feedback latency[1]	to analog: 198ns to digital: 120ns
Minimal digital[2] feedback latency	to analog: 306ns to digital: 166ns
Phase update latency	0 ns
Frequency update latency	<268 ns
Multi OPX+ Functionality	
Multiple OPX+ interconnectivity	Seamless scalability
Inter-controller additive feedback latency	108 ns
Inter-controller Skew	typical: 30-70ps

[1] Time from last bit of analog input pulse to first bit of analog/digital output pulse, including A2D conversion, integration, demodulation, state-estimation, branching, waveform generation and D2A conversion (for analog only). Value may vary between SW versions

[2] The time from external input trigger to first bit of analog/digital output pulse

Analog Out

- Configure any output channel to work in IQ-mode, single-mode, or multi-mode
- Use IQ-mode to generate arbitrary spectra by using quadrature modulations, output the signal from a single channel or a pair of channels going to an IQ mixer
- Use single-mode when connecting a single channel to a single side-band mixer or an AOM, for direct qubit control
- Use multi-mode for flux-lines or gates for easy calibration and compensation of multi-lines cross-talks

IQ Mode

To IQ mixer

Single Mode

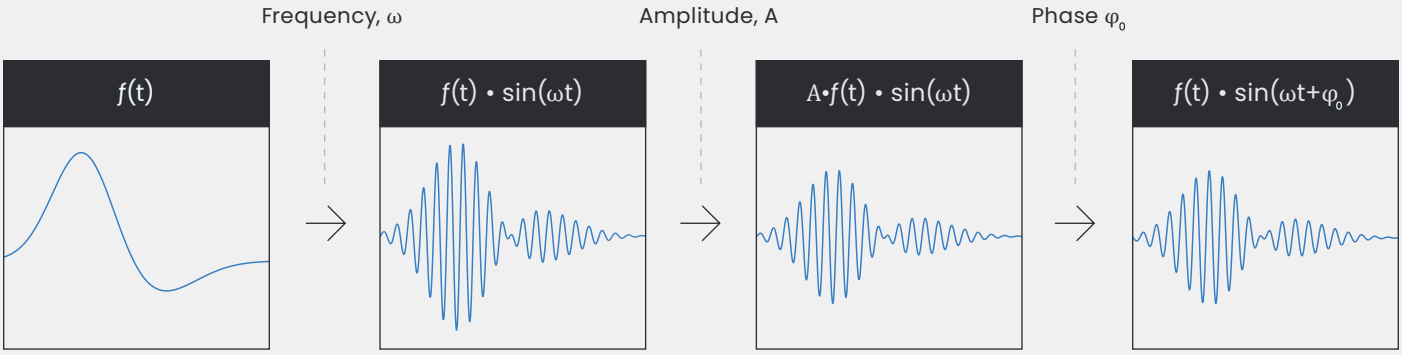
To single side band mixer / AOM / flux-line / other

Multi Mode

To flux-lines / gates / other

Analog Pulse Shaping

- Control pulse frequency, amplitude, phase, and duration
- Automatically shape pulses to fit the calibrated qubit parameters
- Track and control the frame of qubits (perform virtual Z-rotations with zero latency & perfect fidelity)
- Play either arbitrary pulses or constant pulses
- Waveforms are stored in FPGA memory so, when a pulse is playing, utilize the ability to return it to zero, maintain the last value, or ramp it to zero to remove any abruptness

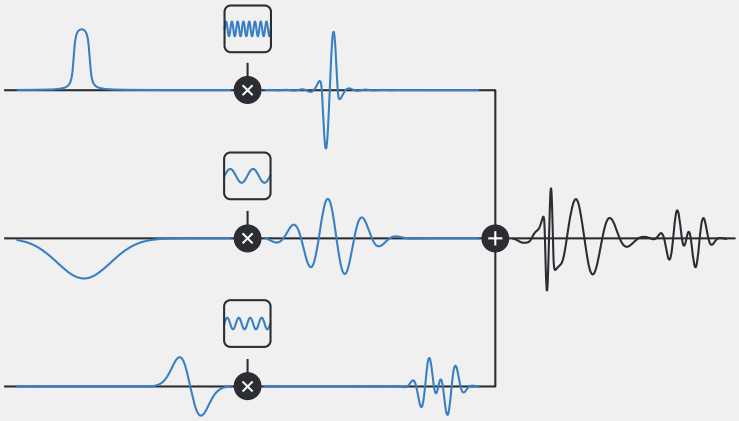


IQ-Mode Additional Features

- Easily program pairs of phase quadrature pulses (IQ-pairs) as if they were a single pulse
- Perform single-side-band frequency modulation
- Calibrate and correct mixer imbalances

Dynamic Resource Allocation

- Play pulses with up to 18 different frequency components to a single output (9 for each channel in IQ-mode)
- Switch back and forth between frequencies with no additional latency, and seamlessly keep track of phases of different frequencies
- Easily calibrate a qubit's resonance frequency by looping over frequencies with ultra-low latency

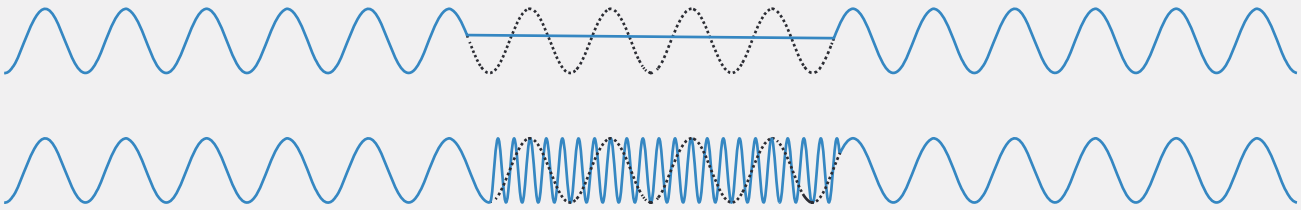


Real-Time Filters

- Configure digital filters on the analog outputs and compensate on the fly for any distortions on the line
- configure both feedback and FIR filters
- Have greater usability in feedback applications due to the minimal latency added (<50 ns)

Real-Time Phase Memory

- Seamlessy preserve phase when switching back and forth between an unlimited number of frequencies



Pulse Types

Pulses can be saved and played in various ways to optimize memory usage according to your needs:

- Arbitrary pulses – Save arbitrary waveforms with a 1 ns time resolution (or any other rate) to be played in real time by the OPX+
- Constant pulses – Play a constant amplitude and frequency for a pre-defined duration or a duration which is determined in real time
- Composed pulses – Compose complex pulses made of several waveforms of different types
- And more

PPU and Compiler

Pulse Compression and Optimal Use of Waveform Memory

Pulses can be saved and played in various ways to optimize memory usage according to your needs:

- Use compressed and constant pulses to save immense amounts of memory. For example, using compressed pulses may effectively decrease the pulse memory by up to six orders of magnitude. Moreover, using constant pulses barely consumes any memory.
- Dramatically reduce memory consumption for most protocols by reusing pulses with real-time pulse shaping. For example, when using a particular pulse with different amplitudes, phases, and frequencies, you only need to save a single pulse in memory, regardless of how many pulse variations there are.

Time to Compile

With the Quantum Orchestration Platform, program compiling and loading times can be orders of magnitude faster than with traditional test equipment.

Analog In

Analog Input

- Simultaneously demodulate and integrate time tag signals' different frequencies from a single input
- Easily calibrate integration weights
- Quickly program complex calculations for state estimation, beyond threshold comparison (Bayesian estimators)

Time Tagging

One nanosecond time-tagging resolution with real-time counting and processing.

- Time tag TTLs with 1 ns resolution with an intuitive time-tagging QUA command
- Count TTLs in user-defined time windows with an intuitive time-tagging QUA command
- Process results of time tagging and counting in real time, employing real-time vector processing

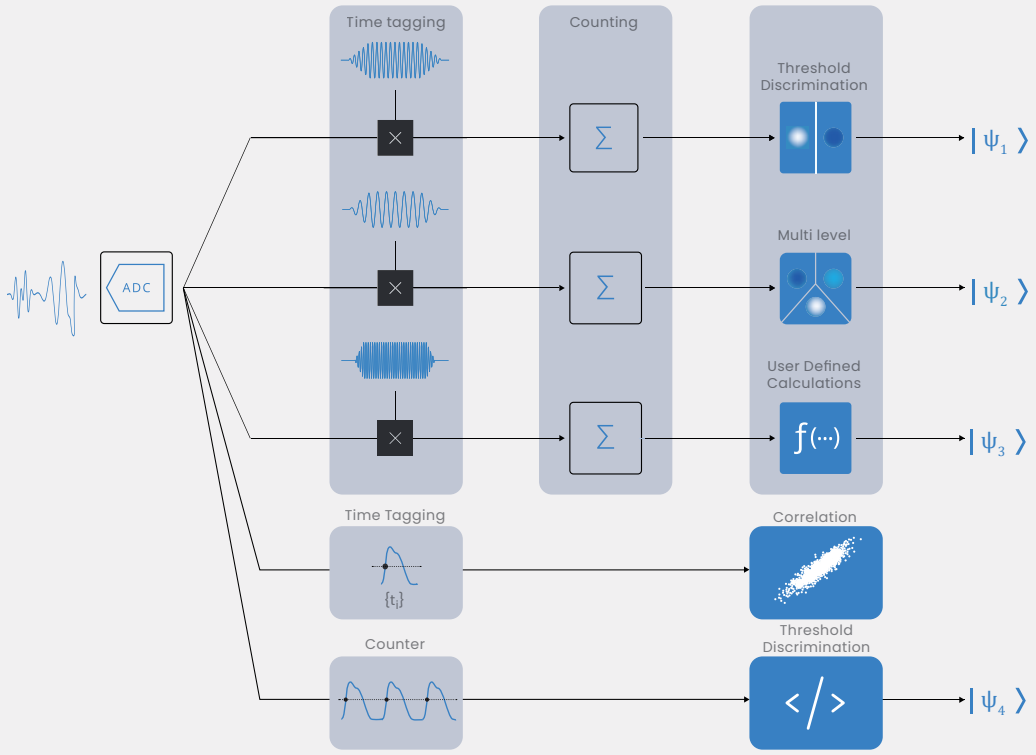
Data Streaming

- Stream acquired data (raw, demodulated, or integrated) or final results of calculations, including state-estimation data, averaged data, etc.
- Perform calculations on the stream from the PPU to the server to the client's PC, saving throughput on your local LAN

Streaming Data In and Out

An extensive framework for streamed processing of results within the OPX+ before they are sent to the client computer to be saved.

- IO values - Data is sent to and from the PPU while the program is running
- Pre-compilation - New waveforms are uploaded after the program is compiled and running



Real-Time Processing and Analysis

Demodulations

- Control the initial phase of the demodulation (rotate the IQ plane) for better thresholding and state-estimation
- Perform continuous measurements and access intermediate measurement results in real time and generate arrays of those results while simultaneously processing them
- Integrate in various ways, including full integration, sliced integration, accumulated integration and moving-window integration
- Integrate with either constant integration weights or arbitrary ones to maximize readout fidelity

Real-Time Processing

- Easily program complex ultra-fast real-time calculations inside the PPU and seamlessly integrate and sync them into any protocol, including complex analysis of input data
- Perform real-time processing from simple arithmetic to complex vector processing
- Access our program libraries, which include simple averaging, density matrix estimation, optimization procedures, frequency demultiplexing, smart calibration procedures, etc.

Feedback

- Use calculation results toreact in real time to feedback on pulse shaping, mixer calibration, integration parameters, and much more
- Use calculation results for real-time program branching
- Run ultra-low latency feedback operations

Digital Out

Digital Pulse Control

- Seamlessly integrate digital pulses into protocols
- Use digital outputs for fast control over external devices (e.g., RF switches, voltage sources, and signal generators)
- Store digital pulses into the dedicated memory and reuse them for any digital output
- Control pulse duration based on real-time feedback from measured data
- Affiliate digital pulses with analog outputs and automatically send digital and analog pulses together
- Easily calibrate optimal delays of digital outputs

Multi OPX+ System

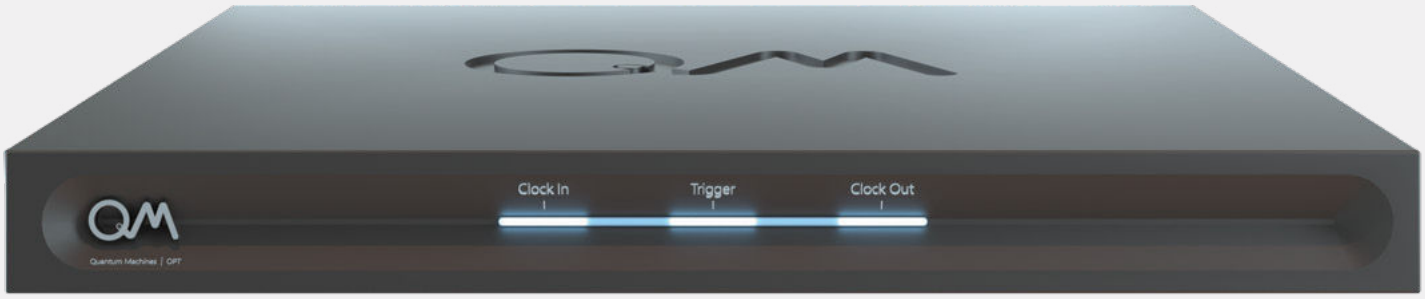
Synchronize All Processors with Ease

Clock Distribution

- The OPT allows for the synchronization of multiple OPXs+. The OPT is a clock distribution system which provides and distributes clocks and triggers, allowing the user to have all OPXs+ and all components within them work with the same clock.
- Multi OPX+ sync allows all FPGAs, DACs, and ADCs to align to a single clock edge.

Clock Source

- The OPX+ works with a single 1 GHz clock, which is used for all major components in the system.
- The clock can be generated in one of three ways:
  - Employing an internal VCXO and a PLL to generate a 1 GHz clock with no external reference
  - Employing an internal PLL to generate the 1 GHz clock from an external 10 MHz reference clock (e.g., an atomic clock)
  - Receiving a 1 GHz clock from an external source to use multiple OPXs+ and have them fully synchronized to the same source clock (see below)



Multi OPX+

- Multiple OPXs+ connected to the same OPT can be seamlessy programmed with QUA as if they were one
- The QUA compiler will take into account any possible latencies which may occur due to sync processes and data transfer, enabling scale up of your control system from a few qubits to hundreds with minimal effort
- Multi OPX+ synchronization is performed thanks to the OPT

QUA : Code Quantum Programs Seamlessly

Implement the Protocols of Your Wildest Dreams, as Easily as Writing Pseudocode

COMPREHENSIVE
Quantum and Classical

QUA unifies universal quantum operations in their 'raw' format (at the pulse level) with universal classical operations used in classical processing (Turing complete) and comprehensive control flow.

EXPRESSIVE
As Much as It Gets

Code even the most advanced programs and run them with the best possible performance.

Natively describe your most challenging experiments, from complex AI-based multi-qubit calibrations to quantum error correction.

SCALABLE
Grows with You

QUA scales with you to enhance your quantum algorithms and experiments of today and tomorrow. With QUA, coding 1,000 qubits is as easy as coding 1 qubit. QUA removes any limitations in implementing your protocols, from the simplest to the most complex.


Quantum Computing

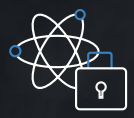

Quantum Sensing


Hybrid Quantum - Classical Algorithmics


Quantum Technologies Research & Development


Quantum Simulations


Quantum Communication


Quantum Firmware Development

QUA is a first of its kind pulse level programming language integrating classical processing into the lowest levels of quantum programming in an unprecedented way. It unifies universal quantum operations in their 'raw' format – all the pulse-level stuff used to control and measure qubits – with universal classical operations used in classical processing – all the good stuff you know from Python, Matlab, Java, etc.

Run Your Quantum Experiments with Ease

- Randomized & cross entropy benchmarking
- Multi-qubit active reset
- Quantum error correction (e.g. cat codes, surface code)
- From Rabi, Ramsey & spectroscopy to neural-nets-based calibrations
- Qubit state tracking and qubit stabilization
- Real-time atom sorting
- Bayesian estimation-based adaptive sensing
- Multi-node entanglement distillation
- [Your Next Dream Protocol Here !]

QUANTUM MACHINES

Code 1,000 Qubits as Easy as 1

Quantum Complete

Native quantum pulse-level operations

```
0  play('pi_pulse', 'qubit1_xy')
1  measure('readout_pulse', 'qubit12', raw_data,  demod.full('cos', result))
```

Turing Complete

Comprehensive classical processing of classical variables

```
2  assign(a, Math.cos(x) * Math.sqrt(y))  # a = cos(x)*sqrt(y)
3  assign(b, Math.abs(z) + Math.ln(w))    # b = abs(z)*ln(w)
```

Comprehensive control flow

```
4  while(...), for(...), if/else(...), switch-case      # nest them !!
```

Quantum Classical

Quantum operations based on classical variables and calculations

```
5  play('pi_pulse'*amp(a), 'q1_xy', duration=b,          # a from line 2
6        chirp=(Math.cos(a) * Math.exp(b), 'Hz/nsec'))  # b from line 3
7  measure('readout_pulse', 'QPC',                       # signal integration
8          integration.full('weights', result))
9  measure('trigger', 'laser',                           # time-tagging
10         time_tagging.analog(timestamps, length, counts))
```

Classical calculations based on quantum measurements

```
11 assign(state_estimation,
12        0.5 * (1 + result * (alpha+beta*C)))    # result from lines 7-8
13 assign(error_syndrome,
14        ancilla_result[0] & ~ancilla_result[1]) # vector of errors
```

Control flow based on classical variables based on quantum measurements

```
15 if_(error_estimation):
16     play('pi'*amp(Math.ArgMax(all_states)), 'q1_xy') # analog feedback
17 while_(error_syndrome == 0):
18     do_something()                    # user-defined QUA macro
```

Comprehensive Timing Control

Absolute timing control (relative to global time-stamp) & relative timing control (sync & async multi-threading)

Pseudocode vs QUA Code - STOP Algorithm (AWS)

```
initialize: t=(d-1)/2; n_diff=0; countSyn=1;
 SynRep=1; n_diff Increase=0;
while test= 0 do
   if n_diff=t then
      test=1
   end
    Measure the error syndrome s_j.
          Store the error syndrome s_j-1
          from the previous round in
          synPreviousRound and the
          current syndrome s_j in
          synCurrentRound.;
    if countSyn>1then
       if synPreviousRound=synCurrentRound
          then
          SynRep=SynRep+1;
          n_diff Increase=0;
       else
          SynRep=0;
          if n_diff Increase=0 then
             n_diff=n_diff+1;
             n_diff Increase=1;
          else
             n_diff Increase=0;
          end
       end
    end
    if SynRep=t-n_diff+1 then
       test=1;
    end
    countSyn=countSyn+1;
end
```

* arXiv:2012.04108

```
0  while_(test == False):
1      if_(n_diff == t):
2          assign(test, 1)
3      assign_vec(synPrevRound, synCurrRound)             # vector processing
4      measure_ZL_syndrome(logical_qubit, synCurrRound) # QUA macro
5      if_(countSyn > 1):
6          if_(synCurrRound == synPrevRound):
7              assign(SynRep, SynRep + 1)
8              assign(n_diffInc, 0)
9          else_():
10             assign(SynRep, 0)
11             if_(n_diffInc == 0):
12                 assign(n_diff, n_diff + 1)
13                 assign(n_diffInc, 1)
14             else_():
15                 assign(n_diffInc, 0)
16     if_(SynRep == t - n_diff + 1):
17         assign(test, 1)
18     assign(countSyn, countSyn + 1)
```

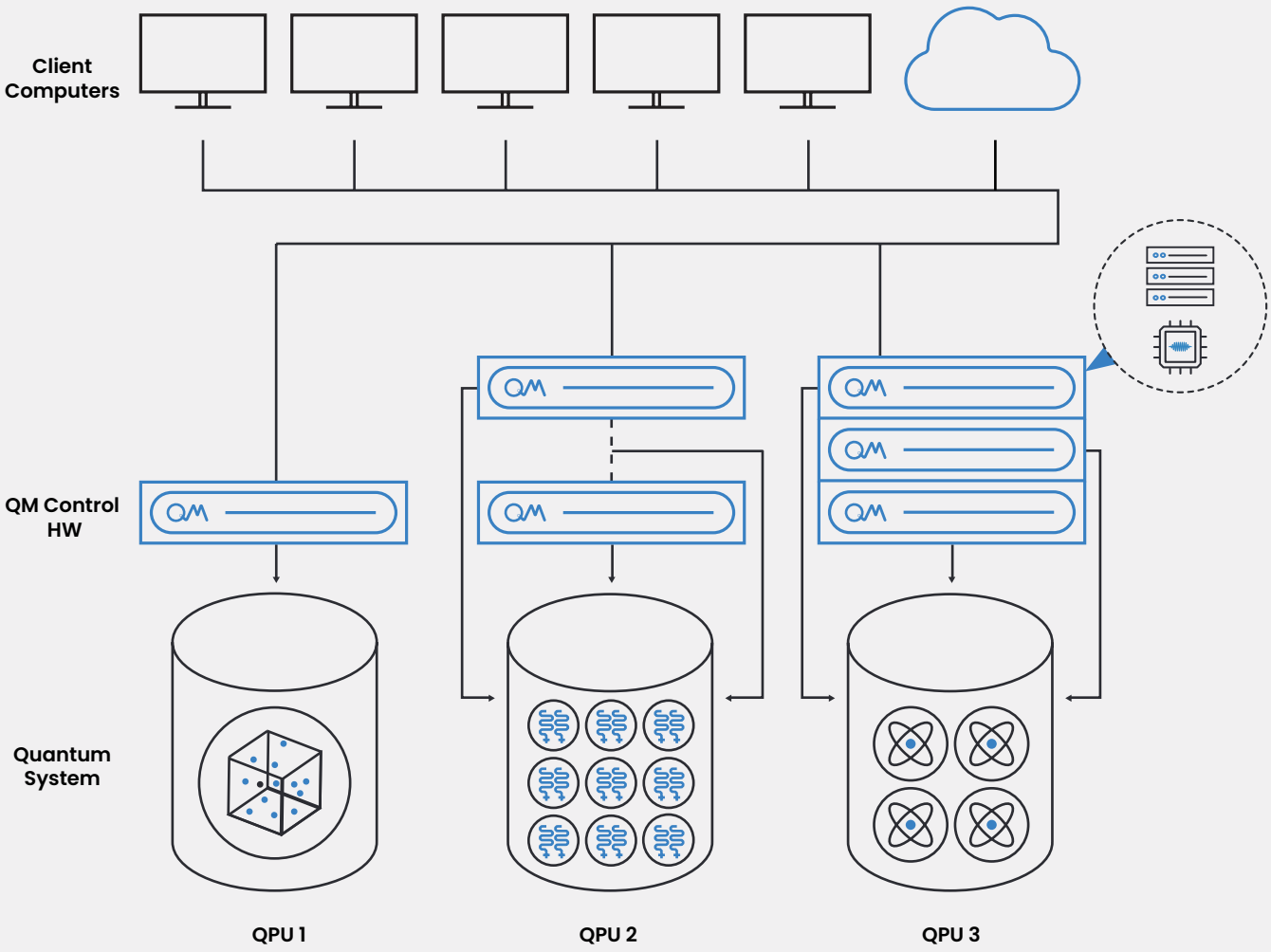
Stream Processing

Classical to Power Quantum

Employ and leverage the stream processing capabilities of the QOP: a new platform for real-time processing of data streams arriving from the OPX+. While the most demanding real-time processing takes place in the PPU within the OPX+, additional real-time processing can take place on a separate dedicated processor. In between the client and the PPU, lies a Linux server capable of performing calculations on the data streamed out of the PPU, allowing you to run real-time processes, such as FFTs, averages, and cross-correlation measurements to hybrid algorithms like VQE and QAOA.

Server Architechture

- A single OPX+ or a cluster of OPXs+ can be connected to multiple quantum devices so that various users on several computers can run experiments on those different quantum devices simultaneously and independently.
- All OPXs+ are connected to a dedicated router, which is connected to the local network. Any user on the local network can send jobs to any OPX+ through the server.
- On-the-fly calculate the data on the server before it even reaches the client PC, supported up to 40 GB ethernet connection



QUANTUM MACHINES

If you wish to learn more:

info@quantum-machines.co

About Quantum Machines

Quantum Machines accelerates the realization of useful quantum computers that will disrupt all industries. Supporting multiple Quantum Processing Unit (QPU) technologies, the company's Quantum Orchestration Platform (QOP) fundamentally redefines the control and operations architecture of quantum processors with unprecedented levels of scalability, performance, and productivity.

Our rich product portfolio, including full stack (hardware and software) quantum control and state-of-the-art quantum electronics empowers academia and national labs, HPC centers, enterprises, and cloud service providers building quantum computers all over the world. To learn more, please visit quantum-machines.co.