

From neutral atom arrays to real world applications

Daniel Garcia Guijo – Technical Project Delivery Manager

01

Introduction
to Pasqal

02

Software
Stack: Low-
level

03

Software
Stack: High-
level

04

Applications

Introduction to Pasqal

Quantum Computing with Neutral Atoms



Founded 5 years ago, backed by
Nobel prize-winning technology

15+ patents and applications

800+ publications

250+ employees

We build neutral atom-based
Quantum Processing Units

350+ qubits

Analog and Digital-Analog mode

Application-driven research and full-
stack approach

Public Roadmap



Technology
PASQAL & affiliated ecosystem

Products

	2022 - 2023	2024 - 2025	2026 - 2027	2028+		
HARDWARE PLATFORM						
Max qubits	200	1,000	10,000			
Addressability	Z add	Z+X add	Addressable 1Q and 2Q gates			
Base repetition rate	1 Hz	3 Hz	10 Hz	100 Hz		
FTQC Program		Atom shuttling	Ultra High-Fidelity Gates	Scalable logical qubits architecture		
HARDWARE ACCELERATED LIBRARIES						
Quantum Matter & Quantum AI	Algorithm Blueprint	Algorithm Development	Production			
QUANTUM PROCESSORS						
Generation	Orion Alpha ~3M gates	Orion Beta ~5M gates On premise delivery	Orion Gamma ~10M gates On premise delivery	Vela ~40M gates	Pegasus ~200M gates	Centaurus FTQC QPU 128+ Logical qubits 200M+ gates
Total hours of QPU for users	500	5-10,000	20-30,000	60-70,000	200-250,000	500-550,000
Factories	France	Canada	Factory 3			
COMMUNITY						
Platform		Learn	Interact	Collaborate		
Open-source Software Stack	Pulser	Qadence	Solvers & Emulators			

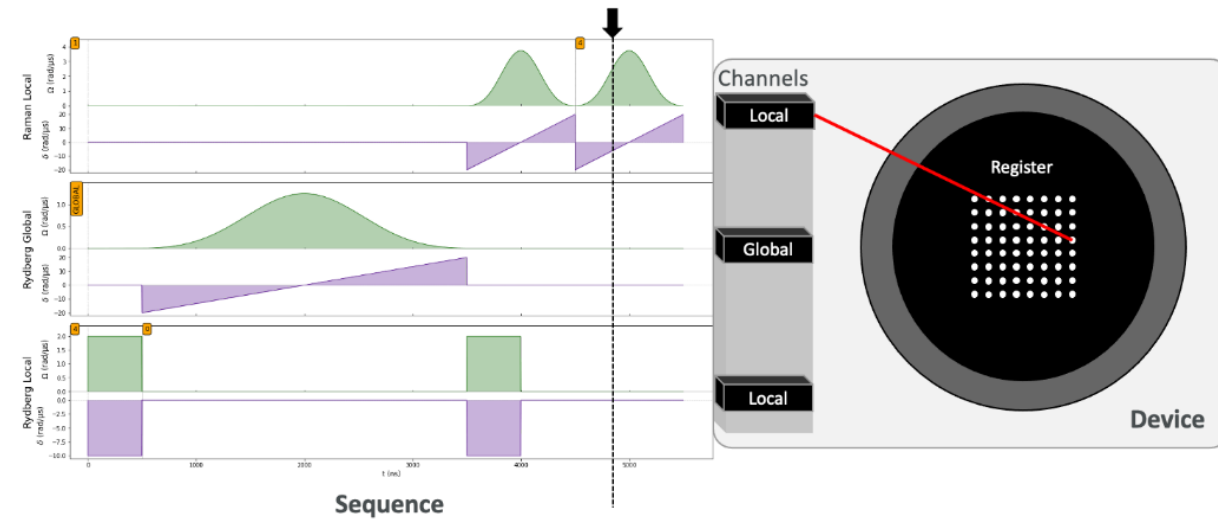
Software Stack: Low-Level

Pulser & Pulser Studio

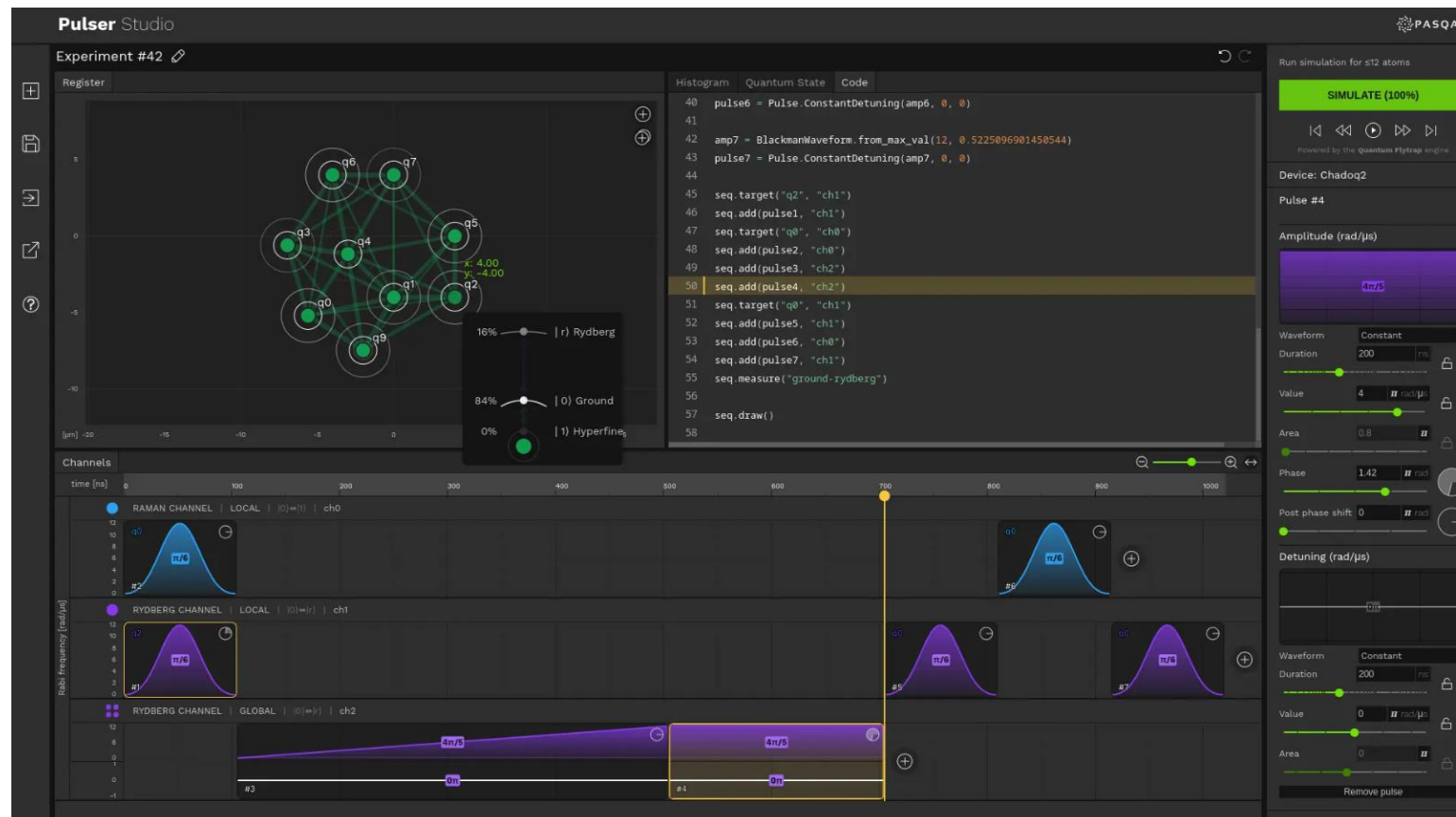


Pulser

- Open-source Python package for programming neutral atom arrays.
- Allows for both **digital** and **analog** quantum simulations.
- Whatever can be done in the device, it can be done with Pulser.



Zero-code neutral atom quantum computing platform



The screenshot displays the Pulser Studio interface for an experiment named "Experiment #42". The main workspace is divided into several panels:

- Register:** A 2D plot showing the spatial arrangement of 10 qubits (q0 to q9) in a triangular lattice. A zoomed-in view of q2 shows its energy levels: |r> Rydberg (16%), |0> Ground (84%), and |1> Hyperfine (0%).
- Code:** A Python script defining the pulse sequence:

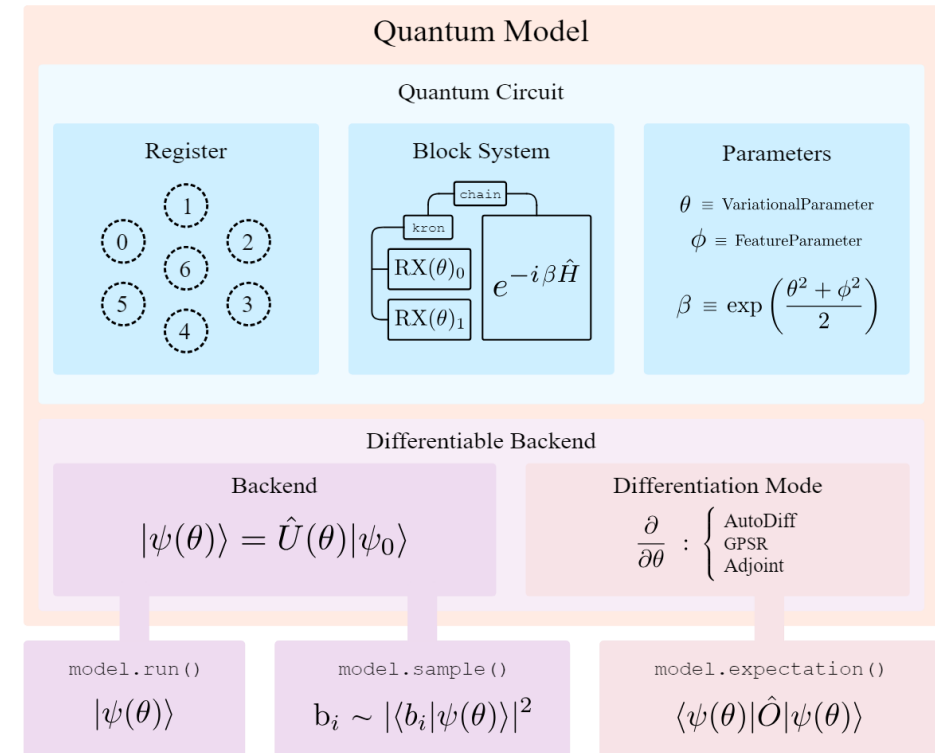
```
40 pulse6 = Pulse.ConstantDetuning(amp6, 0, 0)
41
42 amp7 = BlackmanWaveform.from_max_val(12, 0.5225896981458544)
43 pulse7 = Pulse.ConstantDetuning(amp7, 0, 0)
44
45 seq_target("q2", "ch1")
46 seq_add(pulse1, "ch1")
47 seq_target("q0", "ch0")
48 seq_add(pulse2, "ch0")
49 seq_add(pulse3, "ch2")
50 seq_add(pulse4, "ch2")
51 seq_target("q0", "ch1")
52 seq_add(pulse5, "ch1")
53 seq_add(pulse6, "ch0")
54 seq_add(pulse7, "ch1")
55 seq_measure("ground-rydberg")
56
57 seq.draw()
58
```
- Channels:** A timeline view showing the frequency of three channels over time (0 to 1000 ns). The channels are:
 - RAMAN CHANNEL (LOCAL | 0>=|1>):** ch0, with a pulse at approximately 700 ns.
 - RYDBERG CHANNEL (LOCAL | 0>=|r>):** ch1, with a pulse at approximately 700 ns.
 - RYDBERG CHANNEL (GLOBAL | 0>=|1>):** ch2, with a pulse at approximately 700 ns.
- Simulation Controls:** A panel on the right with a "SIMULATE (100%)" button, playback controls, and device information (Chadog2). Below this are controls for Pulse #4, including Amplitude (4 rad/μs), Duration (200 ns), Value (0 rad/μs), Area (0), and Phase (1.42 rad).

Software Stack: High-Level

Qadence



- First open-source software for digital-analog quantum computing.
- Flexible interface, native differentiability, and focus on real-device execution.
- Aimed at advancing research on variational algorithms built for native digital-analog platforms.



Applications

Real world use cases



Applications

- Develop domain-specific solvers for industrial applications.
- Minimize the need for quantum computing knowledge.
- Provide solvers and emulators to end-users.



Any questions?



Thanks for coming!