Quekit-SDK - An Indigenous SDK for virtual trapped-ion and quantum physics simulation

Vicky Budhiraja[1†], Jitendra Verma[1], V. Ravishankar[2], and Sugam Kumar[3]

[1] QUTE Electronics, Vasant Kunj, New Delhi, 110070, India
[2] Department of Arts, Science, and Design, Indian Institute of Information Technology Dharwad, 580009, India
[3] Inter-University Accelerator Centre, New Delhi, 110067, India.

Synopsis We present Qutekit software development kit for trapped-ion quantum hardware. It provides virtual models of the main hardware used in an ion-trap experiment, such as the ion, laser, AOM, DDS, FPGA, and photon detector. A quantum program written by the user is converted into the required sequence of hardware operations. The software can be used to test these operations on a virtual system before running them on real hardware.

We have designed and developed a hardware-aware software development kit, Qutekit, for modeling various trapped-ion systems, compiling, and executing quantum programs on virtual quantum hardware. It provides virtual hardware setup capabilities, hardware chain workflows, and an execution framework in which user-level quantum instructions are translated into device-level control operations to propagate through a configurable and customisable virtual hardware stack.

The virtual hardware stack represents canonical components such as trapped-ion systems, lasers, acousto-optic modulators (AOM), direct digital synthesizers (DDS), FPGA-based timing control, and photon-detection hardware [1]. A quantum circuit given by the user is converted into a sequence of physical operations. For example, a quantum gate can be converted into laser pulses with defined frequency, phase, amplitude, and duration. The timing of these operations is also defined so that the different hardware units work in the required sequence. Ion state evolution and measurement are included in the virtual system [2]. This creates an execution path from a quantum circuit to a pulse generation sequence, ion-laser interaction, state evolution, and measurement, examined via a single Qutekit SDK framework.

The main software core of Quekit-SDK is written using a high-performance Rust core with a minimalist and simple Python interface. This enables quantum programs to be defined from Python while retaining strongly typed device models and physics-oriented execution in the underlying Rust-based SDK core.

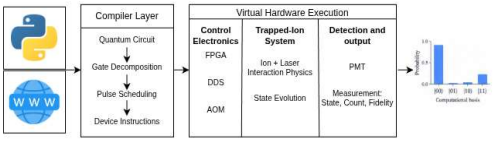

Figure 1. High-level SDK architecture diagram

The present work focuses on trapped-ion systems and provides a virtual hardware environment for developing and testing quantum control programs. The same software structure can later be connected to experimental control electronics.

References

[1] C. Lee,*Characterisation and control of trapped-ion qubit* (Ph.D.thesis, Imperial Coll., London, 2022) **10.25560/** 100200

[2] Liebfried D *et al.* 2000 arXiv:quant-ph/0009105

†E-mail: vicky@quteelectronics.com