QUASAR - QUantum Assembler for Star-topology ARchitectures

Jakab Nádori[1,2], Péter Boross[1], Győző Egri[1], Zoltán Zimborás[5,6], András Pályi[2,3,4]

[1] Qutility @ Faulhorn Labs, Budapest, Hungary [2] Department of Physics of Complex Systems, ELTE, Budapest, Hungary
[3] HUN-REN-BME-BCE Quantum Technology Research Group, Budapest, Hungary [4] Department of Theoretical Physics, BME, Budapest, Hungary
[5] Quantum Computing and Quantum Information Research Group, HUN-REN Wigner Research Centre forPhysics Budapest, Hungary.
[6] Department of Physics, University of Helsinki, Helsinki, Finland

qutility.io
info@qutility.io

Abstract

As quantum processors and quantum technologies continue to scale, efficient hardware-aware compilation is becoming increasingly important for the execution of quantum algorithms and quantum sensing protocols on real devices. Central-spin platforms, such as nitrogen-vacancy (NV) centers in diamond, are promising for both quantum computing and quantum sensing. However, their star-topology connectivity and asymmetric single-qubit gate costs make circuit compilation particularly challenging and motivate architecture-specific approaches.
We present QUASAR (QUantum Assembler for Star-topology ARchitectures), a Python package for compiling quantum circuits on star-topology devices. QUASAR supports both gate-level and pulse-level compilation, generating optimized pulse sequences for NV-center-based quantum computing and sensing applications. At the gate level, building on the unitary decomposition of SQUANDER[1], it combines bottom-up and top-down synthesis strategies, to efficiently compile both shallow and deep circuits. Compared with established compilers such as Qiskit[2], QUASAR can help reduce two-qubit gate count and total pulse execution time while supporting circuits with up to 5 qubits in the benchmarks shown, with additional support for circuits up to 32 qubits via partitioning. In addition, QUASAR supports gate set- and fidelity-aware approximate compilation, allowing a controlled trade-off between circuit depth and gate errors. These improvements are expected to reduce noise accumulation and improve execution fidelity on real hardware.

[1] Péter Rakyta and Zoltán Zimborás. Efficient quantum gate decomposition via adaptive circuit compression, 2022.
[2] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D Nation, Lev S Bishop, Andrew W Cross, Blake R Johnson, and Jay M Gambetta. Quantum computing with Qiskit.

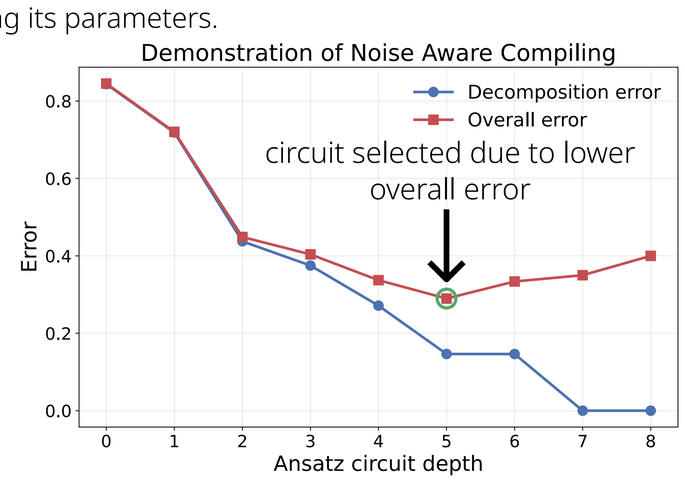
Noise Aware Compilation

To realize theoretical textbook quantum circuits on actual quantum hardware we need to express them in a device native gateset and qubit-topology compliant circuit. We can define a cost function distance F between the quantum circuit U and the approximate device native V(θ)**:**

$$F(U, V(\theta)) = \frac{1}{d^2}|\mathrm{Tr}(U^\dagger V(\theta))|^2$$

With this cost function we can optimize the parameters of V with numerical optimizers (ADAM, BFGS). Increasing the number of parameters and entangling gates lowers the cost function, however as we increase the gate count the actual fidelity of circuits will lower due to noise despite the decomposition being numerically exact (distances lower than 10^{-8}). To balance numerical accuracy against noise-induced fidelity loss in deeper circuits, we introduce a noise-aware search heuristic.
QUASAR's compilation process is built upon SQUANDER, a comprehensive unitary decomposition package. This decomposition relies on fundamental two-qubit sub-circuits known as templates; to determine their optimal placement, SQUANDER employs a tree search algorithm that defines the circuit through its parent nodes while simultaneously optimizing its parameters.

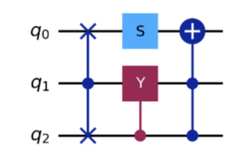
Noise Aware Compilation:
Decomposition tolerance adjusted based on layer infidilety and circuit depth. Leads to shorter pulse times with no real loss in



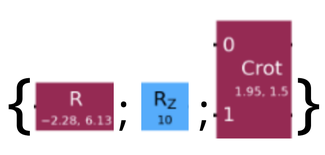
What does QUASAR do?

Inputs

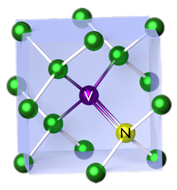
Textbook circuit



Native gateset

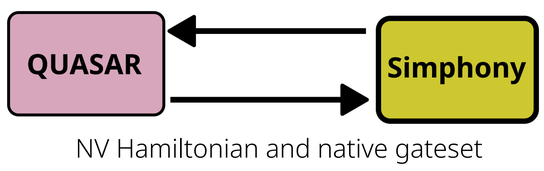


NV Hamiltonian/Simphony model



Phase I: Pre calculation

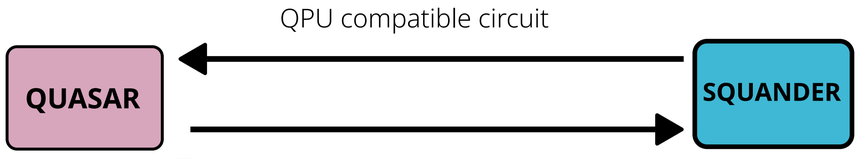
Optimized pulses corresponding to native gates



NV Hamiltonian and native gateset

Phase II: Unitary decomposition (Gate-level compiling)

QPU compatible circuit



Topology, native gateset and qubit mapping

Phase III: Execution on QPU (Pulse-level compiling)



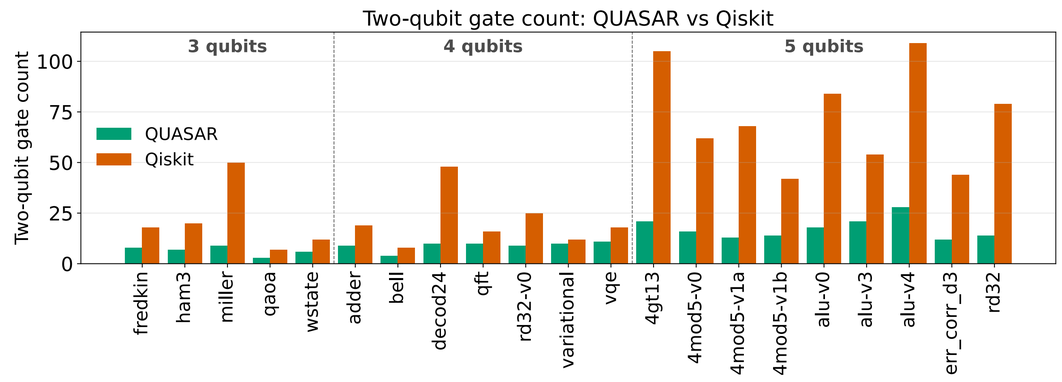
QUASAR connects high-level quantum circuits to hardware execution in three stages. First, it uses the NV Hamiltonian and native gate set to precompute optimized pulse implementations with Simphony. Next, it performs gate-level compilation by rewriting the textbook circuit into a QPU-compatible form, including native-gate decomposition, topology constraints, and qubit mapping. Finally, it translates the compiled circuit into pulse sequences that can be executed directly on the target QPU.

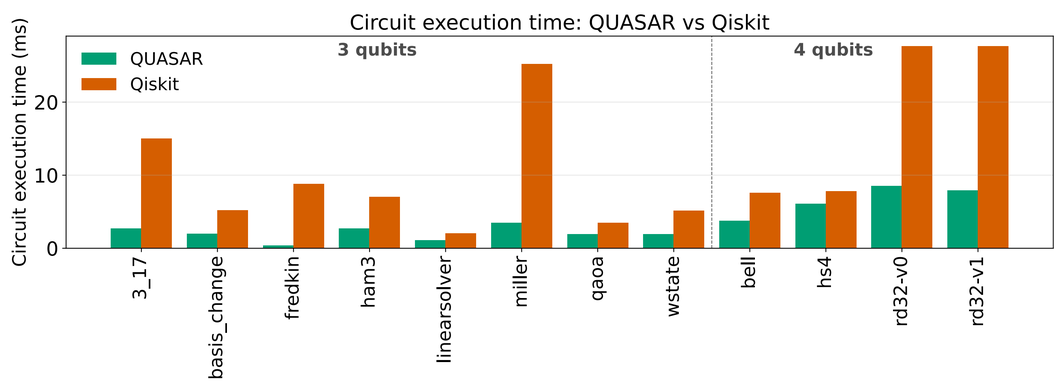
Features:
- 3 gatesets supported (DDrf - CROT gate, CPhase gate, fixed angle DDrf gate)
- Extra Pauli basis change circuits for arbitrary expectation value measurement
- Noise aware decomposition

Future plans:
- Noisy simulations during decomposition
- Support for compilation beyond 5 qubits
- Gate-level fidelity aware decomposition
- Compilation for coupled NV-centers
- Extended noise aware compilation to 5+ qubits

Benchmarking QUASAR against Qiskit

Compiling for the DDrf gateset up to 5 qubits:





QUASAR reduces the two-qubit gate count by **62.8%** and circuit execution by **62.9%** on average compared to Qiskit.

Acknowledgment

This research was supported by the National Research Development and Innovation Office of Hungary within the German-Hungarian EUREKA programme (Project No. 2020-1.2.3-EUREKA-2022-00029).

National Research, Development and Innovation Office Hungary
PROJECT FINANCED FROM THE NRDI FUND