A Tool for Quantum Circuits Analysis

Filip Novák, 2026
Supervisor: doc. Ing. Ondřej Lengál Ph.D

Quantum Simulation

As large, reliable quantum computers are not yet widely available, we rely on classical simulation to design and validate quantum algorithms. Classical simulators model the evolution of a quantum state under a sequence of gates, typically representing the state as a complex-valued vector whose dimension grows as 2^n with the number of qubits. One such simulator is **MEDUSA** [1], which uses an MTBDD-based state representation and forms the basis for our work.

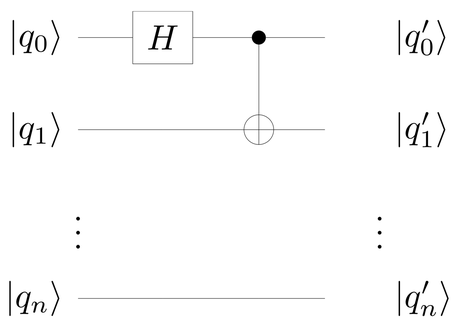

Figure 1: Example quantum circuit

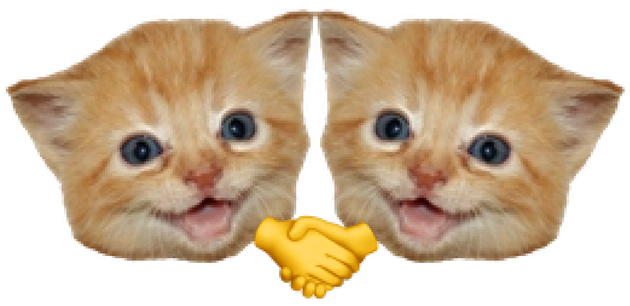

Figure 2: Kittens in quantum entanglement

OpenQASM Frontend

```
"tree": "tree_id",
"traverse":
   "to": "c",
   "action":
      "on": "R", // apply nested action only on R side
      "traverse":
         "to": "t",
      "action":
         "type": "swap" // simple swap of children
```

Figure 4: Custom tree operation representing CX gate in one pass of the tree

$$B_{qc} \cdot T + \overline{B}_{q_c} \cdot \left(B_{q_t} \cdot T_{q_t} + \overline{B}_{q_t} \cdot \overline{T}_{q_t} \right)$$

Figure 5: Update formula representing CX gate

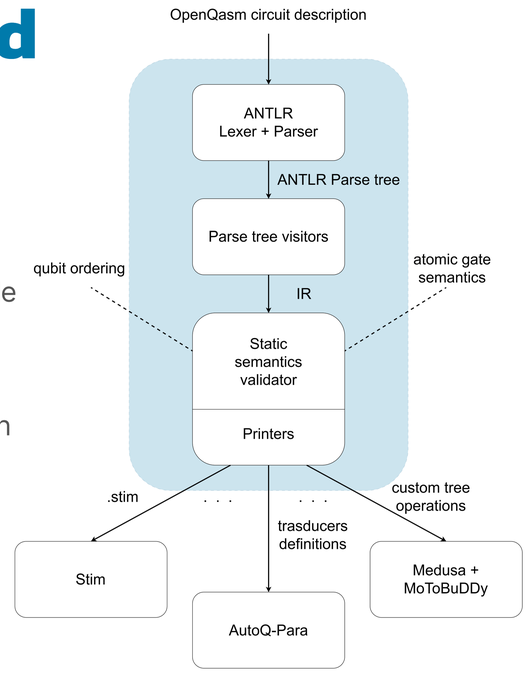

Figure 6: qFront pipeline diagram

MTBDDs

Multi-Terminal Binary Decision Diagrams are an extension of Binary Decision Diagrams, a representation of Boolean functions. MTBDDs extend the domain of values in their terminal nodes from binary to any countable domain:

$$f_x : \{0,1\}^k \to \mathbb{D}$$

To employ them in quantum simulation, each variable corresponds to one qubit, each root-to-leaf path encodes a basis state, and the terminal node holds the corresponding amplitude. Nodes with identical subtrees are shared, so states with repeated or structured amplitude patterns are stored compactly, avoiding the exponential memory that a flat array would require.

MoToBuDDy

MEDUSA originally uses **Sylvan**, a parallel C MTBDD library, whose multi-core synchronization overhead yields no practical benefit here. We instead extended **BuDDy** [2], a lightweight sequential C BDD library, with MTBDD support for a leaner backend.

We also introduce a floating-point representation of complex amplitudes as an alternative to the original algebraic **MPZ**-backed one, which, while exact, restricts the supported gate set. Floating-point lifts this restriction, enabling gates such as **RX** and **RY** at arbitrary angles.

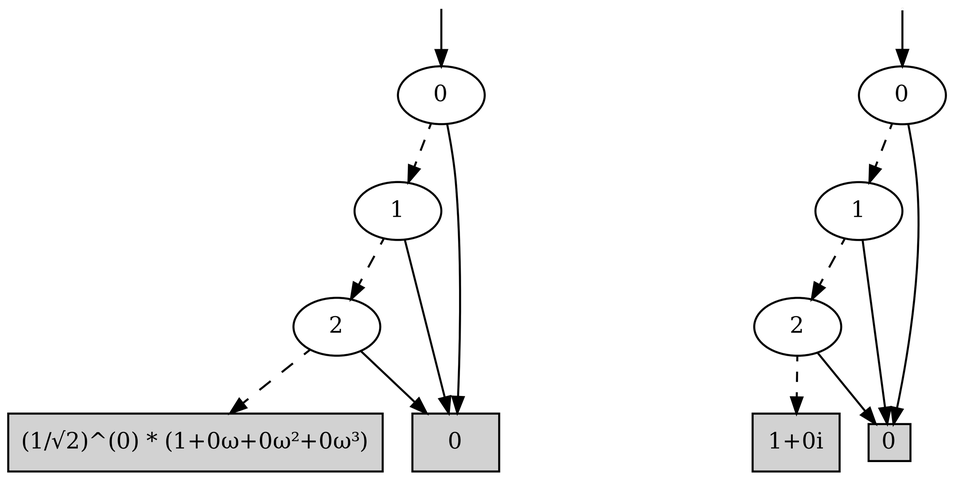

Figure 3: An MTBDD representing the quantum state $|000\rangle$ using algebraic complex number representation (left) and floating-point real/imaginary representation (right).

To quantify floating-point precision loss, the sum of squared amplitudes over all basis states was tracked after each gate application and loop summarization and should equal 1 by definition. The maximum deviation from this value for several precisions is reported in Table [1].

Precision	Max. Deviation
Quadruple (128-bit)	1×10^{-29}
Long Double (80-bit)	1×10^{-14}
Double (64-bit)	1×10^{-11}
Float (32-bit)	1×10^{-2}

Table 1: Maximal probability deviation per floating-point precision (Grover-24)

Benchmarks

Benchmarks were conducted across multiple quantum circuit families, comparing against several state-of-the-art simulators: **Quokka#**, **Quasimodo**, **SliQSim**, **DDSIM**, and **QSylvan**, with a timeout of one hour. A significant speedup is observed on amplitude amplification (Grover) circuits. We report results for the quadruple-precision (f128) variant, as it maintains numerical accuracy throughout. It is worth noting that the double-precision (f64) variant achieves a speedup of roughly **6916×**, 0.454s versus 3140s compared to the original **MEDUSA**. However, both f64 and f32 fail on larger circuits due to floating-point inaccuracies that produce malformed decision trees.

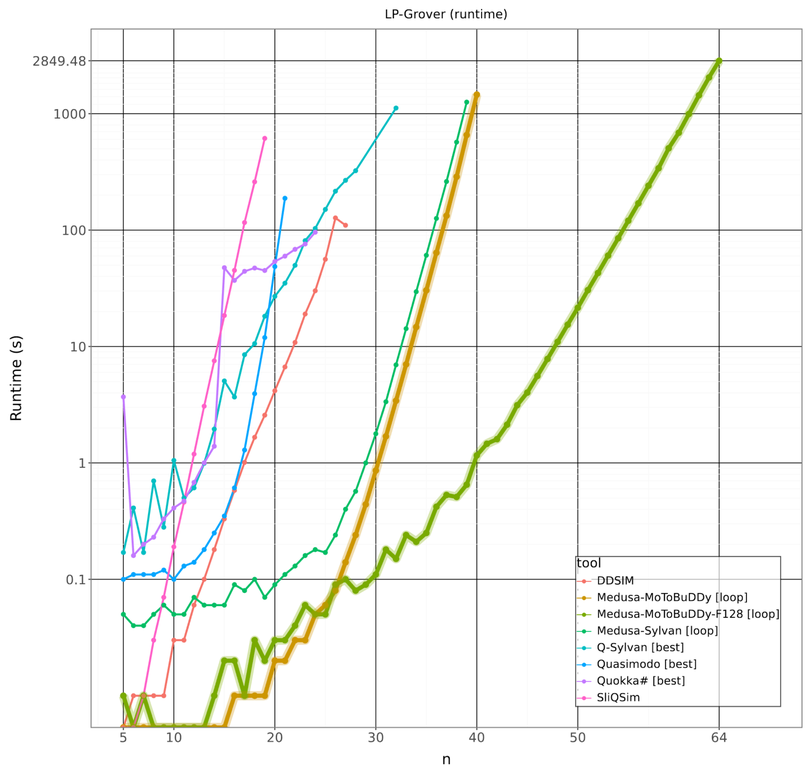

Figure 7: Comparison of simulators on Grover circuits

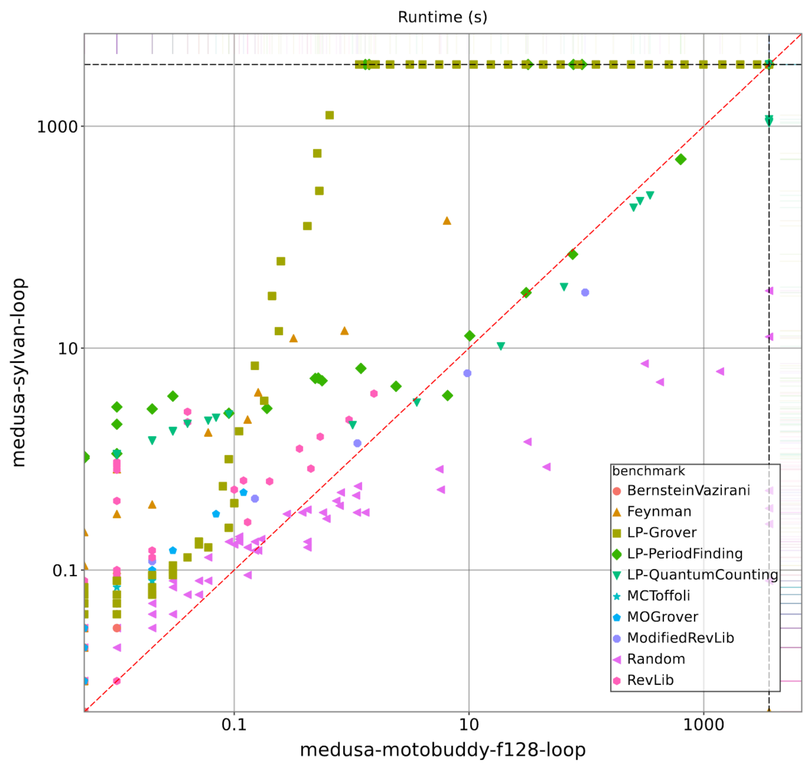

Figure 8: Runtime scatter plot: Sylvan MEDUSA vs. MoToBuDDy backend with floating-point terminals

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026

References:
[1] Chen, T.-F.; Chen, Y.-F.; Jiang, J.-H. R.; Jobranová, S. and Lengál, O. Accelerating Quantum Circuit Simulation with Symbolic Execution and Loop Summarization. In: Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. New York, NY, USA: Association for Computing Machinery, 2025. ISBN 9798400710773. Available at: https://doi.org/10.1145/3676536.3676711.
[2] Lind Nielsen, J. BuDDy: A Binary Decision Diagram Package. Technical Report. Department of Information Technology, Technical University of Denmark, 1999.