BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023

MTBDD-based Quantum Circuit Simulation

Sára Jobranová, 2023

Supervisor: Ing. Ondřej Lengál, Ph.D

Why quantum simulation?

Even though the history of quantum computing dates back to the 1980s, it is useful to simulate the behavior of quantum circuits on classical computers as the hardware still has a largely experimental character. Our implementation of a quantum simulator, **MEDUSA**, converts input quantum circuits specified in OpenQASM (Open Quantum Assembly Language) and computes its end state represented as an MTBDD.

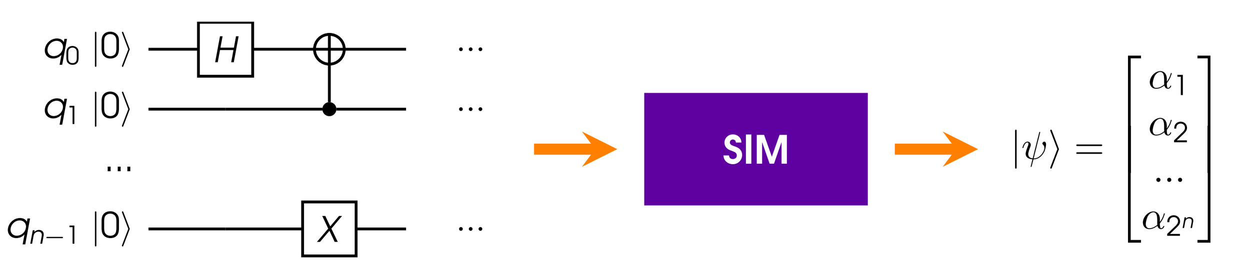


Figure 1: The aim of quantum simulation is to calculate the final state vector of the system

MTBDDs

MTBDDs (***Multi-terminal binary decision diagrams***) encode functions $f(v_1,\dots,v_n)\colon \{0,1\}^n \to \mathbb{D}$. They are a generalised variant of ROBDDs (Reduced ordered binary decision diagram), usually simply called BDDs, where terminals can have an arbitrary value.

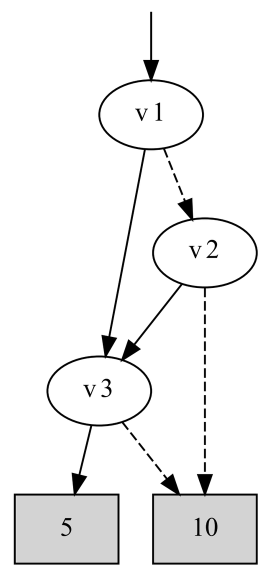


Figure 2: Example MTBDD (a dashed edge denotes a low successor, a solid edge denotes a high successor)

MTBDD-based quantum circuit representation

- Algebraic representation of complex numbers:

$$(1)\quad z = \left(\frac{1}{\sqrt{2}}\right)^k \cdot (a + b\omega + c\omega^2 + d\omega^3),$$

where $a, b, c, d, k \in \mathbb{Z}$, $z \in \mathbb{C}$ and $\omega = e^{\frac{i\pi}{4}}$ [1, 2].

- A circuit's state vector is represented with an MTBDD such that the leaves contain the algebraic representation of the probability amplitudes.
- Gate operations are applied either as universal or as permutation-based update formulae for this MTBDD.
  - ***Universal update formulae*** [1] – defined using the following operators on the function $T(b_{n-1}, \dots, b_0)\colon \{0,1\}^n \to \mathbb{C}$ represented by the system's state's MTBDD:

$$(2)\quad T_{q_t}(b_{n-1},\dots,b_t,\dots,b_0) = T(b_{n-1},\dots,1,\dots,b_0)$$
$$(3)\quad T_{\overline{q_t}}(b_{n-1},\dots,b_t,\dots,b_0) = T(b_{n-1},\dots,0,\dots,b_0)$$
$$(4)\quad B_{q_t}(b_{n-1},\dots,b_0) = b_t$$
$$(5)\quad B_{\overline{q_t}}(b_{n-1},\dots,b_0) = \overline{b_t}$$

Table 1: Universal update formulae (target qubits are denoted as q_t, q'_t, control qubits are denoted as q_c, q'_c if the gate uses them)

Gate	Update formula
$X[q_t]$	$B_{q_t} \cdot T_{\overline{q_t}} + B_{\overline{q_t}} \cdot T_{q_t}$
$Y[q_t]$	$\omega^2 \cdot (B_{q_t} \cdot T_{\overline{q_t}} - B_{\overline{q_t}} \cdot T_{q_t})$
$Z[q_t]$	$B_{\overline{q_t}} \cdot T - B_{q_t} \cdot T$
$H[q_t]$	$\frac{1}{\sqrt{2}} \cdot (T_{\overline{q_t}} + B_{\overline{q_t}} \cdot T_{q_t} - B_{q_t} \cdot T)$
$S[q_t]$	$B_{\overline{q_t}} \cdot T + \omega^2 \cdot B_{q_t} \cdot T$
$T[q_t]$	$B_{\overline{q_t}} \cdot T + \omega \cdot B_{q_t} \cdot T$
$Rx(\frac{\pi}{2})\,[q_t]$	$\frac{1}{\sqrt{2}} \cdot (T - \omega^2 \cdot (B_{q_t} \cdot T_{\overline{q_t}} + B_{\overline{q_t}} \cdot T_{q_t}))$
$Ry(\frac{\pi}{2})\,[q_t]$	$\frac{1}{\sqrt{2}} \cdot (T_{\overline{q_t}} + B_{q_t} \cdot T - B_{\overline{q_t}} \cdot T_{q_t})$
$CNOT[q_c, q_t]$	$B_{\overline{q_c}} \cdot T + B_{q_c} \cdot B_{\overline{q_t}} \cdot T_{q_t} + B_{q_c} \cdot B_{q_t} \cdot T_{\overline{q_t}}$
$CZ[q_c, q_t]$	$B_{\overline{q_c}} \cdot T + B_{\overline{q_t}} \cdot T - B_{\overline{q_c}} \cdot B_{\overline{q_t}} \cdot T - B_{q_c} \cdot B_{q_t} \cdot T$
$Toffoli[q_c, q'_c, q_t]$	$B_{\overline{q_c}} \cdot T + B_{\overline{q_{c'}}} \cdot T - B_{\overline{q_c}} \cdot B_{\overline{q_{c'}}} \cdot T + B_{q_t} \cdot B_{q_c} \cdot B_{q_{c'}} \cdot T_{\overline{q_t}} + B_{\overline{q_t}} \cdot B_{q_c} \cdot B_{q_{c'}} \cdot T_{q_t}$
$Fredkin[q_c, q_t, q'_t]$	$B_{\overline{q_c}} \cdot T + B_{q_t} \cdot B_{q_{t'}} \cdot B_{q_t} \cdot T + B_{\overline{q_t}} \cdot B_{\overline{q_{t'}}} \cdot B_{q_c} \cdot T + B_{q_t} \cdot B_{\overline{q_{t'}}} \cdot B_{q_c} \cdot T_{\overline{q_t} q_{t'}} + B_{\overline{q_t}} \cdot B_{q_{t'}} \cdot B_{q_c} \cdot T_{q_t \overline{q_{t'}}}$

  - ***Permutation-based update formulae*** – cannot be used universally with every gate but are less computationally demanding

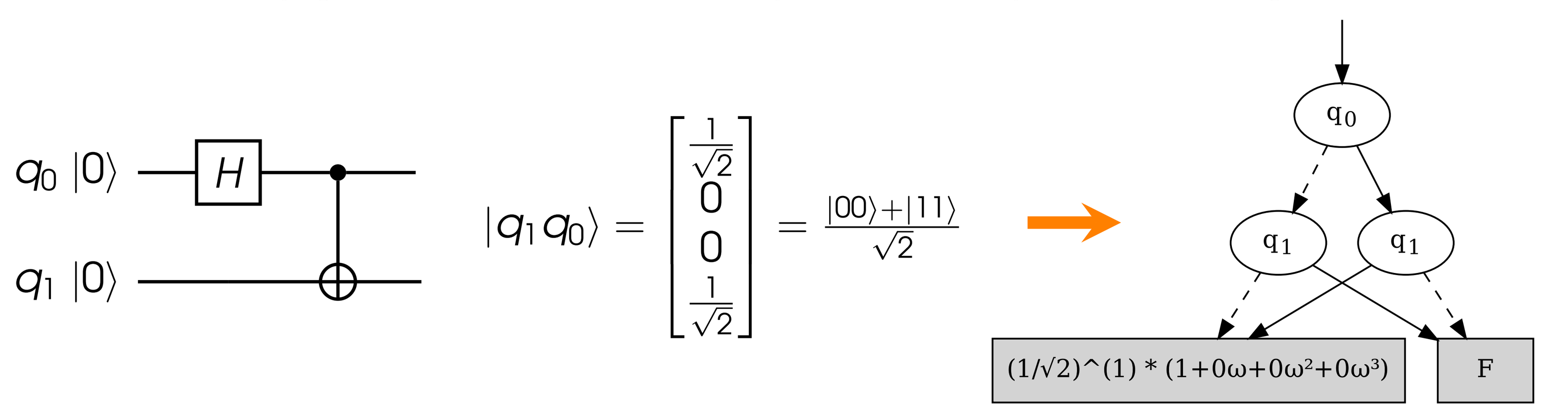


Figure 3: Circuit to create the Bell and its resulting MTBDD (leaf value F denotes zero probability amplitude)

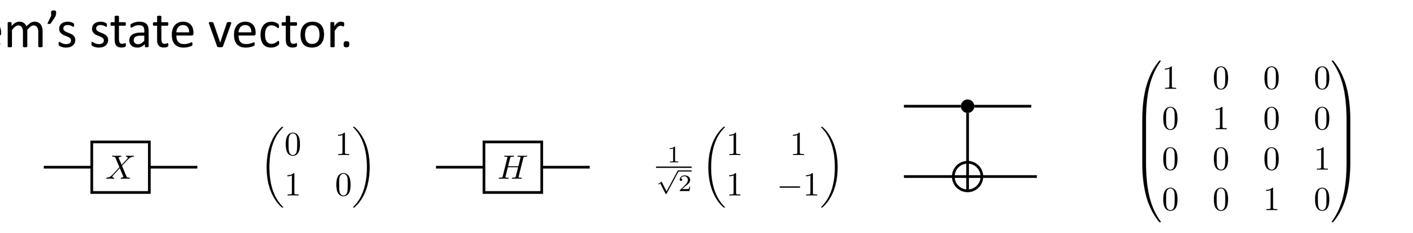
Quantum circuits

Quantum state. A qubit's state $|\psi\rangle$ can be in a superposition of the computational basis states $|0\rangle$ and $|1\rangle$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$ are the ***probability amplitudes*** for the respective basis states. I.e., it is a two-dimensional unit complex vector representing the probabilities that upon measurement its value will be $|0\rangle$ or $|1\rangle$.

Quantum gates are used to perform operations on qubits and hence alter the system's quantum state. They can be represented as unitary matrices, then the update of the system's quantum state is carried out as a matrix multiplication of the (dimension-wise modified) gate matrix with the system's state vector.



(a) Pauli-X gate (a NOT gate equivalent) (b) Hadamard gate (c) CNOT (Controlled-NOT) gate

Figure 4: Some example gates and their matrix representation

Experiments

The experiments consisted of comparing our implemented quantum simulator MEDUSA to the current BDD-based state-of-the-art simulator SliQSim [2] by measuring their respective runtimes for various quantum circuits – e.g., circuits implementing Grover's search algorithm (both single and multi-oracle) or multi-controlled Toffoli gates.

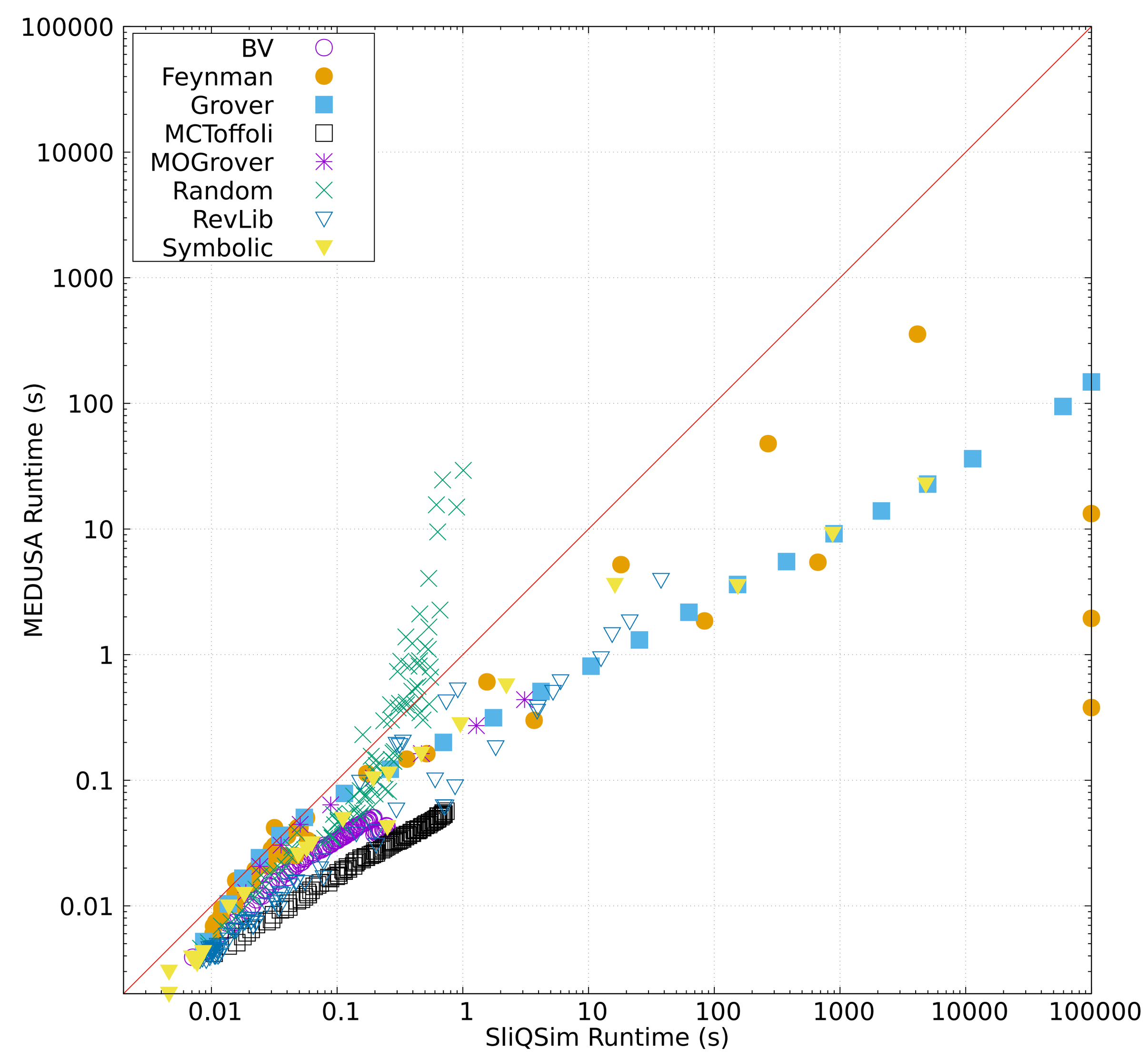


Figure 5: Results for all the benchmark circuits

The used time-out was set to 100 000 seconds (27 hours 46 min 40 s, wall-clock time). Failed tests (either due to the time-out limit or an error) have the according coordinate set to the maximum possible runtime. The axes are logarithmic.

Table 2: Benchmark results for a few of the most complex circuits (#Q and #Gates denote the number of circuit's qubits and gates respectively)

(a) Grover

Benchmark	#Q	#Gates	Runtime (s)	
			MEDUSA	SliQSim
19	38	95 445	13.94	2135
20	40	141 526	22.77	4983
21	42	211 505	36.27	11380
23	46	463 921	94.58	59390
24	48	681 818	148.3	Timed-out

(b) Feynman

Benchmark	#Q	#Gates	Runtime (s)	
			MEDUSA	SliQSim
hwb11	15	87 789	5.438	667.4
hwb12	20	171 482	13.27	2675
gf2^64_mult	192	12 731	5.205	18.13
gf2^128_mult	384	50 043	47.83	268.3
gf2^256_mult	768	198 395	356.0	4132

References

[1] Chen, Y.-F., Chung, K.-M., Lengál, O., Lin, J.-A., Tsai, W.-L. et al. An Automata-based Framework for Verification and Bug Hunting in Quantum Circuits (Technical Report). 2023. Available at https://arxiv.org/abs/2301.07747.

[2] Tsai, Y.-H., Jiang, J.-H. R. and Jhang, C.-S. Bit-Slicing the Hilbert Space: Scaling Up Accurate Quantum Circuit Simulation. In:2021 58th ACM/IEEE Design Automation Conference (DAC). 2021, p. 439-444. DOI: 10.1109/DAC18074.2021.9586191.