CLASSIQ

CLASSIQ.IO

CLASSIQ

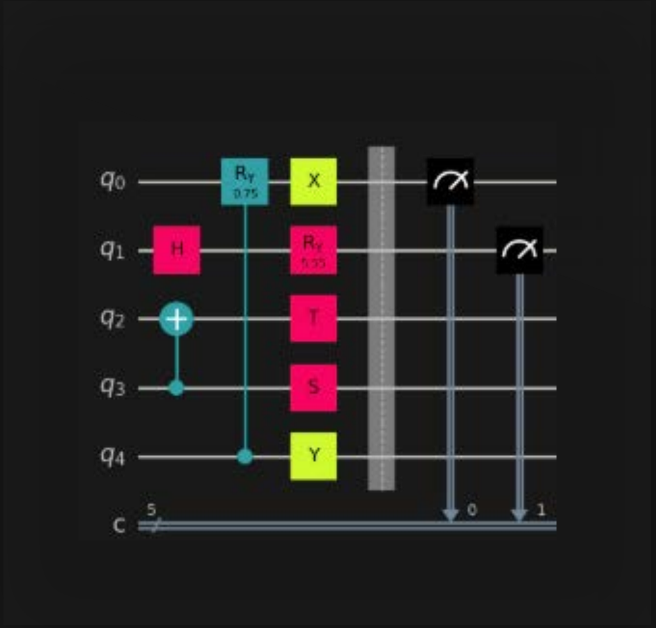
Quantum Computing Software

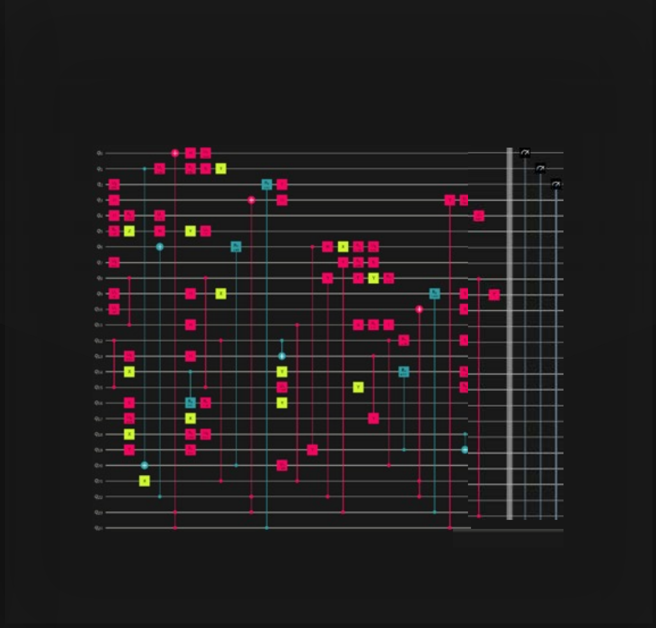
ABOUT CLASSIQ

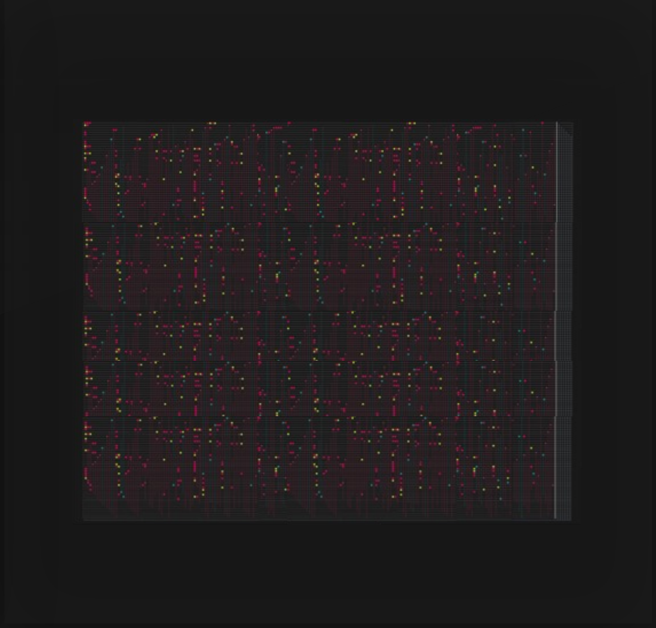
- The leader in quantum computing software
- $63M in funding
- A team of 65 world class experts
- Based in Israel, US, Europe, and Japan
- A line of Fortune-500 enterprise customers and partners
- Used in research and education

GATE LEVEL DESIGN DOES NOT SCALE

It is impossible to design complex quantum circuits
using today's development methods







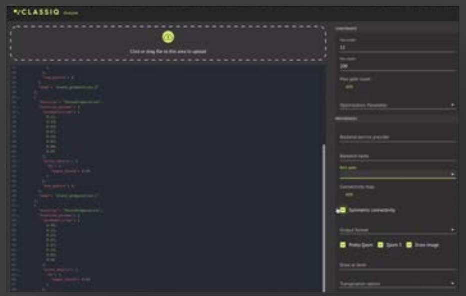
CLASSIQ : THE GATEWAY TO QUANTUM

The Classiq Platform

Design, Debug, Optimize

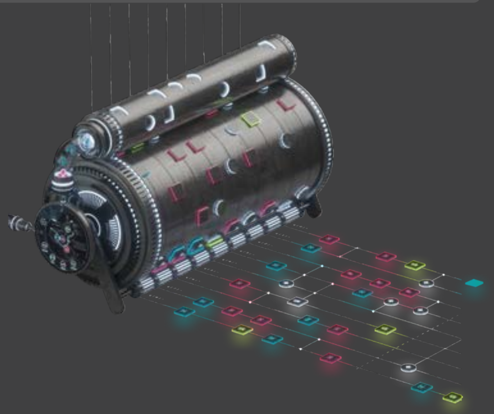
Model → **Compiler** → **Visualizer** →

Model



High-level design of quantum algorithms

Compiler



Synthesize the optimal quantum circuit according to the available resources

Visualizer

Interactive application for circuit visualization

Execute

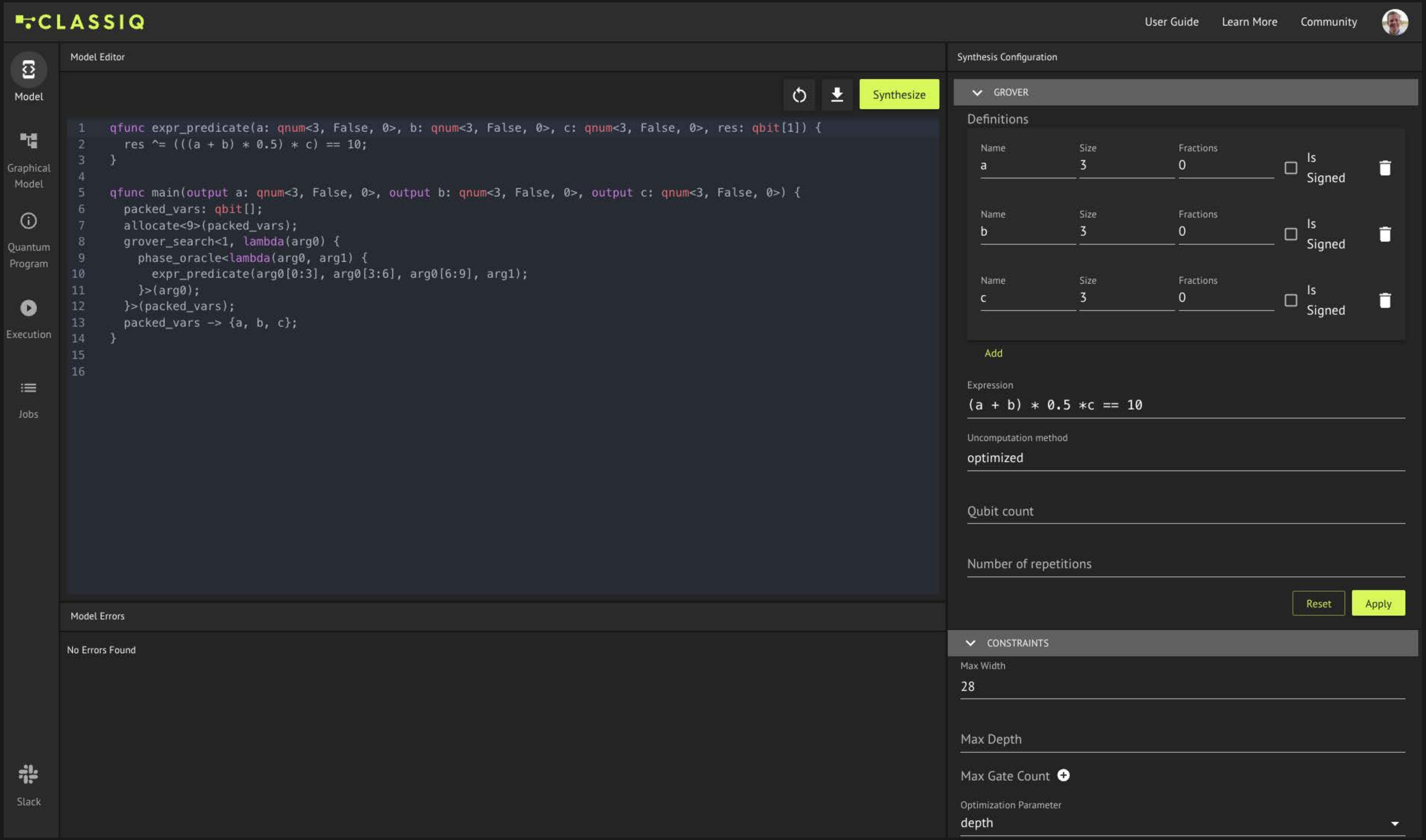
OS

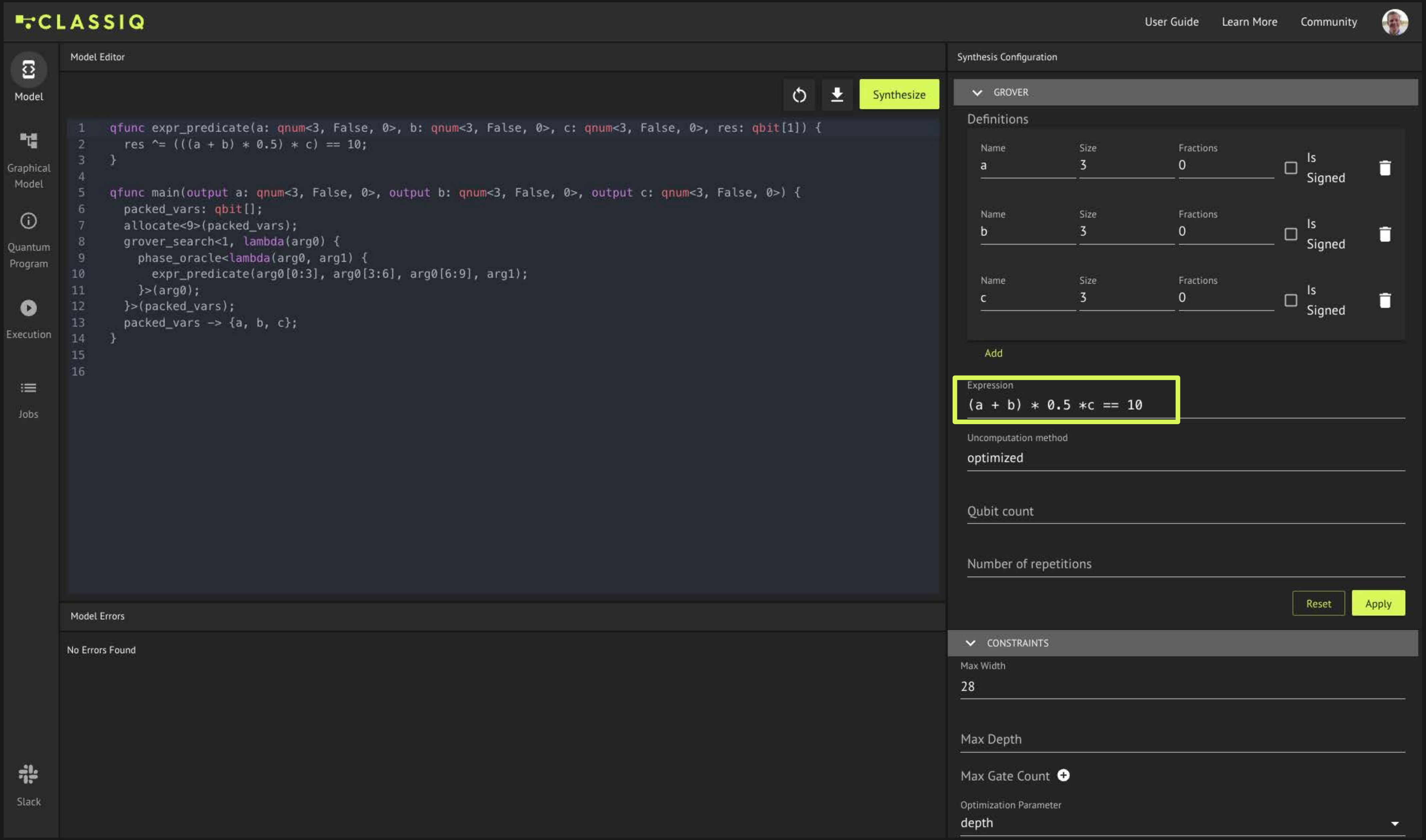
- Azure Quantum
- AWS Braket
- IBM
- Any universal gate-based computer

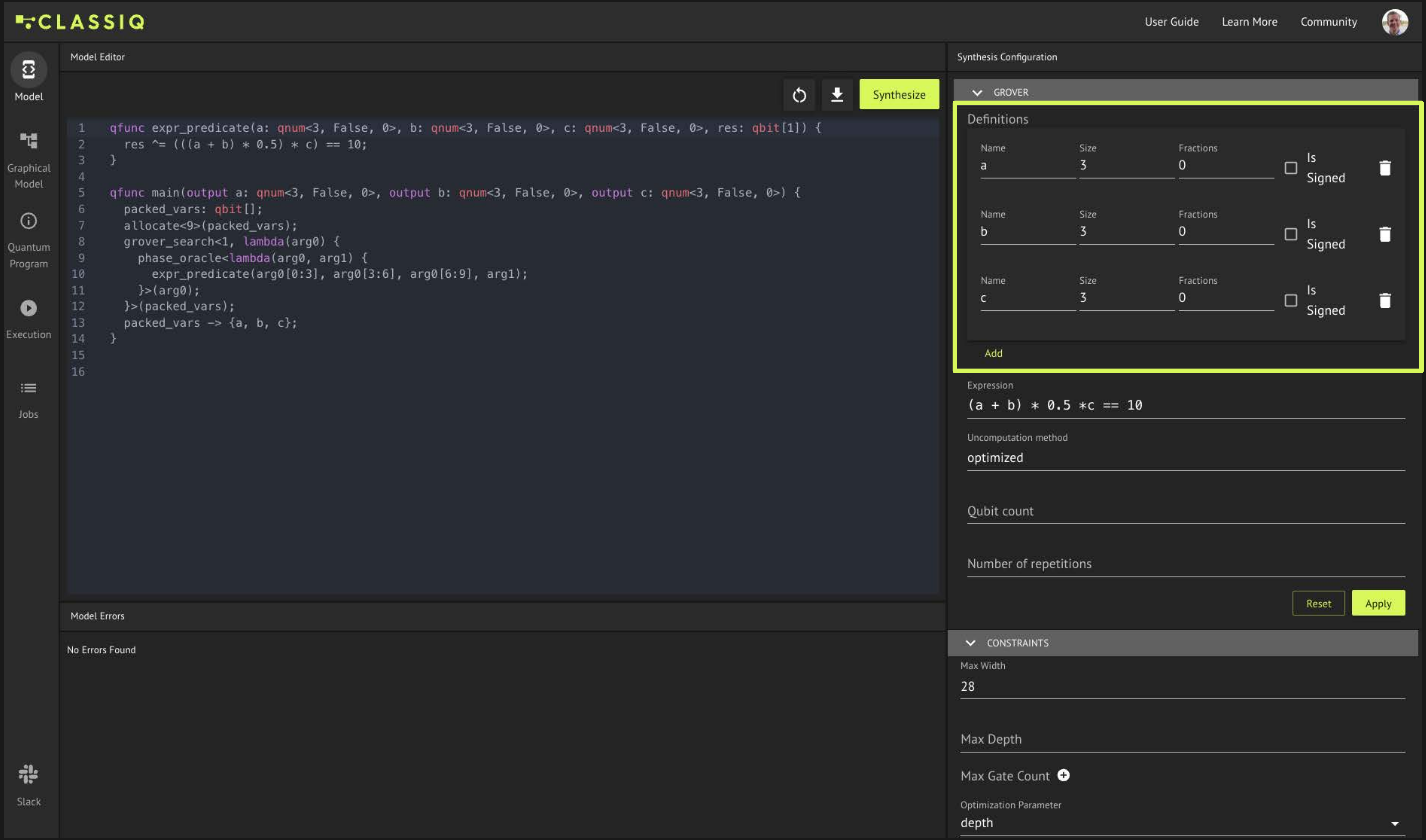
Easy execution of quantum programs on all major quantum hardware vendors

CLASSIQ

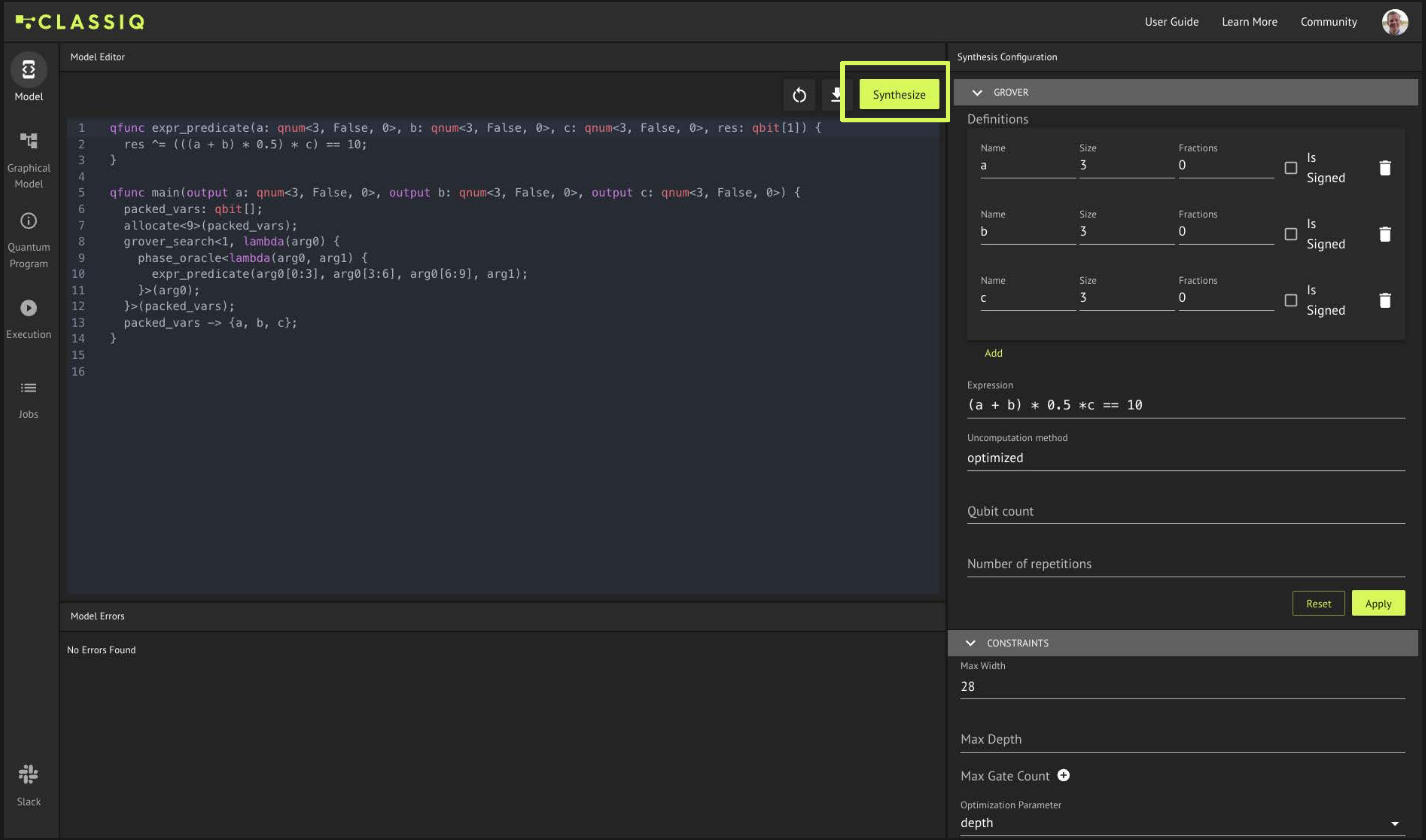
Grover demo

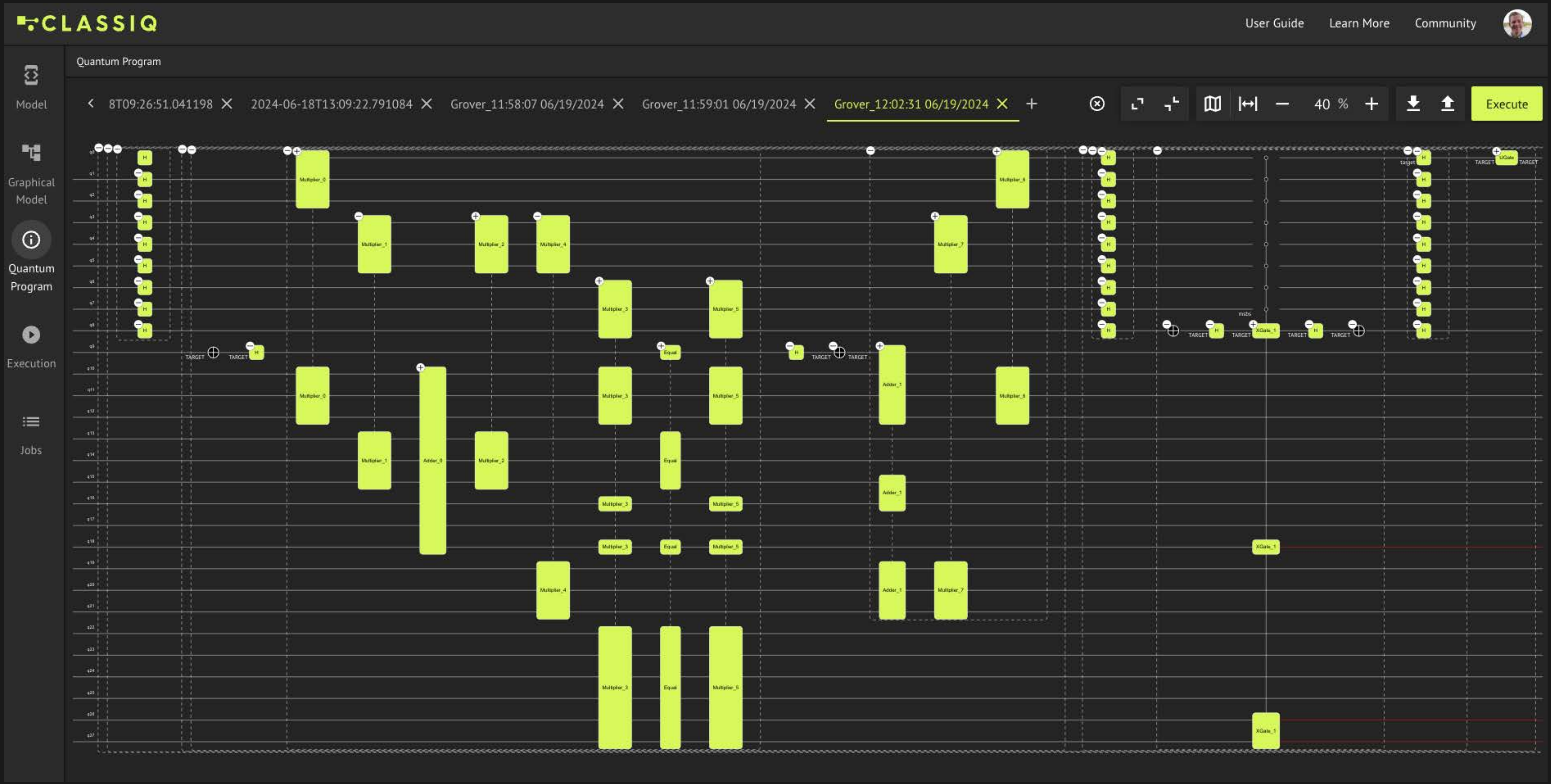

CLASSIQ
User Guide
Learn More
Community
Model
Graphical Model
Quantum Program
Execution
Jobs
Slack
Model Editor
Synthesize
qfunc expr_predicate(a: qnum<3, False, 0>, b: qnum<3, False, 0>, c: qnum<3, False, 0>, res: qbit[1]) {
res ^= (((a + b) * 0.5) * c) == 10;
}
qfunc main(output a: qnum<3, False, 0>, output b: qnum<3, False, 0>, output c: qnum<3, False, 0>) {
packed_vars: qbit[];
allocate<9>(packed_vars);
grover_search<1, lambda(arg0) {
phase_oracle<lambda(arg0, arg1) {
expr_predicate(arg0[0:3], arg0[3:6], arg0[6:9], arg1);
}>(arg0);
}>(packed_vars);
packed_vars -> {a, b, c};
}
Model Errors
No Errors Found
Synthesis Configuration
GROVER
Definitions
Name a Size 3 Fractions 0 Is Signed
Name b Size 3 Fractions 0 Is Signed
Name c Size 3 Fractions 0 Is Signed
Add
Expression
(a + b) * 0.5 *c == 10
Uncomputation method
optimized
Qubit count
Number of repetitions
Reset
Apply
CONSTRAINTS
Max Width
28
Max Depth
Max Gate Count
Optimization Parameter
depth

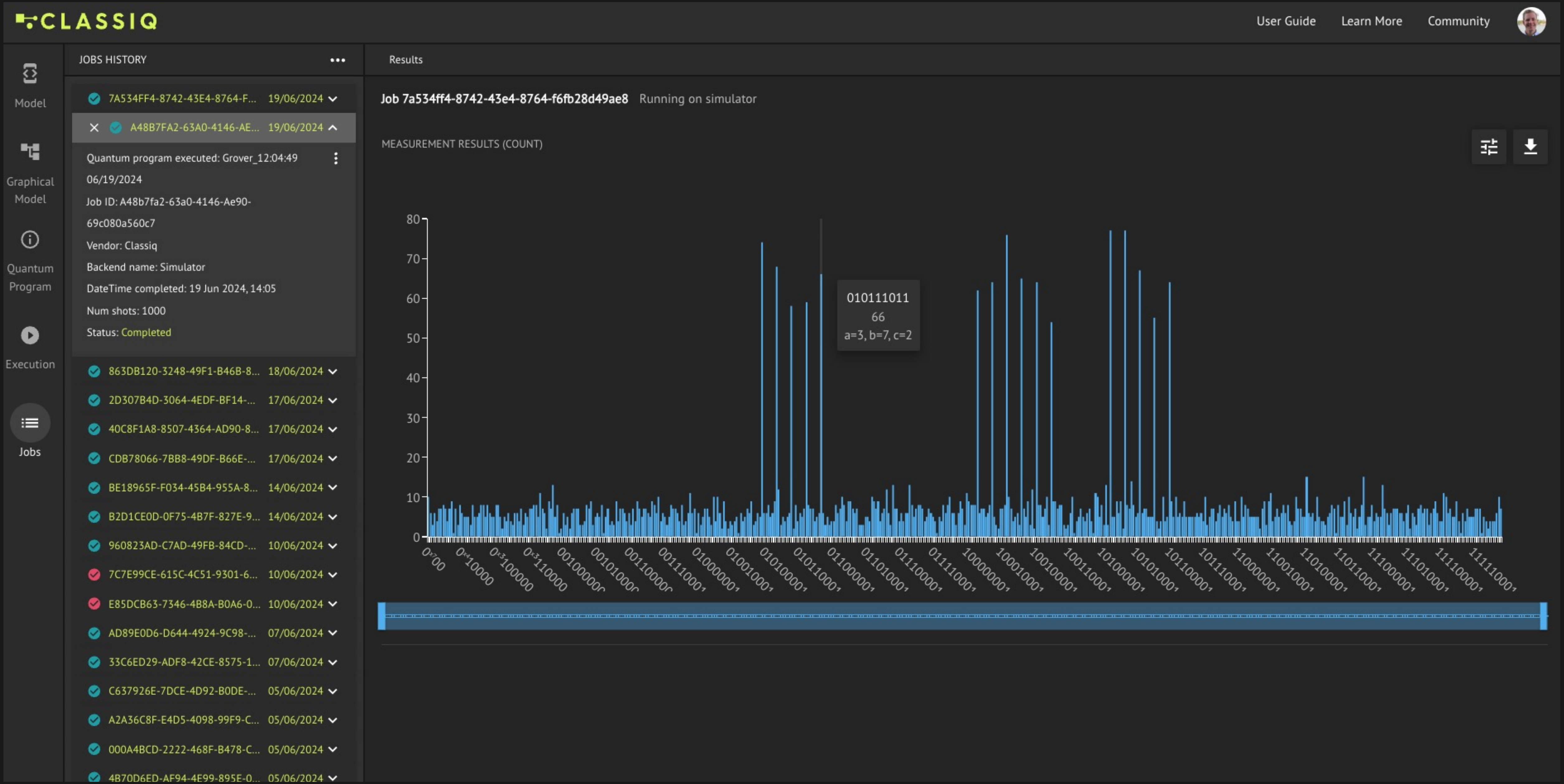

CLASSIQ
User Guide
Learn More
Community
Model
Graphical Model
Quantum Program
Execution
Jobs
Slack
Model Editor
Synthesize
qfunc expr_predicate(a: qnum<3, False, 0>, b: qnum<3, False, 0>, c: qnum<3, False, 0>, res: qbit[1]) {
res ^= (((a + b) * 0.5) * c) == 10;
}
qfunc main(output a: qnum<3, False, 0>, output b: qnum<3, False, 0>, output c: qnum<3, False, 0>) {
packed_vars: qbit[];
allocate<9>(packed_vars);
grover_search<1, lambda(arg0) {
phase_oracle<lambda(arg0, arg1) {
expr_predicate(arg0[0:3], arg0[3:6], arg0[6:9], arg1);
}>(arg0);
}>(packed_vars);
packed_vars -> {a, b, c};
}
Model Errors
No Errors Found
Synthesis Configuration
GROVER
Definitions
Name a Size 3 Fractions 0 Is Signed
Name b Size 3 Fractions 0 Is Signed
Name c Size 3 Fractions 0 Is Signed
Add
Expression
(a + b) * 0.5 *c == 10
Uncomputation method
optimized
Qubit count
Number of repetitions
Reset
Apply
CONSTRAINTS
Max Width
28
Max Depth
Max Gate Count
Optimization Parameter
depth


CLASSIQ
User Guide
Learn More
Community
Model
Graphical Model
Quantum Program
Execution
Jobs
Slack
Model Editor
Synthesize
qfunc expr_predicate(a: qnum<3, False, 0>, b: qnum<3, False, 0>, c: qnum<3, False, 0>, res: qbit[1]) {
res ^= (((a + b) * 0.5) * c) == 10;
}
qfunc main(output a: qnum<3, False, 0>, output b: qnum<3, False, 0>, output c: qnum<3, False, 0>) {
packed_vars: qbit[];
allocate<9>(packed_vars);
grover_search<1, lambda(arg0) {
phase_oracle<lambda(arg0, arg1) {
expr_predicate(arg0[0:3], arg0[3:6], arg0[6:9], arg1);
}>(arg0);
}>(packed_vars);
packed_vars -> {a, b, c};
}
Model Errors
No Errors Found
Synthesis Configuration
GROVER
Definitions
Name a Size 3 Fractions 0 Is Signed
Name b Size 3 Fractions 0 Is Signed
Name c Size 3 Fractions 0 Is Signed
Add
Expression
(a + b) * 0.5 *c == 10
Uncomputation method
optimized
Qubit count
Number of repetitions
Reset
Apply
CONSTRAINTS
Max Width
28
Max Depth
Max Gate Count
Optimization Parameter
depth

CLASSIQ


CLASSIQ
User Guide
Learn More
Community
Model
Graphical Model
Quantum Program
Execution
Jobs
Slack
Model Editor
Synthesize
qfunc expr_predicate(a: qnum<3, False, 0>, b: qnum<3, False, 0>, c: qnum<3, False, 0>, res: qbit[1]) {
res ^= (((a + b) * 0.5) * c) == 10;
}
qfunc main(output a: qnum<3, False, 0>, output b: qnum<3, False, 0>, output c: qnum<3, False, 0>) {
packed_vars: qbit[];
allocate<9>(packed_vars);
grover_search<1, lambda(arg0) {
phase_oracle<lambda(arg0, arg1) {
expr_predicate(arg0[0:3], arg0[3:6], arg0[6:9], arg1);
}>(arg0);
}>(packed_vars);
packed_vars -> {a, b, c};
}
Model Errors
No Errors Found
Synthesis Configuration
GROVER
Definitions
Name
a
Size
3
Fractions
0
Is Signed
Name
b
Size
3
Fractions
0
Is Signed
Name
c
Size
3
Fractions
0
Is Signed
Add
Expression
(a + b) * 0.5 *c == 10
Uncomputation method
optimized
Qubit count
Number of repetitions
Reset
Apply
CONSTRAINTS
Max Width
28
Max Depth
Max Gate Count
Optimization Parameter
depth


CLASSIQ
User Guide
Learn More
Community
Quantum Program
Model
Graphical Model
Quantum Program
Execution
Jobs
8T09:26:51.041198
2024-06-18T13:09:22.791084
Grover_11:58:07 06/19/2024
Grover_11:59:01 06/19/2024
Grover_12:02:31 06/19/2024
40 %
Execute

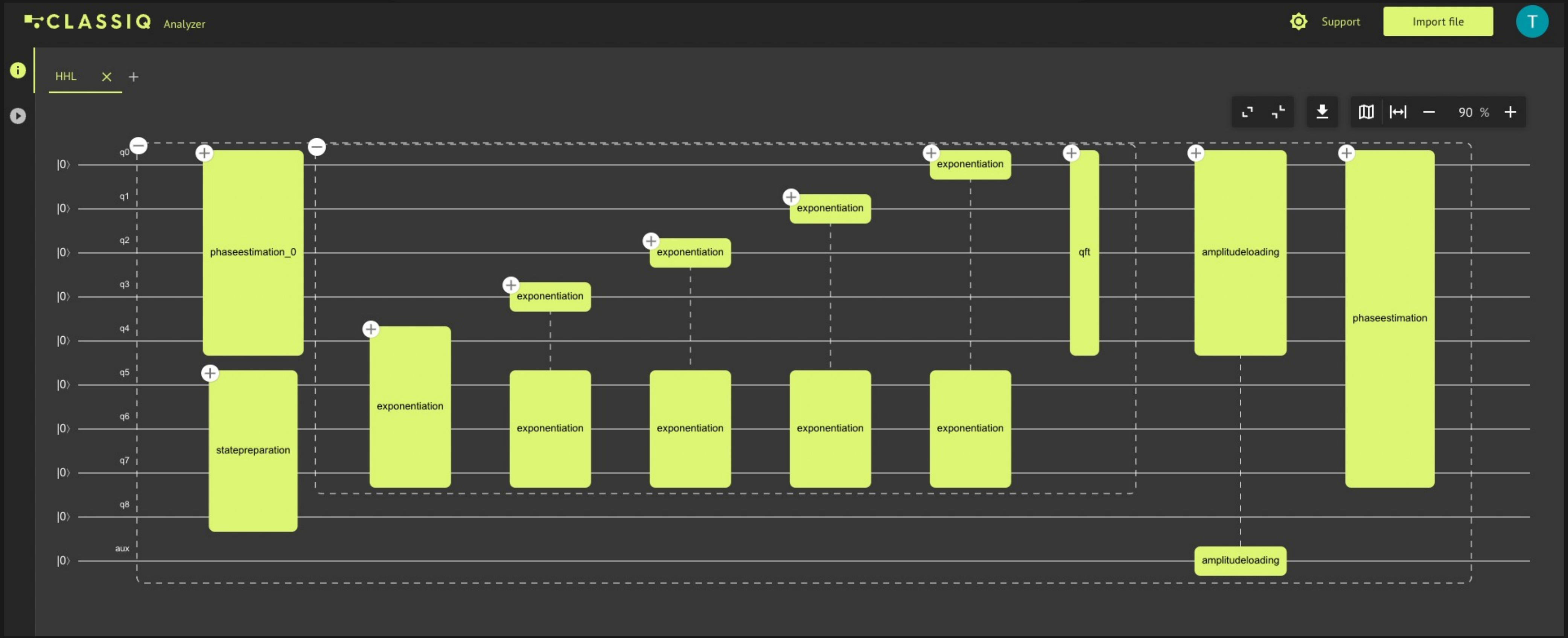

CLASSIQ
User Guide
Learn More
Community
Model
Graphical Model
Quantum Program
Execution
Jobs
JOBS HISTORY
7A534FF4-8742-43E4-8764-F... 19/06/2024
A48B7FA2-63A0-4146-AE... 19/06/2024
Quantum program executed: Grover_12:04:49
06/19/2024
Job ID: A48b7fa2-63a0-4146-Ae90-69c080a560c7
Vendor: Classiq
Backend name: Simulator
DateTime completed: 19 Jun 2024, 14:05
Num shots: 1000
Status: Completed
863DB120-3248-49F1-B46B-8... 18/06/2024
2D307B4D-3064-4EDF-BF14-... 17/06/2024
40C8F1A8-8507-4364-AD90-8... 17/06/2024
CDB78066-7BB8-49DF-B66E-... 17/06/2024
BE18965F-F034-45B4-955A-8... 14/06/2024
B2D1CE0D-0F75-4B7F-827E-9... 14/06/2024
960823AD-C7AD-49FB-84CD-... 10/06/2024
7C7E99CE-615C-4C51-9301-6... 10/06/2024
E85DCB63-7346-4B8A-B0A6-0... 10/06/2024
AD89E0D6-D644-4924-9C98-... 07/06/2024
33C6ED29-ADF8-42CE-8575-1... 07/06/2024
C637926E-7DCE-4D92-B0DE-... 05/06/2024
A2A36C8F-E4D5-4098-99F9-C... 05/06/2024
000A4BCD-2222-468F-B478-C... 05/06/2024
4B70D6ED-AF94-4E99-895E-0... 05/06/2024
Results
Job 7a534ff4-8742-43e4-8764-f6fb28d49ae8 Running on simulator
MEASUREMENT RESULTS (COUNT)
80
70
60
50
40
30
20
10
0
010111011
66
a=3, b=7, c=2

Classiq use cases

CLASSIQ

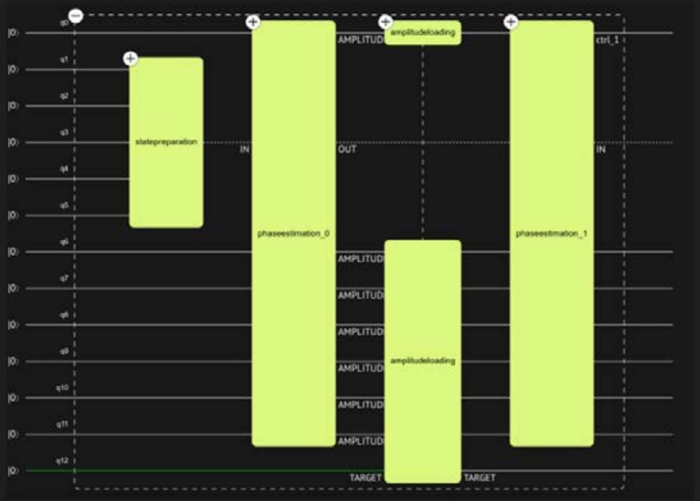
INDUSTRIAL RESEARCH

HHL Based Computational Fluid Dynamics



LARGEST EVER SIMULATION WITH NVIDIA CU-QUANTUM

39 Qubits, 10,000,000 gates



An HHL circuit generated by the Classiq platform. This circuit is built using four quantum function blocks. The accuracy of this quantum linear solver depends on the depth and number of extra qubits employed in the Quantum Phase Estimation block.

NVIDIA
1,397,488 followers
1w •
+ Follow

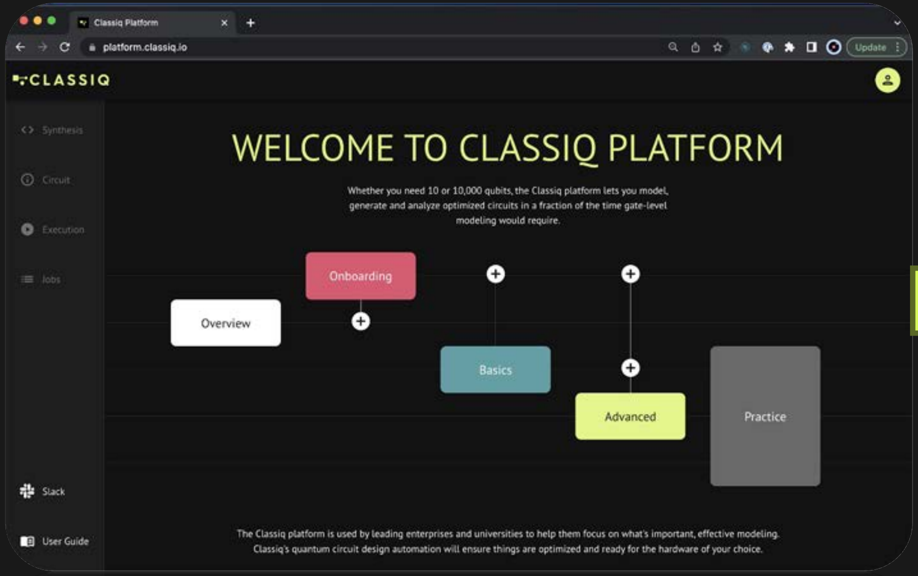
We announced with Rolls-Royce and Classiq Technologies a major breakthrough for computational fluid dynamics, advancing the development of #quantumcomputing in aerospace. #ISC23

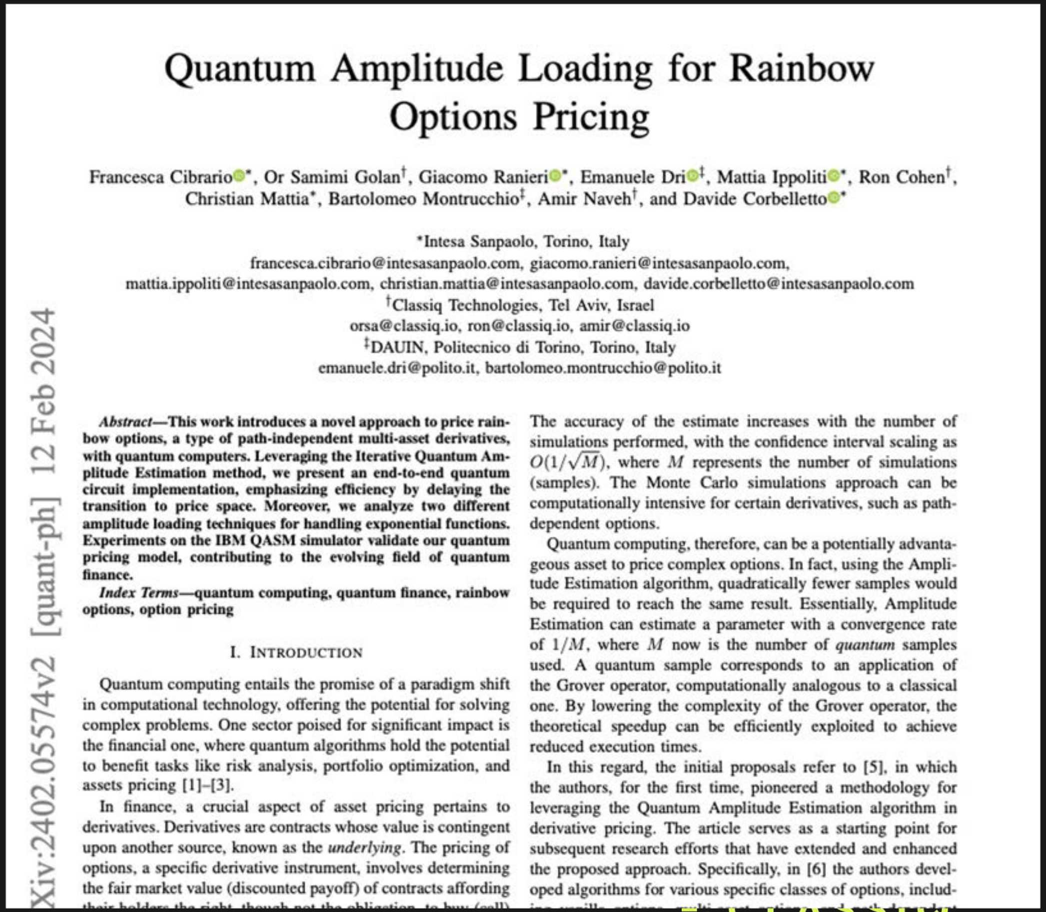
NVIDIA, Rolls-Royce and Classiq Announce Quantum Computing Breakthrough for Computational Fluid Dynamics in Jet Engines

nvidianews.nvidia.com • 3 min read

FROM ZERO TO ADVANCED AND CAPABLE TEAM IN 1 YEAR

Team Building





Quantum Amplitude Loading for Rainbow Options Pricing

Francesca Cibrario*, Or Samimi Golan†, Giacomo Ranieri*, Emanuele Dri‡, Mattia Ippoliti*, Ron Cohen†, Christian Mattia*, Bartolomeo Montrucchio‡, Amir Naveh†, and Davide Corbelletto*

*Intesa Sanpaolo, Torino, Italy
francesca.cibrario@intesasanpaolo.com, giacomo.ranieri@intesasanpaolo.com, mattia.ippoliti@intesasanpaolo.com, christian.mattia@intesasanpaolo.com, davide.corbelletto@intesasanpaolo.com
†Classiq Technologies, Tel Aviv, Israel
orsa@classiq.io, ron@classiq.io, amir@classiq.io
‡DAUIN, Politecnico di Torino, Torino, Italy
emanuele.dri@polito.it, bartolomeo.montrucchio@polito.it

arXiv:2402.05574v2 [quant-ph] 12 Feb 2024

***Abstract*—This work introduces a novel approach to price rainbow options, a type of path-independent multi-asset derivatives, with quantum computers. Leveraging the Iterative Quantum Amplitude Estimation method, we present an end-to-end quantum circuit implementation, emphasizing efficiency by delaying the transition to price space. Moreover, we analyze two different amplitude loading techniques for handling exponential functions. Experiments on the IBM QASM simulator validate our quantum pricing model, contributing to the evolving field of quantum finance.**

***Index Terms*—quantum computing, quantum finance, rainbow options, option pricing**

I. Introduction

Quantum computing entails the promise of a paradigm shift in computational technology, offering the potential for solving complex problems. One sector poised for significant impact is the financial one, where quantum algorithms hold the potential to benefit tasks like risk analysis, portfolio optimization, and assets pricing [1]–[3].

In finance, a crucial aspect of asset pricing pertains to derivatives. Derivatives are contracts whose value is contingent upon another source, known as the *underlying*. The pricing of options, a specific derivative instrument, involves determining the fair market value (discounted payoff) of contracts affording

The accuracy of the estimate increases with the number of simulations performed, with the confidence interval scaling as $O(1/\sqrt{M})$, where M represents the number of simulations (samples). The Monte Carlo simulations approach can be computationally intensive for certain derivatives, such as path-dependent options.

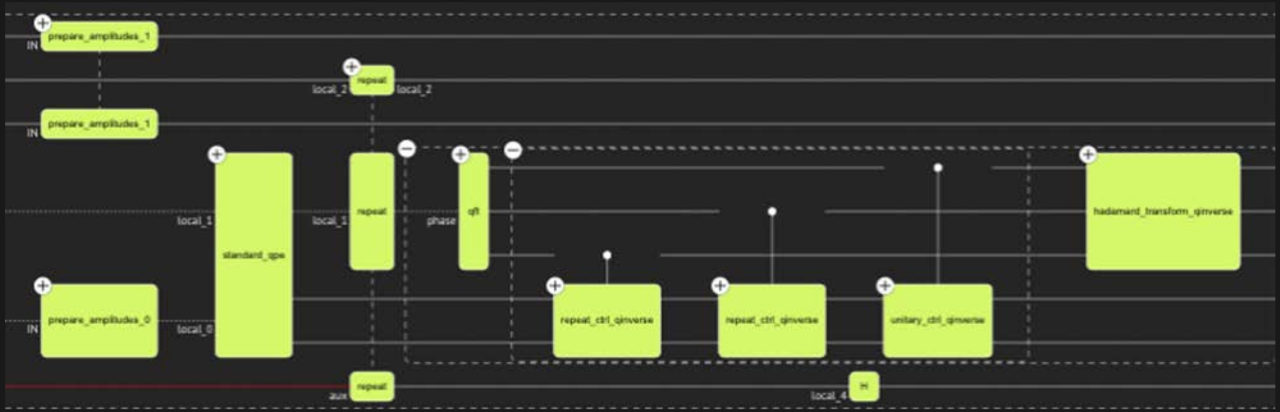
Quantum computing, therefore, can be a potentially advantageous asset to price complex options. In fact, using the Amplitude Estimation algorithm, quadratically fewer samples would be required to reach the same result. Essentially, Amplitude Estimation can estimate a parameter with a convergence rate of $1/M$, where M now is the number of *quantum* samples used. A quantum sample corresponds to an application of the Grover operator, computationally analogous to a classical one. By lowering the complexity of the Grover operator, the theoretical speedup can be efficiently exploited to achieve reduced execution times.

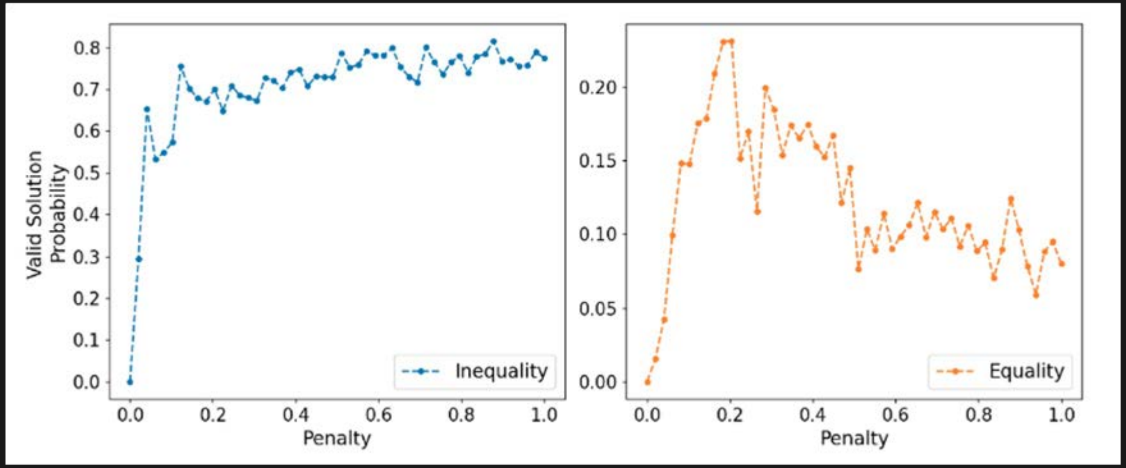
In this regard, the initial proposals refer to [5], in which the authors, for the first time, pioneered a methodology for leveraging the Quantum Amplitude Estimation algorithm in derivative pricing. The article serves as a starting point for subsequent research efforts that have extended and enhanced the proposed approach. Specifically, in [6] the authors developed algorithms for various specific classes of options, includ-

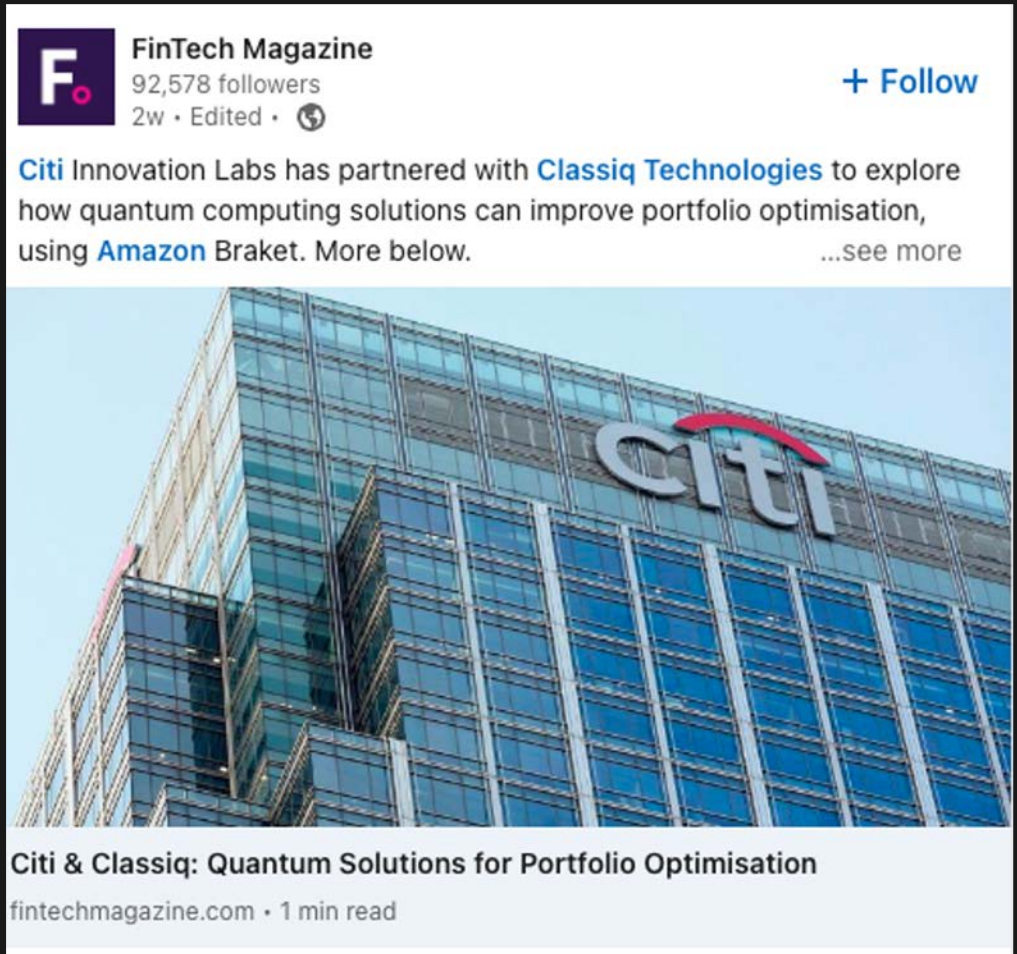
CLASSIQ

QUANTUM CAPABILITY BUILDING

Portfolio Optimization







CLASSIQ

THANK YOU

CLASSIQ.IO