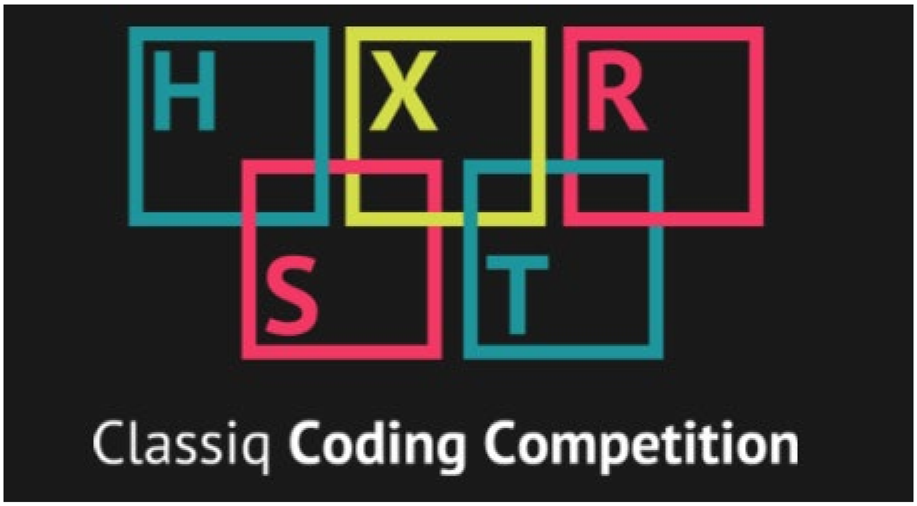
The Quantum Leap
May 24, 2022

Why I'm Excited About the Classiq Coding Competition, and Why You Should be Too - Even if You Don't Understand Quantum Programming



In a prior post entitled "The Case for an Annual 'Quantum Games' Competition", I described how the amount of innovation and technical advancement in Quantum Computing (QC) has been incredible over the past 12 months or so, but how challenging it is to compare machine performance. Should we focus on who has the "most qubits"? [hint, the answer is no] Or highest "quantum volume"? Or which can run the longest before decoherence? Or should we focus on #AQ as IonQ has suggested? How about SupermarQ or QED-C proposed alternative benchmarks?

So, I suggested an annual "Quantum Games" or world Olympics to spur innovation and friendly competition. I volunteered to be a judge so that I could enjoy a front-row seat to watch the competitors give their best to the challenges. I was thrilled that Classiq has since created the "Classiq Coding Competition" and honored to be a judge in this recently announced competition. Let me explain why I'm so excited, but first some more details on the Competition.

Classiq's Worldwide Competition – a $25,000 Challenge to Build the Best Quantum Circuits

As Nir Minerbi, the Classiq CEO noted in the Contest announcement, "Creating efficient quantum algorithms is part engineering, part art. The Classiq Coding Competition is a call to the world's quantum software community to showcase their talents and demonstrate how quantum computing can take humans to new heights. Efficient circuits enhance the ability of any quantum computer to solve important problems."

Minerbi went on to add "You would be surprised how much can be achieved with compact, efficient circuits. The onboard computer used in the Apollo 11 space mission got a man to the moon using just 72 kilobytes of ROM. Quantum computing is taking off, and the need to create elegant and efficient quantum algorithms will exist for years to come. Organizations that manage to fit larger problems into available computers will reap their quantum benefits sooner than others. The Classiq Coding Competition will encourage the creativity and ingenuity required to make this happen and **highlight the art of the possible** (emphasis added) in compact, efficient circuits."

The Competition includes four problems or challenges briefly described below:

Problem	General Description	Success Metric
Log-Normal State Preparation	Many quantum algorithms rely on initializing qubits in a specific state. The promised speedup of the algorithm depends on the ability to prepare the quantum state efficiently.	Shortest circuit depth that provides an error below 0.01
Kakuro – A Constraint Satisfaction Problem	Kakuro is a logic puzzle played on a grid of cells. The challenge is to solve the puzzle using Grover's algorithm.	Minimized number of 2-qubit CX gates
Decomposing a Multi-Controlled Toffoli Gate	Decompose an MCX gate with 14 control qubits, one target qubit and up to five auxiliary qubits.	Shortest circuit depth
Hamiltonian Simulation	Describes the evolution of molecules and solid-state systems by solving the Schrodinger equation. Quantum computers enable such simulation in a scalable manner.	Using the CX and single qubit gates only with up to 10 total qubits, shortest circuit depth and an error below 0.1

While the descriptions in this chart will not resonate with readers unfamiliar with quantum algorithm construction, suffice it to say that it is a disparate and varied set of challenges with many possible solutions, which is part of the reason this is such an interesting challenge.

Here are a few more highlights of the Competition:

- **Submission Deadline**: June 5, 2022 (by midnight US Eastern time)
- **Cash prizes of $25,000**: with $3,000 to Gold medal, $1,500 to the Silver medal and $500 to the Bronze medal winner for each of the four problems. There is also a $1,000 prize to

each of two select winners in the "Youth" category (aged 18 and under) and $1,000 to each of the three most innovative solutions.

- Multiple submissions are allowed
- Submissions may rely on any preferred framework (i.e., Qiskit, Pennylane, etc.), however the Classiq platform may not but used
- For those unable to receive a cash prize, Classiq has designated several worthy charities where the prize can be donated
- A panel of five judges will review submissions
- For more details, please visit the Classiq Competition page [here]

There has been significant and early enthusiasm for the Coding Competition with about 250 registrations received as of today, from all over the world (see map below) and more than 30 early solutions already submitted despite the contest being open for 15 more days.

Why is This Exciting for Non-Coders?

Nir's quote about "**highlighting the art of the possible**" is why I'm so excited and why you should be too. Some think of computer programs as rigid, dull instructions that either solve a problem or return an error. However, the reality is that programming is a true art, and there are nuanced ways of sequencing, connecting and interweaving programming commands. This is especially true with Quantum Computing in the early "NISQ" (noisy intermediate stage) environment where programmers need to deal with a few additional challenges besides simply programming the solution. These include:

- **Building in error correction**: Because of the noise inherent in current QCs, many of the available physical qubits must be used for error correction overhead. Handling this in an efficient manner is quite challenging, especially with the limited numbers of qubits available today.

- **The probabilistic nature of quantum algorithms**: Quantum programs or algorithms are not simply run once with the program returning an answer. Quantum effects are given by probabilities and the algorithms are generally run many times; each time being referred to as a "shot". Often 10,000 shots are performed before an answer is determined. The program needs to "reset" the qubits between shots to allow the next shot to begin.
- **Not all qubits are created equal**: Certain QCs have limitations on which qubits can "entangle" with certain other qubits. Some can only entangle with nearest neighbors and others are more flexible. Some QCs have qubits that can maintain their state longer and so can run deeper algorithms. In some instances, certain qubits within a given QC are less reliable than others, and there are other peculiarities among different QCs. Programmers need to be able to adapt their programs to factor in these differing characteristics.
- **Machines and Development Kits vary**: There are a number of different quantum computers available via the cloud and each has differing constraints and capabilities. In addition, there are a variety of development kits or programming environments (i.e., Qiskit, Q#, Cirq, Strawberry Fields, Forest, etc.).

This is not meant to be a technical primer on programming QCs but rather is intended to showcase how much "art" there is in programming. For those interested in learning more about these various constraints and challenges I encourage you to read Yuval Boger's (Classiq's CMO) excellent post on this topic here.

Quantum Computers are still quite early in their development. Most available machines have limited numbers of qubits and various strengths and weaknesses regarding coherence, connectability and processing speed. Ultimately, Quantum Computers will only be as valuable and impactful as the programs written for them, so seeing creativity and outside-the-box approaches to challenges with these early, noisy, faulty machines will be quite revealing. I'm excited to get a glimpse into the varying approaches entrants use, and creative methods they employ, to overcome some of the challenges.

I look forward to reviewing the submissions in detail with my illustrious co-judges and plan to provide some insights about the competition in a future post. In the meantime, I encourage you to enter the contest and/or tell your friends and colleagues about it. It's a great chance to test your skills, perhaps win some money, and most importantly, earn the bragging rights as a Medalist in the first ever Quantum Computing Coding Competition.

References:

Boger, Yuval, "Why Is It So Difficult to Write Software for Quantum Computers? And What Can Be Done About It", *Quantum Computing Report*, accessed April 10, 2022.

If you enjoyed this post, please visit my website and enter your email to receive future posts and updates: http://quantumleap.blog

Russ Fein is a venture investor with deep interests in Quantum Computing (QC). For more of his thoughts about QC please visit the link to the left. For more information about his firm, please visit Corporate Fuel. Russ can be reached at russ@quantumleap.blog.