HOW TO MAKE A "QUANTUM" COMPUTER

Authors: Alejandro Mata Ali, Sergio Muñiz Subiñas, Jorge Martínez Martín, Miguel Franco Hernando, Javier Sedano.

The rapid rise of quantum computing in sectors such as industry, finance, and energy has sparked strong interest from governments, companies, and investors. One of the key medium to long-term developments is the acquisition of quantum computers by data centers, businesses, and research institutions, which is fueling the growth of a significant new market. Various companies are already offering these devices, ranging from integrated systems for supercomputing centers to smaller, portable units.

However, this expansion brings with it a critical challenge: how can buyers certifiy that the purchased device is genuinely a quantum computer, rather than a classical simulator disguised as one? Addressing this question is crucial to ensuring trust and transparency in the quantum technology market.

To underscore this concern for both businesses and academia—and to encourage further research—we present a simple design of a fully classical device that can convincingly mimic the behavior of a noisy quantum computer. This system can successfully pass algorithmic benchmarks and withstand common challenges aimed at exposing non-quantum behavior, all while operating solely on classical hardware. Strikingly, such a device could be constructed for less than €3000, even when simulating a large number of qubits.

The existence of such technology highlights a significant vulnerability: it opens the door for fraudulent companies and startups to market fake quantum devices with little oversight. This underscores the urgent need for robust certification standards and verification methods to protect the integrity of the emerging quantum computing ecosystem.

SENSORS

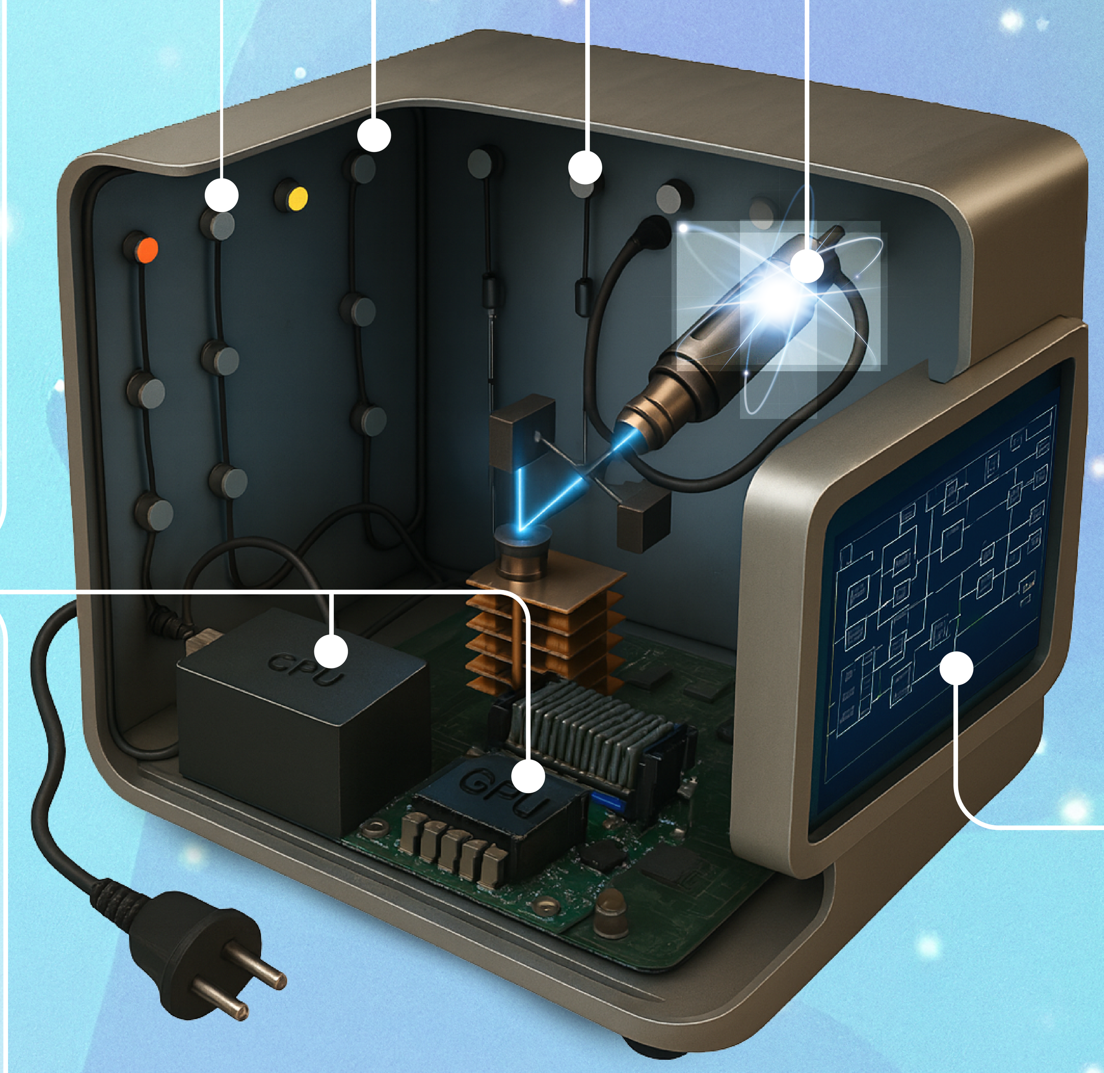
To reproduce the effect of environmental noise in a quantum computer, sensors can be integrated to monitor changes in heat, light, magnetic field and other magnitudes. The device should be designed to cease functioning (or operate with significant degradation) if any sensor is disconnected or if the protective enclosure is opened. This prevents users from testing the quantum computer by opening the device and performing a quantumness test or altering the environmental conditions to check variations in the performance.

The implementation of such mechanism has an estimated expense of about €100.

CLASSICAL COMPUTATION UNIT

For emulation purposes, a classical processing unit equipped with a CPU and, in larger cases, a GPU is employed. The inclusion of these components is justified by the need to transpile quantum circuits and manage complex control systems. In some configurations, they may also support advanced error correction or mitigation methods, depending on the sensor inputs.

The estimated cost is around €1500, for example using a Barebone Zotac Zbox Magnus, with an Intel i7 processor, 64GB RAM and an 4070 RTX GPU. If fewer computational resources are required, the overall cost can be reduced significantly.



THE QUANTUM DEVICE

To reproduce the effect of environmental noise in a quantum computer, sensors can be integrated to monitor changes in heat, light, magnetic field and other magnitudes. The device should be designed to cease functioning (or operate with significant degradation) if any sensor is disconnected or if the protective enclosure is opened. This prevents users from testing the quantum computer by opening the device and performing a quantumness test or altering the environmental conditions to check variations in the performance.

The implementation of such mechanism has an estimated expense of about €100.

TOUCH SCREEN

A touch screen, with some software such as Quirk, to create, implement and execute quantum circuits, and return the results to the user.

The estimated cost is around €100.

CLASSICAL EMULATION

The system must be able to identify the type of quantum circuit and select the most suitable simulation method accordingly. For deep circuits, it employs statevector simulations; for large but narrow circuits, it uses Matrix Product State (MPS) or tensor network simulations; and for Clifford circuits—commonly used in proofs of quantumness—it applies Clifford simulators. All of these approaches can be implemented with the AerSimulator in Qiskit and can be GPU-accelerated for simulations beyond approximately 15 qubits.
Any apparent lack of randomness in the results can be justified as noise or experimental errors. Alternatively, genuine randomness can be incorporated through the use of quantum random number generators. There are three main situations:

SMALL CASE (1-30 QUBITS)

- Digital circuits → Fully simulable with classical resources.
- Analog circuits → Simulable up to ~15 qubits using Trotter decomposition.
- Larger systems → Slower computation.
- Low-depth circuits → Efficiently simulated with tensor networks.
- High-depth circuits → Require statevector simulations.
- Fast approximation methods → Discrepancies can be explained as quantum errors, preserving the appearance of quantum behavior.

MEDIUM CASE (31-34 QUBITS)

- Digital circuits → Fully simulable with classical HPC resources.
- Computation techniques → Standard classical methods apply across different circuit types.
- Large and deep circuits → MPS compression can be applied, with discrepancies attributed to quantum errors.
- Composite system simulation → Circuits can be represented as two interacting subsystems with a defined bond dimension.

LARGE CASE (+35 QUBITS)

- Only narrow circuits can be simulable with techniques such as Tensor Networks; all other types may be entirely faked.
- Verification method → Test correctness by solving specific problems:
  - If the problem is classically solvable → The classical unit can compute and return the solution.
  - If the problem is not classically solvable but the solution is verifiable → The circuit must be deep, allowing discrepancies to be attributed to "quantum errors."

NOISE SIMULATION

The noise model can be either an existing one or an artificial construct. In the latter case, it can be justified under the label of "own new technology" within the quantum device, which prevents direct comparison with known real-world models. Such models can be easily created and applied using the NoiseModel functionality of AerSimulator. However, it is important to note that while simple models are straightforward to implement, more complex noise models demand significantly higher computational resources and execution time.

ERROR CONSISTENCY

The 'quantum' errors must be consistent across the usage. Modifying a single gate should have the expected and predictable impact on the computation, ensuring that the noise model remains consistent. Additionally, the device should account for natural decalibration over time: as more circuits are executed, the simulated noise levels should gradually increase to reflect the 'real quantum' noise increase.
For Tensor Network simulators with extreme compression, it is important to prevent disproportionate output fluctuations caused by small circuit modifications. To mitigate this, selected circuits can be stored along with their computed results, ensuring stability and reproducibility under minor circuit changes.

COMPUTATION TIMES

Computation times depend differently on circuit depth for a simulator compared to a real quantum device. To address this discrepancy, the simulated 'quantum' computer should be exposed as slow enough to cover the worst-case execution time. For circuits that require less time, a Wait command is applied to align the execution duration with the estimated timing of a real device, ensuring consistent and realistic performance modeling.

GOBIERNO DE ESPAÑA · MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES · CDTI INNOVACIÓN · Financiado por la Unión Europea · Plan de Recuperación, Transformación y Resiliencia