Crash Course on Quantum Computing for Engineering Students

Leonel Sousa
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa Lisboa, Portugal, las@inesc-id.pt

***Abstract*—Quantum computing is an interdisciplinary field that spans physics, mathematics, computer engineering, and computer science. Teaching quantum computing effectively within a constrained timeframe presents a challenge, particularly for engineering students with diverse academic backgrounds. This paper presents an approach to integrating quantum computing into a broader course on Technologies of Computing Systems (TCS) through a dedicated two-week module. The course structure combines lectures, interactive problem-solving, lab exercises, and a hands-on project to ensure an engaging and practical learning experience. The curriculum covers fundamental quantum computing principles, quantum gates, circuits, algorithms, and quantum technology. Assessment includes project work, exams, and student feedback, demonstrating increased engagement and comprehension. The results indicate that this quantum computing module successfully bridges the gap between classical and quantum paradigms, fostering student interest and understanding within a short period.**

I. Introduction

Within the last two decades, quantum technologies have progressed significantly, evolving from quantum mechanics and the study of the dynamics of particles at its most fundamental level into a cross-disciplinary field of applied research [1]. Four main different domains have been addressed with quantum technologies: *i*) communication, in which individual or entangled photons allow data transmission in a provable secure way; *ii*) simulation, where quantum systems are used to reproduce the behavior of other, less accessible, quantum systems; *iii*) sensing and metrology, which exploits the high sensitivity of coherent quantum systems to external perturbations to measure physical quantities; and *iv*) computation, which employs quantum effects to set up the large computational state space and perform massively parallel computation, by using quantum superposition, 2^n inputs can be stored in n quantum units of information, quantum bits (qubits), simultaneously [2].

Quantum computing takes a prominent place among the new computing paradigms that are currently being investigated. Moore's law, which predicts that the number of transistors in digital electronics circuits doubles every two years [3], has slowed down, also due to the physical characteristics of transistor technology, which reached dimensions at which quantum effects impact their functionality [4]. Quantum technologies and quantum computing have received significant attention not only from governments and public bodies but also from start-ups to large companies.

Quantum computing has already significant domains of application on optimization, simulation, machine learning, and cryptography [5]. To this end, quantum computing spans different complementary fields, such as physics, mathematics, computer engineering, and computer science. Students to be trained need a background in those areas. Recently, researchers from across academia, government, industry, and national laboratories have proposed a modular quantum engineering course for the first year of undergraduate programs [6]. Although it targets quantum-aware and quantum-proficient engineers at the bachelor's level, there are undergraduate programs without specific courses, like the proposed, on quantum information science and engineering. Thus, this paper presents a course-chapter on quantum computing for engineering graduate students. This course is embedded in a more general course on Technologies of Computing Systems (TCS), which is devoted to new paradigms of computation and emergent technologies for designing computing systems.

This paper describes the experience of teaching a quantum computing course-chapter as part of a 6 ECTS TCS course in an Electrical and Computer Engineering (ECE) MSc program. The skills targeted with this course chapter (2 ECTS) are:

- to identify quantum gates and analyze quantum circuits;
- to analyse quantum algorithms;
- to design and simulate circuits for quantum computation;
- to understand the limits of quantum technology and quantum systems.

The knowledge provided for supporting the skills are:

- principles and math that support quantum computing;
- analysis of the operation of quantum gates and quantum circuits;
- derivation of quantum states and representation of data in the quantum domain;
- tools for testing and designing circuits for quantum computation;
- understanding of quantum technology and quantum systems.

In this course, lectures, interactive problem-solving, lab work, and a practical project work together to create an efficient learning process. All these three components are tightly coupled, pushing forward the knowledge of graduate students and boosting their interest in quantum computing.

The organization of this paper is as follows. Section II provides the context and the organization of the course, including how the learning process is distributed by the different types of classes, and grading, which allows us to assess the success of the proposal and the course-chapter in practice. The subsequent Section III, presents the main topics of quantum computing covered by the course and how they are learned and experienced in the classes.

Section IV discusses the technology behind quantum computing. Section V provides additional information about the labs and a practical project, and Section VI describes how students are evaluated and students' opinions and satisfaction with the course. Finally, Section VII draws the main conclusions and follows up on the impact of the course on other activities of IST.

II. Context and organization of the course

Instituto Superior Técnico (IST), created in 1911, is the faculty of engineering of the Universidade de Lisboa (https://tecnico.ulisboa.pt/en/). The Master on Electrical and Computer Engineering (MEEC) is a two-year MSc general program in Electrical and Computer Engineering offered by IST, 120 ECTS, with seven areas of specialization (https://fenix.tecnico.ulisboa.pt/cursos/meec21). It attracts around 200 new students per year.

TCS is an elective course of the MEEC program with the main objective of providing students with the skills for understanding and designing computing systems based on emergent technologies. The course covers memory technologies and sub-systems, such as resistive memories, and domain-specific accelerators, covering not only the technical issues but also the economic impact [7], and new paradigms of computation, such as processing in memory (PIM) [8] [9], DNA-based computing, and quantum computing. A set of slideshows was prepared to introduce the key concepts, and students have a list of references to technical papers and book chapters to support the learning process. This paper only focuses on the chapter of the course related to quantum computing.

This course assumes students have a background in engineering, and beyond the simple linear algebra operations taught at the BSc level, we provide additional material on tensor algebra, in particular vector-vector and matrix-matrix products that are useful in quantum computing.

The 6 ECTS TCS course runs for seven weeks (quarter), each one including 4 hours of lectures and 3 hours of lab classes, with additional seventeen hours of autonomous work. The last two weeks are fully devoted to quantum computing, which is an independent chapter of this course. Regarding the evaluation, 40% of the final grade comes from a final written exam, and 60% comes from two projects. Since its first edition in 2021/22, the TCS course has been fully taught in English, including all material for lectures, labs, and projects. The work in the first lab class of these last two weeks is guided, introducing projects, materials, and simulators, while in the remaining classes, students have to do the project autonomously. Details about the quantum computing syllabus are provided in the next sections.

III. Quantum computing: knowledge and skills

This section presents the sequence of topics covered and the knowledge and skills acquired in lectures and labs in a symbiotic way. From the start, students are introduced to software simulators to verify the operation of the quantum gates, circuits, and algorithms and analyse the quantum states. Two simulators are adopted in this chapter of the course: the web-based Quirk simulator [10] and the Qiskit open-source software development kit [11]. The former is a web-based simulator with a very user-friendly interface, that allows one to compose quantum circuits by dragging and dropping components, but it is limited to 16 qubits. The latter is able to handle circuits of up to 63 qubits; it is less intuitive to program and use but has the advantage of having associated packages to interface with IBM Quantum systems. Given the nature and duration of this course-chapter, we advise students to use the Quirk simulator, but it is up to them which one they use. The study material is provided at the beginning of the course. Additionally to the set of slideshows exposing the basic concepts, references include two main books [1], [2].

A. Quantum bits and quantum states

The first concept introduced is the basic unit of information of a quantum computer. Starting from a classical bit (cbit), the two states of a cbit can be represented by '0' and '1' in the vector space: $'0' \equiv \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ and $'1' \equiv \begin{pmatrix} 0 & 1 \end{pmatrix}^T$. Immediately, this representation is extended to move from a cbit to a qubit through (1).

$$qubit \equiv \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \ ; \ |p_0|^2 + |p_1|^2 = 1 \tag{1}$$

For complex probability amplitudes ($p_0, p_1 \in \mathbb{C}$), $|p_0|^2$ represents the probability density of a qubit to be in the state '0', and simultaneously in '1' with probability density $|p_1|^2$: the sum of the probability densities should be equal to one; for $p_0 = 1$ or $p_1 = 1$, we are in the classical domain, otherwise the qubit is a superposition of both states. We can also consider the qubit in (1) as a vector in a space defined by a set of orthogonal basis vectors (2):

$$qubit \equiv p_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + p_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \equiv p_0|0\rangle + p_1|1\rangle \ , \tag{2}$$

with $|0\rangle$ and $|1\rangle$ being the Dirac's *ket* notation for states (vectors). In quantum mechanics, it is extensively used the Dirac *bra-ket* notation for computing the amplitude of going from an original state $|o\rangle$ to an ending state $|e\rangle$ through the inner product (3), where $*$ means the complex conjugate and ^{T} the transpose of the vector.

$$\langle o|e\rangle \equiv |o^*\rangle^T |e\rangle \tag{3}$$

After learning how to represent a single qubit, the students are taught how to generalize the vector and Dirac representations for multiple qubits by using the tensor product operation ($\otimes$). At this time, we introduce or revisit, depending on the background of the students, basic tensor algebra. (4) illustrates moving from 1 to 2-qubits $|pq\rangle$, noting that $|p_0|^2 + |p_1|^2 + |q_0|^2 + |q_1|^2 = 1$. It is easy to generalize (4) to n-qubits. The normalization of the complex amplitudes means that the state of the system is a unit in an n-dimensional complex vector space, which is the Hilbert space.

$$|pq\rangle \equiv \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 q_0 \\ p_0 q_1 \\ p_1 q_0 \\ p_1 q_1 \end{pmatrix} \quad (4)$$

While the concept of superposition is introduced in the beginning, we postpone to the end of the lectures the description of two other important concepts: the *measurement effect* and the *quantum entanglement*. Regarding measurement, it is highlighted that when the qubit is measured, it collapses to one of the basic states, which for a single qubit is '0' or '1'. The measurement process irreversibly changes the state of a qubit moving it away from superposition to a new state that is exactly the outcome of the measurement. Therefore, it implies that we cannot collect any additional information about the probabilities associated with an internal state by repeating the measurement.

Quantum entanglement is a complex phenomenon that is difficult to understand. It occurs when a group of particles (qubits) are generated or interact in such a way that the quantum state of a qubit cannot be described independently of the state of others in the group, even when they are physically far away. Although no information is communicated, the behavior of qubits seems to be somehow coordinated, such that measuring a qubit also collapses the other in a correlated state. This coordination happens even across vast stretches of space in an instantaneous way (considering the travel time of light between the entangled qubits). In this course-chapter, we don't go much further explaining the theory behind the entangled state, but when studying quantum gates, examples of circuits that generate qubits in entangled states are provided.

B. Graphical representation of states' space

It is useful to have insightful representations for the states of a qubit, in particular through geometric and graphical means. For the case of qubit states with p_0 and p_1 real numbers, the state of the qubit can be represented as a unit circle in a bi-dimensional space, using the x and y coordinates along the two orthogonal axes. Beyond this particular set of states, students are taught about the possibility of representing the general case of $(p_0, p_1) \in \mathbb{C}^4$ in a 3-D space. Two complex numbers occupy 4-D space, but can be represented in 3-D by introducing the Bloch unitary sphere in Fig. 1. By applying Euler's formulation of complex analysis to (2), students obtain (5). By multiplying (5) by $e^{-i\phi_{p_0}}$, which does not change the probability of each state but only the phase, and by applying the Euler's formula leads to (6), and the representation in the unitary sphere with pair of angles (θ, ϕ) in Fig. 1.

$$|\varphi\rangle \equiv p_0|0\rangle + p_1|1\rangle \equiv R_0 e^{i\phi_{p_0}}|0\rangle + R_1 e^{i\phi_{p_1}}|1\rangle \quad (5)$$

$$\begin{aligned} |\varphi\rangle &\equiv \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle \quad (6) \\ \phi &= \phi_{p_1} - \phi_{p_0} \; ; \; \phi \in [0 : 2\pi[\\ \theta &= 2\cos^{-1} R_0 = 2\sin^{-1} R_1 \; ; \; \theta \in [0 : \pi] \end{aligned}$$

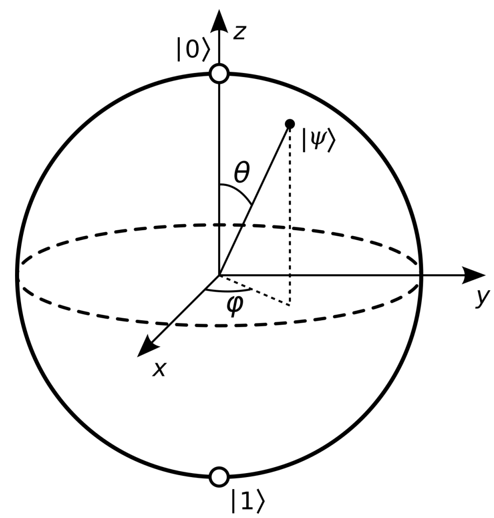


Fig. 1: Bloch sphere

For all experiments conducted by the students, states are analysed in the Bloch sphere, as they are familiar with this representation and identifying transitions between states. When a qubit is in entanglement, its state cannot be individually represented. It is exceptionally considered to be in the center of the sphere.

After these two sections of the course, students attain the skills to represent the state of a quantum system through the state vectors of qubits, and how to represent qubits in the Bloch sphere. The next topic to teach is quantum gates, which define the transitions between quantum states.

C. Quantum gates

Starting by explaining that, like with classical digital circuits, gates in a serial datapath apply the operations sequentially, it is stated that quantum gates can be represented by matrices and their operation by matrix multiplication over the vectors representing the quantum states. Thus, quantum gates operate over quantum states by applying linear algebra operators. This part of the chapter of the course starts with students applying basic quantum gates, like the one in Fig. 2. Students check the state's transition by hand, using tensor and linear algebra like in (7), and by simulation (Fig. 2).

$$\mathrm{C}NOT \times |11\rangle \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad (7)$$

To measure the value of a qubit a meter gate is applied, as illustrated in Fig. 3. Students realise that by collapsing beyond

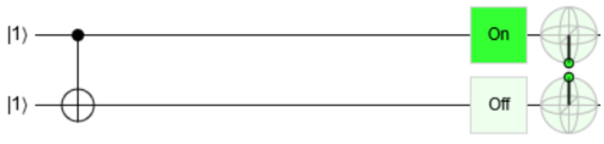


Fig. 2: CNOT gate, with the input $|11\rangle$ (Quirk simulator [10])

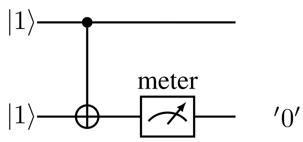


Fig. 3: A meter (gate) outputs the value of the qubit applying the measurement effect

the qubit value to '0' or '1', measurement is not a reversible operation. Whenever used, it changes the value of a qubit and there is no way of moving back to the original value. It means that meters can not be used in a circuit for sensing intermediate values, only at the output to obtain the final result.

The set of basic gates that students analyse and play with are represented in Fig. 4. All operations, implemented by gates, in quantum computing that do not perform measurement are reversible and represented by unitary matrices. It means that for a quantum gate represented by matrix Ψ, $\langle \Psi^* | \Psi \rangle = \mathbf{I}$, with $\mathbf{I}$ being the identity matrix. Like this, we make sure that for the evolution of quantum systems, the sum of probabilities of all possible outcomes always equals the unit.

At this stage, students have to solve simple practical exercises in the lab classes. For example, they must prove a gate of their choosing is unitary. And they play with the simulator to check the transitions between states when different gates are applied in series, like the the quantum circuit in Fig. 5. They compute the state's change produced by each of the two individual gates and compute the matrix corresponding to the whole circuit. Like this, students reach (8), and for the input state $|00\rangle$ the circuit produces the output of the state given by (9).

$$\begin{aligned} H \otimes I &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \\ CNOT \times (H \otimes I) &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix} \end{aligned} \tag{8}$$

$$CNOT \times (H \otimes I) \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{9}$$

It is quite interesting to observe that the state at the output in (9) not only represents superposition but also entangles the

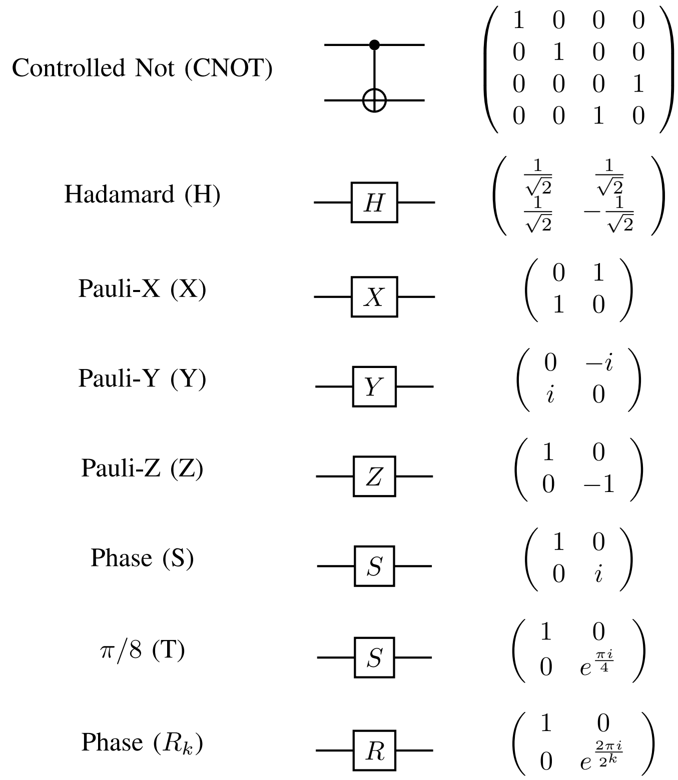


Fig. 4: Basic quantum gates

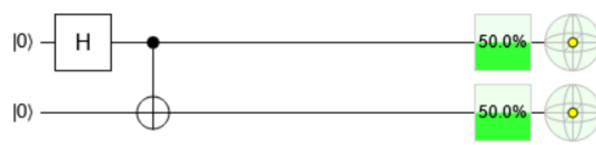


Fig. 5: The control qubit of a CNOT gate is placed in the superposition state: Bell state, simple example of entangled quantum states of two qubits

two quantum bits. According to to (10), it is not possible to separate the behaviour of the two output quantum bits. It is shown through (10) that the quantum state cannot be factored. Thus, students are introduced to the property of entanglement. The quantum bits are somehow coordinated, and measuring one qubit collapses the other in a correlated way. The physical explanation for this phenomenon, which is controversial, is not detailed, but it is simply said that there is no communication, it is quite fast. At this stage, students are challenged to compute the states for the other three states of the quantum input bits, and realize that they form Bell's states. These are specific quantum states of two qubits that represent the simplest (and maximal) examples of quantum entanglement.

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix} \neq \begin{pmatrix}a\\b\end{pmatrix} \otimes \begin{pmatrix}c\\d\end{pmatrix}, \forall(a,b,c,d) \in \mathbb{N}$$
$$(a=0 \vee d=0) \wedge (b=0 \vee c=0) \Rightarrow$$
$$(a.c=0) \vee (b.d=0) \text{ proof of the inequality} \quad (10)$$

At the end of this section of the course, students should have the tools and skills that allow them to analyse any quantum circuit with a defined number of qubits. Moreover, they become familiar with the superposition and entanglement principles.

D. Quantum algorithms

Within the time available in this course-chapter, a couple of seminal quantum algorithms are selected. The Deutsch-Jozsa algorithm and the Shor algorithm are based on the Quantum Fourier Transform (QFT). These are also the algorithms chosen for students' projects in the lab classes.

The choice of the Deutsch algorithm has the purpose of showing how quantum algorithms are developed and how parallelism is used to speed up the solution-finding process. It is also important for showing that while parallel computations are performed, the information we can extract can be limited. The Deutsch Algorithm is a sub-instantiation for 1-qubit of the more general quantum Deutsch–Jozsa algorithm, which provides an exponential speedup over a classic algorithm. For an n-qubit problem, classically $2^{n-1}+1$ computations are required, but only 1 computation using the general quantum algorithm.

The Discrete Fourier Transform (DFT) is an important discrete transform, used to perform Fourier analysis in many practical applications. It takes a finite sequence of equally-spaced input samples (x) into a same-length output sequence of equally-spaced samples (y). The QFT is an important basis of quite relevant algorithms, such as the Shor algorithm [12].

E. Deutsch Algorithm

Students are exposed to the concept of Oracle, observed as a black box but assuming that some physical system can be used to create it in the quantum computer (for example from simple quantum gates). The Deutsch oracle maps $f:\{0,1\} \mapsto \{0,1\}$, which means only one input bit, with two possible values, and one output bit, with also two values $f(0)$ and $f(1)$. The goal of the Deutsch algorithm is to find if the oracle is "balanced", in this case ($f(0) \neq f(1)$), which corresponds to unary identity and negation, or "constant" ($f(0)=f(1)$). The quantum oracle $\mathbb{O}_f$ maps the input 2-qubit $|x\rangle|y\rangle$ into the output 2-qubits according to (11), where $\oplus$ represents the binary XOR operator. Students can check that it is easy to implement this oracle, as it corresponds to an identity gate, but that they do not know the identity of the function $f(x)$. With this oracle and the Deutsch algorithm, it is possible, with a single instantiation by exploiting quantum parallelism, to identify if $f(x)$ is constant or if it is "balanced".

$$\mathbb{O}_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle \quad (11)$$

To evaluate the function with different basis vectors at the same time, students are told to create a special superposition state at the input to the oracle. It is explained that by using Pauli-X (X) and Hadamard (H) gates (Fig. 4), superposition states $|+\rangle|-\rangle$ are created leading to (12).

$$\begin{aligned}|+\rangle|-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \otimes (\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\\ &= \frac{1}{2}(|00\rangle-|01\rangle+|10\rangle-|11\rangle)\end{aligned} \quad (12)$$

Following the proof of the algorithm as in [2], it is shown that by evaluating the quantum Oracle ((11)) the results in (14) are obtained for $f(x)$ constant or non-constant ($\overline{f(x)}$ is the negation of $f(x)$).

$$\mathbb{O}_f|+\rangle|-\rangle = |+\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle-|\overline{f(0)}\rangle) \text{ if } f(0)=f(1) \quad (13)$$

$$\mathbb{O}_f|+\rangle|-\rangle = |-\rangle\frac{1}{\sqrt{2}}(|f(0)\rangle-|\overline{f(0)}\rangle) \text{ if } f(0)\neq f(1) \quad (14)$$

At the output of the Deutsch Oracle, it is easy to identify from the state of the first bit if $f(x)$ is constant or variable, the state will be $|+\rangle$ or $|-\rangle$, respectively. By applying a Hadamard gate (H), ($H|+\rangle=|0\rangle$; $H|-\rangle=|1\rangle$), the outcome of the Deutsch algorithm will be $|0\rangle$ for $f(x)$ constant, and $|1\rangle$ otherwise.

There are "lessons" that students can take from the Deutsch algorithm, which are quite useful for quantum computing in general.

- By taking a superposition of the basis states, they are able to exploit quantum parallelism and to achieve a speedup of 2 even for this simple problem ($n=1$); with more qubits (e.g. 40 qubits), many computations can be performed at the same time (e.g. $2^{40} \approx 10^{12}$).
- Although the Deutsch algorithm leverages quantum parallelism, it only determines whether f(x) is constant or balanced. This limitation is common in quantum algorithms, as they typically extract limited information from parallel computations.

F. Quantum Fourier Transform and Shor's Algorithm

Starting from (15), in the quantum domain, complex numbers are replaced by quantum states. An arbitrary quantum state $\sum_{i=0}^{N-1} x_i|i\rangle$ is mapped through the QFT to the state $\sum_{i=0}^{N-1} y_i|i\rangle$ according to (16).

$$y_k = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j e^{-jk\frac{2\pi i}{N}} \quad (15)$$

$$QFT(|x\rangle) = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1} e^{-xy\frac{2\pi i}{N}}|y\rangle \quad (16)$$

It is important to note that the quantum states in (16) should be represented in binary notation, with each qubit corresponding to a single bit.

As it is usually done for the DFT, (16) can be presented in a matrix form. By considering $\mathbf{W} = e^{i\frac{2}{\pi}}N$ it leads to (17).

$$QFT(|x\rangle) = \begin{pmatrix} W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & \cdots & W^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ W^0 & W^{N-1} & \cdots & W^{(N-1).(N-1)} \end{pmatrix} |y\rangle \tag{17}$$

By deriving in detail the calculations for a 2-qubit system, it is shown that, similarly to the Fast Fourier Transform (FFT), it is possible to achieve a faster version, the QFT with iterative computation [1]. The general equation for iteratively calculating QFT for $N = 2^n$ is (18). Applying the controlled-R quantum gate that applies a relative phase change to the state $|1\rangle$, as presented in Fig. 4, a quantum circuit for implementing the algorithm of the QFT can be found by the students in [1].

$$QFT(|x_1 \cdots x_n\rangle) = \frac{1}{\sqrt{N}}\Big[(|0\rangle + e^{2\pi i 0.x_n}|1\rangle) \otimes (|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle) \cdots \otimes (|0\rangle + e^{2\pi i 0.x_1 \ldots x_n}|1\rangle)\Big] \tag{18}$$

The QFT Quantum Fourier Transform (QFT) is crucial for several quantum algorithms, in particular for the Shor's algorithm [12] analysed by the students. Shor's algorithm is used for finding the prime factors of an integer ($N = P \times Q$), factorizing N by guessing one of its factors and improving that guess in case it is not. Shor's Algorithm starts by taking a random guess P', and checking if it shares a factor with the large number N using Euclid's algorithm. If that is the case, the other factor can be simply obtained as $P = P'$, $Q = N/P$. However, since N is the product of two very large "random" prime numbers, finding a factor is extremely unlikely so, the algorithm will transform P' into a better guess. This new guess comes from the application of Euler's theorem to the relation between P' and N: if P' is not a factor of N, they are coprime numbers and through Euler's theorem can be written as (19).

$$\underset{(coprimes)}{P, N} \rightarrow P'^g = mN + 1 = (P'^{\frac{g}{2}} + 1)(P'^{\frac{g}{2}} - 1) = m \times N \tag{19}$$

The Shor algorithm proceeds to find the periodicity of $f(x) = P'^x \mod N$, which requires separately computing each value of $f(x)$. However, in a quantum system, this issue can be overcome thanks to quantum superposition, which allows the computation of all the images of $f(x) = P'^x \mod N$ at the same time. Given that f(x) is periodic, the resulting state is composed of equal values separated by the period. When the QFT is applied, the results state is composed of the QFT of every image of $f(x)$, which adds together in a state that contains the frequency at which each value repeats $|1/p\rangle$.

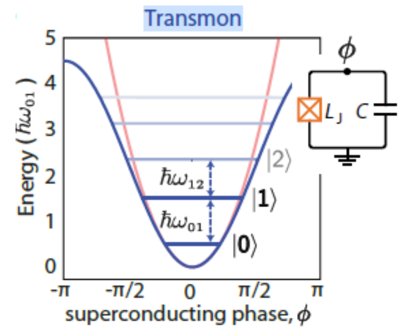


Fig. 6: Transmon qubit, with a Josephson junction [13]

By the end of this section of the course, students have the skills to analyse quantum algorithms and derive quantum circuits to implement them.

IV. Quantum computing technology and systems

It is not possible in such a short period of time to study the physics of the quantum phenomenon and, in detail, the technology behind quantum computers. We have decided to focus the attention on superconducting technology, and quantum computers based on this technology, pointing out the paper [13] for students to read on this topic.

The superconducting qubit has been used on the Noisy Intermediate Scale Quantum (NISQ) technology, in which non-error-corrected qubits are used. Superconducting qubit-based quantum computers are currently dominant, particularly in industry. Recent advances on two-qubit gates implementation, as well as operations on logical qubits in extensible superconducting qubit-based quantum computers, show this technology also holds promise for the longer-term goal of building larger-scale error-corrected quantum computers. [13].

This course-chapter introduces the basic devices for designing superconducting qubit systems and discusses the concept of coherence, the status when the system's behavior can be explained by quantum mechanics.

Contrary to the energy spectrum of a LC harmonic oscillator, the energy spectrum of the transmon qubit in Fig. 6 with a non-linear Josephson junction produces non-equidistant energy levels. The Josephson junction consists of two superconducting electrodes separated by a thin insulating barrier. It allows for the coherent tunnelling of Cooper pairs, which results in a non-linear inductor. Typically, the two lowest levels are used to define a qubit, with $|0\rangle$ corresponding to the ground state and $|1\rangle$ to the excited state. Superconducting circuits have evolved in the last years from two different types of qubits, one based on electric charge and the other based on magnetic flux. The lifetime of a transmon qubit is on the order of a hundred microseconds (Fig 2 of [13]).

The manufacturing of superconducting circuits is a multi-step additive and subtractive fabrication process, which involves lithographic patterning, metal deposition, etching, and controlled oxidation of films of a superconductor, such as aluminium or niobium. Circuits are fabricated on silicon or sapphire substrates, leveraging techniques compatible with silicon CMOS manufacturing. Devices are placed inside a copper or aluminium package that provides an engineered electromagnetic environment thermally anchored to the $\approx$ 10mK (less than -270^{o}C) stage of a dilution refrigerator. Additionally to the qubits, the superconducting circuits comprise resonators and bias lines.

Developing NISQ algorithms will rely on access to quantum computers of increasing complexity. IBM pioneered with a 5-transmon qubit device in 2016, and since then IBM systems have improved. Nowadays, IBM gives access, for example, to D-Wave (see https://cloud.dwavesys.com/), Microsoft to https://azure. microsoft.com/en-us/services/quantum/, and Amazon to https://aws.amazon.com/braket/. Google recently developed a 53-qubit processor, named Sycamore, and reports the first demonstration of quantum computational supremacy, by solving a problem using a quantum computer significantly faster than the best-known algorithm on a classical computer. Sycamore comprises 53 individually controllable transmon-type qubits and 86 couplers used to turn on/off nearest-neighbor 2-qubit interactions.

V. Labs and Projects

Each lab session lasts one and a half hours, with students attending a total of four sessions. They work in groups of three, with the Teaching Assistant actively participating only in the first session. During this initial class, students are introduced to the Quirk and Qiskit simulators, their advantages and limitations are discussed, and a simple demonstration is provided on how to use them to model basic quantum gates and circuits. Although students are free to choose their preferred simulator, they are generally encouraged to use Quirk, as it is more user-friendly, fully web-based, and easier to learn, despite some limitations. Outside of class, students are required to analyze the behavior of a more complex quantum gate using their chosen simulator. A typical example is the *Toffoli* gate, which can be decomposed into six CNOT gates and several single-qubit gates [14]. This exercise provides students with an opportunity to experiment with the simulator, observe how quantum states evolve under different gate operations, and interpret results using the Bloch sphere representation. For the remaining lab sessions, students work independently on their projects.

A couple of quantum algorithms are suggested to students, they select the most interesting one to develop, derive the circuit and simulate its operation. The suggested work always has two stages, a simple first stage that is close to what they easily find in the study book, and a second one that challenges students to go further in the topic. In the first edition of the course, the suggested quantum algorithms are:

- a binary adder in the quantum domain, considering at the input only the two states $|0\rangle$ and $|1\rangle$; in this first stage of the work, students only need to use CNOT and *Toffoli* gates [15]; in this first stage, students have also to extend the quantum circuit to implement an adder with two-input qubits; the challenge for the second stage of the work is to design adding circuits but taking advantage of the superposition, for example, based on the QFT.
- the Deutsh Oracle and the Deutch algorithm to identify the class of unitary operators in the circuit ($f(x)$ constant or balanced); for the second stage of the work, students are challenged to extend the algorithm from 1 to 2 bits, by extending the work to the Deutsch–Jozsa algorithm.

In the week following the final lab session, students must submit a report of four pages, two-column format.

VI. Assessment and Feedback from Students

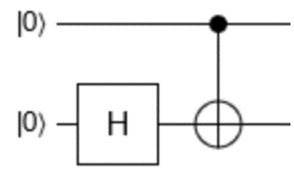
For the course, the assessment, in addition to a final exam, requires a demonstration of the operation of the designed quantum circuits and an individual discussion with each group of the different aspects of the work and the options taken.

An example of the question of the TCS exam on quantum computing is presented. The 6.5 assigned to this question of an exam represents one-third of the 20 total points, approximately the weight of this component of the course.

Technologies of Computing Systems
Exam

---

1) For the quantum circuit based on Hadamard ($\mathbf{H}$) and CNOT quantum gates



[2.5 points] a) Compute the value of the output qubits when the input qubits assume both the value $|0\rangle$.

$$\mathbf{H} \equiv \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix};\ CNOT \equiv \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}.$$

[1.5 points] b) Is the circuit in an entanglement state? Justify the answer and, if it is not the case, give an example of a circuit with quantum entanglement.

[2.5 points] c) Compute the matrix that corresponds to the complete quantum circuit; consider $1 \otimes \mathbf{H} \equiv \begin{pmatrix}\mathbf{H} & \hat{\mathbf{0}}\\ \hat{\mathbf{0}} & \mathbf{H}\end{pmatrix}$.

Confirm the results previously obtained when the input qubits assume the value $|00\rangle$. ■

The experience has shown that some, but few, students show difficulties in facing this completely new approach to computing. These difficulties arise mainly when students have to integrate knowledge of this chapter of the course to extend quantum algorithms for solving more complex problems. This is a component of the course chapter that should be reinforced if one more week is available. The group work and the practical exercises and tools have been revealed to be essential to overcome difficulties. The TCS course, like the majority of the courses in MEEC, is elective and offered in the first of the two years of the master's program.

IST runs a system that monitors the quality of the courses on a yearly basis, called Quality of the Curricular Units (QUC). The QUC is a questionnaire anonymously answered by students about the organization of the courses, the material available to study and to perform experimental work, and also about the professor availability to support students and follow their work. Moreover, as an elective course, the attractiveness also reveals the real interest that the course rouses in students.

In the first edition, the course attracted 58 students, while in the second edition, this number rises to 72 (24% rise). The percentage of students approved, considering the two editions, is around 90% and the average grade is 15.5 out of 20. In fact, in this second edition, it became the most popular course for the specialization areas it was offered. The second edition of the course also saw the enrollment of Erasmus students, originally from (NTNU in Norway, *Université de Grenoble* in France, and *Hochschule München University of Applied Sciences* in Germany. According to the students, this chapter of course on quantum computing is mostly responsible for the widespread interest. Moreover, the course was evaluated by students with 9 out of 9, and was awarded with the seal of Excellence by IST.

Acknowledgment

This work was supported by FCT (Fundação para a Ciência e a Tecnologia, Portugal), and European Defence Fund (EDF) through the projects UIDB/50021/2020, and grant agreement No. 101168112 (SEQURED).

VII. Conclusions and Follow Up

The course-chapter on quantum computing presented in this paper was designed for engineering students and tested strongly positive outcome over two consecutive academic years in an MSc program on Electrical and Computer Engineering. It is an intensive course, with total work hours of 56 hours, from which 14 contact hours in two weeks. Abstracting the computation on most of the course, and only at the very end tackling the technological aspects is a good approach. It allows students to practice quantum circuits and algorithms during the two weeks without facing the additional difficulties of quantum technology, including quantum decoherence. Moreover, interleaving lectures with lab classes and practical projects in a very synchronous way has motivated students to study and learn the main topics and to surpass the novelty that this course chapter carries in terms of quantum computing fundamentals and technology. Students evaluated the quality of this elective course as excellent, and some of them attended the TCS course and later on enrolled in the Minor in Quantum Science and Technologies at IST, which offers three courses in the field of Quantum Technologies, to deepen their knowledge of quantum technology. Some of these students chose to make MSc theses in quantum computing and quantum technologies. An example was an MSc thesis that thoroughly analysed Shor's algorithm and implemented it by using Qiskit, while some of the quantum circuits were tested on the IBM quantum computer [16]. This course aroused the interest in quantum computing at IST, so students who attended the course have organized an in-person discussion panel on quantum computing in the *IST Electrical and Computer Engineering Students Conference* (https://deec.tecnico.ulisboa.pt/noticias/comunidade/jeec-2022). Looking ahead, for computer science students it would be good to reinforce the algorithmic component, while for electrical engineering students, it is crucial to deepen the technological component. For both, it is important to understand the restrictions imposed by NISQ technology. Moreover, to be aware of the longer-term effort of building larger-scale error-corrected quantum computers.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] H. Y. Wong, *Introduction to Quantum Computing*. Springer, 2022.

[3] G. E. Moore, "Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff." *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, 2006.

[4] T. N. Theis and H.-S. P. Wong, "The end of Moore's law: A new beginning for information technology," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[5] A. Bayerstadler, G. Becquin, and et al, "Industry quantum computing applications," *Journal of International Technology and Information Management*, vol. 8, no. 25, pp. 43–71, 2021.

[6] A. Asfaw, A. Blais, K. R. Brown, J. Candelaria, C. Cantwell, L. D. Carr, J. Combes, D. M. Debroy, J. M. Donohue, S. E. Economou, E. Edwards, M. F. J. Fox, S. M. Girvin, A. Ho, H. M. Hurst, Z. Jacob, B. R. Johnson, E. Johnston-Halperin, R. Joynt, E. Kapit, J. Klein-Seetharaman, M. Laforest, H. J. Lewandowski, T. W. Lynn, C. R. H. McRae, C. Merzbacher, S. Michalakis, P. Narang, W. D. Oliver, J. Palsberg, D. P. Pappas, M. G. Raymer, D. J. Reilly, M. Saffman, T. A. Searles, J. H. Shapiro, and C. Singh, "Building a quantum engineering undergraduate program," *IEEE Transactions on Education*, vol. 65, no. 2, pp. 220–242, 2022.

[7] N. C. Thompson and S. Spanuth, "The decline of computers as a general purpose technology," *Communications of the ACM*, vol. 64, no. 3, p. 64–72, 2021.

[8] F. Devaux, "The true processing in memory accelerator," in *2019 IEEE Hot Chips 31 Symposium (HCS)*, 2019, pp. 1–24.

[9] S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, "Compute-in-memory chips for deep learning: Recent trends and prospects," *IEEE Circuits and Systems Magazine*, vol. 21, no. 3, pp. 31–56, 2021.

[10] C. Gidney, "Quirk: Quantum circuit simulator," https://algassert.com/quirk, 2016.

[11] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.

[12] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[13] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting qubits: Current state of play," *Annual Review of Condensed Matter Physics*, vol. 11, no. 1, pp. 369–395, mar 2020.

[14] V. V. Shende and I. L. Markov, "On the CNOT-cost of TOFFOLI gates," vol. 9, no. 5, p. 461–486, 2009.

[15] L. Sousa, "Nonconventional computer arithmetic circuits, systems and applications," *IEEE Circuits and Systems Magazine*, vol. 21, no. 1, pp. 6–40, 2021.

[16] G. B. Valentim, "Quantum computing: From algorithms to implementation," https://fenix.tecnico.ulisboa.pt/cursos/meec/dissertacoes, Lisboa, Portugal, 2021.