



HAL
open science

Full-stack control system for ultracold Rydberg atom quantum computers and simulators

Manuel Morgado

► **To cite this version:**

Manuel Morgado. Full-stack control system for ultracold Rydberg atom quantum computers and simulators. Theoretical and/or physical chemistry. Université de Strasbourg, 2024. English. NNT : 2024STRAF030 . tel-04815996

HAL Id: tel-04815996

<https://theses.hal.science/tel-04815996v1>

Submitted on 3 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE des Sciences Chimiques

THÈSE présentée par:

Manuel Morgado

soutenue le: **26 avril 2024**

pour obtenir le grade de: **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Physique

**Full-stack control system for
ultracold Rydberg atom quantum
computers and simulators**

THÈSE dirigée par:

M. WHITLOCK, Shannon

Prof. Dr., Université de Strasbourg, CESQ

RAPPORTEURS:

Mme. TODRI-SANIAL, Aida

Prof. Dr., Technische Universiteit Eindhoven (TU/e)

M. BIRKL, Gerhard

Prof. Dr., Technische Universität Darmstadt

AUTRES MEMBRES DU

JURY:

M. PUPILLO, Guido

Prof. Dr., Université de Strasbourg, CESQ

M. BAILLEUL, Matthieu (invité)

Dr., Université de Strasbourg, IPCMS

Résumé

Dans cette étude, je relève le défi de contrôler les ordinateurs quantiques et les simulateurs basés sur les architectures d'atomes de Rydberg ultra-froids. Deux conceptions matérielles sont explorées : les atomes individuels et les petites ensembles atomiques, utilisant le Potassium-39 (39K). Je détaille de nouveaux protocoles de portes quantiques, y compris celui pour l'interaction dipolaire de Rydberg. Des méthodes expérimentales pour préparer des états quantiques et réaliser une manipulation globale cohérente sont présentées, avec des configurations avancées utilisant le pompage optique et les champs micro-ondes spatiaux modulés.

De plus, je présente la Boîte à Outils de Traitement de l'Information Quantique Atomique (AQiPT), une architecture logicielle basée sur Python. AQiPT abstrait le matériel, les logiciels et les systèmes quantiques, permettant une programmation complète et un accès transparent à la surveillance, à la gestion des données et aux simulations liées. Ces contributions posent les bases pour une plateforme d'atomes de Rydberg ultra-froids à pile complète, aider à jeter les bases une nouvelle ère pour l'informatique quantique et la simulation numérique.

Mots-clés : atomes ultra-froids, atomes de Rydberg, informatique et simulation quantiques, micro-ondes, atomes individuels, système logiciel quantique, unités de traitement quantique (UTQ).

Résumé en anglais

In this work, I address the challenge of controlling quantum computers and simulators based on ultra-cold Rydberg atom architectures. Two hardware designs are explored: single atoms and small atomic ensembles, utilizing Potassium-39 (39K). I detail quantum gate protocols, including a new one for dipolar Rydberg interaction. Experimental methods for preparing quantum states and achieving coherent global manipulation are introduced, featuring advanced setups with optical pumping and modulated free-space microwave fields.

Additionally, I present the Atomic Quantum Information Processing Toolbox (AQiPT), a Python-based software architecture. AQiPT abstracts quantum hardware, software, and systems, enabling comprehensive programming and seamless access to monitoring, data management, and linked simulations. These contributions lay the groundwork for a full-stack ultracold Rydberg atoms platform, helping lay the foundation for a new era for quantum computing and digital quantum simulation.

Keywords: ultra-cold atoms, Rydberg atoms, quantum computing and simulation, microwave, single-atom, quantum software system, quantum processor units (QPU).

This page was intentionally left blank.

Abstract

In the last 30 years, the field of Quantum Information Processing, Science and Technology has risen as one of the most fascinating, popular and challenging areas of science, due to its duality of combining science and elements of engineering. Recent advances in the classical control of complex quantum systems has encouraged the community to extend the design of current quantum computers and quantum simulators to the limits of software control for the implementation of a universal set of quantum gate operations. This has necessitated the development of new hardware components, such as FPGA based waveform generators with real-time waveform (i.e., electrical signals) generation and feedback. This technology enables precise and fast manipulation of quantum states of the systems while at the same time they are continuously detecting and correcting errors and mitigating the noise from classical hardware affecting the performance. It is an important requirement for quantum computers and simulators to scale beyond the current state-of-the-art of hundreds of qubits to thousands and millions of qubits. This scaling necessarily include more sophisticated classical control hardware as well as the way these are programmed, aware of the dynamics and physical properties of the quantum system that may constrain the manipulation.

In this work, I address the challenge of controlling quantum computers and simulators based on ultra-cold Rydberg atom architectures. Starting by considering two quantum hardware designs for qubits and quantum registers: single atoms and small atomic ensembles, using Potassium-39 (^{39}K). Followed by the quantum gate protocols compiled in this work, including a new gate protocol for dipolar Rydberg interaction. Experimental techniques are also described and implemented for preparing their quantum state and realizing coherent global manipulation with a new setup for optical pumping and arbitrary modulated free-space microwave fields.

Additionally, I designed a novel quantum software system architecture implemented with Python called Atomic Quantum Information Processing Toolbox (AQiPT). The architecture is based in specifications which are abstractions of the quantum device's hardware, software and quantum system. The software includes these abstractions in the different modules that allows theorists, experimentalists and quantum computing users to fully program the quantum device, while still having access to the monitoring, data management and linked simulations from a digital twin.

These contributions form the building blocks of a full-stack ultracold Rydberg atoms platform as an emerging technology for quantum computers and digital quantum simulators. Including the generation of a quantum system that serves as a quantum processor unit and the software system that acts as gateway for different type of users.

Resumé

L'informatique basée sur les technologies des semi-conducteurs a atteint l'une de ses limites fondamentales, où la miniaturisation ultérieure des transistors à 4 nm devient difficile et les effets et phénomènes de la physique quantique tels que l'effet tunnel et le principe d'incertitude deviennent des effets dominants, rendant les transistors peu fiables pour la création de bits. Cependant, cela offre également la possibilité de développer des technologies fondamentalement nouvelles qui exploitent les effets quantiques pour effectuer des tâches plus efficacement que les technologies classiques. Cela met en jeu plusieurs systèmes physiques différents, y compris les technologies des matériaux basées sur la supraconductivité et le silicium, la technologie atomique basée sur les ions et les éléments atomiques, et la technologie photonique basée la propagation de la lumière dans l'espace libre et les puces photoniques.

En 1984, R. Feynman [Fey82] a proposé qu'il serait possible de construire une machine quantique exploitant les principes de la mécanique quantique pour simuler la dynamique des systèmes quantiques, ce que nous appelons maintenant un "simulateur quantique universel". Nous sommes actuellement au cœur d'une révolution de l'informatique quantique, dans laquelle différents modèles pour résoudre des problèmes informatiques en utilisant des états quantiques rivalisent dans le but d'obtenir un avantage quantique. Dans le contexte des atomes neutres, cela inclut : les automates cellulaires quantiques [Win21b], l'annealing quantique [Gla17], la simulation quantique analogique [Ber17a], la simulation quantique numérique [Sig21] et l'informatique quantique numérique [Blu24], avec des cas d'utilisation pour le calcul quantique et la simulation de la chimie [Bar18], de la finance [Orú19] et des applications mathématiques [Whi23].

L'un des principaux défis dans la construction de ces types de machines concerne les systèmes de contrôle, qui deviennent une tâche complexe pour dépasser l'étape du Ère des dispositifs quantiques bruités de taille intermédiaire (Ère NISQ en anglais) [Pre18] avec des centaines de bits quantiques. Cela nécessite un système complet qui nous permette d'initialiser, de manipuler et d'observer les états quantiques. La dynamique du système quantique peut être contrôlée pour former une superposition d'états quantiques, l'une des caractéristiques de la mécanique quantique qui rend le calcul quantique si attrayant pour son "parallélisme quantique" [Deu85]. Cet effet peut être utilisé pour créer une intrication quantique entre une ou plusieurs pièces d'information, conduisant à une puissance de calcul sans précédent pour effectuer certains types de calculs beaucoup plus efficacement que les ordinateurs classiques [Sho94], [Gro96].

Ces piles de systèmes d'information quantique ont certaines caractéristiques en commun, cependant, en fonction de la nature de l'architecture, certaines peuvent être compliquées par le grand nombre de paramètres expérimentaux et les dispositifs classiques-

quantiques utilisés dans la machine.

Dans ma thèse, je présente un pont entre les idées abstraites de l'informatique quantique et les expériences physiques. Cela signifie développer des protocoles expérimentaux, du matériel et des logiciels capables de contrôler des éléments clés du dispositif afin d'accomplir certaines opérations physiques avec le dispositif. Les contributions innovantes de mon travail reposent sur trois points principaux:

- Compiler des démonstrations expérimentales avec des atomes de Rydberg ultra-froids comme une boîte à outils : qubits et portes, y compris un nouveau protocole de porte.
- Développement d'un système quantique pour les systèmes atomiques : logiciel, matériel et architecture logique.
- Mises à niveau expérimentales dans notre configuration à Strasbourg : adressage par micro-ondes à grande portée modulée, piégeage d'un seul atome et conception d'un nouveau système de contrôle.

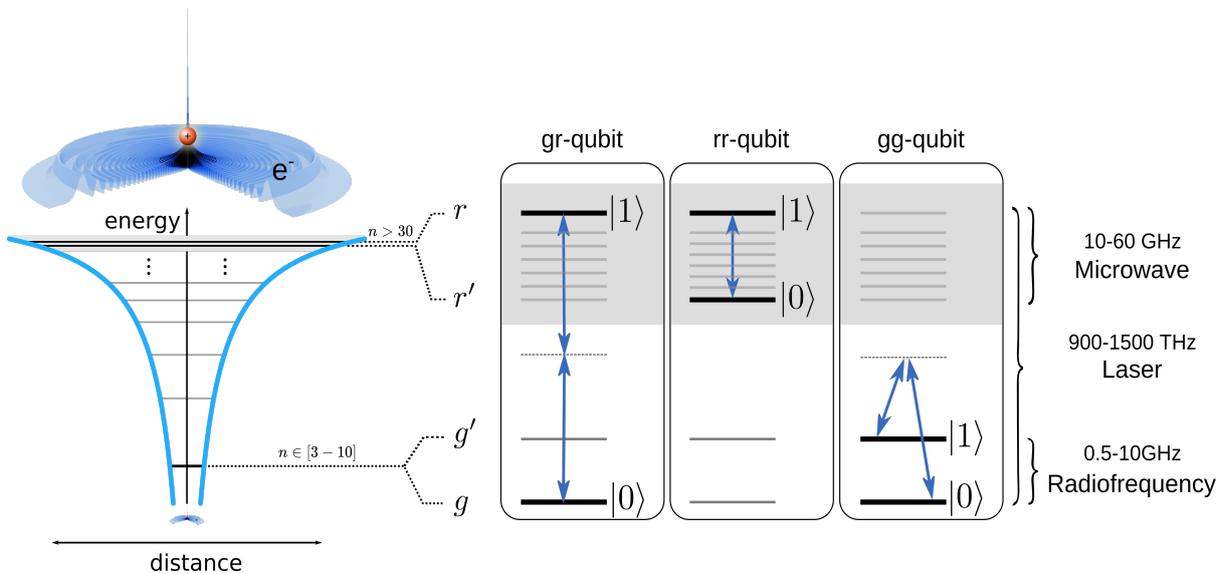


Figure 1: **Encodage des qubits Rydberg.** Classification des qubits interactifs de Rydberg. gr-qubits composés d'un état fondamental et d'un état Rydberg. des rr-qubits composés de deux états Rydberg hautement excités et des gg-qubits composés de deux états fondamentaux. Chiffre tiré de [Mor21]

Qubit de Rydberg à l'état fondamental (gr-qubit)

L'une de mes premières contributions a été de faire une étude bibliographique très détaillée des éléments constitutifs de l'informatique quantique avec des atomes de Rydberg ultra-froids. En rassemblant toutes les avancées de la communauté, les fondements physiques des atomes de Rydberg ultra-froids m'ont permis de définir la boîte à outils de l'informatique quantique de cette architecture. En d'autres termes, nous avons défini le qubit, des ensembles de portes natifs, les modèles de simulation et les limitations. Ces résultats ont été publiés dans un article de revue [Mor21].

En plus de résumer l'état de l'art, nous avons également apporté de nouvelles contributions. La première d'entre elles était une classification des différents types de codage de qubits pour la plateforme Rydberg. Trois types de qubits d'atomes de Rydberg :

Ce qubit est encodé dans un état fondamental et un état de Rydberg fortement excité pour les états $|0\rangle$ et $|1\rangle$ respectivement. Ce type de qubit est généralement manipulé avec des faisceaux laser ayant des longueurs d'onde correspondant à la différence d'énergie entre deux états atomiques généralement définis avec un nombre quantique azimuthal $l=0$ et pour l'état fondamental et excité avec un nombre quantique principal $n < 10$ et $n > 25$ respectivement. La différence d'énergie entre ces niveaux atomiques correspond à $h \times (900 - 1500)\text{THz}$, où la valeur finale dépendra de l'élément atomique et des états sélectionnés. Ce codage de qubit a des durées de vie limitées par la durée de vie de l'état de Rydberg de l'ordre de $\sim 100\mu\text{s}$ et des temps de cohérence de $20\mu\text{s}$ principalement limités par le bruit de phase laser.

Qubit de Rydberg-Rydberg (rr-qubit)

Les états de rr-qubit sont tous deux définis parmi les états de Rydberg. Cela signifie que les deux états du qubit sont deux niveaux de Rydberg différents. Le nombre quantique azimuthal montre différents types d'interactions pour réaliser les portes. Bien que la préparation de ces qubits se fasse à l'aide de faisceaux laser, leur contrôle se fait avec des champs de radiofréquence et micro-ondes d'énergies de l'ordre de $< h \times 1\text{GHz}$ et $< h \times 100\text{GHz}$. Le temps de cohérence typique des qubits individuels est de $22\mu\text{s}$ et de $(4-7)\mu\text{s}$ dans les réseaux.

Qubit à l'état fondamental (gg-qubit)

Ce codage de qubit est la solution actuelle la plus utilisée en raison de ses avantages dans les techniques de lecture et la faible complexité pour les opérations de portes à l'échelle NISQ. De plus, ils sont de bons candidats en tant qu'éléments pour l'informatique quantique tolérante aux erreurs. Les états des qubits gg sont les deux états énergétiques fondamentaux atomiques qui sont contrôlés avec des impulsions laser et des radiofréquences. Ces qubits gg ont une durée de vie de l'ordre des millisecondes.

Élaboration d'un ensemble universel de portes

Une deuxième contribution majeure a été de donner une première vue d'ensemble des différents types de portes logiques quantiques qui peuvent être réalisées en utilisant des interactions médiatisées par les atomes de Rydberg et leurs protocoles correspondants en termes de processus physiques. Nous avons résumé 7 protocoles de portes qui ont été réalisés dans des expériences ou proposés dans la théorie. Ces protocoles sont couramment appelés portes natives, comprenant un ensemble prédéfini de protocoles d'opérations physiques.

En plus de cet ensemble, nous avons présenté un nouveau protocole pour réaliser une porte XY réglable, composée de 4 impulsions. Ces impulsions comprennent:

- A. Impulsion π sur les deux qubits couplant les états $|0\rangle$ à $|r\rangle$
- B. Impulsion 2π hors phase sur la transition $|1\rangle \rightarrow |r'\rangle$ en présence d'une interaction d'échange dipolaire résonante.
- C. Impulsion π sur les deux qubits couplant les états $|r\rangle$ à $|0\rangle$

En choisissant l'angle de la porte pour être de 3π , cette porte effectue une porte i SWAP canonique. En utilisant un champ micro-ondes polarisé, il est possible d'induire des transitions parmi les niveaux de Rydberg avec une grande fidélité. Cela permet l'application d'une correction d'erreur sur les erreurs de fuite des états de Rydberg provenant d'autres portes quantiques telles que les portes CZ. Cette méthode peut transformer les états de fuite dans les qubits de données, présents dans un code QECC en damier [Fow12], en états pour réaliser des calculs. De plus, ce type de technique peut être utilisé pour calibrer et caractériser les portes de phase contrôlée.

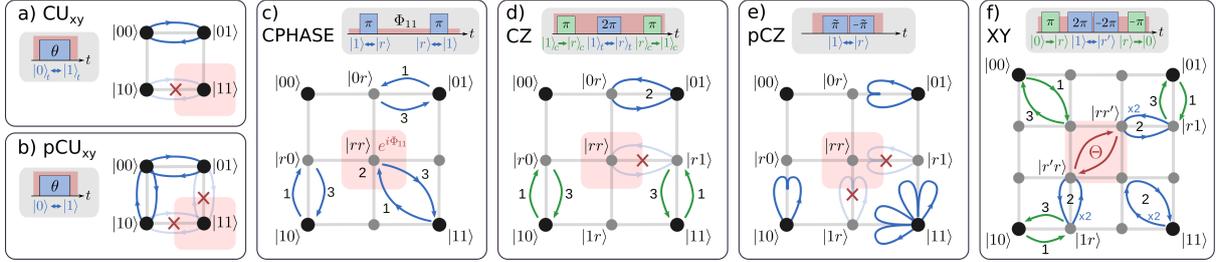


Figure 2: **Ensemble de portes Rydberg natives.** Résumé des séquences d'opérations physiques définies par la porte quantique native pour deux portes qubit. Chiffre tiré de [Morgado2021].

Nouveau système logiciel quantique : AQiPT

Afin de contrôler ces systèmes quantiques physiques, il est nécessaire d'avoir un contrôle complet et précis de l'appareil expérimental, par exemple, l'ordinateur quantique ou le simulateur quantique. Pour cette raison, un système logiciel quantique est nécessaire tout au long de la pile, offrant un accès au contrôle expérimental à différents niveaux. Dans ce travail, j'ai développé un système basé sur Python composé de modules qui permet de définir les protocoles expérimentaux depuis des instructions de bas niveau jusqu'au matériel, jusqu'à des instructions de haut niveau telles que les circuits quantiques. De plus, l'un des modules peut traduire l'instruction expérimentale en un modèle simplifié avec une prise en charge physique du système quantique afin de simuler à l'aide d'un émulateur, alimenté par défaut par un solveur QuTiP [Joh12], mais pouvant être remplacé par l'utilisateur développeur. En général, le logiciel est équipé d'un pilote qui peut être utilisé pour organiser des séquences et des commandes selon les spécifications de l'ordinateur quantique ou du simulateur.

L'abstraction des exigences et des ressources de la machine en spécifications est l'un des piliers de la philosophie de conception logicielle. Cette caractéristique le distingue des autres logiciels disponibles conçus pour différentes architectures. Elle permet au système de gérer une gamme plus diversifiée de dispositifs pour contrôler l'expérience. Par exemple, il prend en compte l'utilisation de modulateurs de lumière, qui peuvent ne pas être nécessaires pour des systèmes tels que les qubits supraconducteurs. Ce système logiciel que nous appelons AQiPT : Atomic Quantum information Processing Toolbox (AQiPT) est également un cadre général où d'autres tâches importantes telles que la surveillance et la gestion des données sont intégrées, permettant une connexion transparente entre la machine quantique physique, les simulations et les techniques de calibration haute performance.

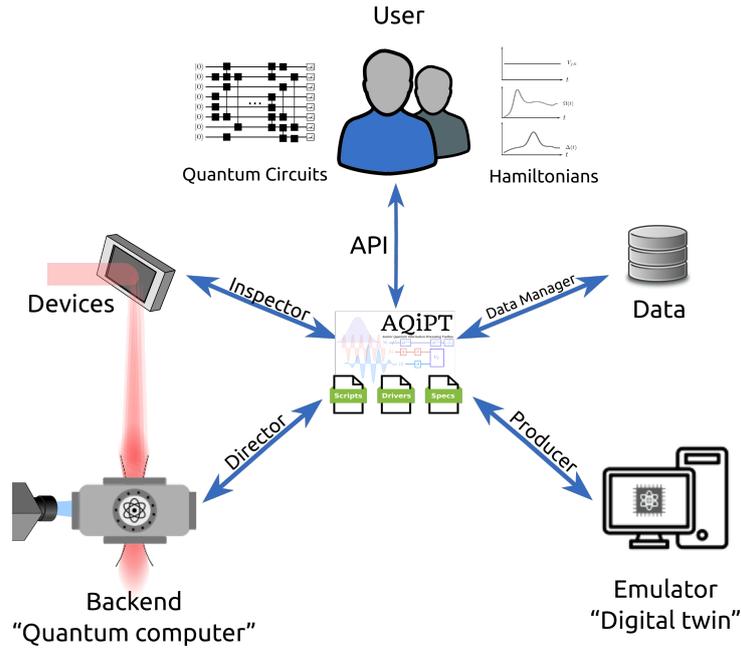


Figure 3: **Architecture AqiPT**. Conception de l'écosystème de modules d'AqiPT aux différents niveaux. Au centre, la spécification atomique, matérielle et logicielle qui est gérée par AqiPT pour les appareils, prépare la séquence expérimentale pour générer les simulations simplifiées.

Génération expérimentale d'architectures QPU

J'ai contribué à la création de vastes réseaux d'ensembles atomiques jouant le rôle de matrices de qubits pour les registres quantiques. Nous avons chargé un réseau de centaines de pièges optiques générés avec un dispositif à micromiroirs numériques (DMD) avec des ensembles d'environ 50 atomes de potassium-39, en utilisant un plan d'atomes de $100\mu\text{m}$ d'épaisseur constituée d'environ cent mille atomes piégés dans un piège optique dipolaire précédemment chargé à partir d'un piège magnéto-optique (MOT) avec des millions d'atomes.

L'utilisation de la technologie DMD nous a permis d'utiliser différentes dispositions spatiales du réseau de pinces optiques, ce qui joue un rôle important dans la configuration d'un processeur quantique universel (QPU) avec une carte de connectivité donnée exploitant les différents types d'interactions de Rydberg.

Une autre manière d'encoder des qubits est de piéger des atomes individuels dans des pinces optiques. Cette technique est largement utilisée dans la communauté, avec l'avantage de simplifier la plupart de la physique à plusieurs corps des ensembles atomiques. Dans notre expérience à Strasbourg, en utilisant des collisions assistées par la lumière pendant des dizaines de millisecondes, il a été possible de piéger un atome unique. Cela a été confirmé en mesurant les statistiques et en trouvant une distribution bimodale lors de la présence et de l'absence de fluorescence d'un atome de potassium unique sur des centaines d'essais expérimentaux. Dans l'ajustement, chaque gaussienne correspond à des événements avec une fluorescence nulle provenant de l'atome détecté sur la caméra EMCCD.

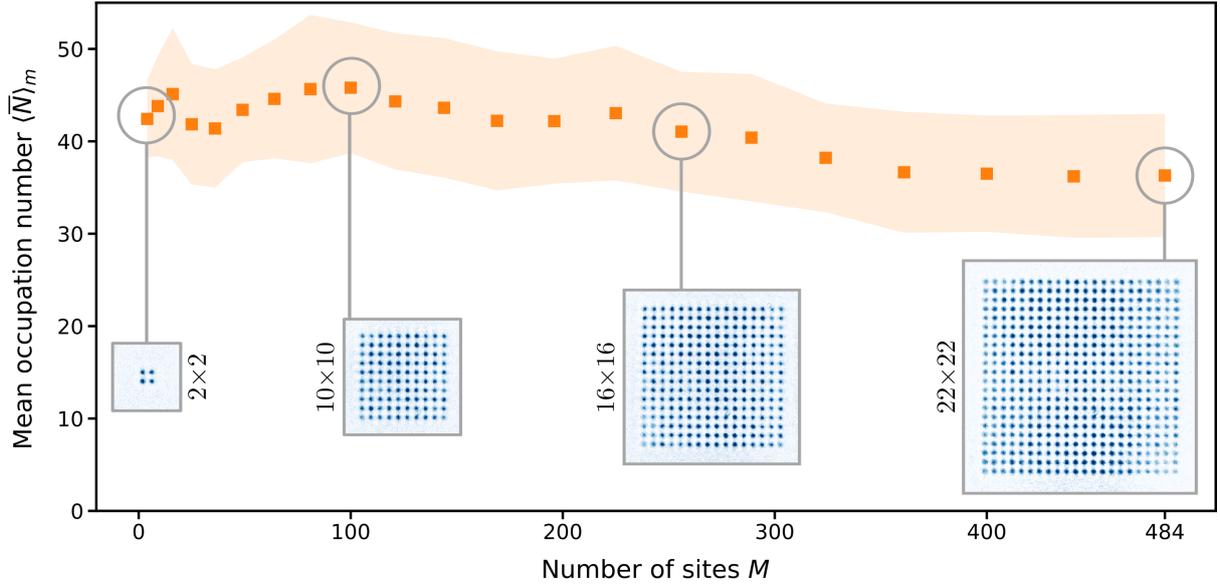


Figure 4: **Réseau d'ensembles atomiques.** Les points de données carrés représentent les nombres d'occupation moyens du tableau pour des tableaux carrés de période de $3,5\mu\text{m}$ ($M = 2 \times 2$ à $M = 22 \times 22$). Le numéro d'occupation reste globalement uniforme et insensible aux pincettes. La région ombrée représente l'uniformité du réseau, tandis que les encarts montrent des images d'absorption pour des réseaux de différentes tailles, en moyenne sur 20 expériences. Chiffre tiré de [Wan20]

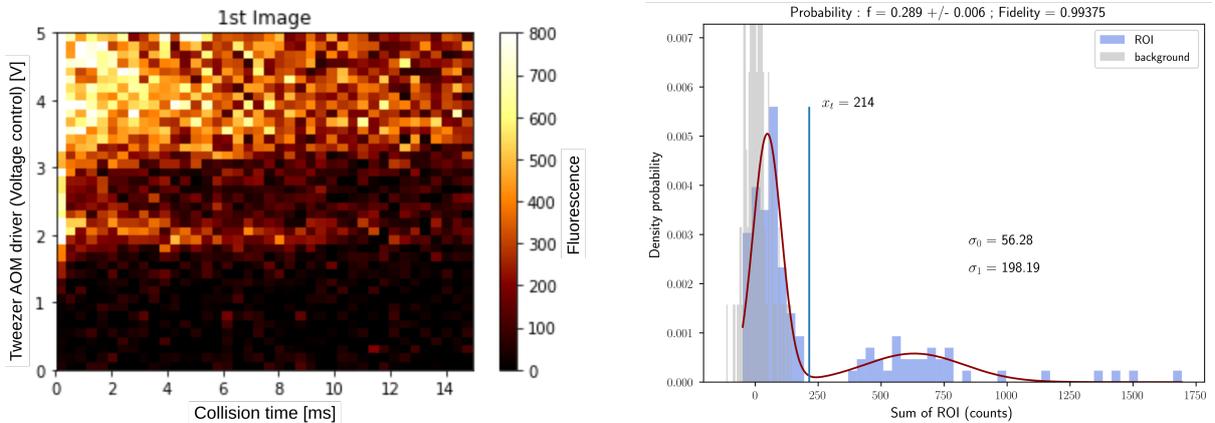


Figure 5: **Piégeage d'un seul atome.** Fluorescence produite pour différentes puissances modulées par la pince en fonction du temps de collision initial (à gauche). Distribution binaire (histogramme) pour la présence/absence d'un seul atome dans la pince optique avec double ajustement gaussien (ligne continue) et un seuil de 214 comptes après application d'un algorithme de suppression de franges et soustraction de fond d'images de fluorescence dans un retour sur investissement de 3×3 pixels à 8×8 binning et gain x300 pour la caméra EMCCD.

Enfin, dans la boîte à outils Rydberg, en plus de l'adressage local généralement effectué par des déflecteurs acousto-optiques bidimensionnels (2D AOD), il est également possible d'avoir un adressage global, qui pourrait être réalisé par des lasers ou, dans le cas des transitions de Rydberg, par des micro-ondes.

J'ai conçu, construit et testé un système de micro-ondes pour le contrôle global des qubits, dans le but d'avoir un adressage global du qubit et plus généralement du registre quantique. En utilisant des réseaux et des ensembles atomiques individuels, nous avons testé le système en réalisant une spectroscopie micro-onde, ainsi que des protocoles Rabi et Ramsey. En pilotant le système (réseaux de qubits rr) avec des micro-ondes, le système a montré des temps de cohérence d'environ 6s avec des fréquences de Rabi très élevées. Ainsi, je présente une proposition de conception utilisant ce type d'encodage avec des qubits gg en tant qu'architecture logique intéressante d'un QPU basé sur des atomes de Rydberg, inspirée par l'architecture classique basée sur des bascules.

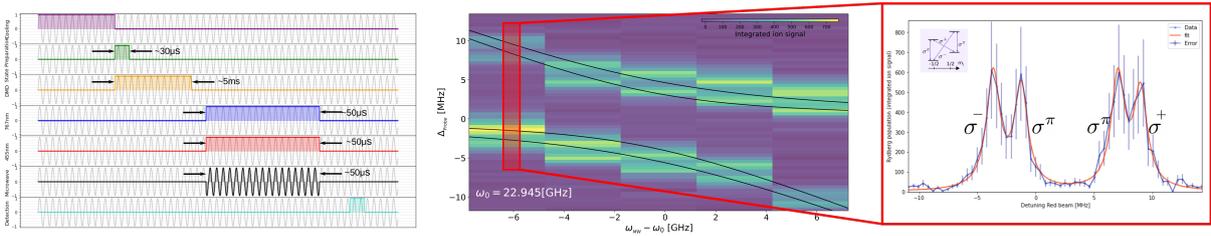


Figure 6: **Système à micro-ondes.** Schéma de séquence d'impulsions expérimentales pour réaliser la spectroscopie (à gauche) Éviter les croisements en spectroscopie micro-ondes montrant une double structure, vraisemblablement due à la polarisation du mélange micro-ondes (à droite).

En résumé, cette thèse offre une vue d'ensemble complète de l'état de l'art du le domaine de la plateforme Rydberg. Elle présente une boîte à outils théorique spécialement conçue pour les ordinateurs quantiques et les simulateurs adaptés à cette technologie, incluant un nouveau protocole de porte pour la correction d'erreurs. La thèse englobe également la conception et la mise en œuvre d'un Système Logiciel Quantique Rydberg pour atomes ultra-froids présentant des implémentations clés au sein d'un environnement Python open source. Ce système facilite le contrôle des Unités de Traitement Quantique (QPU en anglais), la conception de protocoles, la gestion des ressources et la transposition de circuits quantiques - passant des portes natives aux portes canoniques.

Dans le domaine du développement technologique des QPU, la thèse comprend la formulation de protocoles expérimentaux pour piéger des atomes uniques dans des pincettes optiques, en utilisant des collisions assistées par lumière, et la création de réseaux d'ensembles atomiques avec des atomes ultracold de Potassium-39. En outre, en ce qui concerne la manipulation d'ensembles atomiques, une nouvelle conception de l'adressage global des qubits par micro-ondes large bande modulée a été introduite et testée. Cette conception démontre quelques microsecondes de temps de cohérence à grande échelle, y compris dans des réseaux de micro-ensembles d'atomes.

Acknowledgement

First of all, I would like to thank my supervisor Shannon Whitlock and his family for their friendship. Especially him, who introduced me to the world of experimental physics with ultra-cold atoms, with whom I could make the perfect match with my specialty in Quantum Computing. Now, I hope to continue my academic career along this field. Thanks to Tobias, Sayali, Yibo, and Graham from the EQM group, to whom I owe my introduction to the experiment and my beginnings in the group during the master thesis and my first year of the doctorate. To Li Chang, for his teachings of laboratory tricks and tips for fibre coupling and laser locking, as well as always being available to discuss physics ideas and concepts. To Swayangdipta for his excellent hiking advice and open to passionate physics discussions; thanks for being that patient colleague to discuss ideas. To Shuzhe for all those soccer matches and cherry collections. To Tom Dukatzeiler for collaborating and helping with the assembly and measurements with the new microwave setup. To Tom Bienamé for his comments and insights into the experiment.

Thanks to Guido Pupillo and Johannes Schachenmayer for their support in my life at CESQ and for supporting me in bringing the YAO conference to Strasbourg, also to the members of their groups: Tatiana, Sven, Camille and Masella. Also to Paul-Antoine Hervieux and Florian Meinert for their advice during the CSI along the PhD. I would also like to thank my colleagues who joined the EQM group towards the end of my doctorate: Max, who brought a bit more of that German touch to the lab, Amar for his keen interest in learning all the details, and Clément for his enthusiasm in experimental engineering. Best of luck with your doctorates. I would also like to thank the group of DMONS from the IPCMS for borrowing the VNA that we use to understand the functioning of IQ-mixers, also for their tips and suggestions on RF and MW electronics. Thanks to Ramon Szmuk and Nikola Šibalić for helping setting up the OPX and nice discussions about control systems, microwaves and control software.

To my colleagues from QMat and QUSTEC: Guillermo, Vineesha, Deeksha, Apurba, Andreas, Madhav, Carlo, Arya, Grazia, Pietro, Wei, Andris, Patrick, Samiul, Gabe, Alex, Mohammed, Daniel and Denis with whom we enjoyed the beginning of this experience amid the COVID-19 pandemic. We had excellent moments during summer schools, and I truly enjoyed being with all of you. Thanks to Nataliia and Janosh from EUCOR for accompanying us and helping us settle in at the start of the doctorate and being present during the first years.

I also acknowledge our funding source, the French National Research Agency (ANR) through the Programme d'Investissement d'Avenir under contract ANR17-EURE-0024 and QUSTEC funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement MSCA No. 847471 and Horizon Europe programme HORIZON-CL4-2021-DIGITAL-EMERGING-01-30 via the project

101070144 (EuRyQa).

I thank the ISIS Institute and CESQ for allowing me to contribute to the laboratory move in the midst of my Ph.D. and providing all the help to make the transition in record time! (2 months) Special thanks to Raphael Abt, Corinne Ledger, Anna Guyon, Muriel Muzet. Especially grateful to Corinne, Anna, and the ISIS administration for all the help in making my Secondment stay at ETH Zürich more accessible.

My deep and sincere thanks to Jonathan Home for allowing me to do a 6-month Secondment in his Trapped Ion and Quantum Information (TIQI) laboratory at ETH Zürich. During the time I spent in the group, I could not have felt more welcomed and included. I learned a lot about trapped ion experiments, software practices, hardware, experimental setups, quantum engineering, and teamwork. I hope to see more results soon from the Rydberg experiment we worked on and the neighbour GKP experiment. I especially appreciate the time with the Rydberg experiment members Silvan, Woj, Henry, Pavel, the great discussions and questions from Daniel, the great time with the neighbour experiment members of the GKP experiment Martin W, Moritz, and Stephan, as well as from the Cryo experiment Carmelo, Alfredo, Matteo and the control guy Martin S with his infinite patience. I would also like to thank Reina and Stefano for helping me adapting to Zürich.

To all those acquaintances, friends, friends of friends, and relatives who helped me get to Strasbourg for my Master-2, without whom and their financial support, I wouldn't have been able to even start a Ph.D. I will always be grateful for that. To my friends from USB who have helped maintain that spirit of teamwork, passion, and curiosity for physics where I include the group of COF Gustavo, Rosty, Mago, Rubén, Simón, Adri, Moisés, Jeinny, Mary, Rodrigo and others. Also, to those with whom we collaborated to lead COF Alumni USB: Anabel, Julián, Mate, Karly, Kevin, Antonio, Alexandra, Prof. Rita, Prof. Mario and Prof. Jorge, and from which we continue to reap many successes. Huge thanks to Mauricio and Luis as well as to Ines and Julia for helping me make the leap to Strasbourg. Without them, it would not have been possible to be here. Also to Prof. Ladera for supporting me in continue this path, Prof. Castro for giving me the opportunity to co-supervise my first engineering student Angel to which I thanks for be open to keep discussions on programming and his enthusiasm for AQiPT.

I would like to thank my family and all those close friends Stefani, Rubén, Popla, David, Iván, Angelo who are like family to me. To my mom, for always educating me within the framework of excellence and encouraging me to enjoy doing what I love and am passionate about, always giving my best. To my grandmother, who taught me the principles of resilience and hard work, my aunts who were always there to support and connect me with family, and also to my cousin, whom I hope will fall in love with chemistry as much as I have with physics. Last but not least, thanks to my girlfriend's family: Pieter, Greet, Tim, Debby for their support in settling down in Europe and being that second family that helped me not feel intense homesickness. To my girlfriend and life partner, Susan van der Woude, for her unwavering support in pursuing my dreams and everything I want to do.

For all this and more, ¡Gracias totales!

List of publications and open-source projects

Some of the results presented in this thesis have led to the following manuscripts and peer-reviewed publications, which have been written and published along the course of my doctoral studies:

Preparation of hundreds of microscopic atomic ensembles in optical tweezer arrays. Y. Wang, S. Shevate, T. M. Wintermantel, M. Morgado, G. Lothead & S. Whitlock. npj Quantum Information 6, 54 (2020). [Wan20]

Epidemic growth and Griffiths effects on an emergent network of excited atoms. T. M. Wintermantel, M. Buchhold, S. Shevate, M. Morgado, Y. Wang, G. Lothead, S. Diehl & S. Whitlock. Nature Communications volume 12, 103 (2021). [Win21a]

Quantum simulation and computing with Rydberg-interacting qubits. M. Morgado & S. Whitlock. AVS Quantum Sci. 3, 023501 (2021). [Mor21]

Some other projects have led to open-source projects of software and hardware, available in:

Atomic Quantum Information Processing Toolbox (AQIPT). <https://github.com/manuelmorgado/AQiPT>

Modular Amplification Rack-mounted for Voltage and Laser-beams (MARVEL). <https://github.com/manuelmorgado/MARVEL>

Dedicated to those who enjoy calculating, designing and building machines inspired by curiosity.

Contents

| | |
|---|------------|
| Abstract | v |
| Resumé | vii |
| Acknowledgement | xv |
| Thesis outline | 1 |
| 1 Introduction | 3 |
| 1.1 Classical information | 4 |
| 1.2 Quantum information | 8 |
| 1.2.1 Binary digit of quantum information | 9 |
| 1.2.2 Quantum resources | 11 |
| 1.2.3 Quantum logic operations | 12 |
| 1.2.4 Quantum measurements | 14 |
| 1.2.5 Quantum circuits | 14 |
| 1.2.6 Quantum algorithms | 15 |
| 1.2.7 Quantum error correction | 21 |
| 1.3 State-of-the-art in quantum information processing | 23 |
| 1.3.1 Platforms | 23 |
| 1.3.2 Classical control software | 31 |
| 1.3.3 Classical control hardware | 32 |
| 1.3.4 Read-out | 33 |
| 1.3.5 Scalability | 33 |
| 1.4 Main contributions described in this thesis | 33 |
| 2 Quantum computing and simulation with ultra-cold Rydberg atoms | 35 |
| 2.1 Review of criteria for quantum computers and quantum simulators | 35 |
| 2.2 AMO Physics | 39 |
| 2.2.1 Atom-light Hamiltonian | 39 |
| 2.2.2 Optical dipole traps | 39 |
| 2.2.3 Quantum master equation | 41 |
| 2.3 Rydberg Physics | 42 |
| 2.3.1 Rydberg atoms | 42 |
| 2.3.2 Rydberg properties | 44 |
| Dipole matrix elements | |
| Lifetime | |
| Polarizability | |
| 2.3.3 Rydberg interactions | 48 |
| 2.4 Quantum information processing with Rydberg atoms | 52 |

| | | |
|----------|--|-----------|
| 2.4.1 | Rydberg qubits encoding | 52 |
| | Ground-Rydberg qubit (gr-qubit) | |
| | Rydberg-Rydberg qubit (rr-qubit) | |
| | Ground-Ground qubit (gg-qubit) | |
| 2.4.2 | Single qubit Rydberg gates | 57 |
| | U_{xy} (native gate) | |
| | $R_{x,y,z}$ gate | |
| | H (Hadamard) gate | |
| | Pauli $X(\sigma_x)$ gate | |
| | Pauli $Y(\sigma_y)$ gate | |
| | Pauli $Z(\sigma_z)$ | |
| | T gate | |
| | S gate | |
| 2.4.3 | Multi qubit native gates | 66 |
| | CU_{xy} gate | |
| | CPHASE gate | |
| | CZ gate | |
| 2.4.4 | Parallel qubit native gates | 71 |
| | pCU_{xy} gate | |
| | pCZ gate | |
| | C_kZ and C_kNOT gates | |
| | $CNOT^k$ gate | |
| | XY gate | |
| 2.5 | Current state of the art | 76 |
| 2.5.1 | Quantum error correction and noise mitigation | 76 |
| | Physical errors | |
| | Classical manipulation errors | |
| 2.6 | Quantum Full-stack systems | 81 |
| 2.6.1 | Hardware system | 81 |
| 2.6.2 | Software system | 84 |
| 2.6.3 | Frontend | 85 |
| 2.6.4 | Middleware | 85 |
| 2.6.5 | Backend | 85 |
| 3 | Quantum Software Systems for ultra-cold Rydberg atoms architectures | 89 |
| 3.1 | Hardware control system | 89 |
| 3.1.1 | Current technology | 90 |
| 3.1.2 | New generation of classical control | 91 |
| | Installation and features | |
| | Hardware configuration | |
| | Waveform generation | |
| 3.2 | Software control system: AQiPT | 95 |
| 3.2.1 | System architecture | 96 |
| | Specifications | |
| | Workflow | |
| 3.2.2 | Modules | 100 |
| 3.2.3 | Drivers | 114 |
| 3.2.4 | Compiler | 115 |

| | | |
|----------|---|------------|
| 3.2.5 | Learning Materials and Documentation overview | 116 |
| 3.2.6 | Integration | 116 |
| 3.2.7 | Examples | 117 |
| 3.2.8 | Performance | 124 |
| 3.2.9 | License | 127 |
| 3.3 | Challenges and improvements | 128 |
| 4 | Quantum backend for atomic QPUs | 129 |
| 4.1 | Minimal requirements for atomic Quantum backends | 129 |
| 4.1.1 | Atomic element | 130 |
| 4.1.2 | Control hardware | 130 |
| 4.1.3 | Atomic initialization | 136 |
| 4.1.4 | Atomic substrate preparation | 138 |
| 4.1.5 | Manipulation and calibration | 141 |
| 4.1.6 | Measurement and readout | 142 |
| 4.2 | Quantum registers from structured light | 143 |
| 4.2.1 | Single atom loading in an optical tweezer | 143 |
| 4.2.2 | Atomic ensemble loading in an array of optical tweezers | 145 |
| 4.3 | Coherent control | 146 |
| 4.3.1 | New microwave system | 147 |
| | Design | |
| | Hardware setup | |
| | Characterization | |
| 5 | Atomic Quantum Processor Unit | 155 |
| 5.1 | Experimental sequence | 155 |
| 5.2 | Register state preparation | 156 |
| 5.2.1 | Calibration of fields | 156 |
| 5.2.2 | Single atoms | 157 |
| 5.2.3 | Atomic ensembles | 158 |
| | Optical pumping | |
| | Rydberg excitation | |
| | Array generation | |
| 5.3 | Dissipation (depumping) mechanisms | 163 |
| 5.4 | Microwave control of rr-qubits | 164 |
| 5.4.1 | Spectroscopy | 165 |
| 5.4.2 | Lifetime | 165 |
| 5.4.3 | Rabi oscillation and Ramsey interferometry in pancake | 165 |
| 5.4.4 | Rabi oscillation and Ramsey interferometry in tweezer array | 166 |
| 6 | Conclusions & Outlook | 169 |
| 6.1 | Full-stack ultra-cold Rydberg atom quantum computers and simulators | 169 |
| 6.2 | Next generation of the Rydberg platform | 171 |
| 6.2.1 | Quantum backend | 171 |
| | Cryostat systems | |
| | Alkali vs Alkaline-earth species | |
| | Circular Rydberg states | |
| | Fast image detection | |
| | Miniaturization | |

| | | |
|----------|---|------------|
| 6.2.2 | Architecture for QPUs | 173 |
| A | Dirac notation | 175 |
| B | Quantum process state tomography | 179 |
| C | Atomic structures | 181 |
| D | Strasbourg AOM scheme | 183 |
| E | IQ-mixing | 185 |
| F | UML's and flowcharts from AQiPT | 189 |

List of Figures

| | | |
|------|--|------|
| 1 | Encodage des qubits Rydberg | viii |
| 2 | Ensemble de portes Rydberg natives. | x |
| 3 | Architecture AQiPT | xi |
| 4 | Réseau d'ensembles atomiques | xii |
| 5 | Piégeage d'un seul atome | xii |
| 6 | Système à micro-ondes | xiii |
| 1.1 | Classical bit | 6 |
| 1.2 | Turing machine | 6 |
| 1.3 | T flip-flop | 9 |
| 1.4 | Bloch sphere | 10 |
| 1.5 | Quantum Circuit | 15 |
| 1.6 | Quantum Fourier Transform algorithm | 17 |
| 1.7 | Shor's factorization algorithm | 19 |
| 1.8 | Grover's search algorithm | 21 |
| 1.9 | Error correction circuits. | 23 |
| 1.10 | Superconducting qubits | 24 |
| 1.11 | Charge qubit | 25 |
| 1.12 | Flux qubit | 25 |
| 1.13 | Phase qubit | 26 |
| 1.14 | Superconducting Platform. | 26 |
| 1.15 | Semiconducting platform | 27 |
| 1.16 | Ion platform | 28 |
| 1.17 | Ion interaction mechanisms | 30 |
| 2.1 | Criteria for quantum computers and simulators | 36 |
| 2.2 | Quantum devices classification | 38 |
| 2.3 | ^{39}K Quantum defects | 43 |
| 2.4 | ^{39}K Radial wavefunction | 44 |
| 2.5 | Dipole matrix element | 46 |
| 2.6 | Rydberg lifetime for ^{39}K | 47 |
| 2.7 | Polarization and microwave (Rydberg) transitions in ^{39}K | 48 |
| 2.8 | Rydberg mechanisms | 48 |
| 2.9 | Rydberg blockade | 50 |
| 2.10 | Angular dependency of C_6 coefficient | 51 |
| 2.11 | Rydberg qubits | 53 |
| 2.12 | gr-qubit | 54 |
| 2.13 | Ensemble gr-qubit | 55 |
| 2.14 | rr-qubit | 55 |
| 2.15 | gg-qubit | 56 |

| | | |
|------|---|-----|
| 2.16 | Native gate map | 58 |
| 2.17 | U_{xy} | 60 |
| 2.18 | R_x gate performance | 60 |
| 2.19 | R_y gate performance | 61 |
| 2.20 | R_z gate performance | 62 |
| 2.21 | H gate performance | 63 |
| 2.22 | X gate performance | 64 |
| 2.23 | Y gate performance | 64 |
| 2.24 | Z gate performance | 65 |
| 2.25 | T gate performance | 65 |
| 2.26 | S gate performance | 66 |
| 2.27 | CU_{xy} native gate map | 67 |
| 2.28 | CPHASE gate | 68 |
| 2.29 | Model A | 69 |
| 2.30 | Model B | 70 |
| 2.31 | CZ gate | 71 |
| 2.32 | pCU_{xy} gate | 72 |
| 2.33 | pCZ gate | 73 |
| 2.34 | XY new gate protocol | 74 |
| 2.35 | XY gate | 76 |
| 2.36 | Circuit fidelity | 77 |
| 2.37 | D_{\square} | 78 |
| 2.38 | Schematic of the ultra-cold Rydberg platform. | 81 |
| 2.39 | Quantum full-stack | 87 |
| | | |
| 3.1 | General block diagram for AWGs. | 90 |
| 3.2 | General block diagram for FGs and DDSs. | 91 |
| 3.3 | OPX front and back panel | 92 |
| 3.4 | OPX block diagram | 93 |
| 3.5 | New control system installation | 93 |
| 3.6 | OPX performance | 94 |
| 3.7 | AQiPT | 96 |
| 3.8 | AQiPT architecture | 97 |
| 3.9 | AQiPT specifications | 98 |
| 3.10 | AQiPT workflow | 100 |
| 3.11 | AQiPT data structure | 101 |
| 3.12 | Control module of AQiPT | 103 |
| 3.13 | Emulator module of AQiPT | 108 |
| 3.14 | AQiPT Atom specifications from <code>Kernel</code> module | 110 |
| 3.15 | API <code>constant</code> waveform function | 111 |
| 3.16 | API <code>wait</code> waveform function | 112 |
| 3.17 | API <code>digital</code> waveform function | 112 |
| 3.18 | API <code>analog</code> waveform function | 113 |
| 3.19 | API <code>digitals</code> waveform function | 113 |
| 3.20 | API <code>analog</code> s waveform function | 113 |
| 3.21 | Real devices drivers | 114 |
| 3.22 | Devices classes with virtual drivers | 115 |
| 3.23 | AQiPT transpiler | 118 |

| | | |
|------|---|-----|
| 3.24 | QFT algorithm | 119 |
| 3.25 | Gate design simulation. | 120 |
| 3.26 | Setup for backend implementation | 121 |
| 3.27 | Workflow backend implementation | 121 |
| 3.28 | AQiPT <code>atomicModel</code> performance without simulation | 124 |
| 3.29 | AQiPT <code>atomicModel</code> performance with simulation | 125 |
| 3.30 | AQiPT <code>function</code> class performance | 125 |
| 3.31 | AQiPT <code>pulse</code> class performance | 126 |
| 3.32 | AQiPT <code>track</code> class performance | 126 |
| 3.33 | AQiPT <code>sequence</code> class performance | 127 |
| | | |
| 4.1 | Experimental apparatus in Strasbourg | 130 |
| 4.2 | Simplified scheme of potassium-39 (^{39}K) | 131 |
| 4.3 | D-lasers breadboards | 133 |
| 4.4 | Control electronics backend setup | 134 |
| 4.5 | 2D MOT | 137 |
| 4.6 | Science chamber cross-section | 138 |
| 4.7 | Beams in the science chamber | 139 |
| 4.8 | Pancake dipole trap | 139 |
| 4.9 | Digital-Micromirror-Device (DMD) | 141 |
| 4.10 | Optical setup preparing 455nm and 920nm beams | 142 |
| 4.11 | Optical setup for imaging and tweezers | 143 |
| 4.12 | Ion detection hardware | 144 |
| 4.13 | Optical setup for tweezer array with DMD and 780nm | 145 |
| 4.14 | DMD pattern compensation | 146 |
| 4.15 | IQ-mixer | 148 |
| 4.16 | Microwave horn antenna geometry | 149 |
| 4.17 | Microwave hardware setup | 150 |
| 4.18 | Microwave source characterization | 151 |
| 4.19 | DDS characterization | 152 |
| 4.20 | IQ-mixer, LO and DDS characterization | 152 |
| 4.21 | IQ-mixer, LO and AWG characterization | 153 |
| | | |
| 5.1 | Experimental sequence | 155 |
| 5.2 | Fields calibration | 156 |
| 5.3 | Single tweezer: absorption imaging | 158 |
| 5.4 | Single tweezer: fluorescence imaging | 159 |
| 5.5 | Single atom | 159 |
| 5.6 | Optical pumping | 160 |
| 5.7 | Rydberg excitation preparation | 161 |
| 5.8 | Loading array of atomic ensembles | 162 |
| 5.9 | QPU based in an array of atomic ensembles | 163 |
| 5.10 | Dissipation mechanism | 164 |
| 5.11 | Microwave spectroscopy | 165 |
| 5.12 | T_1 time (lifetime) of pancake | 166 |
| 5.13 | Rabi oscillations and Ramsey interferometry in pancake | 167 |
| 5.14 | Rabi oscillations and Ramsey interferometry in tweezer array | 168 |
| | | |
| 6.1 | Flip-flop quantum architecture | 173 |

| | | |
|------|---|-----|
| 6.2 | Rydberg full-stack quantum computers and simulators | 174 |
| C.1 | Potassium-39 atomic energy structure. | 181 |
| C.2 | Other atomic elements energy structure. | 182 |
| D.1 | AOM schematics | 184 |
| E.1 | Normal mixer | 185 |
| E.2 | Mixer example | 186 |
| E.3 | IQ mixing example | 187 |
| E.4 | IQ mixing higher orders | 188 |
| F.1 | UML analysis | 189 |
| F.2 | UML control | 190 |
| F.3 | UML emulator | 191 |
| F.4 | UML kernel | 192 |
| F.5 | <code>Pulse</code> compiler flowchart | 193 |
| F.6 | <code>Track</code> compiler flowchart | 193 |
| F.7 | <code>Experiment</code> compiler flowchart | 194 |
| F.8 | <code>Director</code> software mapper | 195 |
| F.9 | <code>Director</code> hardware mapper | 196 |
| F.10 | Command <code>analog</code> from API | 197 |
| F.11 | Command <code>analogs</code> from API | 198 |
| F.12 | Command <code>compose</code> from API | 199 |
| F.13 | Command <code>constant</code> from API | 200 |
| F.14 | Command <code>digital</code> from API | 201 |
| F.15 | Command <code>digitals</code> from API | 202 |
| F.16 | Command <code>generateDirector</code> from API | 203 |
| F.17 | Command <code>generateProducer</code> from API from API | 204 |
| F.18 | Command <code>generateSpecifications</code> from API | 205 |
| F.19 | Command <code>wait</code> from API | 206 |
| F.20 | AQiPT documentation | 207 |

List of Tables

| | | |
|-----|--|-----|
| 1.1 | Classical logical gates | 8 |
| 1.2 | Quantum logical single-qubit (canonical) gates | 12 |
| 1.3 | Quantum logical multi-qubit (canonical) gates | 13 |
| 1.4 | Properties and features of quantum architectures | 31 |
| 1.5 | Hardware features and control requirements | 32 |
| 2.1 | Rydberg properties | 44 |
| 2.2 | Characteristics of Rydberg qubits | 57 |
| 2.3 | Quantum gates and operations | 79 |
| 2.4 | Atomic elements. | 82 |
| 2.5 | Camera technologies. | 83 |
| 2.6 | Structured light devices. | 83 |
| 4.1 | Microwave antennas | 150 |

Acronyms

- ADC** Analog-to-Digital-Converters.
- AOD** Acousto-Optical-Deflector.
- AOM** Acousto-Optical-Modulator.
- AP** Atom Plane.
- API** Application-Programming-Interface.
- AQiPT** Atomic Quantum Information Processing Toolbox.
- AQS** Analog quantum simulators.
- ARM** Advanced RISC Machines.
- ASIC** Application Specific Integrated Circuit.
- AWG** Arbitrary Waveform Generator.
- BBR** Black-body radiation.
- CMOS** Complementary Metal-Oxide-Semiconductor.
- CPHASE** Controlled-phase gate.
- DAC** Digital-to-Analog Converters.
- DAQ** Data-acquisition.
- DDS** Direct-Digital-Synthesizer.
- DMD** Digital-Micromirror-Devices.
- DP** Double-pass.
- DQS** Digital quantum simulators.
- EIT** Electromagnetic induced transparency.
- EMCCD** Electron-multiplying charge-coupled device.
- EOM** Electro-Optical-Modulator.
- FFT** Fast Fourier Transform.

FG Function Generator.

FPGA Field Programmable Gate Array.

FQS Full quantum-stack.

GRAPE Gradient Ascent Pulse Engineering.

GUI Graphical User Interface.

HW Hardware.

IF Intermediate Frequency.

JJ Josephson junction.

LO Local Oscillator.

LUT Look-Up-Table.

MCP Micro-Channel Plates.

MLA Microlense array.

MOT Magneto-Optical-Trap.

MTS Modulation Transfer Spectroscopy.

MW Microwave.

OBE Optical Bloch equations.

OPT Operator T.

OPX Operator X.

PCB Printed Circuit Board.

PNA Power Network Analyzer.

QAP Quantum Assembly Program.

QC Quantum (gate-based) computers.

QEC Quantum error correction/corrected.

QED Quantum-electro-dynamics.

QFT Quantum Fourier Transform.

QM Quantum Machine.

QPT Quantum process tomography.

QPU Quantum Processing Unit.

RF Radiofrequency.

RWA Rotating-Wave-Approximation.

SDK Software-Development-Kit.

SLM Spatial-Light-Modulator.

SOM System-On-Module.

SP Single-pass.

SPAD Single-Photon Avalanche Diode.

SPAM State-preparation and Measurement.

STIRAP Stimulated Raman adiabatic passage.

SW Software.

SWR Standing-Wave-Ratio.

TA Tapered-Amplifier.

TOF Time of flight.

TTL Transistor-Transistor Logic.

UHV Ultra-high vacuum.

ULE Ultra-low expansion glass.

UML Unified modeling language.

UQC Universal Quantum Computer.

VCO Voltage-Controller-Oscillator.

Thesis outline

This work is presented in six chapters: Chapter 1, provides an introduction to quantum information processing which serves as the motivation for this thesis. This chapter also includes a brief summary of quantum hardware. Chapter 2 provides a comprehensive review of the Rydberg atom platform for quantum computing, which have recently emerged as one of the most competitive physical platforms for quantum computing. Here I include the qubits encoding in this platform, I also show how elementary quantum gates can be realized from physical interactions and a set of gate protocols, error correction and the physics foundations based in a article review of the current state-of-the-art; here it is included a new gate protocol using dipolar Rydberg interactions. In chapter 3, I presented a quantum system design for atomic architectures, including hardware and software conceptual designs for the quantum backend based on our operational quantum simulator. In chapter 4, I give an overview of the experimental setup including our development in the generation of hundreds of optical tweezers using a Digital Micromirror Device. In chapter 5 I report and analyze the achievements on the development of substrates for quantum units of information based on single potassium-39 atoms. This includes the generation of large arrays of optical tweezer with atomic ensembles, observation of coherent driving of Rydberg atoms with microwave fields and first measurements of single-atom trapping. Finally, in Chapter 6, I wrap-up the analysis and insights of the previous chapters, and analyse a series of new elements that can be included in a new generation of these architectures.

This thesis presents a series of experimental, technical and theoretical achievements such as: an assembly of a quantum toolbox for the Rydberg platform, a quantum software system for atomic architectures and a generation of atomic qubits substrate and their coherent control, which link the abstract ideas of quantum computing from the canonical perspective to the backend (native) point of view. Together, these elements represent the basis for a full-stack control system including hardware and software, that is capable of controlling key components to achieve complex physical operations in the device necessary for quantum computing.

The primary contributions of the work lie in three main points:

- Assembly of a toolbox for quantum computers and simulators with Rydberg atoms, based on experimental demonstrations of qubits encoding and quantum gates. This includes a new entangling gate protocol that uses dipolar Rydberg interactions.

Involved chapters: 2 with elements in 3.

- Design of a Quantum System for atomic systems: software, hardware and logic architecture.

Involved chapters: 3 and 6.

- Experimental implementations, observations and first measurements in our setup at Strasbourg of: microwave coherent driving with a modulated large-range global microwave addressing and single-atom trapping.

Involved chapters: 3, 4 and 5.

Introduction

“Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.”

R.P. Feynman (1986)

In this chapter I will introduce the field and topic of this work, together with a brief review of classical information and the state-of-the-art of their quantum counterparts, including the different platforms, exposing the challenges and the need of full-control of the quantum objects that carry the information.

From the Classical to the Quantum world

Classical technology based on silicon will reach soon its fundamental limit, by miniaturising the transistors closer to the atomic scale. Quantum physics effects and phenomena become dominant at this scale and classical bits are no longer reliable due to tunnelling and uncertainty. Therefore, new avenues are being explored, for example the development of 2D materials for the development of semiconducting technology [Roy24] and ARM-like architectures that challenge the computational performance of hypothetical quantum computers, raising a natural question: Do we really need quantum computers? An initial approach to answer this question is by arguing that quantum devices expose correlations that are not accessible with the current technology, potentially enabling new algorithms that exhibit a computational advantage to the classical versions. In the last decades, new technology based on quantum physics has taken an impulse in different platforms using: solid-state technology based on superconductors and semiconductors, atomic technology based on ions and atomic elements, and photonic technology based on free-space and photonic chips.

In 1984 R.Feynman [Fey84] proposed for the first time to build a quantum machine that exploits the principles of quantum mechanics to mimic quantum systems. While various models exist today for executing computational tasks on machines of this nature [Win20], quantum annealing [Ray23], analog quantum simulation [Llo96] [Ber17b], digital quantum simulation [Sig21] and quantum computation [Nie01], we do not yet have demonstrated a model for a full-universal Turing-complete quantum computer. However, prototypes demonstrating building blocks for quantum-controlled (Deutsch model)

$$I(1) \propto -\ln(0.1) = 2.3 \quad . \quad (1.1)$$

The logarithm is taken to measure the uncertainty, facilitating operations and giving a more manageable scale, later the logarithm base is changed to base 2, in order to use a natural description of storage and processing of information in computers. Thus, Shannon's entropy is given by events that are represented by a set of random variables $\{x_i\}$ sampled from a probability distribution and with probabilities $\{p_i\}$, where $\sum_{i=1}^n p_i \log_2(p_i)$ with $0 \leq p_i \leq 1$:

$$H(x) = -\sum_{i=1}^n p_i x_i = -\sum_{i=1}^n p_i \log_2(p_i). \quad (1.2)$$

In the case of a uniform probability distribution of events i.e., $p_i = 1/n$, the maximum value of entropy is $H_{\text{MAX}}(x) = \log_2(n)$. This shows that the uncertainty is maximal. A more graphical example is by thinking of the situation of two images whose pixels are randomly sampled from a picture, where independently each of them shows no information since they do not show any structure; however, when they are summed together they form the picture which shows for example a landscape.

Meanwhile, the J. von Neumann theory converges to a similar expression by considering the entropy (H) of the quantum system given by the density matrix ρ with eigenvalues $\{\lambda_i\}$.

$$H(x) = -\sum_{i=1}^n \lambda_i \log_2(\lambda_i) \geq 0. \quad (1.3)$$

These abstract concepts of information, defined by the probabilities of a sequence of random events and the amount of entropy on them establish the basis for building devices that are able to prepare, manipulate and measure the smallest units of information, called "bits", an acronym for "Binary digit". Coming back to the examples, the bit is defined by the presence of a "1" in a string or piece of the "landscape" in the case of the image example. In contrast, a "0" would represent the absence of information or lack of a piece of "landscape". We will see later that quantum systems use the mathematical description of their state (i.e., wavefunctions) to define this concept in the quantum realm.

Classical bits are implemented in real classical devices and used in current technology, by generating electrical signals using two voltage values in CMOS (Complementary Metal-Oxide-Semiconductor) electronic bits: $<0.4V$ is defined as a bit "0" and $>2.6V$ is defined as a bit "1". On the other hand, there are TTL (Transistor-Transistor Logic) bits which are defined as $<0.8V$ and $>2.2V$ respectively [Sto06]. See also fig. 1.1.

With the capability to generate, manipulate and measure classical information, a natural next step is to come with a model of computation with these building blocks that can execute a sequence of instructions to perform a task (i.e., an algorithm). This raised the interest of D. Hilbert, proposing the *Entscheidungsproblem*, which poses the fundamental question of whether an algorithm is capable of solving all mathematical problems or not. A. Church [Chu36] and A. Turing [Tur36] in 1936 independently tackled Hilbert's challenge by introducing theoretical models of computation that demonstrated the inherent limitations on what could be algorithmically solved. Their models, build

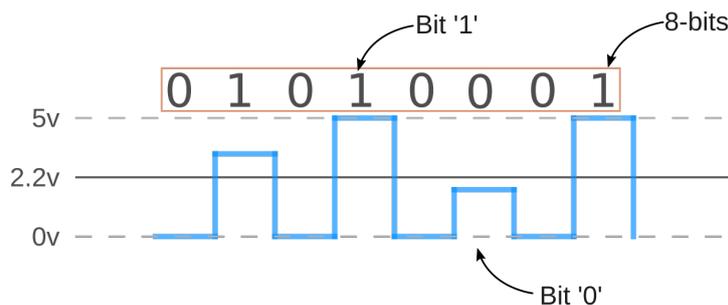


Figure 1.1: **Classical bit**. Representation of an electrical signal where it is possible to define a bit state “0” when the signal is below a voltage threshold of 2.2V and state “1” when it is over the threshold. The complete signal shown is a collection of 8-bits or one byte.

around the fundamental idea of an algorithm, led to the definition of a machine, later called “Turing Machine” composed of four main elements (see also fig. 1.2):

- **Program**: this element holds a finite table of instructions that the machine will execute according to the cell value.
- **State register**: it is the element that stores the machine state including current cell value, head positions and program.
- **Tape**: element that contains multiple values to be processed by the machine. It is divided in cells that contain one value.
- **Read/write tape head**: it is the element that can change the values and can move to specific cells in the tape.

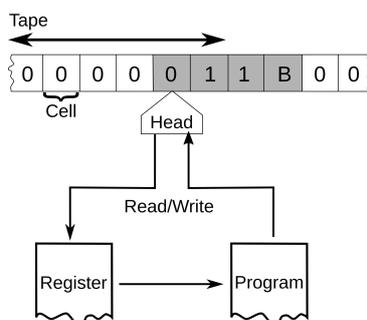


Figure 1.2: **Turing machine**. Scheme of the Turing Machine with its elements: program, register, tape and head. The cells with possible values: 0, 1 or blank, are read by the head and then stored in the state register in order to feed the program that determines the state of the machine and move the tape to a new cell.

Turing machines have the ability to simulate (mimic or execute the steps of) any computation that can be described algorithmically (i.e., a computable process). Meanwhile, the existence of undecidable problems illustrates the limits of what can be done algorithmically. In other words, there are questions or tasks in mathematics that cannot be algorithmically determined or decided as true or false e.g., the halt problem¹, the

¹Halting problem, consist in determining if a task will continue or stop.

homeomorphism problem². On the other hand, a more generalized hypothesis of Turing's work defines the universality of these machines. Where a "Universal Turing Machine" can simulate the behavior of any other Turing machine. In simpler terms, it is a machine that can execute any algorithm that is theoretically possible within the framework of Turing machines.

While the work of Church and Turing laid the foundations of modern computation by establishing an algorithmic model, another perspective on implementing algorithms is provided by the "Circuit model". This model is based on the capabilities of current hardware technology, particularly semiconductor-based electronic transistors. The functionality of these transistors is implemented by configuring certain voltages at their terminals. Moreover, this model will be a more convenient framework to introduce the concepts of quantum computation later on.

This model is represented by a schematic called a "circuit", which comprises wires carrying information in the form of bits with 0 or 1 values, and gates performing operations or transformations on these bit values. The bits are depicted by input and output binary values, while the wires are represented by horizontal lines conveying the bits in time (from left to right). These wires pass over symbols representing gate operations, each of which can take one or more input bit values. Later, we will see a quantum analog represented by the quantum circuits and a classical analog given by a classical control set.

These operations are commonly referred to as logic gates, logic gate operations, or gate operations. Logic gates can be conceptualized as transformations from an input state to an output state, and their actions can be precisely outlined through truth tables. As deterministic operations, these tables remain constant for each gate, illustrating the output values corresponding to all possible inputs, later a quantum counterpart is shown. The foundations of these gate operations are based on the mathematical principles of Boolean algebra. We summarize these gates in table 1.1.

A collection of classical gates provides the foundation for constructing more intricate elements, such as adders, to accomplish more complex tasks [Sto06]. Among these elements, Flip-flops and Latches play a crucial role. They enable the storage of bit information within a dynamic circuit, recognizing that the state of a bit is associated with time-dependent voltage circulating through an electronic circuit (sequential logic). Quantum counterparts are not yet well defined as an embedded component in the quantum model, however Logical quantum bits are good candidates and in chapter 6 I will show a proposal for such a type of component. Electronic flip-flops come in various types and families, for example the T flip-flop (see also fig. 1.3. In this element, the flip-flop states (Q, \bar{Q}) toggles in each clock input cycle when the trigger (T) input is in a high state. Conversely, when the T input is low, Q and \bar{Q} retains their previous values.

Information, represented as bits or flip-flop states, is manipulated through logical operations rooted in algebra. These arithmetic operations form a fundamental set of instructions, facilitating the execution of any computational task. Moreover, the implementation of these gates involves the use of specific voltage configurations across arrays of electronic transistors, each exhibiting similar behaviors based on their arrangement.

²Homeomorphism problem, also know as recognition problem

| Gate | Symbol | Operation | Truth table | | |
|------|---|------------------------|-------------|---|---|
| AND |  | $Q = A \wedge B$ | A | B | Q |
| | | | 0 | 0 | 0 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 1 |
| OR |  | $Q = A \vee B$ | A | B | Q |
| | | | 0 | 0 | 0 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 1 |
| NOT |  | $Q = \neg A$ | A | Q | |
| | | | 0 | 1 | |
| | | | 1 | 0 | |
| NAND |  | $Q = \neg(A \wedge B)$ | A | B | Q |
| | | | 0 | 0 | 1 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 0 |
| NOR |  | $Q = \neg(A \vee B)$ | A | B | Q |
| | | | 0 | 0 | 1 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 0 |
| XOR |  | $Q = A \oplus B$ | A | B | Q |
| | | | 0 | 0 | 0 |
| | | | 0 | 1 | 1 |
| | | | 1 | 0 | 1 |
| | | | 1 | 1 | 0 |
| XNOR |  | $Q = \neg(A \oplus B)$ | A | B | Q |
| | | | 0 | 0 | 1 |
| | | | 0 | 1 | 0 |
| | | | 1 | 0 | 0 |
| | | | 1 | 1 | 1 |

Table 1.1: **Classical logical gates.** Summary of typical gate operations used in the classical circuit model following the corresponding Boolean algebraic operations.

Later, we will explore analogs or equivalents of these gates and their implementations using quantum systems.

1.2 Quantum information

Von Neumann's work showed that there is certain amount of information in quantum systems. However, it was not until Feynman that it became clear that quantum systems can also be considered as machines or devices that can be used to perform a task or calculation, similar to the Turing Machines. Later on, Di Vincenzo proposed a minimal design

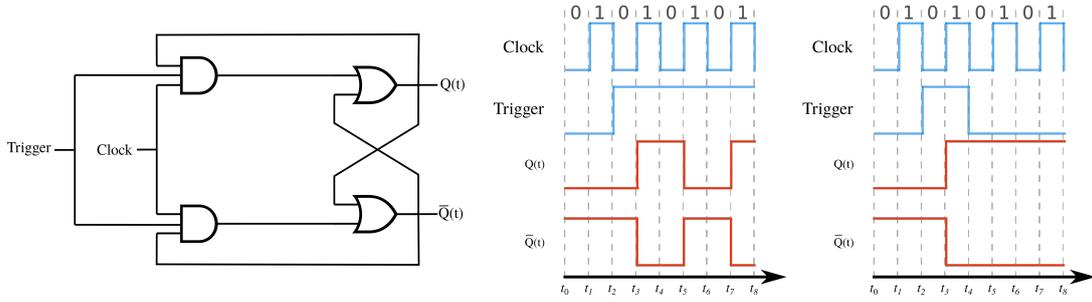


Figure 1.3: **T flip-flop.** **a)** Sequential logic circuit composed by AND and OR gates forming a T flip-flop used in classical computation to store information. **b)** and **c)** Input/Output signal sequence of this type of flip-flop allowing to store a state for a clock cycle when the trigger is activated and deactivated.

and criteria that set minimal requirements for a quantum computer which afterwards was extended by Cirac and Zoller for quantum simulators that I will discuss in detail in the next chapter (see chapter 2).

In the context of the classical circuit model paradigm, quantum computing similarly defines analogous elements. Ultimately, a Universal Quantum Computer (UQC) aims to conduct calculations and simulations equivalent to any other quantum computer or system. It incorporates new types of resources previously discussed, including: quantum entanglement and quantum superposition.

In this section I will be using Dirac's notation for quantum states, where the wavefunction (ψ) of the quantum state in the position or momentum basis can be represented as a "ket" ($|\psi\rangle$) and the equivalent in the dual-space given by its transpose conjugate called "bra" ($\langle\psi|$). A short introduction to Dirac's notation can be found in appendix A. Where the computational states can be represented as column spin vectors:

$$\begin{aligned} |0\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\ |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \end{aligned} \tag{1.4}$$

Additionally, expectation values of operators ($\langle\mathcal{O}\rangle$) are given by:

$$\langle\mathcal{O}\rangle = \langle\psi|\mathcal{O}|\psi\rangle \tag{1.5}$$

1.2.1 Binary digit of quantum information

A binary digit of quantum information, often called quantum bit or *qubit* for short, is the smallest unit of quantum information. The natural way to graphically represent a qubit is with a complex projective plane called Bloch Sphere for qubits (see fig. 1.4).

At the poles we have the two possible states of the qubit and on the equator of the sphere are the equal superposition of states. The superposition of quantum states of

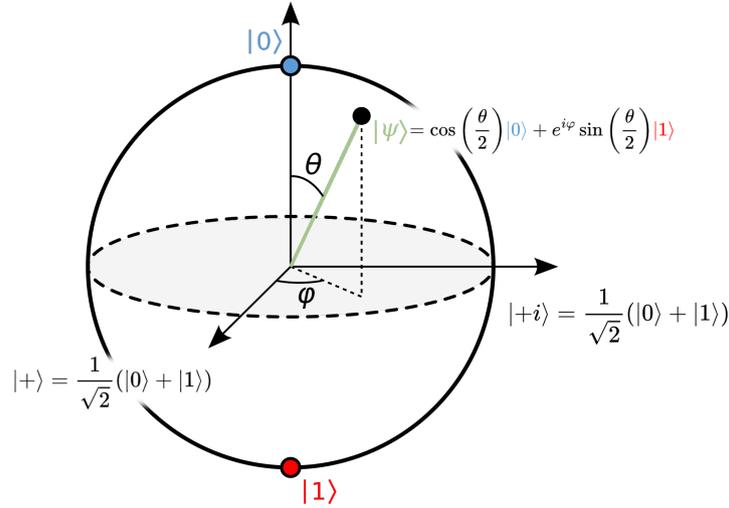


Figure 1.4: **Bloch sphere.** The graphical representation of the qubit state is often shown using spherical-like coordinates of the complex space of the qubit wavefunction, known as the Bloch Sphere. The state in this sphere is described by a fixed normalized radius and two angles: φ and θ . The eigenstates of the qubit constitute the poles of the sphere, meanwhile the equal superposition of the qubit states are located along the equator e.g., $|\pm\rangle, |\pm i\rangle$.

the qubit is one of the resources and it is mathematically denoted by the general linear combination of the eigenstates:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1.6)$$

where α and β are non-zero probability amplitudes of being in the state 0 and 1 respectively, and fulfil the constraint $|\alpha|^2 + |\beta|^2 = 1$. The general expression for a qubit state vector depending on the two angles (similar to spherical coordinates with fixed normalized radius) is described by:

$$|\psi_{(\theta,\phi)}\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)e^{i\phi}|1\rangle, \quad (1.7)$$

where θ and ϕ are the polar and azimuthal angles. On the other hand, the density matrix operator ($\hat{\rho} = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, including quantum states with statistical mixtures) of the qubit system can be written in terms of the Pauli vector ($\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$) and the Bloch vector:

$$\begin{aligned} \hat{\rho} &= \frac{1}{2} (\sigma_0 + \vec{r}_{\text{Bloch}} \cdot \vec{\sigma}), \\ &= \frac{1}{2} (\sigma_0 + r_x \hat{\sigma}_x + r_y \hat{\sigma}_y + r_z \hat{\sigma}_z). \end{aligned} \quad (1.8)$$

Here the Bloch vector is defined as: $\vec{r}_{\text{Bloch}} = (\text{Tr}[\hat{\rho}\hat{\sigma}_x], \text{Tr}[\hat{\rho}\hat{\sigma}_y], \text{Tr}[\hat{\rho}\hat{\sigma}_z])$ where $\text{Tr}[\]$ is the trace operation.

Moreover, it is possible to define a *qudit* which is a d -level system that could bring additional computational power since the generated span is larger than state space generated by the binary system. In fact, often a qudit is the natural form of the used quantum

systems, but by isolating a two-level system it is possible to generate a qubit, which often presents a technical challenge to precisely manipulate it. Another common way to graphically represent qubit/qudit and multi-partite systems is the city-plot of a quantum tomography, to more details refer to appendix B.

1.2.2 Quantum resources

As mentioned above, one of the possible states of the qubit is a superposition state. **Superposition** is an intrinsically quantum mechanical property of a quantum system to have a probability to be in two or more states simultaneously. When considering more than one qubit system, known as a multi-partite system, the state of the composed system can be written as either a product state (separable), a mixed state (statistical mixture) or an entangled state.

If the state of system A is $|\psi_A\rangle$ and the state of system B is $|\psi_B\rangle$, the **product state** ($|\Psi_{\text{product}}\rangle$) is given by:

$$|\Psi_{\text{product}}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle. \quad (1.9)$$

where the symbol \otimes defines the tensor product between two partitions, a detailed explanation can be found in [Nie01]. Thus, this will lead to a density matrix of the shape:

$$\begin{aligned} \hat{\rho}_{\text{product}} &= |\Psi_{\text{product}}\rangle\langle\Psi_{\text{product}}|, \\ &= (|\psi_A\rangle \otimes |\psi_B\rangle) (\langle\psi_A| \otimes \langle\psi_B|). \\ &= \hat{\rho}_A \otimes \hat{\rho}_B = (|\psi_A\rangle\langle\psi_A|) \otimes (|\psi_B\rangle\langle\psi_B|). \end{aligned} \quad (1.10)$$

On the other hand, for a **mixed state** ($|\Psi_{\text{mixed}}\rangle$) with probabilities p_A and p_B associated with states $|\psi_A\rangle$ and $|\psi_B\rangle$ respectively, the density matrix is:

$$\begin{aligned} \hat{\rho}_{\text{mixed}} &= \sum_i p_i |\psi_i\rangle\langle\psi_i|, \\ &= p_A |\psi_A\rangle\langle\psi_A| + p_B |\psi_B\rangle\langle\psi_B|, \\ &= p_A \hat{\rho}_A + p_B \hat{\rho}_B, \end{aligned} \quad (1.11)$$

here the probabilities satisfy the condition $\sum_i p_i = 1$. For an **entangled state** involving two particles A and B, where $|\psi_{\text{entangled}}\rangle$ represents the joint state, the density matrix is:

$$\hat{\rho}_{\text{entangled}} = |\psi_{\text{entangled}}\rangle\langle\psi_{\text{entangled}}|. \quad (1.12)$$

The latter exhibits another important feature of quantum systems used in quantum computing which is quantum entanglement. **Entanglement** is a fundamental phenomenon in quantum mechanics where two or more particles become correlated in such a way that the state of one particle cannot be described independently of the state of the other quantum object.

These quantum resources potentially enable these quantum devices to have speed-ups in higher complexity problems. Two cornerstones of quantum computing are Shor's Quantum Fourier Transform algorithm which, when utilised for number factorization shows an exponential speedup and Grover's search algorithm, which shows quadratic speedup.

1.2.3 Quantum logic operations

The unitary operations (i.e., operators that fulfil the condition $UU^\dagger = 1$) that can be performed over the qubits are called quantum gates, although in this work I will usually refer to them as gates, whenever the context of referring to a quantum system is clear. There is a standard set of quantum canonical operations based on the Pauli matrices, rotation transformations, the Hadamard transformation and controlled gates over pairs of qubits. These canonical gates are summarized in Table 1.2.

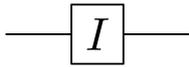
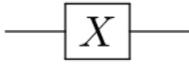
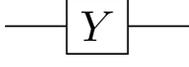
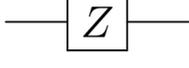
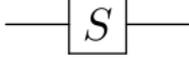
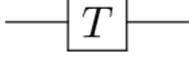
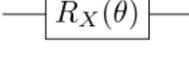
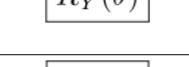
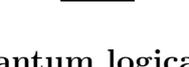
| Gate | Symbol | Operation | Unitary transformation |
|-------------|---|--|--|
| I |  | $I \psi_0\rangle = \alpha 0\rangle + \beta 1\rangle$ | $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ |
| X |  | $X \psi_0\rangle = \beta 0\rangle + \alpha 1\rangle$ | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ |
| Y |  | $Y \psi_0\rangle = -\beta 0\rangle + \alpha 1\rangle$ | $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ |
| Z |  | $Z \psi_0\rangle = \alpha 0\rangle - \beta 1\rangle$ | $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ |
| H |  | $H \psi_0\rangle = \frac{\alpha+\beta}{\sqrt{2}} 1\rangle + \frac{\alpha-\beta}{\sqrt{2}} 0\rangle$ | $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ |
| S |  | $S \psi_0\rangle = \alpha 0\rangle + i\beta 1\rangle$ | $\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ |
| T |  | $T \psi_0\rangle = \alpha 0\rangle + \beta e^{i\pi/4} 1\rangle$ | $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$ |
| R_x |  | $R_x \psi_0\rangle = [\cos(\frac{\theta}{2})\alpha - i\sin(\frac{\theta}{2})\beta] 0\rangle + [-i\sin(\frac{\theta}{2})\alpha + \cos(\frac{\theta}{2})\beta] 1\rangle$ | $\begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$ |
| R_y |  | $R_y \psi_0\rangle = [\cos(\frac{\theta}{2})\alpha - \sin(\frac{\theta}{2})\beta] 0\rangle + [\sin(\frac{\theta}{2})\alpha + \cos(\frac{\theta}{2})\beta] 1\rangle$ | $\begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$ |
| R_z |  | $R_z \psi_0\rangle = e^{-i\frac{\theta}{2}}\alpha 0\rangle + e^{i\frac{\theta}{2}}\beta 1\rangle$ | $\begin{pmatrix} \exp(-i\theta/2) & 0 \\ 0 & \exp(i\theta/2) \end{pmatrix}$ |
| Measurement |  | $ \alpha ^2$ or $ \beta ^2$ | $ 0\rangle 0\rangle$ or $ 1\rangle 1\rangle$ |

Table 1.2: **Quantum logical single-qubit (canonical) gates.** Where $|\psi_0\rangle = \alpha|0\rangle + \beta|1\rangle$ is the initial qubit state and α and β the amplitude probability of the qubit eigenstates $|0\rangle$ and $|1\rangle$.

Multi-qubit gates are also important in the generation of entanglement and development of quantum algorithms, they are summarized in Table 1.3.

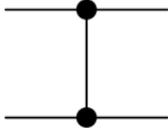
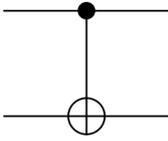
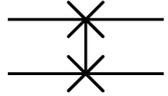
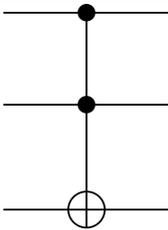
| Gate | Symbol | Operation | Transformation |
|---------------|--|---|--|
| CZ |  | $CZ \psi_0\rangle = a 00\rangle + b 01\rangle + c 10\rangle - d 11\rangle$ | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$ |
| CX |  | $CX \psi_0\rangle = a 00\rangle + b 01\rangle + d 10\rangle + c 11\rangle$ | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ |
| SWAP |  | $SWAP \psi_0\rangle = a 00\rangle + c 01\rangle + b 10\rangle + d 11\rangle$ | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ |
| TOFFOLI (CCX) |  | $CCX \psi_0\rangle = a 000\rangle + b 001\rangle + c 010\rangle + d 011\rangle + e 100\rangle + f 101\rangle + h 110\rangle + g 111\rangle$ | $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ |

Table 1.3: **Quantum logical multi-qubit (canonical) gates.** Where $|\psi_0\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ is the initial qubit state and α and β the the amplitude probability of the qubit eigenstates $|0\rangle$ and $|1\rangle$.

Here we also notice that some of the gates depend on certain values, these types of gates are usually recognized as “Parametric gates”.

As in many aspects of algebra, the notion of distance or a “metric” is used to quantify properties. The performance of a quantum gate is usually reported in terms of a bounded quantity called “gate fidelity” (\mathcal{F}_g), which by itself is not a metric since it does not satisfy all the criteria for a metric (i.e., triangular inequality), however the gate error ε is in fact a metric and it is given by:

$$\varepsilon = 1 - \mathcal{F}_g. \quad (1.13)$$

Where the gate fidelity (\mathcal{F}_g) depending on the target ($\hat{\sigma}$) and obtained density matrix ($\hat{\rho}$) is given by:

$$\mathcal{F}_g(\hat{\rho}, \hat{\sigma}) = \left| \text{Tr} \left(\sqrt{\sqrt{\hat{\sigma}} \hat{\rho} \sqrt{\hat{\sigma}}} \right) \right|^2. \quad (1.14)$$

This measure can also be extended to the resultant state of the composite system following a quantum circuit. However, quantifying the performance in this context is not well-defined, and it remains an active area of research [Wac21].

1.2.4 Quantum measurements

In classical theory, measuring the system is relatively simple. Since each measurement comes from average magnitudes from microscopic properties where classical and semi-classical physics are dominant.

On the other hand, in quantum mechanics this is very different since the theory is based on the concept of wavefunctions represented by state vectors describing the state of the system using probability amplitudes, which makes the theory non-deterministic. Additionally, when the measurement is performed the system no longer exists in the original state, this is known as the “collapse of the wavefunction”. Therefore, one way to describe measurements is by “projecting” or collapsing the state vector into one of the possible states, measuring the probability of the system being in that state. This is achieved by using a measurement operator. This can be represented by calculating the probability of the state $p(m)$ whenever a measurement is applied:

$$p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle, \quad (1.15)$$

where ψ represents the state of the system, M_m and M_m^\dagger the measurement operator and its transpose conjugate. This kind of measurement is also understood as a non-unitary gate (i.e., $M_m^\dagger M_m \neq 1$), because it is not possible to reconstruct the state after collapsing the wavefunction.

Throughout this work in chapter 2, chapter 4 and chapter 5 I will mention different ways to measure atomic systems: absorption imaging, field-ionization and fluorescence imaging. These techniques are destructive techniques which alter the system in an irreversible way even beyond an ideal projective measurement. However, they can be used to realize non-destructive measurements with clever solutions, including additional subsystems or ancillary states or qubits.

1.2.5 Quantum circuits

In order to construct a formalism that can represent the tasks that will be executed by quantum system (i.e., quantum computers and simulators), there are different models of computation: adiabatic quantum computing [Alb18], quantum cellular automata [Len93], measurement-based quantum computing [Bri09] and the gate based quantum computing [Nie01].

Similarly to the classical gate model, the quantum circuit model is represented by a schematic of the timeline evolution of the qubit states (i.e., evolution in time from left-to-right, horizontal axis in fig. 1.5). At the beginning of each line the initial state of the qubit is usually specified. Along this line we find square boxes representing the different operations over the qubit state at different sections of the schematic. Different qubit evolution lines can be connected transversally with other lines that link two boxes (one in each qubit) representing two or multiple qubit gates. The multi-qubit gates are also represented by a single rectangular box addressing many qubit evolution times. Within the schematics of quantum circuits there are two characteristics that are important: the number of layers or vertical columns of gates that occur simultaneously and the width of the circuit which is given by the number of qubits of the circuit. All these elements are represented in fig. 1.5.

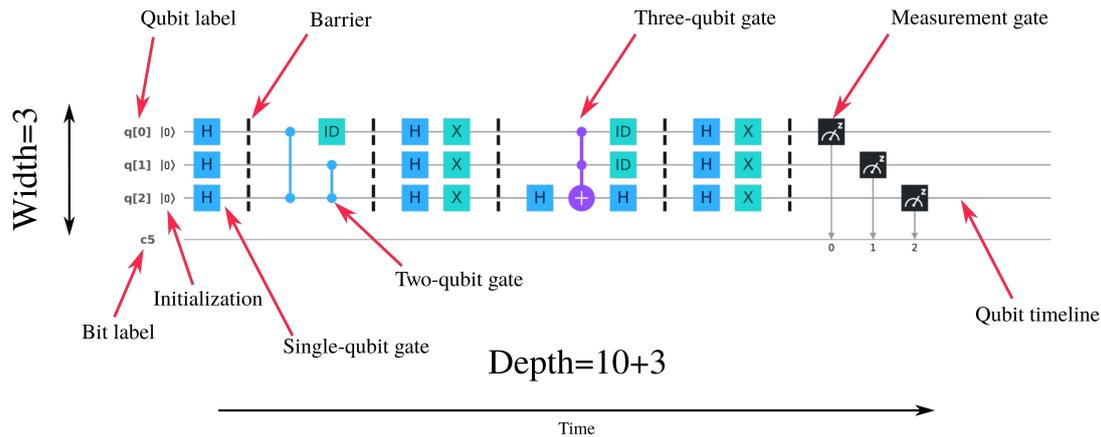


Figure 1.5: **Quantum Circuit**. The main properties of a quantum circuit are: width (number of qubits) and depth (number of gate layers). Within the circuit diagram it is pointed the qubit label with its timeline and state initialization. Changes in the initial state depends on execution of single-qubit and multi-qubit gates. Barriers introduced by the software development kits Qiskit [Qis23] allowing visualization and separation of layers along the circuit. A final measurement is performed and the values are stored in a classical bit.

1.2.6 Quantum algorithms

As mentioned before, quantum algorithms could potentially bring certain advantages in hard problems and enable them to address intractable problems. These algorithms in the context of quantum computation try to solve arbitrary tasks and they are usually represented by quantum circuits composed of gates. In the future, it is very likely that currently developed quantum algorithms will serve as primitives for programming quantum devices. Here, I briefly go through the cornerstones of quantum algorithms that in fact show certain theoretical advantages. I will discuss the Quantum Fourier Transform (QFT) replicated later in chapter 3 with the developed software (see chapter 3).

In general, current trends in the development of quantum algorithms are the semi-classical algorithms [Per14], this means that they have a partial intervention of classical hardware or software which sometimes are enhanced by machine learning.

Quantum Fourier Transform

An important algorithm in quantum computing which serves as workhorse is the Quantum Fourier Transform (QFT) algorithm, see fig. 1.6. Similarly to its classical counterpart, this algorithm represents a basis transformation of a problem that shows periodic behaviour or structures into a space where the problem is simplified. Often, this transformation in the classical implementation brings the calculation from the real or time domain to the Fourier or frequency domain.

In order to have a glance of the QFT algorithm we will take a look at the Hadamard operation which is the most simple implementation of the QFT, making use of interference phenomena. The Hadamard operation takes the state from the basis defined by $\{|0\rangle, |1\rangle\}$ i.e., Z -basis to the X -basis given by $\{|+\rangle, |-\rangle\}$. This means that the QFT is represented by a local rotation operation over the Z -basis eigenstates. This can be

generally represented by the operator:

$$R_n = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^n} \end{pmatrix}, \quad (1.16)$$

where n is the qubit or operation index or term of the algorithm and $2\pi/\varphi$ is the rotation angle. When the operation is applied repeatedly for k -times, the operator is rewritten using the base-ten representation of $k = k_0 + k_12^1 + k_22^2 + \dots + k_{n-1}2^{n-1}$ as:

$$R_n^k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^n}k} \end{pmatrix}, \quad (1.17)$$

where

$$e^{\frac{2\pi i}{2^n}k} = e^{\frac{2\pi i k_0}{2^n}} \cdot e^{\frac{2\pi i k_1}{2^{n-1}}} \dots \cdot e^{\frac{2\pi i k_{n-1}}{2}}, \quad (1.18)$$

therefore the generalized k -times rotation operation is:

$$R_n^k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^n}k_0} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{n-1}}k_1} \end{pmatrix} \dots \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2}k_{n-1}} \end{pmatrix}. \quad (1.19)$$

In terms of quantum operations i.e., $|\tilde{\psi}\rangle = QFT|\psi\rangle$, the sequence of rotations can be written as:

$$|\tilde{\psi}\rangle = \frac{1}{\sqrt{N}} \sum_{\phi=0}^{N-1} e^{\frac{2\pi i \psi \phi}{N}} |\phi\rangle, \quad (1.20)$$

where $N = 2^n$ is the basis state. Defining ϕ similarly as k as a binary string i.e., $\phi = \phi_0 + \phi_12^1 + \phi_22^2 + \dots + \phi_{n-1}2^{n-1} \equiv \phi_{n-1}\phi_{n-2}\dots\phi_1\phi_0$, and eq. (1.18) the operation can be written as:

$$|\tilde{\psi}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\phi=0}^{2^n-1} \prod_{\ell=1}^n e^{\frac{2\pi i \psi \phi_\ell}{2^\ell}} |\phi_1\phi_2\phi_3\dots\phi_n\rangle, \quad (1.21)$$

where the notation $|\phi_1\phi_2\phi_3\dots\phi_n\rangle$ represents the iteration over all bit-strings of the 2^n basis-states i.e., $\sum_{\phi=0}^{2^n-1} = \sum_{\phi_1=0}^1 \sum_{\phi_2=0}^1 \dots \sum_{\phi_n=0}^1$. This leads to a final state of the form:

$$|\tilde{\psi}\rangle = \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i(\psi)}{2^1}} |1\rangle \right) \otimes \left(|0\rangle + e^{\frac{2\pi i(\psi)}{2^2}} |1\rangle \right) \dots \otimes \left(|0\rangle + e^{\frac{2\pi i(\psi)}{2^n}} |1\rangle \right). \quad (1.22)$$

Its circuit representation is shown in fig. 1.6. For the simple case of the QFT using 1 qubit (Hadamard transformation), the transformation is:

$$QFT|0\rangle = \frac{1}{\sqrt{2^1}} \sum_{\phi=0}^1 e^{\frac{2\pi(0)\phi}{2^1}} |\phi\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle), \quad (1.23)$$

$$QFT|1\rangle = \frac{1}{\sqrt{2^1}} \sum_{\phi=0}^1 e^{\frac{2\pi(1)\phi}{2^1}} |\phi\rangle = \frac{1}{\sqrt{2}} \left(e^{\frac{2\pi i(0)}{2}} |0\rangle + e^{\frac{2\pi i(1)}{2}} |1\rangle \right) = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle). \quad (1.24)$$

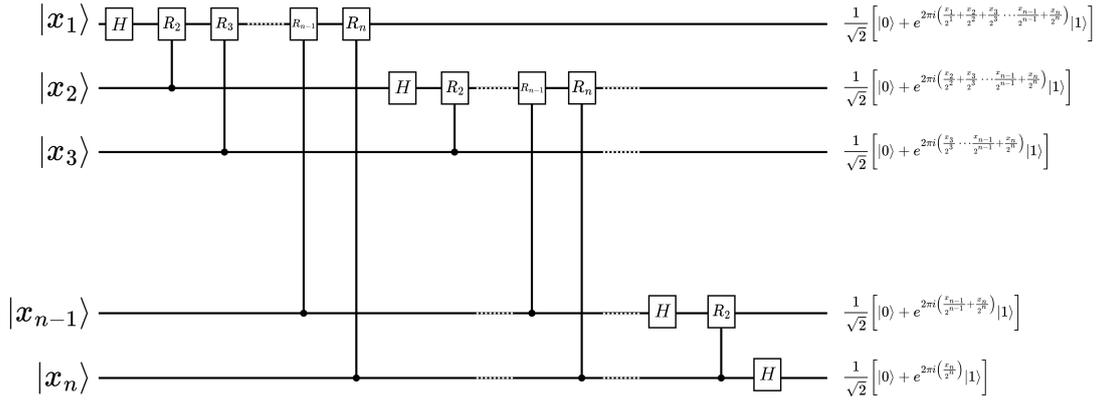


Figure 1.6: **Quantum Fourier Transform algorithm.** Quantum circuit representation of the Quantum Fourier Transform algorithm described in the main text, showing the conditional rotational gates over the qubits of the circuit and the phase accumulation on the state in the right side according to the width of the circuit.

Shor's factorization algorithm

Created by P. Shor in 1994 [Sho94], this algorithm aims to factor a given integer (N) into prime numbers, providing an efficient quantum solution for a hard computational problem, see fig. 1.7. This algorithm's significance lies in its ability to be executed on a quantum computer in polynomial time, approximately $O(\log(N))$, which holds profound implications for decrypting cryptographic keys.

Shor's algorithm consists of two parts. The first can be carried out on a classical computer to reduce the factorization problem to a problem of *finding the order* (determining the cardinality of the group for each factor). The second part involves applying the quantum algorithm to solve the problem of "finding the order".

The classical part consist of the following steps:

C.1 Choose a pseudo-random number $1 < a < N$.

C.2 Calculate the greatest common divisor between a and N i.e., $\gcd(a, N)$.

- If the greatest common divisor is not equal to 1, then it is a non-trivial factor of N and the cycle closes.
- Otherwise, use the quantum algorithm to find the order. To find the order or period r of the following function:

$$f(x) \rightarrow a^x \equiv \pmod{N},$$

that is, the smallest positive integer r for which the condition holds:

$$f(x + r) = f(x).$$

C.3 If r is odd, return to step 1.

C.4 If $a^{r/2} \equiv -1 \pmod{N}$, return to step 1.

C.5 The greatest common divisors of $a^{r/2} \pm 1$ and N that are non-trivial factors of N . On the contrary go back to step **C.1**.

Subsequently, the quantum part of the algorithm follows the steps listed below. The schematic representation of the quantum circuit is shown in fig. 1.7.

Q.1 Create the initial state register $|R\rangle$ as a superposition of S states.

$$|R\rangle = S^{-1/2} \sum_{x=0}^{S-1} |x\rangle|0\rangle,$$

where S satisfies $N^2 \leq S \leq 2N^2$.

Q.2 Construct the quantum function $f(x)$ and apply it to the initial register, which will remain a superposition of the S states.

$$|R'\rangle = S^{-1/2} \sum_{x=0}^{S-1} |x, f(x)\rangle.$$

Q.3 Apply the Quantum Fourier Transform to the register, which is defined as:

$$U_{\text{QFT}}|x\rangle = S^{-1/2} \sum_{y=0}^{S-1} e^{2\pi ixy/S} |y\rangle. \quad (1.25)$$

such that we obtain:

$$U_{\text{QFT}}|R'\rangle = S^{-1} \sum_{x=0}^{S-1} \sum_{y=0}^{S-1} e^{2\pi ixy/S} |y\rangle |f(x)\rangle.$$

A slightly more abstract view allows us to understand that each term in the first summation corresponds to different trajectories of the quantum interference.

Q.4 Perform the measurement, where a result is obtained with x_0 at the input and $f(x_0)$ at the output. Since $f(x)$ is periodic, the probability of measuring a certain y is given by:

$$S^{-1} \left| \sum_{x: f(x)=f(x_0)} e^{2\pi ixy/S} \right|^2 = S^{-1} \left| \sum_b e^{2\pi i(x_0+rb)y/S} \right|^2,$$

where it can be observed that the higher this probability, the closer the number $y \cdot r/S$ is to an integer.

Q.5 Convert y/S into an irreducible fraction, from which we can extract the denominator r' , which is a potential candidate for the value of the order r .

Q.6 Verify that $f(x) = f(x + r')$. If this condition is met, the algorithm terminates. Otherwise, obtain more candidates for r using values close to y or multiples of r , and repeat the process.

Q.7 If no candidates are found, return to step **Q.1** of the algorithm.

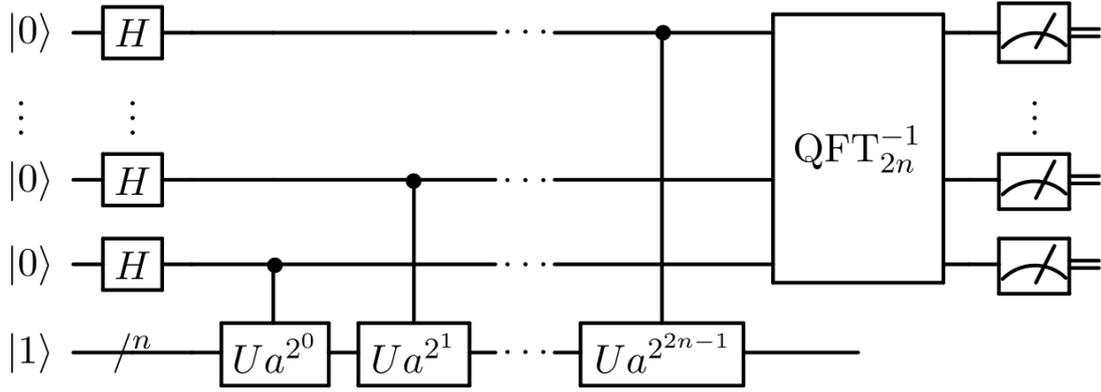


Figure 1.7: **Shor's factorization algorithm.** Quantum circuit representation of Shor's factorization algorithm described in the main text composed by the sequence of conditional rotational gates and a final QFT algorithm implemented in chapter 3.

Grover's search algorithm

This algorithm consists of obtaining a specific desired output (a state) from a generic input (a set of data) [Gro96], see also fig. 1.8. Its utility lies in searching for particular data within a database, which generally may be unordered. On a classical computer, this procedure would require a number of evaluations on the order of the number of elements in a database (N), while its quantum counterpart would only require a number of evaluations of the order of \sqrt{N} . To carry out this algorithm, it is important to define three objects:

- **Oracle** (U_w). It represents a function that distinguishes the state to be searched, which we will denote as the winner.

$$f(x) := \begin{cases} 0 & \forall x \neq w \text{ loser} \\ 1 & \forall x = w \text{ winner} \end{cases}, \quad (1.26)$$

such that it is possible to write the application of the oracle as:

$$U_w|x\rangle = (-1)^{f(x)}|x\rangle.$$

- **Uniform superposition** ($|s\rangle$). This is the initial state of all the data, without knowing the winner.

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad (1.27)$$

- **Operators** (U_s and U_w). These operators allow the winner to be identified through a reflection in the complex plane \mathbb{C}^n generated by $|s\rangle$ and $|w\rangle$ (winner), in addition to "blurring" the probabilities of the non-winners in order to obtain the winner clearly.

$$U_s = 2|s\rangle\langle s| - \mathbf{1}, \quad (\text{Diffusion operator}) \quad (1.28a)$$

$$U_w = \mathbf{1} - 2|s\rangle\langle s|, \quad (\text{Reflection operator}) \quad (1.28b)$$

where the projections correspond to:

$$\begin{aligned} \langle w|s\rangle &= \frac{1}{\sqrt{N}}, \\ \langle s|s\rangle &= 1, \\ \langle w|x\rangle &= 0. \end{aligned}$$

The steps to follow to execute Grover's algorithm, as depicted in fig. 1.8, are:

A. *Initialize the system state.* Here all possible states have the same probability.

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

B. Apply $r(N)$ iterations (converge asymptotically to $(\mathcal{O}\sqrt{N})$)

(a) Apply the reflection operator U_w

$$\begin{aligned} U_w|s\rangle &= (\mathbf{1} - 2|w\rangle\langle w|)|s\rangle = |s\rangle - 2|w\rangle\langle w|s\rangle, \\ &= |s\rangle - 2\langle w|x\rangle |w\rangle - 2\langle w|w\rangle|w\rangle, \\ &= |s\rangle - \frac{2}{\sqrt{N}}|w\rangle. \end{aligned} \quad (1.30)$$

(b) Apply the winner selector operator U_s (see fig. 1.8 (right))

$$\begin{aligned} U_s \left(|s\rangle - \frac{1}{\sqrt{N}}|w\rangle \right) &= (2|s\rangle\langle s| - \mathbf{1}) \left(|s\rangle - \frac{1}{\sqrt{N}}|w\rangle \right), \\ &= 2|s\rangle\langle s|s\rangle - |s\rangle - \frac{4}{\sqrt{N}}|s\rangle\langle s|w\rangle + \frac{2}{\sqrt{N}}|w\rangle, \\ &= 2|s\rangle - |s\rangle - \frac{4}{\sqrt{N}} \frac{1}{\sqrt{N}}|s\rangle + \frac{2}{\sqrt{N}}|w\rangle, \\ &= \left(1 - \frac{4}{N} \right) |s\rangle + \frac{2}{\sqrt{N}}|w\rangle, \\ &= \left(\frac{N-4}{N} \right) |s\rangle + \frac{2}{\sqrt{N}}|w\rangle. \end{aligned} \quad (1.31)$$

C. Execute the *measurement* gate.

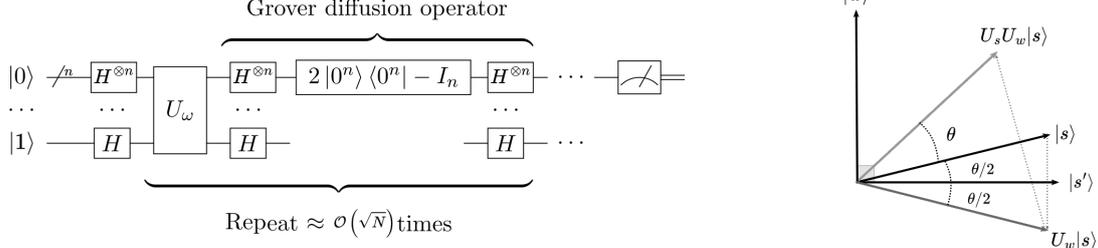


Figure 1.8: **Grover's search algorithm.** Quantum circuit representation of Grover's search algorithm described in the main text and the geometrical visualization of the state vector evolution along the steps of the algorithm of the searched state.

1.2.7 Quantum error correction

In quantum information processing as well as in any model of computation it is important to consider the presence of errors in the manipulation, communication and storage of information. For the sake of completeness, I briefly explain the basis of quantum error correction, which is later linked with the possible application of quantum native gates in the Rydberg platform (see chapter 2).

Due to the fragile nature of the quantum systems used for quantum computers and simulators, the units of information are prone to be affected by their environment and in some case by the internal dynamics of the system. These effects produce undesired state evolutions and therefore must be corrected. The noise channels of errors that can be corrected during the computation of generalized qubits are:

- **Bit-flip channel:** an error channel that consists of flipping the states $|0\rangle \leftrightarrow |1\rangle$ with probability $0 \leq p \leq 1$. This channel is given by the map over the system density matrix $\varepsilon(\hat{\rho}_s)$:

$$\varepsilon(\hat{\rho}_s) = (1 - p)\hat{\rho}_s + p\hat{\sigma}_x\hat{\rho}_s\hat{\sigma}_x. \quad (1.32)$$

- **Phase-flip channel:** the error channel consist in the introduction a phase over the qubit state $|1\rangle \rightarrow -|1\rangle$. The error channel is given by the map:

$$\varepsilon(\hat{\rho}_s) = (1 - p)\hat{\rho}_s + p\hat{\sigma}_z\hat{\rho}_s\hat{\sigma}_z. \quad (1.33)$$

- **Depolarizing channel:** it represents the evolution of the system or the noise driving the quantum states into a completely mixed state i.e., states represented out of the shell of the Bloch Sphere:

$$\varepsilon(\hat{\rho}_s) = (1 - p)\hat{\rho}_s + \frac{p}{2}\mathbf{1}. \quad (1.34)$$

- **Amplitude-Damping channel:** this channel relates to the energy dissipation phenomena in the quantum system. e.g., spontaneous photon emission

$$\varepsilon(\hat{\rho}_s) = \epsilon_0\hat{\rho}_s\epsilon_0^\dagger + \epsilon_1\hat{\rho}_s\epsilon_1^\dagger, \quad (1.35)$$

where

$$\epsilon_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}, \quad (1.36)$$

$$\epsilon_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}. \quad (1.37)$$

However, this is a simplification of the possible errors channels in real systems. For example, additional channels related to physical phenomena such as the loss of a trapped atom. These are known as erasure errors, that imply a complete loss of the qubit and requires its full reset.

As in the classical theory of error correction, the redundancy of information is a key element for the detection and correction of quantum errors. Moreover, the development of error correction codes in both classical and quantum theories is an interesting and vastly active field of research [Con22].

The ideas behind the techniques for error correction that are easily translated to the quantum version can be shown with a simple example of using a logical encoding that later will be defined as a logical bit or logical qubit, where 3 bits or qubits are used to encode a logical 0 or 1. The units of information are streamed through a circuit with a certain probability that the bits change from 0 to 1 with a probability $p = 0.1$ and a probably $1 - p$ to be changed from 1 to 0.

Thus the calculation follows the probabilities to obtain the logical 0 or 1 depending on the number of errors occurring in the logical bit/qubit. In this sense it is clear that for the 000 the probability is $(1-p)(1-p)(1-p) = (1-p)^3$, as for one bit/qubit error probability 100 or 010 or 001 the probability follows $p(1-p)^2 + p(1-p)^2 + p(1-p)^2 = 3p(1-p)^2$, similarly for two bits/qubits errors is $p^2(1-p)$ and finally for three bits/qubits the probability is p^3 . Therefore, the role of the logical bit/qubit come to play by being the probability of having the logical 0 with probability of the first two cases of errors i.e., $p_0 = (1-p)^3 + 3p(1-p)^2$ and for the logical 1 the last two cases i.e., $p_1 = 3p(1-p)^2 + p^3$. This calculation leads to the fact that the probability of having a logical 0 is higher than the single bit 0, likewise for the logical 1.

In the analysis the process is defined as a code, where 4 stages are defined: encoding, detection, correction and decoding, with the error channels applied in between the encoding and the detection. The redundancy consists of two ancilla qubits, which are corrected accordingly with the parity measurement made over the syndrome error qubits. This code for bit-flip and phase-flip errors can be represented by the circuits depicted in fig. 1.9.

This circuit represents an example of error correction over a logical qubit made of three physical qubits. Error correction codes can be generalized for larger system, leading to codes based on stabilizer operators for example on those based in Kitaev codes [Kit97] such as: surface code [Fow12]. There are also other types known as repetition code, like the popular color code [Lan11] in the trapped ion community.

Quantum error correction schemes for the atomic platform has been proposed [Luk18], while practical implementations are still in its early days, some proof-of-concept for additional requirements has been demonstrated, for example mid-circuit measurement capability [Gra23b].

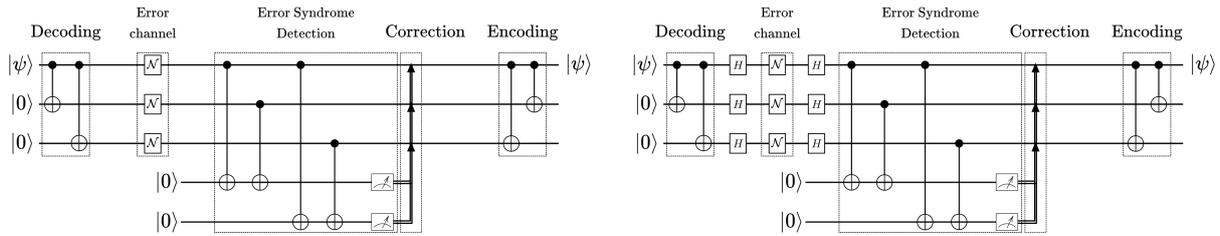


Figure 1.9: **Error correction circuits.** On the left the 3 (data) qubits Steane error correction code for the detection and correction of bit-flips and the phase flips in the right. Additional 2 qubits are added as ancillary for the detection of errors via syndrome detection, meanwhile the outcome of their measurements are used to execute classical conditional operations over the data qubits. A decoder and encoder gates are set to entangle and untangle the qubits at the beginning and the end of the code.

1.3 State-of-the-art in quantum information processing

The current effort on developing different qubits' architectures can be classified in different platforms with different advantages and challenges at the fundamental and technical level. However, the composition of an ultimate quantum machine will depend on the application and in the case of a Universal Quantum Computer (UQC) it is not clear if it will be composed by one or many architectures for different tasks within the machine, for instance: long-term storage, short-term storage, communication, fast and slow processing, non-demolition measurement, and so on.

1.3.1 Platforms

Here I summarize the different promising architectures for gate-based quantum processors with their current state-of-the-art and their advantages, in a comparative way.

- **Superconductors**

Superconducting qubits harness superconducting currents in electrical circuits at ultra-low temperatures using superconducting materials (e.g., Nb, Nb-Ti, Nb₃Sn, Al), representing a cutting-edge frontier in quantum computing. By exploiting the non-linearity of Josephson junctions (JJ) [Bar13] and microwave resonators, they offer promising controllability and scalability. Fabrication techniques such as deposition and lithography enable their production, positioning them as leading contenders for large-scale quantum processors.

The physics behind using superconducting circuits is to consider typical oscillating circuits e.g., the Inductor-capacitor (LC) circuit that represents a harmonic oscillator. However, in order to break the symmetry between the energy gaps, a non-linear element (Josephson junction) was introduced, creating the anharmonic oscillator needed to create qubits see fig. 1.10. Depending on the type of elements and the control properties of the qubit, this platform is divided in three main families of qubits: charge, flux and phase qubits.

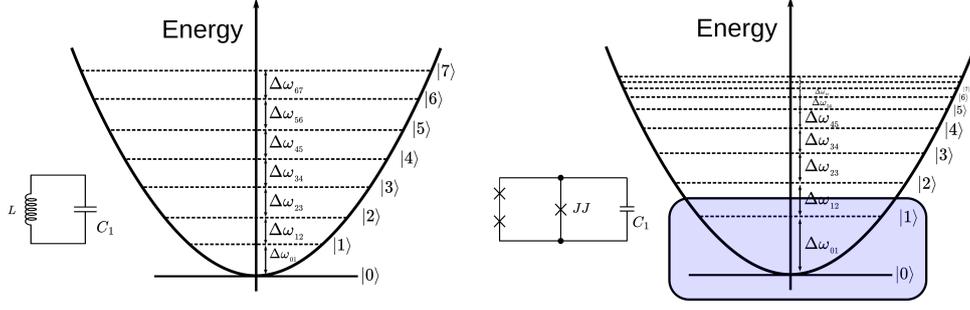


Figure 1.10: **Superconducting qubits.** Harmonic oscillator quadratic potential with the energy levels equally spaced in the left and anharmonic oscillator with a quadratic potential approximation to the cosine potential and the energy levels not equally spaced due to the anharmonicity of the JJ term in the Hamiltonian.

The qubit control independently on the type of superconducting qubit, is achieved by using microwave signals from a generator over a qubit drive after being mixed with radio-frequency frequencies (e.g., IQ-mixing). This type of control allows rotations over the x and y axis of the Bloch sphere and by tuning the phase with a virtual z rotations can be realized. The qubit-qubit interactions occurs with capacitive and inductive couplings or a combination of both, however the exact implementation depend on the exact design of the qubit.

Charge qubits: they are the most simple design of superconductors, and are often called “Cooper Pair Box” (CPB), the qubit is mainly composed of a capacitor and a JJ. When the circuit reaches very low temperatures i.e., 20mK the wavefunction of the Cooper pairs in the island can be described by the Hamiltonian in eq. (1.38). Where the eigenstates are given by Mathieu functions obtained from solving the eigenstate equation $\hat{\mathcal{H}}|\psi_k(\phi)\rangle = E_k|\psi_k(\phi)\rangle$ and selecting the lowest two states for the qubit states:

$$\hat{\mathcal{H}}_{CPB} = \underbrace{\frac{\hat{Q}^2}{2C}}_{\text{Capacitor}} - \underbrace{E_J \cos\left(\frac{2\pi\hat{\phi}}{\phi_0}\right)}_{\text{JJ}}, \quad (1.38)$$

here the $\hat{Q} = 2e\hat{n}$, $\hat{\phi} = 2\pi\hat{\phi}/\phi_0$ are the charge and flux operators in terms of the number of Cooper pairs operators \hat{n} and the electron charge e , C , E_J and ϕ_0 are the capacitance of the capacitor, the energy of the junction and the offset of the superconducting phase, respectively (see fig. 1.11).

Flux qubits: this kind of qubit family introduces an inductor in the system which changes the effective potential of the system. Similar to the charge qubit, the qubit states are defined by the lowest states of the Hamiltonian, representing the relative phase difference between the superconducting currents ($\sim 300\text{nA}$). This Hamiltonian is written in eq. (1.39):

$$\hat{\mathcal{H}}_{flux} = \underbrace{\frac{\hat{Q}^2}{2C_J}}_{\text{Capacitor}} + \underbrace{\frac{\hat{\phi}^2}{2L}}_{\text{Inductor}} - \underbrace{E_J \cos\left[\frac{2e}{\hbar}(\hat{\phi} - \Phi_{ext})\right]}_{\text{JJ}}, \quad (1.39)$$

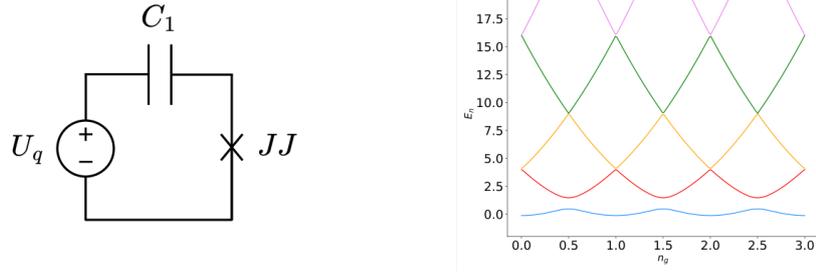


Figure 1.11: **Charge qubit.** Schematic of the qubit circuit and the eigenfunctions given by Mathieu functions as solutions of the Schrödinger equation, exhibiting a sweet spot where the separation between the first two states is maximal when setting an optimal value of the initial number of Cooper pairs in the island.

here the capacitance is assigned to the junction C_J , L is the inductance $\hat{\phi}$ is the flux operator and Φ_{ext} is the external magnetic flux quanta (see fig. 1.12).

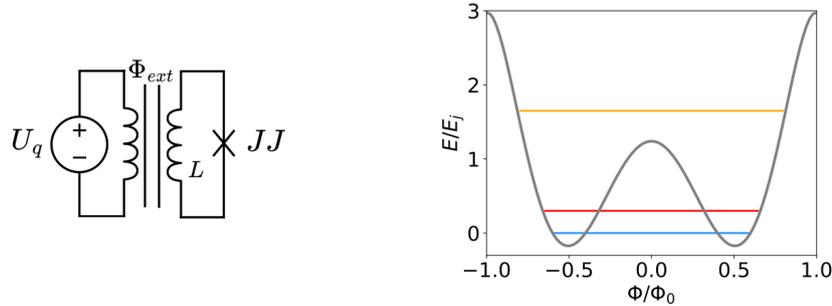


Figure 1.12: **Flux qubit.** Schematic of the qubit circuit with an inductor and the JJ, leading to a double-well potential obtain from its Hamiltonian.

Phase qubits:

A phase qubit utilizes a larger, current-biased Josephson junction compared to charge qubits. It operates in the zero-voltage state with a non-zero current bias. To enhance non-linearity, the biased current is typically maintained close to the critical current I_c of the junction. This configuration maximizes the sensitivity of the qubit to changes in the phase difference across the junction, enabling precise control and manipulation of its quantum state. The Hamiltonian is given by:

$$\hat{\mathcal{H}}_{\text{phase}} = \underbrace{\frac{E_{CJ}}{2e} \hat{Q}^2}_{\text{Capacitor \& Inductor}} - \underbrace{I \varphi_0 \hat{\delta}}_{\text{Supercurrent}} - \underbrace{I_0 \varphi_0 \cos(\hat{\delta})}_{\text{JJ}}, \quad (1.40)$$

where I is the supercurrent at the junction, φ_0 is a scale factor and $\hat{\delta}$ the phase difference operator. This leads to a washboard potential as shown in fig. 1.13.

Generally the backend of superconducting qubits is composed of a cryogenic system that cools down the system in stages, up to a temperature of 10mK in the last stage, where the superconducting quantum register chip is located, see fig. 1.14. This kind of setup shares similarities with the ones used in semiconductors.

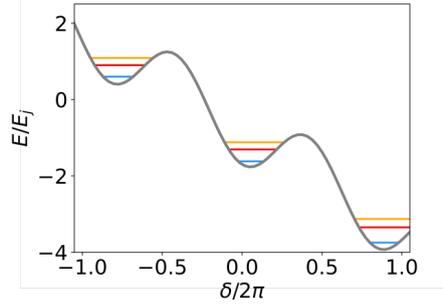
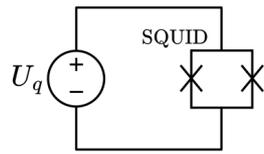


Figure 1.13: **Phase qubit.** Schematic of the qubit circuit with a voltage source and array of JJ or SQUID, and the washboard potential obtain from its Hamiltonian.

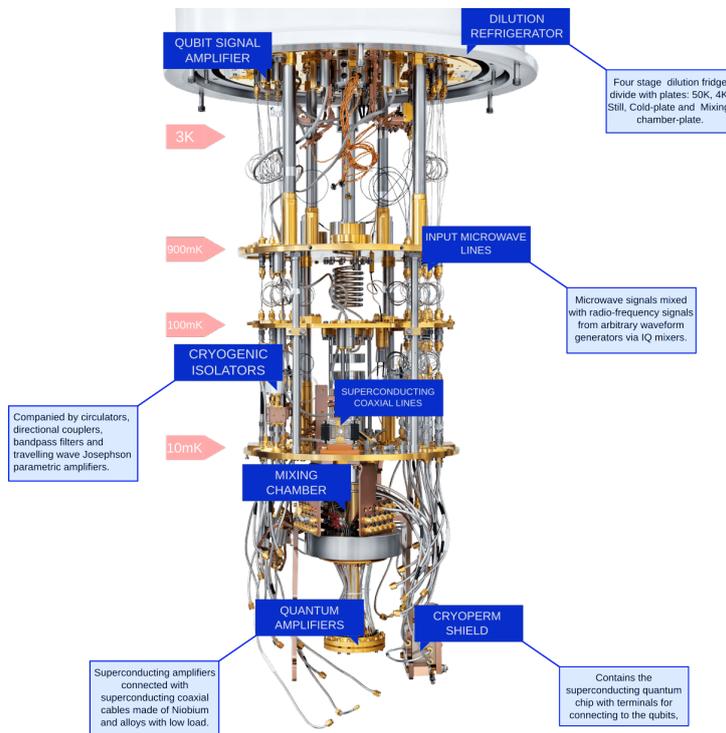


Figure 1.14: **Superconducting Platform.** Schematic of the superconducting apparatus composed mainly of the cryogenic setup fridge, where the superconducting chip with the qubits is hosted at the bottom of the fridge chandelier passing through a series of microwave components: attenuators, mixers and circulators from the top at 3K until the bottom with temperatures of $\sim 10\text{mK}$. Experimental setup from Rigetti computing Inc.

- **Semi-conductors**

Qubits based in this technology have a large number of variations e.g., single and multi spin states of electrons and nuclei spins states of doped semi-conductors. This technology is mostly enabled by the advanced fabrication techniques and expertise from the semiconducting chip community.

The general Hamiltonian that governs the qubits is given by electron-spin Hamil-

tonian (Heisenberg exchange) and the single-electron Zeeman term:

$$\hat{\mathcal{H}}(t) = \frac{1}{4} \sum_{\langle i,j \rangle} J_{i,j}(t) \hat{\sigma}_i \cdot \hat{\sigma}_j + \frac{1}{2} \sum_i g_i \mu_B \hat{B} \cdot \hat{\sigma}_i \quad (1.41)$$

A general classification of the qubits (see fig. 1.15) with this platform is:

- **Loss-DiVincenzo qubit**, which encodes the qubit in the spin-1/2 electron states. This is the simplest encoding of a qubit in the architecture where single- and two- qubits gates are achieved with static and oscillatory electromagnetic fields i.e., spin-spin coupling by exchange interaction and effective local magnetic field.
- **Donor qubit**, that encodes the qubit states in the nuclear spins of donor atoms in silicon e.g., ^{31}P making them highly coherent.
- **Singlet-triplet qubit**, is encoded in a group of electron multi-spin states where the two qubit states $|\uparrow\downarrow\rangle$ and $|\downarrow\uparrow\rangle$ define the qubit states. This is defined as flip-flop qubit too. These type of qubits are also controlled with static and oscillatory electric and magnetic fields or combinations of both and using capacitive coupling.
- **Exchange-only qubit**, is defined within a sub-space with a constant total spin. This type of qubit can take different forms by choosing high magnetic fields synchronized with the exchange coupling and by tuning the different couplings, enabling different sub-spaces of the system.

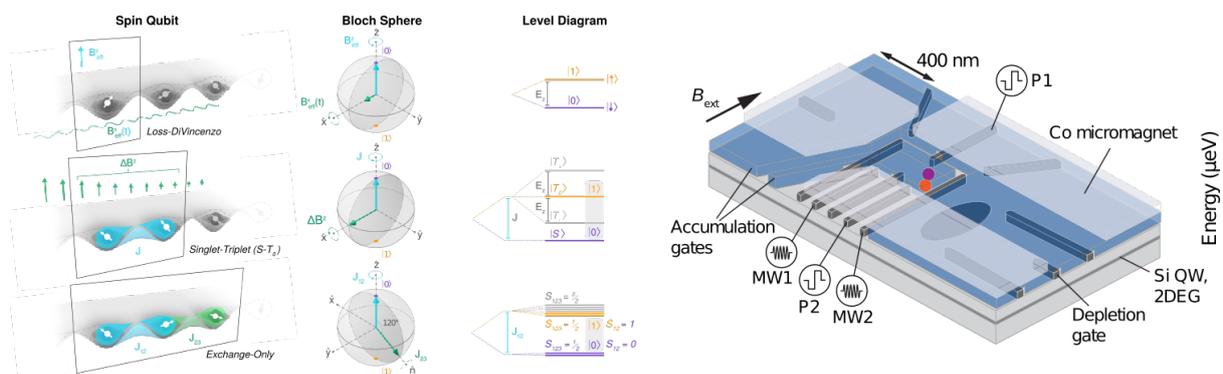


Figure 1.15: **Semiconducting platform.** Figure adapted from [Bur23] and [Wat18] showing the the types of qubit encoding in semi-conducting platforms with three main types of qubits encoding using the electronic, nuclear and spin states of the crystalline energy structure with their representation in the Bloch sphere and the energy level diagram. Schematic of the typical semi-conducting chip hosting the qubit at the nanometer scale with connections to microwave sources, as well as DC sources.

A much more detailed review and analysis of the technology of the platform can be found in [Bur23]. The chip where the qubits are, is shown in fig. 1.15(left), a more detailed diagram of the backend can be appreciated in [Sch22].

- **Trapped-ions**

Ion qubits were used as a pioneering qubit realistic model in gate-based quantum computing proposed by I. Cirac and P. Zoller [Cir95] and implemented by the group of R. Blatt [Sch03], where the quantum information is stored in the internal states of individual ions. These ions are typically trapped using electromagnetic fields in a device called an ion trap. Today's trap implementations are based on three types of technologies: static electric and magnetic potentials known as Penning traps [Bri12], static and oscillating electric fields known as Paul traps [Boh16] and planar traps with integrated optics which use RF and DC electrodes to confine the ions out of the chip's surface but close enough to control them. See fig. 1.16.

The ions are prepared by exciting atoms from the periodic table groups: IIA and IIB e.g., Be, Mg, Ca, Sr, Ba, Zn, Cd, Hg, and Yb. Where ^{40}Ca , ^9Be , ^{139}Ba and ^{171}Yb are currently the most used elements.

The qubit is encoded in the electronic or nuclear spin state of the ion, which can

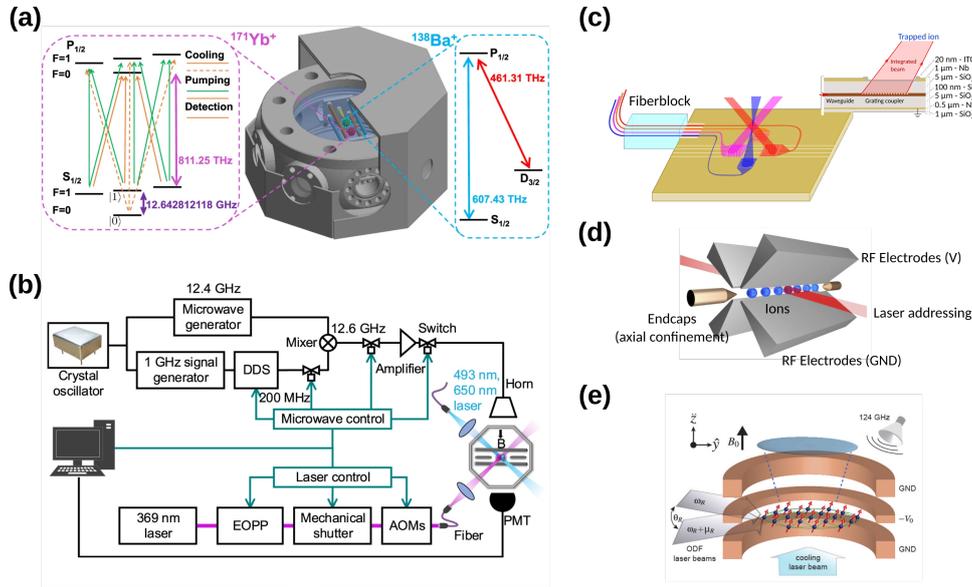


Figure 1.16: **Ion platform.** Vacuum system where the ion-chip substrate resides with the atomic element the representation of the meaningful atomic structure is presented for Yb^+ and Ba^+ . The experimental setup layout of a typical ion platform with the presence of the electronic control including, radio-frequency and microwave generators, locking control system and their interaction with lasers and opto-mechanical devices e.g., AOMs that prepare the light beams interacting with the atoms (Figure adapted from [Wan21]). An example scheme of a planar ion trap chip with integrated optics adapted from [Nif20] and a Paul trap made of 4 razor blades generating the RF field to control the ions together with perpendicular laser beams and endcaps for axial confinement, figure adapted from [Elt16]. Penning trap architecture for the ion platform with electrodes for DC fields and side addressing of the ions via laser beams, where additional control is realized with external antennas, figure adapted from [Bri12].

be manipulated using laser beams or microwave fields. In the trapped-ion community there are two types of qubit encoding: the **Optical qubit** with one qubit

state in a fundamental ground-state and the other one in a meta-stable state with a half-life of around a second with couplings achieved by laser beams in the visible spectrum e.g., 200-900nm; the **RF (hyperfine) qubit** on the other hand encodes the qubit states in two hyperfine ground-states of the atom that have very long half-life of around tens of minutes and coupling performed by RF and MW fields e.g., 1-100GHz. The state preparation includes auxiliary short-lived states that allows to pump the electrons to a fiducial ground-state. The qubit free Hamiltonian can be written as:

$$\hat{\mathcal{H}}_0 = \hbar\omega_z \hat{a}^\dagger \hat{a} + \frac{\hbar}{2} \omega_0 \hat{\sigma}_z, \quad (1.42)$$

where $\hat{\sigma}_z$ is the Pauli Z operator, ω_0 and ω_z is the transition and trap frequency respectively and \hat{a}, \hat{a}^\dagger are the annihilation and creation operators. Ion qubits have been studied extensively for their long coherence times [Wan21] and high-fidelity operations [Hog23], making them promising candidates for building scalable quantum computers. The control mechanisms for ionic qubits for single qubit operations are directly made by the optical and RF/MW fields. Meanwhile two or more qubit operations are based in three kinds of processes (see fig. 1.17): Blue sideband, Red sideband and Carrier explained later in this section, that consider the chain of ions as an harmonic oscillator with a phonon state or mode $|n\rangle$ from the interaction given by the Hamiltonian:

$$\hat{\mathcal{H}}' = \frac{1}{2} \hbar \Omega (\hat{\sigma}_+ + \hat{\sigma}_-) [e^{i(kz - \omega t + \phi)} + e^{-i(kz - \omega t + \phi)}], \quad (1.43)$$

here $\hat{\sigma}_+ (\hat{\sigma}_-)$ are the raising(lowering) operators, k, ω, ϕ are the wave-vector, frequency and phase of the laser and Ω is the coupling strength. The interaction processes, depicted in fig. 1.17, are given by the terms:

Blue sideband

If the laser detuning is chosen to be $\Delta = \omega_z$, this Hamiltonian induces transitions where the atoms get into the excited state and increases the phonon mode by 1 i.e., $|0, n\rangle \leftrightarrow |1, n+1\rangle$. The Hamiltonian is given by:

$$\hat{\mathcal{H}}_I^{(\text{bsb})} \approx \frac{1}{2} \hbar \eta \Omega (\hat{a}^\dagger \hat{\sigma}_+ e^{i\bar{\phi}} + \hat{a} \hat{\sigma}_- e^{-i\bar{\phi}}), \quad (1.44)$$

where η is a scale factor.

Red sideband

If the laser detuning is chosen to be $\Delta = -\omega_z$, this Hamiltonian induces transitions where the atom gets into the excited state and decreases the phonon mode by 1 i.e., $|0, n+1\rangle \leftrightarrow |1, n\rangle$. Hamiltonian can be expressed as:

$$\hat{\mathcal{H}}_I^{(\text{rsb})} \approx \frac{1}{2} \hbar \eta \Omega (\hat{a} \hat{\sigma}_+ e^{i\bar{\phi}} + \hat{a}^\dagger \hat{\sigma}_- e^{-i\bar{\phi}}). \quad (1.45)$$

Carrier

When the frequency of the laser matches the frequency of the atomic transition,

i.e., $\Delta = 0$, the vibrational levels remain unchanged, such that $|0, n\rangle \leftrightarrow |1, n\rangle$. Hamiltonian is given by:

$$\hat{\mathcal{H}}_I^{(c)} \approx \frac{1}{2} \hbar \Omega (\hat{\sigma}_+ e^{i\phi} + \hat{\sigma}_- e^{-i\phi}). \quad (1.46)$$

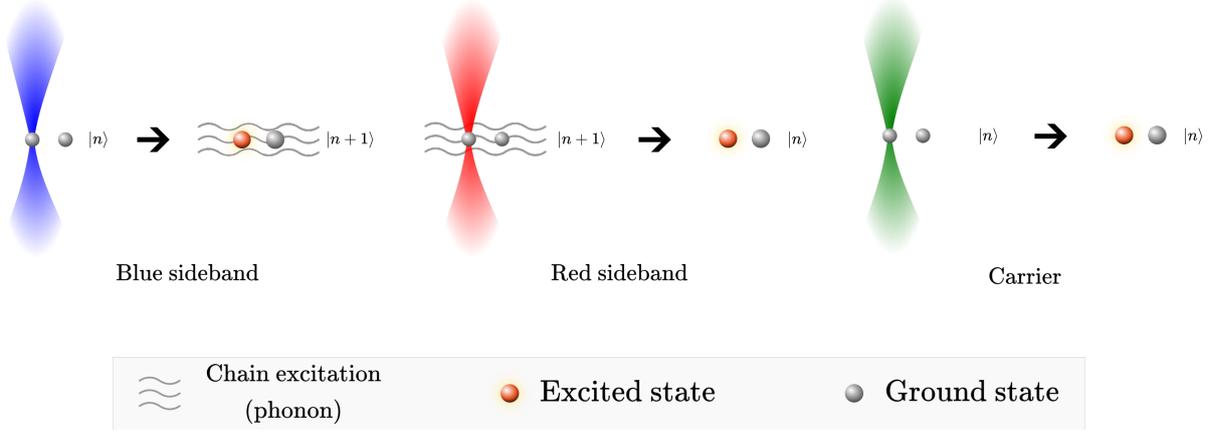


Figure 1.17: **Ion interaction mechanisms.** Blue sideband mechanism (left), driving the transitions with the excited state and an increase of one additional mode. Red sideband mechanism, driving the transition where the atom is excited and the mode decreases by one. Carrier mechanism, where driving the atom state to an excited state meanwhile the vibrational modes remain unchanged.

Detection of ion qubits can be implemented in a non-demolition fashion by using a short-lived auxiliary states that cannot decay to the second qubit state due to selection rules. A laser beam over the ion, couples the first qubit state and quickly decay producing a photon that can be detected only if it was initially in the first qubit state, not affecting the qubit if its state correspond to the second qubit state.

- **Cold-atoms**

This platform is the main topic of this work and a full description of the qubit encoding, physical implementation and gates can be found in chapter 2; the software system for the platform can be found in chapter 3, and in chapter 4 and chapter 5 the description of the apparatus for quantum computers and simulators as well as the quantum processor unit based in Rydberg-interacting cold-atoms quantum systems is discussed.

Additionally, it is important to highlight that this platform is particularly versatile for quantum computers and simulators since it shows the possibility to have single shot measurement to all qubits, similar to the multiplexing measurement in superconductors, as well as the controllable features using microwaves, when the atoms are in the Rydberg state manifold (see chapter 2) and fast gate operations of the nanosecond scale. This platform can also use optical transitions that allow operations with low crosstalks to idle qubits and large connectivity similar to the ion platform while still being able to shelve the qubits into isolated states for avoiding errors similar to semi-conducting architecture but with the possibility to still have modulated addressing for optimal control operations.

Platforms mentioned in this section with their advantages and disadvantages are summarized in table 1.4.

| Platform | $T_{\text{Coherence}}$ | τ_{gate} | Max. # qubits | Routing | Advantages | Technical challenges |
|-----------------------|--------------------------------------|---------------------------------------|---|---|--|---|
| Superconductor | $16.04 \pm 4 \mu\text{s}$ [Aru19] | 12 ns [Aru19] | 1121 (IBM condor) | Constraint by couplers and bus design. Surface acoustic waves. Charge transport. Super-exchange. Spin-SWAPs. Spin-CTAP. Capacitive coupling. Spin-chain coupling. | Industrial fabrication techniques and fast readout. Large family of qubits. | Scaling cryogenics setup. Increasing spectral resolution of electronics to increase fan-out control. Charge noise. |
| Semiconductor | $77 \pm 14 \mu\text{s}$ [Phi22] | ~ 200 ns [Phi22] | 6 [Phi22] | Nearest-neighbours. SWAP network qubit routing. | Natural integration with classical processors. Miniaturization due to small scale of the system (electrons). Large family of qubits. | Scaling cryogenics setup. Qubit interaction and fan-out control. Thermal noise, charge noise and charge motion due to micromagnet field gradients |
| Ions | 5500 ± 670 s [Wan21] | 480 ns [Sch18] | 56 (Honeywell #2) | All-to-all connectivity (shuttling) | Long coherence times. Highest gate fidelities. | Scaling the number of trapped ions with cross-talk during multi- and single-qubit operation. |
| Cold-atoms | 40 ± 7 s [Bar22] | 51 ns [Mad20] 6.51ns [Che22] | 1167 [Pau24] (^{85}Rb) 1225 [Nor24] (^{171}Yb) 6100 [Man] (^{133}Cs) | All-to-all connectivity (tunable) using dynamical rearrangement [Blu24] | Naturally scalable to $>10\,000$ qubits. Tunable interactions. Arbitrary geometries. Non-demolitive readout techniques. | Fast readout. Laser power. Fast real-time calibration close to gate times. |

Table 1.4: **Properties and features of quantum architectures.**

The four architectures discussed above, are candidates for QPU substrates. However, the QPU is controlled by classical electronics that are equipped with all the standard capabilities of manipulation, readout and feedback to the backend. Commercially there is a group of hardware solutions available that are based on fast FPGA waveform generation using Digital-to-Analog Converters (DAC), embedded in Printed Circuit Board (PCB) cards that are often set in electronic racks devices with multiple channels synchronised to each other. Additionally, these devices have digitizers commonly called Direct-Digital-Synthesizers (DDS) for driving the waveforms with RF frequency signals. More features are included in these systems such as DC-current generators for TTL signals and input signals using Analog-to-Digital-Converters (ADC). These technologies are shared among the different architectures for precise time-control of the system, where the requirements vary depending on the fundamental limits of the physics involved.

In the case of super- and semi-conductors the manipulation and readout share the same nature of being electrical signals, meanwhile current readout of ion and ultra-cold atoms can be controlled by electrical signals but the state-of-the-art is based on imaging (i.e., absorption and fluorescence imaging). However, other ways of readout of these systems as well as in photons are starting to use single photon sensors.

The requirements for the classical hardware depend on the intrinsic resonance frequency of the qubits, the gate time, readout time and the number of channels with amplitude, frequency and phase modulation. This is discussed in the next sections.

1.3.2 Classical control software

Classical hardware requires a low-level software that defines a set of instructions that allows it to set a sequence of electrical waveforms that are sent via different channels to the different components of the quantum device. The set of instructions for the classical

| Platform | Driving element | Carrier frequency | Time resolution (s) | Analog I/O | Digital I/O |
|-----------------|---|----------------------|---------------------|------------|-------------|
| Superconductors | Resonators | <12GHz | 10^{-9} | Yes | Yes |
| Semiconductors | Electrodes, resonators | <20GHz | 10^{-8} | Yes | Yes |
| Ions | Electrodes, optical modulators | <100GHz <500MHz | 10^{-4} | Yes | Yes |
| Cold-atoms | Optical modulators, RF- and MW-antennas | <500MHz (4-40)GHz | 10^{-9} | Yes | Yes |

Table 1.5: **Hardware features and control requirements.**

control hardware form part of the Firmware of the Quantum Software System that will be presented in chapter 3.

Nowadays, control software used on atomic quantum devices is not yet a full software system where diverse hardware, data management and analysis, simulation and monitoring are included in a single framework.

Instead, they are based on atom gas software experiments, LabVIEW programs or in-house solutions that do not consider all of the different features needed for quantum computers and simulators [Sta13] or even a generalized abstraction of a given experiment. Additionally, more tailored quantum computing software, for example Qiskit [Qis23] is more oriented towards quantum computing applications and development and does not offer features for configuration, connection and control of any arbitrary setup. Although there are currently efforts to implement an open-source unified software [Eft21], there are no implementations for ultra-cold atoms devices yet. Chapter 3 will present a quantum software system for atomic architectures that tackles this challenge, not just abstracting the backend and bridging to quantum computing tasks, but also connecting to modules for simulation and configuration that are aware of the QPU nature, besides taking into account tools for analyzing data at different levels.

It is natural that the software system of a quantum device will also evolve together with the backend. Therefore, when scaling quantum devices in, for example, the number of qubits, a software that is modular and extensible, would be required for accommodating all the operations for manipulation, calibrations, correction and measurement for all the qubits in the quantum device. In the long term, this means calculating waveforms for an arbitrary number of gate operations for millions of qubits, restricted to the physical constraints such as: connectivity, quantum circuit executed, Rabi couplings and detunings for the specific gate protocol for the qubit.

1.3.3 Classical control hardware

The quantum backend classical control electronics includes a series of devices to generate waveforms to orchestrate triggering and signal modulation going to the different components of the backend. Recent commercial solutions include controllers with <120 channels i.e., RF analog channels and I/O digital channels. These controllers can be concatenated

with a master synchronizing a set of these controllers.

These controllers for some solutions include the possibility to add arbitrary waveform and function generators, DC generators, direct-digital-synthesizers, clocks, microwave generators [Kas20] and more towards the direct integration to high-performance-computer and data-centers.

Some of these devices that are programmed to precisely control digital-to-analog-converters come with low-level instruction firmware [Mac], others with high-level GUI [Zür23] and many others with wrappers to serve as software-development-kits as Sinara from ARTIQ [Nie23] [Key20].

1.3.4 Read-out

Besides control, the qubit readout or measurement is also an important matter that plays a fundamental role when scaling in the number of qubits and moving towards a fault-tolerance quantum computer. From quantum mechanics, we know that it is not possible to directly measure a quantum state without collapsing the state, one solution is to implement quantum non-demolition measurements, that are being used to correct errors by doing live-feedback from this measurements back to the backend [Gra23b].

Current readout times for fluorescence imaging of atomic arrays are reported to be $< 800\mu\text{s}$ with 99.8% [Blu24]. Meanwhile in solid-state platforms this is around 140ns with 99.5% and 96.9% for two- and three-state readout fidelity in superconductors [Che23].

Dual-species designs will also contribute to the development of non-demolition readout by using an ancillary atomic qubit to execute readout [Ber17a].

1.3.5 Scalability

Increasing the number of qubits in the QPU is a challenging task where only a few groups have demonstrated the generation of arrays with >1000 atoms [Nor24] [Pau24] [Gyg24] [Man24]. Bigger arrays with tens of thousands of qubits are on the horizon since, they are at the moment just limited by the laser power.

Proposals for applying quantum gate operations over a large number of qubits include hybrid system between optical deflectors and light modulators [Gra23b] taking advantage of the fast deflection capabilities of the former and the large addressing and beam wave-front shaping of the latter.

Additionally, a large amount of processing might include atom shuttling for entanglement generation within these large arrays of atoms for creating logical qubits necessary for quantum error correction and fault-tolerance quantum computing schemes [Blu24].

Going to the scale of 10^5 qubits at the modularity of 10^3 qubits per module would require electronics with >200 channels, including the possibility of using quadrature modulation.

1.4 Main contributions described in this thesis

The main challenge in quantum computing with atoms addressed in this thesis is to establish a full-stack for controlling, in a standard framework, with all types of elements and instruments that a quantum device may require. This implies controlling static devices, that might be used for monitoring, as well as dynamically (controlled) devices that are used to ultimately control the quantum processor unit (QPU) via quantum connectors

which are those type of device that can serve as a bridge between the classical control and the quantum phenomena, for instance: acousto-optical modulators, electro-optical modulators, antennas etc.

As part of the quantum stack, the QPU is the main component. In chapter 2 I explore an emerging architecture that in the last 5 years has gained momentum, which is based on Rydberg states of ultra-cold atoms. I defined different ways to encode the minimal unit of quantum information: first by considering different atomic elements, considering different atomic levels and also different physical systems where I considered small atomic ensembles and single atoms as substrate for the creation of quantum units of information. I compile a series of demonstrated experimental protocols and put them in the context of a gate set for Rydberg-interacting atomic qubits, including a new 4-pulse gate operation based on Rydberg interactions.

In chapter 3, I show the developed Python software based on specifications that allows to: design protocols from low-level instructions (i.e., experimental sequences) to high-level instructions (i.e., quantum circuits), emulate a simplified model of the system and the protocol with a hardware-aware simulation of the quantum dynamics, monitor and management of the experiment and the data coming out of the device. It is an open-source software that allows different users modify it to their own flavour.

Controlling qubits represents another challenging topic. In chapter 4 and chapter 5, I developed and tested a global addressing microwave system that drive transitions between Rydberg states for manipulating the information encoded in arrays of qubits. This kind of global addressing is suitable for generating quantum entanglement and it is a key piece for useful quantum algorithms.

Finally, I contribute to the generation of large arrays of optical tweezers loaded with atomic ensembles and first measurements of loading single potassium atoms using inelastic collisions in a single optical tweezer that serve as substrate for QPUs of the platform with less complex dynamics and on which the gate protocols designed can be applied in a more natural way.

Quantum computing and simulation with ultra-cold Rydberg atoms

“The concepts ‘system’, ‘apparatus’, ‘environment’, immediately imply an artificial division of the world, and an intention to neglect, or take only schematic account of, the interaction across the split.”

S. Bell

In this chapter, I review the requirements of quantum devices, including simulators and computers. I also explore the necessities of a comprehensive quantum full-stack for the quantum backend of the Quantum Processing Unit (QPU), specifically based on ultra-cold atoms, as utilized throughout this work. The discussion encompasses the fundamental physics of the atomic architecture backend, the current state of both hardware and software, and the design philosophy employed for the new toolbox, which consists of building blocks for quantum devices utilizing Rydberg atoms

This section of the thesis chapter is based on insights and findings from our co-authored article, “Quantum simulation and computing with Rydberg-interacting qubits” - M.Morgado & S.Whitlock. AVS Quantum Sci. 3, 023501 (2021) [Mor21], serving as a key reference.

2.1 Review of criteria for quantum computers and quantum simulators

Quantum computers have been developed following the minimal criteria defined by D. Di Vincenzo [DiV00] and more recently complemented for simulators by I. Cirac & P. Zoller [Cir12]. See fig. 2.1 for an overview of the criteria.

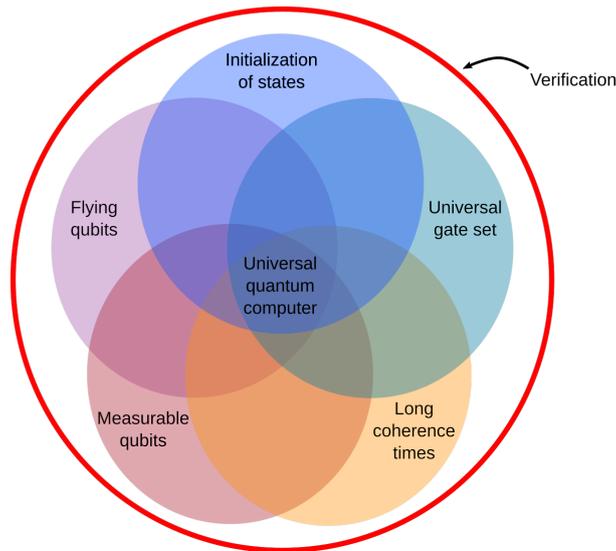


Figure 2.1: **Criteria for quantum computers and simulators.** Graphical representation of the Di Vincenzo criteria for quantum computers, complemented by I.Cirac & Zoller criteria for quantum simulators.

In the following these criteria are discussed in more depth:

A. *Physical system with well-defined qubits.*

The quantum physical system may consist of a collection of qubits, where each has two distinguishable state levels (e.g., the spin-1/2 *Up* and *Down* states of a quantum particle). Moreover, the term “well-defined” involves the physical parameters of the qubit being precisely known, such as the internal Hamiltonian determining its eigenstates, coupling with other qubits, and interaction with external fields that could be used for qubit state control. This control is commonly employed to ensure low probabilities in undesired energy levels.

In the case of quantum simulators, the requirement relaxes by just having quantum particles e.g., bosons, fermions, pseudo-spins that are located in the space and as a system has a large number of degrees of freedom.

B. *Ability to initialize qubits in a simple reference state, such as: $|000\dots\rangle$.*

For computation, the basic requirement is that qubits are initialized to a known value at the beginning of performing the algorithm or calculation. A second, slightly more technical reason, is that qubits with low entropy levels are required for applying quantum error correction.

For simulations, it is necessary that the state is known and that it can at least be prepared approximately.

C. *Prolonged decoherence time, much greater than the gate operation time.*

The concept of decoherence time is fundamental in this part of the criteria because it defines the characteristic time at which a qubit state (e.g., $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$) becomes a mixed state. This time, which characterizes the qubit's dynamics, may depend on the initial state and, in a special way, on the decay state known as the “leakage state.” A system may have different coherence times depending on its degrees of freedom. However, only one is required with which successful computation can be performed, and error correction has an acceptable fault tolerance (the ratio between gate execution time and qubit coherence time is in the range of $10^4 - 10^5$).

Although different non-unitary mechanisms e.g., dissipation of quantum systems limit the coherence time of the qubit, it is sometimes used in favor to engineer longer coherence times and therefore longer quantum computation and quantum simulation time.

D. *Compatible with a “universal quantum gate set”.*

As discussed in previous sections, quantum gates represent the key component of quantum computing, where quantum algorithms correspond to a sequence of unitary transformations acting on a certain number of qubits, which would simply leave the work of obtaining the corresponding Hamiltonian for each gate. However, this is a very simplistic view of reality, as deriving such Hamiltonians is not straightforward (especially for gates requiring 3 qubits, e.g., the Toffoli gate). Some error correction codes require a new set of elementary gate operations, acting in sets of physical qubits called “Logical Qubits”.

In the case of quantum simulators that are not gate-based this criteria corresponds to determining Hamiltonians, including those that cannot be simulated classically in an efficient way.

E. *Specific Qubit measurement capability.*

The last condition established by Di Vincenzo requires the possibility of measuring specific qubits. In an ideal measurement, a “0” is produced with probability p , and a “1” with probability $1 - p$ for a density matrix:

$$\hat{\rho} = p|0\rangle\langle 0| + \alpha|0\rangle\langle 1| + \alpha^*|1\rangle\langle 0| + (1 - p)|1\rangle\langle 1|.$$

Now, if the measurement is “non-destructive,” i.e., if we have a qubit in “0”, then this measurement leaves the qubit in the state $|0\rangle$, which can then be used for state preparation in the next requirement (although it can be fulfilled in other ways). Gottesman *et al.* [Got99] showed that if qubits are initialized in a maximally entangled state (e.g., Bell basis $|\Psi^\pm\rangle = |01\rangle \pm |10\rangle$, $|\Phi^\pm\rangle = |00\rangle \pm |11\rangle$), then having single-bit gates is sufficient.

F. *Verification.*

While this condition was not included in the original Di Vincenzo criteria, I.Cirac & P.Zoller include methods or techniques that allows to verify the outcome of the simulator or computer in order to check if the executed tasks are correct or not. This concept has recently settled the idea of a “Digital Quantum Twin”, borrowed from the system engineering field introduced first by M.Grieves [Gri17] and some initial efforts by Jaschke et al [Jak00].

These criteria do not define and include the need of a full quantum-stack (FQS) for these kind of machines. The criteria of a FQS is required for engineering and designing an architecture that is capable of achieving a fault-tolerant quantum device either for computation or simulation. By defining a FQS we set technical constraints that help define fundamental limits that shape the criteria of Di Vincenzo, Cirac & Zoller.

Therefore two elements that must be taken into account at the moment of designing and constructing these machines and should be included in the previous criteria are:

- **Fast feedback hardware** capabilities of the state machine for enabling error correction. Also includes a middleware that allows management and monitoring of the state for calibrations for a large number of units of information.
- **Software system** for handling the design of tasks at: high-level for example for the design of quantum circuits and algorithms, middle-level for design gate, simulation, readout and feedback protocols, and for low-level of the stack for specific protocols of preparation of the quantum information register.

Based on these criteria and considering the new added points, quantum machines can be classified in three main types, as depicted in fig. 2.2.

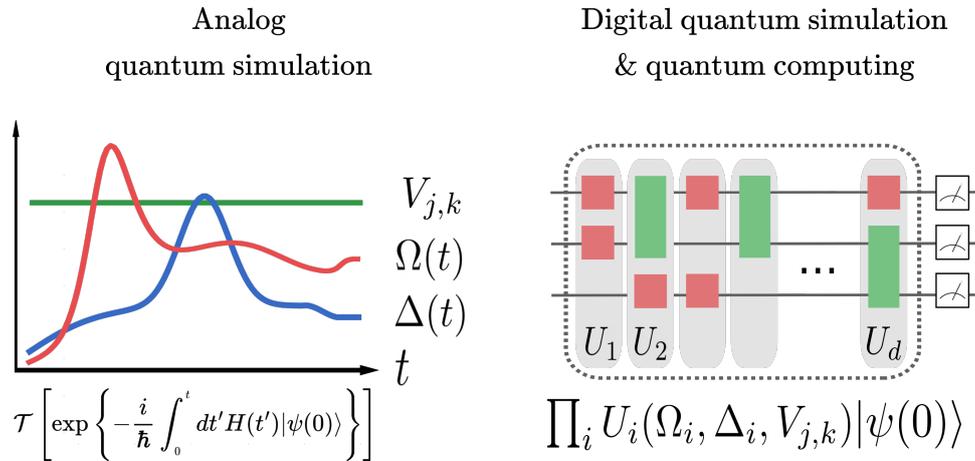


Figure 2.2: **Quantum devices classification.** Analog quantum simulators described in the main text with evolution of the system in a free manner (left). Digital quantum simulators and quantum computer with a discretized sequence of operations applied to a set of units of information (qubits).

- **Analog quantum simulators (AQS):** these kind of machines consist in a quantum physical system that mimics another quantum system or a specific class of models that can be reproduced on them, following closely a Hamiltonian.
- **Digital quantum simulators (DQS):** they are similar to AQS, but they encode the quantum state of the target system in a quantum register e.g., array of qubits, and simulate their time evolution by a sequential programmable quantum operations, which are not necessary to be unitary.

- **Quantum (gate-based) computers (QC):** these machines take DQS a step further, by making use of a set of fully addressable and controllable qubits, accompanied by a universal set of quantum logic gates operations, that allow to implement quantum algorithms for solving classically intractable computational problems [Ghe21] [Fit17].

A key difference between the different kinds of quantum machines is the level of controllability and “digitization” of the operations that can be performed. However, the control and implementation over ultra-cold atom platforms are constrained to the physics that govern the dynamics of the system, which is described in the next section.

2.2 AMO Physics

2.2.1 Atom-light Hamiltonian

In order to build the quantum gates protocols under the Rotating-Wave-Approximation (RWA), I introduce the radiation-matter interaction Hamiltonians, including the terms for the laser coupling and the laser detuning.

$$\hat{\mathcal{H}}_j^{ab} = \left(\frac{\Omega_j(t)}{2} e^{i\varphi_j(t)} |a\rangle_j \langle b| + \text{h.c.} \right) - \Delta_j(t) |b\rangle_j \langle b|. \quad (2.1)$$

Here $\hbar = 1$, for simplicity, and the $\Omega_j(t)$ is the Rabi coupling strength, proportional to the laser intensity between the atomic energy levels $|a\rangle_j$ and $|b\rangle_j$, the laser phase is given by the phasor $e^{i\varphi_j(t)}$; this term is Hermitian and therefore is accompanied by the Hermitian conjugated. The second term is diagonal, representing the detuning ($\Delta_j = \omega_j - \omega_0$) of the laser relative to the resonance frequency (ω_0) at the states $|b\rangle_j$. The index j corresponds to the atom index.

For the cases where the transition is dipole forbidden, the eq. (2.1) the Hamiltonian change to an effective form where the laser phase change to a relative phase, assuming that the laser between the two couplings are phase-coherent fields and making use of a large detuning (δ_e) in the intermediate state $|e\rangle$ for adiabatically eliminate it, the Rabi coupling changes to $\Omega \approx \frac{\Omega_a \Omega_b}{2\delta_e}$. The effective detuning of the system is then $\Delta = \Delta_{2ph} + (\Omega_a^2 - \Omega_b^2)/(4\delta_e)$, where the 2-photon detuning is given by: $\Delta_{2ph} = \omega_a + \omega_b - \omega_0$.

In a Λ -system additional dephase and decay mechanisms are given by the natural spontaneous decay rate (Γ) in the intermediate state and the coupling strength in each transition:

$$\gamma_{a(b)} = \frac{\Omega_{a(b)}}{4\delta_e^2} \Gamma, \quad (2.2)$$

for the states a and b .

On the other hand, semi-classical effects as the thermal motion of atoms that introduces random frequency shifts and dephasing is proportional to the effective wavevector and the atom velocity $|\vec{k}_a \pm \vec{k}_b| v_{th}$ which can be included in the phase factor of the laser.

2.2.2 Optical dipole traps

Development of optical tweezers enabled the trapping and manipulation of atoms in the space allowing to generate the arbitrary arrays of single atoms and small atomic ensembles

that serve as a substrate for qubits. This means that atoms were not longer constrained to the sub-wavelength distances exposed by optical lattices, which is a positive development in terms of tuneability of interactions. The optical tweezers are also described as out-of-resonance dipole traps, that can be described as an electric field \vec{E} with frequency ω which induces a dipole moment \vec{d} that oscillates at the field frequency:

$$\vec{E}(\vec{r}, t) = |\vec{E}| \exp\{-i\omega t\} \hat{e}_p + h.c., \quad (2.3)$$

where \hat{e}_p is the unitary polarization vector. Additionally, the dipole moment vector is described in a similar way since it is proportional to the field:

$$\vec{d} = \alpha \vec{E}, \quad (2.4)$$

where α is the complex polarizability. Due to the induced dipole moment, the atom experiences a conservative force which is proportional to the interaction potential:

$$\vec{F}_{\text{dip}}(\vec{r}) = -\nabla U_{\text{dip}}(\vec{r}) = \frac{1}{2\epsilon_0 c} \text{Re}(\alpha) \nabla I(\vec{r}), \quad (2.5)$$

where c is the speed of light, ϵ_0 the vacuum permittivity and $I = 2\epsilon_0 c |\vec{E}|^2$ is the field intensity, and $U_{\text{dip}} = -\frac{1}{2} \langle \vec{p} \vec{E} \rangle = -\frac{1}{2\epsilon_0 c} \text{Re}(\alpha) I$ is the interaction potential.

The scattering rate in a dipole trap can be understood as a continuous process of photon scattering from repeated photon absorption and subsequent spontaneous re-emission. The scattering rate from this process is described by:

$$\begin{aligned} \Gamma_{\text{sc}}(\vec{r}) &= \frac{P_{\text{abs}}}{\hbar\omega}, \\ &= \frac{1}{\hbar\epsilon_0 c} \text{Im}(\alpha) I(\vec{r}), \end{aligned} \quad (2.6)$$

where the power absorbed and the remitted dipole radiation is:

$$P_{\text{abs}} = \langle \vec{p} \vec{E} \rangle = 2\omega \text{Im} \left(\vec{d} | \vec{E} \right) = \frac{\omega}{\epsilon_0 c} \text{Im}(\alpha) I. \quad (2.7)$$

The polarizability can be obtained from considering the simple model of damped forced oscillator e.g., $\ddot{x} + \Gamma_\omega \dot{x} + \omega_0^2 x = -eE(t)/m_e$. Solving the dynamics for x the polarizability expression obtained is:

$$\alpha = \frac{e^2}{m_e \omega_0^2} \frac{1}{-\omega^2 - i\omega\Gamma_\omega}, \quad (2.8)$$

where the damping rate is given by:

$$\Gamma_\omega = \frac{e^2 \omega^2}{6\pi\epsilon_0 m_e c^3}. \quad (2.9)$$

The last expression is equivalent to the semi-classical result when calculating the dipole matrix element of allowed transitions considering selection rules and neglecting the saturation effects in a two-level system. Additionally, this expression can be simplified when considering a dipole potential in the rotating wave approximation (RWA) that leads to the general expression for the potential:

$$U_{\text{dip}}(\vec{r}) = \frac{3\pi c^2}{2\omega_0^3} \frac{\Gamma}{\Delta} I(\vec{r}). \quad (2.10)$$

and the scattering rate:

$$\Gamma_{\text{sc}}(\vec{r}) = \frac{3\pi c^2}{2\omega_0^3} \left(\frac{\Gamma}{\Delta} \right)^2 I(\vec{r}). \quad (2.11)$$

This expressions and concept are later considered in our experiment where we generate and use dipole traps to hold a large (red detuned i.e., $\Delta < 0$) atomic cloud from where we load tightly focused dipole traps (or also called optical tweezers) that we use to hold the substrate for atomic qubits made of tens of atoms or blue detuned with single atoms ($\Delta > 0$).

2.2.3 Quantum master equation

Another important tool when treating atomic systems is the use of quantum master equation formalism, that represent a way to describe the system and solve their dynamics. In fact, the emulator module of the software developed in this work is based in this mathematical formalism (see chapter 3). Here the quantum master equation, equivalent to the Schrödinger's equation but includes dissipative terms linked to dephasing and spontaneous decay. The general expressions is:

$$\dot{\hat{\rho}}(t) = -\frac{i}{\hbar} [\hat{\mathcal{H}}, \hat{\rho}] + \mathcal{L}(\hat{\rho}), \quad (2.12)$$

where $\hat{\rho}$ is the density matrix of the system and $\mathcal{L}(\hat{\rho})$ are the Lindbladians or also known as Lindblad superoperator:

$$\mathcal{L}(\hat{\rho}) = \sum_i \gamma_i \left(L_i \hat{\rho} L_i^\dagger - \frac{1}{2} \{ L_i^\dagger L_i, \hat{\rho} \} \right). \quad (2.13)$$

Here the L_i, L_i^\dagger are the “jump” operators that describes that describes the non-unitary terms of the dynamics, meanwhile the symbols $\{A, B\} = AB + BA$ represent the anti-commutator. For the case of a three-level system the Hamiltonian can be written in the RWA as:

$$\hat{\mathcal{H}} = (\hbar/2) \begin{pmatrix} 0 & \Omega_1 & 0 \\ \Omega_1 & -2\Delta_1 & \Omega_2 \\ 0 & \Omega_2 & -2(\Delta_1 + \Delta_2) \end{pmatrix} \quad (2.14)$$

where Ω_i are the coupling strengths between two of the atomic states and Δ_i are the detunings at the RWA. The non-unitary dynamics are then described by the Lindblad superoperator or often in this work called Lindbladian. This super operator describe the mechanisms such as the spontaneous decay and the dephasing terms. As matrix form the spontaneous decay term of the Lindbladian is:

$$\mathcal{L}(\hat{\rho}) = \begin{pmatrix} \Gamma_1 \hat{\rho}_{ee} & -\frac{1}{2} \Gamma_1 \hat{\rho}_{ge} & -\frac{1}{2} \Gamma_2 \hat{\rho}_{gr} \\ -\frac{1}{2} \Gamma_1 \hat{\rho}_{eg} & -\Gamma_1 \hat{\rho}_{ee} + \Gamma_2 \hat{\rho}_{rr} & -\frac{1}{2} (\Gamma_1 + \Gamma_2) \hat{\rho}_{er} \\ -\frac{1}{2} \Gamma_2 \hat{\rho}_{rg} & -\frac{1}{2} (\Gamma_1 + \Gamma_2) \hat{\rho}_{re} & -\Gamma_2 \hat{\rho}_{rr} \end{pmatrix}, \quad (2.15)$$

where Γ_i are the spontaneous decay rate at the i -state, $\hat{\rho}_{ij}$ is the ij -th density matrix element. The matrix can also be written with the Dirac's notation using the raising and

lower operators $\hat{\sigma}_i = \sqrt{\Gamma_i}|a\rangle\langle b|$. The Lindbladian term for the dephasing mechanism can also take the form of a matrix:

$$\mathcal{L}_d(\hat{\rho}) = \begin{pmatrix} 0 & -\gamma_1\hat{\rho}_{ge} & -(\gamma_1 + \gamma_2)\hat{\rho}_{gr} \\ -\gamma_1\hat{\rho}_{eg} & 0 & -\gamma_2\hat{\rho}_{er} \\ -(\gamma_1 + \gamma_2)\hat{\rho}_{rg} & -\gamma_2\hat{\rho}_{re} & 0 \end{pmatrix}, \quad (2.16)$$

where γ_i are the dephasing rate for the i -th state.

One can then now derive from the Optical Bloch Equation (OBE) by using the Lindbladians and Hamiltonian just above describe (eq. (2.12) with eq. (2.14), eq. (2.15) and eq. (2.16)), leading to the complex valued coupled differential equation system of first order, for the 3-level system:

$$\begin{aligned} \dot{\hat{\rho}}_{gg} &= \Gamma_1\hat{\rho}_{ee} + \frac{i}{2}(\Omega_1\hat{\rho}_{ge} - \Omega_1^*\hat{\rho}_{eg}), \\ \dot{\hat{\rho}}_{ee} &= -\Gamma_1\hat{\rho}_{ee} + \Gamma_2\hat{\rho}_{rr} - \frac{i}{2}(\Omega_1\hat{\rho}_{ge} - \Omega_1^*\hat{\rho}_{eg}) + \frac{i}{2}(\Omega_2\hat{\rho}_{er} - \Omega_2^*\hat{\rho}_{re}), \\ \dot{\hat{\rho}}_{rr} &= -\Gamma_2\hat{\rho}_{rr} - \frac{i}{2}(\Omega_2\hat{\rho}_{er} - \Omega_2^*\hat{\rho}_{re}), \\ \dot{\hat{\rho}}_{ge} &= -\left(i\Delta_1 + \frac{\Gamma_1}{2} + \gamma_1\right)\hat{\rho}_{ge} - \frac{i\Omega_1^*}{2}(\hat{\rho}_{ee} - \hat{\rho}_{gg}) + \frac{i\Omega_2}{2}\hat{\rho}_{gr}, \\ \dot{\hat{\rho}}_{er} &= -\left(i\Delta_2 + \frac{\Gamma_1 + \Gamma_2}{2} + \gamma_2\right)\hat{\rho}_{er} - \frac{i\Omega_2^*}{2}(\hat{\rho}_{rr} - \hat{\rho}_{ee}) - \frac{i\Omega_1}{2}\hat{\rho}_{gr}, \\ \dot{\hat{\rho}}_{gr} &= -\left(i(\Delta_1 + \Delta_2) + \frac{\Gamma_2}{2} + \gamma_1 + \gamma_2\right)\hat{\rho}_{gr} + \frac{i\Omega_2^*}{2}\hat{\rho}_{ge} - \frac{i\Omega_1^*}{2}\hat{\rho}_{er}. \end{aligned} \quad (2.17)$$

These equations describe the unitary and non-unitary dynamics of the quantum system, that can be used for engineer quantum gate protocols as shown in section 2.4.2 and section 2.4.3 with time-dependent Hamiltonians. In terms of simulations, these equations govern the numerical solutions in the emulator module of the software presented in chapter 3.

2.3 Rydberg Physics

2.3.1 Rydberg atoms

A Rydberg atom is an atom excited to a highly atomic state in which the most outer bounded of its electrons occupies an orbit far from the nucleus. These orbits are typically characterized by high principal quantum number $n \in [20 - 100]$, meaning the electron is in a state with high energy and far from the nucleus, similar to the Hydrogen model. Rydberg atoms exhibit unique properties due to their highly excited states, including large sensitivity to external electric and magnetic fields, long lifetimes, and the ability to interact strongly with other atoms and photons. J. Rydberg (1888), was the first in describe the spectral lines series associated with transitions between these highly excited states. The energy levels below the ionization threshold E_i series are described by the expression:

$$E = E_i - \frac{R_y^*}{(n - \delta_{nlj})^2}, \quad (2.18)$$

where the effective principal quantum number $n^* = n - \delta_{nlj}$ and δ_{nlj} is the Rydberg electron quantum defect. The quantum defects describes the increase of the binding energy of the atom relative to the hydrogenic energy, due to the penetration and polarization of the inner electron shells with low orbital angular momentum $\ell < 3$ and parameterized by the Ritz expansion:

$$\delta_{nlj} = \delta_0 + \frac{\delta_2}{(n - \delta_0)^2} + \frac{\delta_4}{(n - \delta_0)^4} + \dots, \quad (2.19)$$

where δ_i are the Rydberg-Ritz coefficients [Jas48]. For ^{39}K for high n the value of the quantum defect obtained from ARC software [Šib17]: $\delta(S_{1/2}) = 2.18(9)$, $\delta(P_{1/2}) = 1.72(7)$, $\delta(P_{3/2}) = 1.72(4)$, $\delta(D_{5/2}) = 0.27(6)$ (see fig. 2.3).

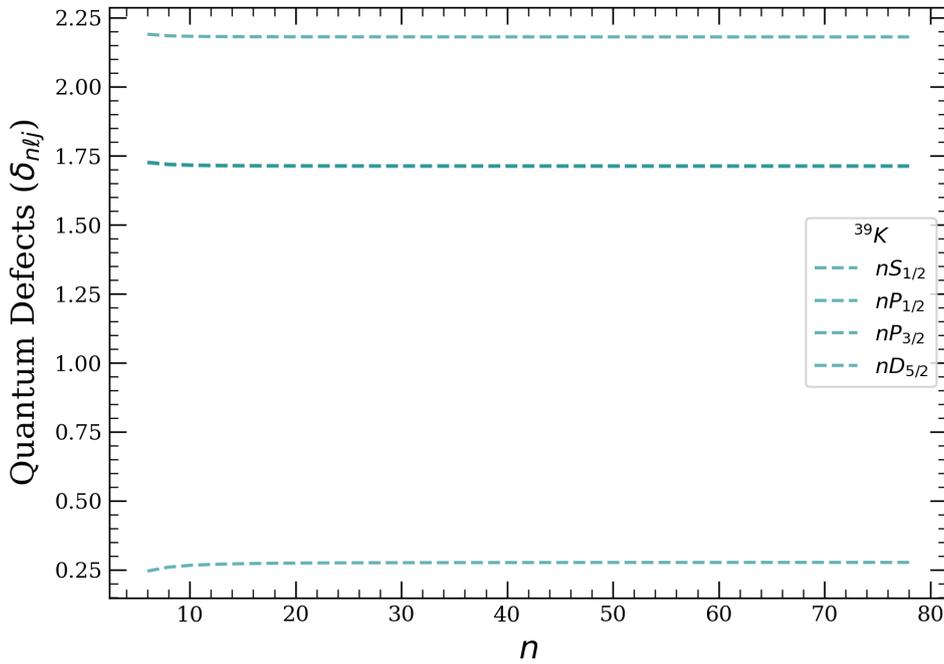


Figure 2.3: ^{39}K **Quantum defects**. Quantum defects for potassium-39 in the states $S_{1/2}$, $P_{1/2}$, $P_{3/2}$, $D_{5/2}$.

The Rydberg constant corrected for the reduced mass $R_y^* = \frac{m_e}{m+m_e} R_{y\infty}$ and the Rydberg constant is given by:

$$R_{y\infty} = \frac{m_e e^4}{8\varepsilon_0^2 h^3 c} = 1.097(37) \times 10^7 \text{m}^{-1}, \quad (2.20)$$

with the rest mass of the electron m_e , e is the elementary charge, ε_0 is the permittivity of free space, h the Planck constant and c the speed of light in the vacuum.

The size of a Rydberg atom is typically in the order of hundreds of nanometers ($\sim 400\text{nm}$), whereas the typical size of atoms in the ground-state is in the order of tenths of nanometers ($\sim 0.1\text{nm}$). This relative size difference is similar to the ratio in size between a football ball nr.5 ($\sim 0.7\text{m}$) and the size of the Real Madrid CF stadium ($\sim 228\text{m}$ long). The scale, in the case of potassium, can be seen in fig. 2.4.

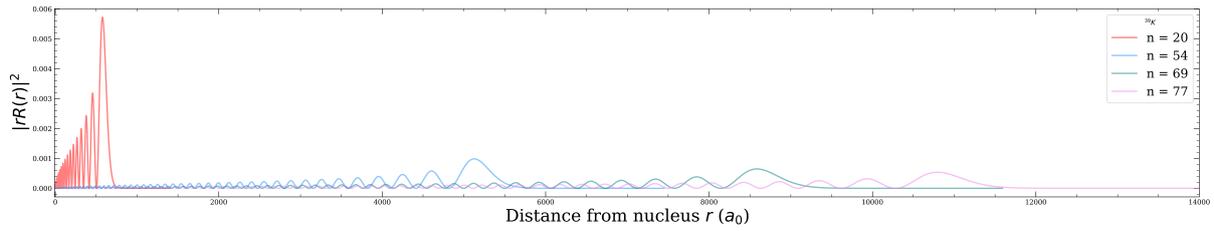


Figure 2.4: ^{39}K **Radial wavefunction**. Radial wavefunction of the Rydberg atom for the states with principal quantum number (n) 20, 54, 69 and 77 of potassium-39.

2.3.2 Rydberg properties

Rydberg states in atoms and molecules is one the most active topics in atomic physics, since they have versatile features when used in experimental setups, as for example: imaging and field-ionization readout, microwave manipulation and connection to optical transitions, relatively long time-scales constrained to the Rydberg lifetime and tuneable interactions.

One main characteristic of the Rydberg atoms is the dependency on the principal quantum number n over many of its properties summarized in table 2.1.

| Property | n Dependency |
|-------------------------------------|----------------|
| Atomic radius | n^2 |
| Binding energy | n^{-2} |
| Natural lifetime | n^3 |
| Ionization field | n^{-4} |
| Maximum permanent dipole moment | n^2 |
| Dipole-Dipole interaction frequency | n^4 |
| Van der Waals interactions | n^{11} |

Table 2.1: **Rydberg properties**. Properties of Rydberg atoms that depends strongly in the principal quantum number n , enabling the choice of the Rydberg state to be control parameter.

Dipole matrix elements

Many of the relevant properties and quantities of Rydberg atoms are obtained from the calculation of dipole matrix elements, which describes the expectation value of the electric dipole transition which are proportional to $|\langle e|\hat{E} \cdot \vec{r}|g\rangle|^2$, where $\{|e\rangle, |g\rangle\}$ are the eigenstates of the system in the written in the basis of some angular momentum quantum number. e.g., $|g\rangle = |\ell, m_\ell\rangle$. Generally, the dipole matrix elements are given by:

$$\langle n, \ell, m_\ell | r_q | n', \ell', m'_\ell \rangle = (-1)^{\ell-m_\ell} \begin{pmatrix} \ell & 1 & \ell' \\ -m_\ell & q & m'_\ell \end{pmatrix} \langle \ell || r || \ell' \rangle, \quad (2.21)$$

where the polarization $q = -1, 0, 1$, which corresponds to σ^+ , π and σ^- , which depends on the reduced matrix element given by:

$$\langle \ell || r || \ell' \rangle = (-1)^\ell \sqrt{(2\ell+1)(2\ell'+1)} \begin{pmatrix} \ell & 1 & \ell' \\ 0 & 0 & 0 \end{pmatrix} \mathcal{R}_{n\ell \rightarrow n'\ell'}, \quad (2.22)$$

with the round braces denoting the Wigner-3j symbols and $\mathcal{R}_{n\ell \rightarrow n'\ell'}$ is the radial matrix element:

$$\mathcal{R}_{n\ell \rightarrow n'\ell'} = \int_{r_i}^{r_o} R_{n,\ell}(r) r R_{n',\ell'}(r) r^2 dr, \quad (2.23)$$

where the atomic wavefunctions can be separated in a radial and spherical components i.e., $\psi(r, \theta, \phi) = Y_{\ell, m_\ell}(\theta, \phi) R_{n\ell}(r)$. Similarly, when highly excited states are considered then the hyperfine structure splitting can be neglected and therefore the fine structure splitting produced by the Hamiltonian term of the spin-orbit interaction i.e., $\hat{\mathcal{H}} = \beta \hat{L} \cdot \hat{S}$, which is described by the basis of (j, m_j) and the matrix elements:

$$\langle n, \ell, j, m_j | r_q | n', \ell', j', m'_j \rangle = (-1)^{j-m_j} \begin{pmatrix} j & 1 & j' \\ -m_j & q & m'_j \end{pmatrix} \langle j || r || j' \rangle, \quad (2.24)$$

and the reduced matrix element:

$$\langle j || r || j' \rangle = (-1)^{\ell+s+j'+1} \delta_{s,s'} \sqrt{(2j+1)(2j'+1)} \left\{ \begin{matrix} j & 1 & j' \\ \ell' & s & \ell \end{matrix} \right\} \langle \ell || r || \ell' \rangle \quad (2.25)$$

here the curly braces denote the Wigner-6j symbol. The dipole matrix elements of the transitions between states $4S_{1/2}$ and $4P_{1/2}$ is shown in the fig. 2.5.

Lifetime

Rydberg states are characterized of having relative long radiative lifetimes, for example states nS live in the order of tens of microseconds for $n \sim 30$. The lifetime of these states are link on the orbit of the electron around the nuclei, in other words on the orbital angular moment ℓ , scaling faster with n for circular Rydberg states. The lifetime is determined from the Einstein A-coefficients:

$$\frac{1}{\tau_0} = \sum_{n', \ell'} A_{n\ell \rightarrow n'\ell'}. \quad (2.26)$$

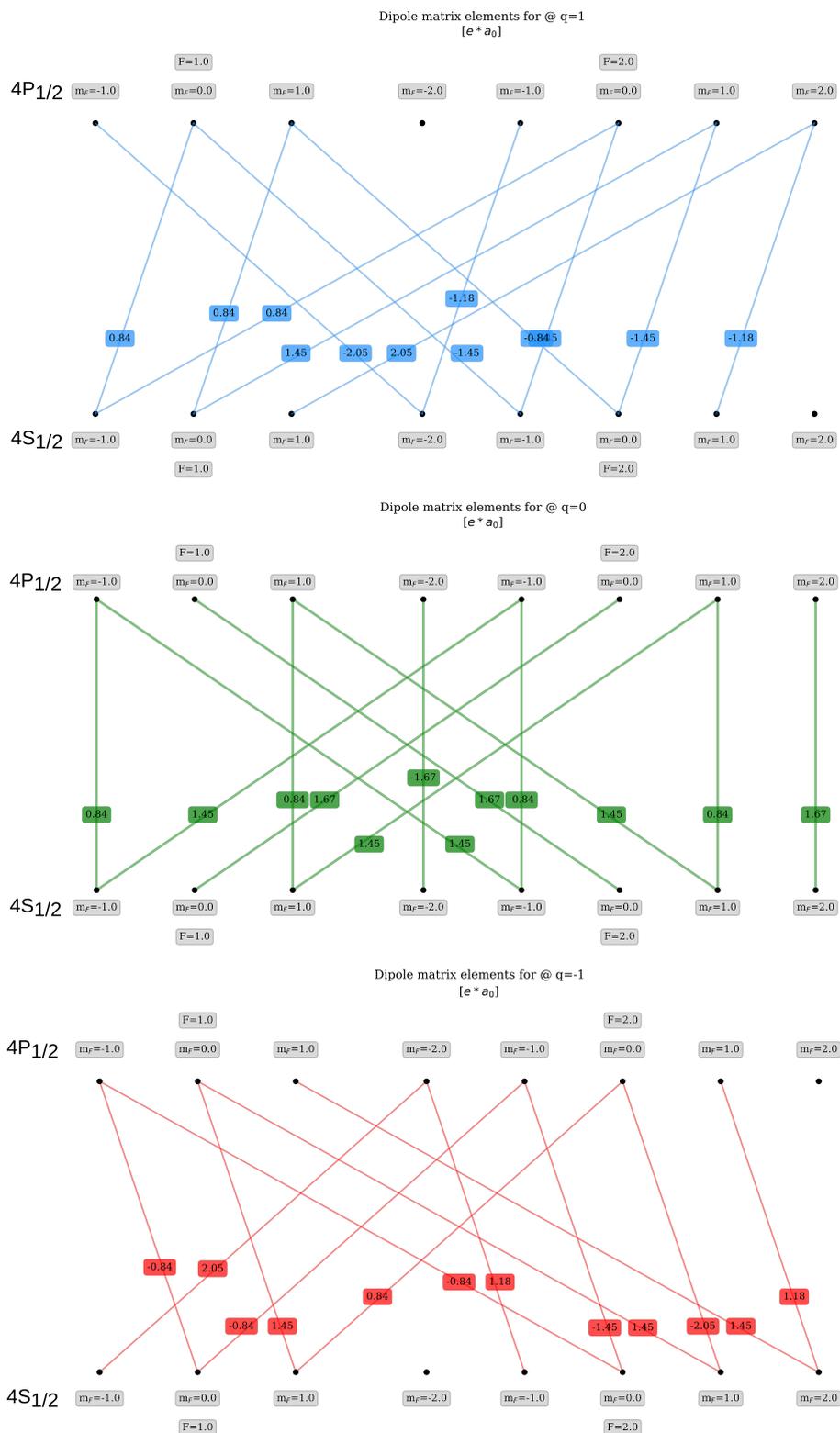


Figure 2.5: **Dipole matrix element.** Dipole matrix elements of potassium-39 for the states $4S_{1/2}$ and $4P_{1/2}$ and their sublevels for the different polarizations: $q = 0$ (green), $q = 1$ (blue) and $q = -1$ (red).

The sum run over all possible dipole coupled states that satisfy $E_{n'\ell'} < E_{n\ell}$ and the A-coefficient follows the expression:

$$A_{n\ell \rightarrow n'\ell'} = \frac{4\omega_{nn'}^3}{3c^3} \frac{\ell_{\max}}{2\ell + 1} \mathcal{R}_{n\ell \rightarrow n'\ell'}^2, \quad (2.27)$$

here the A-coefficient depends on the third power of the frequency transitions which results in the lowest energy transitions being the dominant decay channels for radiative decay.

On the other hand, at finite temperature the black-body radiation (BBR) becomes non-neglectable, reducing the lifetime of the state, see fig. 2.6. The BBR diffuses the state to upper and lower energy Rydberg states and it is described by the expression:

$$\frac{1}{\tau_{\text{BBR}}} = \sum_{n'\ell'} \frac{A_{n\ell \rightarrow n'\ell'}}{\exp(\omega_{nn'}/k_B T) - 1}, \quad (2.28)$$

where k_B is the Boltzmann constant and T the temperature. The BBR characteristics, as function of the principal quantum number “n” are depicted in fig. 2.6.

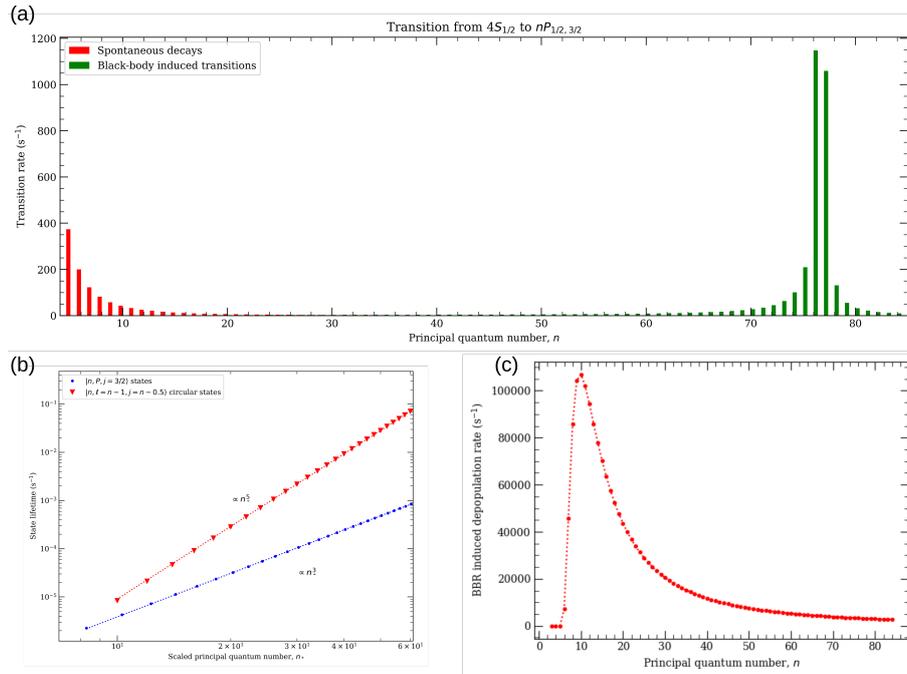


Figure 2.6: **Rydberg lifetime for ^{39}K** (a) Shows the finite temperature effects by showing the strength of the different decay channels for the state in ^{39}K at 300K showing the contribution from radiative decay (red columns) and black-body induced transitions (green). (b) Shows the lifetime in for a Rydberg state of the form $|ketn, P, j = 3/2\rangle$ with n^3 dependency similarly for a circular Rydberg atom $\ell = n - 1$ exhibit a dependency proportional to n^5 . (c) Induced state depopulation due to BBR in potassium-39.

This leads then to an effective lifetime determined by:

$$\frac{1}{\tau_{\text{eff}}} = \frac{1}{\tau_0} + \frac{1}{\tau_{\text{BBR}}}. \quad (2.29)$$

Polarizability

The polarizability of Rydberg atoms scales as n^7 , this is because if we re-write the scalar polarizability as function of the dipole matrix elements and the energy difference:

$$\alpha_0 = 2e^2 \sum_{n' \ell' j' \neq n \ell j} \frac{|\langle n, \ell, j, m_j | r_0 | n', \ell', j', m'_j \rangle|^2}{E_{n' \ell' j'} - E_{n \ell j}}. \quad (2.30)$$

where the dipole matrix element scales with n^2 and the energy difference as n^{-3} . The scalar polarizability for ground-states and a Rydberg state used in potassium-39 later in this work (see chapter 5) are shown in fig. 2.7.

Additionally transition within the Rydberg manifold using free-space microwaves are dipole coupled in the range of gigahertz. A map of these couplings between two Rydberg states of the form $n_1 S_{1/2} \leftrightarrow n_2 P_{1/2}$ is shown in fig. 2.7.

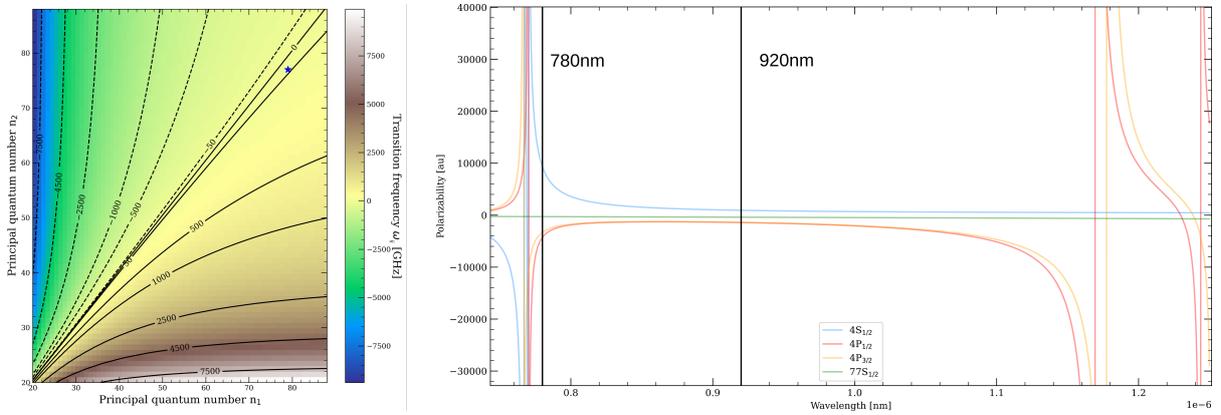


Figure 2.7: **Polarization and microwave (Rydberg) transitions in ^{39}K .** Microwave transitions frequency values to be use as the frequency of the field in order to couple the system and the polarization within the range of available lasers in the lab, distinguishing between the two wavelengths used for optical tweezer generation (780nm) and 920nm for single-atom trapping (see chapter 5.)².

2.3.3 Rydberg interactions

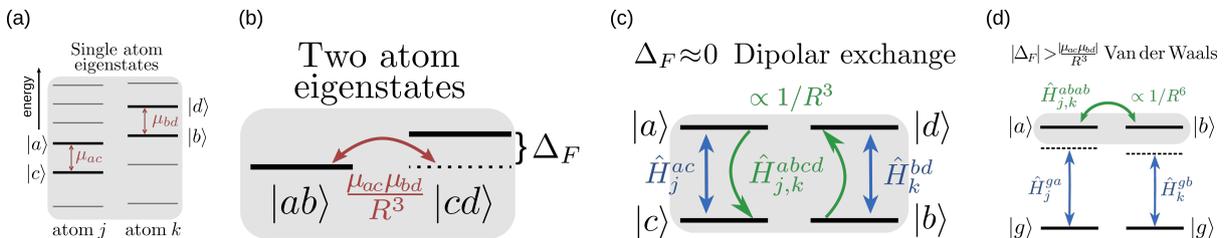


Figure 2.8: **Rydberg mechanisms.** (a) Rydberg manifold energy level diagram for two atoms with dipole-coupled pair states with corresponding dipole moments $\mu_{a,c}$ and $\mu_{b,d}$. (b) Pair state diagram for the two atom eigenbasis. (c) Interaction mechanism diagram for resonant interactions proportional to $1/R^3$ and (d) non-resonant interactions proportional to $1/R^6$.

The Rydberg atoms are known by long-range and tuneable interactions. This is used in quantum information processing to engineer Hamiltonians for many-body physics in the case of simulations and in the case of quantum computing it is used to engineer interaction terms that are used in two- and multi-qubit gates to perform conditional operations (using Rydberg blockade) and state transfer-like operations (using dipolar exchange). The general form of this Hamiltonian term is given by:

$$\hat{\mathcal{H}}_{j,k}^{abcd} = \frac{V_{j,k}}{2} |ab\rangle\langle cd| + \text{h.c.} , \quad (2.31)$$

where the states of the j and k are (a, c) and (b, d) respectively. $V_{j,k}$ is the interaction strength which depends on the distance between the atoms and the chosen Rydberg states³.

Rydberg interactions can be used to affect the motion of atoms but also to affect the internal state manipulation of the atoms due to their large transition dipole moment origin [Wüs11][Hag12][Sec16][Pło18][Gam20][Ska20]. The operator that describes the dipole-dipole interaction between two atoms j and k is given by:

$$\hat{V} = \frac{1}{4\pi\epsilon_0} \frac{\hat{\mu}_j \cdot \hat{\mu}_k - 3(\hat{\mu}_j \cdot \vec{n})(\hat{\mu}_k \cdot \vec{n})}{R_{j,k}^3}, \quad (2.32)$$

where $\hat{\mu}_j$ is the electric dipole operator over the atom j , \vec{n} is the unitary vector in the direction of the atom separation with distance $R_{j,k}$. This expression neglects the angular dependency of the dipole-coupled Rydberg states and assumes that only two dipole-coupled Rydberg states per atom e.g., $a \leftrightarrow c$ and $b \leftrightarrow d$ with similar transition energy $E_a - E_c \approx E_d - E_b$ (see fig. 2.8). The states before mentioned can be related to the single atom state basis or they could refer to Rydberg states that belong to different types of qubit encoding e.g., states from data and ancilla qubits used to encode a logical qubit.

The Hamiltonian of two atoms, written in the pair basis states i.e., $\{|ab\rangle, |cd\rangle\}$ is then:

$$\hat{\mathcal{H}} = \left(\frac{\mu_{ac}\mu_{bd}}{R_{j,k}^3} |ab\rangle\langle cd| + \text{h.c.} \right) + \Delta_F |cd\rangle\langle cd|, \quad (2.33)$$

where $\mu_{\alpha\beta} = |\alpha|^2 \hat{\mu} \beta / \sqrt{4\pi\epsilon_0}$ and the Förster defect is $\Delta_F = E_c + E_d - (E_a + E_b)$ with the lower order energy state neglected. Depending on the energy scale, this interaction term might take different shapes according to the type of interaction that is performed (see fig. 2.8).

The **Rydberg blockade** phenomena is a conditional mechanism which, in the presence of two or more atoms, the Rydberg excitation in one of the atoms, results in a suppression of a second or subsequent excitation due to an energy level shift for the neighbour atoms (see fig. 2.9 (left)). In fig. 2.9 (right) it is shown the calculation of the pair-states energy as a function of the distance between atoms, where the energy is shifted up, the closer they get.

- **Resonant interaction** ($\Delta_F \approx 0$)

³In the Dirac's notation the state vector $|\alpha\beta\rangle = |a\rangle_j \otimes |\beta\rangle_k$ represents the product state between the Hilbert space between both atoms.

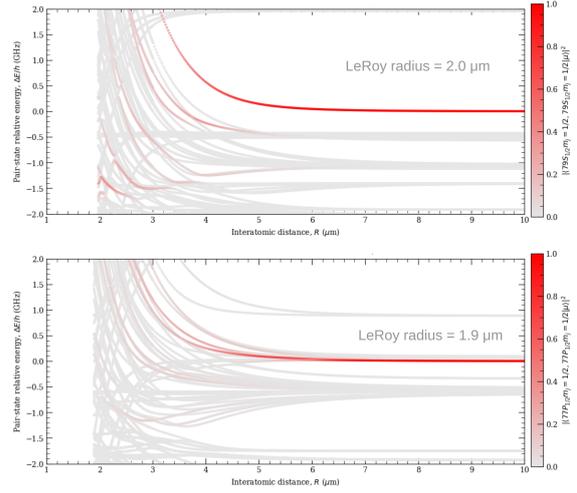
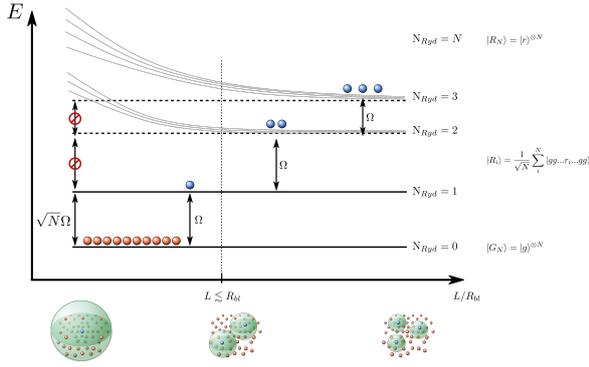


Figure 2.9: **Rydberg blockade** Rydberg blockade diagram of the suppression of secondary excitation of atoms, due to an energy shift. Pair-state relative energies that change with the distance between atoms and the red zone depict the high population of Rydberg states.

This type of interactions involves the coherent dipolar exchange of two dipole-coupled Rydberg states with different parity in the angular momentum (e.g., $a = d = nS$ and $b = c = nP$) with an interaction strength that is proportional to $1/R^3$.

$$\hat{\mathcal{H}} = \frac{\mu_{ac}\mu_{bd}}{R_{j,k}^3} |ab\rangle\langle cd| + h.c., \quad (2.34)$$

$$= \hat{\mathcal{H}}_{abcd}^{j,k} \left(V_{j,k} = \frac{2\mu_{ac}\mu_{bd}}{R_{j,k}^3} \right). \quad (2.35)$$

By tuning the Föster resonances with external electric fields it is possible to modulate the effective Rydberg-Rydberg interaction strength.

- **Non-resonant interaction** ($\Delta_F \gg |\mu_{ac}\mu_{bd}|/R_{j,k}^3$)

A second order perturbation of this term drives the system into a low energy subspace:

$$\hat{\mathcal{H}} = -\frac{C_6}{R_{j,k}^6} |ab\rangle\langle ab| = \hat{\mathcal{H}}_{j,k}^{abab} \left(V_{j,k} = -\frac{C_6}{R_{j,k}^6} \right), \quad (2.36)$$

where $C_6 = \frac{|\mu_{ac}|^2|\mu_{bd}|^2}{\Delta_F}$ is the van der Waals coefficient, that determines the energy shift of the state $|ab\rangle$ by $-C_6/R_{j,k}^6$. This type of interaction involves Rydberg state that are the same or have the same parity in the angular momentum i.e., $a = b = nS$ or $a = nS$ and $b = n'S$.

More generally, the Rydberg-Rydberg interactions must be consider as a total contribution of the different types of interactions, where some can be neglected under certain conditions. Additionally, the angular dependency of the Rydberg interaction can be often neglected in alkali atoms since they are isotropic and with simple geometries i.e., $V_{j,k} \propto 1 - 3\cos^2(\theta_{j,k})$ ⁴. In fig. 2.10 it is possible to observe the angle dependency of the

⁴Software for the calculation of properties of neutral and Rydberg atoms are available in open repositories.

interaction (C_6 coefficient) for nS states and between nP states. The interaction between

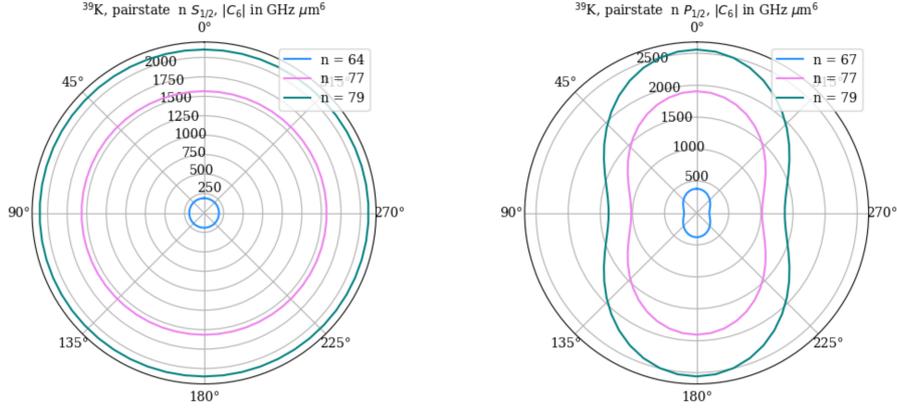


Figure 2.10: **Angular dependency of C_6 coefficient** For nS states (left) and nP states (right) of potassium-39.

Rydberg states serve generating a toolbox for quantum simulation with a set of models that can be generated are: Quantum Ising model, Quantum XY model and Quantum XXZ model.

- A. **Quantum Ising model:** the neighbour interaction model is implemented using transverse and longitudinal fields, using the Rydberg blockade (i.e., diagonal two-body Rydberg-Rydberg interaction), represented by the Hamiltonian:

$$\hat{\mathcal{H}} = \sum_j \left[\hat{\mathcal{H}}_j^{gr} (\varphi_j = 0) + \frac{1}{2} \sum_{k \neq j} \hat{\mathcal{H}}_{j,k}^{rrrr} \right], \quad (2.37)$$

where g and r are the ground and Rydberg states respectively. This model is equivalent to the spin Hamiltonian:

$$\hat{\mathcal{H}} = \frac{1}{2} \sum_j \left[\Omega_j(t) \hat{X}_j + (\Delta_j(t) - \mathcal{I}_j) \hat{Z}_j + \sum_{k \neq j} \frac{V_{j,k}}{4} \hat{Z}_j \hat{Z}_k \right], \quad (2.38)$$

where in the context of the model \hat{X}_j, \hat{Z}_j are the Pauli's σ_x and σ_y operators.

A special case of this model is the PXP model, where the Hamiltonian is given by:

$$\hat{\mathcal{H}} = \frac{1}{2} \sum_j \Omega_j(t) \hat{P}_{j-1}^0 \hat{X}_j \hat{P}_{j+1}^0, \quad (2.39)$$

for the case when $\Delta_j = 0$ and $a < R_b < 2a$ (nearest neighbor blockade) with $V_{j,k} \approx 0$. Here $\hat{P}_j^0 = |0\rangle_j \langle 0|$ is a projection operator of the state 0 acting over the atom j .

- B. **Quantum XY model:** a spin model which describe a spin-exchange interactions between Rydberg states. The model can be mapped onto a hopping hardcore bosons on a graph or using the Jordan-Wigner transformation for non-interacting fermions. The model for spin-1/2 can be realized with atoms in the Rydberg states with orbital angular momentum difference equal to one ($\delta\ell = 1$), for example where the Rydberg

states involved are $|s\rangle = nS_{1/2}$ and $|p\rangle = nP_{1/2}$. In the regime of resonant dipolar exchange interaction the Hamiltonian of the model can be natively implemented as:

$$\hat{\mathcal{H}} = \sum_j \left[\hat{\mathcal{H}}_j^{sp}(\varphi_j = 0) + \frac{1}{2} \sum_{k \neq j} \hat{\mathcal{H}}_{j,k}^{spps} \right]. \quad (2.40)$$

This expression is equivalent to the spin-1/2 Hamiltonian:

$$\hat{\mathcal{H}} = \frac{1}{2} \sum_j \left[\Omega_j(t) \hat{X}_j + \Delta_j(t) \hat{Z}_j + \sum_{k \neq j} \frac{V_{j,k}}{4} \left(\hat{X}_j \hat{X}_k + \hat{Y}_j \hat{Y}_k \right) \right]. \quad (2.41)$$

C. **Quantum XXZ model:** this spin model is studied in the context of spin-orbit coupled Rydberg systems where the effective $\delta\ell = 0, 2$. Here the van der Waals interactions are mediated via a additional intermediate state with large Föster defect, for instance: $nS \leftrightarrow n'P \leftrightarrow (n+1)S$, the native Hamiltonian is:

$$\hat{\mathcal{H}} = \sum_j \left[\hat{H}_j^{rr'}(\varphi = 0) + \frac{1}{2} \sum_{k \neq j} \hat{H}_{j,k}^{rr'r'r} (V_{j,k} = 2J_{j,k}) + \frac{1}{2} \sum_{k \neq j} \hat{H}_{j,k}^{rr'r'r'} (V_{j,k} = 2\delta J_{j,k}) \right]. \quad (2.42)$$

Thus, the equivalent to the anisotropic spin- $\frac{1}{2}$ XXZ-Hamiltonian is given by:

$$\hat{\mathcal{H}} = \frac{1}{2} \sum_j \left[\Omega_j(t) \hat{X}_j + (\Delta_j(t) - \mathcal{I}_j) \hat{Z}_j + \frac{1}{2} \sum_{k \neq j} J_{j,k} \left(\hat{X}_j \hat{X}_k + \hat{Y}_j \hat{Y}_k + \delta \hat{Z}_j \hat{Z}_k \right) \right], \quad (2.43)$$

where $\mathcal{I}_j = \sum_{k, j \neq k} \delta J_{j,k}$, interaction strength $J_{j,k} = -C_6/R_{j,k}^6$ and the anisotropy parameter δ that is Rydberg state dependent.

2.4 Quantum information processing with Rydberg atoms

In the previous chapter we described the foundations of quantum information processing describing the basic components of the theory: qubits, gates, circuits and algorithms. However, these elements are not motivated by the physics of the quantum mechanical objects, which means that the qubit or more generally, the qudit state are representations of real energy levels of the quantum object. Similarly, the quantum gates are just algebraic representations of the outcome of actions of different Hamiltonians over the system dynamics. The effect of these actions plus the effect of the environment will ultimately define the physical operation over the defined qubit/qudit. These are the so-called “native gate operations”, which are the natural processes by which the quantum system undergoes a dynamic evolution.

2.4.1 Rydberg qubits encoding

Like any other architectural framework, qubit encoding in atomic systems unleashes a variety of codifications. Although the nature of atomic energy levels aligns more closely

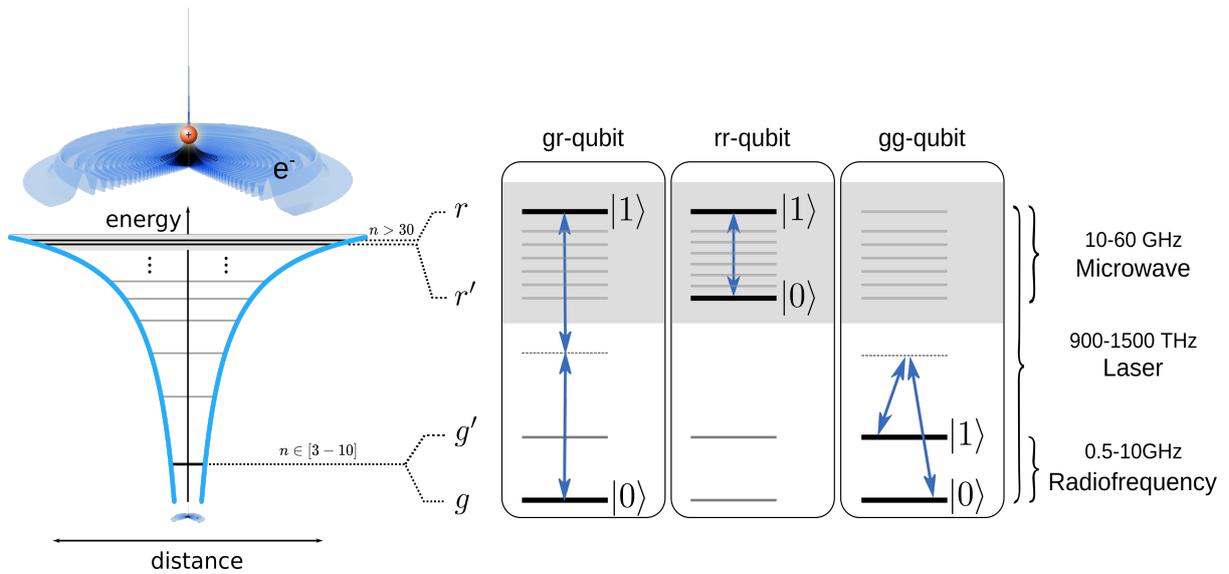


Figure 2.11: **Rydberg qubits.** (Left) The atomic potential of an atom with the quantized states or energy levels of the electronic radial wavefunction. In this model, ground-states are well defined for the principal quantum number n in the range between 3 and 10; meanwhile highly-excited Rydberg states are situated for values of n above 30. (Right) Rydberg qubit encodings given by the type of states of the qubit: gr-qubit coupled with lasers, gg-qubit also coupled with laser with two-photon (Raman) transitions and rr-qubit coupled with microwaves.

with a qudit than a qubit, this work specifically focuses on quantum information with qubits. Thus this inherently qudit behavior, prompt intriguing inquiries about extending the native gate set to accommodate more levels. As mentioned earlier, the intricate energy level configuration of Rydberg atoms, combined with sophisticated techniques for coherently manipulating internal states with laser beams, radio-frequency and microwave fields, opens up numerous possibilities for quantum information storage and manipulation. The Rydberg qubits are then classified into three primary types (i.e., gr-qubits, rr-qubits and gg-qubits), distinguished by the number of weakly-interacting states, also referred as ground-states, and the number of strongly-interacting states, normally referred to as Rydberg states, which might compose the qubit states. See fig. 2.11 for an overview.

Depending on the type of atoms (alkaline or alkaline-earth), the ground state has different values of n and ℓ , additionally, within the manifold the qubit state can be prepared in a specific m_F state that exhibits insensibility to magnetic fields. For, for example, alkaline atoms such as ^{39}K , it could be the state $|4S_{1/2}, F = 1, m_F = 0\rangle$ and for ^{87}Rb it is $|5S_{1/2}, F = 1, m_F = 0\rangle$. Meanwhile for alkaline earth-atoms such as: ^{87}Sr it is possible to use the ground states from the singlet or triplet manifolds, this means using either a low-lying F, m_F states of 1S_0 or metastable F, m_F states: 3P_0 or 3P_1 or $3P_2$. For ^{171}Yb the low-lying F, m_F states belong to 1S_0 and metastable F, m_F states: 3P_0 or 3P_1 . More information about the atomic energy diagrams can be found in appendix C.

These types of encoding are compatible with single-atom or atomic ensembles approaches. Thus taking advantage of fast preparation and readout as well as using the ensemble to replenish the qubit after an erasure error or for a qubit reset. However this

remains challenging due to possible fluctuations of the initial number of loaded atoms.

Ground-Rydberg qubit (gr-qubit)

The gr-qubit (see also fig. 2.12) is encoded in a ground state and a highly-excited Rydberg state for the state $|0\rangle \equiv |g\rangle$ and $|1\rangle \equiv |r\rangle$ respectively. This kind of qubit is usually manipulated with laser beams with wavelengths that match the energy difference between the two atomic states that are usually defined with azimuthal quantum number $\ell = 0$ and for the ground and excited state with principal quantum number $n < 10$ and $n > 25$ respectively. The energy difference between these atomic levels corresponds to (900-1500)THz, where the final value will depend on the atomic element and the selected states. This qubit encoding has lifetimes limited by the Rydberg state lifetime in the order of $\sim 100\mu\text{s}$ and coherence times of $20\mu\text{s}$ mainly limited by laser phase noise. This qubit exhibits magic wavelengths that allows it to trap ground and Rydberg states simultaneously.

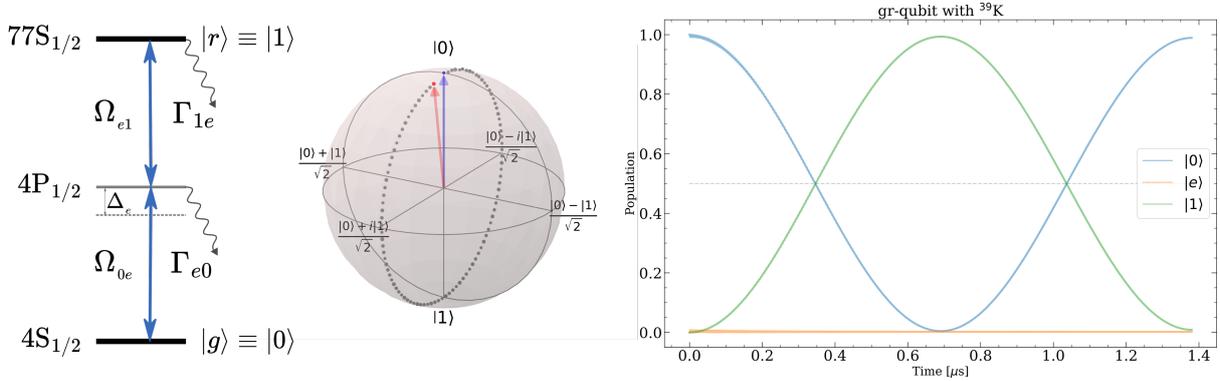


Figure 2.12: **gr-qubit**. Atomic energy level diagram of the model for this qubit. Trajectory of the qubit states on the Bloch sphere. Population dynamics from the solution of a quantum master equation at the RWA approximation, using parameters of ^{39}K . Rabi couplings $\Omega_{0e}, \Omega_{e1} = 2\pi \times 8\text{MHz}$, $\Delta_e = -2\pi \times 176\text{MHz}$, $\Gamma_{e0} = 2\pi \times 5.928\text{MHz}$, $\Gamma_{re} = 2\pi \times 323\text{Hz}$. Simulations realized with AQiPT (see chapter 3).

Here the qubit control using lasers can be achieved either by single-photon transitions or by multi-photon transitions. The most common way is to use two-photon transition, since it is a good compromise between technical complexity (e.g., laser locking, time-control of laser pulses) and performance. Using adiabatic protocols such as stimulated Raman adiabatic passage (STIRAP), it is possible to obtain high fidelities.

Another way to engineer this type of qubit is to use atomic ensembles which allow to exploit the phenomena of \sqrt{N} enhancement in the Rabi coupling due to the Rydberg blockade, see fig. 2.13. However, a linked challenge to this encoding when using atomic ensembles is that the coupling performance to the other qubit state is determined by the fluctuations in the number of atoms in the ensemble, making it difficult to calibrate the frequency of the qubit transition. Another challenge is related to the qubit-qubit interaction since the interactions between ensembles could generate undesirable cross-talks.

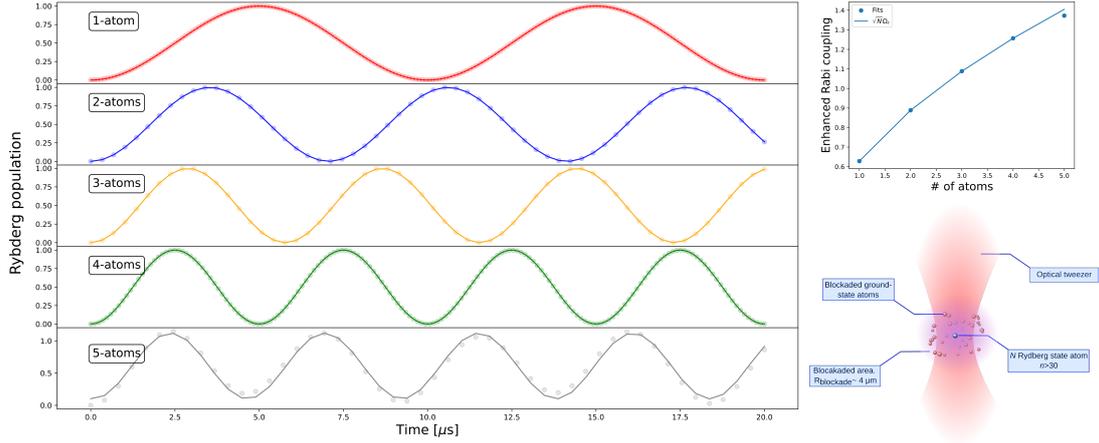


Figure 2.13: **Ensemble gr-qubit**. Level energy diagram of the collective states of the ensemble qubit (left). Graphical representation of the ensemble qubit in a tightly focused far-off-resonant tweezer beam, showing the blockade area by the Rydberg atom (in blue) over the ground-states atoms (in red). This type of qubit exhibits a \sqrt{N} enhancement on the Rabi coupling, where N is the number of atoms, the effect of the Rabi oscillation is shown from simulations for up to 5 atoms.

Rydberg-Rydberg qubit (rr-qubit)

Utilizing two distinct Rydberg states for qubit encoding, denoted as rr-qubits ($|0\rangle \equiv |r\rangle$ and $|1\rangle \equiv |r'\rangle$), provides a versatile platform for tailoring Rydberg-Rydberg interactions, see fig. 2.14 for an overview. This encompasses the manipulation of long-range dipolar-exchange interactions expressed as $\frac{C_3}{R^3}(|10\rangle\langle 01| + |01\rangle\langle 10|)$. The effectiveness of interactions between Rydberg states is most pronounced when the principal quantum number n of the qubit states are closely matched in the configuration n and $n - 1$, and when azimuthal quantum numbers $\delta = |\ell - \ell'| = 0, 1, 2$. This gives rise to typical energy splittings of rr-qubits within the range of 5 to 80GHz.

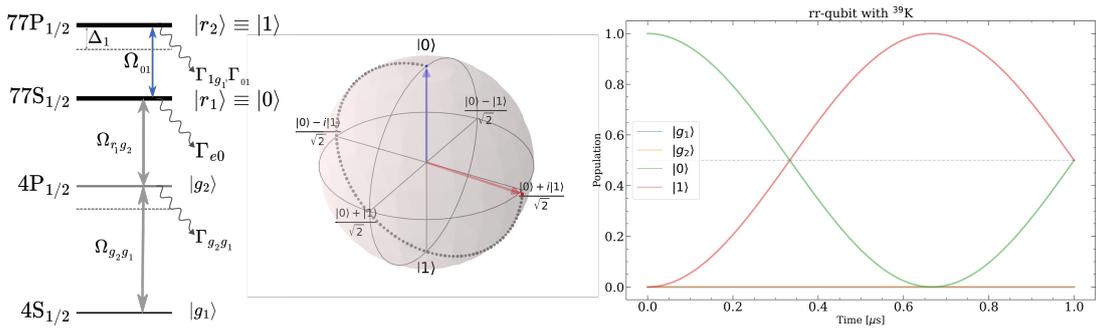


Figure 2.14: **rr-qubit**. Atomic energy level diagram of the model for this qubit. Trajectory of the qubit states on the Bloch sphere. Population dynamics from the solution of a quantum master equation at the RWA approximation, using parameters of ^{39}K . Rabi couplings $\Omega_{01} = 2\pi \times 0.375\text{MHz}$, $\Delta_e = 0\text{MHz}$, $\Gamma_{g_1g_2} = 2\pi \times 5.93\text{MHz}$, $\Gamma_{r_1g_2} = 2\pi \times 323.0\text{Hz}$, $\Gamma_{r_2g_1} = 2\pi \times 94.4\text{Hz}$, $\Gamma_{r_2r_1} = 2\pi \times 352.1\text{Hz}$. Simulations realized with AQiPT (see chapter 3).

To initialize rr-qubits, one can employ optical excitation pulses from the ground state or robust STIRAP state transfer. The qubit's state manipulation can be done using

microwave coupling fields, presenting advantages such as low-noise sources for steering transitions and minimal Doppler dephasing. However, this approach introduces challenges in precisely addressing individual qubits and the polarisation of the field.

Similarly to gr-qubits, coherence times face limitations due to the complexities associated with trapping Rydberg states and the finite Rydberg state lifetime of approximately $100\mu\text{s}$. Recent experiments with alkaline-earth Rydberg atoms confined in optical tweezers have demonstrated coherent dynamics of single atoms with coherence times reaching $T_2^* = 22\mu\text{s}$. Furthermore, coherent dynamics in small arrays of alkali atoms have showcased lifetimes of approximately 4 to $7\mu\text{s}$, coupled with typical nearest-neighbour interaction strengths ranging from 1 to 2.5MHz.

A potential avenue for extending qubit lifetimes involves the utilization of qubits formed by two circular Rydberg states $|0\rangle \equiv |C\rangle$ and $|1\rangle \equiv |C'\rangle$, boasting large principal quantum numbers n and n' and maximal azimuthal $\ell = n - 1$ and $\ell' = n' - 1$ quantum numbers. These states are promising for prolonging lifetimes of the states in the Rydberg manifold suppressing the effects from the blackbody radiation due to low cryogenic temperatures and the use of microwave cavities. However, the integration of laser trapping with cryogenic environments comes at the expense of heightened technical intricacies.

Ground-Ground qubit (gg-qubit)

The gg-qubit is encoded in two long-lived low-lying atomic states ($|0\rangle \equiv |g\rangle$, $|1\rangle \equiv |g'\rangle$), see fig. 2.15 for an overview. It exhibits larger coherence times and switchable interactions by using a combination of laser and field control, making them prime candidates for universal quantum computing. This encoding can involve magnetically insensitive hyperfine sublevels of the electronic ground state or the electronic ground state coupled with a metastable excited state (e.g., alkaline-earth species). Hyperfine qubit energy splittings fall in the (1 - 10) GHz range, while metastable qubits involve optical frequencies.

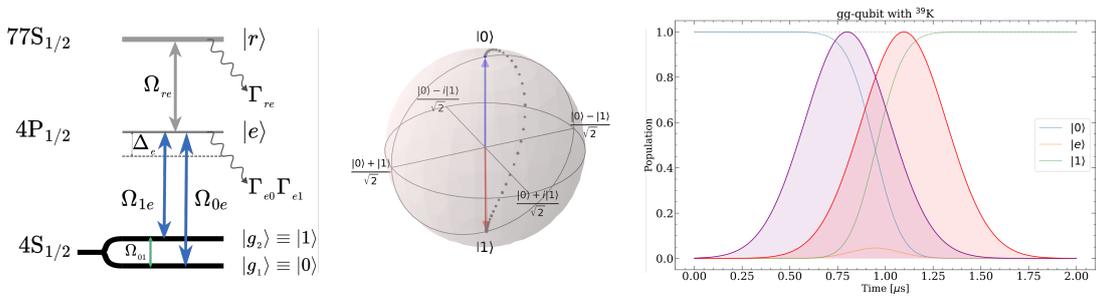


Figure 2.15: **gg-qubit**. Atomic energy level diagram of the model for this qubit. Trajectory of the qubit states on the Bloch sphere. Population dynamics from the solution of a quantum master equation at the Rotating Wave Approximation (RWA), using parameters of ^{39}K . Double gaussian STIRAP protocol with Rabi couplings $\Omega_{0e}, \Omega_{1e} = 2\pi \times 5\text{MHz}$, $\Delta e = 8\text{MHz}$, $\Gamma_{e0}, \Gamma_{e1} = 2\pi \times 5.93\text{MHz}$. Simulations realized with AQiPT (see chapter 3).

The typical coherence times of gg-qubits extends to (1 - 20) ms for single qubit dephasing times without spin echo pulses or magic trapping techniques which could boost them to even longer lifetimes. This prolonged coherence is a significant advantage over gr- and rr-qubits. Single qubit gates demonstrate high fidelities, exceeding $\mathcal{F} > 0.9999$ for global manipulation and $\mathcal{F} = 0.992$ for local addressing with low cross-talk.

Unlike gr- and rr-qubits, gg-qubits exhibit weak interactions, necessitating mediation through precisely timed and pulse-shaped optical fields that momentarily excite and de-excite them via Rydberg states. This approach has been employed to demonstrate two-qubit quantum logic gates with fidelities as high as $\mathcal{F} = 0.995(2)$ for CZ gates and a Bell fidelity of $\mathcal{F} = 0.980(2)$ without SPAM correction with individual qubit addressing [Eve23]. Alternatively, gg-qubits can be induced to interact by weakly admixing some Rydberg state character to the ground states using off-resonant laser coupling (Rydberg dressed qubits). This opens possibilities for realizing new many-body phases with engineered long-range interactions and spatially/temporally controlling interactions for gate operations.

These different types of encoding present a series of advantages and disadvantages for either quantum computing and simulation and the ultimate device i.e., quantum computer or simulator might dynamically exploit their features by using different encoding according to the task required to be performed. In table 2.2 I summarized the types of Rydberg qubit encoding and their main properties.

| Encoding | Manipulation | Lifetimes | Advantages | Disadvantages |
|------------------|---------------------------|---|---|---|
| gr-qubits | Lasers | $< 100\mu\text{s}$ | State-of-the-art gates. Spin models. | Short lifetimes. Always-ON interactions. |
| rr-qubits | Microwaves | $< 100\mu\text{s}$ ($\sim 10\text{ms}$) | Versatile interactions. | Short lifetime (No for circular Rydberg states) |
| gg-qubits | Raman lasers & microwaves | $>100\text{ ms}$ | Long lifetimes | Complex qubit manipulation. |

Table 2.2: **Characteristics of Rydberg qubits.**

2.4.2 Single qubit Rydberg gates

In quantum computing and in quantum simulation the processing of the information in the system is done via physical operations that change the state of the system and manage the quantum resources. For instance the generation of entanglement in arrays of qubits has been recently demonstrated with: GHZ-states with 6 qubits [Gra22] and 48 physical entangled qubits [Blu24].

A sequence of physical operations which are the result of the action of a series of Hamiltonians can be engineered to perform a unitary transformation that can be obtained from the exponential matrix of the evolution operator following the expression:

$$\hat{U}(t) = \exp \left\{ i\hat{\mathcal{H}}\Delta t/\hbar \right\}. \quad (2.44)$$

In this way the canonical gates will be performed by the natural mechanisms of the system, which are recognized as the native gates. Here I present the native gate set for the

Rydberg qubits, together with algebraic representations and the sequence of Hamiltonian-pulses, which are the time-dependency of each Hamiltonian term.

To facilitate the explanation of the implementation of the Rydberg native gates, I introduce an innovative diagram (see fig. 2.16) to represent the truth tables that comprehend the series of physical operations that produce the different couplings between the atomic states i.e., computational and ancillary states, following a sequence of Hamiltonian-pulses that we define as a protocol. These diagrams also include the strong Rydberg-Rydberg interaction regions and a bordered color ring denoting the phase of the state. These diagrams are a tool for representing the logic operation over the qudit state.

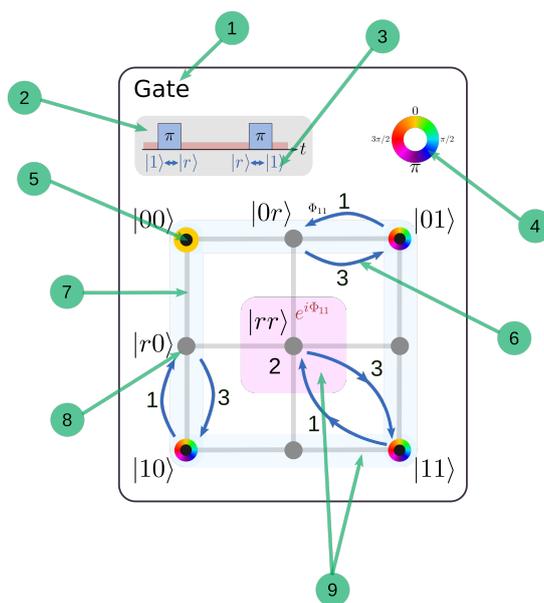


Figure 2.16: **Native gate map.** State diagram of the physical operation for the implementation of Rydberg native gates. (1) Name of the gate. (2) Pulse protocol. (3) Transition associated to the applied pulse. (4) Colormap phase of the state. (5) Computational states. (6) State transitions. (7) State link due possible transition. (8) ancillary states. (9) Transition regions: interacting sub-space (red) and non-interacting (blue).

Atomic systems possess the capability to perform operations involving physical manipulations on individual atoms (i.e., qubits/qudits) or sets of atoms (i.e., quantum registers) by using local and global addressing. The local addressing can be executed by 1D and 2D AODs which address cycle time faster than other devices such as DMDs and SLMs. Due to this difference in the response time, there are two types of functionalities where the local addressing is used:

- 1) Static functionality, for generating the structured light for generating (i.e., atom-light substrate) the array of atoms that form the quantum register at the real and Fourier plane.

- 2) Dynamic functionality, used to generate AC-Stark shifts to complement global gate protocols, sequences or arbitrary single-atom operations performing parallel-, single- and multi-qubit gates, and atom rearrangement with μs time-scale.

On the other hand, the global addressing can be performed by high-power global laser beams or by a microwave field generated using highly coherent sources (for example, as the one used along this work with low single-side-band phase noise $< -134 \text{ dBc/Hz}$, see chapter 4) emitted by a polarized horn and helical antennas. Arrays of helical antennas [And21] have also shown polarization control of the electro-magnetic field. The global addresses are used for state preparation of the quantum register (i.e., optical pumping, circularization of Rydberg states) as well as for global gates (i.e., simultaneous single qubit gates).

Combinations of the technology, using AODs, DMDs, SLMs have shown that it is possible to hybridize the two functionalities [Zha24], [Gra23a] opening new possibilities of combined physical operations along the quantum circuit or algorithm to be executed for computing or simulation.

In previous sections, I reviewed the physics toolbox for building the different physical processes that can be utilized to generate evolution of the systems that can be described as unitary transformations. These transformations or gates are explained below by describing the time-dependent Hamiltonian and their linked unitary transformation for single-qubit gates, two-qubit gates and multi-qubit gates including the parallel qubit gates.

U_{xy} (native gate)

Considering the time evolution operator of the atom-light interaction Hamiltonian from the eq. (2.1), this Hamiltonian leads to an arbitrary rotation operator as follows:

$$\begin{aligned} U &= \exp \left\{ -i \hat{\mathcal{H}}_j^{01} \tau_g \right\} \\ &= \exp \left\{ i \left(\frac{\Delta}{2} \right) \tau_g \right\} \times \exp \left\{ -i \left(\frac{\tilde{\Omega}}{2} \vec{v} \cdot \vec{\sigma} \right) \tau_g \right\}, \end{aligned} \quad (2.45)$$

where τ_g is the gate time, the generalized Rabi frequency is given by $\tilde{\Omega} = \sqrt{\Omega^2 + \Delta^2}$ and we define the vectors $\vec{v} = \tilde{\Omega}^{-1} \{ \Omega \cos(\phi), -\Omega \sin(\phi), \Delta \}$ and $\vec{\sigma} = \{ \hat{X}, \hat{Y}, \hat{Z} \}$. The inner product of these vectors represents the effective coupling between the qubit states in the case of resonance (i.e., $\Delta = 0$) and the decoherence term due to the detuning of the coupling. For the former, we can exponentiate the matrix and obtain the unitary transformation for any arbitrary single qubit native gate, denoted by $U_{xy}(\theta, \phi)$:

$$U_{xy}(\theta, \phi) = \begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2) e^{i\varphi} \\ -i \sin(\theta/2) e^{-i\varphi} & \cos(\theta/2) \end{pmatrix}, \quad (2.46)$$

with $\theta = \Omega \tau_g$. The parameters of the Hamiltonian: Ω, Δ and ϕ are dependent on the coupling field: power, frequency difference to the transition and phase coherence. It

is possible with the current technology addressed in previous sections to modulate the amplitude, frequency and phase of the field, either from a coherent laser beam or a free space field e.g., microwave or radio-frequency.

The symbol of this gate is shown in fig. 2.17.

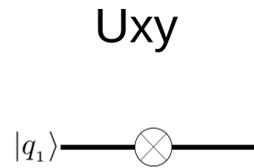


Figure 2.17: U_{xy} . Quantum circuit symbol of the gate, with the qubit state label.

It is possible to use a sequence of this type of gate, each with possibly different parameter values, to perform another gate known as a composed gate. For example, to form a set of canonical parametric single qubit gates, known as rotation gates which depend of one (angle) parameter.

$R_{x,y,z}$ gate

A more efficient approach to implementing the R_z gate involves utilizing off-resonance transitions ($\Delta \neq 0$), thereby obviating the necessity for executing three native rotations. This technique has been experimentally implemented for two qubit gates with two and one pulse sequence, generating a phase accumulation in the state of the qubit.

Rotations around the x axis are given by a single $U_{xy}(\theta, \varphi)$, where $\varphi = 0$:

$$\begin{aligned} R_x(\theta) &= U_{xy}(\theta, 0), \\ &= \begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}. \end{aligned} \quad (2.47)$$

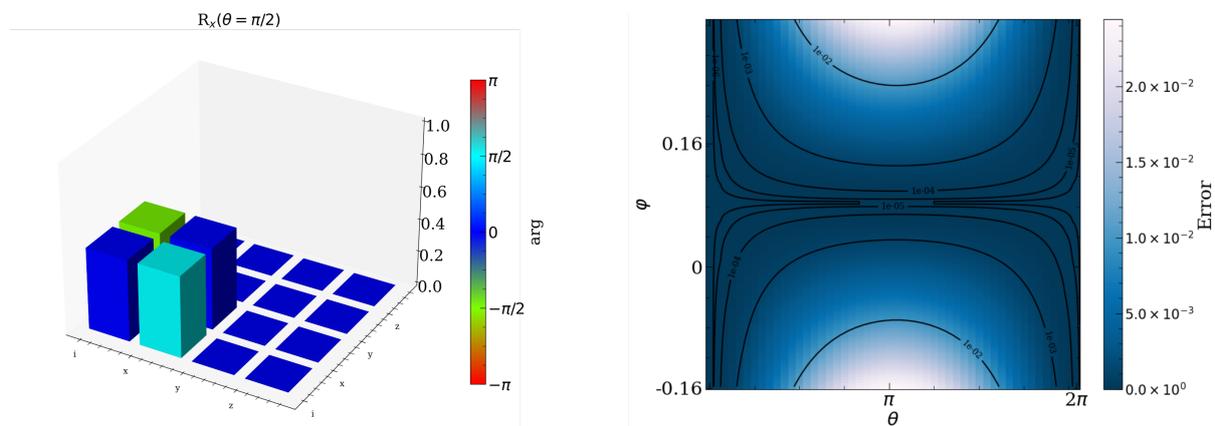


Figure 2.18: R_x gate performance. Quantum state tomography of the $R_x(\theta)$ gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates with a fix φ and a variable θ parameter (right).

Similarly the rotations around the y axis use the same structure with $\varphi = -\pi/2$ instead i.e., $e^{-i(\pm\varphi)} = \mp i$:

$$\begin{aligned} R_y(\theta) &= U_{xy}(\theta, -\pi/2), \\ &= \begin{pmatrix} \cos(\theta/2) & (-i)^2 \sin(\theta/2) \\ -i^2 \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}. \end{aligned} \quad (2.48)$$

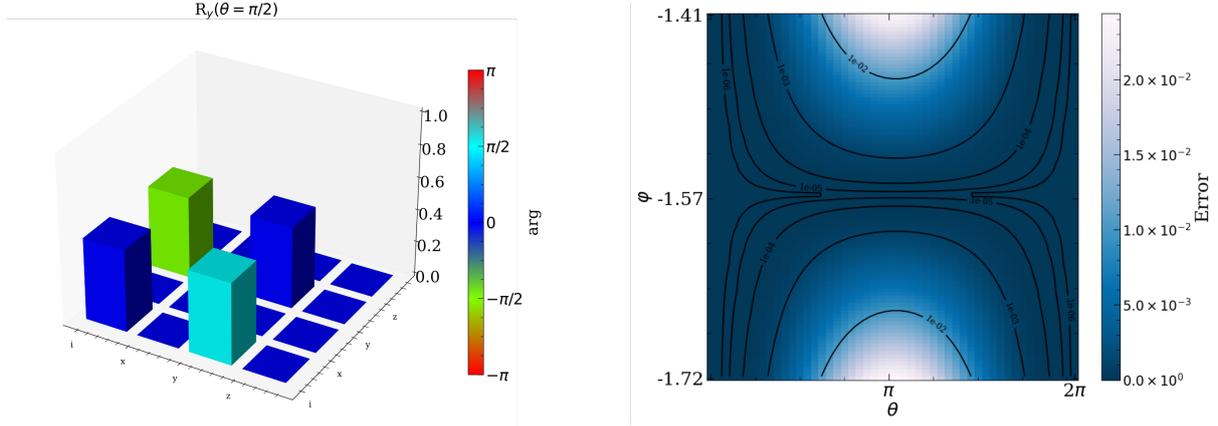


Figure 2.19: R_y gate performance. Quantum state tomography of the $R_y(\theta)$ gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error with a fix φ and a variable θ parameter (right).

Lastly, the rotations around the vertical axis or z axis use a composition of three $U_{xy}(\theta, \varphi)$ gates as follows:

$$\begin{aligned} R_z(\theta) &= U_{xy}(\pi/2, \pi/2)U_{xy}(\theta, 0)U_{xy}(\pi/2, -\pi/2), \\ &= \begin{pmatrix} C_{\pi/4} & -i^2 S_{\pi/4} \\ (-i)^2 S_{\pi/4} & C_{\pi/4} \end{pmatrix} \begin{pmatrix} C_{\theta/2} & -i S_{\theta/2} \\ -i S_{\theta/2} & C_{\theta/2} \end{pmatrix} \begin{pmatrix} C_{\pi/4} & (-i)^2 S_{\pi/4} \\ i^2 S_{\pi/4} & C_{\pi/4} \end{pmatrix}, \\ &= \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} e^{-i\frac{\theta}{2}} & -\frac{\sqrt{2}}{2} e^{i\frac{\theta}{2}} \\ \frac{\sqrt{2}}{2} e^{-i\frac{\theta}{2}} & \frac{\sqrt{2}}{2} e^{i\frac{\theta}{2}} \end{pmatrix}, \\ &= \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}. \end{aligned} \quad (2.50)$$

Here we use the notation $C_\theta \equiv \cos(\theta)$ and $S_\theta \equiv \sin(\theta)$. Moreover, z axis rotations can be achieved by using off-resonant light to induce AC-Stark shifts making the qubit to dephase without requiring three gates, but this requires precise control of the qubit phase or in other words, the laser phase.

In figures fig. 2.18, fig. 2.19 and fig. 2.20 the quantum process tomography is shown (see details in appendix B) that corresponds to the canonical versions. Additionally, it is possible to observe lower errors (blue areas) for short fluctuations of φ being robust for values of θ close to 0 and 2π in the case of R_x and R_y and being less robust in the case of R_z rotations.

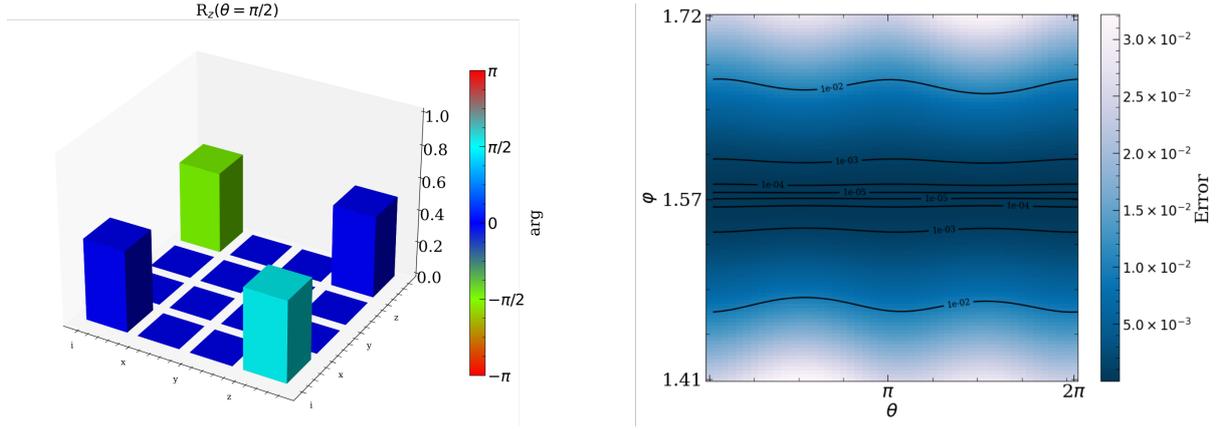


Figure 2.20: R_z **gate performance**. Quantum state tomography of the $R_z(\theta)$ gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error in the fix (φ) and variable (θ) parameter (right).

H (Hadamard) gate

Another important example of a canonical gate that can be generated with the native rotation gate is the Hadamard gate, within a global phase factor of $\pi/2$, using a composition of 2 rotational gates:

$$\begin{aligned}
 H &= U_{xy}(\pi, 0)U_{xy}(\pi/2, -\pi/2), & (2.51) \\
 &= \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}, \\
 &= -i\frac{\sqrt{2}}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \\
 &= e^{-i\frac{\pi}{2}} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. & (2.52)
 \end{aligned}$$

Canonical single qubit gates can also be performed by the native gate of the Rydberg platform using any type of encoding. As well as other canonical gates that can be achieved also as a compositions of native gates or directly using a single U_{xy} .

The gate performance is shown in fig. 2.21.

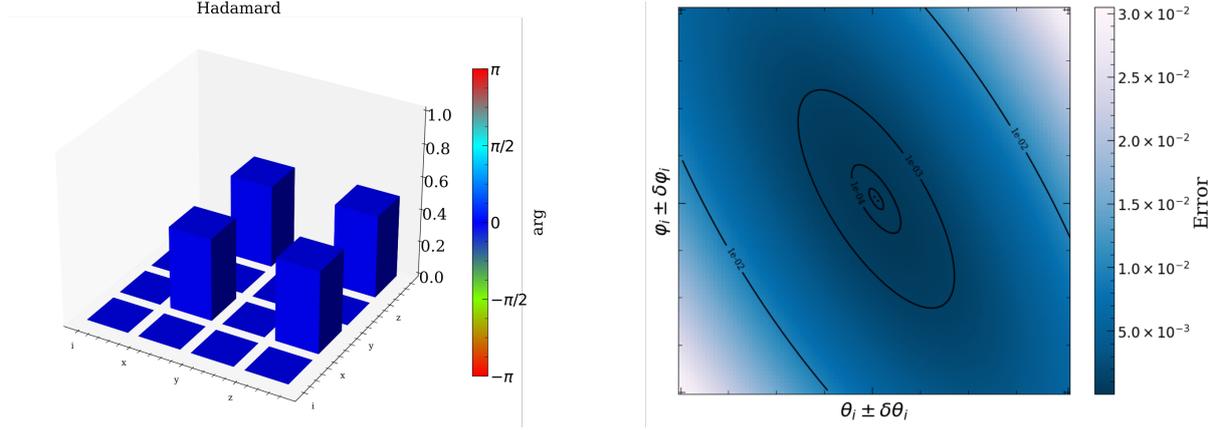


Figure 2.21: **H gate performance.** Quantum state tomography of the H gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

Pauli $X(\sigma_x)$ gate

This gate is also known as bit-flip gate, analog to the classical NOT gate. For this gate, a single native gate with: $\theta = \pi$ and $\varphi = \pi$ can be used as follows:

$$X = e^{-i\frac{\pi}{2}} U_{xy}(\pi, \pi), \quad (2.53)$$

$$= e^{-i\frac{\pi}{2}} \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} = e^{-i\frac{\pi}{2}} i \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$= e^{-i\frac{\pi}{2}} e^{i\frac{\pi}{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x. \quad (2.54)$$

This implementation includes a global phase of $\pi/2$. The gate performance is shown in fig. 2.22.

Pauli $Y(\sigma_y)$ gate

This gate is performed using a single native gate with: $\theta = \pi$ and $\varphi = \pi/2$ as follows:

$$Y = e^{-i\frac{\pi}{2}} U_{xy}(\pi, \pi/2), \quad (2.55)$$

$$= e^{-i\frac{\pi}{2}} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = e^{-i\frac{\pi}{2}} (-i^3)(-i) \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix},$$

$$= e^{-i\frac{\pi}{2}} e^{i\frac{\pi}{2}} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \sigma_y. \quad (2.56)$$

This implementation include a global phase of $\pi/2$. The gate performance is shown in fig. 2.23.

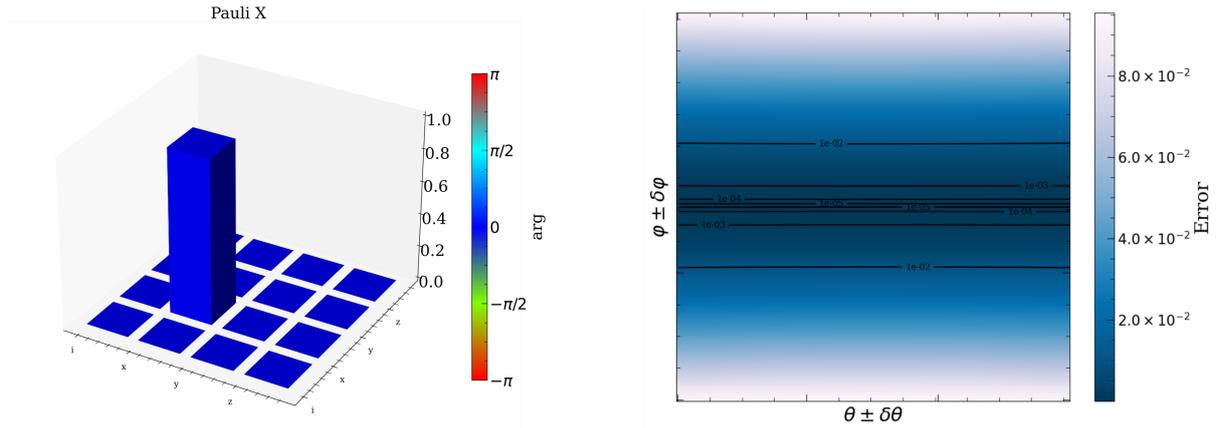


Figure 2.22: **X gate performance.** Quantum state tomography of the X gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

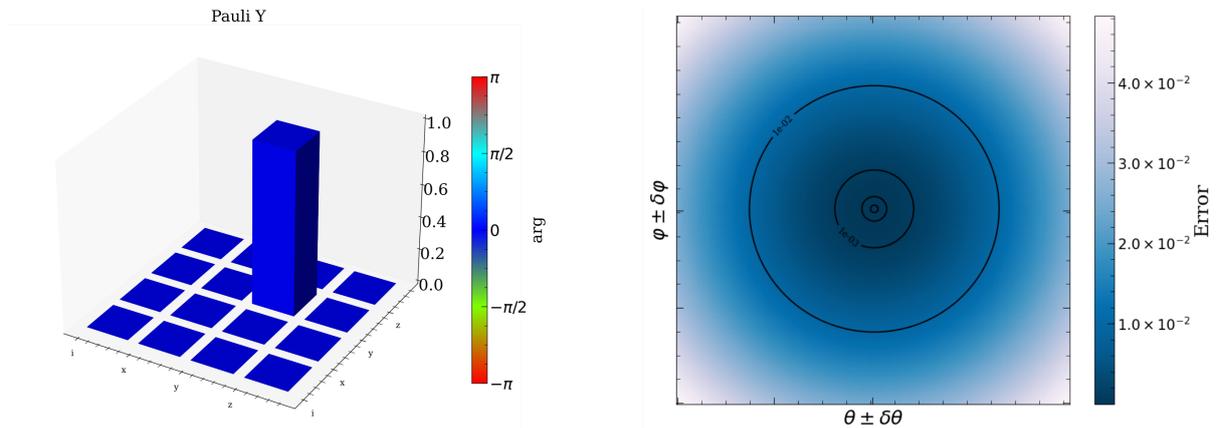


Figure 2.23: **Y gate performance.** Quantum state tomography of the Y gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

Pauli Z(σ_z)

This Pauli gate can be either been seen as a three U_{xy} gates, also with a global phase of $\pi/2$ or a combination of R_y and X gates.

$$Z = R_y(-\pi/2) \sigma_x R_y(\pi/2) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_z. \quad (2.57)$$

The gate performance is shown in fig. 2.24.

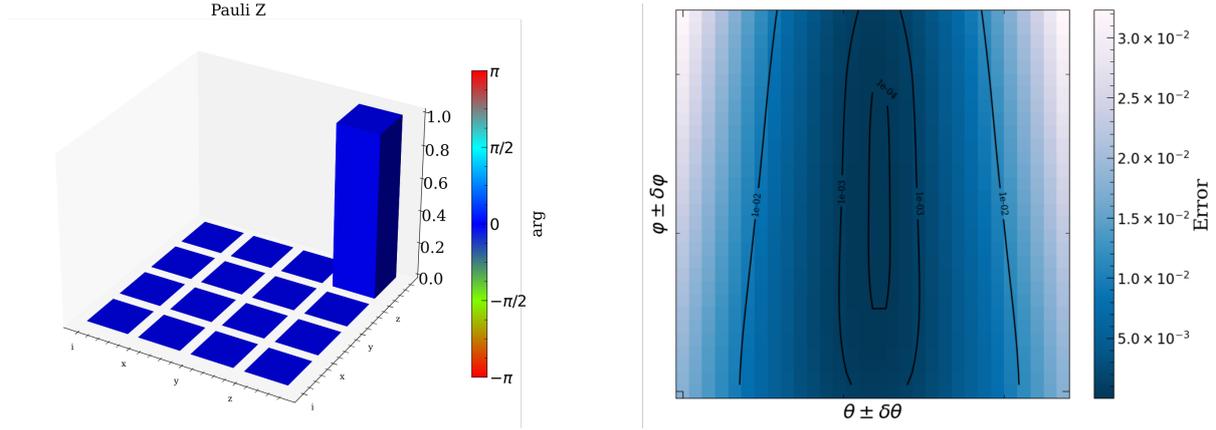


Figure 2.24: **Z gate performance.** Quantum state tomography of the Z gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

T gate

This gate produces a rotation over the z-axis of $\pi/4$ and it is composed of three native rotations:

$$T = e^{i\frac{\pi}{8}} U_{xy}(\pi/2, \pi/2) U_{xy}(\pi/4, 0) U_{xy}(\pi/2, -\pi/2), \quad (2.58)$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{pmatrix}. \quad (2.59)$$

The gate performance is shown in fig. 2.25.

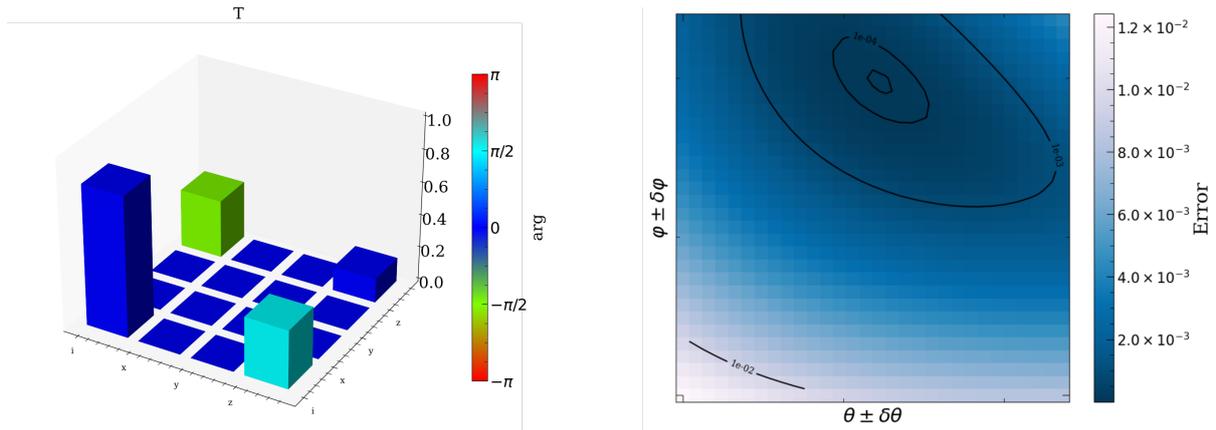


Figure 2.25: **T gate performance.** Quantum state tomography of the T gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

S gate

This gate is similar to the T gate, realizing a partial rotation of the qubit state around the z -axis. In the case of the S gate the rotation corresponds to $\pi/2$ and it is an expensive gate with 6 rotation native gates i.e., $S = T^2$. However, it is possible to achieve this kind of rotation gates around the z -axis by introducing AC-stark shifts made by locally focused laser beams e.g., optical tweezers.

$$S = T^2, \quad (2.60)$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \quad (2.61)$$

The gate performance is shown in fig. 2.26.

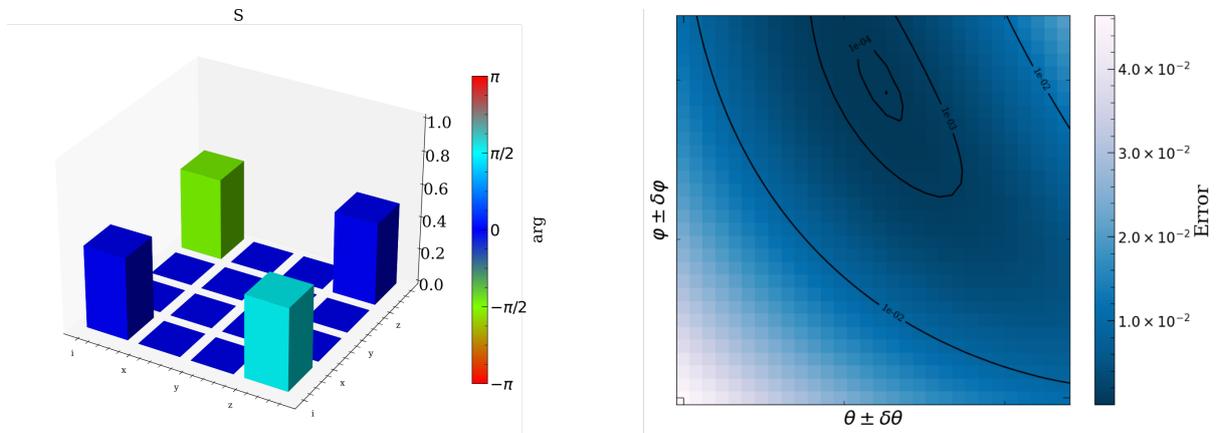


Figure 2.26: **S gate performance.** Quantum state tomography of the S gate where the colorbar describes the complex phase of the fidelity bar in the city plot (left). Heatmap of the gate error for values of the input parameters of the native $U_{x,y}$ gate with 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

2.4.3 Multi qubit native gates

Interactions play a key role in the implementation of two or more qubit gates. In previous sections I presented a toolbox for the tunable interactions using Rydberg qubits, some of the following gates exploit the resonant interactions based on the dipolar exchange between two Rydberg states of different parities (i.e., long-range interactions), meanwhile the other group of gates take advantage of the Rydberg blockade coming from the van der Waals interactions (i.e., short range interactions).

CU_{xy} gate

This controlled two-qubit (control and target) gate, realizes a conditional native rotation U_{xy} over the target qubit (e.g., $|t\rangle$) due to the state of the control qubit (e.g., $|c\rangle$). In an entangling operation a CU_{xy} gate can be implemented with $\theta = \pi$ and $\phi = 0$ to perform a canonical CNOT gate, preceded and followed by a $R_z(-\pi/2)$ gate over the control qubit.

The evolution operator for this gate carries the single-atom Hamiltonian term over the target qubit and the strong interactions between control and target during the gate time τ_g , as it is shown in the equation below:

$$\hat{U} = \exp \left\{ -i \left(\hat{\mathcal{H}}_t^{01} + \hat{\mathcal{H}}_{c,t}^{1111} \right) \tau_g \right\}. \quad (2.62)$$

This represents the following Hamiltonian sequence:

Additionally, expanding the exponential matrix one can obtain the unity transformation that can be use in a quantum circuit:

$$CU_{xy} = \begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2)e^{i\varphi} & 0 & 0 \\ -i \sin(\theta/2)e^{-i\varphi} & \cos(\theta/2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.63)$$

The truth table can be obtained by applying the gate to the possible input states, following the ‘‘Physical truth table’’ that is represented in fig. 2.27 with its circuit representation and the dynamics of the gate.

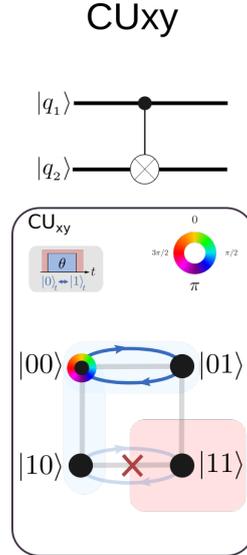


Figure 2.27: CU_{xy} **native gate map**. State diagram of the physical operation for the implementation of Rydberg native gates. In the case of gr-qubits the resonance condition has to be satisfied (e.g., $\Delta = 0$) and the gate parameter $\theta = \Omega_t \tau_g$, where $|V_{c,t}| \gg \Omega_t$. The accumulated phase by the interactions can be compensated by choosing $|V_{c,t}| = 2\pi m / \tau_g$.

This gate is shown using a simple implementation with gr-qubits. However, it is also possible to implement it with gg-qubits or rr-qubits. In these two cases the condition on the interaction strength $|V_{c,t}|$ can be relaxed when the system is constrained to the computational basis with strong diagonal van der Waals interactions.

CPHASE gate

Local phase control is important for quantum algorithms such as Quantum Fourier Transform (QFT) [Wei01], where the use of a specific case of the controlled-phase gate or CPHASE plays a key role. Additionally, an advantage of the always-on interactions is that it can be used to introduce such a phase when the qubit is encoded as gg-qubit. However, this can also be used in gr- and rr-qubit encodings, where the van der Waals interaction strength must be stronger than the dipolar interaction.

The time evolution operator can be summarized in 3 pulses with always-on interactions:

$$\hat{U} = \exp \left\{ -i \left(\hat{\mathcal{H}}_c^{r1} + \hat{\mathcal{H}}_t^{r1} \right) \tau_1 \right\} \times \exp \left\{ -i \hat{\mathcal{H}}_{c,t}^{rrrr} \tau_2 \right\} \times \exp \left\{ -i \left(\hat{\mathcal{H}}_c^{r1} + \hat{\mathcal{H}}_t^{r1} \right) \tau_1 \right\}. \quad (2.64)$$

Obtaining the unitary matrix associated to the gate:

$$\text{CPHASE} = \begin{pmatrix} e^{i\Phi_{00}} & 0 & 0 & 0 \\ 0 & e^{i\Phi_{01}} & 0 & 0 \\ 0 & 0 & e^{i\Phi_{10}} & 0 \\ 0 & 0 & 0 & e^{i\Phi_{11}} \end{pmatrix}. \quad (2.65)$$

This gate becomes an entangling gate when $(\Phi_{00} - \Phi_{11}) - (\Phi_{10} + \Phi_{01}) \neq 2\pi n \quad \forall n \in \mathbb{Z}$. Similar to the previous gate, one can obtain the truth table by applying the obtained matrix to the different possible computational inputs. However, only one input (i.e., $|11\rangle$) demonstrate that both qubits states couple to the Rydberg state, allowing it to accumulate a phase of $\Phi_{11} = (C_6/R_{c,t}^6)\tau_2$.

An overview of the CPHASE is shown in fig. 2.28.

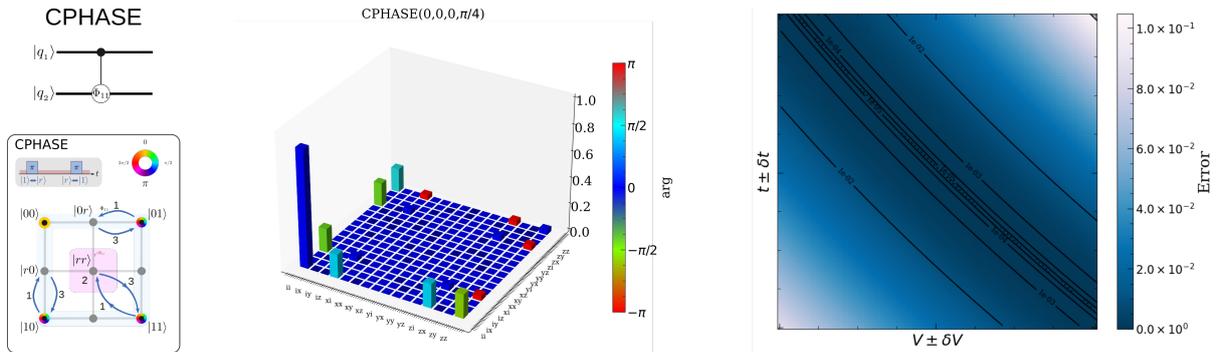


Figure 2.28: CPHASE gate. State diagram of the physical operation for the implementation of Rydberg native gates and the corresponding symbol at the quantum circuit formalism (left). Quantum state tomography for the parameters $\Phi_{00} = \Phi_{01} = \Phi_{10} = 0$ and $\Phi_{11} = \pi/4$ (center) and heatmap of gate performance based on the error due to 10% error rates ($\delta\varphi_i, \delta\theta_i$) in each parameter (φ_i, θ_i) of the two U_{xy} gates simultaneously (right).

Due to the interaction dependency e.g., interaction strength and population transfer, the gate protocol is not as robust as the CU_{xy} . However, this can be improved by using STIRAP for optimal populations to the Rydberg states instead of π -pulses and including

additional fields to control the interaction strength during the protocol e.g., electric-field cancellation and large magnetic fields to increase the splitting between neighboring Rydberg states.

Doppler dephasing and fluctuating light shifts could be cancelled by applying Rydberg dressing and adiabatically varying two-atom light-shifts alternated with spin-echo sequences.

CZ gate

The previous protocol forms a family of phase gates that can lead to an important gate in quantum computing that is used for the implementation of the canonical CNOT gates, assisted by Hadamard gates. The key ingredient for fast execution of CZ gates is the use of the Rydberg blockade. There are various models for the implementation of these type of gates, Jaksch et al [Jak00] introduced two models: Model A (see fig. 2.29) and Model B (see fig. 2.30).

Model A: The model assumes $|V_{c,t}| \ll \Omega$ and consists of applying a global laser pulse over both qubits, with an interaction time where it accumulates a phase $\phi = \Delta t V_{int}$. A final global pulse is added in the protocol, for retrieving the states into the computational space.

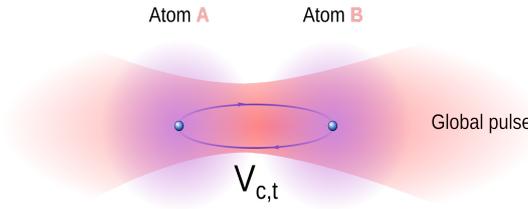


Figure 2.29: **Model A.** Global pulse applied in the atoms A and B with low interaction strength regime between control and target qubits, accumulating a phase ϕ .

Model B: The global pulses model is an appealing implementation in the hardware, although it lacks the flexibility of selecting the specific pair of qubits without adding complexity. Indeed it is possible to add an additional Stark-shift in the non used qubits, but for large quantum registers the requirements in terms of laser power will be considerable. This can be solved by using local addressing using AODs. This second model consists of a laser π -pulse applied over one of the atoms, followed by a 2π -pulse over the second atom and a final π -pulse over the first atom involved. This will lead to an interaction time:

$$\Delta t \approx 2\pi \left(\frac{1}{\Omega_1} + \frac{1}{\Omega_2} \right), \quad (2.66)$$

where $\Omega_1 = \Omega_2 = \Omega$ and the $|V_{c,t}| \gg \Omega$. Therefore, the operation can be summarized with the following time evolution operator:

$$\hat{U} = \exp \left\{ -i \hat{\mathcal{H}}_c^{r1} \tau_1 \right\} \times \exp \left\{ -i \left(\hat{\mathcal{H}}_t^{r1} + \hat{\mathcal{H}}_{c,t}^{rrrr} \right) \tau_2 \right\} \times \exp \left\{ -i \hat{\mathcal{H}}_c^{r1} \tau_1 \right\}. \quad (2.67)$$

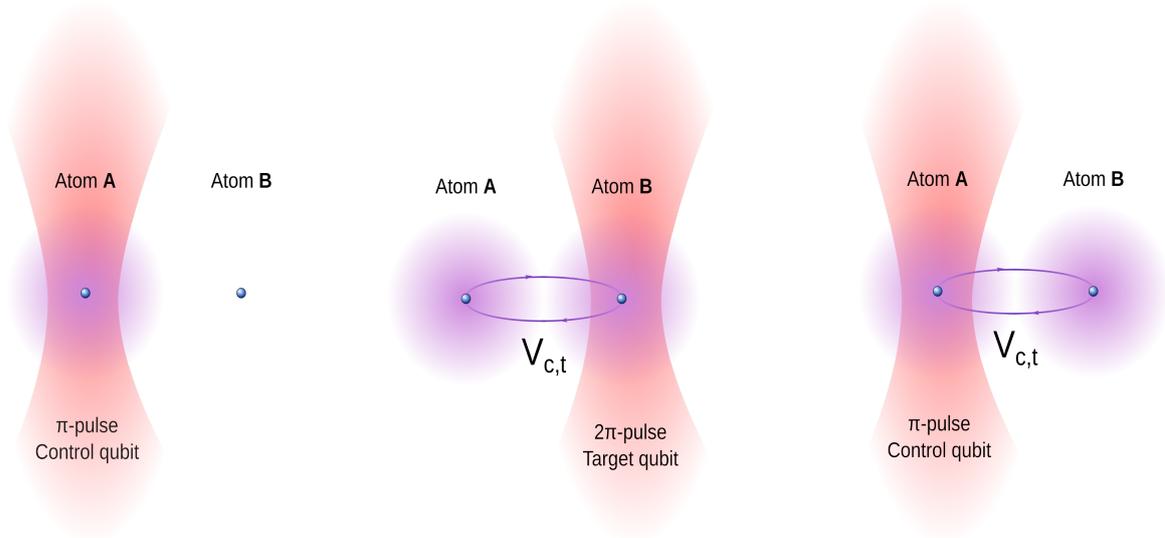


Figure 2.30: **Model B.** A three steps sequence: (left) π -pulse over the control qubit or atom A, bringing the system into $i|rg\rangle$ state. Then a 2π -pulse over the target qubit or atom B and that is detuned from the double Rydberg state $|rr\rangle$ by an interaction strength $|V_{c,t}|$, accumulating a small phase proportional to $\phi = \pi\Omega_2/2|V_{c,t}|$ (center) and a final π -pulse over the control qubit leaving the system in the state $e^{i(\pi-\phi)}|gg\rangle$ (right).

after accumulating a phase proportional to the interaction time Δt .

The gate time (Δt) results from the time taken by the 3 pulses with τ_1 and τ_2 pulse gates, which are dependent of the respective Rabi coupling. When calculating the matrix exponential the obtained unitary matrix associated to the gate is given by:

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (2.68)$$

This gate protocol is easily mapped to gr- and gg- qubits, a similar protocol can be followed with rr-qubit encoding where a third Rydberg state with a much larger principal quantum number n i.e., larger blockade radius, is used for the phase accumulation step.

The physical truth table for this gate is represented in fig. 2.31.

The model B can also be used to perform a CNOT gate when using three π -pulse on the target qubit instead of a single 2π -pulse. These pulses represent the transitions $|1\rangle_t \rightarrow |r\rangle_t$, $|r\rangle_t \rightarrow |0\rangle_t$ and $|r\rangle_t \rightarrow |1\rangle_t$ that in the non-blockaded regime exchange the target state. Additionally, another way to perform the CNOT gate is by including a Hadamard gate before and after and a native CZ in between.

The next 2-qubit quantum gates have a special characteristic intrinsic to the Rydberg platform, which is the parallel implementation of qubit gates over the control and target qubits which is facilitated by the use of global pulses. This subset of gates is promising for the implementation of quantum error correction codes, since the gate satisfy the need of simultaneously implementing a group of gates over the quantum registers.

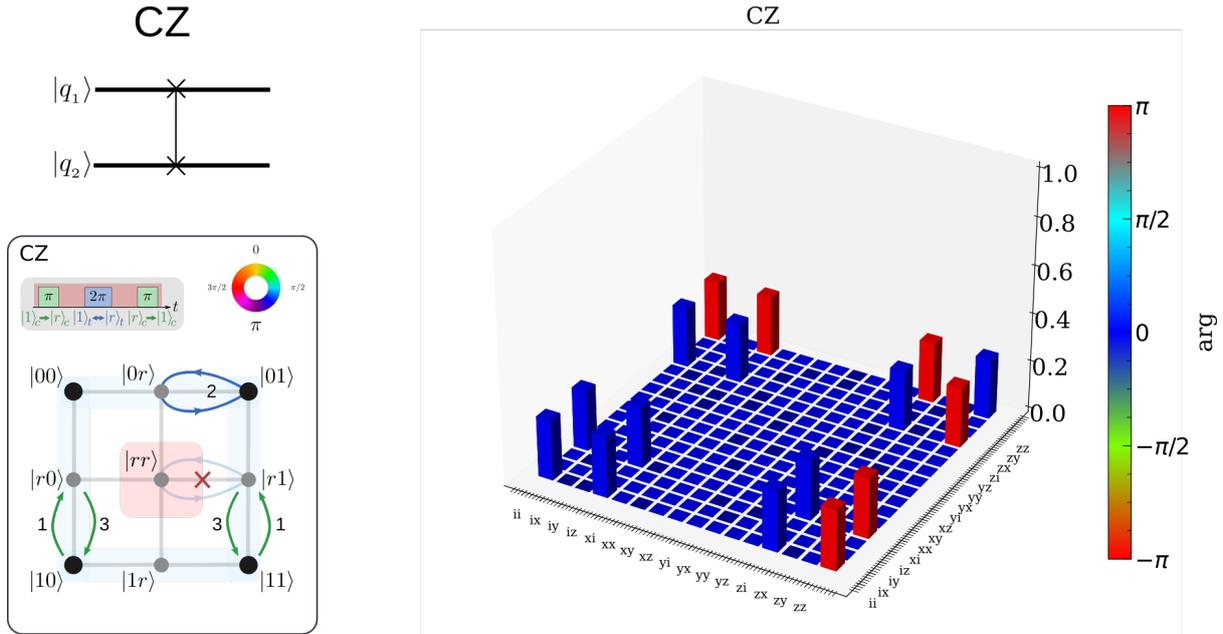


Figure 2.31: **CZ gate**. State diagram of the physical operation for the implementation of Rydberg native gates and the correspondent symbol at the quantum circuit formalism (left). Quantum state tomography for the CZ gate (right).

Recent explorations with these protocols have found optimal time-dependencies of the Hamiltonians using techniques from the well-known field of engineering in optimal control [Wer07] e.g., Gradient Ascent Pulse Engineering (GRAPE) where the main algorithm considers the model A but considering that the interaction strength is larger than the coupling $|V_{c,t}| \gg \Omega$ and thus optimizes the time-dependency of the laser-phase pulse [Jan22][Pag22].

2.4.4 Parallel qubit native gates

pCU_{xy} gate

Typical controlled gates involving two qubits only create conditional physical operations over transitions occurring in the target qubit. However, with the parallel gates we can apply the same conditional over the transitions on both qubits i.e., control and target, simultaneously.

This gate can be summarized as a sequence of Hamiltonian terms in the time evolution operator as:

$$\hat{U} = \exp \left\{ -i \left(\hat{\mathcal{H}}_c^{01} + \hat{\mathcal{H}}_t^{01} + \hat{\mathcal{H}}_{c,t}^{1111} \right) \tau_g \right\}. \quad (2.69)$$

Here the distinction between the control and target qubit does not exist, since the physical operation addressing each qubit individually happens in parallel. The associated unitary transformation depends on two parameters:

$$pCU_{xy} = \begin{pmatrix} \cos(\theta/2) & S(\theta, \varphi) & S(\theta, \varphi) & 0 \\ S(\theta, -\varphi) & \cos^2(\theta/4) & -\sin^2(\theta/4) & 0 \\ S(\theta, -\varphi) & -\sin^2(\theta/4) & \cos^2(\theta/4) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (2.70)$$

where $S(\theta, \varphi) = -i \sin(\theta/2)e^{i\varphi}/\sqrt{2}$, $\Omega = \theta/\sqrt{2}\tau_g$ and the interaction strength is $|V_{c,t}| = 2\pi m/\tau_g \gg \Omega$ when resonance light is used and the accumulated phase is compensated. A maximally entangled Bell state between two qubits (i.e., $(|01\rangle + |10\rangle)/\sqrt{2}$) has been observed by Gaetan et al [Gaë10], describing both qubits realizing collective Rabi oscillations since they are within the blockaded regime and then there is no distinction between both control and target qubit encoded as gr- or dressed gg-qubit.

A simple but general protocol to realize this kind of operation is by starting in the ground state of the system i.e., $|00\rangle$, then $\Omega_c = \Omega_t = \Omega = \pi/\sqrt{2}$ being the Rabi coupling for both qubits by on resonant laser beams $\Delta_c = \Delta_t = \Delta = 0$ at the strong blockade regime $|V_{c,t}| \gg \Omega$. The state diagram of this operation is presented in fig. 2.32.

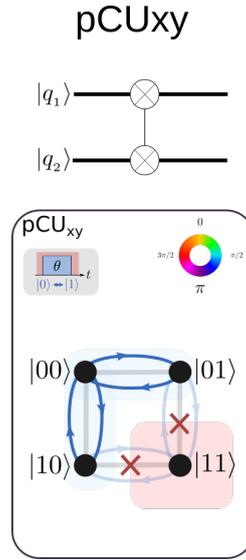


Figure 2.32: pCU_{xy} gate. State diagram of the physical operation for the implementation of the pCU_{xy} Rydberg native gate and the correspondent symbol at the quantum circuit formalism.

pCZ gate

Simultaneous operations over both qubits come natural when considering the Rydberg Blockade based gates. A parallel CZ gate protocol allows the transition of one of the atoms into the Rydberg state, meanwhile the second remains blockaded in the intermediate state $|1\rangle$. This is a two consecutive π -pulse with always-on van der Waals interaction.

This protocol is more susceptible to loss and dephasing compared to a single continuous pulse with global amplitude and frequency modulated laser pulse which minimizes the time in the Rydberg state. A modification to this protocol was proposed and experimentally implemented by Levine et al. [Con22], in this protocol a phase difference of ϕ

is introduced to reduce these effects. The modified sequence by Levine et al [Con22] can be summarized as follow:

$$\hat{U} = \exp \left\{ -i \left(\hat{\mathcal{H}}_{c,t}^{rrrr} + \sum_{\alpha \in \{c,t\}} \hat{\mathcal{H}}_{\alpha}^{r1}(\varphi_{\alpha} = \varphi) \right) \frac{\tau_g}{2} \right\} \times \exp \left\{ -i \left(\hat{\mathcal{H}}_{c,t}^{rrrr} + \sum_{\alpha \in \{c,t\}} \hat{\mathcal{H}}_{\alpha}^{r1}(\varphi_{\alpha} = 0) \right) \frac{\tau_g}{2} \right\}, \quad (2.71)$$

where the blockade strength is larger than the Rabi coupling and detuning i.e., $|V_{c,t}| \gg \Omega_{\alpha}, |\Delta_{\alpha}|$. By choosing $\Omega_{\alpha} = \Omega$, $\Delta_{\alpha} = 0.377\Omega$, $\varphi = 3.90242$ and a gate time of $\tau_g = 2.7328\pi/\Omega$ a parallel CZ gate that can be derived from the CPHASE gate transformation eq. (2.65) with $\Phi_{00} = 0$, $\Phi_{01} = \Phi_{10} = \phi_1$ and $\Phi_{11} = 2\phi_1 - \pi$.

The physical truth table operation is represented in fig. 2.33.

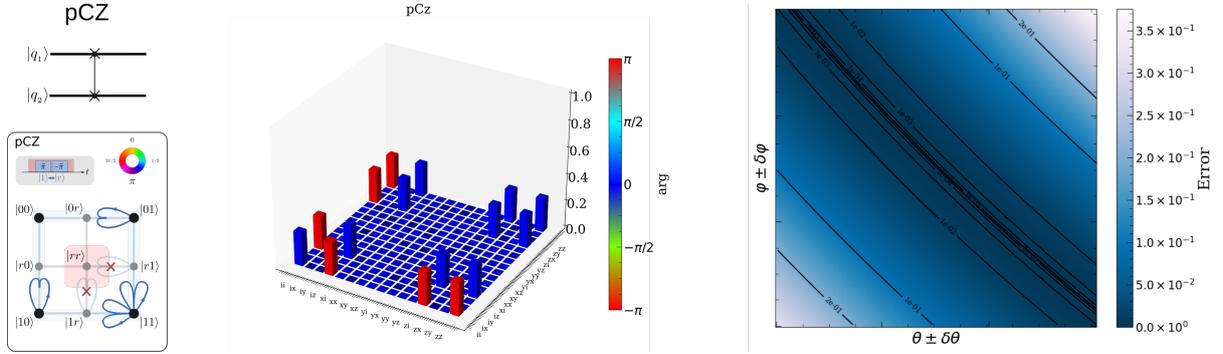


Figure 2.33: *pCZ* gate. State diagram of the physical operation for the implementation of *pCZ* Rydberg native gate and the correspondent symbol at the quantum circuit formalism (left). Quantum state tomography for the the gate (center) and the error performance for values θ and ϕ from the native CU_{xy} gate, considering 10% error rates ($\delta\tau_g, \delta V_i$) in each parameter (τ_i, V_i) gate pulses simultaneously (right).

C_kZ and C_kNOT gates

The phenomenon of the Rydberg blockade is the key of multi-qubit native gates. Previously we described the CZ gate as a blockade gate, therefore a C_kZ becomes a generalization with k control qubits. The unitary transformation is given by:

$$C_kZ = \begin{pmatrix} \mathbb{1}_k & 0 \\ 0 & \hat{Z} \end{pmatrix}, \quad (2.72)$$

where $\mathbb{1}_k$ is the multidimensional identity operator with dimension $2^{k+1} - 2$ and \hat{Z} the Pauli-Z operator applied over the target qubit. Using time modulated resonant beams and minimizing undesired cross talked couplings between control qubits with tuned asymmetric interactions, it is possible to perform this gate. These kind of interactions are achieved by choosing large spatial separation, or choosing two Rydberg states with strong interstate

interactions or using microwave Rydberg dressing, that ultimately enable asymmetric Rydberg blockade. The sequence of the gate follows as:

$$\begin{aligned} \hat{U} = & \exp \left\{ -i \sum_k \hat{\mathcal{H}}_k^{r0} (\Omega_k = \Omega_c, \varphi_k = \pi) \tau_1 \right\} \\ & \times \exp \left\{ -i \left(\hat{\mathcal{H}}_t^{r'1} (\Omega_t, \varphi_t = 0) + \sum_k \hat{\mathcal{H}}_{k,t}^{rr'rr'} \right) \tau_2 \right\} \\ & \times \exp \left\{ -i \sum_k \hat{\mathcal{H}}_k^{r0} (\Omega_k = \Omega_c, \varphi_k = 0) \tau_1 \right\}. \end{aligned} \quad (2.73)$$

where k is the control qubits index and t is the target qubit index. Here the resonance condition $\Delta_k = \Delta_c = 0$ and $|V_{k,t}| \gg \Omega_t$. Generalizations of this gate include shortcuts to adiabaticity [She19] and atoms coupled to optical cavities [Lia15]. A generalization replacing the \hat{Z} operation by a $U_{x,y}$ using three pulses. Using detuned laser beams in the anti-blockade regime it is possible to create a multi-qubit phase gate [Shi18].

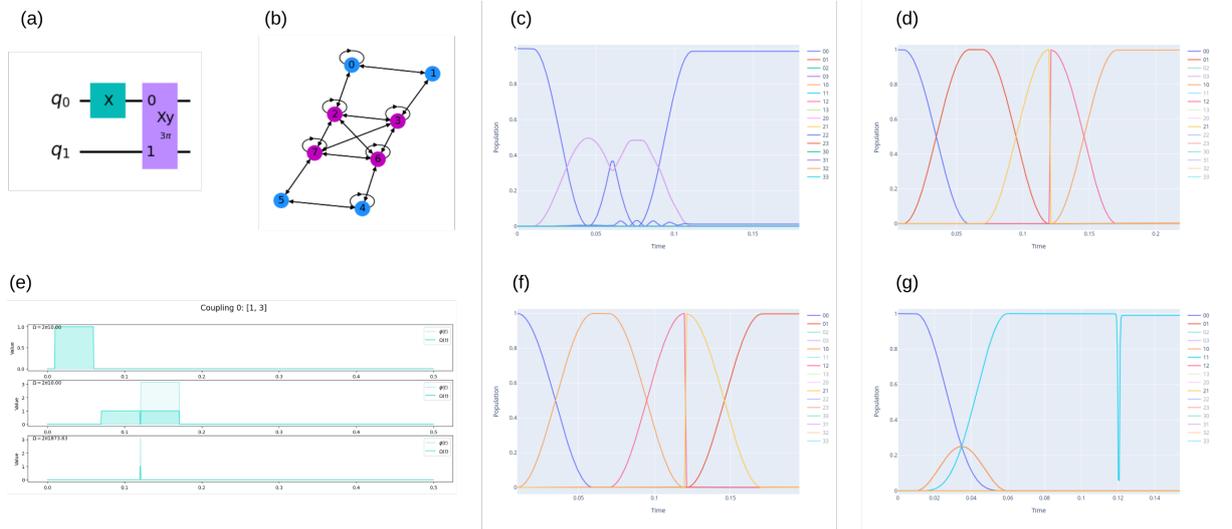


Figure 2.34: **XY new gate protocol.** (a) Quantum circuit representation. (b) State graph of the quantum register of 2 atoms with 4-level system. (e) Schedule of the Hamiltonian terms including an extra pulse for an X -gate for preparing the state. (c) Dynamic simulation for the input states to the gate: 00 (d) Dynamic simulation for the input states to the gate: 01 (f) Dynamic simulation for the input states to the gate: 10 (g) Dynamic simulation for the input states to the gate: 11. Simulations realized with AQiPT (see chapter 3).

$CNOT^k$ gate

Differently than the previous gate, the $CNOT^k$ gate uses a single control qubit to conditioned multiple target qubits instead. $CNOT^k$ gates play an important role in the development of topological quantum error correction codes being used for the creating of plaquettes for syndrome's parity measurement. Different techniques have been proposed for realizing this family of gates, for instance: use of electromagnetic induced transparency (EIT) to entangle multiple qubits with a single control qubit [Ras20] and the adiabatic evolution of dark-states and resonance exchange interactions [Kha20].

XY gate

In addition to the previous gates, this work introduces a new protocol for long-range range dipolar interactions. It is composed of 4 pulses. These pulses consist of:

1. π -pulse on both qubits coupling $|0\rangle$ to $|r\rangle$ states. i.e., $\tau_1\Omega_c = \pi$
2. Two 2π -pulse with a π -phase difference performing the $|1\rangle \rightarrow |r'\rangle$ transition in the presence of a resonant dipolar exchange interaction. i.e., $\tau_2\Omega_t = 2\pi$. Further improvements can be done with optimal control of the amplitude together with a smooth variation of the phase from one pulse to another.
3. π -pulse on both qubits coupling $|r\rangle$ to $|0\rangle$ states.

The following time evolution operator summarizes this protocol:

$$\begin{aligned}
 \hat{U} = & \exp \left\{ -i \sum_{\alpha \in \{c,t\}} \hat{\mathcal{H}}_{\alpha}^{r^0} (\Omega_{\alpha} = \Omega_1, \varphi = \pi) \tau_1 \right\} \\
 & \times \exp \left\{ -i \left(\hat{\mathcal{H}}_{c,t}^{rr'r'r} + \sum_{\alpha \in \{c,t\}} \hat{\mathcal{H}}_{\alpha}^{r'1} (\Omega_{\alpha} = \Omega_2, \varphi = \pi) \right) \tau_2 \right\} \\
 & \times \exp \left\{ -i \left(\hat{\mathcal{H}}_{c,t}^{rr'r'r} + \sum_{\alpha \in \{c,t\}} \hat{\mathcal{H}}_{\alpha}^{r'1} (\Omega_{\alpha} = \Omega_2, \varphi = 0) \right) \tau_2 \right\} \\
 & \times \exp \left\{ -i \sum_{\alpha \in \{c,t\}} \hat{\mathcal{H}}_{\alpha}^{r^0} (\Omega_{\alpha} = \Omega_1, \varphi = 0) \tau_1 \right\}.
 \end{aligned} \tag{2.74}$$

This protocol fulfills the resonance condition $\Delta_{\alpha} = 0$, $\tau_1\Omega_1 = \pi$, $\tau_2\Omega_2 = 2\pi$ where $\tau_2 = \Theta/V_{c,t}$. The involved Rydberg states are dipole-coupled e.g., nS and nP . The duration of the pulses determines the rotation angle of the gate, meanwhile the Rabi coupling is calibrated to fit two 2π rotations. The unitary transformation is given by the matrix:

$$\text{XY}(\Theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta/2) & -i \sin(\Theta/2) & 0 \\ 0 & -i \sin(\Theta/2) & \cos(\Theta/2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.75}$$

By choosing the angle of the gate to be $\Theta = 3\pi$, this gate performs a canonical iSWAP gate. By using a polarized microwave field it is possible to drive transitions in the Rydberg manifold with high fidelities. The gate protocol has been simulated for different input states values e.g., 00, 01, 10 and 11 (fig. 2.34) using a quantum master equation simulation of the time-dependent Hamiltonian of a 4-level system with 2 ground-states and 2 Rydberg states with always-ON interactions. This protocol is included in the transpiler of the software detailed in chapter 3.

XY gates enable the application of error correction over leakage errors of Rydberg states coming from other quantum gates such as CZ gates. This method can transform leakage states in data qubits, present in a checkered-board QEC code [Fow12], into

computational states. Additionally, this type of technique can be used to calibrate and characterize the controlled-phase gates. The physical truth table is shown in fig. 2.35.

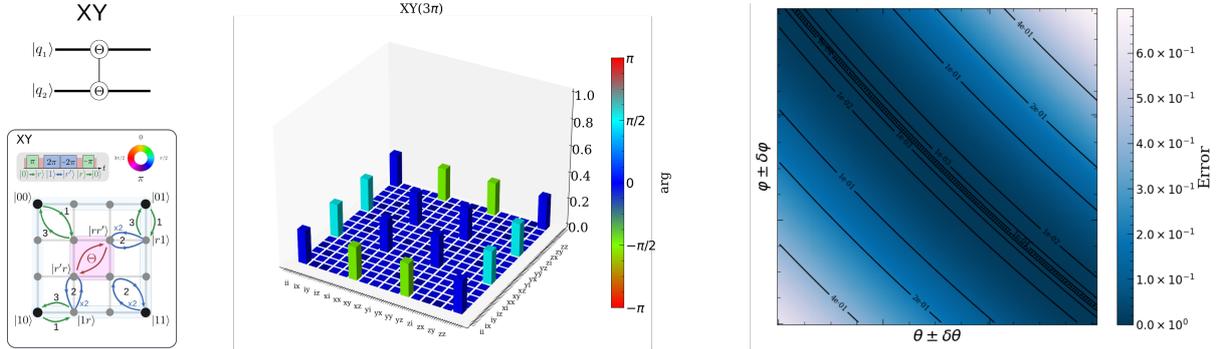


Figure 2.35: **XY gate**. State diagram of the physical operation for the implementation of the XY Rydberg native gate (left). Quantum state tomography of the gate (center) and the gate error performance as function of the parameter Θ and the interaction time τ with a 10% error rates ($\delta t_i, \delta V_i$) in each parameter (t_i, V_i) in the gate pulses simultaneously with $\Theta = 3\pi$ (right).

2.5 Current state of the art

Measuring the performance of quantum devices is in general a difficult task since the same nature of a possible advantage is a challenge to design a meaningful figure of merit. In the last years IBM has introduced quality (Quantum Volume [Cro19]) and speed (Circuit-Layer-Operations-Per-Second [Wac21]) metrics for quantum circuits which are inspired by the superconducting platform. In general it is possible to roughly estimate the quantum circuit fidelity in terms of the gate fidelity and the W number of qubits (width) and the D number of layers in the circuit (depth), see fig. 2.36.

$$\mathcal{F}_{qc} = \mathcal{F}_g^{W \cdot D}. \quad (2.76)$$

Although fidelity is still a handy tool for describing the performance, a compatible metric for the atomic platform including computers and simulators that we denote as D_\square (see fig. 2.37) is more useful.

$$D_\square \equiv \operatorname{argmax}_{n \leq N} \min[n, d(n)]. \quad (2.77)$$

2.5.1 Quantum error correction and noise mitigation

With the gate toolbox it is natural that the platform presents errors due to noise, interactions with the environment and imperfections in the hardware and methods. A specific type of quantum algorithms, called quantum error correction codes (QEC codes), are dedicated to correct these errors at the level of the quantum gates. This means that errors are corrected at the qubit sub-space. However, as it was presented before, quantum systems are commonly more complex than 2-level systems and therefore other types of

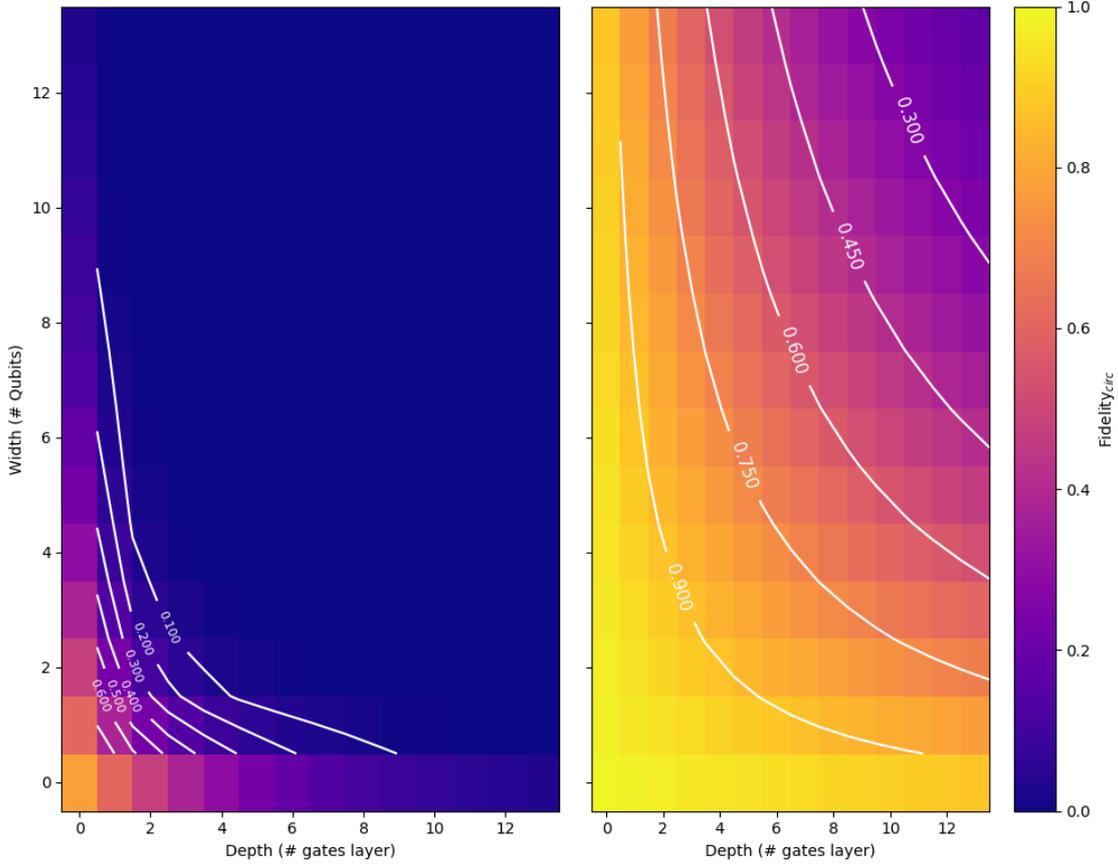


Figure 2.36: **Circuit fidelity.** Fidelities for the lower and higher gate fidelity in experimental implementations of Rydberg gates.

errors appears in the system which are linked to the properties and phenomena of the quantum hardware. In this section I will address the possible sources of errors in two parts: Physical error channels and Classical manipulation error channels.

Physical errors

These type of errors depend on the properties of the atoms, their interaction with light and the energy state properties. A good estimation example for Cs atoms can be found in [Gra19] and an implementation of correction techniques including Black-body radiation of Rydberg states for Rb is presented in [Con22].

- 1. Rydberg excitation dephasing.** In the case of a 2-photon Rydberg excitation, which usually includes two wavelengths, there is a mismatch on the wavevector inversely proportional to the difference of the wavelengths ($\Delta\lambda$), leading to stochastic values of a phase (ϕ) in atoms excited to Rydberg states for τ_{Ryd} . Thus, the Doppler dephasing error according to [Wil10] and [Gra19] is given by the mean stochastic phase:

$$\varepsilon_{RED} = \langle e^{i\phi} \rangle = e^{-\left(\frac{\tau_{Ryd}}{T_D}\right)}, \quad (2.78)$$

where the Doppler dephasing rate $T_D = \frac{\Delta\lambda}{2\pi} \sqrt{\frac{2M_{atom}}{k_B T_{atom}}}$ with M_{atom} and T_{atom} the mass and temperature of the atom respectively. This type of errors will limit the

$D_{\square}(\tau_g, N)$ vs T_{vac}

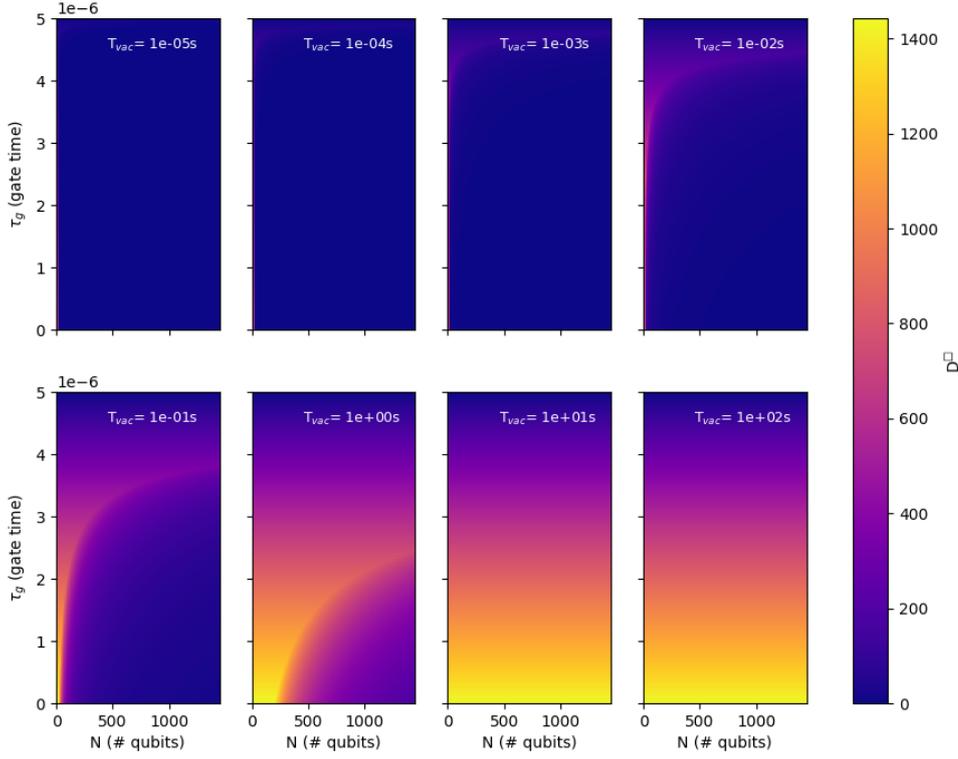


Figure 2.37: D_{\square} . Figure of merit for the Rydberg platform and quantum simulations for different values of the gate time in eq. (2.77).

time of gates that requires Rydberg interactions or phase accumulation in the Rydberg manifold, but also can be suppressed by modulating the laser's phase or using single-photon Rydberg excitations.

- Rydberg lifetime.** When the atom is in a highly excited Rydberg state, the atom is coupled to the vacuum bath driving the atom to be out of the Rydberg manifold. Therefore the error due to this decay process is determined by the Rydberg lifetime Γ_r and given by:

$$\varepsilon_{RHL} = \sum_{\alpha \supset |r\rangle} 1 - e^{-(t_{\alpha} \Gamma_r)}, \quad (2.79)$$

where t_{α} is the time on the state α that is coupled to the Rydberg subspace.

- Intermediate states lifetime.** Similar to the previous type of error, in a two-photon Rydberg excitation there is an intermediate state coupled to the Rydberg state that is detuned. This leads to photon scattering due to spontaneous emission with probability:

$$\varepsilon_{IHL} = \frac{\pi \Gamma_{|i\rangle}}{4 \Delta} \quad (2.80)$$

where $|i\rangle$ is the intermediate state with lifetime $\Gamma_{|i\rangle}$ and detuning Δ . This expression is valid for equal coupling strength and in the blockade regime.

| Year, reference | Qubit | Operation | Fidelity | T_2^* | τ_g |
|-----------------|--------------------------|---|--|-------------------------|--|
| 2016, [Orú19] | ^{133}Cs (gg) | pCU $_{xy}(\pi)$ | $\geq 0.81(2)^{\text{i,iii}}$ | | $2 \mu\text{s}$ |
| 2017, [Zen17] | $^{87-85}\text{Rb}$ (gg) | Heteronuclear CZ \rightarrow CNOT | $0.73(1)^{\text{ii}}$ | | $6 \mu\text{s}$ |
| 2018, [Lev18] | ^{87}Rb (gr) | pCU $_{xy}(\pi)$ | $\geq 0.97(3)^{\text{i,iii}}$ | $4.5(1) \mu\text{s}$ | 177 ns |
| 2018, [Pic18] | ^{133}Cs (gg) | pCU $_{xy}(\pi)$ | $\geq 0.81(5)^{\text{i,iii}}$ | $10(2) \text{ms}$ | $1.85 \mu\text{s}$ |
| 2019, [Gra19] | ^{133}Cs (gg) | CZ \rightarrow CNOT | $0.89^{\text{i,iii}}$ | $\leq 1.6 \text{ms}$ | $1.12 \mu\text{s}$ |
| 2019, [Lev19] | ^{87}Rb (gg) | pCZ pCZ \rightarrow CNOT C $_2$ Z \rightarrow Toffoli | $\geq 0.974(3)^{\text{i,iii}}$ $\geq 0.965(3)^{\text{i,ii}}$ $\geq 0.870(4)^{\text{i,ii}}$ | | $0.4 \mu\text{s}$ $1.2 \mu\text{s}$ |
| 2020, [Zha20] | $^{88}\text{Sr}^+$ (gg) | CPHASE \rightarrow CZ | $0.78(3)^{\text{iii}}$ | | 700 ns |
| 2020, [Mad20] | ^{88}Sr (gr) | pCU $_{xy}(\pi)$ | $\geq 0.991(4)^{\text{i,iii}}$ | $\approx 2 \mu\text{s}$ | 51 ns |
| 2023, [Eve23] | ^{87}Rb (gg) | CZ | $0.993(5)$ | $3 \mu\text{s}$ | 239 ns |
| 2024, [Blu24] | ^{87}Rb (gg) | CZ CCZ | $0.993(5)$ $0.998(5)$ | $3 \mu\text{s}$ | 239 ns |

Table 2.3: Reported quantum operations and quantum gates realized with Rydberg qubits and performance parameters. T_2^* refers to the qubit coherence time measured via Ramsey interferometry without spin echo pulses (coherence times with echo pulses are typically an order of magnitude longer). τ_g is the operation time. For results prior to 2016, see the review by [Saf16] ⁱFidelity corrected for SPAM errors, ⁱⁱgate fidelity and ⁱⁱⁱentanglement (Bell) fidelity. For reference, a fidelity of $\mathcal{F} = 0.99$ corresponds to an estimated achievable circuit depth of $D_{\square} = 10$.

4. State leakage. A rotation of the blockaded states are produced by imperfect but strong blockades. The error is defined by the mean blockade shift \mathbf{B} [Saf10]:

$$\varepsilon_{SL} = \frac{\Omega^2}{8\mathbf{B}^2}. \quad (2.81)$$

5. Atom motion. One characteristic of the atomic platform is that atoms after performing a task are reloaded for new task runs, this means that the position and initial velocity of the atoms are not always exactly the same. Therefore the measurements are usually reported as an average of these multiple realizations. Graham et al [Gra19] calculated the population error in a state $|i\rangle$ in a π -pulse and in the Rydberg state for a 2π -pulse, considering a Gaussian distributed atom position i.e., $\rho(\vec{r})$ with 2D width σ and axial width σ_z due to a Gaussian beam field amplitude distribution for each Rydberg excitation laser $f(\vec{r})$.

For the atom probability being in the state $|i\rangle$:

$$\langle P_{|i\rangle}(t) \rangle = \int \rho(\vec{r}) \left[\cos^2 \left(\frac{\Omega_R t}{2} f(\vec{r}) \right) + \frac{\Delta^2(\vec{r})}{\Omega_R^2(\vec{r}) + \Delta^2(\vec{r})} \sin^2 \left(\frac{\Omega_R t}{2} f(\vec{r}) \right) \right] d\vec{r}, \quad (2.82)$$

and for the Rydberg state $|r\rangle$:

$$\langle P_{|R\rangle}(t) \rangle = \int \rho(\vec{r}) \left[\frac{\Omega_R^2(\vec{r})}{\Omega_R^2(\vec{r}) + \Delta^2(\vec{r})} \sin^2 \left(\frac{\Omega_R t}{2} f(\vec{r}) \right) \right] d\vec{r}, \quad (2.83)$$

where Ω_R is the Rabi coupling and $\Delta(\vec{r}) = \Delta_0 + \Delta_1 f_1^2(\vec{r}) + \Delta_2 f_2^2(\vec{r})$ is the two-photon detuning and the constant detuning and the Δ_1, Δ_2 are the Stark-shift coefficients.

6. **Laser noise.** It is a hard type of error to quantify but its source comes from laser beams' phase and the noise in the intensity which produce additional unwanted Stark-shifts.
7. **Cross-talk.** When atoms are excited into Rydberg states the addressed atoms can influence the state control over other pairs while blockading (i.e., imperfect or undesired Rydberg blockade) or interacting resonantly (i.e., dipole-dipole interactions) or creating energy shifts. This can be avoided with clever timing of the operations and synchronizing the times of the atoms in the Rydberg manifold. Black-body radiation (BBR) also contributes to these type of errors since the atoms in a Rydberg state will have a diffuse probability of being in a Rydberg's neighboring states.
8. **SPAM errors.** The State-preparation and Measurement (SPAM) errors include those that determine the atom loss probability in the traps, due to background collisions after being measured. Similarly, when the atom state is measured errors can be introduced by fluctuations in the beam polarization and leakage errors by interrogation beams. Finally the preparation of the atom state by using for example the optical pumping could introduce errors in the final prepared state due to leaks into states generated by fluctuations in magnetic fields that change the selected magnetic state.

Classical manipulation errors

Besides all the errors listed in the previous section, that are more related to the quantum system, it is also important to consider errors coming from the classical control electronics. Electronic noise is always present in devices and they can considerably affect the outcoming state of an operation due to electronic noise.

The first type of classical noise is the distortion of electrical signals coming from the waveform generators. This noise is intrinsic to the amplified RF signals that are used to drive the opto-mechanical components controlling the laser fields. These distortions can be characterized or analyzed with Filter functions in order to pre-distort the waveforms in order to obtain a desired shape using open-loop optimization of non-linear distortions characterized by Volterra series [Sin23]. The second type is linked to the time-precision of the waveforms played in the classical control, that can enable or tune interactions at undesired times.

All the above mentioned errors can be directly suppressed at the hardware level by applying non-unitary processes over the multi-level (qudit) space. Proposals for the continuous correction of BBR and spontaneous decay to build fault-tolerant protocols have already been proposed for the platform by I. Cong et al [Con22]; however, standard error correction codes like surface code using robust native gates e.g., pCZ , $iSWAP$, $C_k NOT$ and $CNOT^k$ and unaltered electrical waveforms can be added in order to reach low physical and logical errors up to fault-tolerant thresholds. First experimental implementations of continuous measurement and correction of errors [Gra23b] include real-time feedback based on new generation controllers are described in more detail in chapter 3.

2.6 Quantum Full-stack systems

Current experiments are the prototypes of what we consider quantum backends, ultimately their commercial version for serving as quantum computers and simulators will be scaled down but still conserving their most fundamental elements. These elements can be abstracted in two types: hardware system and software system, these compose what it is commonly referred as a quantum full-stack. Clearly each component might differ from one implementation to another, but conserving the essence of the functionality and the key properties, this is what I call, specifications.

There is an additional type of specification which is constrained by the first two and the nature of the Quantum Processor Unit (QPU), which is the computational specification. In chapter 3 I will analyze one possible specification based on an implementation of a full-stack for the ultracold Rydberg atom platform.

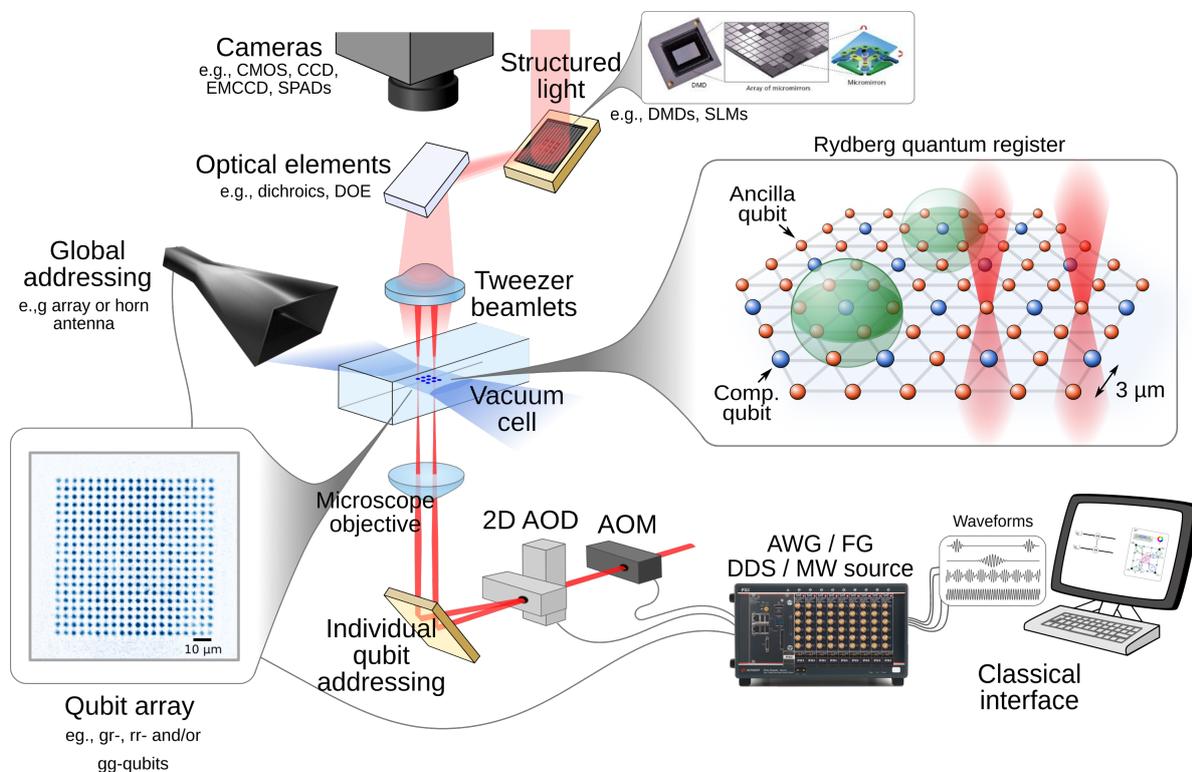


Figure 2.38: Schematic of the ultra-cold Rydberg platform.

2.6.1 Hardware system

In the scheme from fig. 2.38 we observe the vision of a generalized Rydberg platform. In chapter 4 and chapter 5 our approach in Strasbourg for the backend and the QPU is explained in more detail. Here, I will break-down the architecture in terms of instruments and classical devices.

- **Atom source**

The source represents the building block of the substrate of the QPU. The atoms used in the architecture can be alkaline (for example: ^{39}K , ^{87}Rb , Cs) or alkaline-

earth (e.g., ^{88}Sr , ^{174}Yb). The atoms are produced from enriched metals with these elements or from abbreviation ovens (see table 2.4).

The atoms are mainly kept in ultra-high vacuum (UHV) at approximately 6.9×10^{-9} bar in a vacuum chamber made of galvanized steel equipped with viewports or glass cells. The former design offers lower surface effect when using Rydberg states, meanwhile the second option allows a larger optical access and closer view of the atoms. More advantages and disadvantages from both designs are listed below:

| Type | Typical atomic elements | Advantages | Disadvantages |
|---------------------|--|---|---|
| Alkaline | $^{85-87}\text{Rb}$, ^{39}K , ^{133}Cs | Simple atomic structure i.e., single valence electron. Commercial laser wavelengths. Well-known laser cooling techniques e.g., Doppler, evaporative and Sisyphus cooling. | State-of-the-art gates. Spin models. |
| Alkaline-earth-like | ^{171}Yb , $^{87-88}\text{Sr}$, ^{40}Ca | Two electrons in the outer shell which pair off eliminating hyperfine interaction. Singlet and Triplet states that allows to use nuclear and electronic states for qubit encoding. Broad and narrow line transitions for efficient cooling. Presence of long-life and coherent clock and meta-stable states. QND measurement techniques with secondary electron. Significant presence of magic wavelengths. | Complex laser cooling and trapping which can lead to error leakage. Limited experience in the community with the atomic physics of atomic elements such as Ytterbium. |
| Molecules | RbCs | Presence of vibrational states e.g., qudit architectures. Hybrid implementations with solid-state and photonic devices. Simpler scalable manufacturability. | Control and addressability due to the complex energy level structure that produce a dense spectrum of energy levels. |

Table 2.4: **Atomic elements.** Summary of the possible atomic and molecular elements that can be use for the QPU, considering advantages and disadvantages. The most commonly used alkaline is rubidium and for alkaline-earth-like strontium and Ytterbium are commonly used, although calcium is widely used in the ion trap community for Rydberg ions [Mok20].

- **Lasers**

Highly coherent light sources are important for the manipulation of atoms, used for cooling and trapping atoms, excitation transitions, atom arrangement, induced light-shifts and structured light.

Power, frequency and phase stabilization are critical to avoid noise and errors due to the atom-light interactions, creating leakage to undesired states, slower state transitions, short trapping of atoms (i.e., erasure errors). Additionally, the need for a large number of wavelengths besides existing new techniques for multiple wavelength laser locking techniques, will require very broadband optics which typically reduce the transmission efficiencies and larger optical aberration.

- **Cameras and detectors**

Readout in atomic architectures has been extensively studied and fast techniques for readout represent one of the important challenges in optimization. Atomic states are currently measured with imaging, since it represents a scalability that goes with the number of qubits. There are three main types of camera technology: CCD, CMOS and SPAD, usually connected to dedicated FPGA boards for the data loading and transmission. Currently the limitation lies in the time-scale of $< 1\text{ms}$ with quantum efficiencies of 98% for EMCCDs, 90% for CMOS and 50% with SPAD cameras.

There are many advantages and disadvantages of using each type of cameras, table 2.5 summarizes them.

| Type | Properties |
|-------------|--|
| SPAD arrays | Low quantum efficiency $< 10\%$ with low noise. Low temporal resolution under nanosecond scale with high frame rate. |
| sCMOS | Quantum efficiency of 25% with low noise. Nanosecond time resolution. |
| CCD | High quantum efficiency of 85% with low noise at milisecond time-scale. |

Table 2.5: **Camera technologies.** Summary table of the camera technologies for realizing acquisition of fluorescence or saturated absorption imaging of atomic arrays.

Rydberg atoms can also be detected with field ionization, and therefore can be detected with avalanche sensors such as Micro-Channel Plates (MCPs), however this type of readout requires a more complex method for spatial resolution.

- **Structured light devices**

The substrate of the quantum register is generally programmed as a real-space array with the coordinates of the locations of the qubits (i.e., real-space) or with the phase and frequencies of such an image that projects the Fourier transform in the Fourier plane. The latter allows programming the wavefront light-phase that can be used for trapping certain states and to reduce dephasing noise. Here it is possible to use of Digital Micromirror Devices (DMDs) or Spatial-Light-Modulators (SLMs) with different advantages and disadvantages, these are summarized in the table below (table 2.6).

| Type | Number of qubits/atoms | Properties |
|----------------------------------|----------------------------|---|
| Spatial Light Modulator (SLM) | ~ 400 | Re-configurable geometries. Low electronic noise. High diffraction efficiency $> 90\%$. 120Hz refreshing rate. |
| Digital-Micromirror-Device (DMD) | > 400 | High power threshold damaged. Total efficiency $< 15\%$. Fast refreshing rate of 22KHz. |
| Optical elements | 49 (DOE) > 1000 (MLA) | High power threshold damaged. Not reconfigurable |

Table 2.6: **Structured light devices.** Summary of the technologies to generate arrays of optical tweezers that is imprinted over the atomic cloud.

- **Light modulators**

Optical modulators play a fundamental role in changing detunings and laser beam intensities, including on/off switching. The modulators work based on acousto-optical (i.e., AOMs and AODs) and electro-optical effects (i.e., EOMs) driven by RF

electrical signals. These modulators are used for local addressing for gate operations and atom rearrangement.

- **Optical system**

Large numerical aperture objectives are crucial for accessing the atoms and therefore the qubits are trapped with μm distances. Furthermore, it enables the concentration of laser beams from structured light into extremely confined regions, ensuring the precise positioning of atoms at very low temperatures.

- **Free-field emitters**

Single and array antennas and electrodes are often used to drive the atoms with electric and electro-magnetic fields in free space. These fields in the case of microwaves and radio-frequencies correspond to some hundreds of MHz to tens of GHz. These emitters are used to drive atomic transitions globally, since local addressing with such a large wavelength is a complex task.

- **Analog/Digital signal generators**

Classical devices for generating waveforms such as: Direct-Digital-Synthesizers (DDSs), Arbitrary Waveform Generators (AWGs), Function Generators (FG) and DC sources are used to drive and orchestrate the devices along the apparatus. If they require a trigger: TTL signals, often referred to in this work as Digital signals are used to drive the device. On the other hand, if the device is a parametrized control, then an analog signal is used. The latter allows not just to have a full control on an envelope signal, but also the carrier frequency of the signal, often required in devices like light-modulators.

- **Classical controller**

A classical computer is required for the general control of the devices, for monitoring and data acquisition, it is the classical interface for the user.

2.6.2 Software system

In delving into the software structure of the full-stack, it is imperative to first establish a comprehensive understanding of the pivotal role this software plays in computers and simulators. This subsection explains the fundamental features and functionalities of a software system for this architecture device.

The experimental apparatus initially requires a general purpose software that is open to generate arbitrary sequences and that usually is open for integration of new elements and based on a graphical user interface for monitoring certain quantities.

On the other hand, software for quantum computers and simulators, although having complete flexibility and control, keeps the software, hardware, and system specifications fixed. In this section I will expand on the complete stack of quantum computers and simulators in a general way, while in the next chapter (chapter 3) I will expand these concepts with the software system developed during this thesis.

The full-stack system is depicted as a hierarchical and non-oriented diagram (see fig. 2.39), offering a visual representation of independent but interconnected components organized at specific levels based on the degree of abstraction associated with their corresponding actions and roles. In the diagram below it is presented a Rydberg quantum

computer system stack inspired on the work from other approaches based on different architectures [Fu17], [Bal21].

The stack presented in this work is divided in three main blocks denoted as: **Frontend**, dedicated to quantum information processing; **Middleware**, dedicated to bridge the low-level machine instructions with the high-level quantum algorithms and tasks; **Backend**, which is where the quantum processor unit layer is located and where the classical control system together with the quantum system and the quantum-classical elements lie.

Additionally, a parallel accelerating component is a **Digital Twin**, which offers simulation tools at the different levels e.g., **Simulator for Quantum Circuits**, **Dynamic Simulator** and **AMO Physics simulator**. The simulations contribute to the verification and calibration tasks. In the next chapter (chapter 3) I will dive into details about the software system developed during this work.

2.6.3 Frontend

On the more external level the applications that are desired to be executed in a quantum computer or simulator (e.g., for optimization tasks). It is composed by a sequence of **Quantum Instructions** that are part of a finite set of quantum commands (e.g., Shor factorization), as it happens in a programming language in classical computing. These instructions are the result of one or more **Quantum Primitives** (e.g., Quantum Fourier Transform - QFT). Which can require verification or not from a simulation or the quantum system in the **Quantum Verification** component. For these algorithms to be fault-tolerant it is required to be corrected with the **Quantum Error Correction** component, being able to detect and correct errors using a correction code scheme e.g., surface code. The **Gate compiler** compresses the quantum circuit of canonical gates and reorganizes the gates in the circuit depending on the real quantum register map and the gates that can be reduced whenever possible e.g., using the unitarity of the quantum gates. The circuit is represented in terms of elements of the **Canonical Gates** component, that are universal in quantum computation. Along this abstraction block there is a manager component of the different processes called **Runtime**, managing the repetition of the QEC code, the repetitions of the instruction for collecting statistics etc.

2.6.4 Middleware

The **Quantum Gate Transpiler**, translates and compiles the **Quantum Native gates** into the canonical gates. Each native gate is calibrated for each qubit using the parameters of the **Waveform operation** with the **Waveform Calibration** component. Each pulse operation associated with the gate operation can be optimized with the tools within the **Quantum Optimal Control tools** component and electrical signals can be calibrated as well. The parser that analyzes which physical qubits are assigned to the qubits in the quantum circuit and constrains the gates accordingly with the connectivity map is the **Register Mapper**.

2.6.5 Backend

This is the lower layer of abstraction of the stack and connects with the middleware via the **Device Scheduler** component, which revises the pulsed gates into a device set of instructions. The **Initial Configuration** parameters as well as the specifications (i.e.,

hardware, software and atom) in the `Initial configuration` component are used to construct the assembly program of the QPU. It is crucial to have the `Devices Drivers` e.g., higher-level controllers and the `Device Instructions` that are used in the schedule. The component of the block `Atom array generation` is dedicated to produce given geometries of the atomic processor or to perform re-arrangement of atoms for defect-free arrays [Sch20].

First, it is important to generate the substrate of the “quantum-chip” which in the case of atoms is generated in every new shot. Therefore it is important to have a sequence of `Device Primitives` and `Atomic Primitives` to generate the substrate for qubits. These primitives are composed from more fundamental and abstract components which are `Data`, `Digital` and `Analog` commands which are generated from the `Firmware of Passive, Active and Supporting devices`. Ultimately, the `Quantum physical system` i.e., ultracold atoms located in the core of the quantum computer or simulator, manipulating and carrying the quantum information and the conform the `Quantum Processor Unit (QPU)`.

An important part of this layer to work is the sub-block with all the specification of the backend i.e., `Hardware specs.`, `Software specs` and `Atom specs.` that give the awareness of the backend components, layout, devices, instruments, firmware and calibration to the software system, together with the `Channel mapper` that contains the map of the devices channels between each other and the `Data` for storage and analysis.

Summarizing, I have now introduce an assembled set of building blocks for quantum computing with ultracold Rydberg atoms including qubit encoding, a universal quantum native gate set. Furthermore, I have analyze the possible errors and quantum software stack architecture for quantum computers and quantum simulators.

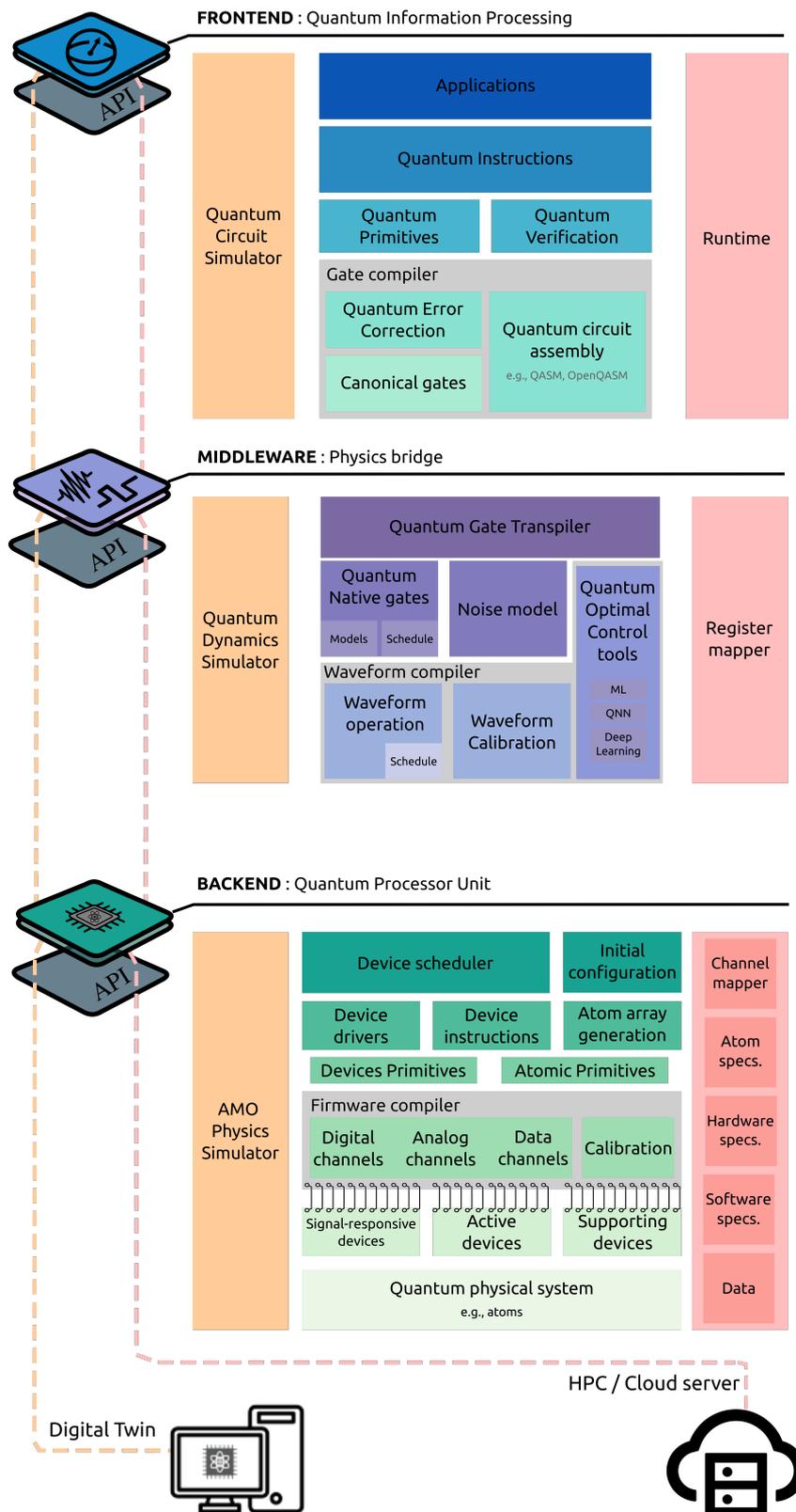


Figure 2.39: **Quantum full-stack**. Scheme of the full-quantum stack proposed in this work for the ultra-cold Rydberg platform divided in three layers: frontend, middleware and backend, accompanied by a **Digital Twin** and assisted by **High-Performance-Computer (HPC)/Cloud server** for verification of the devices' task and acceleration of parallelizable tasks for the execution and optimization of instructions along the stack.

Quantum Software Systems for ultra-cold Rydberg atoms architectures

“A computer is a state machine. Basically, all it is, do something and then find the next state to go to.”

K.Thompson (2019)

In this chapter I will summarize the current state-of-the-art for gates and qubits, including manipulation techniques for qubit addressing. Subsequently the experimental upgrades I developed in the lab in the control hardware and software will be discussed. The experimental noise and error analysis together with the link challenges are also addressed. The chapter ends with a recap of the experimental upgrades and a proposal for ongoing upgrades.

3.1 Hardware control system

Quantum systems are very fragile and in the case of the atomic platform the natural way of control and manipulate the states is via the interaction with light and electromagnetic fields produced by lasers and signal generators. Signal generators play a fundamental role for controlling quantum computers and simulators. In order to execute this control with the incident fields over the atoms as described in the previous chapter, electrical waveforms and data are used to drive these quantum-classical devices that change the properties of light e.g., frequency, intensity, and phase or the spatial structure. Different technologies have been developed over the years: DDSs, FGs and AWGs that use more fundamental components such as Field Programmable Gate Arrays (FPGAs), Microcontrollers or Application Specific Integrated Circuits (ASICs), Digital-Analog-Converter (DACs) and Analog-Digital-Converters (ADCs),

The hardware control is able to generate electrical signals using hard-wired and hard-coded parameters that define analytic functions sampled from crystal oscillators or RLC circuits or loaded from the RAM memory of the devices. In this section we will introduce the operation of current and new generations of devices that I have installed in the laboratory implementing first tests and later a software control system based on the capabilities and features of these new generation of controllers.

3.1.1 Current technology

The current technology for the generation of electrical waveforms is based in FPGAs and ASICs that are internally programmed with logic circuits that produce binary outputs that are transmitted to DACs, converting them in analog signals that afterwards are filtered and amplified for their final use. However this is limited to the internal configuration of the FPGA and therefore gives a very rigid set of possible outcomes since this component can only be flashed at the beginning of the control sequence since its compilation are in the time scale of the control sequence. Another possibility is to load the bit strings directly from memory in what are called Look-Up-Tables (LUTs), this method would be limited by the time it requires to load the pre-calculated values into memory i.e., memory bandwidth and then be passed to the DACs.

The output signals from these controllers are usually synchronized with an internal oscillator (clock), it is also possible to synchronize it with an external clock in order to keep it synchronized with other devices. Additionally, external triggers can be used for defining the start of the functioning of these devices. Each of these devices are based in a specific architecture and I would like to briefly summarize them here, to be used as a reference for the next section, where I describe the new hardware.

An ultimate goal of these devices is to generate arbitrary waveforms, these are waveforms or signals that are defined by the user or programmer. AWGs architectures are generally described by a block diagram where a clock sampler allows an address generator to sample a value after receiving a trigger, accessing the device memory where a processor has stored waveform values. This sets the values that are send to the DAC in order to apply a filter, amplification and pre-amplifications that will be coming out of the device. See fig. 3.1 for the block diagram.

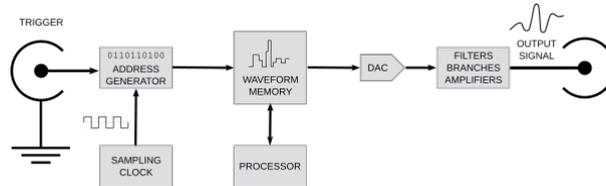


Figure 3.1: **General block diagram for AWGs.** Schematic functioning of AWGs by using a sampling clock and a processor to determine the bit strings passing to the DAC before being prepared by filters and amplifiers for the final output through a terminal.

On the other hand, FGs and DDSs use a phase accumulator block that permits these devices to have a very high precision in the repetition output cycle. See fig. 3.2 for the corresponding block diagram.

This advantage for FGs and DDSs also produces a negative effect of neglecting points from the sampling for large cycle frequency and of repeating points for slow cycle frequencies, not being possible to determine all the sampling points of the waveforms. When FGs are equipped with AWGs capabilities, the changes imply modifications that involve output filters based in linear-phase (Bessel) design and elliptical filters, which are good for non-sharp waveforms. The changes also include the type of memory for the waveforms, changing from ROM memory to RAM memory, being capable to change the number of samples per repetition. Thus, for narrow pulses the phase accumulation in this architec-

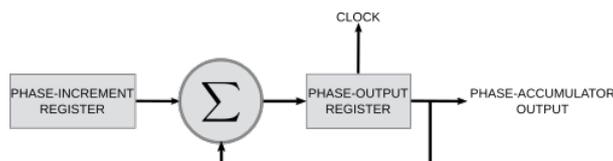


Figure 3.2: **General block diagram for FGs and DDSs.** Schematic functioning of a DDS producing an output signal according to the phase accumulation parametrized by an internal clock.

ture becomes more problematic forcing one to choose specific repetition rates that are integer submultiples of the clock frequency.

Another important aspect of waveform generators is the phase coherence, which links to the presence of discontinuities in non-constant frequency waveforms. In other words, when changing for example the frequency of a sinusoidal signal the phase change according to the number of data points that are skipped by the phase generator and it is sent to the DAC. This explains why DDSs and FGs are good and commonly used for linear frequency sweeps, but not for arbitrary waveforms which might include many non-linear changes of the signal's frequency.

Regarding AWGs, they differ from FGs and DDSs by the possibility to split the memory into multiple segments allowing it to generate and run a segment many times before another one is executed. This in turn allows AWGs to fully synchronize outputs with a guaranteed phase coherency when frequency jumps are present. An advantage of this architecture is that this memory segmentation allows us to branch the possible output signals conditionally, which is an important feature for realizing quantum error correction.

3.1.2 New generation of classical control

Devices such as AWGs and DDSs are widely used in the community and they have been performing well. However, they fall short in fulfilling high level requirements in terms of scalability, precision and speed for generating robust and fast gates close to the error threshold, and they lack the necessary features for realizing fast branching of waveforms in order to achieve quantum error correction.

In this section I will describe a new generation of commercial classical control that I have installed in the laboratory and upon which I have built a software system that enables the exploitation of the properties of this new generation of devices. This new device is a commercial generator from the company Quantum Machines (QM) called Operator X (OPX). See fig. 3.3 for an image of the OPX front and back panel.

The workflow of the OPX consist in programming the instructions with the OPX QUA Python SDK in a computer. The QUA code is executed and the instructions are send via LAN to the System-On-Module (SOM) of the OPX which has a dedicated module to compile the program into low-level language and transmitted to the FPGA pulses processors which are combined with internal oscillators signals. Then these waveforms are converted to analog signals by passing them to DACs when the FPGA is triggered. Inversely this can be done for analog input signals that are passed through ADCs for later demodulation such that it can be used on the computer.

Each of the OPX counts with 10 analog and 10 digital channels with 1GSa/s 16-bits vertical resolution along a 400 MHz frequency bandwidth. The jitter of this device is

$< 1\text{ps}$ and it can be synchronized with external clocks at 10, 100, and 1000 MHz. Rise and fall times of the output signals falls well below current state-of-the-art gate times ($\sim 50\text{ns}$).

Regarding capabilities for real-time features, the phase update latency of the waveform is suppressed to $< 32\text{ns}$ ¹ and the frequency update latency is below 268ns. The feedback latency from the last output value to the branch of the first value of the analog (digital) signal is 198ns (120ns) meanwhile the trigger latency from an external analog (digital) signal to the first value from the terminals of the OPX is 306ns (166ns)².

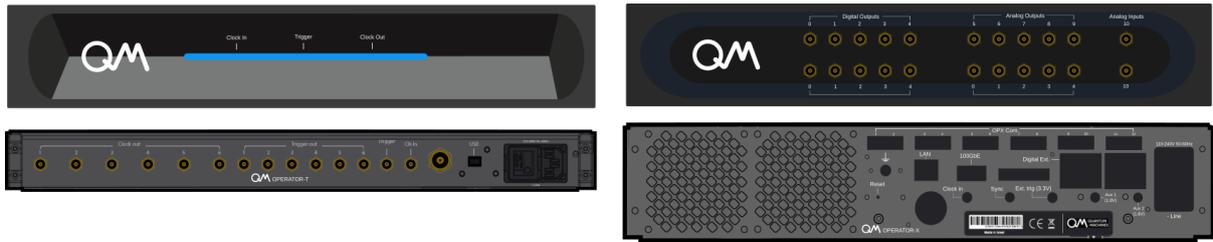


Figure 3.3: **OPX front and back panel.** Front and back panel of the OPT (left) and OPX (right). Images inspired from the documentation of the hardware².

The simplified block diagram of the OPX can be seen in fig. 3.4, where it is possible to note the subtle differences of the architecture with the one presented previously for AWGs, including an additional block “Thread” that feeds the Waveform Generator, showing that it is an architectures based on CPU rather than memory. This also allows the device to use a Data Processing Block to feedback calculations of the waveform in order to realize on-the-fly waveform modifications.

Installation and features

The installation of the OPX consisted in mounting two OPX (master and slave) units with the Operator T (OPT) and connecting it to the LAN. The OPT produces up to 6 clock output signals that were distributed to the 2 OPX. Additionally there is a USB connection from OPT to the master OPX and an auxiliary connection from the OPX master to the OPX slave. The OPX units also have 12 fast interconnections via optical fibers. The connection diagram is shown in fig. 3.5 with the picture of the installation in the laboratory.

An important feature of QUA is the possibility to embed in the QUA program macros which are functions based on QUA semantics but python syntax. Also it is possible to introduce conditionals, loops and definitions of variables (e.g., integers, booleans, fix point number) that will be passed to the low-level language of the FPGA. Additional features are, demodulation of input signals, application of output filters, simulation of the device, external triggering setup, alignment of waveforms and streaming of data to the user computer or server. More details of the implementation of these features are continuously updated in the documentation [Mac].

¹OPX specs sheet, available by request in <https://www.quantum-machines.co/see-full-opx-spec-sheet/>

²Documentation of the OPX hardware available in https://docs.quantum-machines.co/1.1.7/qm-qua-sdk/docs/Hardware/OPX_hardware/.

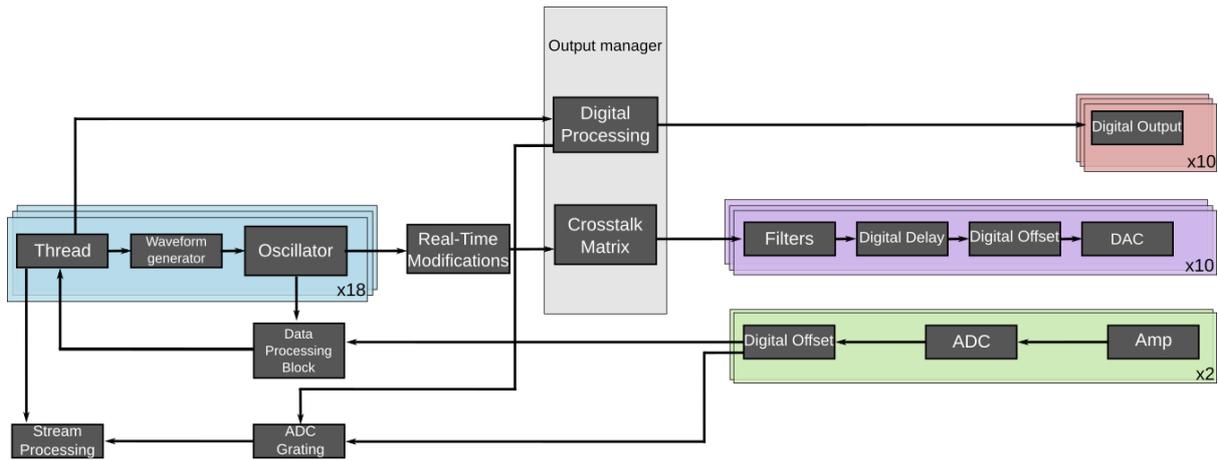


Figure 3.4: **OPX block diagram.** Schematic of the workflow in the OPX hardware. Signal generated by 18 oscillators and mixed with AWGs is streamed internally to the output manager that transmit the signal to digital and analog outputs by passing through modules for real-time time modifications, allowing the change of the waveforms in the flight and modifications according to internal noise between channels with the cross-talk matrix and later on passing to filters and offset modules. Images are inspired from the documentation of the hardware².

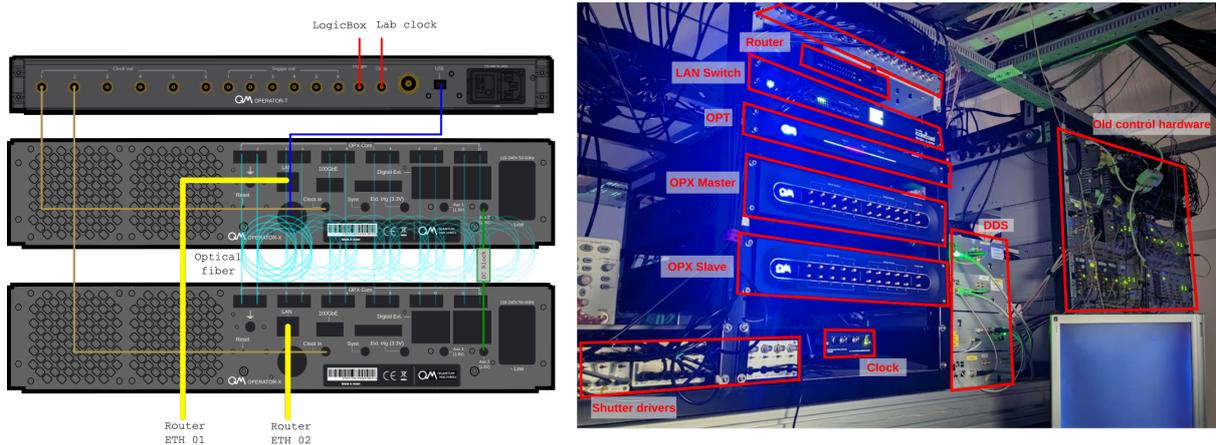
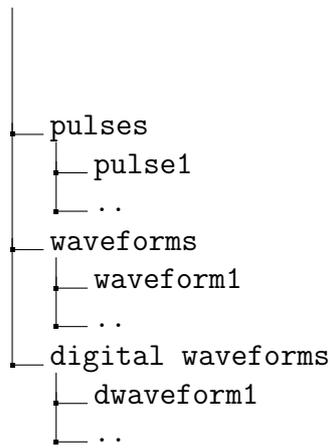


Figure 3.5: **New control system installation** Connection map between 2 OPX units and the OPT and the lab integration with other components.

Hardware configuration

A fundamental element in preparing the unit for use is the setup of the configuration of the hardware which is set in the same python script where the QUA program is scripted. In the script it is defined as a dictionary with the following structure:

```
configuration/
├── controllers
│   ├── con1
│   └── con2
├── elements
│   └── element1
└── ..
```



The time for initializing the OPX, setting the configuration and generating the program is in the order of hundreds of microseconds and writing the program in the device's board takes in the order of hundreds of milliseconds while executing the simulation takes a couple of seconds, but executing the job takes hundreds of milliseconds. The job execution shows a slightly faster performed time than the simulation timing i.e., $\sim 2s$ compared to $\sim 800ms$, with a better convergence in the executing time by realizing 100 simulations and 100 executed jobs with 3 channels playing 3 digital constant pulses and analog stroboscopic pulse with 100 cycles. This can be appreciated in fig. 3.6.

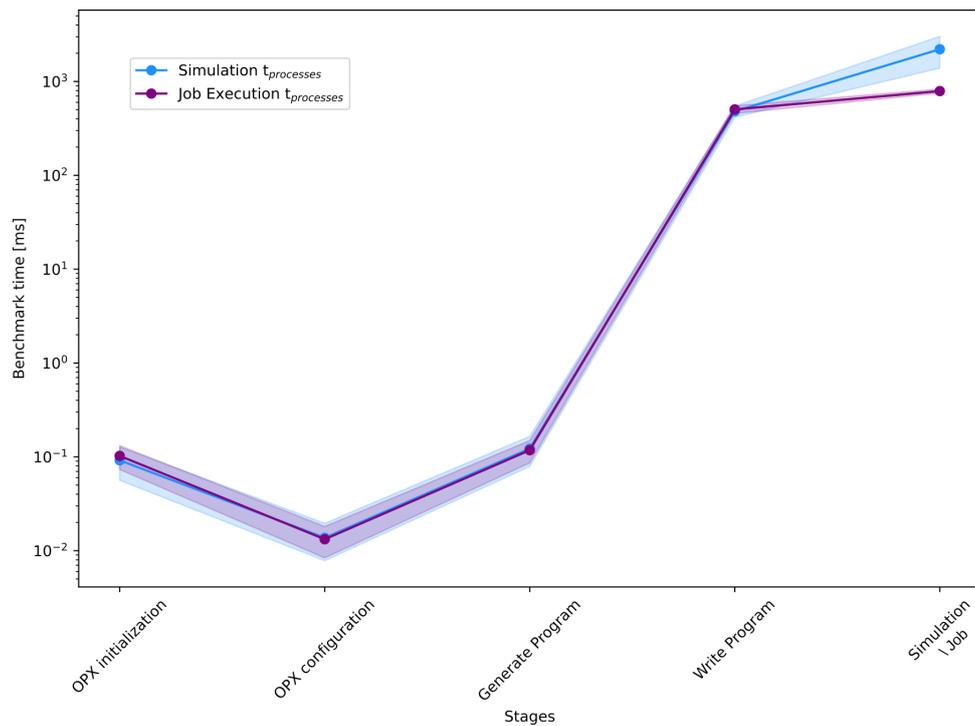


Figure 3.6: **OPX performance.** Average time over 100 repetitions of the different stages when programming a OPX unit.

Waveform generation

For programming the OPX using the python SDK in order to generate the electrical waveforms the workflow of the python script:

Algorithm 1 Main Program

Import QUA and Python libraries.

Definition macros, variables and Python objects.

Definition configuration Python dictionary.

Initialize the OPX QM Python class by giving the IP address.

Set configuration of the OPX instance by passing the configuration.

Initialize the OPX Manager.

Connect to OPX.

Write the commands to the program.

Procedure: Simulation [simulation time].

Execute simulation by passing the simulation time duration.

Output plot the result.

End

Procedure: Job.

Execute the job.

End

3.2 Software control system: AQiPT

The Atomic Quantum Information Processing Toolbox or AQiPT developed during this thesis is a modular monolithic open-source software system designed to serve as a universal framework for implementing experimental sequences in atomic platforms. AQiPT is used for designing and effectively compiling composed waveforms, configurations and instructions while accommodating the unique setup features of different classical hardware. It's a modular architecture, based in specifications and scripting programming, incorporates an Application-Programming-Interface (API) to minimize the semantics and dependency from the devices' commands. What sets AQiPT apart is its seamless integration of quantum information processing tasks and applications with folded processes (see fig. 3.7). This inclusive approach facilitates cross-talk between fundamental atomic theoretical simulations and embedded formalism, allowing for the creation of experimental sequences centered around either waveforms or instructions.

AQiPT allows users to craft personalized experimental sequences tailored to the specific design of their systems. It also tightly integrates with theoretical simulations and formal methods in creating experimental protocols for sequence design enhances analytical capabilities and supports the development of new experiments. The platform's pythonic nature ensures compatibility with a wide spectrum of software packages, leveraging Python APIs/SDKs referenced in the specifications. This adaptability fosters smooth integration with various software tools, amplifying AQiPT's versatility.

Considering the large scope of the software system, what is presented in this section refers to the general design and the fundamental pieces coded during this work. Continuous development of the software is carried out under the umbrella of the EuRyQa project and associated institutions [EuR] and the AQiPT open-software contributors.

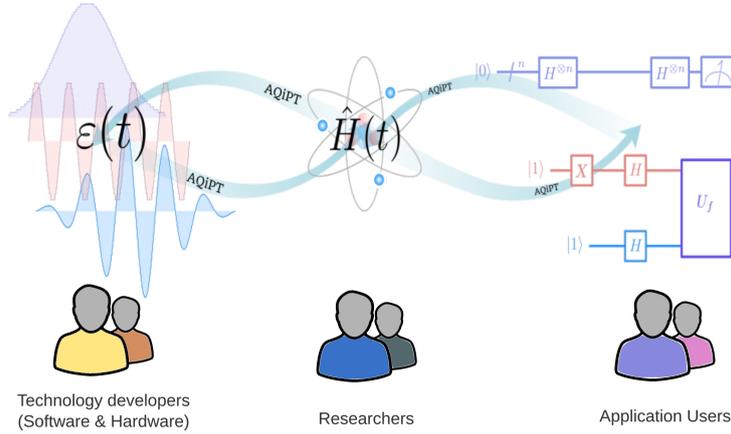


Figure 3.7: **AQiPT**. The software brings tools that bridges different layers of the quantum stack between the applications with quantum circuits (right) to machine instructions (left) using the experimental and atomic physics as a connection.

3.2.1 System architecture

An initial version I developed for AQiPT contains the main pillars for the software system that connects each of the three layers of the full-stack model presented before. The main elements for this interconnection are distributed in different modules as Python classes that we denoted: `Director` (`Control` module), `Producer` (`Emulator` module), `Data Manager` (`Analysis` module), `Inspector` (`DAQ` module) and the `API` (`Interface` module). These modules are depicted in fig. 3.8.

Considering the part of the workflow that will be explained in detail in the next section, these elements gather all the information contained in the specifications and drivers in order to abstract into Python all the available resources and pass it to the compiler, set simplified models for simulation, handle and analyze the meaningful data, share relevant quantities as well as offer a gateway to interact with the stack.

Additionally, this architecture includes a depository with relevant files and data. The code documentation and learning material has been generated in order to guide the user on how to create different instances and setup tasks with the different components and modules of the software system.

Specifications

As previously discussed, the quantum computer or simulator’s specifications play a crucial role in controlling and managing both classical and quantum resources. The specifications are an abstraction of the machine or setup, including the description and properties of the relevant devices i.e., active, passive and idle. After being abstracted, they are organized into three “containers,” referred to as “specifications”: atom, hardware and software.

Hardware specifications are used to abstract the components, instruments and devices within the quantum computer or simulator in order to assign the the corresponding physical channels of the controlling devices as well as for setting default and initial values to the concerning elements. It is used to associate the instance of the drivers loaded in the AQiPT driver libraries to the controlled devices (see more details in fig. 3.10). Additionally, it is used for generating the theoretical model to calculate quantities as Rabi couplings, Zeeman splitting from the magnetic fields generated by coils among others. This

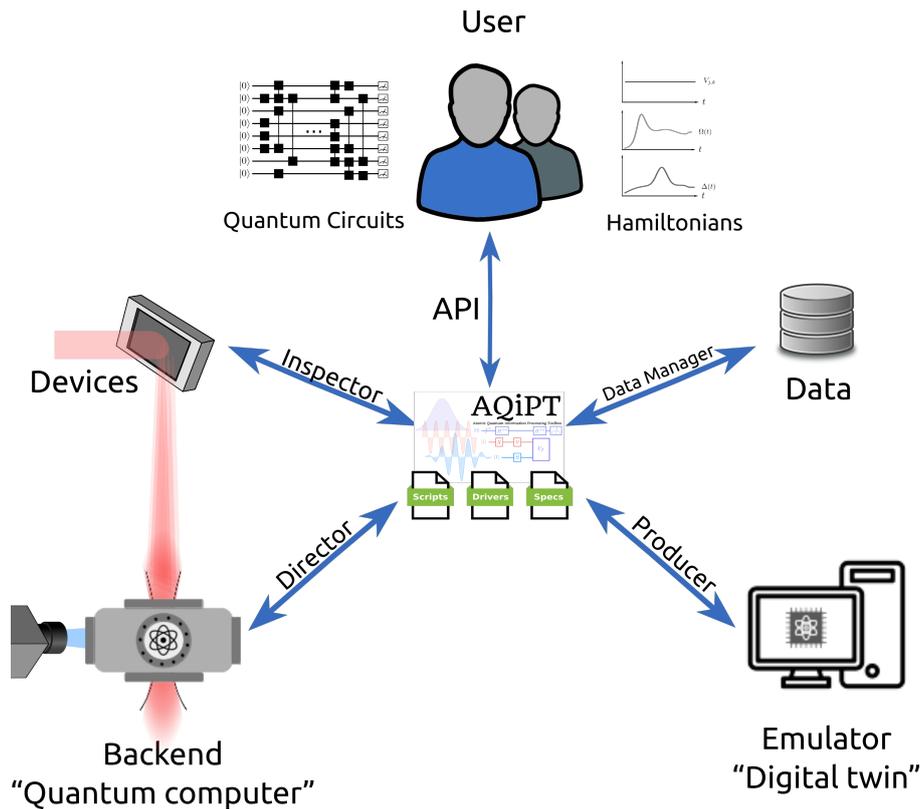


Figure 3.8: **AQiPT architecture** General architecture of AQiPT showing the software inter-connectivity of the modules and classes with the necessary functionalities. Starting from based components such as: scripts, drivers and specifications. A **director** class interacting with the backend components, a **producer** class managing the related simulation models, an **inspector** that monitor and set configurations of the devices and a **data manager** for handling the data. Also an **API** that serves as an interface between the User and the software AQiPT.

specification comprises two blocks: the first block, “Metadata,” contains general information such as date, time, author(s), and affiliation(s). In contrast, the main block, “Setup,” encompasses fundamental details, including driver pointers, properties, and calibration pointers for various components in the atomic platform implementation. These components consist of lasers, Tapered-Amplifiers (TAs), Acousto-Optical-Modulators (AOMs), Acousto-Optical-Deflectors (AODs), cameras, Arbitrary-Waveform-Generators (AWGs), Direct-Digital-Synthesizers (DDSs), Digital-Micromirror-Devices (DMDs), Spatial-Light-Modulators (SLMs), IQ-mixers, shutters, and more.

```

1      {"AOM": "AOM_example",
2        "model": "AOMO_3110_120",
3        "ID": "0x002",
4        "EXT_CHANNEL": "0x33452",
5        "RF_frequency": 98.486,
6        "RF_frequency_UNIT": "MHz",
7        "RF_Amplitude": 0.0,
8        "RF_Amplitude_UNIT": "V",
9        "RF_phase": 0,

```


might be considered in the Hamiltonian terms. It can also be used for calibration with the backend as comparison of the results from the QPU and the expected from the model. This calibration consist of three blocks (see fig. 3.9). The initial block, “Metadata” holds general information such as date, time, author(s), and affiliation(s). The second block comprises details about the pertinent parameters of the atom species, the states of interest, and additional parameters related to atomic and light-matter interactions. These are stored in the third block, which contains the calculated values for each ij -state, including energy, transition frequency, dipole matrix element, etc. Further explanations and details can be found in the `kernel` module.

The hardware and software specifications are set in two JSON files which is a standard file format and data interchange format with human-readable text, meant to contain objects consisting of attribute-value pairs and arrays. The atom configuration is stored in an Apache Parquet file format. This format is column-oriented, offering efficient data compression and enhanced performance, making it well-suited for handling large volumes of complex data in bulk.

Workflow

Due to the hybrid experimental-theoretical nature of the software system, the workflow closely resembles an operating system. It involves not only the generation and calculation of an accurate model of the system but also includes overseeing and managing the experiment. This includes providing tools for storing, monitoring and analyzing data from the device.

The workflow initiates with Software Specifications (`SW Specs.`), which can be created either from a JSON file or through the AQiPT toolbox control module (i.e., `control` module). Subsequently, AQiPT instances are generated and loaded. These instances are then employed to produce `pulses`, `tracks`, or `sequences`, which are in turn loaded into a new specification file. This process is facilitated by the `WTYPE/FTYPE - cmd mapper`, where `WTYPE` pertains to waveform type, and `FTYPE` corresponds to file type. The relevant information about command arguments (`cmd-arg`) is provided to the Director.

The Hardware Specifications (`HW Specs.`) can be loaded from the JSON file. After abstracting with AQiPT components, the hardware fetches the driver commands associated with it. These commands are then provided to the Director.

The `Director`, in turn, aligns the commands with their respective command arguments while compiling the schedule defining the `Quantum Assembly Program (QAP)`. Subsequently, the `QAP` is executed within the experiment instance, directly influencing the implemented setup.

Both the `HW` and `SW Specs.` are stored in the `DataManager` alongside data from passive actors and devices, such as acquired images from cameras. Simultaneously, the theoretical model generated by the `Producer`, using information from the `Director` or the specification, stores data within the `Data Manager` of the atomic specifications, atomic model, and atomic quantum register instances. The `Producer` also retains information about the quantum computing software in case the user opts for the higher gateway with quantum circuits.

Ultimately, the data stored in the `DataManager` is accessible through the `Inspector`. This allows the generation of data instances for post-analysis using any arbitrary terminal with AQiPT modules or from disk using the `Analysis` and `DAQ` modules.

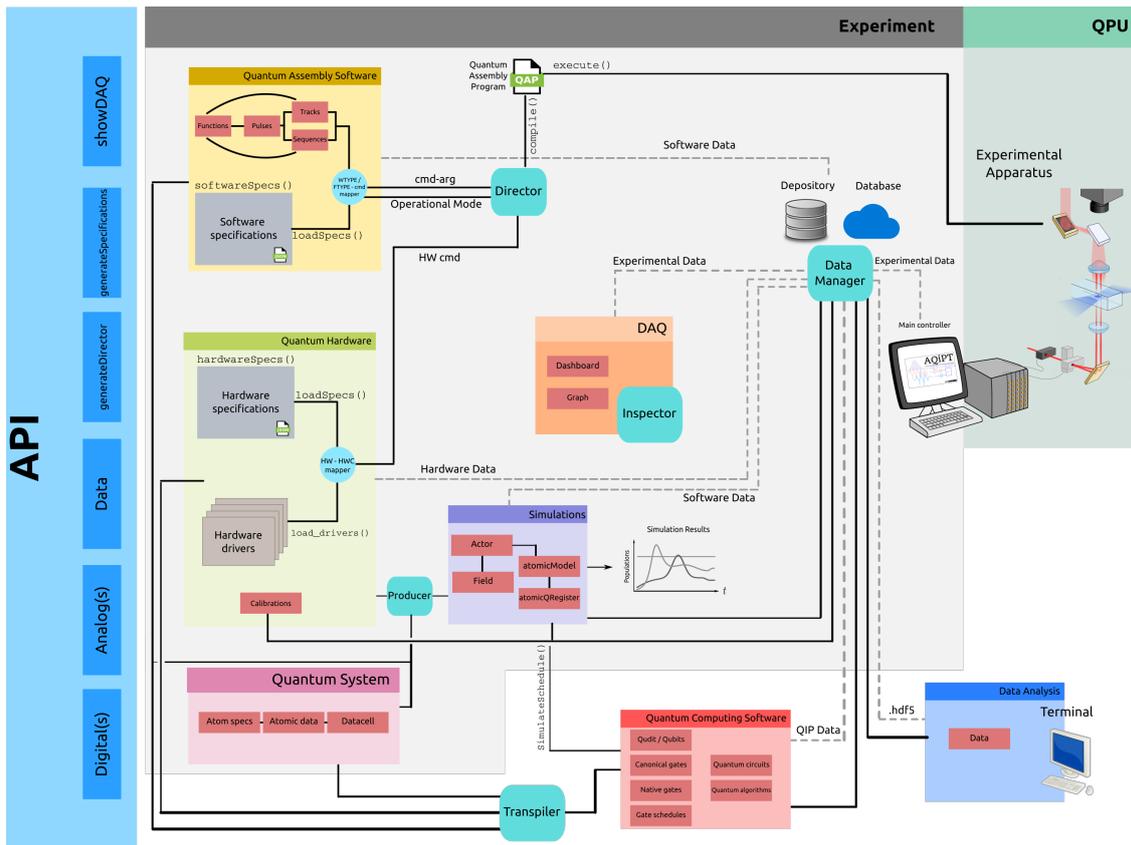


Figure 3.10: **AQiPT workflow.** Block diagram of the functioning of the software for executing tasks in the quantum device as well as generating simulations from the quantum computing level, as from the Hamiltonian or backend level. Initial commands use the minimal set of the API instructions to generate the hardware and software abstractions or they can be load from the specification file to generate the command and arguments sets that compose the **Quantum Assembly Program (QAP)** that execute the instructions in the backend. Similarly, it can generate a basic model for simulating the physics of such instructions or to generate the **QAP** from or dynamic simulation from high-level quantum programs i.e., quantum circuits. Data coming from the experiment as well as from simulations is saved in the depository or database locally or in cloud storage, accessible by the terminal computer.

3.2.2 Modules

A precise control exerted over atomic systems requires a scalable and modular software for programming classical hardware. The software I am presenting in this section has the capability to encompass a large quantity and complexity of analog and digital channels for driving elements such as AOMs and EOM, the instructions can be recycle as well as be managed in a modular way, such that becomes easier to replace, merge and extract elements of the pulse sequence. Although new technologies and protocols are alleviating these challenges, a high-level of control is paramount. Below, I provide an overview of classes and commands that function as tools within AQiPT modules, acting as components of the different layers along the stack (chapter 2) presenting experimental awareness. This design aims to offer a modular, flexible and extensible solution.

The seamless integration with hardware enables the development of abstract processes

for atomic systems, such as cooling, trapping, atom rearrangement, state preparation, and gate operations, through an easily accessible library. The scheme below illustrates the modules and classes contained within AQiPT.

In this initial version of AQiPT, there are 8 internal modules and 3 significant directory structures. The main sub-structures are: the compiler (e.g., for quantum circuits, quantum gates, waveforms, and programs), depository (for data), and hardware (for real and virtual drivers). The 8 main modules are: Analysis, Control, DAQ, Datalogger, Directory, Emulator, Interface and Kernel, these modules will be discussed in the following.

```

1 from AQiPT import AQiPTcore as aqipt
2 from AQiPT.modules.kernel import AQiPTkernel as kernel
3
4 from AQiPT.depository.APIs.API import *
5
6 from AQiPT.modules.analysis import AQiPTanalysis as analysis
7 from AQiPT.modules.control import AQiPTcontrol as control
8 from AQiPT.modules.daq import AQiPTdaq as daq
9 from AQiPT.modules.emulator import AQiPTemulator as emulator
10
11 from AQiPT.modules.datalogger import AQiPTdatalogger as datalogger
12 from AQiPT.modules.directory import AQiPTdirectory as dirPath
13 from AQiPT.modules.interface import AQiPTinterface as interface
14
15 from AQiPT.compiler import AQiPTcompiler as compiler

```

Listing 3.2: Importing AQiPT modules.

A. Analysis (data analysis)

This module is designed to serve as a toolkit wrapper for the analysis of diverse data instances, encompassing Image Data types, Table Data, Text Data, Array Data, and AQiPT Data, all managed by the Data Manager. Within each data-type class, functions for various purposes, such as fitting, FFT, operations, averaging, etc., are embedded. These functions are adaptable, allowing modifications based on specific data analysis requirements.

The AQiPT data structure is summarized in fig. 3.11.

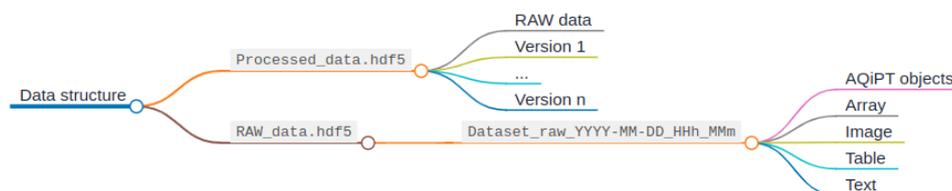


Figure 3.11: **AQiPT data structure.** Tree diagram of the data structure stored by the Data Manager in optimized HDF5 files, containing the raw data and the versions of possible data analysis that the user decide to store.

The **Data Manager** plays a crucial role in storing raw data originating from acquisition devices, waveforms, variables, AQiPT objects, and additional metadata. This data is stored in an HDF5 file, organized within a named time-stamp group with

five sub-groups (each corresponding to a specific data type). This organizational structure facilitates easy access to the data using Python and a file explorer-like interface compatible with open-source HDF5 software. The storage process's latency is not a bottleneck during the actual experiment, as it is stored asynchronously. Data is first instantiated as Python structures and subsequently passed to the data file. In the case of feedback data, there is no requirement for it to be stored before utilization for such purposes.

Furthermore, the `DataManager` generates an additional empty copy of the HDF5 file containing raw data. This feature enables the versioning of analyzed data through different steps, providing the flexibility to revert to any stage of the data processing during the analysis.

This module is closely linked to the `DAQ` module, as our software offers the option to utilize data either in memory (via the `Inspector`) or from stored data on disk (HDF5 file).

```

1 dmanager = analysis.DataManager(aqipt.directory.data_depository_dir
2   , 'Andor testing')
3 camera_params={'FanMode': 2,
4               'AcquisitionMode': 3,
5               'TriggerMode': 0,
6               ...}
7 ixon897 = drivers._daq_dev.andor();
8 ixon897.Configure(args=camera_params);
9
10 image= [];
11 ixon897.StartAcquisition();
12 time.sleep(0.4) ;
13 ixon897.GetAcquiredData(image);
14 image= np.array(image);
15 image= image.reshape((ixon897.width, ixon897.height));
16 ixon897.ShutDown();
17
18 andor_dm.add_RAW_data(new_data = image,
19                       group_label = "Dataset_raw_2023-05-30_15h09m"
20                       ,
21                       subgroup_label = 'Image',
22                       data_label = 'Raw image 1');

```

Listing 3.3: **Example of iXon 897 camera.** Images are stored using the `DataManager` class into HDF5 file.

B. `Control` (sequence formalism)

The `Control` module serves as the core of experimental awareness based on waveforms, facilitating the generation of instructions through the formalism for experimental sequences, drivers, and experimental control software.

Within this module, classes for various building blocks of waveforms, including: `function`, `pulse`, `track`, and `sequence`, are situated. Additionally, the module incorporates classes for experiment and `director`. As outlined in preceding sections, the primary functions of the first two classes are to collect information about experimental waveforms, load the experimental setup, and compile the waveforms into the channels of the hardware, considering the awareness of the setup

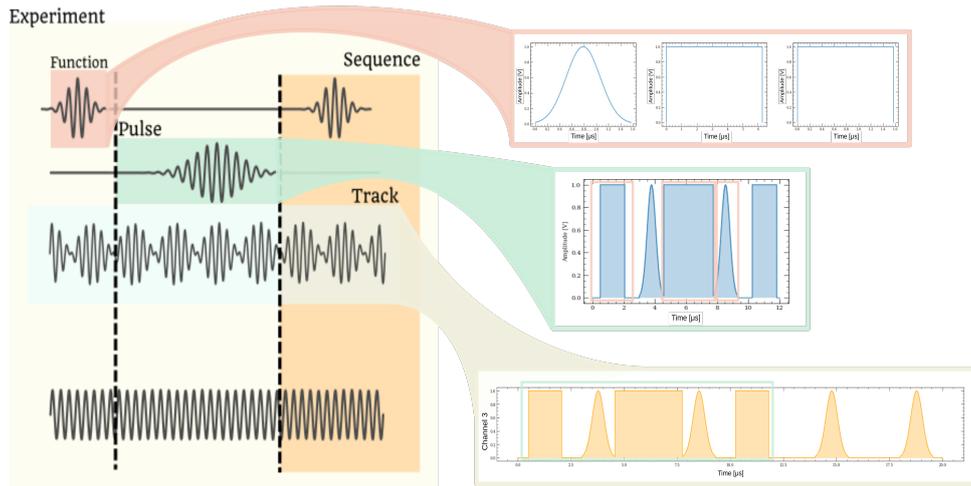


Figure 3.12: **Control module of AQiPT.** **Function** instances (red shadow) that can be reused for the construction of **Pulse** instances defined in a pulse-time window (green shadow). **Tracks** are defined for all time domain of a specific channel (blue shadow) build from stack of pulses. **Sequence** instances (yellow shadow) vertical stack of pulses.

itself. An additional class from the **Emulator** module, known as the **producer**, can be passed to the experiment class. This **producer** class acts as a bridge between the middleware and the backend layer, reducing the disparity between the apparatus and the associated instructions for the quantum system, resembling operations described by the Hamiltonian. Producers play a fundamental role in mapping waveforms and instructions from the experimental implementation to simulations based on simple but representative theoretical models created with the **emulator** module.

Mathematical functions which are included for the waveform generation as part of the **function** class are: **step**, **ramp**, **parabolas**, **sinusoidal**, **chirp**, **chirplet**, **gaussian chirplet**, **gaussian**, **triangular**, **sawtooth**, **quadrature**, and interpolated **splines**. In addition to these basic functions, this class also incorporates a series of mathematical class-methods for setting the area under the curve and applying filter functions. By utilizing a set of function instances, it becomes possible to create a pulse instance that compiles the set of functions based on a list of time-flags relative to the pulse time. The arrays of float values are generated and they can be digitized or not to a given bit-depth in order to be aware of the resolution of signal generators. The additional class-method functions to serve general editing purposes for combining, summing, merging, removing or overwriting.

Using instances of pulses, it is possible to generate instances of tracks or sequences. In the case of sequences, a set of pulses or functions (or a combination of both) can be used. However, in both scenarios, a list of time-flags is not necessary, as the tracks are constructed sequentially. Each pulse is designed and compiled one after the other. For sequences, the vertical blocks maintain a rigid block timing using the longest time-base of the function or pulse in the stack, preventing cross-time-reference between waveforms. Each sequence block is played after the other in

the experiment instance schedule. In track mode, it can be considered as a single block sequence with a time-base equal to the full experimental running time (see fig. 3.12).

The `experiment` instance gathers all the necessary information to execute experimental runs or processes. It contains the functionality to assemble software commands' arguments with hardware commands, generating a schedule that will be compiled into the devices just before execution. This enhances operational speed and is executed via the Main Controller. The `director` class instance, given to the `experiment` instance, plays a crucial role in this process. It contains in memory all the HW & SW Specs. with an embedded mapper, including pointers to on-memory driver instances.

While executing and generating experimental sequences, it is also possible to extract an approximated theoretical model of the system. The producer takes information from the director or the specifications instances and generates a time-dependent model using the actors' instance from the kernel module. Ultimately, it retrieves a model from the emulator module.

In the `Control` module, it is crucial to establish ab-initio specifications, encompassing parameters such as sampling rate, bit depth, and the dynamic time of functions employed in the waveforms. These specifications can be articulated either as instructions in the SW specs file or as a Python dictionary, as illustrated below (see listing 3.4):

```

1#dictionary of general hardware parameters
2hardware_params = {'sampling':int(1e3), #1kS/s
3                  'bitdepth':16, #16-bit
4                  'time_dyn':1}; #time of sampling
5#get params in AQiPT object from dictionary
6dynamic_params = aqipt.general_params(hardware_params);
7tbase = dynamic_params.timebase(); #obtain time-base from params
   object
8

```

Listing 3.4: Use case gate transpilation at Frontend layer.

The instantiating of the function requires a Python dictionary with the parameters of the function (see listing 3.5):

```

1funcs_lst=[]; #arrays of functions to store
2
3#specifications of AQiPT functions
4function_args_lst = [{'amp': 1, 't_o':0.5*np.pi/2, 'width': 0.5*np.
   pi/2,
5                      'tp_window': np.pi/2,
6                      'name': 'pi/2-step', 'color': color_lst[0],
7                      'type': None},
8                      {'amp': 1, 't_o': 0.5*2*np.pi, 'width': 0.5*2*
   np.pi,
9                      'tp_window': 2*np.pi,
10                     'name': '2pi-step', 'color': color_lst[2],
11                     'type': None},
12                     {'g_Amp':1, 'g_center': 0.5*np.pi/2, 'g_std'
   :0.5*np.pi/2*1/4,
13                     'tp_window': np.pi/2,

```

```

14         'name': 'pi/2-gaussian', 'color': color_lst
    [3],
15         'type': None},
16     {'Amp':1, 'freq': 3*np.pi/2, 'phase':0.5*np.pi
    /2*1/4,
17         'tp_window': np.pi/2,
18         'name': 'sine', 'color': color_lst[3],
19         'type': None}];
20

```

Listing 3.5: Use case gate transpilation at Frontend layer.

C. DAQ (data management)

The measurement of the system is a critical aspect, and the type of acquisition depends on the instruments used in the experiment. Consequently, a set of general-purpose functionalities for image and signal acquisition has been consolidated in this module.

In a laboratory implementation using scripting languages, a Graphical User Interface (GUI) is not inherently required. However, it proves beneficial to monitor real-time images from fast cameras and visualize other programmed elements, such as waveforms and image patterns for devices like DMDs and SLMs.

To achieve user-friendly monitoring, the DAQ module integrates classes for Dashboards that contain and display such data through browser interfaces using plots from `Plotly` in a locally running server of `Dash`. This functionality is encapsulated by an instance of the `Inspector` class. Additionally, it is possible to generate instances of graph classes to plot 3D surfaces, 2D color maps and 1D plots.

D. Datalogger

The `Datalogger` module is ultimately tasked with recording all transmitted instructions and managing data processes. It facilitates the recovery of former states and configurations throughout the software.

E. Directory

This module handles the folder directory structure of the modules, specifications, data, and other resources of AQiPT. It also manages the path directories to drivers, which is crucial for the proper functioning of AQiPT's parsers i.e., matching memory addresses or IDs to the physical devices and channels.

```

1 directory = dirPath.directory_tree({'path': '~/AQiPT/modules/
    directory/', 'filename': 'directories_ubuntu.json', 'printON':
    False})

```

Listing 3.6: Directory module. Example for generating an instance of the file tree.

Accompanying this module is a JSON file containing the entire directory structure of AQiPT, formatted for both Ubuntu and Windows operating systems. When the directory tree is instantiated, these path directories to folders are loaded and subsequently utilized in other modules for accessing the different folders and files required.

F. Emulator (Digital twin)

The `Emulator` module encompasses seven main classes dedicated to bridging theory and experiments, as discussed in previous sections. The `atomicModel` class plays a key role in creating a single-particle Hamiltonian model based on the atomic physics of the system, with the possibility to use the data from the kernel main function. An example of the use of this class can be found in the code from listing 3.7.

```
1 #rrQubit
2 psi0 = 0; #initial state
3
4 #nr of levels in the quantum bi. Nrlevels = 2 for qubits, Nrlevels
   = d for qudits
5 Nrlevels = 2;
6
7 #couplings list within the states [i,j], couplingValue, pulse
8 couplings = {'Coupling0': [[0,1], 2*np.pi*0, None ]};
9
10 #detuning list in the state i as [i,i], detuningValue, pulse
11 detunings = {'Detuning0': [[1,1], 0, None]};
12
13 #dissipator list within the states [i,j], dissipatorStrenght
14 dissipators = {'Dissipator0': [[0,1], 0]};
15
16 #label of Rydberg states in the atomicModel (optional)
17 Rydbergstates = {'RydbergStates': [0,1], 'l_values':[0,1]};
18
19 #wrapper dictionary for the dynamic params
20 rrQubit_sim_params = {'couplings': couplings,
21                       'detunings': detunings,
22                       'dissipators': dissipators,
23                       'rydbergstates': Rydbergstates};
24
25 #model class of AQiPT model(timeOfdynamics, NumberOfLevels,
   InitialState, DynamicParams, NameLabel)
26 rrQubit = emulator.atomicModel(tbase, Nrlevels, psi0,
   rrQubit_sim_params, name = 'rr-qubit');
27
28 #Graph-Map of the atomicModel
29 rrQubit_Map = rrQubit.modelMap(plotON=True, figure_size=(3,3));
30
31 #building Time-indepnedent Hamiltonian
32 rrQubit.buildHamiltonian();
33
34 #building Lindbladians
35 rrQubit.buildLindbladians();
36
37 #building Observables
38 rrQubit.buildObservables();
39
40 #playing simulation. mode='free' for time-independent | mode='
   control' for time-dependent
41 rrQubit.playSim(mode='free');
42
43 #ploting results
44 rrQubit.showResults(figure_size=(18,5), resultTitle='Free dynamics:
   '+rrQubit._name);
```

Listing 3.7: Emulator module. Generating a rr-qubit instance using the `atomicModel` class.

This class serves as a building block for the `atomicQRegister`, which is responsible for creating registers with collections of qubits, considering all available mechanisms between them, such as driving terms and interactions.

```

1 coordinates = [(0,0,0),
2               (0,20,0),
3               (0,40,0)];
4 qpu = emulator.atomicQRegister(physicalRegisters=[rrQubit,
5                                                  rrQubit,
6                                                  rrQubit],
7                                   initState='100',
8                                   name='3 rr-qubits quantum register',
9                                   connectivity=['All', []],
10                                  layout=coordinates);
11 qpu.buildNHamiltonian();
12 qpu.registerMap(plotON=True, figure_size=(3,3));
13
14 qpu.compile(plotON=True);
15
16 qpu.buildInteractions(c6=-2*np.pi*17950, c3=2*np.pi*7950);
17 qpu.buildNLindbladians();
18
19 pop_pss = qt.basis(2**3, 4).proj();
20 pop_sps = qt.basis(2**3, 2).proj();
21 pop_ssp = qt.basis(2**3, 1).proj();
22
23 qpu.buildNObservables(observables=[pop_ssp]);
24 qpu.buildNinitState();
25
26 qpu.playSim(mode='free');
27
28 qpu.showResults(resultTitle='QPU Dynamics. PRL 114, 113 002 (2015)',
29                , figure_size=(10,6));
30 plt.legend(labels=[ r'|ssp$\rangle$'])
31 plt.xlim(0,6)

```

Listing 3.8: `Emulator` module. Generating a QPU instance using 3 rr-qubits: result reproduction from Barredo et al PRL 114, 113002 (2015).

In fig. 3.13 I demonstrate the use of the `atomicModel` class of AQiPT to reproduce the dipole-dipole interactions in an array of rr-qubits i.e., `atomicQRegister`.

This module exhibit a set of features that are not part of QuTiP [Joh12] and that bring advantage when developing complex simulations. For instance one of the features in this module is that the `atomicModel` class enable the creation of an arbitrary n -level system i.e., Hamiltonian by just introducing the atom-light couplings between states with their time dependency and simplifying the user the job of manually define an eigenbasis for each model, similarly for the Lindbladians. It do also have the possibility to define some of the involved states as Rydberg states with special properties for when integrated in an atomic register. Moreover, this

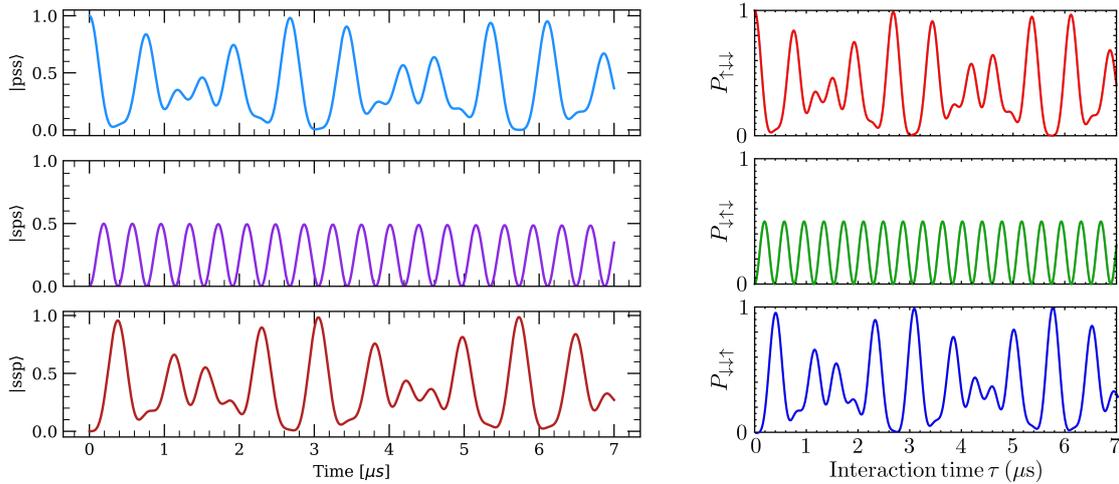


Figure 3.13: **Emulator module of AQiPT**. Results from the simulation of an QPU using 3 rr-qubits from the emulator module (left), reproducing the results from Barredo et al [Bar15] (right).

class produce a graph of the model that allows to depict the model in a graphical way. The plotting of the dynamics from default observables like the populations of the states or in the case of a two-level system in the Bloch sphere are also included features. All these features allows to automatize simulation of scans by changing the model parameters and recompiling the Hamiltonian without generating all the parameter all together for each scan run.

Additionally, in this module it is possible to describe the physics occurring in the actual optical setup with additional specific classes:

`optElement` : for optical elements.

`beam` : for the modification of optical beams.

`OptSetup` : for the general optical setup, encompassing beams and optical components.

Instances of these classes aim to serve as tools for articulating the hardware setup based on specifications and subsequently develop an accurate yet simple model describing the meaningful physics.

As highlighted in previous modules, the `Producer` class is defined to generate a set of instruments that define the different Hamiltonian terms of single-atom models. These models are instances of the `Actor` class, which contains the role and all the information of the elements of the setup. Additionally, it includes the time waveform schedule and atomic specifications. This class further instantiates a `Field` element containing information about all physical parameters for semi-classical simulation if required. Actors can be instantiated as `optical`, `acoustic`, `magnetic`, `microwave`, `RF`, `electric`, or `gravity` fields.

```
1 omega = 2*np.pi*5;
2 transitions = [[[0,14],[1,13],[2,12]]];
3 MyProducer = emulator.producer(name='My producer',
```

```

4         atom_specs=atom_specs,
5         HW_specs=hardware_specs,
6         SW_specs=software_specs);
7 eqm_producer.addActors(actorName='My Actor',
8         deviceLst=['Laser', 'AOM', 'Shutter', 'AWG'
9         ],
10        roleDict={'args':transitions, omega,
11        MySequence._Stack[0].digiWaveform *MySequence._Stack[1].
12        digiWaveform}));

```

Listing 3.9: `Emulator` module. Example of Producer and actors.

G. Kernel (resource management)

The `kernel` of AQiPT encapsulates all the pertinent information about the atomic element, which can be utilized for simulations describing the qubit/qudit system with an atomic model and/or atomic Quantum register. The details and calculations regarding the states and properties of interest of the atoms are encapsulated in a data class named `atomicSpecs`. This data class comprises metadata and another data class called `atomicData`, which contains blocks of data classes or data cells holding the values of the calculations for all ij -states combinations, as depicted in fig. 3.14.

```

1 atom_specs_params = {'4RabiFreq' : [hardware_specs.hardwareLST['
2     laser'] [0], hardware_specs.hardwareLST['laser'] [1]],
3     '4SMap': {'nrange': [50,53],
4     'lmax': 25,
5     'bz': 0.1,
6     'erange': [0, 2.e3],
7     'n': 600,
8     'progressbar': True,
9     'unit': 1,
10    'highlightstate': True,
11    'highlightcolour': 'red'},
12    '4BBR': {'dn': 30,
13    'lj_vals': [[0, 0.5], [1, 0.5]],
14    'mintemp': 0.1,
15    'maxtemp': 300},
16    '4C6': {'phi': 0,
17    'theta': np.linspace(0,2*np.pi,30),
18    'dn': 5,
19    'dEmax': 25e9,
20    'ploton': True} }; #extra
21
22 parameters
23
24 atom_specs = kernel.atomSpecs(atom=Potassium39(),
25     states=['4S_1/2, 1/2', '4P_1/2, 1/2'
26     ],
27     metadata={'Date and Hour': time.ctime
28     (),
29     'User/author': 'M. Morgado'
30     , 'Affiliation': 'CESQ', 'Project': 'AQiPT',
31     'Atom': 'Potassium39', '
32     States': str(['4S_1/2, 1/2', '4P_1/2, 1/2']),
33     'Comments': 'EQM RydSim 39K
34     .' },
35     extraParams=atom_specs_params);

```

```
28 atom_specs.set_atomicData(bfield_ON=True);
```

Listing 3.10: `Kernel` module. Generating atomic specification for the considered states of potassium-39 (4S and 4P states).

Here it is also possible to have a list of *Variables* that can be used along the script as an AQiPT instance:

```
1 VAR = kernel.variables(name='My_experiment');
2 VAR.load_fromSpecs(path='~/AQiPT/compiler');
```

Listing 3.11: `Kernel` module. Generating instance for variables.

Furthermore, the hardware and software specifications classes are also located in this module. An instance of the hardware specifications serves as a parser for a JSON file containing all the requisite information about the atomic physics setup. This includes double- and single-pass configurations, electronic mixers, lasers, and any additional devices. This information can then be unwrapped in an experiment instance along with the waveforms. Additionally, in this module, it is possible to have a list of Variables that can be utilized throughout the script as an AQiPT instance.

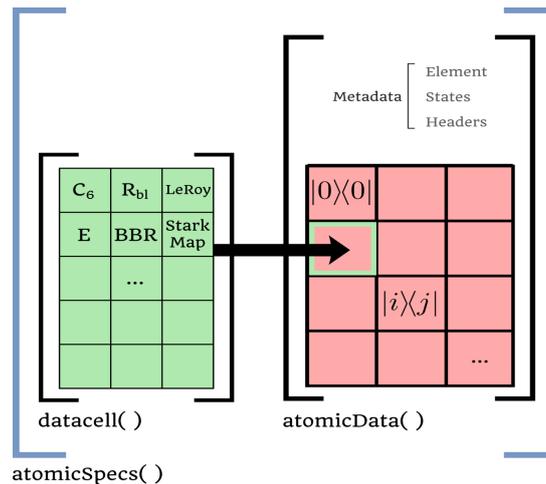


Figure 3.14: **AQiPT Atom specifications from `Kernel` module.** Dataclass of the atomic specification, based in a general class `atomicData()` containing a set of sub dataclasses `datacell()` containing all the calculation and properties of the atom stored for each pair $|i\rangle\langle j|$, including the metadata.

H. Interface

Application-Programming-Interface (API)

The `API` is designed to stack a set of commands that simplify the use of the tools from the main modules. In this way, users have the possibility to generate their own functions for specific routines at different levels of the stack. It is important to note that the `API` works as an interface, which mean that it is also possible to use the same routines directly from the modules commands. Within the `API` it is possible to find three use targets: at the backend for management of the machine with additional simulations; at the middleware for design of waveforms that later

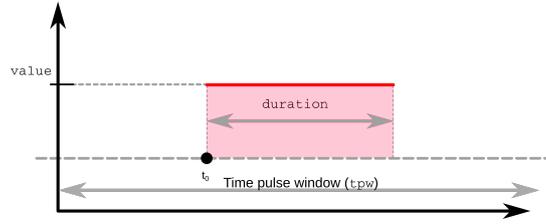


Figure 3.15: **API constant waveform function.**

can be use in experiments or simulations and finally at the frontend, where the commands are dedicated to handle task at the quantum circuit level.

Backend level

`showDAQ()`

It is a simple command that shows different graphics and data acquired by idle devices e.g., waveforms generated by the API as well as by the control module.

`generateSpecifications()` Loads the `.json` file file specification for hardware and software in order to generate the channels specifications, which consist in a database of all the channels that has been abstracted from the real channels in the devices for control of the setup. The flowchart of this command is shown below in appendix F.

`generateDirector()` Generates the director instance for the control of the setup by using the specification before mentioned with the possibility to include calibrations for future implementations. The flowchart of this command can be see in appendix F.

`generateProducer()` Similarly to the previous command, this command generates the producer instance that allows to generate a simplified model to be simulated using the `emulator` module. A flowchart of the functioning of the command can be seen in appendix F.

`runSimulation()` As the models can be run multiple times in the models of the `emulator`, it is possible to run these simulations from the `API` by passing the producer containing the models. The process is described by the flowchart in appendix F.

`prepareAcquisition()` Each device at the beginning of the running of the waveforms requires to set certain default values of the configurations. This command is focused in the configuration of acquisition devices. In this case an implementation for an iXon 897 camera using the developed driver is setting the parameters for starting the acquisition e.g., trigger mode, read mode, exposure time, amplification gain etc.

Additionally, there are a minimal set of waveform commands to be used in the context of the instances of `function`, `pulse`, `track` and sequence of the `control` module:

`constant()` This command generates a waveform with constant value of amplitude for a given duration in a time window (see fig. 3.15). The flowchart of this command can be found in appendix F.

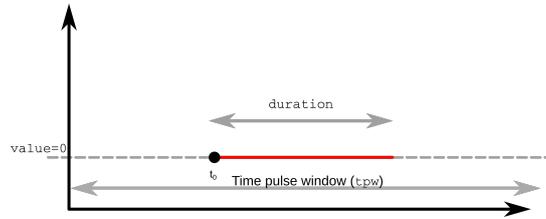


Figure 3.16: **API wait waveform function.** Instruction for enforcing a waiting time with a zero value waveform by a duration time starting a t_0 time.

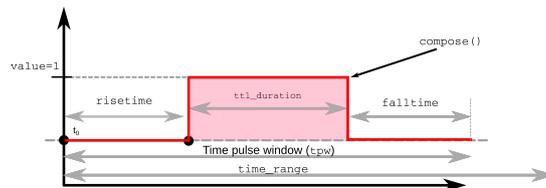


Figure 3.17: **API digital waveform function.** A single digital waveform defined by a step function from the **control** module, defining a waveform for TTL-like signals starting at t_0 time for a high value during TTL duration time considering the possible rise and fall times due to response and relaxation time along a time pulse window, an instruction usually set for digital or DC output signals.

wait() Similar to the constant function, but with an amplitude set to zero. See fig. 3.16 for a graphical representation. This command is used to generate the commands for **digital(s)** waveforms, in order to consider the rising and falling times coming from imperfection in the electronics of relaxation of opto-electronic systems.

compose() This command is used to generate the building blocks of **digital** and **analog** waveform(s). It allows to compose a chain of waveforms at given the necessary arguments of the waveforms to be generated and the flag-times that determine the relative starting time of each waveform inside the composed one. The flowchart of how this command works can be found in appendix F.

digital() As discussed before, the possibility of generating TTL pulses for digital control (i.e., HIGH or LOW) are fundamental to turn ON or OFF devices along the control sequence. This command of the **API** allow to generate such kinds of waveforms and link it directly to the device command. Therefore it requires the ID channel or the name identifier set in the hardware specification. Further details of the command arguments can be visualize in fig. 3.17 and the flowchart of the functioning of this command in appendix F.

analog() For the command to generate analog signals it is the same idea, but instead of receiving just time parameters and amplitude, the arguments might change depending on the specific function (from the **control** module) or if it is the case, the samples to be extrapolated (i.e., in **splines**) or a given array that is desired to be used. Additionally, the command gives the option to introduce a carrier frequency of the waveform, this is useful for those envelops that are mixed with RF or MW sources. The graphical representation of this command is in fig. 3.18 and the flowchart associated to the command in appendix F.

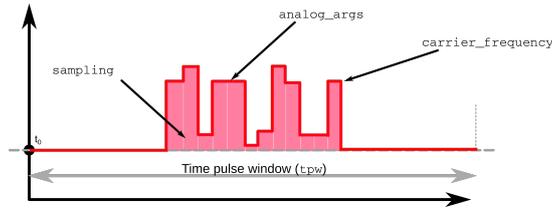


Figure 3.18: **API analog waveform function.** A single analog waveform defined by either analytic functions or point-defined **splines** from the **control** module, according to the function arguments passed to the instruction. The waveform will start at the t_0 time of the waveform. It is mixed with a carrier frequency functions to generate a full waveform including the RF nature or to passed to the hardware instruction.

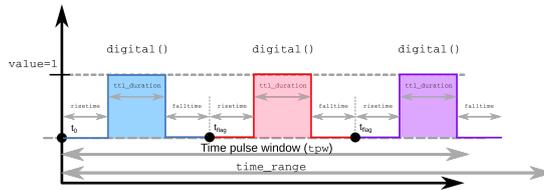


Figure 3.19: **API digitals waveform function.** The instruction sets a sequence of digital waveforms defined by a list of TTL durations and timeflag list to set the absolute initial time for each along the waveform for the pulse time window.

digitals() This command is an extension of the **digital()** by using the command **compose()**. This allows to set a chain of square waveforms or TTLs at given time-flags. The command checks if there are conflicts between the waveforms and raises a warning accordingly. A graphical representation of the output of the command can be observed in fig. 3.19 and the corresponding flowchart in appendix F

analog() Similarly to the **digitals()** commands, this generates a chain of analog signals according to a list of parameters for each waveform and the time-flag where each waveform starts. Moreover, a list of carrier frequencies can be passed in order to mix the envelopes of each analog waveform with its fast oscillating term. The fig. 3.20 shows a pictorial example of two analog waveforms (one after each other) using this command, whose operation is described in the flowchart in the appendix F.

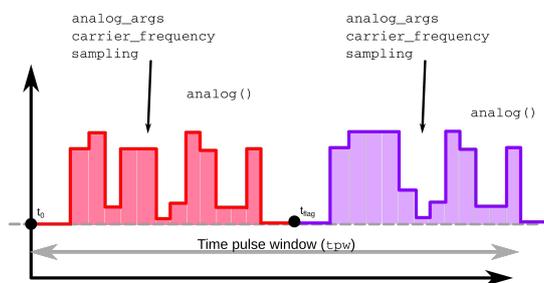


Figure 3.20: **API analogs waveform function.** The instruction sets a sequence of analog waveforms defined by a list of control module function arguments and carrier frequency and sampling list, allowing to set different carrier frequency for each analog waveform as well as the time resolution or sampling rate.

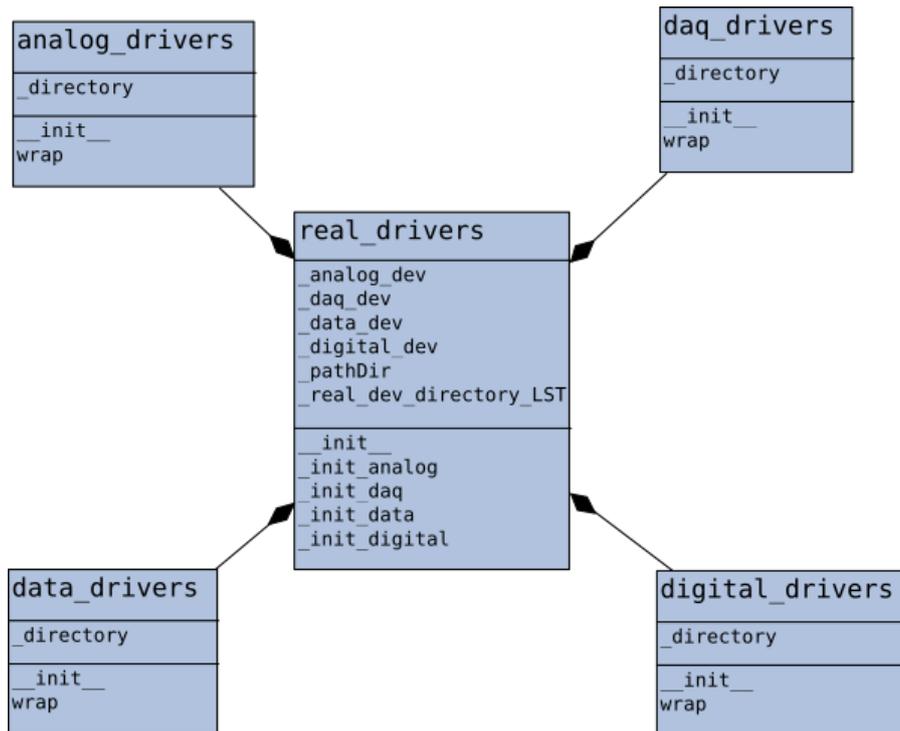


Figure 3.21: **Real devices drivers.** Class diagram of the different types of real devices wrapped in a single class from where we can extract the required driver contained in each class wrap attribute.

`data()` This command is meant to pass more complex information or data-structures to specialized devices such as the structured-light-modulators i.e., SLMs and DMDs. For example, passing images to generate the array shapes.

Middleware and frontend commands of the API will be added in coming versions of the software, but in the meantime it is possible to realize any of the tasks at these levels with the modules.

3.2.3 Drivers

The Backend is composed not just of the quantum system but also of all the classical components that either control the different physical operations related to the Hamiltonian terms that represent the quantum phenomena e.g., AOMs, EOMs, resonant beams or the semi-classical physics that composes the substrate that holds the quantum system e.g., optical tweezers, atom sources. Therefore, normally these devices are controlled directly or indirectly with software, which is based on standard command protocols i.e., Serial-Peripheral-Interface (SPI) and others based on a low-level programming language built by the manufacturer e.g., Python wrappers such as QUA from Quantum Machines and `KeysightSD1` from Keysight Technologies. These drivers are organized in the software system as part of the real Hardware abstraction: Analog, Data-acquisition (DAQ), Data and Digital. Each of these drivers are wrapped in an abstract Python object that is later linked accordingly to the Hardware specifications. See fig. 3.21 for the class diagram.

Virtual devices have also been included in the software system with virtual drivers, which can be used for troubleshooting the backend development. See fig. 3.22 for the



Figure 3.22: **Devices classes with virtual drivers.** Abstraction of classes for devices with custom functions to mimic the real devices instructions (i.e., virtual driver) but also possible outputs, such that it can be used to simulate a hypothetical experimental sequence.

class diagram.

3.2.4 Compiler

Compiling as seen in the paradigm of three layers can be found at different levels. In AQiPT the included compiler does not just transform code and instructions from a classical language as it is Python to another like C or C++. Although this kind compilation happens when the software executes commands from the devices' wrappers (i.e., firmware instructions compilation), it does compile the input arguments given to the commands of the modules or `API` into the AQiPT objects and instructions that afterwards enable the execution of different other task.

One of the AQiPT compilers is for the backend, that is executed when compiling the `experiment` class. This compilation generates a sequence of hardware instruc-

tions by creating in the director a map of the software instructions and the hardware set by the specifications. Then, when the `experiment` class is compiled, an assembler matches the commands from the firmware wrappers with the instructions. Afterwards, the `experiment` class instance is executed together with the real devices actions.

Another compiler lies in the `emulator` module. In this case, it compiles the theoretical models to be simulated, for example an `atomicModel` or `atomicQRegister`. The compilation of waveforms takes the arguments of the `pulses` and `tracks` that are used in the hardware schedule and gate protocols, and calculates synthesize the waveform values.

A third compiler included in this version of AQiPT is the translator compiler (i.e., transpiler), in this case the compiler brings instructions between the quantum circuit formalism i.e., quantum native gates to the native version of the platform i.e., quantum canonical gates. The circuit construction uses the framework of the open-source software Qiskit and delivers the schedules and simulations from the `emulator` and `control` modules. This compiler was part of an engineering thesis project which I co-supervised [Álv23].

3.2.5 Learning Materials and Documentation overview

All the online documentation has been generated by Sphinx which takes the internal code documentation with the descriptions and explanation of the commands and classes.

Within AQiPT I tried to employ a straightforward and intuitive documentation scheme for its quantum information processing software [Mor], encompassing:

- Tutorials: Dedicated to hands-on learning of the framework.
- How-to guides: Concentrating on the fundamental building blocks for developing solutions with AQiPT.
- Explanatory examples: Focused on learning and understanding specific concepts using the tools provided by AQiPT.

Much of the documentation is executed through Jupyter notebooks, facilitating interactive calculations and tool utilization. It also includes explanations complemented by embedded equations and figures.

3.2.6 Integration

The advantage of AQiPT being Python-based and modular lies in the flexibility to remove and replace modules according to specific needs. However, it is acknowledged that this process typically entails a substantial effort, akin to installing any classical peripheral into a personal computer. The approach of AQiPT is to seamlessly integrate new modules or module versions directly into the existing modules. Leveraging its open-source nature, this integration is straightforward, even if the new modules are in other languages like C++ (compiled). This is achieved using tools such as `Ctypes` and the `Pybinding` Python module. Alternatively, for integration with modules written in Julia, `PyJulia` is employed.

3.2.7 Examples

- Frontend

This implementation begins with a high-level abstraction, utilizing quantum circuit formalism for quantum computing. It involves implementing a quantum circuit based on the classes provided by the open-source software package `Qiskit`. Subsequently, an AQiPT Rydberg quantum circuit class can be generated by transpiling the circuit with canonical gates into native gates. These native gates are then mapped into an expandable set of schedules for a `RydbergQubit` or, more broadly, for a `RydbergQuantumRegister`. Each of these classes contains schedules associated with different terms of the Hamiltonian, adhering to the gate protocols of the mapping.

```
1  # bancked for the circuit
2  sim_config = aqipt._SimulationConfig(time_simulation=0.2)
3  circuit_backend = aqipt.BackendConfig(simulation_config=
4  sim_config)
5
6  # quantum circuit with 1 qubit and 2 Hadamards
7  qc = compiler.RydbergQuantumCircuit(1)
8  qc.h(0)
9  qc.h(0)
10 qc.draw(output="mpl")
11
12 # transpiled circuit into native algebraic gates and pulse
13 # schedule of Hamiltonians
14 pulsed_circuit = compiler.Transpiler(backend_config=
15 circuit_backend)
16 ryd_schedule = pulsed_circuit.transpile(qc)
17 rydberg_register = pulsed_circuit.build_transpiled_circuit(
18 init_state=qt.basis(4 ,0))
19 ryd_schedule.plot_schedule()
```

Listing 3.12: Use case gate transpilation at Frontend layer.

Upon completion of the implementation, we derive the equivalent pulse schedule for all qubits and the physical realization of gates for the canonical quantum circuit comprising two consecutive Hadamard gates, see fig. 3.23.

Another example of the software at this level is the implementation of the QFT algorithm internally in AQiPT, see fig. 3.24. The implementation of this algorithm in the software was part of a co-supervised undergrad thesis [Álv23].

```
1  import qutip as qt
2  qft_backend = aqipt.BackendConfig( simulation_config = aqipt.
3  _SimulationConfig(time_simulation=1.5) )
4
5  qc = compiler.RydbergQuantumCircuit(3)
6  qc.qft(num_qubits=2)
7  qc.draw(output="mpl")
8
9  psi0=(qt.basis(64,0));
10 pulsed_circuit = compiler.Transpiler(backend_config=qft_backend
11 )
12 ryd_schedule = pulsed_circuit.transpile(qc)
```

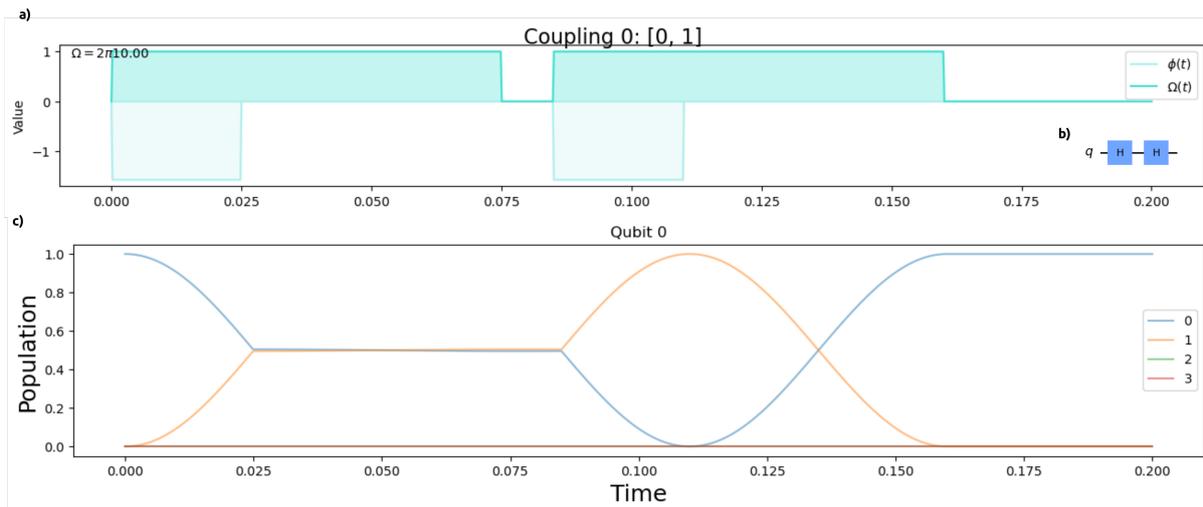


Figure 3.23: **AQiPT transpiler.** Application of the example of a quantum circuit with two consecutive Hadamard gates applied to a single qubit circuit generated using Qiskit tools (top-miniature), generating the Hamiltonian schedule of the 4-level system with 2 ground states and 2 Rydberg states (top) and the calculated dynamics from the emulator module.

```

11 rydberg_register = pulsed_circuit.build_transpiled_circuit(
12     init_state=psi0)
13 rydberg_register.atomic_qregister.showResults(plot_mode="
matplotlib", figure_size=(18, 6))

```

Listing 3.13: QFT algorithm implemented in AQiPT transpiler.

- Middleware

Applications of the software for this layer are related to the fact of designing quantum gate operations at the level of time-dependent Hamiltonians, the case for a $R_x(\theta)$ gate is shown in fig. 3.25. Internally the software can do the bridge between the low-level hardware waveform signals with higher level layer, compiling quantum circuits and the contained operations on it as native gates.

The example presented here is the design of a $R_x(\theta)$ native gate in a 2-level system with gg-qubit encoding.

```

1 ggQubit_psi0 = 0; #initial state
2
3 #couplings list within the states [i,j], couplingValue, pulse
4 couplings = {'Coupling0': [[0,1], 2*np.pi*1, None ]};
5
6 #detuning list in the state i as [i,i], detuningValue, pulse
7 detunings = {'Detuning0': [[1,1], 0, None]};
8
9 #dissipator list within the states [i,j], dissipatorStrenght
10 dissipators = {'Dissipator0': [[0,1], 0]};
11
12 #label of Rydberg states in the atomicModel (optional)
13 Rydbergstates = {'RydbergStates': [], 'l_values': []};
14

```

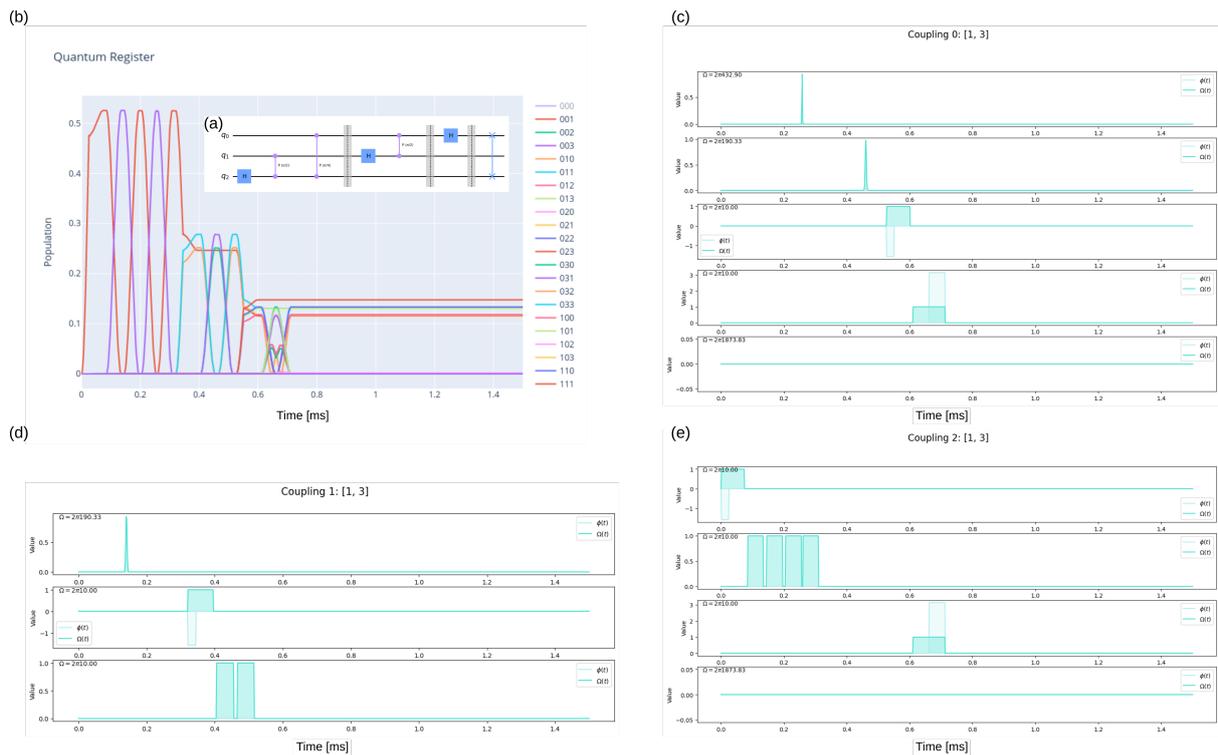


Figure 3.24: **QFT algorithm.** (a) Quantum circuit of a 2 qubit QFT algorithm. (b) Dynamic simulation of the quantum circuit schedule. (c) Schedule of the time-dependency of the Hamiltonian associated to the QFT quantum circuit. (d) Schedule of the time-dependency of the Hamiltonian associated to the QFT quantum circuit. (e) Schedule of the time-dependency of the Hamiltonian associated to the QFT quantum circuit.

```

15 #wrapper dictionary for the dynamic params
16 ggQubit_sim_params = {'couplings': couplings,
17                       'detunings': detunings,
18                       'dissipators': dissipators,
19                       'rydbergstates': Rydbergstates};
20
21 #model class of AQiPT model(timeOfdynamics, NumberOfLevels,
22   InitialState, DynamicParams, NameLabel)
23 ggQubit = emulator.atomicModel(tbase, Nrlevels, ggQubit_psi0,
24   ggQubit_sim_params, name = 'gg-qubit');
25
26 #Graph-Map of the atomicModel
27 ggQubit_Map = ggQubit.modelMap(plotON=True, figure_size=(3,3));
28
29 #building Time-indepnedent Hamiltonian
30 ggQubit.buildHamiltonian();
31
32 #building Lindbladans
33 ggQubit.buildLindbladans();
34
35 #building Observables
36 ggQubit.buildObservables();
37
38 #setting new time dependencies
39 ggQubit.dynParams['couplings']['Coupling0'][2]=funcs_lst[0]; #laser
40   beam coupling state 0 to state 1 with Omega_Rabi = 1 * square
41   pulse pi/2
42 ggQubit.dynParams['detunings']['Detuning0'][2]=funcs_lst[1]; #laser

```

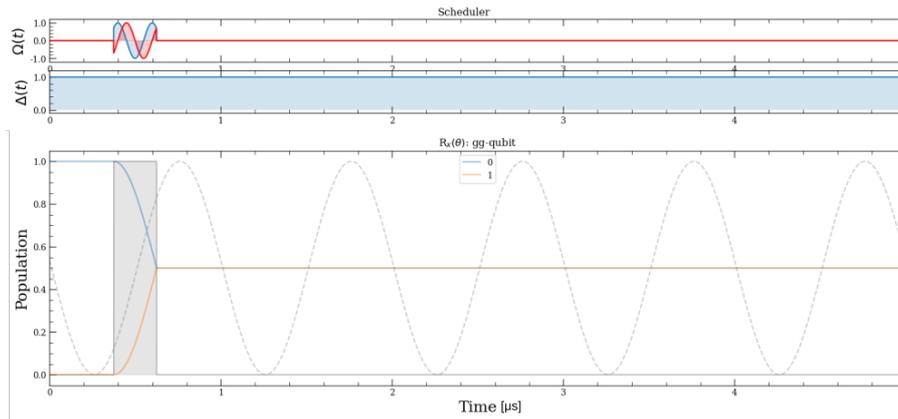


Figure 3.25: **Gate design simulation.** Time dependency of the Hamiltonian parameters for intensity of the laser and phase (i.e., $\Omega(t)$ and $\varphi(t)$).

```

39 beam coupling state 0 to state 1 on constant resonance
40
41 ggQubit.recompile(mode='control');
42 ggQubit.playSim(mode='control');
43 ggQubit.showResults(figure_size=(18,5), resultTitle=r'R$_{x}(\theta)$: '+ggQubit._name);
44

```

Listing 3.14: Use case gate design of $R_x(\theta)$.

- Backend

fig. 3.26 shows the setup for the backend implementation. This implementation of AQiPT serves the purpose of functioning as hardware control software for optical pumping of atoms. The objective is to prepare cold atoms of Potassium-39 in magnetic-insensitive Zeeman states of a low-lying ground state, specifically $|4S_{1/2}, F = 1, m_F = 0\rangle$. The setup involves two laser beams addressing the atoms after passing through a shutter and an AOM: the Cooler and Repumper, coupling $|4S_{1/2}, F = 2 \rightarrow 4P_{1/2}, F = 2\rangle$ and $|4S_{1/2}, F = 1\rangle \rightarrow |4P_{1/2}, F = 2\rangle$, respectively. A set of coils is used to create a homogeneous magnetic field that interacts with the atom, producing Zeeman splitting of the hyperfine structure. An additional Radio Frequency (RF) field is applied using an antenna connected to an RF source to couple the two insensitive magnetic states.

To implement AQiPT (fig. 3.27), the following steps can be followed:

- Step. 1** Load AQiPT libraries.
- Step. 2** Generate and load specifications.
- Step. 3** Generate and load variables.
- Step. 4** Generate hardware and dynamic parameters.
- Step. 5** Define parameters for analog instructions.
- Step. 6** Create sequence blocks with `API` instructions.
- Step. 7** Update software specifications with new sequence blocks and variables.

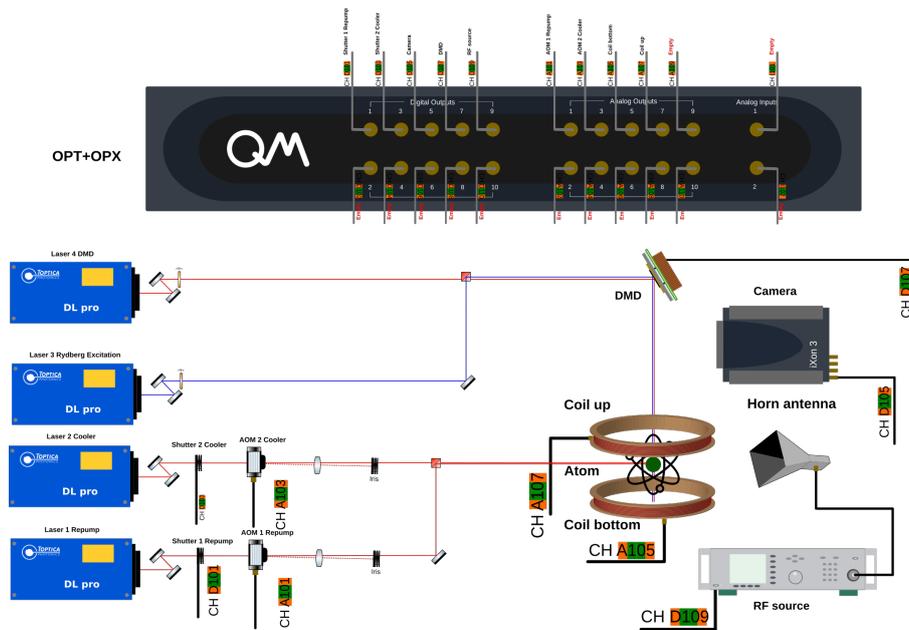


Figure 3.26: **Setup for backend implementation.** A simplified experimental scheme for a hypothetical for optical pumping using 2 lasers with single pass AOMs for intensity control, magnetic coils, RF antenna driven by a RF source and a camera for detection. Additional connection scheme for an OPX unit channels for each controlled elements.

Step. 8 Generate director.

Step. 9 Generate producer for simulation.

Step. 10 Add actors.

Step. 11 Create an experiment with the director, producer, and database.

Step. 12 Compile experiment.

Step. 13 Execute experiment.

Step. 14 Execute simulation.

Step. 15 Show data acquisition.

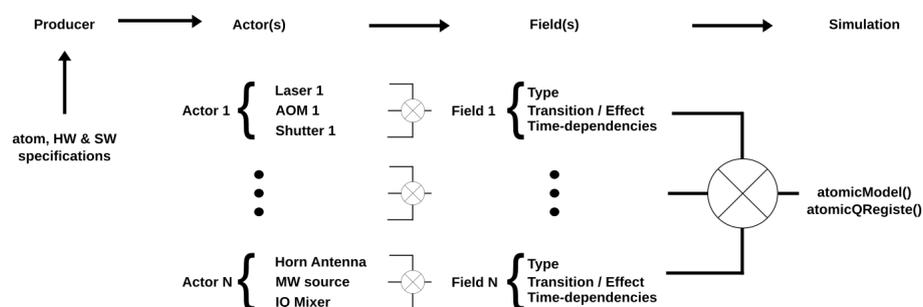


Figure 3.27: **Workflow backend implementation.** Flow diagram for generating an experimental sequence but also a simplified model generated by the `director` and `producer`.

This process will execute an experimental sequence and will simulate the linked model for such a protocol, based on the actions realized by different elements of the experiment in terms of time-dependent Hamiltonians.

In this example, the OPX+ is considered as the Main Controller, producing electrical signals for driving mechanical shutters using digital pulses, AOMs, and coils using analog pulses, and triggering the RF antenna with digital pulses to the pre-configured RF source with the carrier signal that needs to be played. The waveform schedule is programmed with high-level instruction `API` commands in a single sequence instance, which sends the waveform to the channels of devices as specified. The schedule is then compiled later in the software specification file.

```

1 #load specifications
2 specifications = generateSpecifications(mode = 'waveforms', hwPATH
   = '~/AQiPT/hardware', swPATH = '~/AQiPT/compiler');
3 my_HWSpecs, my_SWSpecs, my_channels = specifications['Hardware'],
   specifications['Software'], specifications['HWChannels'];
4
5 #define parameters for waveforms
6 AOM_Repump = {'amp': 1, 't_o':5.5+0.1, 'width': 0.5, 'tp_window':7,
   'name': 'AOM_Repump', 'type': "square"};
7
8 AOM_Cooler = {'amp':1, 't_o':0+0.1, 'width': 5, 'tp_window':7, '
   name': 'AOM_Cooler', 'type': "square"};
9
10 RF_field = {'amp': 1, 't_o':0, 'width': 0.5, 'tp_window':0.5, '
   name': 'RF_field', 'type': "square"};
11
12 #initialize sequence with the experimental instructions
13 Seq_params = aqipt.general_params({'sampling':int(1e3), 'bitdepth'
   :16, 'time_dyn':1} );
14
15 State_preparation = control.sequence('State preparation', tSequence
   =None, stack=[ ]);
16
17 State_preparation.stack2Sequence([digitals('Coil_up', [0, 7], [0],
   [3.5], 'B-field +'),
18
   digitals('Coil_bottom', [0, 7],
19
   [0], [3.5], 'B-field -'),
   digitals('Shutter_1', [0, 7],
20
   [5.5], [0.5], 'Shutter Repump'),
   digitals('Shutter_2', [0, 7],
21
   [0], [5], 'Shutter Cool'),
   analog('AOM_1', AOM_Repump,
22
   Seq_params.sampling, None, 16, 'AOM Repumper SP'),
   analog('AOM_2', AOM_Cooler,
23
   Seq_params.sampling, None, 16, 'AOM Cooler SP'),
   digitals('RF_source', [0, 7],
24
   [5,6], [.5,.5], 'RF source'),
   analogs('Horn_antenna', [0, 7],
25
   [5, 6], [RF_field, RF_field],
   [500, 500], [None, None],
26
   [8, 2], 'RF field',
   aqipt.general_params({'
27
   sampling':int(1e3), 'bitdepth':16, 'time_dyn':1}),
   None, False, [[-1,1],
   [-1,1]]) ] , _IDs=True);

```

```

28
29 State_preparation_plot = State_preparation.plotSequence(plotMode='
    dynamic-subplots', dashboard=True); plt.close('all');
30
31 #update the software specs with new sequence
32 my_SWSpecs.update(waveforms=[State_preparation, State_preparation],
    variables = None, save_waveforms=True);
33
34 director4Exp = generateDirector(my_HWSpecs, my_SWSpecs,
    calibrations=None);
35
36 #prepare model for simulation
37 producer4Exp = emulator.producer(atom=atom,
38                                 name='test_producer',
39                                 director=director4Exp);
40
41 bfield_states = [{'l':0, 'j':0.5, 'f':1, 's':0.5, 'mf':-1},
42                  {'l':0, 'j':0.5, 'f':1, 's':0.5, 'mf':0},
43                  {'l':0, 'j':0.5, 'f':1, 's':0.5, 'mf':1},
44                  {...}]];
45 producer4Exp.addActors('Repumper', 'optical', ['Laser_1', 'AOM_1',
    'Shutter_1', 'OPX_1'], {'role':'coupling', 'args'
    :[[[0,14],[1,13],[2,12]], 2*np.pi*5, State_preparation._Stack
    [2].digiWaveform *State_preparation._Stack[4].digiWaveform]);
46 producer4Exp.addActors('Cooler', 'optical', ['Laser_2', 'AOM_2', '
    Shutter_2', 'OPX_1'], {'role':'coupling', 'args'
    :[[[7,14],[6,13],[5,12],[4,11],[3,12],[4,13],[5,14],[6,15]], 2*
    np.pi*1, State_preparation._Stack[3].digiWaveform *
    State_preparation._Stack[5].digiWaveform]);
47 producer4Exp.addActors('RF', 'rf', ['Horn_antenna', 'RF_source', '
    OPX_1'], {'role':'coupling', 'args':[[1,5], 2*np.pi*1,
    State_preparation._Stack[6].digiWaveform]);
48 producer4Exp.addActors('B-field', 'magnetic', ['Coil_up', '
    Coil_bottom', 'OPX_1'], {'role':'zeeman-splitting', 'args':[{'
    atom':atom, 'bfield':0.0001, 'state_lst':bfield_states}]});
49 producer4Exp.compile(psi0=emulator.AM_superpositionState('012', 16)
    , t_Hamiltonian=True);
50
51 #initializa experiment
52 qpu_backend = control.experiment(atom=atom, waveforms=None,
53                                 hardware = my_HWSpecs,
54                                 director = director4Exp,
55                                 operationMode = 'sequence',
56                                 database=None,
57                                 producer=producer4Exp,
58                                 metadata = 'Experiment for Optical
    pumping. ');
59
60 qpu_backend.compile();
61
62 qpu_backend.execute(); #execute experimental sequence

```

Listing 3.15: Implementation of optical pumping experimental sequence and simulation.

In addition to the experimental sequence described above, it is possible to generate a theoretical model under the rotating wave approximation. This can be done with a set of actors defining specific physical processes, or in mathematical terms, Hamiltonian operator terms for coupling, detuning, or energy shifting the energy

states of the model. The time dependencies of these operators are defined from the waveform schedule, combining the waveforms of the set of elements or instruments into a single time-dependency array passed to the model. The flow diagram on how the model for the simulation is generated is described in the figure below.

```

1 producer4Exp.compile(psi0=emulator.AM_superpositionState('012', 16)
    , t_Hamiltonian=True, simulation_time=np.linspace(0,21,21000));
2 producer4Exp.runSimulation();

```

Listing 3.16: Implementation of optical pumping experimental sequence and simulation.

3.2.8 Performance

In this section I present a series of measurements of the time benchmark of the main functions and instantiation of the classes defined in the modules of AQiPT. The measurements are using the Python package `time`. These results could be further improved by parallelization of the code and GPU acceleration with packages as `CUDA`³ which is out of the scope of this thesis.

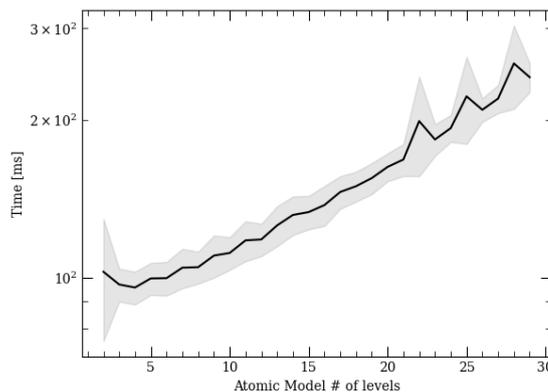


Figure 3.28: **AQiPT atomicModel performance without simulation.** A linear growth below 300 milliseconds executing time of the generation of a single atomic model from the emulator with up to 30 level system and 15 couplings without executing the simulation.

First we observe the performance of the benchmark of the `emulator` module class as the `atomicModel` by measuring the time spent in generating the Python object for models at different number of levels, not including the simulation time (see fig. 3.28) and including the simulation time (see fig. 3.29) averaged over 100 trials. The performed time increased up to 300ms (without simulations) and up to 1s (simulation with 1000 points in the time domain) for models that reach 30 levels in the system and 15 arbitrary couplings.

Another set of classes instantiation that takes an important computational time is the generation of waveforms or arrays of points that represent the time dependency of the Hamiltonian of the system. I have introduced the main elements of the `control` module,

³NVIDIA® CUDA® Toolkit provides a development environment for creating high-performance, GPU-accelerated applications. <https://developer.nvidia.com/cuda-toolkit>

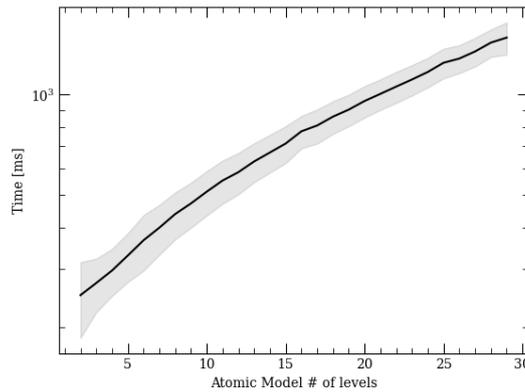


Figure 3.29: **AQiPT atomicModel performance with simulation.** A linear growth of the atomic model generation and simulation execution of a single atomic model with up to 30 levels and 15 couplings.

here I show the time performed by the software on generating the functions with up to 1 million points functions like: step, Gaussian and a quadrature function (see fig. 3.30) taking up to hundreds of microseconds in generating 1 million points for the quadrature waveform which agrees with the approximately the double of the time taken by a simple waveform as the Gaussian function, since the quadrature calculates two waveforms i.e., real and imaginary component.

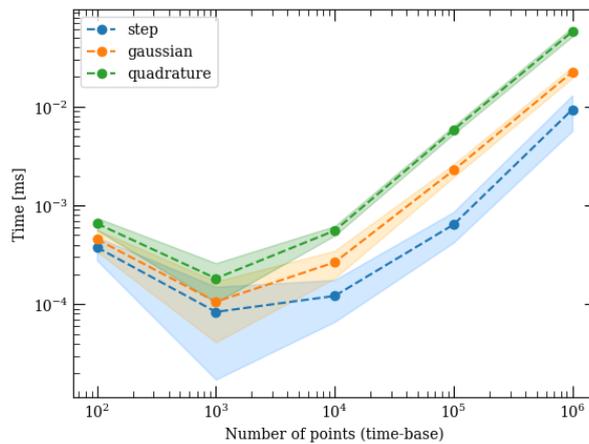


Figure 3.30: **AQiPT function class performance.** Measurement of time performance of waveform generation with AQiPT control module for three different functions: step, Gaussian and quadrature (real and imaginary sinusoidal functions). By measuring the waveform generation time while increasing the number of points defining the waveform.

Additionally by joining many **function** objects, an estimated time for creating instances of **pulse**, requires up to 100 functions is below 0.5ms. In other words, 100 thousands (10^5) points waveform has been performed around 100 times, within 0.5ms time, see fig. 3.31.

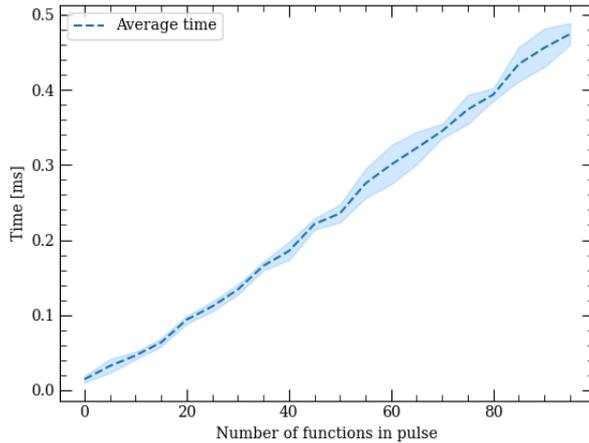


Figure 3.31: **AQiPT pulse class performance.** Measurement of time performance of waveform generation with AQiPT control module for pulses composed by sequence of functions with 1000 points each with up to 100 functions with performance below 0.5ms.

Similarly, classes instances of track using the largest pulse from the previous benchmark take up to 6ms when 100 of pulses are stack behind each other to form the track fig. 3.32. If each channel in the experimental sequence is parallelized (e.g., 1 core/1 track), a reasonable number of channels (e.g., 1000 channels) for 1000 qubits, then it is possible to compile within 6ms a 100 gates circuit.

3

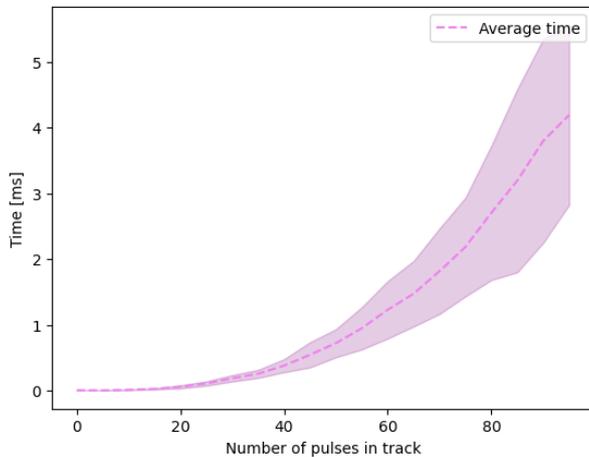


Figure 3.32: **AQiPT track class performance.**

Using the same pulses prepared for the track benchmark, the sequence benchmark requires a relative short time, since it does stack (vertically) copies of the tracks. For 1000 copies of the tracks in the same sequence, it requires up to $0.6\mu s$. Here is where the pythonic nature of the software can be use in favor, since it does not create a full new object but instead creates a register with pointers to the memory allocation of that object, allowing to use it flexibly along the script. See fig. 3.33.

Finally, other performed times were measured for specification loading into memory

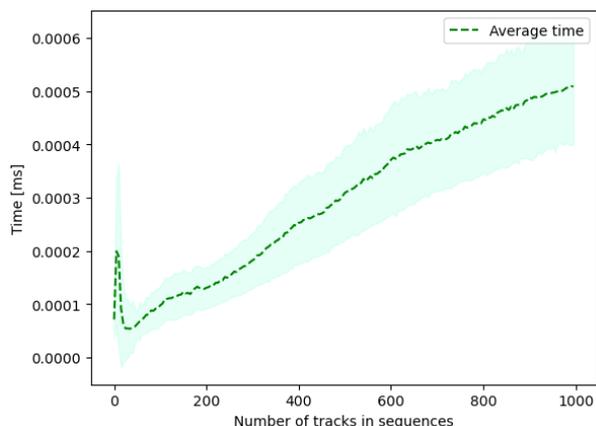


Figure 3.33: **AQiPT** `sequence` class performance.

from the JSON files (118ms) and 253ms for software specification update, the creation (1ms) of producers and actors (129ms), as well as well as the director (408ms) and the experiment, including the compilation, execution and loading of a **DAQ** monitor of the waveforms (10s). The compilation of the simplified theoretical model took on average 400ms.

3.2.9 License

AQiPT is an open source software, and all of its source code is freely available online under the BSD-3 license. The github repository is divided into eight separate folders containing the compiler codes with pre-defined code generating models for atoms, gates, circuits, waveforms and programs for experimental sequences. The configuration folder contains relevant files for the configuration of devices and AQiPT modules. Inside the depository folder it is possible to store any associated data related to simulations and/or hardware runs. Sphinx documentation is located in the docs folder and the hardware folder contains all the used drivers by AQiPT, including the wrappers for virtual devices and real devices that are divided in: Analog, DAQ, Data and Digital like devices. The main code of AQiPT lies in the modules folder where the 8 modules described in the previous sections are located. Exemplary jupyter notebooks and scripts can be found in the notebooks folder. Many other useful elements such as logos, relevant figures, diagrams and more can be found in the resource folder.

Additionally, there is a general script in the repository which contains many utilities used in the modules but they can also serve as tools.

To install AQiPT, users can either clone the main github repository directly and install via `pip install`, or alternatively download and install the package through the PyPi or Anaconda package repositories (see [Mor]).

AQiPT can also be cited. The relevant information can be readily accessed by executing the cite function from the utilities, as follows:

```

1 from AQiPT import AQiPTcore as aqiPT
2
3 aqiPT.cite()

```

Listing 3.17: Command for citing the AQiPT software.

3.3 Challenges and improvements

- Technical

In addressing technical challenges, we encounter limitations in generating a large number of floating points that constitute the waveforms of the experiment for visualization, simulation and signal generation using internal memory of the waveforms generator. These limitations impede our ability to control the resolution down to the smallest time-step when we scale to a large amount of qubits. As the number of qubits increases, the demand for waveforms also rises, which in turn increases the volume of data that must be managed during the computation. Consequently, developing an experimental sequence using a waveform-centered approach becomes challenging. However, with the presence of compiling-on-the-fly control hardware, it becomes feasible to adopt a more instruction-centered approach.

In the community of software engineering, there are ongoing debates over several programming languages, each with its set of popular options. Likewise, the quantum computing domain faces a similar dilemma. Solutions can be found using compiled languages such as C and C++, or interpreted languages like Python. However, hybrid solutions (pre-compiled), such as the Julia language, offer an intriguing middle ground, albeit with compromises in versatility and reliability. This compromise affects the availability of drivers for classical hardware, resulting in a range of challenges when it comes to integrating them, which varies depending on the specific experimental implementation. This challenge can be overcome by considering AQiPT as a “*mise à point*” software enabling pre-configuration of the experimental devices, making them ready to function. Significant advances in the field of Domain-specific languages (DSLs) such as Codon [Sha23] position Python as a strong candidate, enhancing performance close to compiled programming languages.

- Conceptual

Because of the technical challenges involved, the ideal notion of a universal, uniform, and standardized framework for a quantum stack with complete controllability becomes contingent on the implementation. This framework is not only limited by the hardware employed but also by the time-scale of the quantum architecture, resulting in challenges when integrating various abstract layers.

As a result the parser of the experimental setup might necessitate drivers in multiple programming languages to enable comprehensive control of the hardware and experimental sequences, all while preserving the implementation layout.

In summary, a new hardware generation is now available for quantum atomic architectures and specifically in our quantum simulator in Strasbourg, with features as fast feedback and generation of waveforms on-the-fly.

Moreover, a more flexible and general way to control the system has been developed along this work. Which is based on Python language with the capability to also generate simple but accurate simulations and link hardware with quantum computing features such as quantum circuits using the native gate set demonstrated with examples at different levels of the proposed quantum software stack.

Quantum backend for atomic QPUs

“People who are really serious about software, should make their own hardware.”
A. Kay (1977)

This chapter presents a summary of the experimental apparatus (backend) after reassembly at our new facilities at the European Center for Quantum Sciences (CESQ). It describes the experimental setup requirements and components of our setup for generating ultracold atomic gasses trapped in optical traps. In this chapter I also introduce the contributions on the generation of hundreds of optical tweezers and their coherent control using a new global control using microwaves. The chapter ends with a recap of the experimental upgrades and planned future upgrades.

This section of the thesis chapter partially presents findings and insights from a co-authored article, “Preparation of hundreds of microscopic atomic ensembles in optical tweezer arrays”. - Y. Wang, S. Shevate, T.M. Wintermantel, M. Morgado, G. Lohead & S. Whitlock. npj Quantum Information volume 6, Article number: 54 (2020), serving as a key reference [Wan20].

4.1 Minimal requirements for atomic Quantum backends

Similar to other architectures, the platform fulfills a minimal set of requirements that allows it to control the quantum system, which ultimately represents the QPU. For atomic systems this minimal set of requirements can be summarized as follows:

- Atomic element
- Environment
- Atomic initialization
- Atomic substrate preparation
- Manipulation and calibration

- Measurement and readout

Using this set of requirements I will explain the revision of the experimental setup that we have built in Strasbourg with a bottom-up approach, see fig. 4.1.

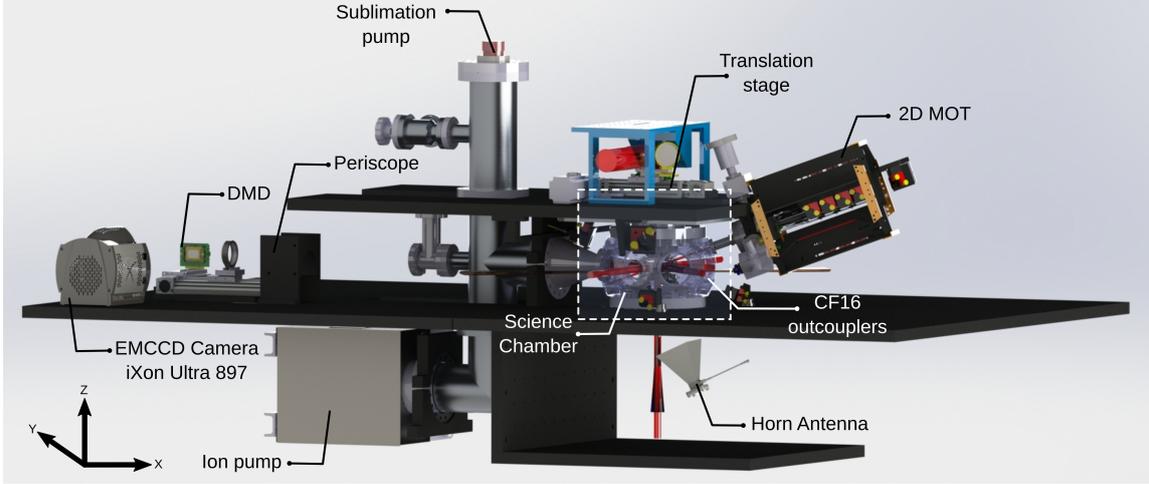


Figure 4.1: **Experimental apparatus in Strasbourg.** General visualization of the elements in the experimental setup, including the 2D MOT, Science chamber, camera, DMD, antenna and vacuum system.

4.1.1 Atomic element

In this chapter and the following chapter, in which the experimental results are discussed, I will focus the explanation around the atomic species of potassium-39 (^{39}K). An enriched dispenser with natural isotope abundances of Potassium¹ has been chosen: 93% for ^{39}K , 0.01% for ^{40}K and 7% ^{41}K . The selection of this alkali atom is due to its simple atomic structure (i.e., Hydrogen-like) with one electron in the outer energy level. The structure of this element gives the flexibility of using a standard scheme with D1 transitions i.e., $|4S_{1/2}\rangle \leftrightarrow |4P_{1/2}\rangle$ (770nm) and D2 transitions i.e., $|4S_{1/2}\rangle_{x\text{-over}} \leftrightarrow |4P_{3/2}\rangle$ (767nm). The D1 and D2 transitions can be used for Gray-molasses cooling thanks to the fast decay from intermediate P states with a $\sim 6\text{MHz}$ linewidth.

Transitions to the Rydberg state with the standard scheme involves the use of a 461nm blue laser to Rydberg states, starting at 4P states. It is also possible (and previously installed in our group) to use a 287nm UV laser for single photon transitions (i.e., $4S_{1/2} \leftrightarrow nP$). When using the inverted scheme of ^{39}K the first transition is 960nm to 5P states and 750nm to Rydberg states. A simplified scheme of potassium-39 is shown in the diagram fig. 4.2 and a complete map of the possible transitions in ^{39}K can be found in appendix C. Calculations of some of its properties can be found in chapter 2.

4.1.2 Control hardware

In order to introduce the atom in a region of control, the backend should have a mechanical structure that supports all the optical elements, mechanical parts and electrical components; these offer good isolation to the quantum system, avoiding noise and errors

¹Alvatec AS-3-K-100-F with natural abundance distribution.

^{39}K Energy Scheme

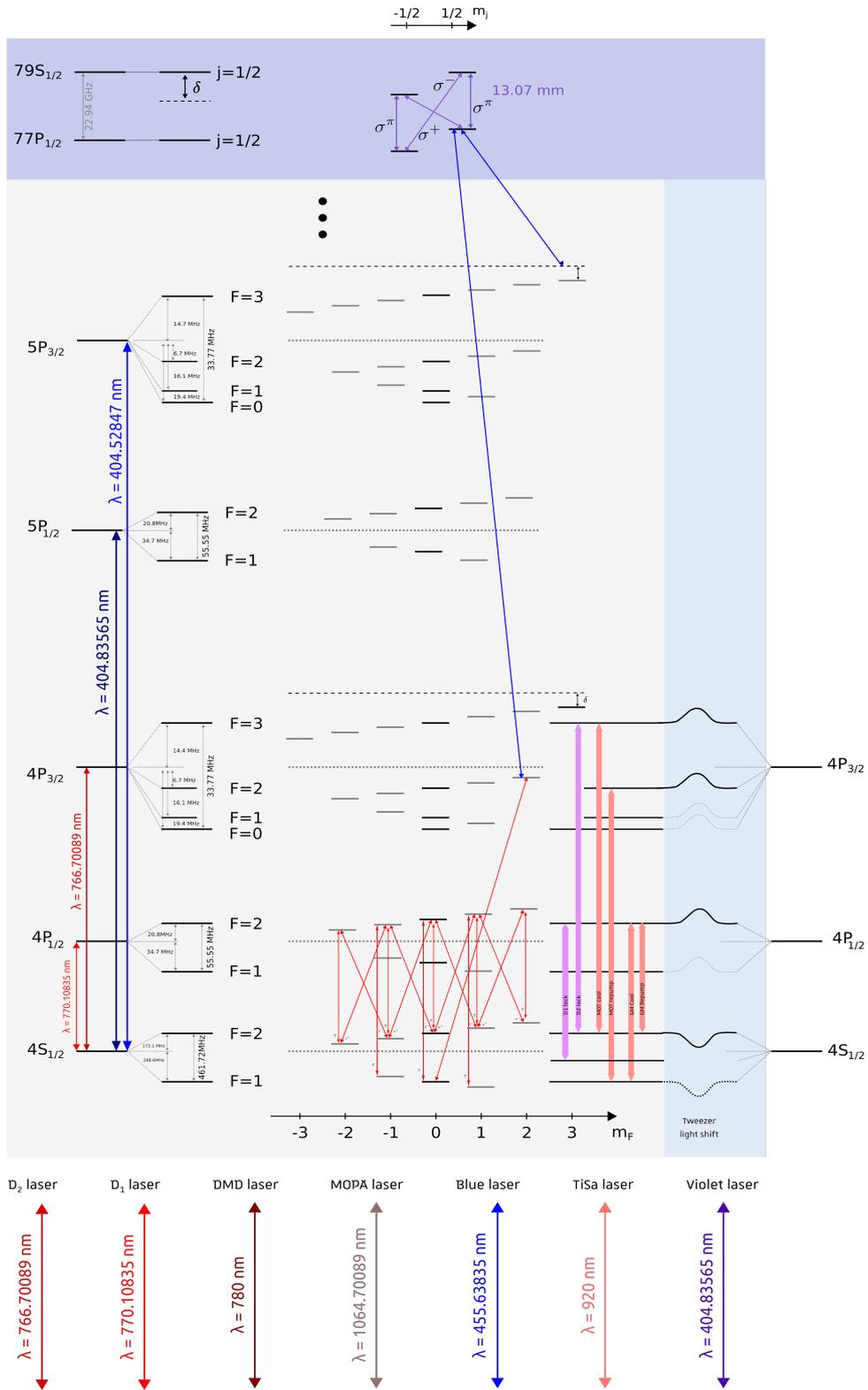


Figure 4.2: **Simplified scheme of potassium-39 (^{39}K).** Atomic energy level structure with energy gaps between fine and hyperfine states and optical transitions in the grey area and microwave transitions of Rydberg states in the purple area.

from the interactions from the environment while allowing to certain level of manipulation (see fig. 4.1). The first step is then to use a ultra-high vacuum (UHV) system that allows it to host the atom but at the same time to interact with light using laser beams. In our apparatus we use a 8" spherical square stainless steel chamber that we use as the Science chamber².

The maintenance of a pressure of 3×10^{-9} mbar in the science chamber is facilitated by the utilization of an ion pump³ and a titanium-sublimation pump⁴. Additionally, the chamber is outfitted with six CF63, four CF40, and sixteen CF16 stainless steel viewports⁵, guaranteeing extensive optical access (see fig. 4.1). Its 8-inch diameter allows for the accommodation of various components within the vacuum, including the objective lens⁶, a radio-frequency (RF) coil, and a Rydberg Micro-channel plate (MCP) detector⁷ equipped with electric field compensation electrodes (see fig. 4.6).

Two more important elements of the environment are: the laser-beam preparation and the electronics. As mentioned above for ³⁹K in the standard scheme it is possible to use D1 and D2 transitions for cooling, trapping, state preparation, manipulation and readout. We make use of a single-pass (SP) and a double-pass (DP) optical configuration to control the amplitude and frequency of the beam respectively, for scanning and calibration in the sequences of the backend.

In the first optical breadboard two Toptica DL Pro diode lasers are located, they are set to the wavelengths: 767nm (D1) and 770nm (D2). They have installed a SP and a DP, being the second attached to a atom vapor cell for Modulation Transfer Spectroscopy (MTS) for locking to the spectroscopy using a Digilock electronics and software, but with the alternative to be lock to an ultra-stable ULE reference cavity. The locked laser beam is coupled to a fiber that delivers the light with $> 85\%$ coupling efficiency of ~ 40 mW laser power of the D-lasers in a second optical breadboard, these beams seed an initial Tapered Amplifier (TA)⁸. The TA amplifies 5mW up to 2W, which splits the beam later into three branches. Each branch contains a second TA with a SP and a DP using those AOMs to create the required frequencies and amplitudes for the cooler and repump for the 3D MOT beams (3D cooler & 3D repump) and the 2D MOT beams (2D cooler & 2D repump), the pusher beam in the 2D MOT as well as side and vertical absorption imaging beams (see fig. 4.7). All of these beams are coupled into optical fibers, transporting the light to 2D and 3D MOT distribution boxes where beams for different purposes are mixed and then split into the different shared beam paths that later go to the science chamber and the 2D MOT.

²Kimball Physics; 8 Multi-CF Spherical Square MCF800-SphSq-G2E4C4A16.

³Agilent Technologies; VacIon Plus 150 Ion Pump StarCell.

⁴Agilent Technologies; TSP Cartridge Filament Source

⁵Steel code 316LN. The viewports are manufactured by UKAEA out of Heraeus Suprasil 3001. They were anti-reflection coated for 460nm (two-photon excitation to the Rydberg state in combination with 767nm light), 575nm (optional two-photon single-wavelength excitation, not used in this thesis), 767nm (MOT and imaging light), 1064nm (high-power dipole traps).

⁶Asphericon A45-32 HPX. Aspheric lens with numerical aperture 0.61 and 32mm focal length. The planar side faces the atomic cloud and was coated with an anti-reflection coating for 767nm combined with an indium thin oxide (ITO) coating as the last layer (Evaporated Coating Inc. #939). This increases the electric conductivity to prevent the build-up of space charges on the glass surface as Rydberg atoms are sensitive to electric fields. The convex side was only anti-reflection coated (Evaporated Coating Inc. #6408) alone. The transmission of the ITO is specified as $<98\%$. In combination with an external lens, the combined optical array has a 47 fold design magnification.

⁷Hamamatsu F1551-21S.

⁸Based on TA chips from Eagleyard (EYP-TPA-0780-01000-3006-CMT03-0000).

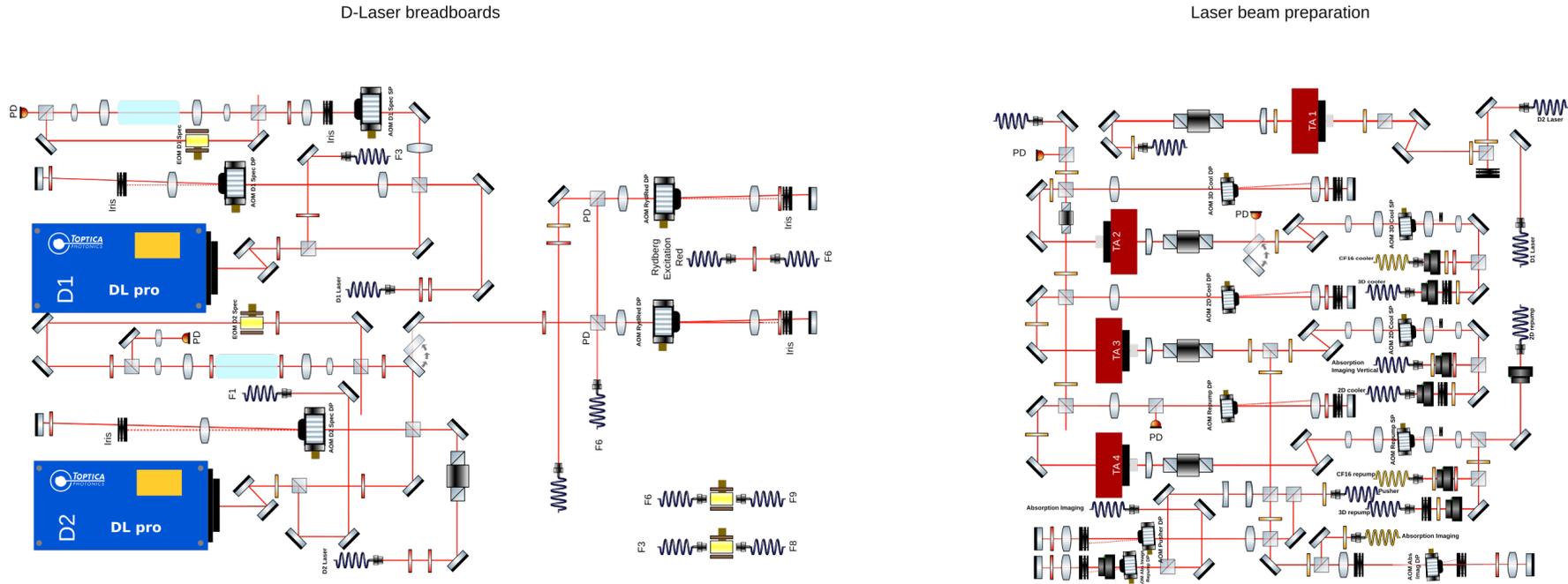


Figure 4.3: **D-lasers breadboards.** Breadboard setup for laser beam used for the generation of 2D-MOT, 3D-MOT, absorption (top and side) and fluorescence imaging.

A full scheme of the optical setup with the frequencies and DP and SP configuration can be visualized in the scheme in appendix D.

In contrast, the electronics of the backend can be just as complex as the optical setup, see fig. 4.4. Some of these electronics fall into the category of “supporting devices”, meaning they do not change dynamically during the experimental run. Examples of such passive electronics include laser locking systems, power supplies, and monitoring oscilloscopes. These can be further automated but it is not necessary for the functioning of quantum computers and simulators. There is another classification of electronics denoted as “active devices” which are usually fully controlled during the experimental run. The active electronics can be separated in the main controller, the AOM drivers, the shutter drivers, the DDS, the microwave source and the AWG. These are discussed below.

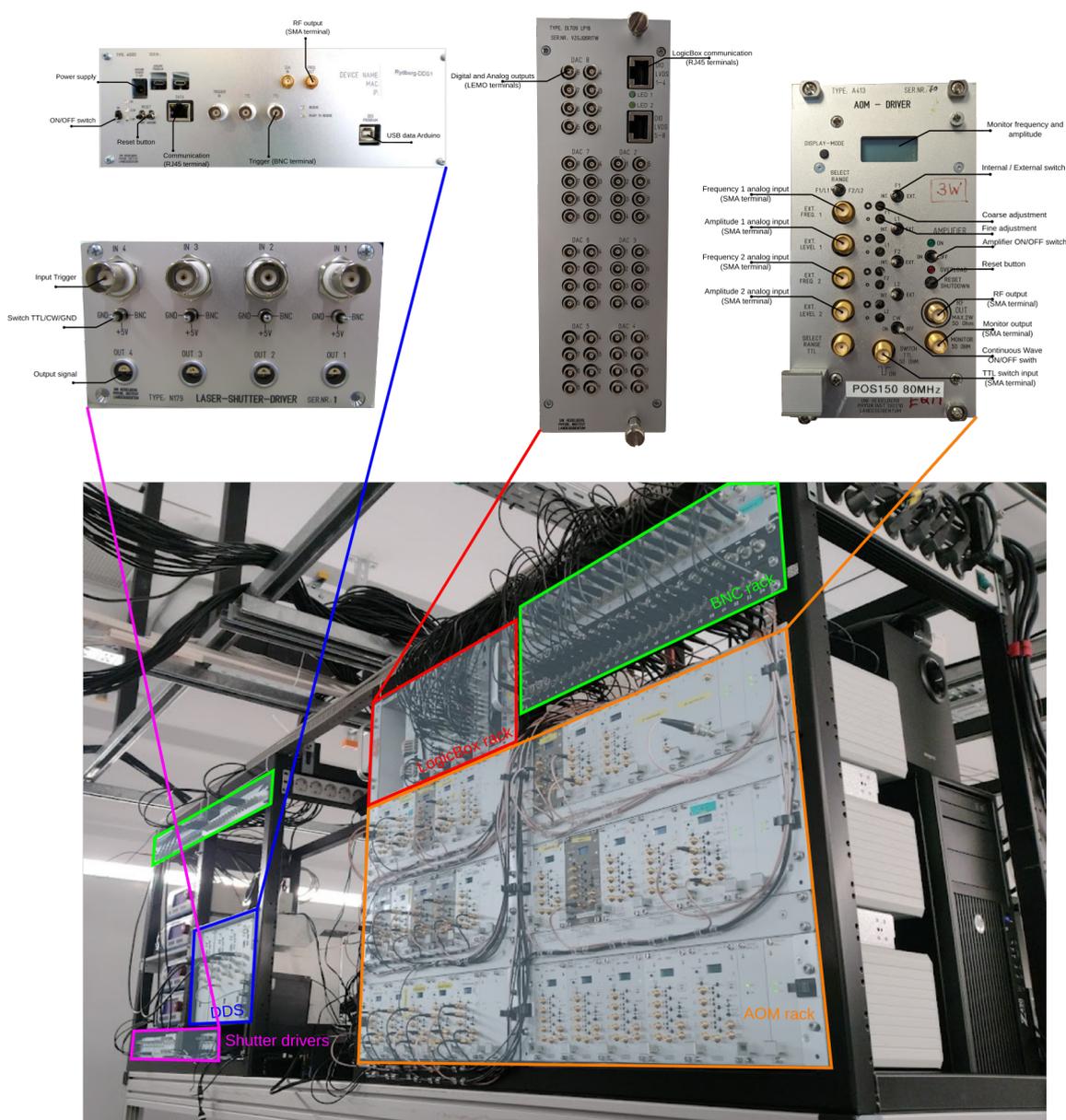


Figure 4.4: **Control electronics backend setup.** Electronics for backend control with main controller (LogicBox), AOM drivers, DDS, shutter drivers, connected to different parts of the experiment with BNC rack bridges.

- Main controller

Currently we are using a FPGA based controller (Xilinx Spartan 3) called LogicBox (DL-709) developed in the Universitaet Heidelberg Electronic workshop [PIH]. It contains a group of submodules with: 20 ports for digital pulses LVDS-TTL SU707, 8 ports for analog inputs Analog-Digital-Converter (ADC) SU721 and 8 ports for analog outputs Digital-Analog-Converter (DAC) SU741-2. The LogicBox has a 10ns time-resolution and 14-bit vertical resolution. We use 4 logicboxes where one of them is set as the master and the other three as slaves, including one LogicBox with two 40-pin connectors for controlling electro-mechanical translation stages ⁹.

Besides having DACs to generate some analog signals, the device does not have capabilities for arbitrary waveform generation, nor the bandwidth to generate RF signals capable of driving our AOMs and EOMs. This is critical for a fully controllable and flexible system. Therefore we use another device to drive these components using analog and digital signals from the LogicBoxes' channels. See the red box in fig. 4.4.

- AOM drivers

In order to drive our AOMs we need an RF signal with $\sim 2W$ power. Our AOM drivers fulfill the power requirement with rack U3 card based on a Voltage-Controller-Oscillator (VCOs) and an additional amplifier. It takes 2 analog inputs to control the frequency and amplitude up to 5V; one digital input (TTL) to activate/deactivate and one RF output.

These devices work as slaves of the main controllers in order to drive the AOMs of the optical setup. We have stacked them with air cooled stages as shown in the orange box of fig. 4.4.

- Shutter drivers

Custom-made mechanical shutters are controlled with digital inputs (< 5 Volts) generated by the shutter driver with a typical rise time of 2ms and fall time of 3.5ms. These drivers (purple box fig. 4.4) are connected to the main controller via a BNC bridge (green box fig. 4.4).

- Direct digital synthesis

For the generation of RF and low-frequency MW signals ($< 2GHz$) the DDS design implements a DDS board (Model) controlled with an Arduino DUE program separately with 9.52ns resolution and 12-bit resolution. The DDS has clock input and external triggering capabilities and connects to the experimental software via analog and digital driving from LogicBoxes and is connected to the network via ethernet.

- Microwave source

Transitions between hyperfine ground-states are achievable with fields within 2GHz carrier frequency, which is possible to achieve with DDSs and AWGs. Meanwhile transitions in the Rydberg manifold can be from a few gigahertz up to $\sim 100GHz$.

⁹FESTO Axis model S/N: EGC-50-120-TB-10H-KF-GK. Motor S/N: EMMS-ST-57-S-SE-G2. Controller S/N: CMMO-ST-C5-1-DIOP (1512316). Power Supply S/N: CACN-3A-1-10. Axial-Set S/N: EAMM-A-L27-57A.

A microwave source used in the experiment with a large bandwidth from 9KHz up to 40GHz¹⁰, with capabilities for external clocking and triggering. More details concerning the design and calibration can be found at the end of this chapter (see section 4.3.1).

- Arbitrary waveform generation

In order to control the amplitude, frequency and phase of the waveforms for AOM drivers and signal mixing with higher frequency signal generators, AWGs play a key role in generating any envelope shape signal i.e., arbitrary waveform. As shown before (see chapter 3), I installed a device called OPX from the company Quantum Machines which assumes the full control of the experiment, and it also presents these capabilities. Additionally, we also make use of rack based AWGs originally designed for the Microwave setup (see more details in the next section) which also posses a bandwidth of $< 500\text{MHz}$ at 16-bit vertical resolution and 2ns time resolution (i.e., 500MSa/s) with possibilities of external clocking and triggering¹¹. Additionally we use a 19-inch rack chassis¹², that allows to stack a series of AWGs and keep them synchronized and control by a main computer¹³ in the rack. The channels of the AWG can be amplified and used directly over optical modulators.

With this set of devices it is then possible to prepare atoms and start the manipulation.

4.1.3 Atomic initialization

Within this section, I delve into the laser system's pivotal role in achieving low temperatures and high atomic densities. Its primary function is to effectively load an optical trap, serving as substrate for arrays of atomic ensembles and single atoms. The laser cooling approach, encompassing 2D MOT, 3D MOT, compressed MOT, and D1 gray molasses, enables the attainment of temperatures around $\sim \mu\text{K}$ for the atomic sample, accompanied by high atomic densities reaching 10^{11}cm^{-3} .

Beyond its cooling applications, the laser system serves additional purposes such as optical pumping, Rydberg excitation and readout by imaging through saturated absorption and fluorescence. The system accommodates for bosonic ^{39}K .

A 2-dimensional Magneto-Optical-Trap (2D MOT) design setup comprises of a glass cell equipped to house up to three dispensers, with one dispenser subjected to an electrical DC current of 1.9A for controlled heating (see fig. 4.5). Potassium atoms released from this heated dispenser undergoes a cooling and trapping process facilitated by two orthogonal pairs of four beams, complemented by two magnetic coil pairs that generate a gradient field. This meticulous arrangement serves to reduce velocity components in the two directions perpendicular to the symmetry axis of the glass cell, resulting in the formation of an atomic beam directed into the science chamber through a dedicated differential pumping tube for the 3D MOT loading.

An additional near-resonant laser beam, known as the “pusher beam”, is strategically aligned along this axis, effectively propelling the cooled atoms toward the science chamber. To ensure optimal conditions within the 2D MOT setup, the glass cell is intricately

¹⁰Keysight - N5173B EXG X-Series Microwave Analog Signal Generator.

¹¹Keysight - M3201A PXIe Arbitrary Waveform Generator, 500MSa/s, 16 bit.

¹²Keysight - M9010A PXIe Chassis: 10-slot, 3U, 24GB/s, Gen 3.

¹³Keysight - M9037A PXIe Embedded Controller: Intel i7, 4GB RAM, 240GB SSD.

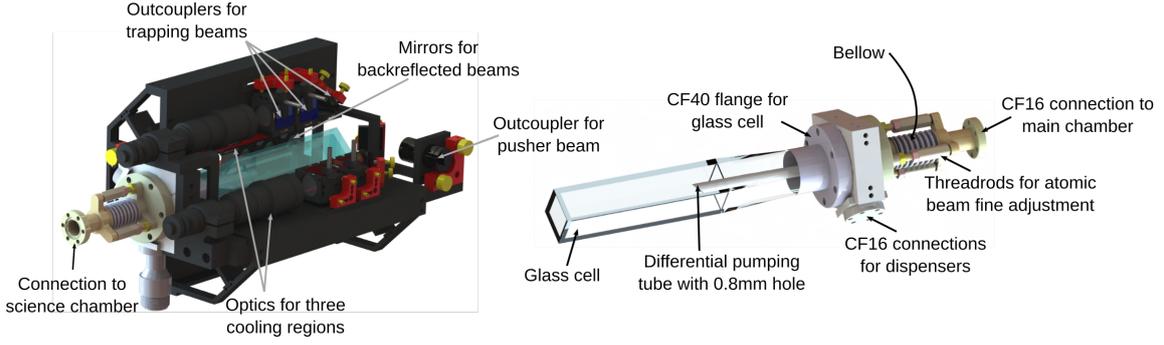


Figure 4.5: **2D MOT.** Elements of the 2D MOT setup (left) and the internal elements of the glass cell containing the atoms (right) before being push by beam inside the science chamber to generate the 3D MOT.

connected to an additional ion pump exclusively dedicated to maintaining the desired environmental parameters.

A 3-dimensional Magneto-optical-Trap (3D MOT) is achieved using three orthogonal pairs of counter-propagating beams with circular polarized light (see fig. 4.7), complemented by a gradient magnetic field in a quadrupole configuration. In potassium-39 the narrow hyperfine splitting within the $4P_{3/2}$ manifold imposes a limitation on the achievable temperature, restricting it to approximately $\sim 500\mu\text{K}$. In other words, atoms moving out of the trap will couple to a cooling resonance transition that pushes the atom toward the center of the MOT (i.e., zero magnetic field point) due to the scattering of this transition ($|4S_{1/2}, F = 2\rangle \leftrightarrow |4P_{3/2}, F = 3\rangle$ for D2 MOT cooling and $|4S_{1/2}, F = 2\rangle \leftrightarrow |4P_{1/2}, F = 2\rangle$ for additional D1 gray molasses). A subsequent cooling stage is implemented using Λ -enhanced gray molasses on the D1 line ($4P_{1/2}$). This cooling stage employs the same beam path as the 3D MOT beams and reaches temperatures close to $10\mu\text{K}$.

Afterwards we compress the 3D MOT by increasing the magnetic field gradient with 200A^{14} , from $7\text{G}/\text{cm}$ to $27\text{G}/\text{cm}$ (with a conversion factor of $9.8\text{G}/\text{cmV}$ for the hollow-core coils)¹⁵. It is also possible to fine tune the magnetic field by changing the current in smaller coils located around the CF63 viewports in the x and y directions¹⁶, the compensation factors of the gradient field are: $0.73\text{G}/\text{cmV}$, $0.73\text{G}/\text{cmV}$ and $1.9\text{G}/\text{cmV}$ for the x, y and z directions. This stage is what we call the 3D MOT compress or cMOT. Overlapping the cMOT with a far-of-resonance ‘‘Pancake’’ dipole trap we obtain a 2-dimensional atom sheet with atomic densities of 10^5cm^{-3} (flat horizontal beam, fig. 4.7). A D1 gray molasses is applied to reduce the temperature up to $10\mu\text{K}$ and confine the atoms in a quasi-flat region i.e., 2D-plane. It earns its name from its geometry, characterized by a vertical dimension noticeably narrower than the two horizontal axes.

As described in previous group theses [Win21b], [WIN16], the pancake trap geometry setup is generated by an incoming 1064nm beam, harnessed from a Mephisto MOPA laser based in Nd:YAG crystal resonator that generates approximately 55W . The beam counts with a high-power single-pass AOM for precise control of the intensity. This beam starts

¹⁴Delta Elektronika SM6000 SM 30-200 supplying up to 200A and 30V .

¹⁵One coil consists of two identical pancakes, where each one consists of two layers with 12 windings each. The inner (outer) diameter of the windings is 19cm (29.6cm).

¹⁶Each x direction coil is made out of 10 layers of 2 windings with an inner diameter of 120mm . Each y direction coil is made out of 3 layers of 10 windings with an inner diameter of 192mm .

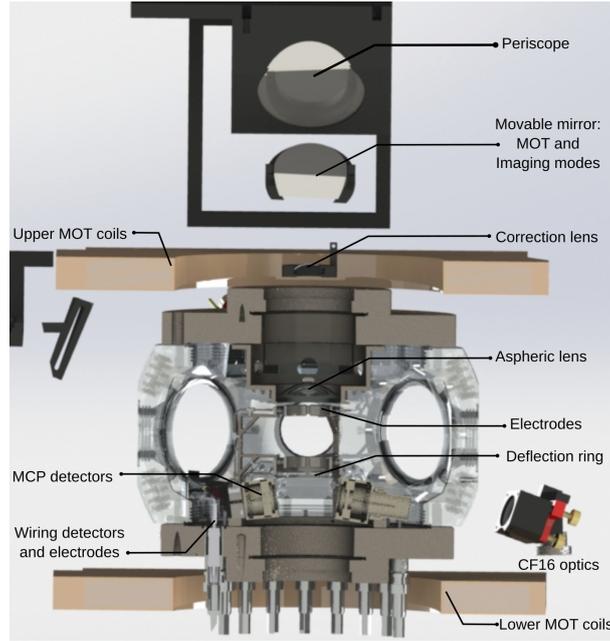


Figure 4.6: **Science chamber cross-section.** Chamber hosting the 3D MOT with CF16 beams used for inducing collisions and driving transitions for fluorescence detection. Objective aspheric lens, electrodes for electric-field ionization and MCP detectors or ionized Rydberg states.

with a vertical expansion facilitated by a concave cylindrical lens ($f_z = -22.7\text{mm}$). Upon encountering the first of the two spherical doublet lenses ($f_s = 202.8\text{mm}$), the beam is skillfully recollimated along the z -axis. Subsequently, a second concave cylindrical lens, orthogonal to the first and featuring a focal length of $f_x = -25.9\text{mm}$, takes center stage along the x -axis, resulting in a pronounced elongation of the beam in the horizontal direction. This deliberate configuration produces a tightly focused vertical beam after traversing the second spherical lens (f_s), with the waist along the x -axis exhibiting a broader profile $\sigma_x = 29\mu\text{m}$ and $\sigma_y = 75\mu\text{m}$. This becomes particularly significant when we decrease the trap power for evaporative cooling, bringing the dimensions close to or below the resolution limit of the imaging system. This leads to an atomic cloud with an atomic density of $2 \times 10^5\text{cm}^{-3}$ at a temperature of $20\mu\text{K}$.

4.1.4 Atomic substrate preparation

After the atoms are loaded into the Pancake dipole trap a structured light is imprinted from the top of the science chamber into the atomic cloud, being focused over the plane defined by the atoms trapped in the pancake by an aspheric lens.

Absorption imaging of the pancake dipole trap is shown in fig. 4.8.

A DMD¹⁷ provides an arbitrary array of optical tweezers (i.e., structured light) which is programmable by loading real-space images, however it is possible to use the Fourier space. Recent work with other techniques to generate large arrays of static tweezer arrays using microlens arrays (MLA) combined with SLM has shown outstanding results in

¹⁷Vialux V-7001 (micromirror array: 1024×768 , micromirror pitch: $13.68 \mu\text{m}$. Active mirror-array area: $14 \times 10.5 \text{mm}^2$. Damage threshold: 25W/cm^2 . Fill factor: 92 %). CAD models provided by the manufacturer.

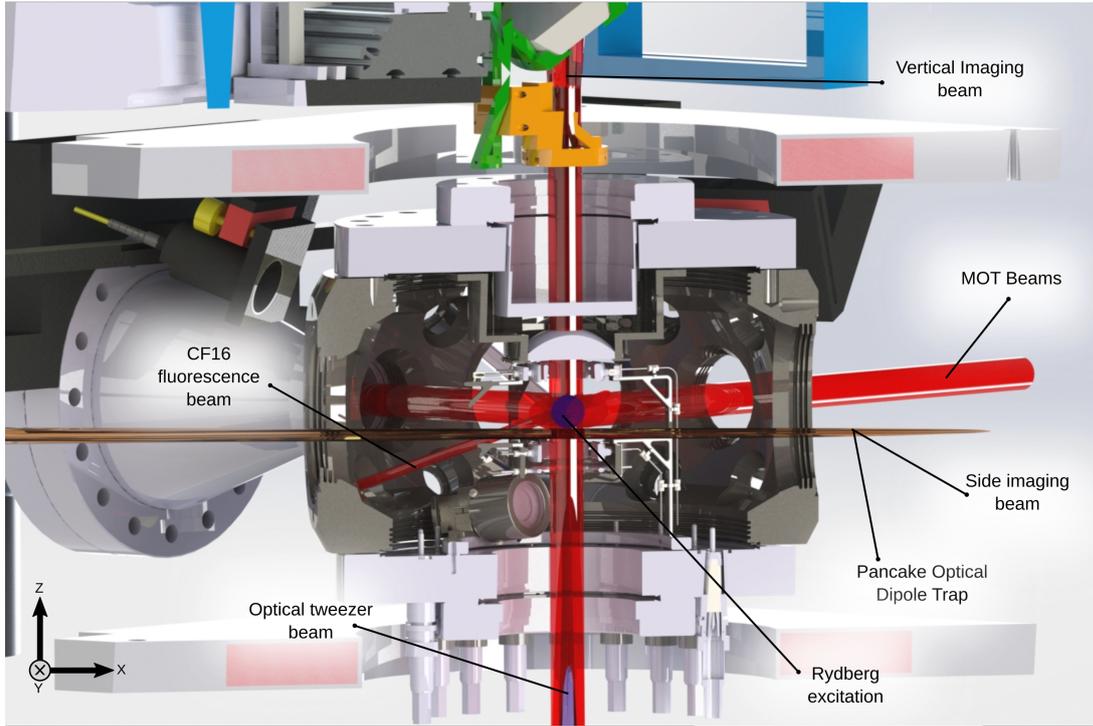


Figure 4.7: **Beams in the science chamber.** 3D version of beams for the 3D MOT, 2D MOT, CF16 beams and optical tweezers inside the science chamber. Additionally, the structures of the deflection ring and electrodes for directing ionized atoms to the MCP detectors are shown.

terms of the scalability in the number of tweezers [Pau24].

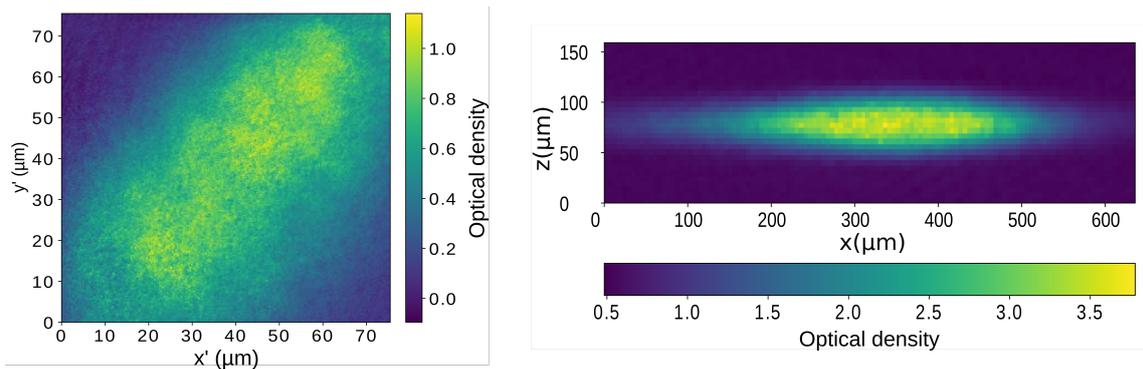


Figure 4.8: **Pancake dipole trap.** Absorption image from top view of the dipole “pancake” trap with normalized optical density (left). Absorption imaging of the side imaging of the atomic cloud after a time of flight (right).

Our DMD consist of a square array of 1024×768 pixels, where each micromirror has an area of $12.6 \times 12.6 \mu\text{m}^2$. The micromirrors are mounted in a yoke that stands in a torsion hinge, that can tilt $\pm 12^\circ$ along the diagonal axis of the micromirror¹⁸, helped by

¹⁸A 45° inclination should be taken into account when installing the device in order to align the rotation axis with the vertical axis and pixel are in a non-rotated frame.

a pair of electrodes that tune the configuration. These mechanism allow to configure and retain the pixel value as a binary ON or OFF in a integrated CMOS memory until the next updated value. The refreshing rate of this value can vary between 4 to 32 kilohertz, depending on the model. The images (i.e., values of the pixel array) are loaded into the memory (RAM) of the controller, where the firmware stored in the FPGA contains the fundamental instructions to manipulate and load the images and set the values in the device. There are two options for interfacing the DMD that communicates with a computer via USB port: one is via a program with a graphical user interface¹⁹ or with the Python driver wrapper²⁰, the integrated driver including instructions for the generation of the real-space images for the DMD is included in AQiPT. This driver is composed uses the low-level instructions from ALP-4.1 while using high-level commands to define the geometry of the atomic array or pattern that will be imprinted on top of the atomic cloud.

The DMD surface behaves as a 2-dimensional blazed grating, reflecting in many orders the incident beam, following the expression:

$$2d(\sin(\theta_B)) = m\lambda, \quad (4.1)$$

where d is the pitch between pixels (micromirrors), λ is the wavelength of the incident beam and θ_B is the angle between the mirror's normal and the chip normal (see fig. 4.9). The expression can be written in terms of the incident and diffracted beam angle i.e., θ_{in} and θ_{out} by the using the relation: $\sin(\theta_B) = [\sin(\theta_{in}) + \sin(\theta_{out})] / 2$.

Initial experiments used a 780nm diode laser output which is amplified by a TA and is then coupled into an optical fiber to create a well-defined Gaussian mode (see fig. 4.13). The 780nm wavelength has been chosen as a compromise that involves a balance between minimizing off-resonant scattering rates and maximizing trap depth at a constant power. Additionally, this wavelengths proximity to the D1 and D2 wavelengths match the optics of the high-resolution setup for tweezer and imaging. The DMD reflects approximately 85% of 400mW of collimated light after the fiber. Depending on the micromirror state, it reflects the incoming light either toward (white pixels) or away from (black pixels) the high-resolution imaging setup. An illustrative DMD pattern in fig. 4.9 demonstrates a nonlinear adjustment in pattern spot sizes to correct for the Gaussian envelope, enhancing uniform microtrap filling.

The reflected tweezer trap light then traverses the high-resolution imaging system. As was shown before in fig. 4.6, it is composed of a large focal length lens and an aspherical objective lens within the vacuum chamber. This process effectively generates a demagnified image of the DMD pattern onto the atoms confined in the pancake trap. After 200ms of overlap time, the atoms are loaded from the pancake trap into the optical tweezers. Subsequently the pancake trap intensity is ramped down to zero, the pancake is deactivated leaving the atoms trapped in the tweezers. Read-out of the tweezer arrays is feasible through high-resolution fluorescence or absorption imaging along the z -axis. A dichroic mirror²¹ separates the imaging light (767nm or 770nm) from the tweezer trap light at 780nm.

¹⁹EasyProject.

²⁰Suite ALP-4.1. driver and controller firmware as well as ViALUX' FPGA logic design for the Virtex 5 on the EVM board.

²¹Edmund optics: FILTER 769NM X 41NM BP 93T 25.0D.

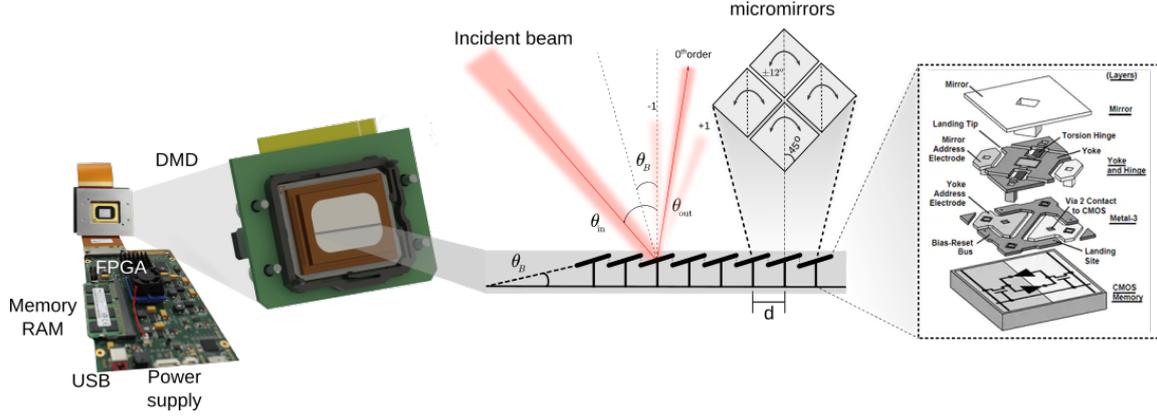


Figure 4.9: **Digital-Micromirror-Device (DMD).**

The imaging of the optical tweezers are described in fig. 4.11, where the effective DMD pixel size (L_{chip} , i.e., including filling factor and micromirror pitch) is demagnified 53 times at the atom plane (AP):

$$L_{AP} = \frac{L_{\text{chip}} \times \kappa}{M}. \quad (4.2)$$

where κ is the fill factor and M is the magnification. For the filling factor of our DMD model $\kappa = 0.92$ the pixel size at the atom plane is 245nm. Where the resolution (ΔL) as obtained from the Rayleigh criteria is:

$$\Delta L = 0.61 \times \left(\frac{\lambda}{\text{NA}} \right). \quad (4.3)$$

When using the numerical aperture and a wavelength of 780nm the resolution is $0.79\mu\text{m}$, compared to the observed cloud sizes in arrays and single-atoms it represents 1/3 of the radial size.

4.1.5 Manipulation and calibration

Lasers and free-space fields e.g., radio-frequency and microwave, are the tools used for manipulation of the atomic states of the atoms in the tweezer arrays. Two types of manipulation are highly relevant to this work: creation of Rydberg states and coupling between Rydberg states.

For the first one we prepare our atoms in a well-defined hyperfine ground-state by using circular polarized light for the cooler and linear polarization for the repumper. This prepares the state in a $|4S_{1/2}F = 2, m_F = 2\rangle$ which is later coupled with a 766.7nm laser to the $|4P_{3/2}F = 3, m_F = 3\rangle$ and then excited to a Rydberg state $|77P_{1/2}, m_j = 1/2\rangle$, using a 455.63nm laser light. This type of control would be key in the consideration of ground and ground-ground qubit encoding. When the atom is in the Rydberg manifold the wavelength of the transitions between states are in the microwave regime. This means that it is no longer possible to use optical transitions made by laser beams, here microwave fields emitted by high-gain polarized antennas are needed. In the experiment the antenna design used is an aluminum commercial linear polarized horn antenna with a large bandwidth (4-40)GHz. The control with microwaves is critical when using ground-ground qubit encoding. A new design of microwave control made along my thesis is detailed later in the chapter, see section 4.3.1.

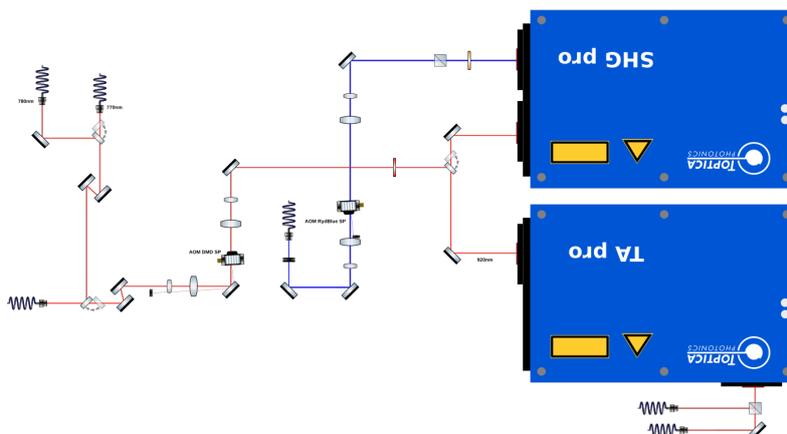


Figure 4.10: **Optical setup preparing 455.63nm and 920nm beams.** Optics arrangement for the generation of a blue beam for Rydberg excitation from a second harmonic generation Toptica module from a 920nm seed laser beam. Additional 770nm and 767nm beams are used for alignment purposes.

4.1.6 Measurement and readout

In our setup and during the development of this work, we used several ways to measure the state of the atoms. The first type of readout is via imaging, there are two possible techniques that can be used with imaging: **saturated absorption imaging** and **fluorescence imaging**.

In **saturated absorption imaging** we shine the atoms with resonant light, that, as the name of the technique suggests, is absorbed by the medium at the center of the Gaussian beam. This method typically requires more than one atom in order to observe the shadow due to the relatively small resonant absorption cross-section $3\lambda^2/2\pi$, which requires multiple consecutive shots for obtaining the final imaging. In our backend it is possible to realize absorption imaging from the side of the science chamber using one of the CF40 partially sharing the path with the pancake trap with a dichroic mirror²² and a CMOS camera²³ and from the top of the science chamber, sharing the path with the tweezers light with EMCCD camera²⁴, see fig. 4.11.

The second imaging technique, **fluorescence imaging**, consists in shining near-resonant light to the atoms that scatter photons out from fast-decaying excited states, which are collected by the aspheric lens in the path of the EMCCD camera.

Besides imaging, the second type of readout is by **electric-field ionization**. Eight electrodes are located close to the atom plane in a set of 4 above and 4 below the plane, surrounding the atom. The electrodes are connected pair-wise orthogonally to each pair and crossing from bottom to top. An additional deflection ring is located below the lower set of electrodes to correct and deflect the trajectory of ionized atoms to successfully reach

²²Thorlabs DMLP950L.

²³IDS UI-3240ML-NIR.

²⁴Andor iXon Ultra 897 UCS-EXF. CAD models provided by the manufacturer.

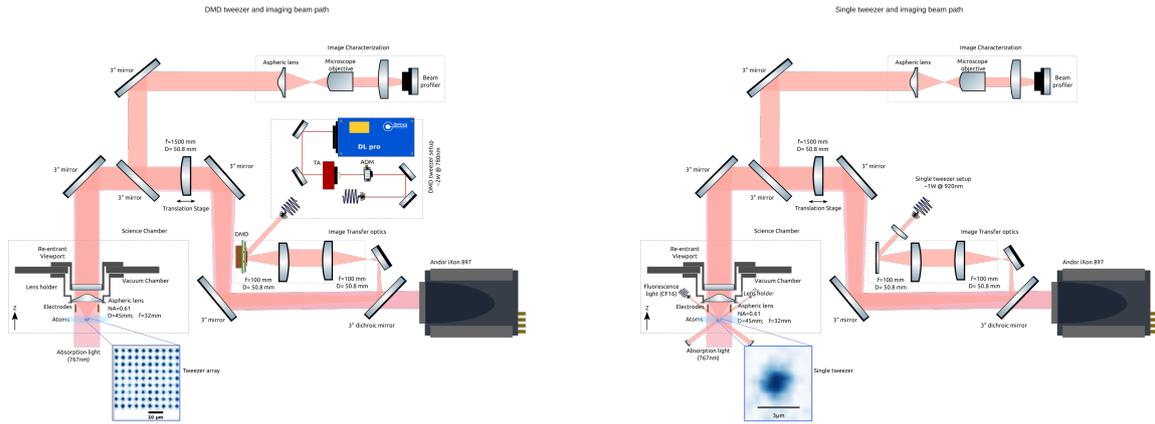


Figure 4.11: **Optical setup for imaging and tweezers.** Optical setup for single tweezer projected in the science chamber with imaging beam path and auxiliary imaging setup for tests (left). Similar setup for tweezer arrays in the real plane, substituting a reflectable mirror by a DMD (right).

the MCP detector²⁵ which signal is amplified²⁶ and acquired by a fast oscilloscope²⁷. The electrodes and deflection rings are simultaneously activated for a detection time. The atoms are ionized with a sudden strong electric field change with approximately $\pm 2.5\text{kV}$ using very fast switches²⁸ from the compensation values of the electric field set by DC source²⁹ (see details in fig. 4.12).

4.2 Quantum registers from structured light

Quantum registers are the core of quantum computers and simulators. Their geometry and properties vary depending on the platform and the architecture approach. In our backend we studied two approaches: single atoms and small atomic ensembles trapped in optical tweezers. For the first we constrained to a single tweezer, for the second we used the largest array possible to generate a register with hundreds of small ensembles trapped with tweezers. Ideally, a quantum computer and simulator should count with the possibility of generating this large arrays of optical tweezers containing single atoms. But due to the low diffraction efficiency that increases the requirement of laser beam power for the structured light we constrained the scope of this thesis to study the methodology for preparing single atoms in optical tweezers and exploring technical features and challenges for using a DMD for generating a large array.

Below I present the technical aspects, properties and components of the backend for this study, in chapter 5 I address the the details and analysis of the findings.

4.2.1 Single atom loading in an optical tweezer

The loading of single atoms in optical tweezers has become a traditional approach for quantum computers and simulator backends, with demonstrations with alkali atoms such

²⁵Hamamatsu F1551-21S.

²⁶Mini-circuits ZFL-500LN+.

²⁷TeledyneLeCroy WaveRunner 8254M-MS with up to 4GHz bandwidth and 40GS/s sampling rate.

²⁸Behlke HV switches.

²⁹Omega OM-USB-3103, 8-channel, 16-bit analog voltage, output range $\pm 10\text{V}$.

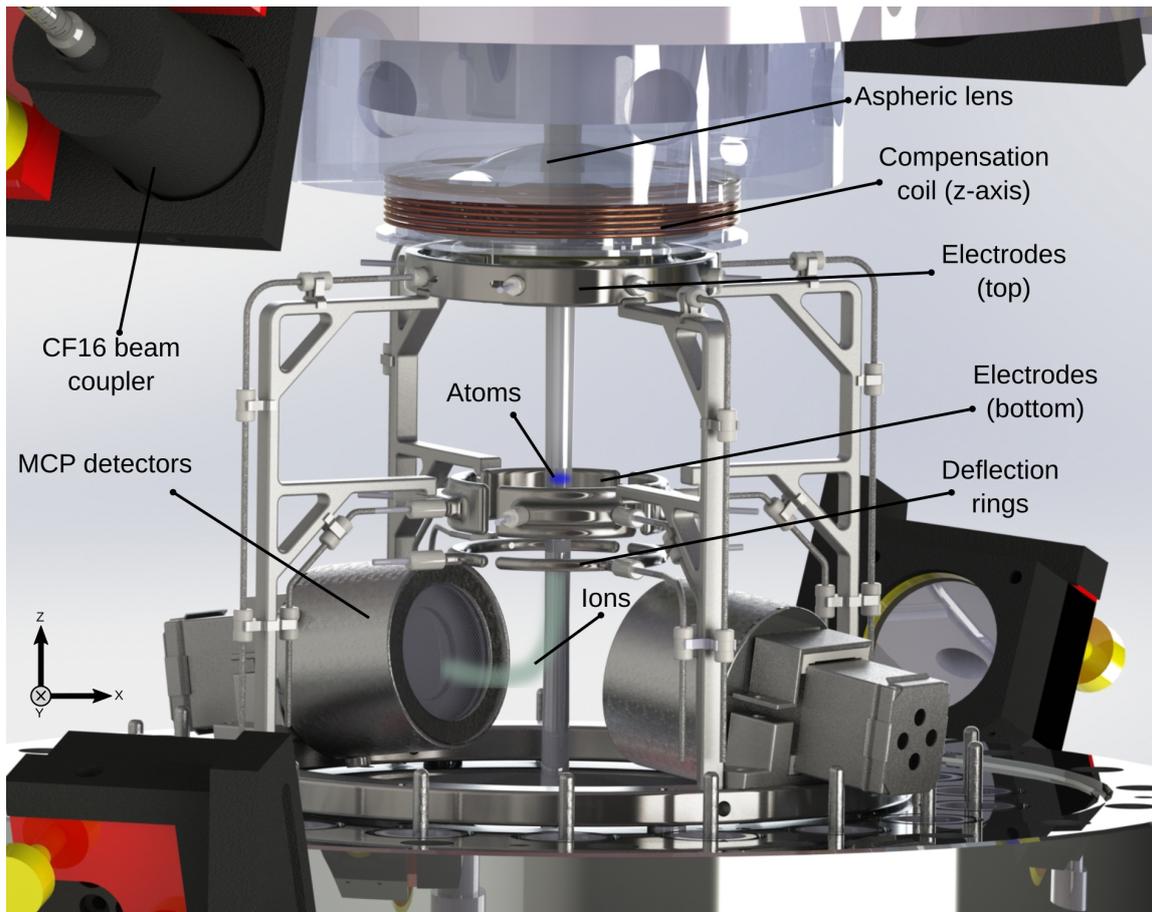


Figure 4.12: **Ion detection hardware.** Fast switching to strong electric fields by upper and lower electrodes ionizing and canalizing the ion to the MCP detector, helped by deflection rings.

as rubidium and cesium, but also with alkaline-earth-like atoms such as ytterbium and strontium. The choice for potassium-39 carries some challenges since cooling schemes can be a little more complex due to the narrow hyperfine splitting in the excited state, but when they are overcome, local light-shifts generated by the polarization gradient can be used to change the differential splitting bringing additional tools for coherent control of gg-qubits with high-fidelity.

For preparing single atoms in an optical tweezer we set a 920nm laser beam³⁰ from the Rydberg excitation laser (455.63nm), with a SP configuration for intensity control and coupled to a single mode optical fiber³¹ (see fig. 4.10). After being expanded the beam is passed through a $4f$ configuration with two lenses of 100mm focus length and one lens with 1500mm focus length to collimate the beam previous to the aspheric lens, see scheme details in fig. 4.11 (right).

For cooling the atom (i.e., gray molasses) and realizing fluorescence imaging a set of 3 CF16 back reflected couplers are disposed with equal intensity power for cooler and repumper beams (i.e., TA2 and TA4 branches in fig. 4.3) of the D1 transitions after passing through an integrated optics splitter.

Due to the small scattering rate of the D1 transitions a full cover of the optical paths

³⁰Toptica TA-pro.

³¹NKT Photonics. LMA-PM-15. Single mode 15 μ m polarization-maintaining fiber.

was necessary to reduce noise in the EMCCD camera. Also, in order to avoid too many internal reflections in the science chamber only one CF16 was selected based on the background noise. To the taken images a background subtraction and fringe removal algorithm were applied (see chapter 5).

4.2.2 Atomic ensemble loading in an array of optical tweezers

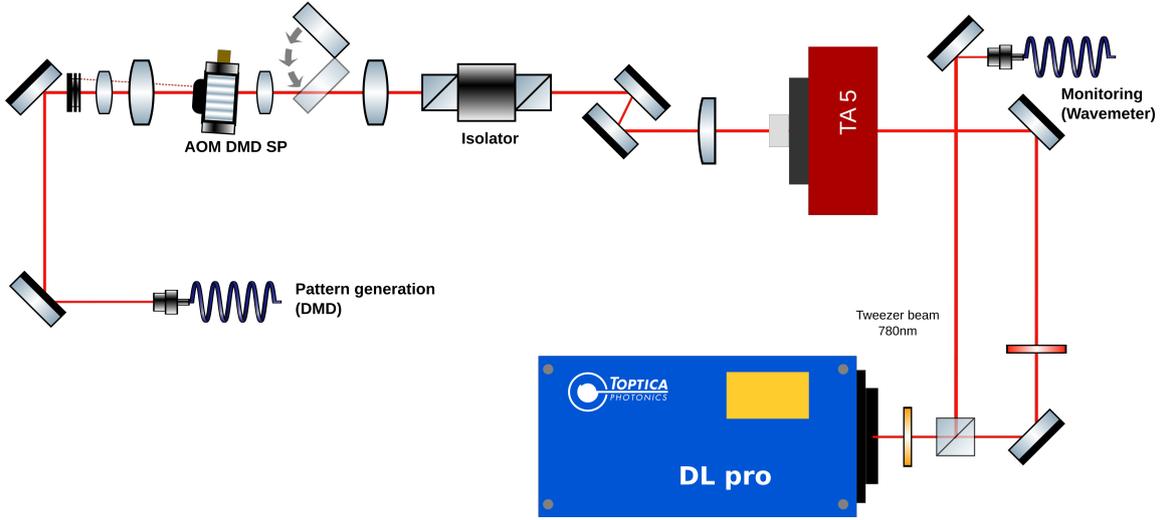


Figure 4.13: **Optical setup for tweezer array with DMD and 780nm.** Bread-board setup for laser beam used for the generation of 780nm tweezer array, illuminating a DMD.

In order to generate arrays of optical tweezers, we designed the optical setup for the ECDL Laser beam³² preparation to illuminate the DMD (see fig. 4.13) with a SP configuration to control the intensity. The output beam couples to a single mode optical fiber that creates an incident Gaussian beam over the arranged mirrors of the DMD chip. The light is structured and passed through a $4f$ configuration similar to the single tweezer and collimated on the aspheric lens, which focuses the optical tweezers at the atom plane, overlapping the focus point with the center of the pancake trap (see fig. 4.11 (left)).

The preparation of the real-space images to be displayed by the DMD are generated as PNG file (i.e., .png extension) with the correct bit-depth and then loaded to the memory of the DMD using the software EasyProject. However, a newer and flexible driver is implemented in AQiPT, allowing more modular realization of the real-space images and connecting directly to the device from Python, using the ALP 4.3 firmware wrapper.

Patterns were characterized using an auxiliary imaging path to a Beam profiler³³. Additional compensation of the beam profile to produce uniform trap-depths across the arrays of optical tweezers was implemented by using an inverted Gaussian dependency site size in terms of the number of pixels i.e., $A_m = A_{min}e^{gr_m^2}$, where A_m is the m -th profile Gaussian compensation at a distance r_m , A_{min} the center site number of pixels with non-effective compensation and g is the compensation parameter (see fig. 4.14).

³²Toptica DL-pro.

³³LBP2-HR-VIS2: array 1928*1448, pixel size 3.69 μ m, sensor size 7.1*5.3mm, resolution 37 μ m. ~10 pixels.

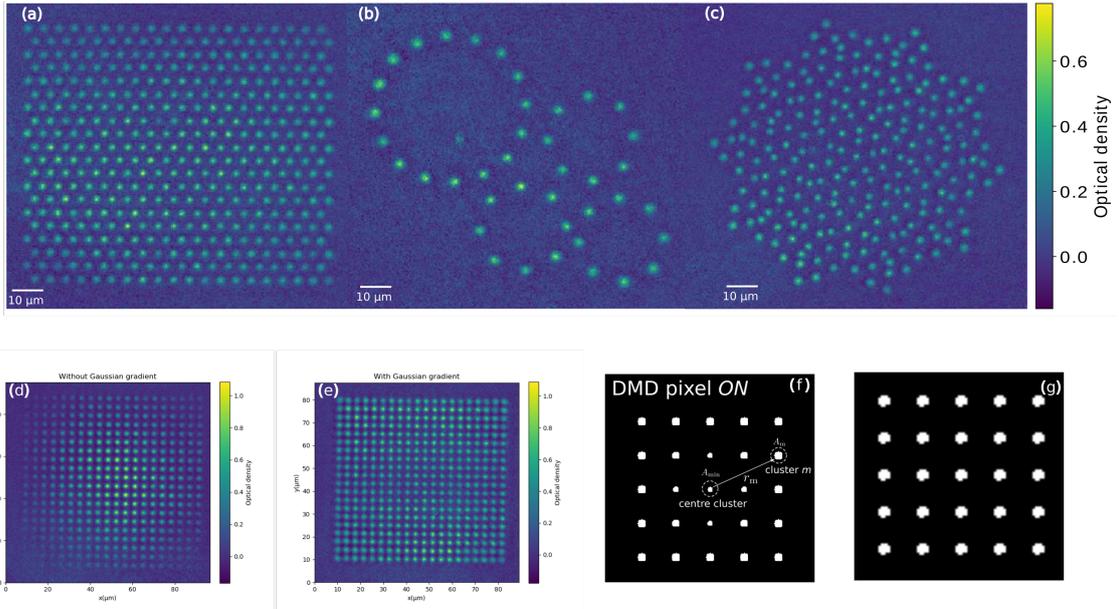


Figure 4.14: **DMD pattern compensation** DMD pattern compensation. (a-c) Average over 20 images of different geometries of arrays of atomic ensembles filled with $\lesssim 36$ potassium-39 atoms: 400-sites triangular lattice with $4\mu\text{m}$ spacing, 40-sites ring structure and 226-sites Penrose quasicrystal lattice. Pattern (d) without Gaussian wavefront compensation and (e) with Gaussian wavefront compensation over a 400-sites square array and (f) adaptation of the raw image given to the DMD according to the distance from the center of illumination to the m -th site of an initially uniform pattern (g). Figures (a-c) adapted from [Wan20] and (d-g) adapted from the group thesis [She21].

4.3 Coherent control

Global addressing of qubits is an important tool for generating single qubit rotations or gates simultaneously over the quantum register. When using atoms in the Rydberg state, this type of control comes naturally. When assisted by local addressing e.g., 2D AOD setup, generating a light-shift that can detune the microwave transition it is possible to have a large number of these single qubit or even two qubit operations by using global control. The use of microwave technology over superconducting qubits is very advanced due its large use and development in other fields such as: communication networks, radar, navigation, among others. The technology has a large catalog of techniques for quantum computing e.g., IQ-mixing, QED cavities and more.

The Rydberg platform can benefit from this technology as it is also continuously developed for superconducting quantum processors. This collaboration not only opens up diverse applications but also enriches it with unique intrinsic characteristics. For example, it mitigates constraints on frequency separation when allocating a tone for each qubit. Moreover, the heightened sensitivity to electric fields exhibited by Rydberg atoms diminishes the necessity for utilizing excessively large microwave field intensities.

4.3.1 New microwave system

One of the experimental contributions during this work is the design, installation and use of a microwave system for global control of Rydberg state transitions. In the context of quantum computing, this is a fundamental element for using rr-qubit encoding and $XY(\theta)$ gates. Meanwhile, their use can be extended for quantum simulation in order to perform Hamiltonian models in large ensembles (e.g., pancake traps or arrays of atomic ensembles).

Design

Control of qubits typically involves generating arbitrary, flexible, and precise waveforms to modulate the physical operations, closely adhering to Hamiltonian terms. This implies having: amplitude, frequency and phase modulation. A long coherence signal with carrier frequencies in the order of gigahertz for Rydberg states qubit control based on microwaves and of a few gigahertz for single qubit operation between hyperfine levels. Additionally, the design involves a strict budget in the losses of the microwave signal along the system in order to be able to apply very large Rabi couplings between Rydberg states.

The designed system has as a main purpose the generation of large-bandwidth microwave field generator. It consisted in these main elements, all of them with 50 Ω impedance matching and SMA connection by the exception of the microwave source, where we use a 2.92mm to SMA adapter impedance matching³⁴:

- Microwave source.

For the design, I explored options for different manufacturers but they were either short range bandwidth with up to 20GHz or with low power (i.e., below 10dB) or with limited software options. The selected model from Keysight (EXG X-Series Signal Generator N5173B) has a frequency bandwidth from 9kHz to 40GHz with a with a standing-wave-ratio (SWR) from 1.7:1 for low frequencies (< 2 GHz) up to 1.6:1 for the highest frequency band (i.e., > 20 to 40GHz) at 0dB and better performance for high power (> 5 dB) with SWR 1.2:1 and 1.4:1 respectively to each frequency band. The microwave source also have a low single-side-band (SSB) phase noise from -134dBc/Hz at low frequencies (i.e., ~ 200 MHz) up to -91dBc/Hz at 40GHz frequency, which is important to limit the dephasing noise.

The output power falls in the range of 23dBm to 15dBm for frequency band of 9kHz to 3.2GHz and 31.8GHz to 40GHz.

One of the properties of the source is a intrinsic amplitude switching speed with < 2 ms for standard-commands-for-programmable-instruments (SCPI) mode and sweep mode, where it can do frequencies sweeps from list values (3201 points) or defined by command i.e., from frequency 1 to frequency 2 (65535 points). It can be programmed via USB, ethernet or GPIB ports using SCPI commands³⁵. A SCPI-based driver for this source is included in AQiPT.

- Arbitrary waveform generator.

In order to modulate the signal out from the microwave source, we use a low-frequency waveform generator and frequency mixer; however, for maximal arbitrary

³⁴Rosenberger: Adaptor RPC-2.92 JACK – RPC-2.40 JACK.

³⁵Keysight: X-Series Signal Generators SCPI Command Reference.

control and precision I have chosen an AWG with a frequency range of 200MHz which would also be possible to use for drive AOMs in a different application. The selected AWG was an election based in compromise between cost and scalability, in case more channels were required in the experiment, since the Keysight M3201A PXIe Arbitrary Waveform Generator is based in a rack architecture, could in principle scale to 40 analog channels in a single PXIe rack. As mentioned before, this AWG architecture also counts with a PC controller embedded in the rack. The characteristics of this AWG with 500MSa/s give a time-resolution of 2ns with 16-bits vertical resolutions. The four channels in the AWG card can be paired to realize digital IQ modulation at the level of the FPGA, with a frequency band up to 400MHz. This AWG has also a good phase-noise SSB of up to -133dBc/Hz at 10 KHz offset along its frequency range.

Also this AWG has a 2GB onboard memory RAM where the waveform are loaded, either using the graphical user interface or the Python framework (KeysightSD1) from the SD1 3.x software³⁶.

Additionally, we later also used a new generation of AWG with on-the-fly waveform generation and fast-feedback features, called OPX (see more details in chapter 3).

- IQ-mixer for precise modulation

A key element of the design was the IQ-mixer, this element allows to precisely modulate the microwave signal (e.g., local oscillator or LO) by mixing it with two intermediate frequency (IF) signals (see fig. 4.15). A IQ-mixer from Marki has been selected³⁷. This mixer allow us to do side-band modulation, single-side-band up-and down-conversion, in order to manipulate the transitions in the Rydberg state within the frequency band (10-37)GHz. The mixer accepts up to 19dBm for the LO input and 11dBm for the IF inputs. The phase balance of the mixer is 5° being 0° the case of a perfect mixer; this means that the matching the relations between phases between the quadrature signals (I and Q). Similarly the matching in the signal levels given by the amplitude balance is 0.01dB. A high isolation with 37dB between LO, IF and the resulting signal guarantees highly-suppressed leakage noise.

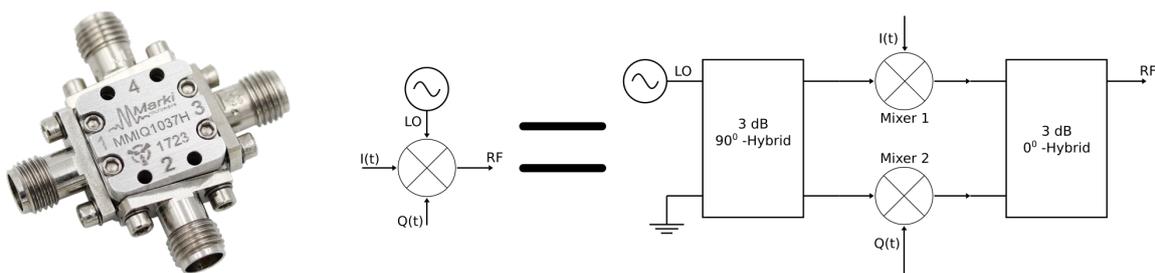


Figure 4.15: **IQ-mixer.** IQ-mixer real component picture (left) and its circuitry symbol (center) and internal circuit block diagram (right).

- Low-loss microwave flexible cables

In order to take maximum advantage of the power generated by the microwave source and the AWG, it is necessary to use microwave cables with optimal transmission. Rigid or semi-rigid waveguides are an option, however not very practical

³⁶Keysight: SD1 3.x Software for M31xxA Digitizers, M32xxA AWGs and M33xxA Combos.

³⁷Marki Microwave: MMIQ-1037H.

for a very compact optical setup, where interfering could be problematic. In the design I considered armored microwave cables by mini-circuits with a wide frequency range with low loss and 50Ω impedance. One limitation of these cables is the availability length of $\sim 1.2\text{m}$, which make hard to keep the microwave source out of the science chamber, considering that Rydberg atoms are very sensitive to electric fields. There is one important parameter of this cable, which is the insertion loss which represents the power lost after passing through the cable which is in the range of 1.3dB for low frequencies and 3.41dB for up to 40GHz.

- Microwave antenna

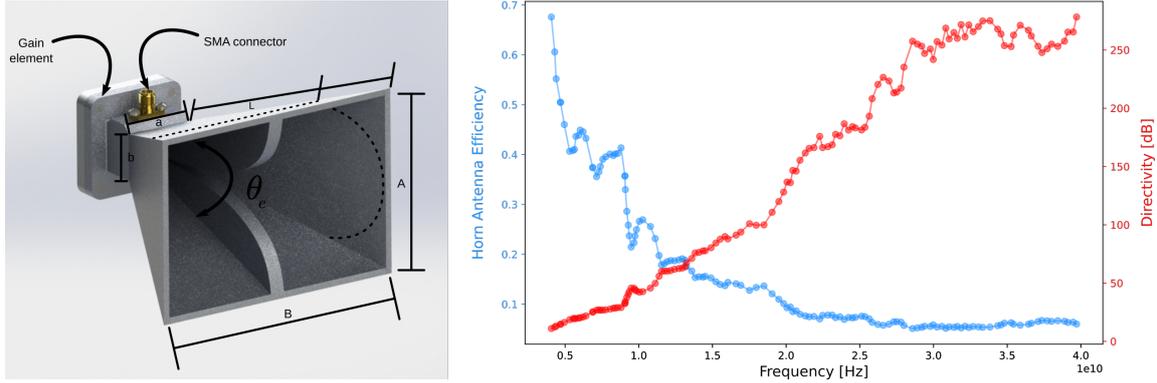


Figure 4.16: **Microwave horn antenna geometry.** Efficiency and directivity from frequency range from 4GHz to 40GHz.

In this design we have selected a simple design of an aluminum linear polarized horn antenna³⁸ with a maximum SWR 2:1 and a gain of 13dB. In the far-field approximation the gain (G) of the antenna is given by:

$$G = \eta_e D, \quad (4.4)$$

where D is the measure of how well an antenna is focuses its radiation or often called directivity and η_e is the antenna efficiency, which also related with the antenna gain via the wavelength of the field and the geometry of the (horn) antenna (see fig. 4.16 (right)), by;

$$G = \frac{4\pi A_e}{\lambda^2} \eta_e, \quad (4.5)$$

where A_e is the effective area of the antenna and λ is the field wavelength.

Hardware setup

The initial setup of the hardware consisted in a sequential connection of the microwave source followed by the IQ-mixer, where additional two terminals from AWG carrying the quadrature signals were connected. The mixed signal produced by the mixer is passed to the horn antenna via the semi-rigid microwave cable in to the antenna that is set under the science chamber looking upward (see fig. 4.17). The antenna was finely positioned to maximize the effect of the atoms by monitoring the effect in the ion signal.

³⁸A-INFO: LB-40400-KF. Linearly polarized horn antenna with (4-40)GHz frequency bandwidth. Size: 65mm x 51.8mm x 41.8mm.

| Antenna | Advantages | Disadvantages |
|--------------------------|--|---|
| Horn antenna | Small minor lobes. Good impedance matching. High directivity. Narrow principal beam. | Design highly dependent on the flare angle. |
| Helical antenna | Simple design and high directivity. Wide bandwidth. Circular polarization control. | Antenna size is larger and efficiency scale down with the number of turns |
| Lens antenna | Robust design and modular beam around the propagation axis. | Bulky and heavy and complex design. |
| Parabolic antenna | Suppression of minor lobes. Facilities for beam adjustment. | Small designs obstruct part of the beam power. |

Table 4.1: **Microwave antennas.** Comparison between microwaves antennas design with advantages and disadvantages.

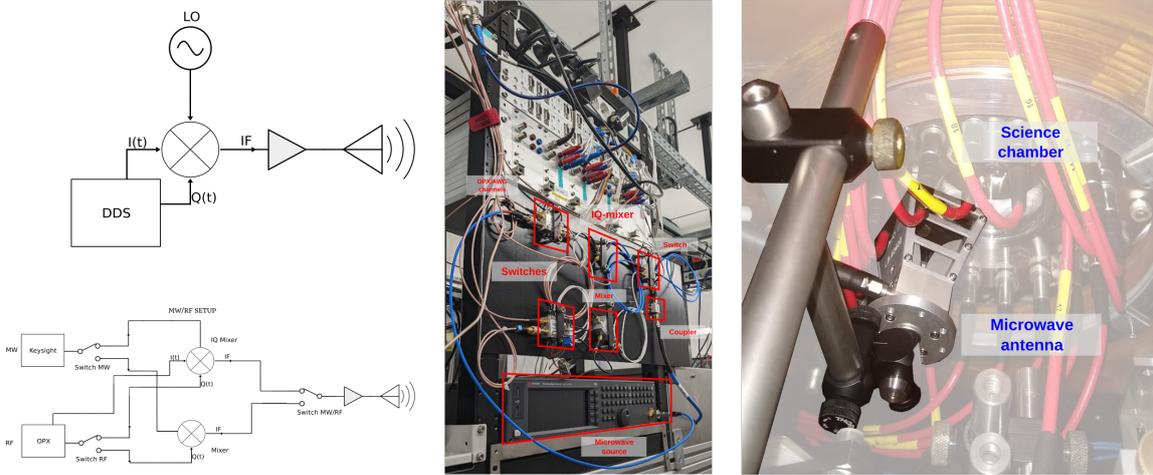


Figure 4.17: **Microwave hardware setup.** Initial setup of the microwave system (left-top) and second version of the setup using OPX channels instead (left-bottom). Experimental setup with the components (right).

Characterization

Before starting to do experiments on the atomic system directly, I tested the system in two configurations with Power Network Analyzer (PNA), one using the system driven by the LO and a DDS for one channel of the IQ-mixer, since the second can also behave like a normal mixer when only one of the IQ terminals is used. The second configuration was replacing DDS by the AWG using both IQ channels as shown in fig. 4.20 and fig. 4.21.

To measure the performance of the side-band suppression I propose a figure of merit that quantifies the efficiency of the noisy 1-side-band suppression \tilde{M} given by:

$$\tilde{M} = 1 - \frac{\left| \int_{\nu_{LO}-2w_{LO}}^{\nu_{LO}+2w_{LO}} S_{LO}(\nu) d\nu \right|}{\left| \int_{\nu_{SB}-2w_{SB}}^{\nu_{SB}+2w_{SB}} S_{SB}(\nu) d\nu \right|}, \quad (4.6)$$

where \tilde{M} is bound within the range $[0, 1]$ and ν_{SB} is the frequency value of the side-band,

ν_{LO} is the frequency value of the local oscillator, w_{SB} is the width of the side-band peak and w_{LO} is the width of the local oscillator peak. The ratio between integrals gives an idea of the side-band suppression with certain noise, values closer to 1 represent a “good” performance, while “bad” performance values are closer to 0.

Performance of the microwave source for frequencies in the typical use range for Rydberg transitions i.e., 10GHz, 15GHz, 18GHz and 25GHz are shown in fig. 4.18. Values of \tilde{M} deviate less than 2% of the maximum value possible representing a 40dBc suppression of the highest spurious frequency for three different powers -10dBm, 0dBm and 10dBm.

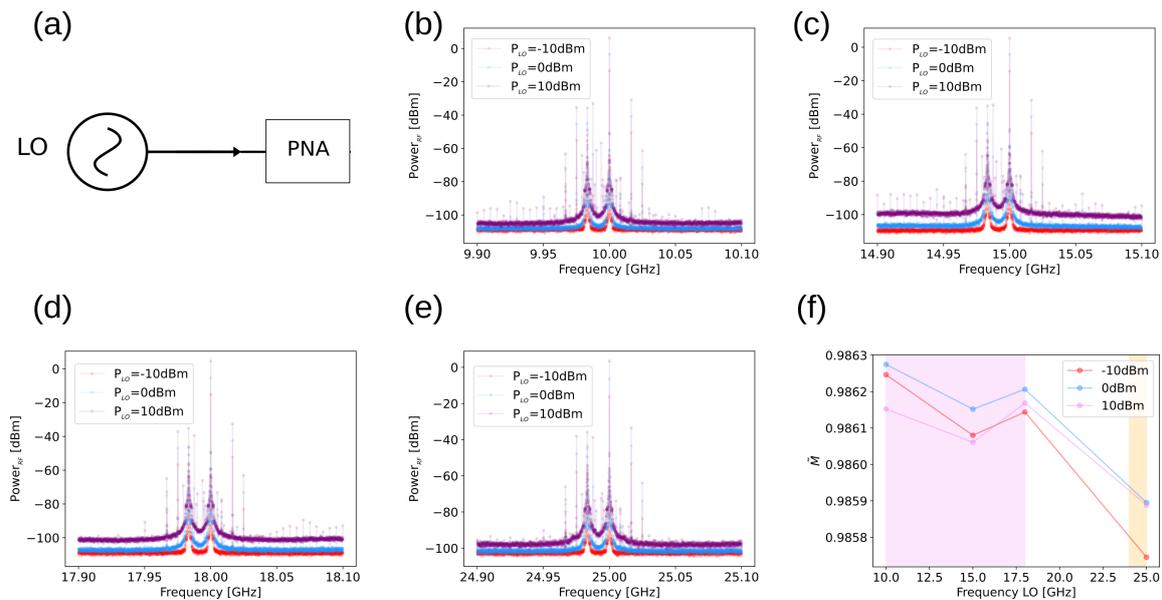


Figure 4.18: **Microwave source characterization.** (a) Setup scheme for characterizing spurious frequencies. (b-e) Spectrum power for 3 input powers of the microwave source for 10GHz, 15GHz, 18GHz and 25GHz center frequency. (f) The (\tilde{M}) performance for the different center frequencies of the source, with region of higher (purple) and lower (yellow) performance.

Additionally, I checked the spectrum of our DDS (see fig. 4.19). It exhibits spurious frequencies with more than 60dBc out of the range of ± 2 MHz at 60MHz frequency and ~ 50 dBc within that range, which is within the range of good signals in microwave systems, where the spurious frequencies can couple secondary transitions within the Rydberg manifold.

The first configuration can be seen in fig. 4.20, where it is possible to observe the result of the signal mixing with double side-bands at 60MHz from the carrier, and additional spurious tones due second harmonics with more than 40dBc, also proved by the estimation of the \tilde{M} value with deviations below 3% from the maximal value of \tilde{M} . Additional power broadening is observed for the largest LO power of about ~ 10 kHz for the carrier and ~ 1 MHz for the sidebands, which is attributed to enhancement of non-linear effects of the internal components of the source and mixer.

In the second configuration in fig. 4.21, the estimation of the performance metric shows

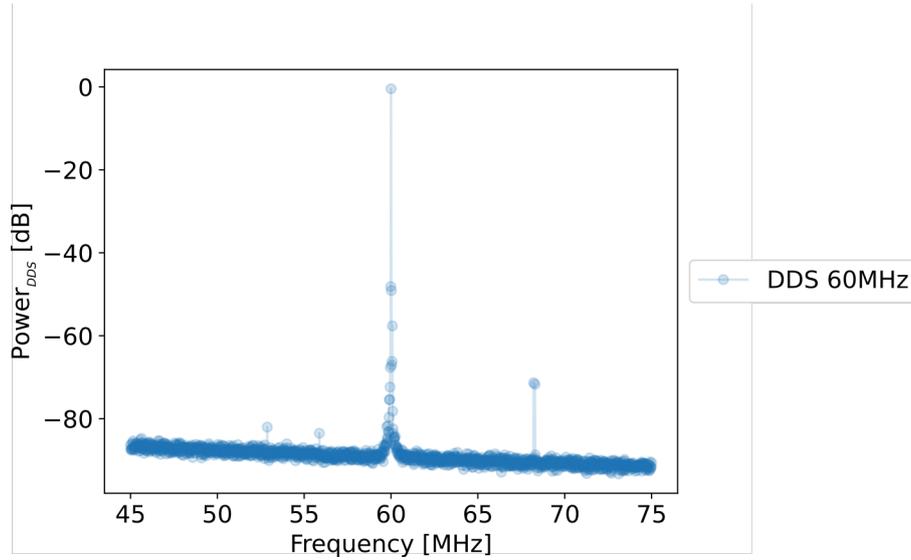


Figure 4.19: **DDS characterization.** Spectrum power of DDS for a set frequency of 60MHz, with spurious frequencies above -50dBc.

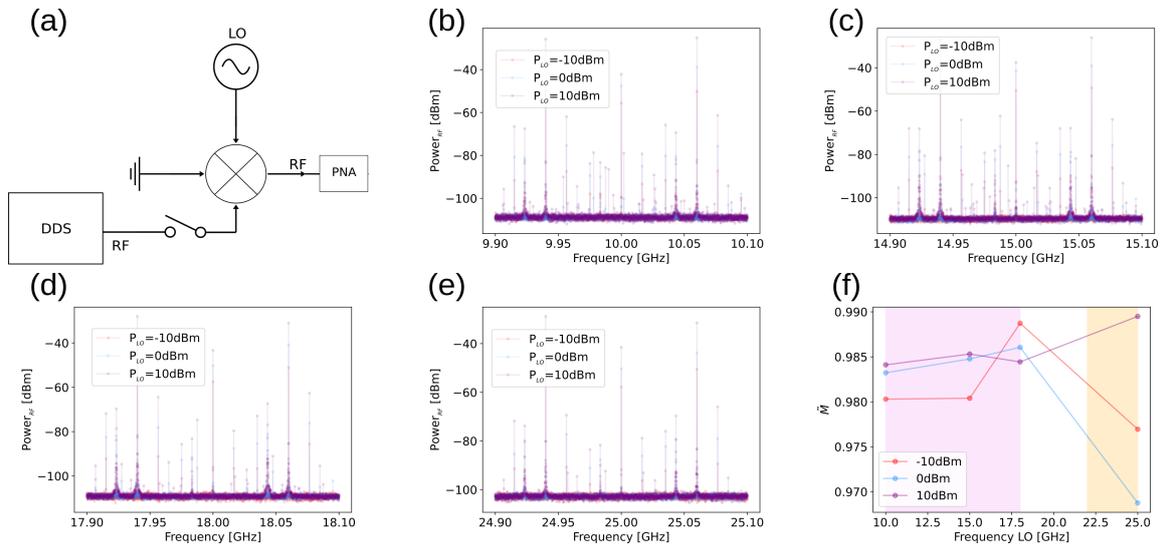


Figure 4.20: **IQ-mixer LO and DDS characterization.** (a) Setup scheme for characterizing spurious frequencies of the setup. (b-e) Output power for 3 input powers of the microwave source for 10GHz, 15GHz, 18GHz and 25GHz center frequency. (f) The \tilde{M} performance for the different center frequencies of the source, with converging value region (purple) and diverging value region (yellow) in the performance.

an improvement with the higher the difference between the carrier and IQ frequencies; measured for different powers of the source, being the highest power the one with best performance. The side-band suppression of left bands relative to the right bands is 40dB for the higher LO power (15dBm) and even higher suppression can be achieved for fine tuning of the IQ-mixer balance.

In general quantum backends comprise a series of electronics, optics and mechanical elements that serves in the generation of the atomic substrates for sets of quantum bits that ultimately will form the quantum processor unit. It allows to cool down the atoms in order to reduce the decoherence and assist in the state preparation and loading of those

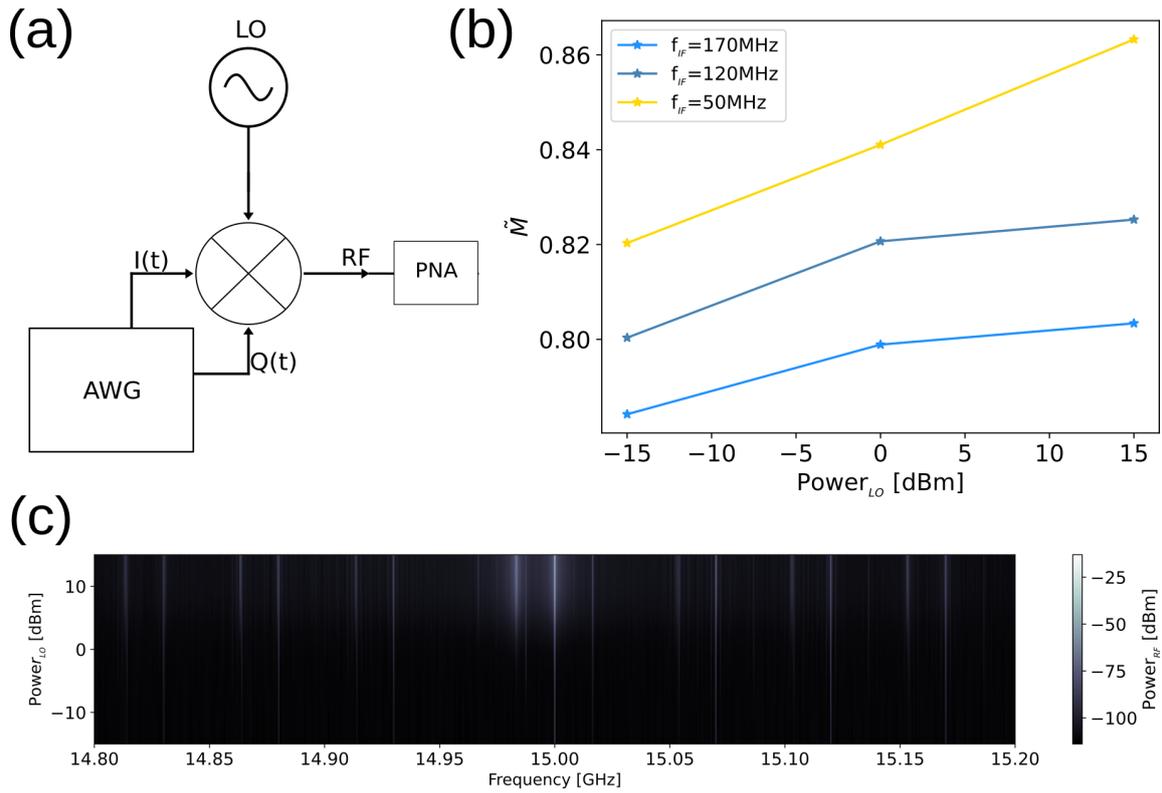


Figure 4.21: **IQ-mixer, LO and AWG characterization.** (a) Setup scheme for characterizing spurious frequencies of the setup. (b) The \tilde{M} performance for different IQ frequencies and different power of the microwave source. (c) Power spectrogram of the RF output for different LO frequencies and powers.

atoms into optical tweezers, where later quantum operations are performed. I presented the architecture of our backend in Strasbourg and the technique for loading the atoms in MOTs and tweezers, together with the techniques and hardware that are used for detection and manipulation. Finally, I presented the design of a new microwave system installed in the backend for global control and its characterization.

Atomic Quantum Processor Unit

“Quantum computing will fail if the qubit errors are not improved significantly from where they are now, for all qubits.”

J. Martinis (2022)

This chapter describes the experiments and observations I realize for generating the substrate of a Quantum Processor Unit using atomic architectures. I explore the possibility of using ensembles of atoms as well as single atoms trapped in optical tweezers. For the former I realize Rabi and Ramsey experiments, in order to characterize the rr-qubits controlled by global addressing in a pancake trap and arrays of optical tweezers.

5.1 Experimental sequence

In our backend we can currently program a series of sequences using LabView that can be stack and play sequentially, this control is much less advance than the one described in chapter 3 but still allow us to test some simple experimental protocols. In our approach described in chapter 4 we prepare the atoms in a MOT for later passing them to a flat dipole trap where we do evaporation cooling and finally load the atoms. After the manipulation (microwave driving, for example) we can detect the system by using absorption imaging from top or side using resonant light, as well as field-ionization where a large electric field. In the later, the electric field is turned ON ionizing the Rydberg atoms accelerating and focusing them in a MCP detector that produce an amplified signal that is read by a fast oscilloscope, this trace in the oscilloscope carries the real ion signal that is analyzed, determining the single, double or multiple events using the *Orthogonal Matching Pursuit algorithm*. A general schematic of the sequence can be found in fig. 5.1

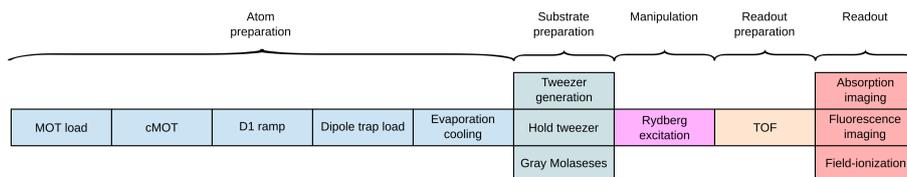


Figure 5.1: **Experimental sequence.** Experimental setup stages for atomic preparation, substrate preparation, manipulation, readout preparation and readout.

5.2 Register state preparation

The state preparation of a register involve two steps, the state preparation for generating the substrate or quantum system that will hold the qubit state, and the preparation of the specific state of the qubit i.e., the correspondent physical state for $|0\rangle$ and $|1\rangle$. Below I will summarize the experiments done for generating the substrate for qubits using single atoms (for gg- and gr- qubits) and atomic ensembles (for rr-qubits).

5.2.1 Calibration of fields

In order to start the experiments with microwaves over Rydberg atoms, it is important to calibrate the straight electric fields with the laser detuning to the Rydberg state. Scanning the frequency of the laser for different values of the electric field, we can observe the resonances where the energy shift is cancelled fig. 5.2. Then for a new value of the detuning we can select the value of the electric fields to be adjusted, see fig. 5.2 (left). Meanwhile the calibration of the microwave power over the number of Rydberg atoms is shown in fig. 5.2 (right), where we observe a quadratic dependency of the Rabi frequency over the number of atoms included.

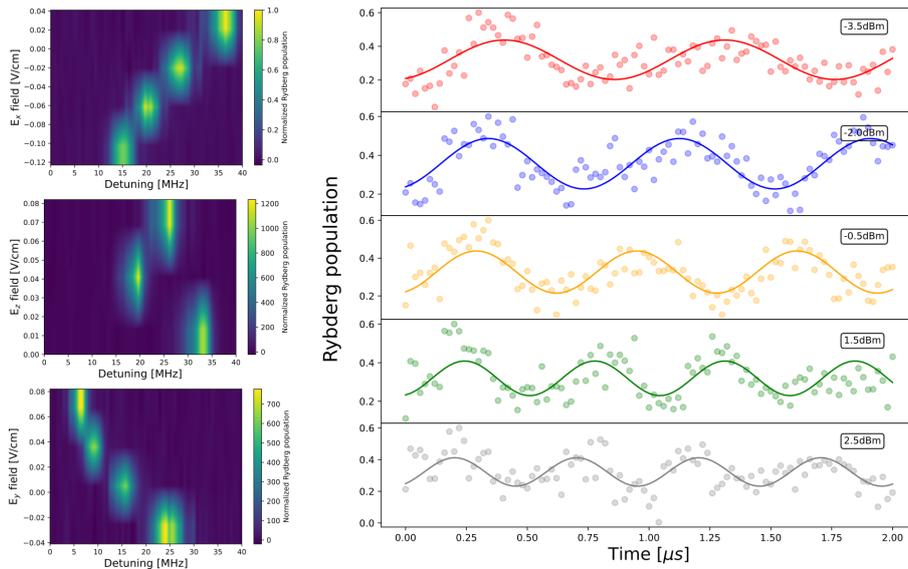


Figure 5.2: **Fields calibration.** Electric field calibration of the Stark shift from electrodes in the 3 directions (left) and the microwave power (right).

In chapter 4 I elaborated on the characterization of the elements of the new microwave setup. In the following I extend the characterization of the previous chapter to direct observations on atoms. These observations include direct effects on the integrated ion signal of the Rydberg population, by changing the power from the microwave source we observe a $\sqrt{P_{LO}}$ dependency. The Rabi frequency varies from 1.1(1)MHz to 1.9(9)MHz when the power of the microwave source is changed from -3.5dBm to 2.5dBm. The noise in the measurement can be attributed to different reasons: atomic motion of the Rydberg atoms which produce gain a phase or get lost when completed the cycle to the other Rydberg state, this is due to they do not remain trap when excited to this highly-excited state; interactions between them, imperfect blockade, allowing inhomogeneous dipole-dipole interactions between Rydberg atoms and internal reflections of the microwave field

inside the chamber. These reasons are clearly a decoherence factor and it will reduce the coherence time of the qubit which is fundamentally limited to hundreds of microseconds of the Rydberg lifetime.

The internal reflections in the chamber were minimized by positioning the horn antenna as vertically straight as possible avoiding the microwave field to have another components. Additionally, I will later discuss a characterization of the blockade effect in the preparation of atoms in the Rydberg state and measuring the effect of interactions on the coherence of arrays of atomic ensembles in tweezers by using different geometries.

5.2.2 Single atoms

The current state of the art is based in the generation of large arrays loaded single-atoms in tweezers. By default, this approach cannot generate defect-free arrays and should be assisted by atom re-arrangement which is carried in time-scales of hundreds of microseconds by driving the frequencies of 2D AODs that can move the atoms from one side to another in the array. This also include a pre-calculation with high complexity for the optimal set of $N/2$ moves that will create a defect-free array of N single atoms [Sch20].

Previously our experiment focused exclusively on large atomic ensembles and micro-sized clouds of 10-50 atoms in tweezer arrays. In this section I outline our approach to load a single ^{39}K atom in a single optical tweezer using 920nm laser in our backend as a substrate for QPUs. Firstly, the focus of the tweezer is aligned with the pancake trap in order to efficiently load the atoms. The alignment is realize by first using resonant light and depleting the atomic cloud and optimizing the speed of depletion, for better overlap of the tweezer focus with the pancake the depletion occurs faster. Afterwards, the beam light was changed for the 920nm laser as show in fig. 4.10, this moves the focus of the tweezer by approximately 250(4)nm. Using absorption imaging we image the tweezer and measure the waist of the tweezer with around $4\mu\text{m}$ using around 150mW power out of the fiber transporting the 920nm light. The atoms exhibit a trapping time of around 784.20(8)ms measured from exponential decay fittings in absorption images made from hold-time experiments fig. 5.3.

In order to reach single atom detection, we use fluorescence imaging using resonant D1 light. This light produce pair-wise loss leaving the trap with one atom or none. In this way it is possible to probabilistically load the tweezer, independently on the number of atoms but depending on the parity number of atoms i.e., even or odd, making it robust against atom number fluctuations. In the approach followed in this work a red molasses scheme is perform by the D1 transitions (cooler and repumper) used for both: assisted collisions and fluorescence imaging. This means that it is important to characterize the number of atoms with the imaging beam time and tune the time for collisions. In fig. 5.4 we observe the atom preparation for different initial collision time pulse from 0ms to 4.5ms, here the difference between 4.5ms and longer is almost zero, showing that the accumulative fluorescence vary at the constant rate of one atom scattering rate. In other words, by having a 50% probability to load the atom in the tweezer, we collect the half of the total possible amount, in consequence the accumulated fluorescence would increase, meanwhile for more atoms the trend increase with the amount of exposure time. The collected fluorescence has a very fast initial increase at the beginning of the imaging pulse and small increase after approximately 20ms due to the loss of atoms by assisted-collisions and the remain of a single atom contributing to the detected signal. In the

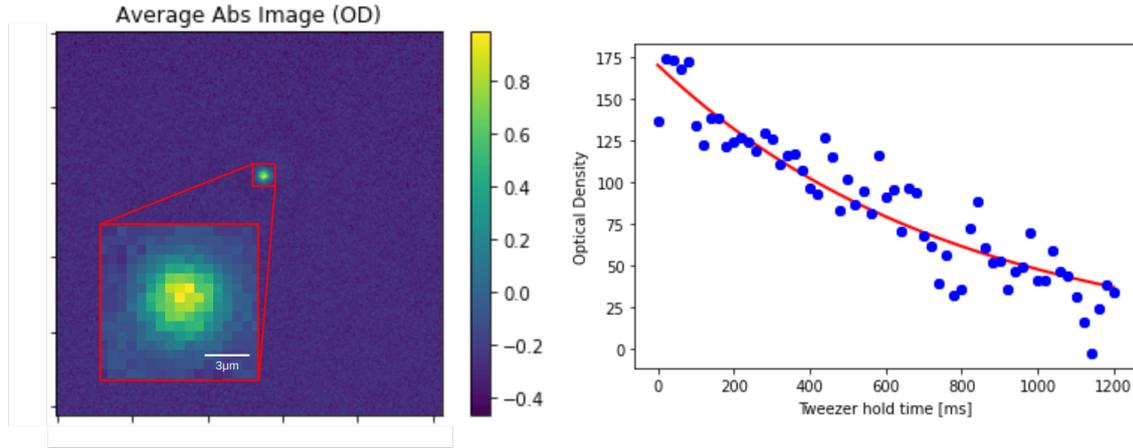


Figure 5.3: **Single tweezer: absorption imaging.** Absorption imaging of a single optical tweezer with tens of atoms with a width of $2.320(2)\mu\text{m}$ from a fitted Gaussian function (left) and the trapping time of the atoms for 784ms from exponential decay fitted function (right).

case of fluorescence the size of the Gaussian fit is around $3\mu\text{m}$ with a signal-to-noise-ratio of 20dB thanks to the covering of the optical path. The trap depth of the tweezer is estimated in 1.74mK with a trap frequency of 0.41MHz, and a maximum scattering rate of 18.19Hz.

When the light shifts due to the optical tweezer are compensated (20MHz), an additional fine tune of the tweezer power during imaging is performed. The fluorescence is shown as function of the tweezer power and the collision time in fig. 5.5 (left) and it was optimized for relative low-power (4V in the AOM controller) and short collision times ($\sim 10\text{ms}$) that shows high fluorescence, leading to a binary distribution with peaks for the presence and absence of a single atom signal fig. 5.5 (right) within a region of interest (ROI) of 3×3 pixels. In other words, when the signal of an atom probabilistically appears within a ROI after the inelastic collisions, the summed value of the pixels, proportional to the acquired photons coincides to the case of an event in the right peak of the histogram. On the other hand, when the atom is absent the sum of pixels in the ROI will fall as an event of the left-hand side of the histogram, exhibiting a threshold that discriminates the first type of event from the second. A similar ROI located in a noisy area, corresponds to the background noise and coincides with the second type of events, where no atom is present.

The image analysis is performed after applying a fringe removal algorithm adapted from [Ock10] and background subtraction of a nearest-neighbour patch. We can fit a double Gaussian with widths of: 56.20(8) counts and 198.10(9) counts. The distribution is represented as a histogram in blue in fig. 5.5 (right) with a fidelity of 0.9937(5) and the threshold for distinguish both cases resulted in 214 counts.

5.2.3 Atomic ensembles

Another approach to generate the substrate for qubits and ultimately QPUs is using atomic ensembles. This approach will carry a different set of challenges but it is an interesting architecture for tackling a fundamental limit in the qubit reset due to its loss out of the system, without requiring atom re-arrangement in the order of hundreds of

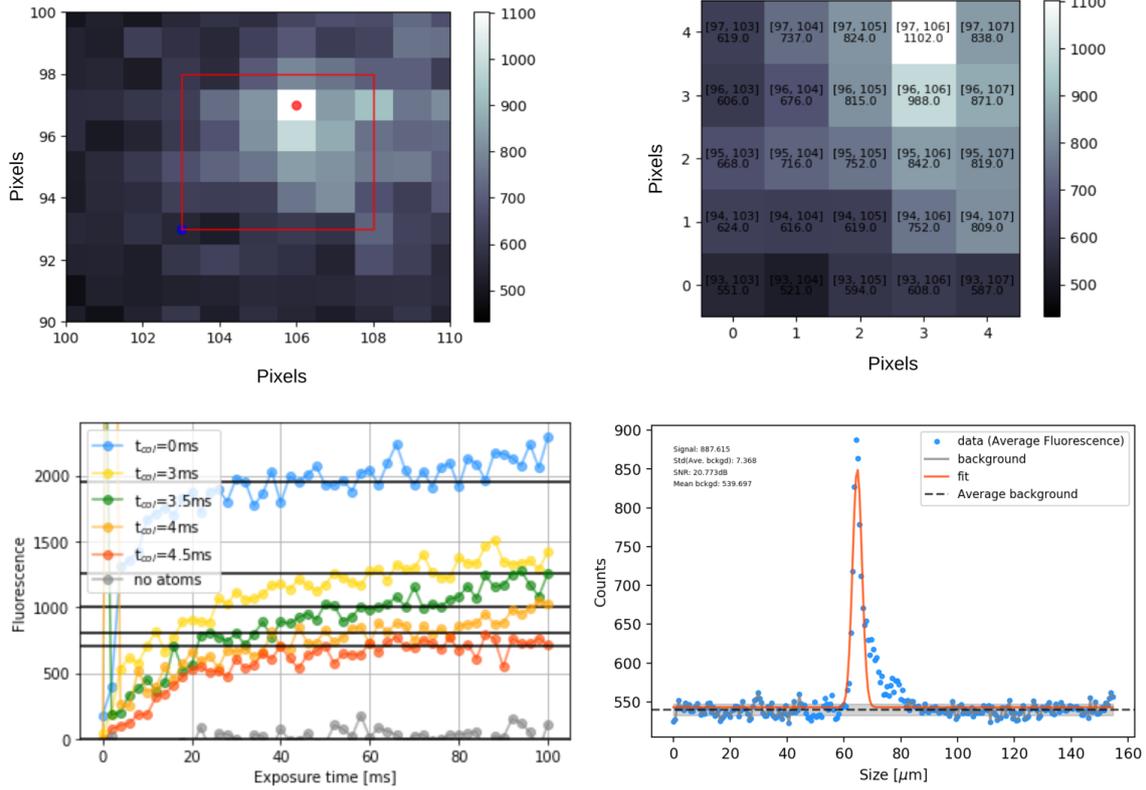


Figure 5.4: **Single tweezer: fluorescence imaging.** (a) Fluorescence imaging with the EMCCD iXon 897 camera using 8x8 binning, x300 gain. (b) Zoom into the region of interest (red box) with the fluorescence pixel value. (c) Fluorescence for exposure time in the case of different initial collision times in the tweezer. (d) Fitting of transversal section of fluorescence imaging.

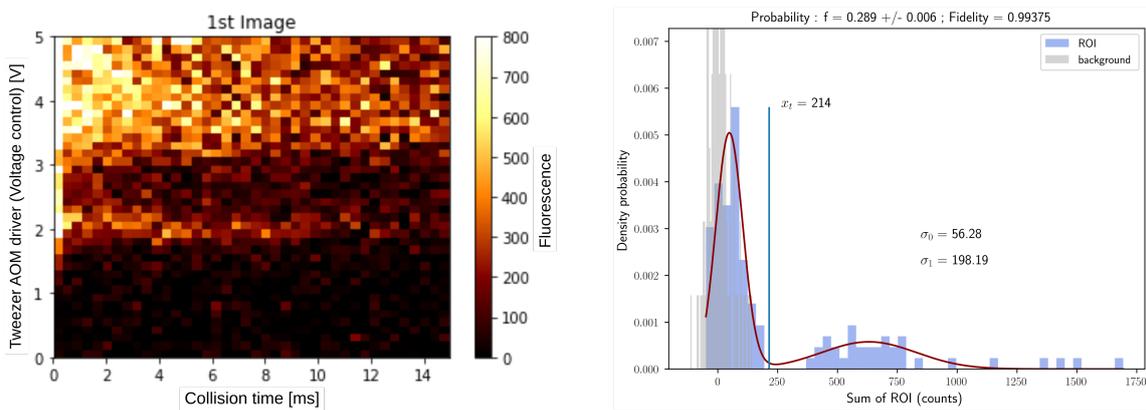


Figure 5.5: **Single atom.** Fluorescence produced for different tweezer modulated power as function of the initial collision time (left). Binary distribution (histogram) for the presence/absence of a single-atom in the optical tweezer with double Gaussian fit (solid line) and a threshold of 214 counts after applying a fringe removal algorithm and background subtraction of fluorescence images in a ROI of 3x3 pixels at 8x8 binning and x300 gain for the EMCCD camera.

microseconds.

Optical pumping

In order to prepare the substrate for the qubit based in Rydberg states, it is necessary to first prepare the atom in a single ground-state that can be address by the circularly polarized excitation lasers (766nm and 455nm) in order to have a pure excited state in a single atom from the atomic ensemble. The state mixture of different $4S_{1/2}$ ground-states would translate into a mixture of Rydberg states with different angular momentum (m_J), introducing population leakage and a more complex dynamics in the system. The characteristics of this Rydberg excitation is more widely presented in the following section. Here I show the effect of a technical upgrade in the optical breadboard where a new beam was added at wavelength of 770nm (cooler) with a circular polarization. This beam illuminates the ^{39}K atoms producing transitions from the Zeeman $|4S_{1/2}, F = 2\rangle$ states to the rapid decay $|4P_{1/2}, F = 2\rangle$ in the presence of a magnetic field that introduce a splitting of such states. Also in the presence of a linearly polarized 770nm light from the vertical imaging path (repumper) that couples $|4S_{1/2}, F = 1\rangle$ with $|4P_{1/2}, F = 2\rangle$, prepares the atom in a single Zeeman state that is uncouple or dark due to selection rules (e.g., $|4S_{1/2}, F = 2\rangle, m_F = 2\rangle$). The atoms in this state are later address by the excitation laser that couple them with the states $|4P_{3/2}, F = 2\rangle$ and afterwards to $|nS_{1/2}, m_J = -1/2\rangle$, where only one m_J state is possible due to the polarization of light.

The sequence of this process is summarized in the plot of the fig. 5.6(a) where the atoms in the pancake trap (1064nm) stays on for approximately 4ms, followed by $5\mu\text{s}$ pulse of the 770nm for optical pumping cooler (from the added beam) and repump (from vertical imaging beam path), relative to the quantization axis defined in the z -axis of the science chamber. This pulse is followed by an approximately $40\mu\text{s}$ excitation beams to the Rydberg state via a two photon process. A magnetic field is kept on during this part of the sequence for an approximately time of 10ms.

In the fig. 5.6(b,c) is shown the optical setups at the breadboards for the beam preparation and at the entry of the science chamber. The effect of the optical pumping in the spectroscopy when detuning the excitation lasers shows the transition of one ground-state to the Rydberg states i.e., $m_J = -1/2$ see fig. 5.6(d) and without effect of the optical pumping with all possible transitions from the all ground-states to the all possible Rydberg states $m_J = -1/2, 1/2$, see fig. 5.6(e).

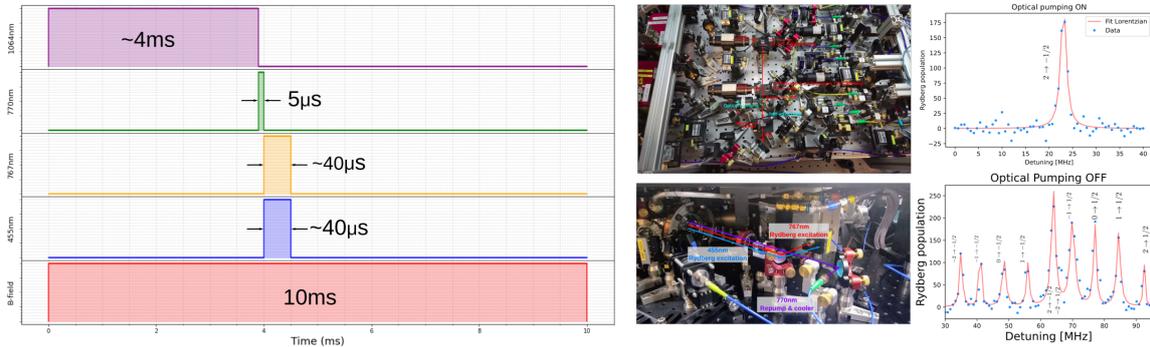


Figure 5.6: **Optical pumping.** (a) Scheme of experimental sequence of pulse for the optical pumping. (b-c) Experimental optical setup for the beam preparation and the aligned beam into the science chamber. (d-e) Spectroscopy showing the effect of the optical pumping, achieving a single and multiple transitions for the presence and absence of the optical pumping.

Rydberg excitation

After applying optical pumping and exciting the atoms into the Rydberg state, it is possible that more than one atom is in the Rydberg state. In order to characterize the excitation probability, the time of the pulse excitation is increased and the ionized Rydberg atom signal is collected for each pulse time (see fig. 5.7). The average probability of excitation to Rydberg states is 54% for pulse times above $0.8\mu\text{s}$. The statistics of the probabilities for the different number of ionized Rydberg atoms that are integrated per shot can be observe in fig. 5.7. The estimated total mean number of Rydberg excitations ($79\text{S}_{1/2}$) in a small atomic ensemble with ~ 50 atoms in the blockade radius of $6\mu\text{m}$ is 0.84 ± 0.11 , with a standard deviation from bootstrapping is 0.8 ± 0.07 taken. This is split in 40% for none Rydberg atom events, 40% for double Rydberg atom events and 20% for triple Rydberg atom events when using resonant light for $1\mu\text{s}$.

On the other hand, when consider simple statistical models for the fully blocked regime where the excitation number is given by a binomial distribution ($p(n)$) where the probability of 1 excitation is p_1 and the probability of 0 excitation is $1 - p_1$, it is possible to write the probability of detecting an ion due to the ionization of one Rydberg atom as a weighted sum (integral) of the probability distribution by the detection efficiency given by another binomial distribution, that depends on the detection efficiency ($\eta \approx 50\%$). In the case of a non-blockaded atomic cloud the excitation pops-up randomly and independent from one to another, this can be described as a Poissonian distribution with a mean (λ), which serves as the probability excitation function. This allow to discriminate between a fully or non-blockaded regime. In the case of fig. 5.7 it shows it is somewhere in between and can be considered partially blocked.

Therefore, using excitation times close to $1\mu\text{s}$ shows a good blockaded Rydberg excitation and with a slightly detuned beams it is possible to improve the preparation of single excitation. This has been later confirmed in our group by checking the blockaded fraction using longer pulses and out of resonance.

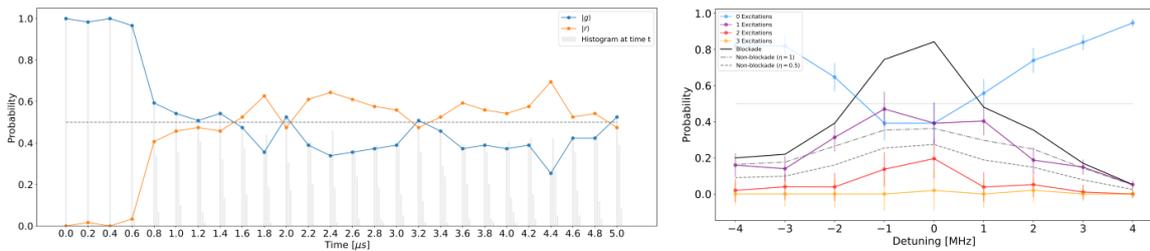


Figure 5.7: **Rydberg excitation preparation.** Probability of Rydberg atom preparation contrasted with the ground state probability for different Rydberg excitation pulse time, showing an average over the 50% probability of having a Rydberg atom (left). Distribution of the average number of Rydberg atoms presence in the microtrap e.g., 0, 1, 2 and 3 Rydberg excitations with a 0.84 mean. Black and gray lines shows the case of blockaded and non-blockaded (with and without detection efficiency) from a statistical model (right).

Array generation

The results presented in this thesis summarize the findings from our work [Mor21], [Whi23], [Wan20] and more details can be found in [She21].

After loading the atoms in the pancake trap as described in chapter 4 we transfer the atoms into the tweezers. To generate the tweezers we illuminate the DMD with a collimated 780nm Gaussian light beam with a 4.3mm waist and a peak intensity of $1.44\text{W}/\text{cm}^2$. Thus, we directly image the atoms as described in fig. 4.11. In this setup, each $(13\mu\text{m})^2$ pixels of the DMD equivalent to $(245\text{nm})^2$ at the atom plane.

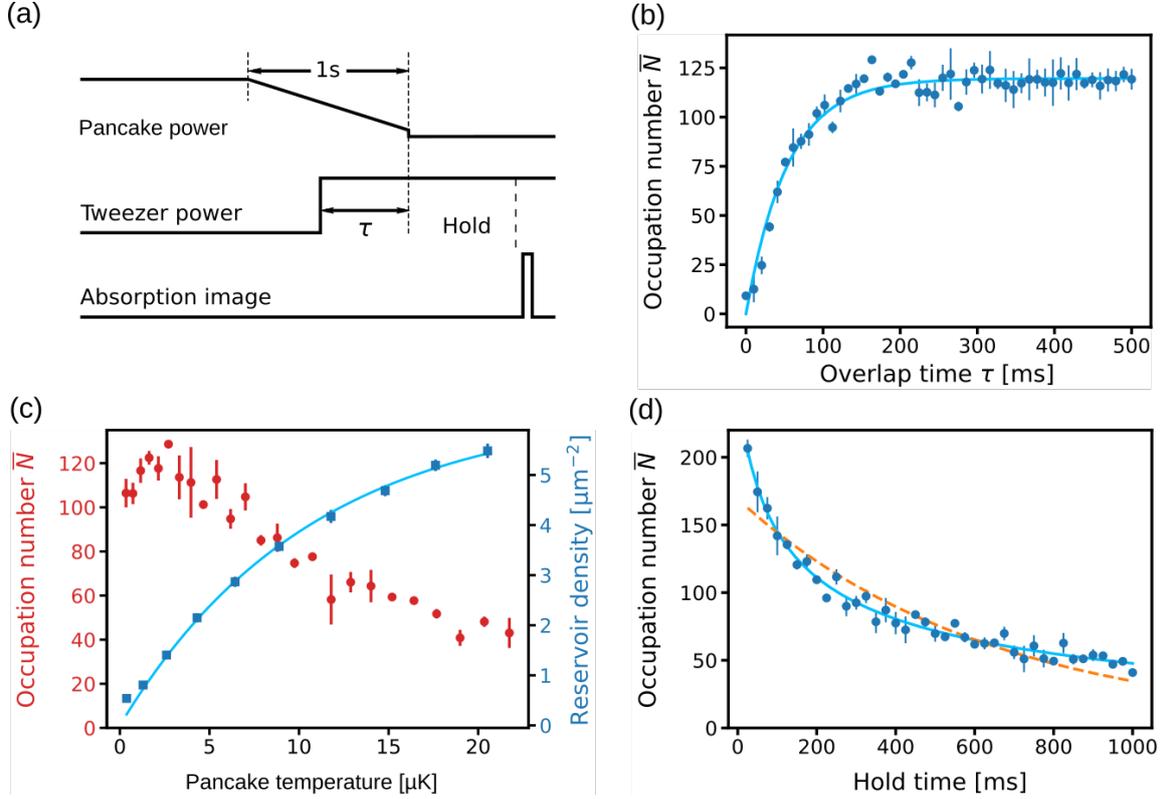


Figure 5.8: **Loading array of atomic ensembles.**(a) Schematic of the experimental sequence of loading the atoms from the pancake trap into the optical tweezer. (b) Effect of the overlapping time between the ramp-down of the power of the pancake trap and the switch-ON moment of the tweezers. (c) Characterization of the pancake trap occupation number as function of the temperature. (d) Holding time characterization considering the one-body loss (dashed), the two- and three-body loss model.

The arrays of tweezers shown in fig. 5.8 are created using different patterns where each site is formed by a region with typical number of pixels between 20-100 pixels. Using saturated absorption imaging technique we detect the atoms as discussed in chapter 4. The probe laser is resonant with the $4S_{1/2} \rightarrow 4P_{3/2}$ transition of ^{39}K at 767nm and with an intensity of $I = 2.1 \times I_{sat}$. An exposure time of $10\mu\text{s}$ is used to image the atoms onto a EMCCD camera using the same optics as for the DMD light patterns (see chapter 4). One advantage of this approach is the fact that it is possible to perform readout of the atomic ensemble within $10\mu\text{s}$, which is an important advantage over the single atom approach.

We used evaporatively cooling of the atoms in the pancake trap while overlapping 200ms with the DMD structured light pattern while ramping down the power, leading an optimal loading of the tweezers in (fig. 5.8). At the end of the ramp the pancake trap is turn OFF being only hold by the tweezers. The characterization of the loading process for a single tweezer of 100pixels and $90\mu\text{W}$ is shown in fig. 5.8.

The overlap between the ramp-down of the pancake trap power and the switch ON of

the tweezers removes selectively the particles with average higher energy. A subsequent thermalization of the remaining particles occurs through elastic collisions. Inelastic processes such as light-assisted inelastic collisions are suppressed by the far off-resonance tweezer beams which minimizes loss and heating of atoms. fig. 5.8 shows that the mean occupation number $\bar{N} = \langle N \rangle_i$ (where the index i it relates with the i -th experimental realization) depends on the final temperature of the pancake trap, with a maximal value $\bar{N} = 120 \pm 5$ at temperatures of $T = 2\mu\text{K}$. Using TOF measurements the temperature of the atoms in single tweezers is $17 \pm 1\mu\text{K}$. fig. 5.8 shows the lifetime of the atoms in the tweezers. One-body decay (dashed curve in fig. 5.8) is dominated by the a three-body and one-body decay model which provides a good fit (solid curve in fig. 5.8). The expression of the model is as follows:

$$\eta = \frac{e^{-k_1 t}}{\sqrt{1 + (k_3/k_1)(1 - e^{-2k_1 t})}}, \quad (5.1)$$

where k_3 is a three-body loss rate and k_1 is one-body loss rate. From fit of the model the three-body and one-body decay constants $k_3=1110\text{ms}$, $k_1=3100\text{ms}$.

An averaged mean occupation number $\langle N \rangle_m$ (where m is the tweezer position) for square arrays with different numbers of sites is shown in fig. 5.9. The orange symbols represent the occupation number averaged over the entire array and over 20 experimental realizations. The experiments show that $\langle N \rangle_m \approx 40$ is approximately constant for tweezer arrays with different numbers of sites in the range of 4 up to 484.

The uniformity could be further improved by adapting compensation of the sites for each tweezer individually and by adapting the wavefront phase of the tweezer beams using SLMs or deformable mirrors.

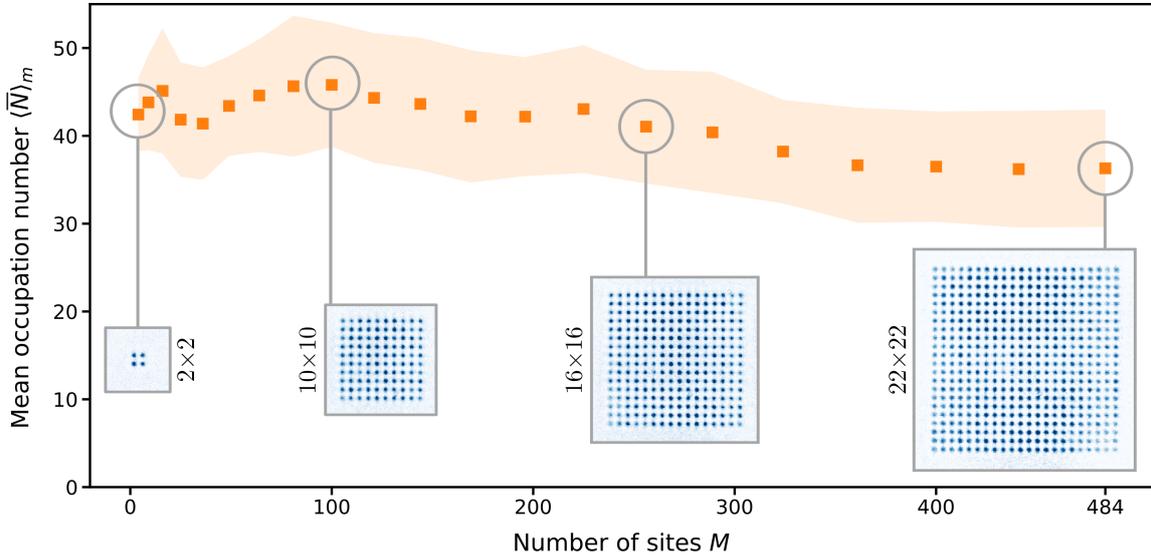


Figure 5.9: **QPU based in an array of atomic ensembles.** Mean occupation number and variance depending on the array size of atomic ensembles.

5.3 Dissipation (depumping) mechanisms

In the atomic architectures it is common to find natural channels of decay that dissipate the state of the quantum system. This is often linked to decoherence and non-unitary

processes that damage the quality of the qubits and the operations. However, it can be used to purify and make more coherent the state of the qubit and operations. This dissipation mechanism we name “depumping” since it pumps down the states in undesired Rydberg sub-states to fast decay intermediate state. We use a blue laser (455nm) to induce the depumping from the excited Rydberg state into an intermediate state 4P. This reduce the leakage to other Rydberg states (10%) and also undesired interactions with other Rydberg atomic states. In fig. 5.10 it is possible to observe the effect of this beam over a large atomic cloud as the pancake with a rapid decay rate of $\Gamma_{\text{depump}} = 48.6(1)\text{kHz}$ obtain from fitting an exponential decay. In the experimental sequence the depumping beam is turn on after the microwave control. On the other hand, when the depumping laser is OFF, the lifetime of the Rydberg states is $\sim 200\mu\text{s}$ which is consistent with the Rydberg life-time.

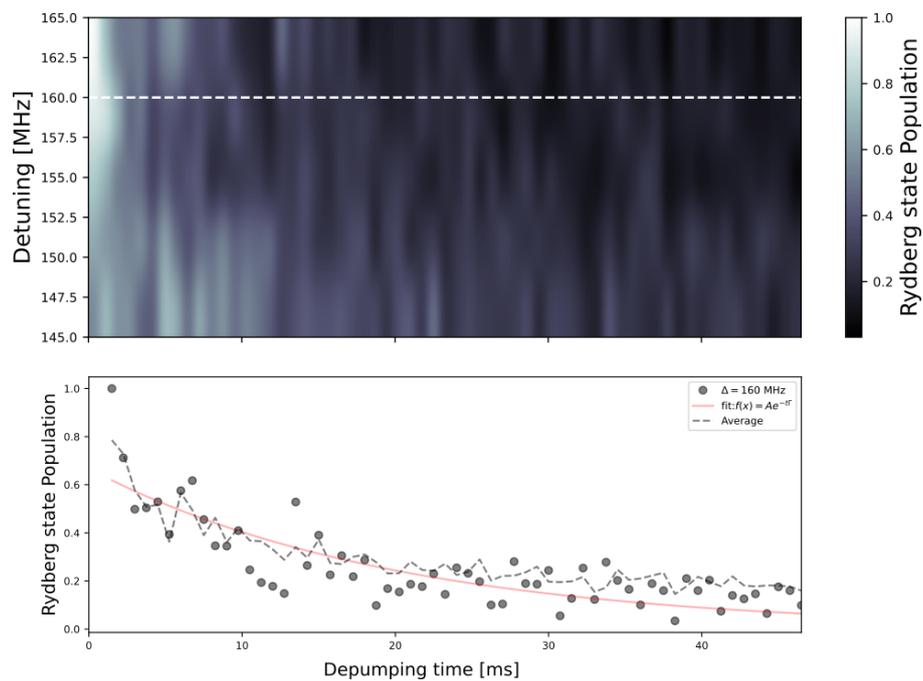


Figure 5.10: **Dissipation mechanism.** Engineered dissipation effect on the Rydberg population for different detunings (top) and a cross-section from the dashed line with an exponential fit (solid red) exhibiting $\sim 48\text{kHz}$ depumping rate (bottom).

5.4 Microwave control of rr-qubits

The rr-qubits are not extensively studied for forming fundamental blocks for quantum computers, but quantum simulation. Here we show a set of experiments we are dedicated to study the behaviour and aspects of this type of encoding. The substrate of a single Rydberg atom from the atomic ensemble is ensured by using optical pumping and the depumping laser after the microwave control. In the following sections I show the experiments using the new microwave setup for the pancake trap and the atomic ensembles.

5.4.1 Spectroscopy

A microwave spectroscopy of the couple two level system (i.e., $79S_{1/2}$, $77P_{1/2}$) by the new microwave system has been realized by driving the microwave source during the excitation pulse time for around $100\mu s$ with a high power of 18dBm out of the source at ± 6 MHz detune from the 22.945GHz transition frequency. Additionally, the excitation laser has been detuned by 20MHz and the total number of ions has been collected (see fig. 5.11). The avoiding crossing in the spectroscopy, shows that for this power is possible to achieve an approximate 7.5MHz Rabi frequency and a substructure in the spectroscopy that is attributed to additional components of the microwave field that couples to secondary m_J Rydberg states, in other words the transitions:

- π -transitions:
 $|79S_{1/2}, m_J = 1/2\rangle \leftrightarrow |77P_{1/2}, m_J = 1/2\rangle$
 $|79S_{1/2}, m_J = -1/2\rangle \leftrightarrow |77P_{1/2}, m_J = -1/2\rangle$.
- σ^+ -transitions:
 $|79S_{1/2}, m_J = -1/2\rangle \leftrightarrow |77P_{1/2}, m_J = 1/2\rangle$.
- σ^- -transitions:
 $|79S_{1/2}, m_J = 1/2\rangle \leftrightarrow |77P_{1/2}, m_J = -1/2\rangle$.

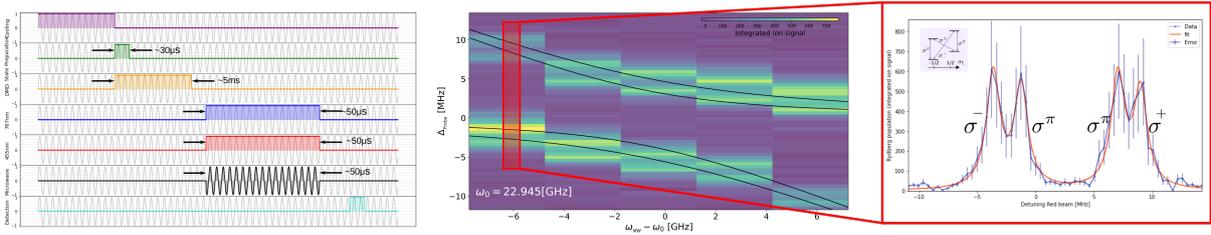


Figure 5.11: **Microwave spectroscopy.** Scheme of experimental pulse sequence for realizing the spectroscopy (left) Avoiding crossing in microwave spectroscopy showing a double structure, presumably from microwave mix polarization (right).

5.4.2 Lifetime

Characterizing the remaining number of Rydberg atoms after the Rydberg excitation, gives an idea of the maximum time where the qubit or atom is present for realizing operations. Here the atom could decay, or because the Rydberg state is not trapped, could be moving out of the trap. The lifetime is shown in fig. 5.12.

5.4.3 Rabi oscillation and Ramsey interferometry in pancake

The new microwave system allows the control of atoms in the Rydberg state, either coming from single atoms or atomic ensembles. In this section I am showing a macroscopic effect of the new system, this is creating single Rydberg atoms using a counter-propagating beams (767nm and 455nm) with opposite circular polarization aligned with the quantization axis along the z -direction of the science chamber. The control driving setup was described in previous chapter (see chapter 4, fig. 4.19), realizing a Rabi oscillations experiment, with a similar sequence to the one presented fig. 5.13 but varying

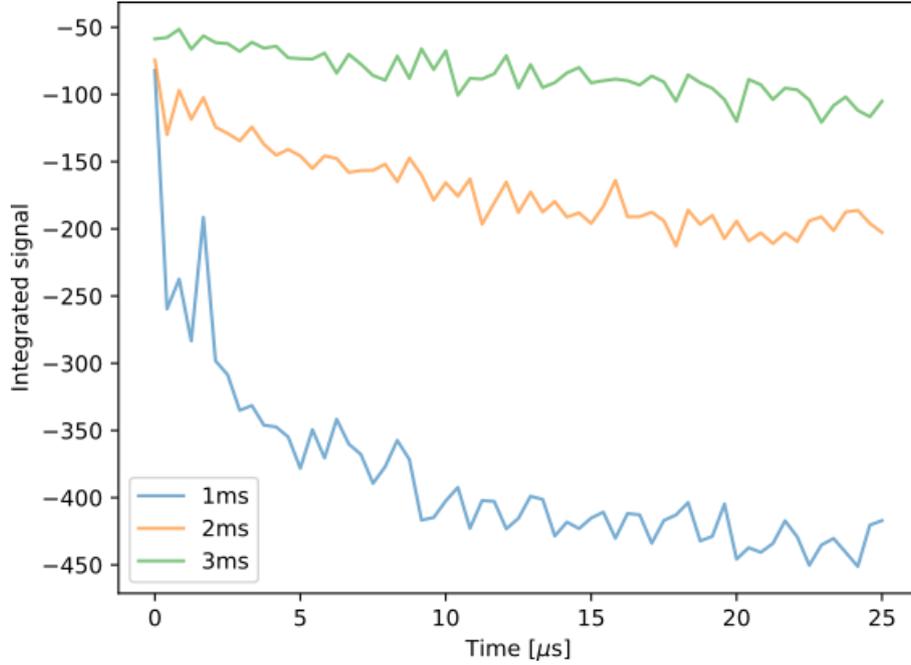


Figure 5.12: **T_1 time (lifetime) of pancake.** Lifetimes of the atoms in the pancake trap after different time of flight (TOF) within $25\mu\text{s}$ window time.

the ON time of the microwave sequence. The result is a coherent driving (oscillations at $\Omega_{\text{Rabi}} = (1.12 \pm 0.01)\text{MHz}$) of the Rydberg atoms generated in the pancake trap for up to $6\mu\text{s}$, before the oscillation get fully damped due to decoherence effects with a damping rate $\gamma = (0.38 + 0.03)\text{MHz}$. These effects are attributed to the fact that the systems started with a higher initial number of atoms that increase the probabilities of collision and then give a higher initial momentum to the atoms in the excited state. Additionally, a Ramsey sequence has been realized with a contrast¹ of 41.16% and coherence time of $6\mu\text{s}$.

Ramsey oscillations can be fitted by 3 types of functions: **Lorentzian**, to represent resonant processes with a characteristic width such as magnetic field fluctuations, collisional broadening, or interactions with nearby surfaces. **Gaussian**, that may indicate the presence of random noise or fluctuations in the experimental setup such as thermal fluctuations, electronic noise, or shot noise in detectors. **Exponential decay**, which are related to relaxation processes as for example: spontaneous emission, phase diffusion, or other. In order to compare the quality of the fitting for the three functions and determine which one is adjusted better to the data, the calculate coefficient of determination (R^2), which are $R_{\text{Lorentzian}}^2 = 0.737$, $R_{\text{Gaussian}}^2 = 0.118$ and $R_{\text{Exp}}^2 = 0.727$ respectively. This show that the system is having influence from both Markovian and non-Markovian processes.

5.4.4 Rabi oscillation and Ramsey interferometry in tweezer array

Naturally, a QPU would be composed by discrete elements or qubits that are well characterized and located in specific spatial coordinates in order to realize operations over them.

¹Ramsey contrast calculated as: $\mathcal{C} = \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}}$. Here I_{max} and I_{min} are the average of the maximum and the average of the minimum values of the signal.

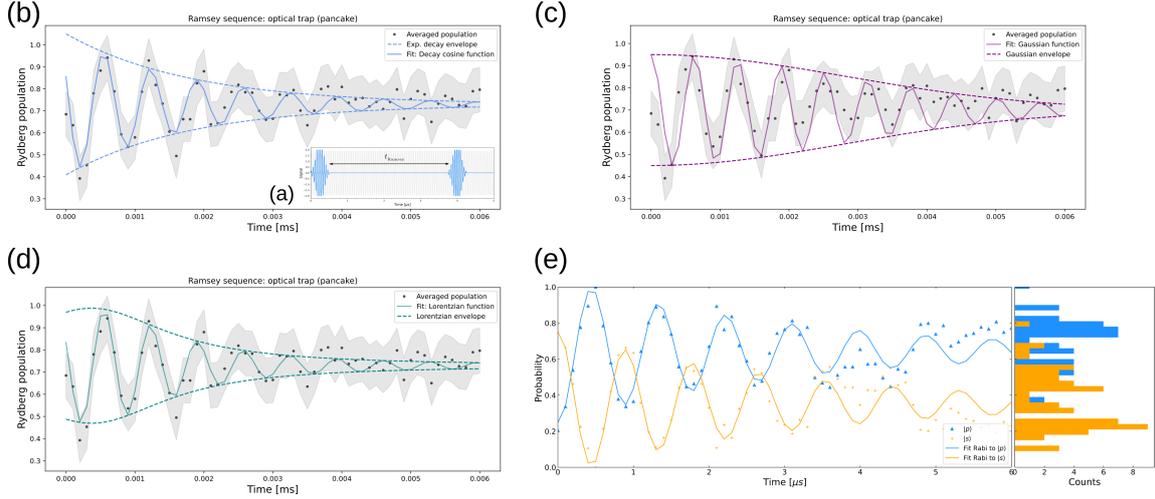


Figure 5.13: **Rabi oscillations and Ramsey interferometry in pancake.** (a) Ramsey fringes for an exponential decay model in the pancake. (b) A Gaussian model. (c) Lorentzian model and (d) Rabi oscillations between $79S_{1/2}$ and $77P_{1/2}$ Rydberg states.

Therefore, the same characterization using arrays of atomic ensembles was performed in arrays of 6×3 arrays where we use a separation of 60 ($36.23 \mu\text{m}$) and 15 ($9.06 \mu\text{m}$) pixel between pairs of nearest neighbours, see fig. 5.14.

For both microtrap geometries the interactions are important and it is expected that the Rabi oscillations and Ramsey fringes get damped. Although a similar behaviour as for the case of the pancake is not observed, it is possible to observe that the oscillations are lower in amplitude (from 0.2 to 0.7 for the fitted function) as they were detuned, this could be related to the induced light-shifts of the microtraps light. The Ramsey contrast for both cases is 84.21% for the 15 pixel separation and 68.69% for the 60 pixel separation. The calculated coefficient of determination (R^2) are lower due to the fluctuations of the detected atoms, also the always-ON interactions affects the coherent driving of the microwave (i.e., less atoms in Rydberg states that can interact, but blocked instead), this can be notice the reduction of the contrast and the decrease on the quality of the fit.

The atomic motion generates phase shifts that agrees with the observations. Estimating the motional speed at an estimated temperature of $15 \mu\text{K}$ is 0.0565m/s and the effective k vector module is $5.662 \times 10^6 \text{m}^{-1}$, gives a coherence time of $T_2 = 4.42 \mu\text{s}$.

In general, it is possible to develop QPU architectures based in loaded single-atoms or atomic ensembles in arrays of optical tweezers, generated from structured light that can be in the case of our experiment in Strasbourg with a DMD but could be generated from static elements e.g., MLAs, DOE, etc. However, both approaches has their challenges, being the second one the studied one more in depth in this chapter. The found sources of decoherence were the always-ON interactions for the case of rr-qubit encodings, but also the deterministic preparation of single Rydberg atoms with a high-detection efficiency. Other source of errors is the state leakage from for example the initial state preparation using optical pumping and the Rydberg excitation.

Additionally, direct observation of the new microwave system has been observed in the atoms, where strong Rabi frequencies can be achieve and improve by purifying the mi-

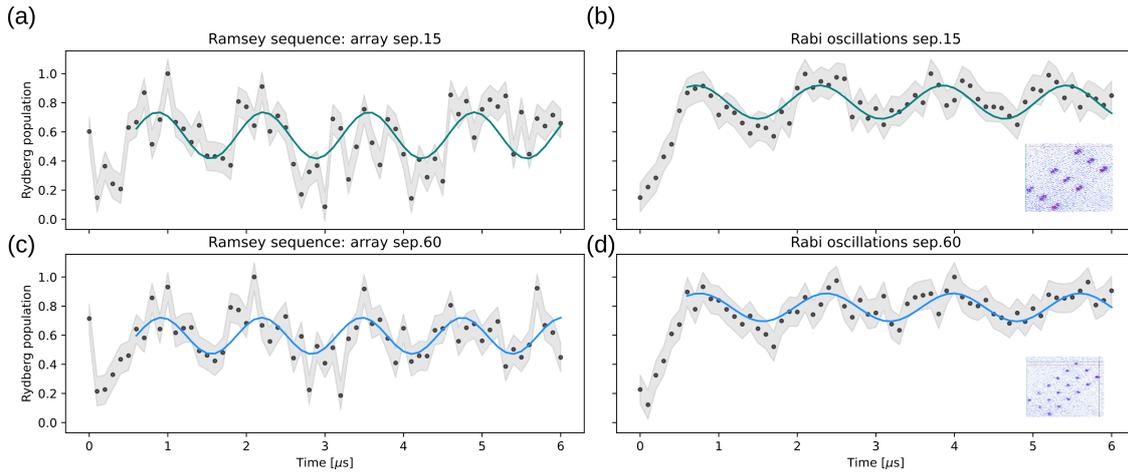


Figure 5.14: **Rabi oscillations and Ramsey interferometry in tweezer array.** Ramsey fringes (left) and Rabi oscillations (right) for Rydberg atoms generated in arrays of optical tweezers, holding microtraps a distance of 15 and 60 pixels from each other.

crowave field polarization. An engineered dissipative channel was implemented to tackle this problem and thus clean the undesired Rydberg states.

A single-atom approach will eliminate many of the experimental challenges, rising others as for example the low single atom detection and the preparation of single atoms in large arrays, that undoubtedly will lead to atom re-arrangement.

Conclusions & Outlook

“In natural science, Nature has given us a world and we’re just to discover its laws. In computers, we can stuff laws into it and create a world.”

A. Kay (1986)

In the first part of this final chapter I wrap up the results and analysis of the chapters presented before and in the second part, I analyze different technical and computational aspects of a fully operational and independent architecture quantum processor unit based on ultra-cold Rydberg atoms and discuss how the architecture integrates into a Universal Quantum Computer design, with quantum hybrid nature.

6.1 Full-stack ultra-cold Rydberg atom quantum computers and simulators

Quantum computers and simulators are composed by more than the quantum systems, it does requires a level of abstractions for the realization of computations, a system to control them with precision and the classical and quantum elements that ultimately serves as quantum processor units.

As addressed in chapter 2 the initial step is to define a set of universal operations that are perform in the ultra-cold Rydberg atom platform. These gates are physically performed by a sequence of operations that define specific transformation over the basic units of information or qubits. One features that places this platform as a clear contender for universal quantum computers is the flexibility of realizing different types of qubit encoding (i.e., gg-, rr- and gr-qubit) with the same atom or system, this open the opportunity to define a more complex computational schemes that bring advantage over the classical counterparts. Furthermore, a set of gates has been presented in the chapter were I divided in single qubit gates based in native rotations of the qubit states, two qubit gates based in the Rydberg blockaded effect and dipole-dipole interactions where a new gate protocol is introduced for $XY(\theta)$ gates based in dipole-dipole interactions that can be used for performing i SWAP operations, which are key for error correction schemes that are in charge of correcting leakage; a final type of gates are introduced, denominated as *parallel gates* which are a sub of the so called multi-qubit gates.

Additionally, quantum computers and digital quantum simulators distinguish themselves from the analog quantum simulators and atomic gas experiments from the fact that they have a full-control system that allows users to access the machines via different levels of abstractions and that they can connect to large classical infrastructures such as High-Performance-Computers accelerating the required computational tasks, as well as giving access via cloud services for user interaction but also with other quantum machines, enabling quantum communication channels.

A quantum software system for this architecture based in specifications called AQiPT has been introduced in chapter 3 as a open-source and Python based software has been develop with a set of modules and a API that constitute the different layers of the quantum software full-stack. This software system is base and inspired in on-the-fly waveform generation and fast-feedback of a new generation of classical hardware that I have also install in our machine. The software also share capabilities not just for hardware and experimental control but also generation of simple theoretical models generated from the experimental sequences that allows to have a better understanding of the experiment. It does also link to quantum computing applications such as quantum circuits and gates that are translated into quantum master equation simulation and waveform instructions that can simply be used for the experimental sequence.

Complementing the full-stack for ultra-cold Rydberg atom quantum computers and simulation is the generation of quantum processor units that are manipulated in order to execute calculations and perform certain task. In chapter 4 and chapter 5 I have introduce a new microwave system into the setup for global control of array of qubits in the Rydberg manifold. Another finding has also been include in the substrate generation for realizing qubits in this platform based on neutral single trapped Potassium-39 atoms and arrays of atomic ensembles of the same atomic element. The former, based in light assisted collisions of resonant light that enhance pair-wise losses in the initially trapped atomic cloud, ending with single atoms that can be measured with fluorescence with fast and sensitive EMCCD cameras. Here the efficiency of the single-atom loading can be certainly improve by using orthogonal polarized light relative to the tweezer beams, the use of D2 transitions for the collection of scattered light which will increase the amount of scattered photons due its shorter lifetimes, use of more stable electronics that introduce smaller amplitude fluctuations on the light from AOM modulation and also more complex protocols using a stroboscopic fashion of the tweezer beam that reduce the interaction between the repulsive potential of the dipole trap and the tweezers beam, together with phase modulation of wavefront of the traps via SLMs or deformable mirrors.

On the other hand, a generation of large arrays of atomic ensembles has been presented using arrays with up to 484 shallow microtraps with a mean number of 36 atoms per trap, that can be used for encoding ensemble qubits that offer the opportunity of fast read-out using EIT detection. Here the use of compensated shapes for each microtrap using a Gaussian distribution was a free phase-control key element that enables this achievement.

Finally, the microwave system has been use to globally control arrays of 3x5 single atomic ensembles with a well prepared single Rydberg atom on it. The microwave field is used to realize Rabi oscillation experiments between two Rydberg states e.g., $77S_{1/2}$ and $79P_{1/2}$. Moreover, a full characterization of the system has been realize together with

microwave spectroscopy that showed one of the signatures of a two-level system as the avoided-crossing and Ramsey interferometry that lead to observe coherent control over large atomic ensembles in a pancake dipole trap as well as in arrays of micro-ensembles with up to $\sim 6\mu\text{s}$ coherence time.

In this sense, this work has introduced a model for quantum computation, it has developed a model for the quantum software system as well as explore possible realizations of potassium-39 QPUs for quantum computers or digital quantum simulators, completing a full-stack. Next, I will give an outlook for future work on a new generation of the Rydberg platform.

6.2 Next generation of the Rydberg platform

“Get a good idea and stay with it. Dog it, and work at it until it’s done right.”
W. Disney

6.2.1 Quantum backend

New generations of backends with larger number of qubits i.e., hundreds of thousands or even millions of qubits will be enabled with the rising of more powerful lasers and new techniques that allow to correct optical setups for effectively. On the other hand, new control electronics will enable the control of larger number beams to control the qubits with high-fidelities as well as new scalable software that facilitates programming the waveforms for the backend.

A set of important elements for the next generation of the backend are:

Cryostat systems

The interaction with the environment by the atoms is one of the main sources of decoherence. With a cryostat system, it is possible to isolate even more the system and reduce the black-body radiation that is critical for rr-qubit encodings, besides giving the option to easily upgrade the quantum backend since the system does not require to bake the system at high temperature and vacuum as the traditional approach.

Alkali vs Alkaline-earth species

Currently, alkali atoms are the working-horse of this architecture with different advantages mentioned in previous chapters. However, Alkaline-earth has shown state-of-the-art performance in terms of gate fidelities.

One approach is to use more than one atomic element in a way that is possible to encode one element for the computation and the second atomic element for the readout of the system in a surface code [Fow12] fashion. However, as proposed by Anand et al [Ana24] the alkaline-earth like atoms can use its secondary electron to do a non-demolition readout, facilitating and increasing the readout time.

The approach of dual species also facilitates the development of QPUs architecture (as the one presented in the next section) which requires different programmable Rydberg

interaction strengths, such that some interactions can be turned ON/OFF in a subset of atoms, as well as the inter-species interactions.

In Strasbourg, the new generation quantum computer based on Rydberg atoms Atomic Quantum Computer As A Service (aQCess) will be based on this technology using potassium and ytterbium

Circular Rydberg states

During my QUSTEC secondment I explored the use of circular Rydberg states, and appreciate their use for quantum computing with this architecture. Their strong Rydberg interactions and their long-lifetimes would be a perfect match for encoding rr-qubits. This can increase the Rydberg lifetimes to a few milliseconds [Höl24] or almost 12 times the typical Rydberg lifetime at room temperature; possibly scaling up to 30ms [Rav23].

A challenge of using circular Rydberg atoms is the increase in the complexity of the experiment, since it will require a new set of electrodes or possibly using an array of helical antennas that allows the phase control of the field by tuning the phase of the simultaneous signals of each of the antennas in the array, and then generate the circularization bringing the atom to a maximal orbital momentum $\ell = n - 1$ by going up in the ladder with equidistant energy gaps.

Fast image detection

One of the remaining bottlenecks of the atomic architecture is the readout speed. This is fundamentally limited by the scattering rate of photons produced by fluorescence. Faster imaging techniques as saturated absorption imaging or EIT imaging would require a cloud or reservoir from which it a shadow could be observed, for example using an array of small atomic ensembles developed in chapter 5. Additionally, faster acquisition could be achieved by using customized Application Specific Integrated Circuits (ASICs) attached to the sensor instead of the general and out-of-the-box camera sensors which come with a vast set of features that are use-specific and that slow-down the process.

Moreover, a fast readout would enable fast feedback to the pulses used for atom re-arrangement and the gate protocols presented in chapter 2, adapting them to optimal versions with robust performance.

Miniaturization

Scaling down the backend while increasing the number of qubit is one of the challenges for quantum computers in general. For the atomic system there are many approaches that can realize a large number of qubits with up to tens of thousands atoms trapped in arrays of optical tweezers, but only a few are in the direction miniaturizing the backend in self. By using integrated photonics would be possible to stack several arrays of atoms and scale the system very quickly [Isi23], while minimizing surface decoherent interactions with protocols that includes separation of atoms further away from the surface before Rydberg-based two qubit gate operations.

The miniaturization of this platform using integrated photonics additionally would facilitate the integration with the electronics that potentially minimize and compensate the noise and errors from the classical control electronics.

6.2.2 Architecture for QPUs

Recent architectures based on atom space re-arrangement have been released [Gyg24] [Blu24]. This architecture requires to move atoms with 2D AODs for storage and entangling regions using hundreds of microseconds for these physical operations. Although a defect-free arrays would require atom re-arrangement a possible architecture can be develop based in a grid of vertical (TV) and horizontal (TH) set of qubits (transmission lines) which could be composed by one type of atomic element using gg-qubit encoding and embedded grid of small clusters (core) e.g., 2x2 rr-qubit encoded atoms. Similar to the ideas of classical flip-flops these core will have always-ON interactions that leads to have a natural clock due to the initialization of its elements in Rydberg states with different parities. In this way, only certain operation over the transmission lines will be allowed depending on the clock cycle of the core, leading to different outputs of the qubits that compose these lines and allowing to entangle with other cores via the TV and TH lines (see fig. 6.1).

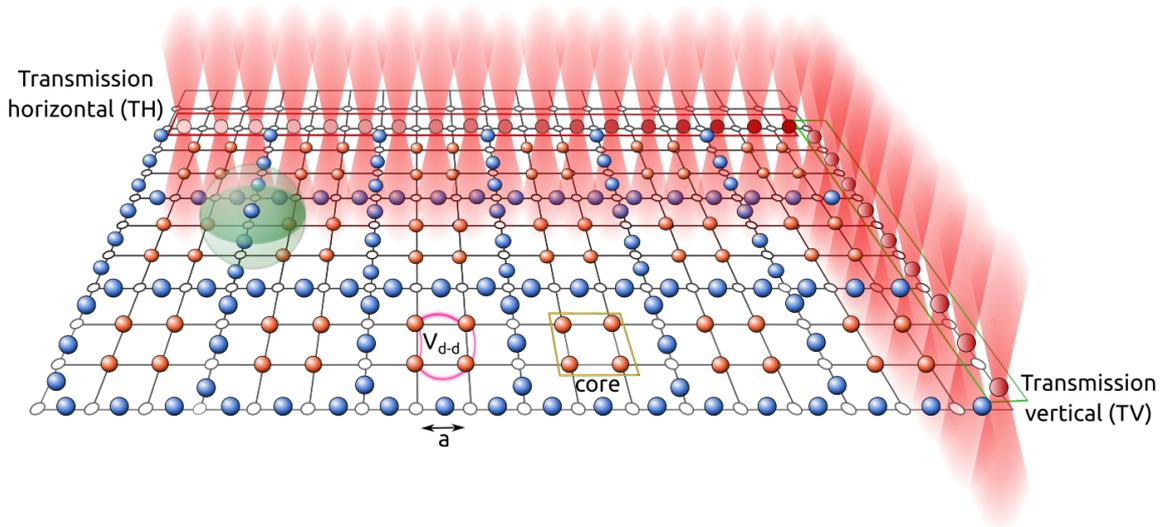


Figure 6.1: **Flip-flop quantum architecture.** Architecture based in core of rr-qubits (orange) with always-ON interactions generating an intrinsic clock, enabling or disabling operations in horizontal and vertical sets of gg-qubits (blue) that allow to access the information in the cores.

The architecture based on this kind of computation based in spatial address and collective encoding of the data qubits will suppress the need of moving the atoms around to entangled and storage regions that generates overhead in the system. The use of multiple species could bring an advantage in tuning the transitions in the different atomic elements.

In terms of storage, the proposed architecture stores dynamically the information in the collective state of the core, similar to a T flip-flop, only being able to change when it is at the right edge of the clock.

In conclusion, this work has developed a set of native gates for the Rydberg platform that allows the execution of quantum algorithms and quantum computing tasks. The qubits used in the platform has be categorize in three different types, enabling the design of more complex computational architectures. Additionally, I have presented a quantum

software system designed to control the resources and processes of a quantum backend using ultracold Rydberg atoms, by abstracting the operations performed over the backend and its hardware composition and the properties of the atomic quantum system in specifications. This abstraction not only allows to control the classical hardware but also use it to precisely manipulate the quantum processor unit. A new generation of commercial hardware for precise, fast and real-time waveform generation has been installed in the lab and integrated in the new software.

Along this work it has also been developed two approaches for the generation of the substrate of qubits based in small atomic ensembles and single-atoms, as well as the generation of a large number of optical tweezers using a DMD for structured light imprinted over the atomic cloud. Finally, a coherent global control of array of atomic ensembles as well as large atomic clouds has been demonstrated with a new microwave system designed and installed in our backend in Strasbourg.

All these elements represent fundamental parts in the composition of the full-stack of quantum computers and digital quantum simulators, with a next generation in the horizon, where new computational architectures will be enabled, see also fig. 6.2.

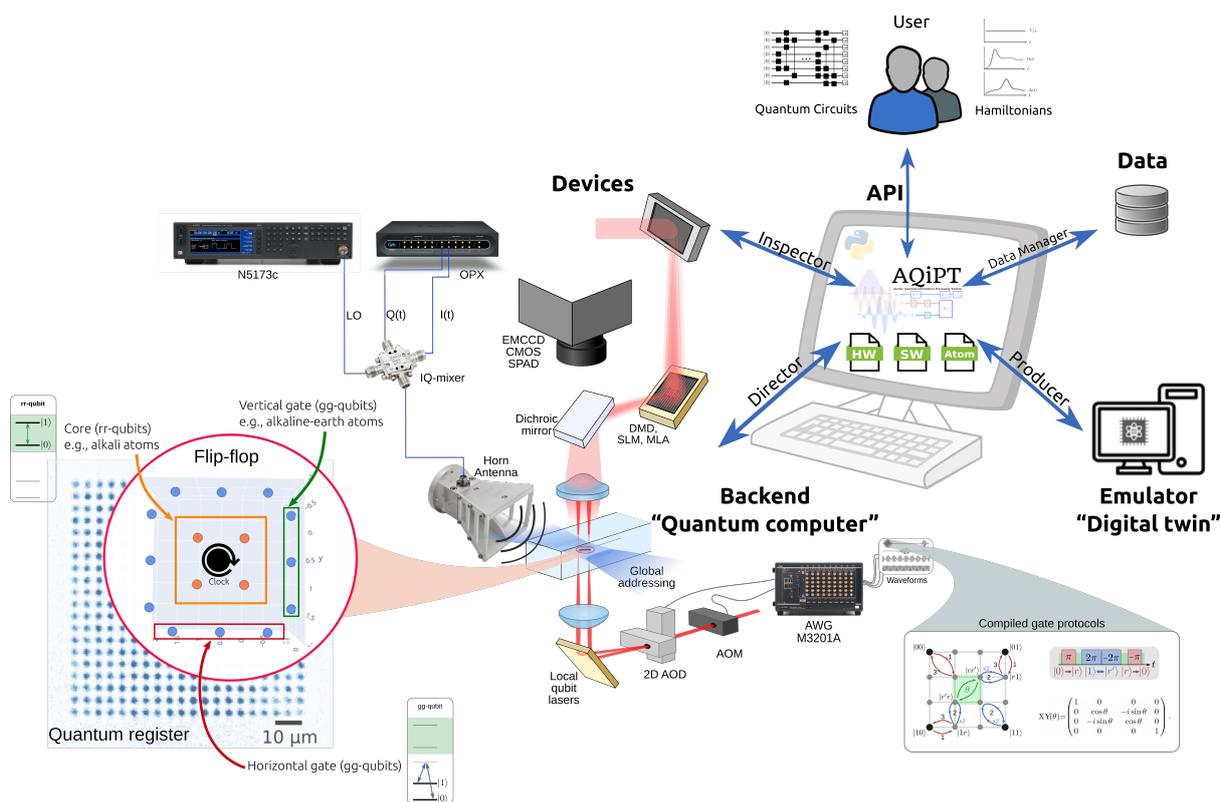


Figure 6.2: **Rydberg full-stack quantum computers and simulators.** Full-stack system for ultracold Rydberg atom quantum computers and digital quantum simulators.

Dirac notation

In quantum mechanics the standard notation of vectors and quantum states is the so-called *Bra-Ket* notation, where each wave vector state is assigned a ket, which mathematically can be represented as a column vector:

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_N \end{pmatrix}. \quad (\text{A.1})$$

An example of this notation are the $|0\rangle$ and $|1\rangle$ kets, which are, in a 2 dimensional Hilbert space (\mathcal{H}), defined as: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Kets have the same properties as vectors. More specifically, kets have the following properties in regards to addition:

1. $|\alpha\rangle + |\beta\rangle \in \mathcal{H}$
2. $|\alpha\rangle + |\beta\rangle = |\beta\rangle + |\alpha\rangle$
3. $|\alpha\rangle + (|\beta\rangle + |\gamma\rangle) = (|\alpha\rangle + |\beta\rangle) + |\gamma\rangle$

For all $|\alpha\rangle, |\beta\rangle$ and $|\gamma\rangle \in \mathcal{H}$. Additionally, kets have the following properties for operations with scalars:

1. $c(|\alpha\rangle + |\beta\rangle) = c|\alpha\rangle + c|\beta\rangle$
2. $(c + d)|\alpha\rangle = c|\alpha\rangle + d|\alpha\rangle$
3. $(cd)|\alpha\rangle = c(d|\alpha\rangle)$

For $|\alpha\rangle \in \mathcal{H}$ and $c \in \mathbb{C} \longrightarrow c|\alpha\rangle \in \mathcal{H}$.

Inner product

Besides the operations mentioned above an inner or scalar product between two vectors can also be defined. In order to define an inner product the dual vector of the ket needs to be defined. This is called the *bra*, hence the name *bra-ket* notation. For each ket $|\psi\rangle$ a vector from the dual space, the bra, can be defined as the Hermitian conjugate:

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} \Rightarrow \langle\psi| = (\alpha_1, \dots, \alpha_N)^*. \quad (\text{A.2})$$

The inner product between a bra and a ket is subsequently defined as: $\langle\phi|\psi\rangle$, where $\langle\phi|\psi\rangle \in \mathbb{C}$. For two kets $|\psi\rangle$ and $|\phi\rangle$:

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} \quad \text{and} \quad |\phi\rangle = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{pmatrix},$$

the inner products corresponds to:

$$\langle\phi|\psi\rangle = (\beta_1^*, \beta_2^*, \dots, \beta_N^*) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} = \beta_1^* \alpha_1 + \beta_2^* \alpha_2 + \dots + \beta_N^* \alpha_N. \quad (\text{A.3})$$

In a more compact notation this can be written as:

$$\langle\phi|\psi\rangle = \sum_{j=1}^N \beta_j^* \alpha_j \in \mathbb{C} \quad (\text{A.4})$$

In quantum mechanics, this quantity is a measure of the probability amplitude for the state ϕ to collapse into the state ψ .

Properties of the inner product

The inner product has the following properties:

1. $c_1 |\psi_1\rangle + c_2 |\psi_2\rangle \longrightarrow c_1^* \langle\psi_1| + c_2^* \langle\psi_2|$ with $c_1, c_2 \in \mathbb{C}$
2. $\| |\alpha\rangle \| = \sqrt{\langle\alpha|\alpha\rangle}$
3. $\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*$
4. $\langle\phi_1|(c_1 |\psi_1\rangle + c_2 |\psi_2\rangle) = c_1 \langle\phi_1|\psi_1\rangle + c_2 \langle\phi_1|\psi_2\rangle$
 $(c_1 \langle\phi_1| + c_2 \langle\phi_2|)|\psi\rangle = c_1 \langle\phi_1|\psi\rangle + c_2 \langle\phi_2|\psi\rangle$ with $c_1, c_2 \in \mathbb{C}$
5. $\langle\alpha|\alpha\rangle \geq 0$

6. $|\alpha\rangle_{\text{NORMALIZED}} = \frac{|\alpha\rangle}{\|\alpha\|}$
7. For $\langle\psi|\phi\rangle = 0 \Rightarrow |\phi\rangle$ is orthogonal to $|\psi\rangle$
8. For $\langle\psi|\psi\rangle = 1 \Rightarrow |\psi\rangle$ is a normalized vector
9. $|c\alpha\rangle \equiv c|\alpha\rangle$
 $\langle c\alpha| \equiv c^*\langle\alpha|$, for $c \in \mathbb{C}$
10. $\langle c\alpha|\beta\rangle = c^*\langle\alpha|\beta\rangle$
11. $|\langle\alpha|\beta\rangle|^2 \leq \langle\alpha|\alpha\rangle\langle\beta|\beta\rangle$ (**Cauchy-Schwarz inequality**)

Outer product

In a similar fashion it is also possible to define an outer product $|\phi\rangle\langle\psi|$. For two kets $|\phi\rangle$ and $|\psi\rangle$ this outer product is defined as:

$$|\psi\rangle\langle\phi| = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} (\beta_1^*, \dots, \beta_N^*) = \begin{pmatrix} \alpha_1\beta_1^* & \dots & \alpha_1\beta_N^* \\ \vdots & \dots & \vdots \\ \alpha_N\beta_1^* & \dots & \alpha_N\beta_N^* \end{pmatrix}. \quad (\text{A.5})$$

The outer product has the following properties:

- $(|\phi\rangle\langle\psi|)|\rho\rangle = \langle\psi|\rho\rangle|\phi\rangle$
 where $|\phi\rangle\langle\psi|$ is linear operator which maps the ket $|\rho\rangle$ to the ket $\underbrace{\langle\psi|\rho\rangle}_{\in\mathbb{C}}|\phi\rangle$.
- $|\psi\rangle\langle\psi|$ is an orthogonal projector onto the subspace spanned by $|\psi\rangle$.
- $(|\psi\rangle\langle\psi|)^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi|$.

Linear combinations

Given n linearly independent vectors $|\alpha_n\rangle$, which form a complete set of vectors (i.e., basis), any vector can be written as a linear combination of these basis vectors. In the case of an orthonormal basis the vectors spanning the space are defined as:

$$|\alpha_1\rangle, |\alpha_2\rangle, \dots, |\alpha_n\rangle \rightarrow \langle\alpha_i|\alpha_j\rangle = \delta_{ij}, \quad i, j = 1, 2, \dots, n \quad (\text{A.6})$$

Any vector $|\alpha\rangle$ can then be written in terms of these basis vectors as:

$$|\alpha\rangle = \sum_{i=1}^n a_i |\alpha_i\rangle, \quad (\text{A.7})$$

such that: $a_i = \langle\alpha_i|\alpha\rangle$.

The norm of the vector is defined as:

$$\|\alpha\| = \sqrt{\langle\alpha|\alpha\rangle} = \sqrt{\sum_i a_i^* a_i} = \sqrt{\sum_i |a_i|^2}. \quad (\text{A.8})$$

Quantum process state tomography

Quantum process tomography (QPT) serves as a robust method within quantum information processing for assessing the efficacy of quantum gates operating with a limited number of qubits. It offers invaluable insights, both empirically and theoretically, into the nuanced effects of processes on quantum states, enabling thorough examinations of noise-induced degradation and imperfections that compromise gate functionality. While conventional metrics like fidelity or distance measures provide singular assessments of deviation from ideal behavior, QPT furnishes a comprehensive analysis, elucidating the specific errors introduced by various imperfections.

This methodological approach entails the construction of a transformation matrix to delineate the intricate alterations undergone by the density matrix of a quantum system during a given process, such as gate operation. The subsequent decomposition of this transformation into operator bases facilitates a granular understanding of the discernible modifications imparted upon input states.

The drawbacks of this figure of merit is that the matrix that characterize the different noise channels will scale rapidly with the number of qubits. So, the goal of the QPT is build this transformation matrix [Moh08]. Considering the quantum map $\epsilon(\hat{\rho}_{\text{in}}) = \hat{\rho}_{\text{out}}$, that formally is written as:

$$\epsilon(\hat{\rho}_{\text{in}}) = \hat{\rho}_{\text{out}} = \sum_i^{N^2} \hat{A}_i \hat{\rho}_{\text{in}} \hat{A}_i^\dagger, \quad (\text{B.1})$$

where $N = 2^n$ is the number of states for n qubits and $\hat{A}_i, \hat{A}_i^\dagger$ are the operation elements. The map can be now expressed in terms of a given orthonormal basis $\{\hat{B}_i\}_i^{N^2}$, fulfilling the condition $\text{Tr}[\hat{B}_i, \hat{B}_i^\dagger] = N\delta_{ij}$, leading to the map:

$$\epsilon(\rho_{\text{in}}) = \rho_{\text{out}} = \sum_{mn} \chi_{mn} B_m \rho_{\text{in}} B_n^\dagger, \quad (\text{B.2})$$

where $\chi_{mn} = \sum_{ij} b_{im} b_{jn}^*$ and $\hat{A}_i = \sum_m b_{im} \hat{B}_m$.

Since along the work I mostly used numerical calculations, the calculation of the QPT has been also done numerically with its implementation in QuTiP [Joh12], which does not have access to the quantum map. Therefore, the software uses an implementation

that calculates the propagator U for the density matrix in the superoperator form, for which it is possible to write:

$$\epsilon(\tilde{\rho}_{\text{in}}) = U\tilde{\rho}_{\text{in}} = \tilde{\rho}_{\text{out}}, \quad (\text{B.3})$$

where $\tilde{\rho}$ is the vector representation of the density matrix $\hat{\rho}$. The superoperator form is then written as:

$$\tilde{\rho}_{\text{out}} = \sum_{mn} \chi_{mn} \tilde{B}_m \tilde{B}_n^\dagger \tilde{\rho}_{\text{in}} = U\tilde{\rho}_{\text{in}}. \quad (\text{B.4})$$

Therefore, an operator U can be defined as:

$$U = \sum_{mn} \chi_{mn} \tilde{B}_m \tilde{B}_n^\dagger. \quad (\text{B.5})$$

This is a linear equation system of $N^2 \times N^2$ equation corresponding to the elements of χ , which can be solve by writing the superoperator propagator as N^4 vectors and the superoperator productor $\tilde{B}_m \tilde{B}_n^\dagger$ as a matrix M with dimensions $N^4 \times N^4$:

$$U_I = \sum_J^{N^4} M_{IJ} \chi_J, \quad (\text{B.6})$$

which solution is:

$$\chi = M^{-1}U. \quad (\text{B.7})$$

Atomic structures

In our experiment we use potassium-39 that is an alkali atom with a single outer electron. A large schematic of the atomic energy structure with the frequency difference and associated wavelengths, considering only the fine structure is shown in fig. C.1.

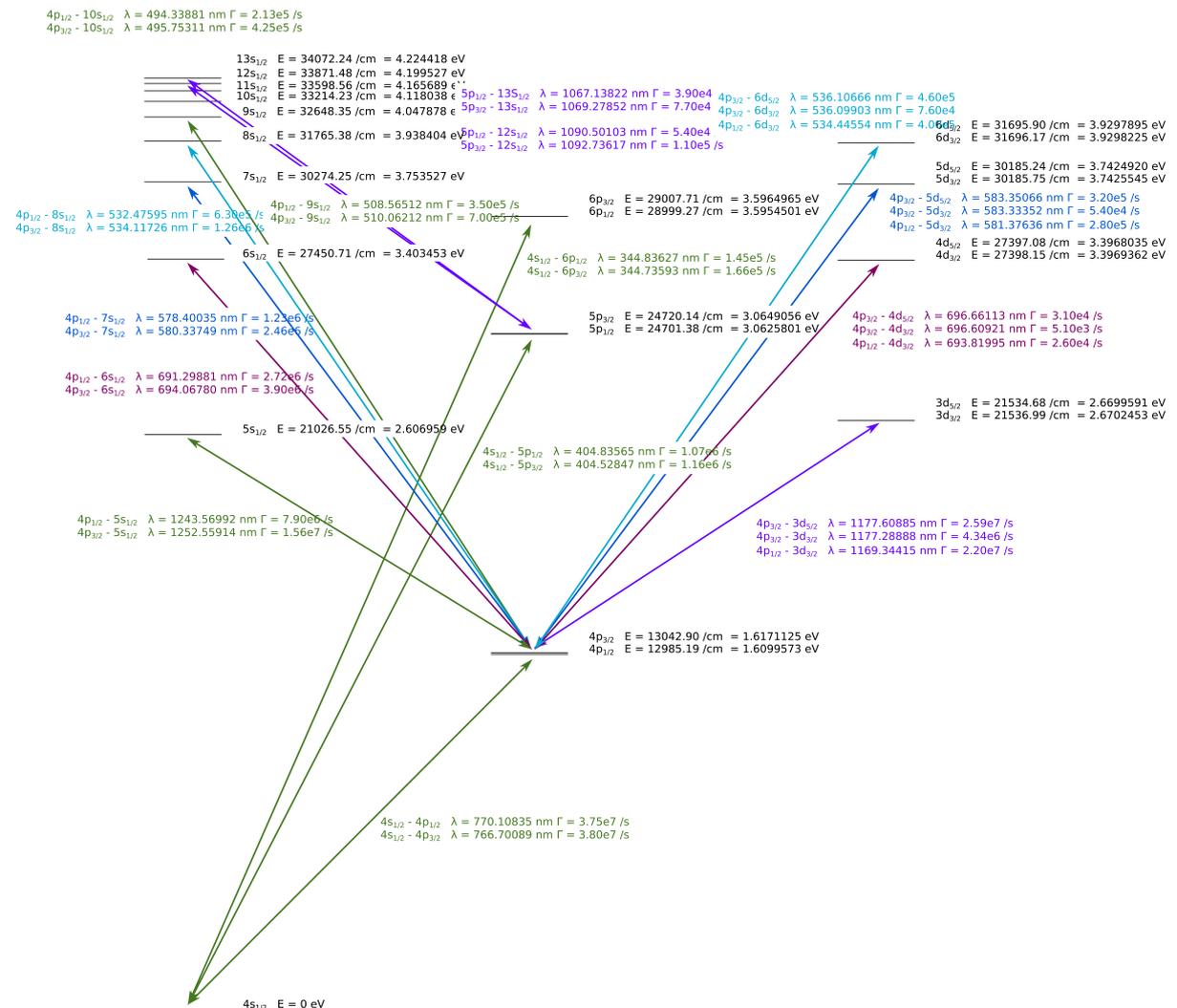


Figure C.1: Potassium-39 atomic energy structure.

Additionally I have added as reference the atomic energy structure for calcium, taken from [Vog09], in whose group I worked during my secondment, see fig. C.2. The atomic energy structure of rubidium [Par08], ytterbium [Kob18] and strontium [Bow19] are also shown in this figure.

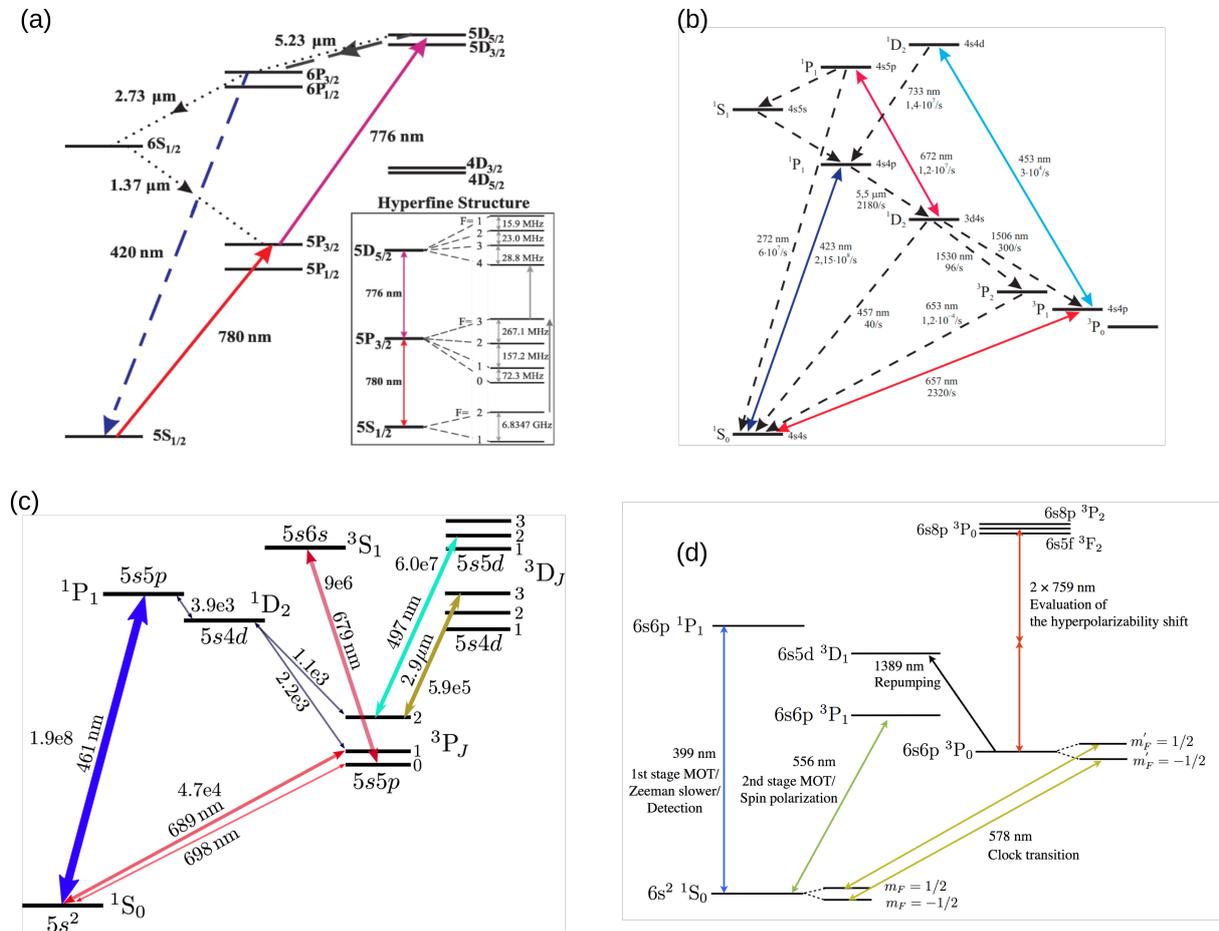


Figure C.2: **Other atomic elements energy structure.** (a) Rubidium-87 atomic energy structure with additionally zoom into the Zeeman sub-levels, from [Par08] (b) Calcium-40 atomic structure exhibiting a blue-cooling transition and a red (clock) transition for a second stage of cooling, from [Vog09] (c) Strontium-88 energy scheme, showing also a two stage transition for cooling the atoms, from [Bow19] (d) Ytterbium-171 with a double clock transition and similar two cooling transitions, from [Kob18].

Strasbourg AOM scheme

Schematics of the AOM setup with double pass (DP) and single pass (SP) configuration. D-lasers beams initially splitted for laser locking to atomic cell spectroscopy and to the cross-over referenced to a Fabry-Perot cavity. The prepared D1 and D2 beams are amplified by a tapered-amplifier (TA) and splitting in 3 branches with a SP and a DP. The branches then prepare 11 beams for the 2D MOT cooling and repumper, 3D cooler and repumper, fluorescence imaging cooler and repumper (labeled CF16), pusher beam, horizontal and vertical imaging, optical pumping and first (red) step for Rydberg excitation. Additional beams from MOPA Mephisto laser for pancake trap, off-resonance beam for tweezers and second (blue) step for Rydberg excitation are present in the scheme.

In this schematic we use the SP to control the intensity of the beam, while the DP is used to scan the frequency of the beam. The calibration of the VCO voltage controllers that are controlling the signal to the AOMs is realized via a linear calibration, this calibration of the type $y = mx + b$ is later used to calculate the final value of the frequency of the beam, by concatenating the calibrations of the AOMs along a chain line from the scheme in fig. D.1.

It is important to remark that the DP of the D1 laser (i.e., “D1 Spec 1”) was changed to SP for optimizing the laser beam power.

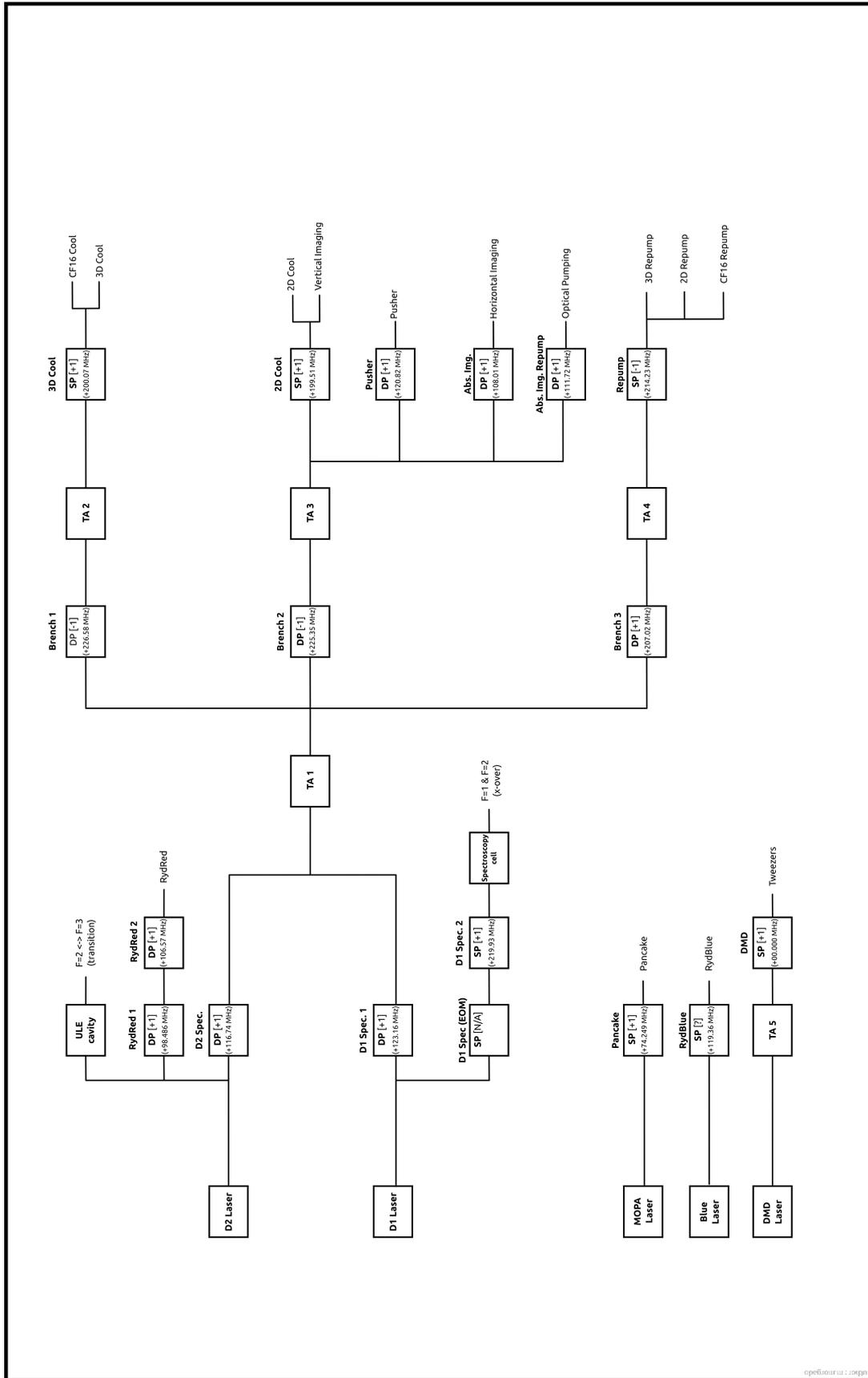


Figure D.1: **AOM schematics.** Schematics of the AOM setup with double pass (DP) and single pass (SP) configuration chain in the system. Simplified and equivalent version of the breadboards optics scheme (see fig. 4.3).

IQ-mixing

In signal processing, mixing is an important and often use technique, beside the drawback it might have by creating spurious frequencies and signals, it also enable different type of modulation i.e., amplitude, frequency and phase modulation in time of a given signal that can be use to drive different devices, in the case of a Rydberg platform, it can be used to drive AOMs and Microwave signals.

In normal mixing, two sinusoidal signals are combined (i.e., s_1 and s_2) which are functions with two different frequencies and same amplitude. Using a normal mixer (fig. E.1), which consists in 3 terminals: RF input signal, local oscillator (LO) and RF output signal.

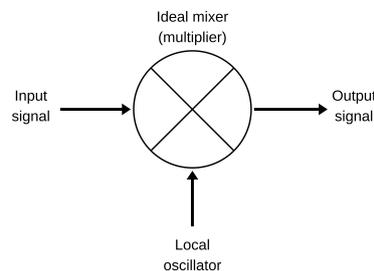


Figure E.1: **Normal mixer.** Mixer with 3 terminals: RF, LO and output.

This will give us a total signal ($s(t)$) that contains the signal of the highest frequency with side bands (i.e., $f_{s1} \pm f_{s2}$). This is mathematically represented by:

$$s(t)_{\text{total}} = [1 + \cos(\alpha)] \cos(\beta) \quad (\text{E.1})$$

$$= \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta) + \cos(\alpha), \quad (\text{E.2})$$

where $\alpha = 2\pi\omega_1$ and $\beta = 2\pi\omega_2$. The examples in fig. E.2 show the effect of normal mixing between 15Hz LO signal and a modulated signal of 2Hz.

The technique of IQ mixing is useful for side-band suppression and up/down-conversion being the later implementations with IQ mixing very versatile and convenient for control of AOMs and microwaves in the experiments. We use IQ mixing, that consist in a passive electronic element that has the role of combining a LO signal together with the quadrature signal coming from an AWG or other, this two signals has usually a phase difference

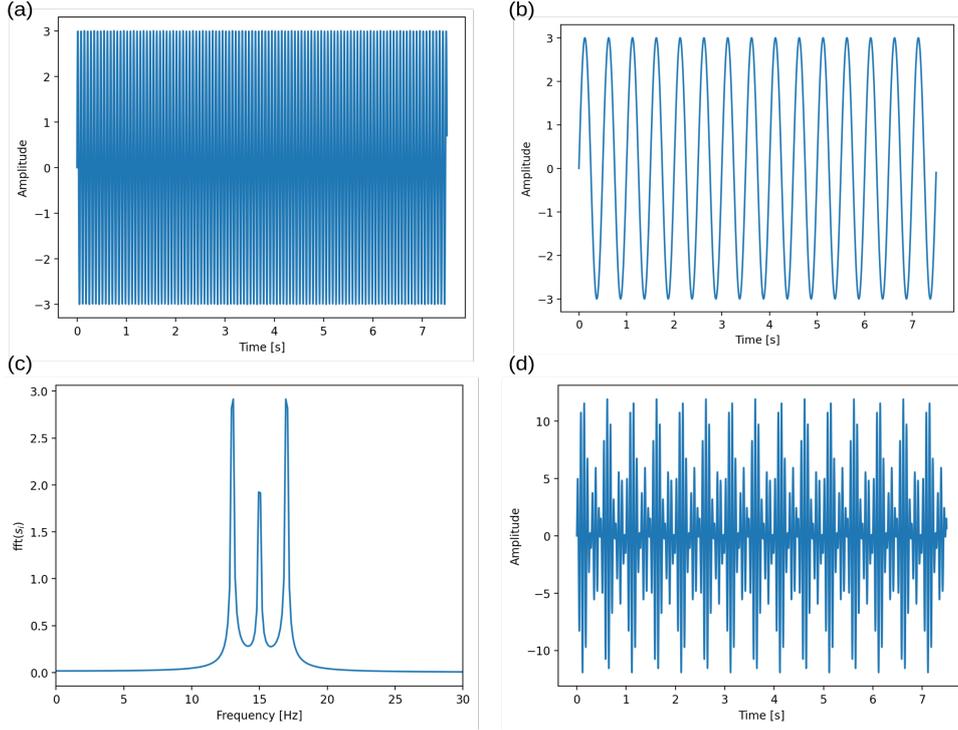


Figure E.2: **Mixer example.** a) LO signal, b) Input signal. c) Mixed signal with multiple frequencies. d) Spectrum showing the (15 ± 2) Hz.

of 90 degrees or $\pi/2$. The schematic of an IQ-mixer has been shown in chapter 4, fig. 4.15.

However, to understand the use of an IQ mixer and the role of the input signals to implement up and down conversion or sideband suppression or more, one should consider the following calculations: starting with functions of the signal of a LO and the quadrature signals I and Q (signals with a phase difference):

$$V_{LO}(t) = A_{LO} \cos(2\pi f_{LO} t), \quad (\text{E.3})$$

$$V_I(t) = A_I \cos(2\pi f_I t), \quad (\text{E.4})$$

$$V_Q(t) = A_Q \cos(2\pi f_Q t - \phi), \quad (\text{E.5})$$

with the frequencies: f_{LO} , f_I and f_Q and amplitudes: A_{LO} , A_I and A_Q . At the IQ-mixer, the local oscillator signal is split in the hybrid coupler with a phase of 90 degrees, for the case of the first it will look like:

$$V_{LO}(t) \Big|_{0^\circ} = \frac{A_{LO}}{\sqrt{2}} \cos(2\pi f_{LO} t), \quad (\text{E.6})$$

$$V_{LO}(t) \Big|_{90^\circ} = \frac{A_{LO}}{\sqrt{2}} \sin(2\pi f_{LO} t), \quad (\text{E.7})$$

After first mixing:

$$V_{M1}(t) = V_I \frac{A_{LO}}{\sqrt{2}} \cos(2\pi f_{LO} t) \quad (\text{E.8})$$

$$= \frac{A_{LO} A_Q(t)}{\sqrt{8}} \{ \cos(2\pi(f_{LO} + f_I)t) + \sin(2\pi(f_{LO} - f_I)t) \}, \quad (\text{E.9})$$

and

$$V_{M2}(t) = V_Q \frac{A_{LO}}{\sqrt{2}} \sin(2\pi f_{LO} t) \quad (\text{E.10})$$

$$= \frac{A_{LO} A_Q(t)}{\sqrt{8}} \{ \sin(2\pi(f_{LO} + f_Q)t - \phi) + \sin(2\pi(f_{LO} - f_Q)t + \phi) \}, \quad (\text{E.11})$$

This is trivial to check-out, by just considering the property: $\cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta)$. The second coupler after mixing with I and Q happens without phase shift and then the V_{IF} is:

$$V_{IF}(t) = V_{M1}(t) + V_{M2}(t) = \frac{A_{LO} A_Q(t)}{\sqrt{2}} \cos(2\pi(f_{LO} - f_{RF})t) \quad (\text{E.12})$$

where $\phi = \pi/2$, $f_I = f_Q = f_{IF}$ and $f_I = f_Q = f_{RF}$. This results are simple to simulate, the resulting spectrum with the sideband suppression is shown in fig. E.3

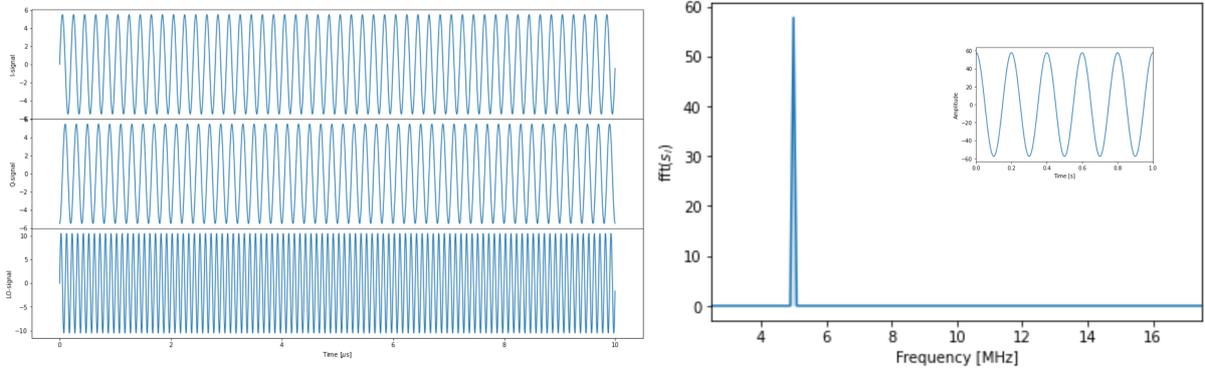


Figure E.3: **IQ mixing example.** Signals of the quadrature with 5MHz frequency and the LO with 10 MHz (left). Spectrum of the mixed signal showing sideband suppression and the time dependent signal (right).

In general, one can consider non-linear effects given by the intermodulation distortions that introduce spurious frequencies with different tones in terms of the input frequencies and amplitudes that change the contribution of each frequency on the spectrum. In this case, we can consider the output signal modeled by an Taylor expansion up to third order where the first order is related to the DC offset, the second term related to the attenuation and/or amplification of the signal and the third order giving the mixing of the signals.

The input signal includes all the input tones/frequencies. We will see two cases:

1. LO frequency and the modulation frequency (RF) with the same amplitude but different frequency.

2. RF signal has an imbalance in the amplitudes for I and Q i.e., unbalanced IQ-mixer.

The Taylor expansion for the output signal is given by:

$$I_o = a_0 + a_1 I_i + a_2 I_i^2 + a_3 I_i^3 + \dots \quad . \quad (\text{E.13})$$

In the first case the input signal is given by:

$$I_i = V_0 (\cos(2\pi f_1 t) + \cos(2\pi f_2 t)) \quad (\text{E.14})$$

and the output signal is:

$$\begin{aligned} I_o = & a_0 + a_1 V_0 \cos(2\pi f_1 t) + a_1 V_0 \cos(2\pi f_2 t) + \frac{a_2}{2} V_0^2 (1 + \cos(4\pi f_1 t)) + \\ & + \frac{a_2}{2} V_0^2 (1 + \cos(4\pi f_2 t)) + a_2 V_0^2 \cos(2\pi(f_2 - f_1)t) + a_2 V_0^2 \cos(2\pi(f_1 + f_2)t) + \mathcal{O}(I_i^3). \end{aligned} \quad (\text{E.15})$$

The numerical result of this analysis is shown in fig. E.4.

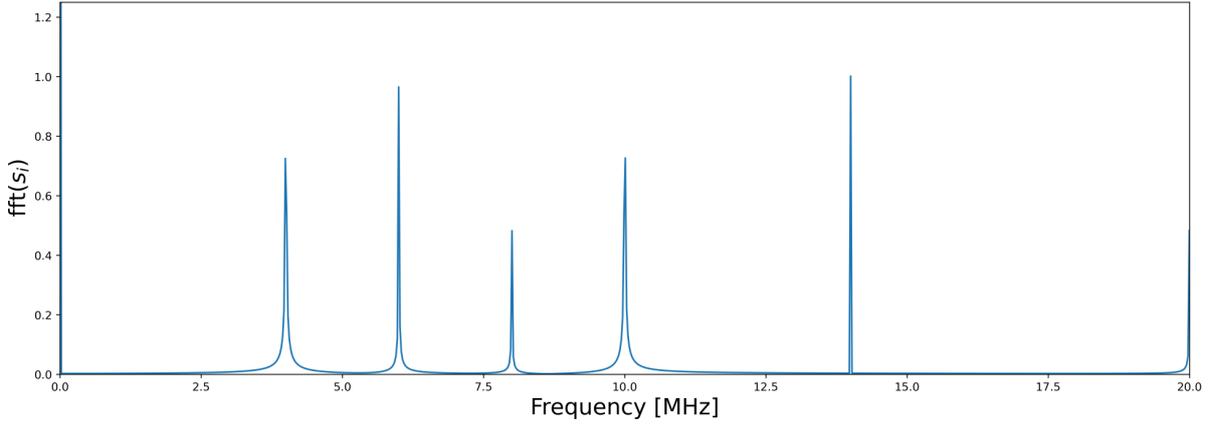


Figure E.4: **IQ mixing higher orders.** Non-linear effects on the IQ-mixing, that consider input signals at 4MHz and a LO frequency of 10MHz. Here it is possible to observe the second harmonics and the difference of these secondary peaks with the carrier frequency.

UML's and flowcharts from AQiPT

In this appendix I added the flowchart of the main functions and classes of AQiPT extensively discussed in chapter 3 as complementary material for understanding the internal functioning of the software, also as part of the documentation of the software.

Unified modeling language (UML) is a general-purpose visual modeling language that provide a standard description of the design of a system. The UMLs are used in software to describe processes, code, structures, workflows and class diagrams. Here I used the case of class and component diagrams to depict the internal structure and functionality of the code.

For the module of `analysis` the associated UML is:

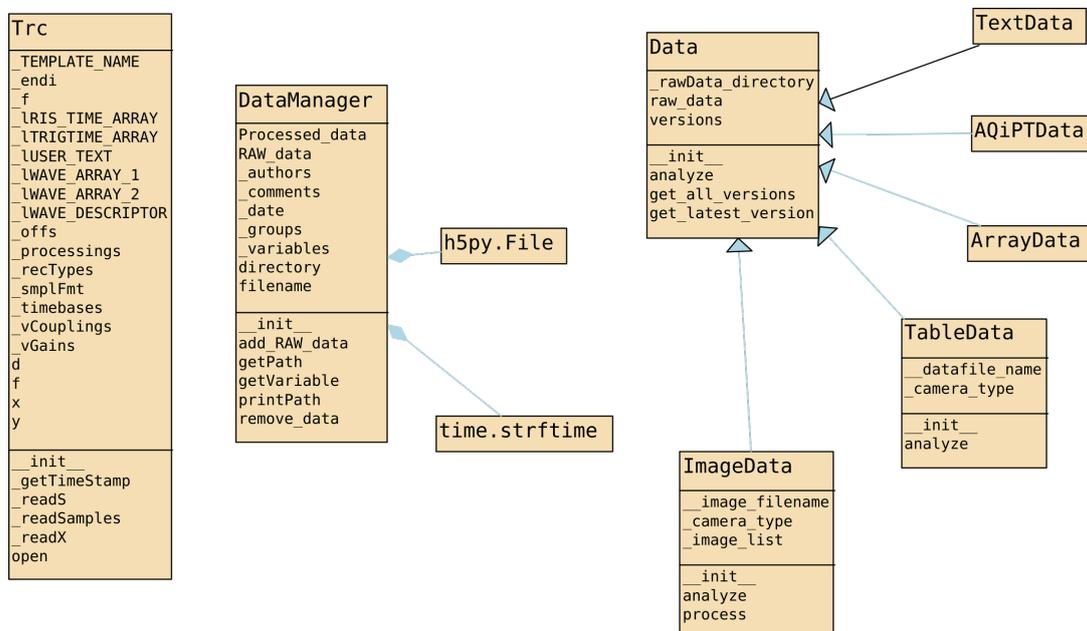


Figure F.1: UML analysis. My caption

For the module of `control` its UML is shown below:

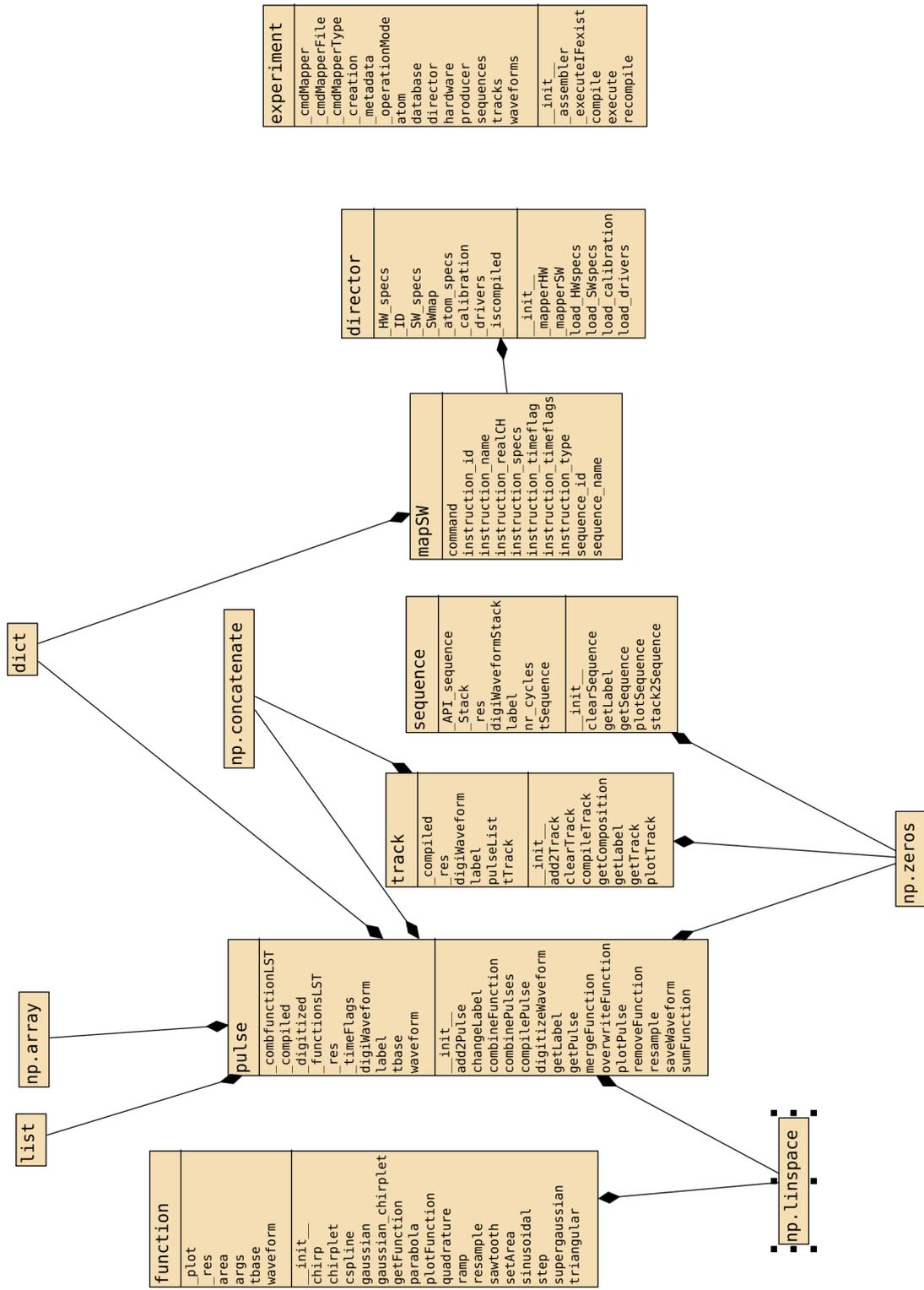


Figure F.2: UML control. My caption

Additionally, I present below the flowcharts of the functions of some of the classes of the modules that take part in the compilation process at different levels of the stack. The first compiler shown is the one used for creating the array of the `pulse` class of the `control` module.

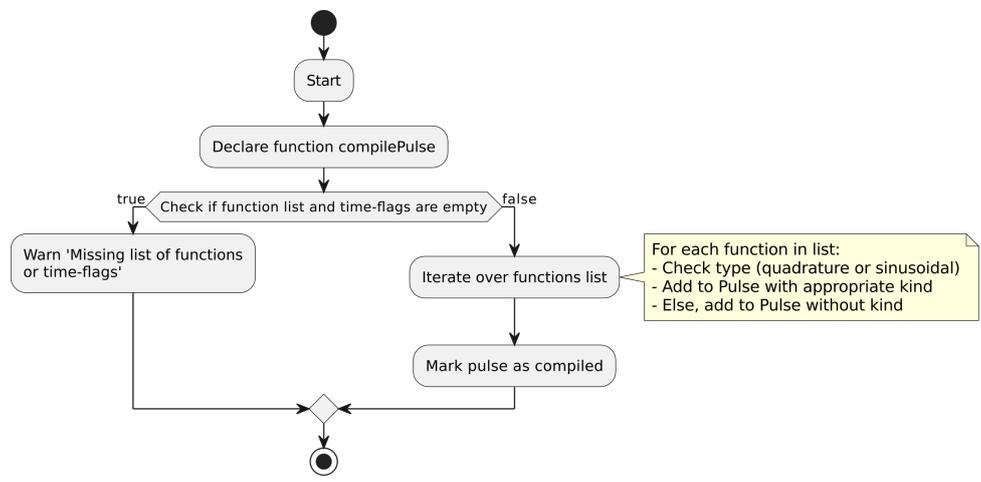


Figure F.5: `Pulse` compiler flowchart. My caption

The compiler for the `track` class of the `control` module is shown fig. F.6.

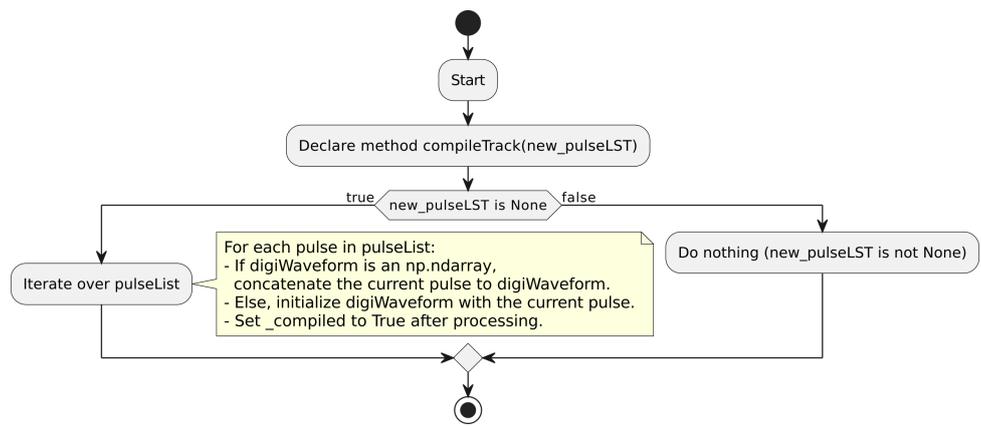


Figure F.6: `Track` compiler flowchart. My caption

The compiler that directly connects to the devices and instruments of the backend is the one that compiles the `experiment` class instance, this compiler is the one in charge to create the quantum assembly program (QAP) and uses two sub compilers embedded in the `director`, that generates the abstractions of the instruments with all properties and an instance of the driver linked to the instrument abstraction in Python.

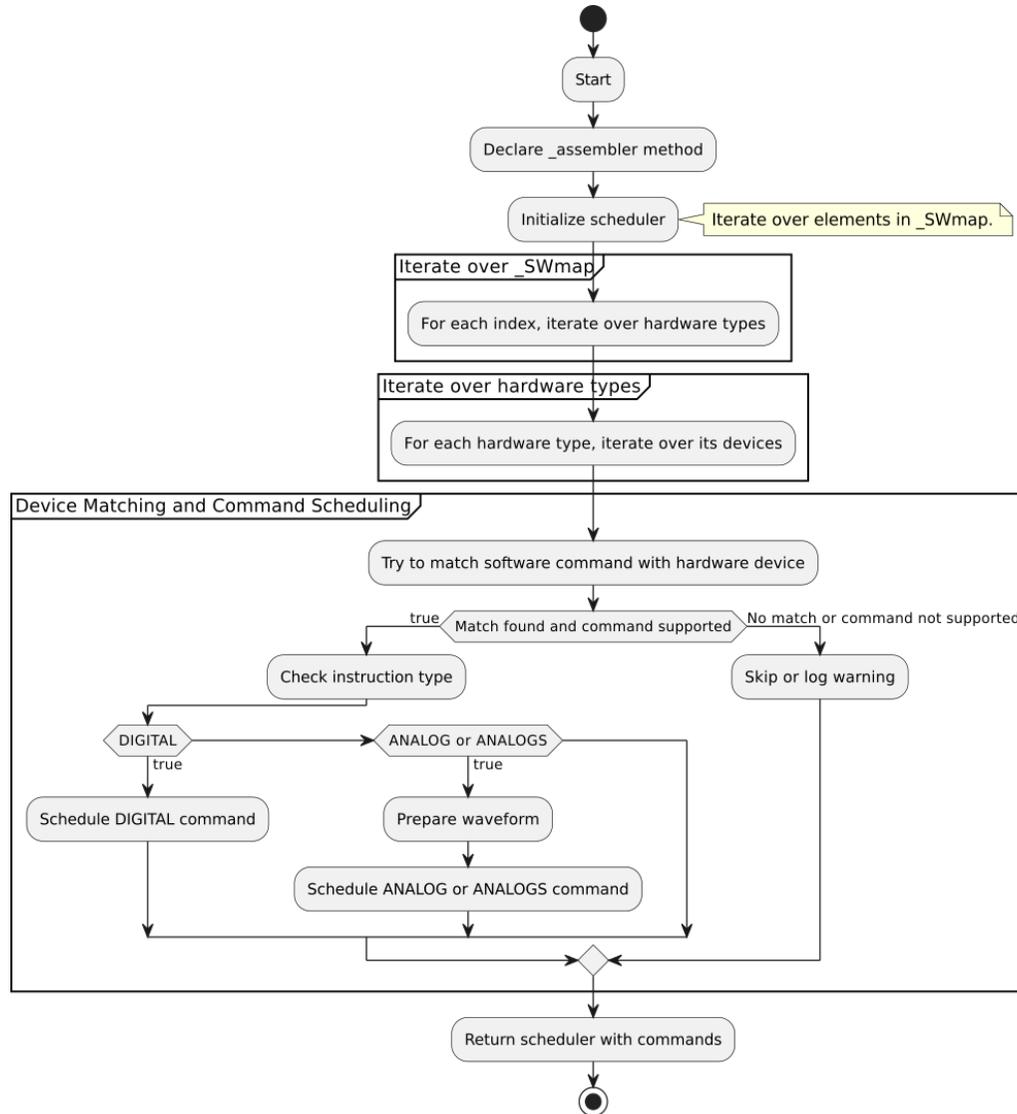


Figure F.7: Experiment compiler flowchart.

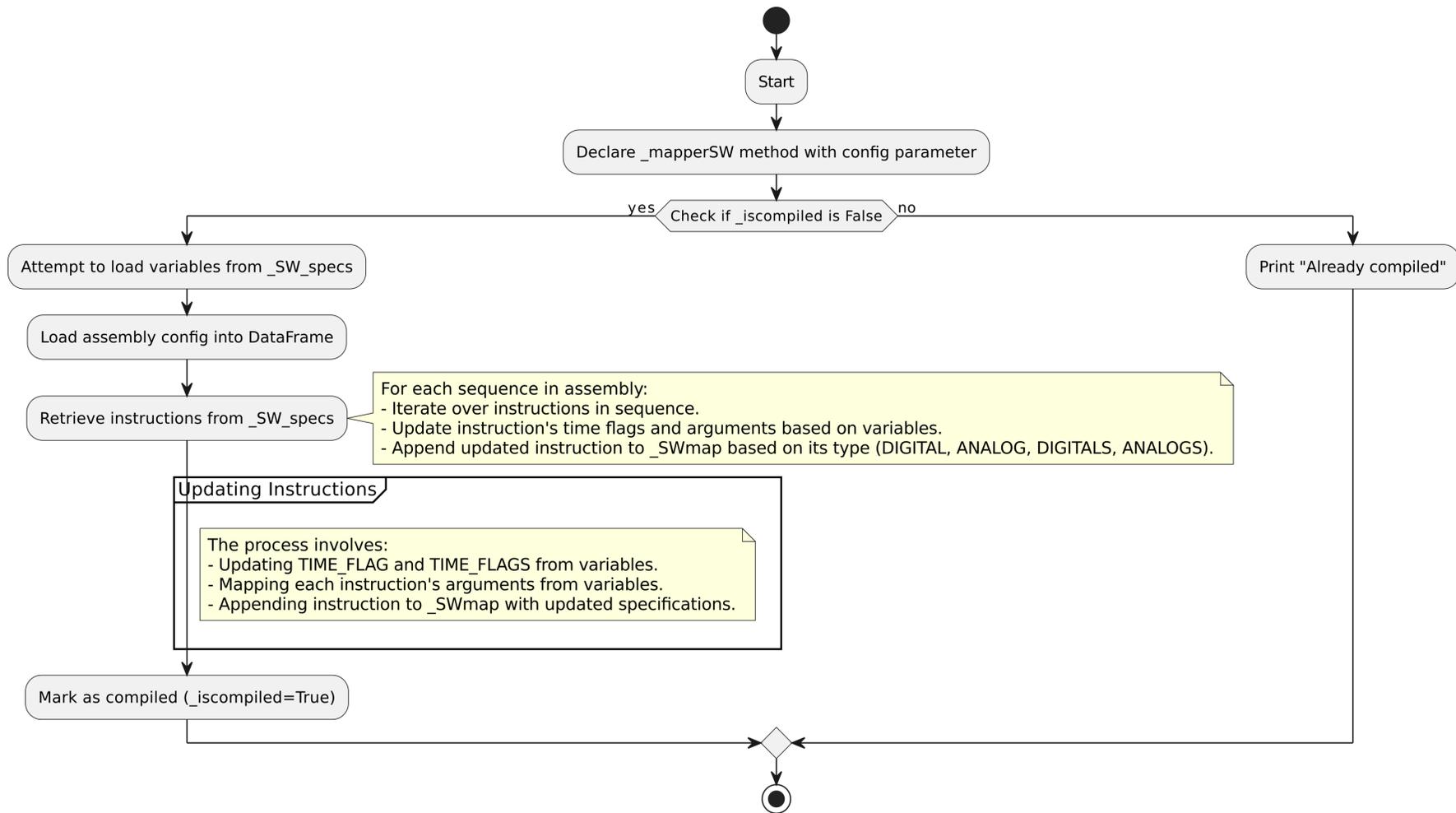


Figure F.8: **Director software mapper**. The subcompiler or as called in the software, the software mapper, is dedicated to pack the waveforms and instructions and unpack it in the Software specifications.

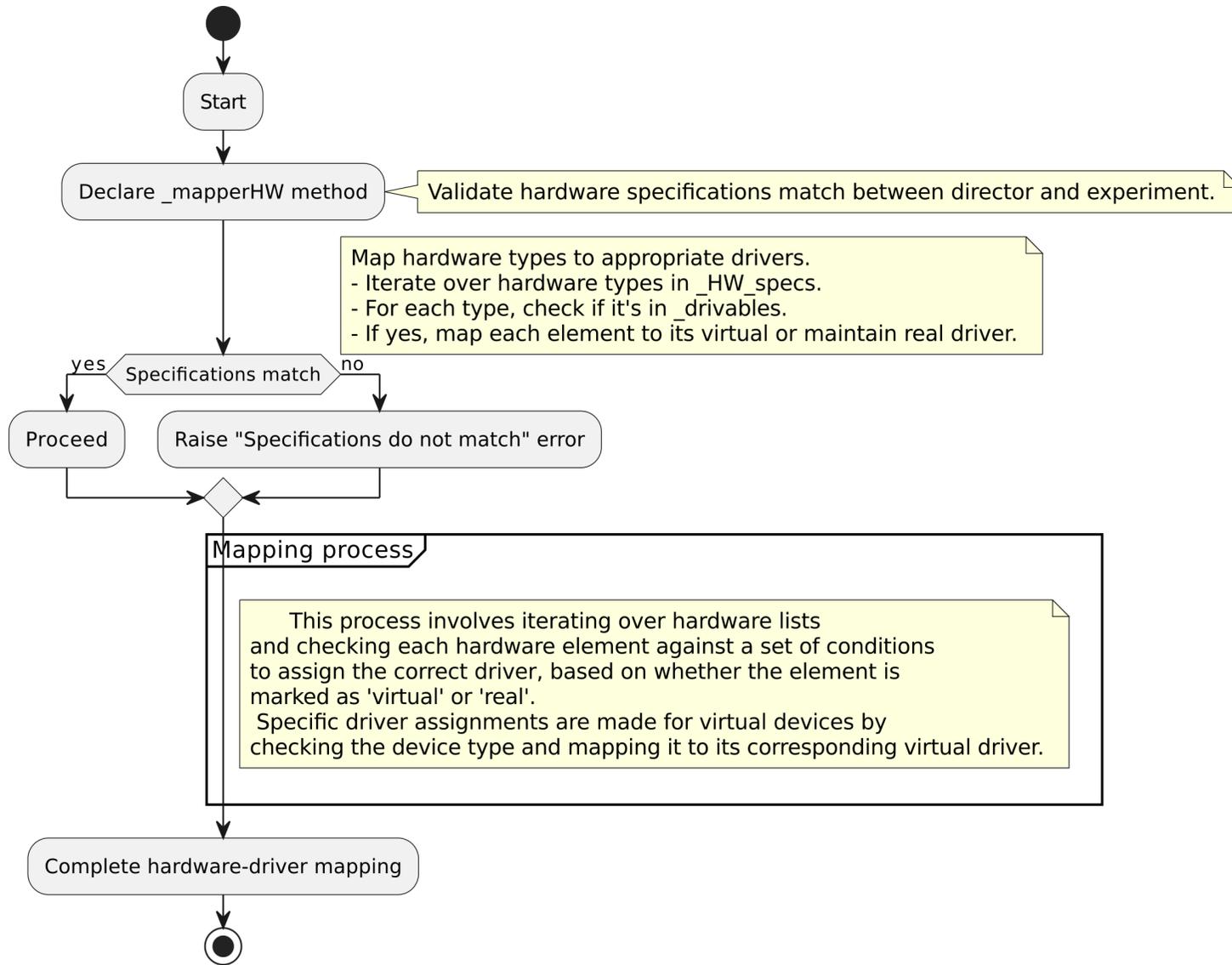


Figure F.9: **Director hardware mapper**. The subcompiler or as called in the software, the hardware mapper, is dedicated to pack the abstracted components of the backend into the Hardware specifications.

Figure F.10: Command `analog` from API.

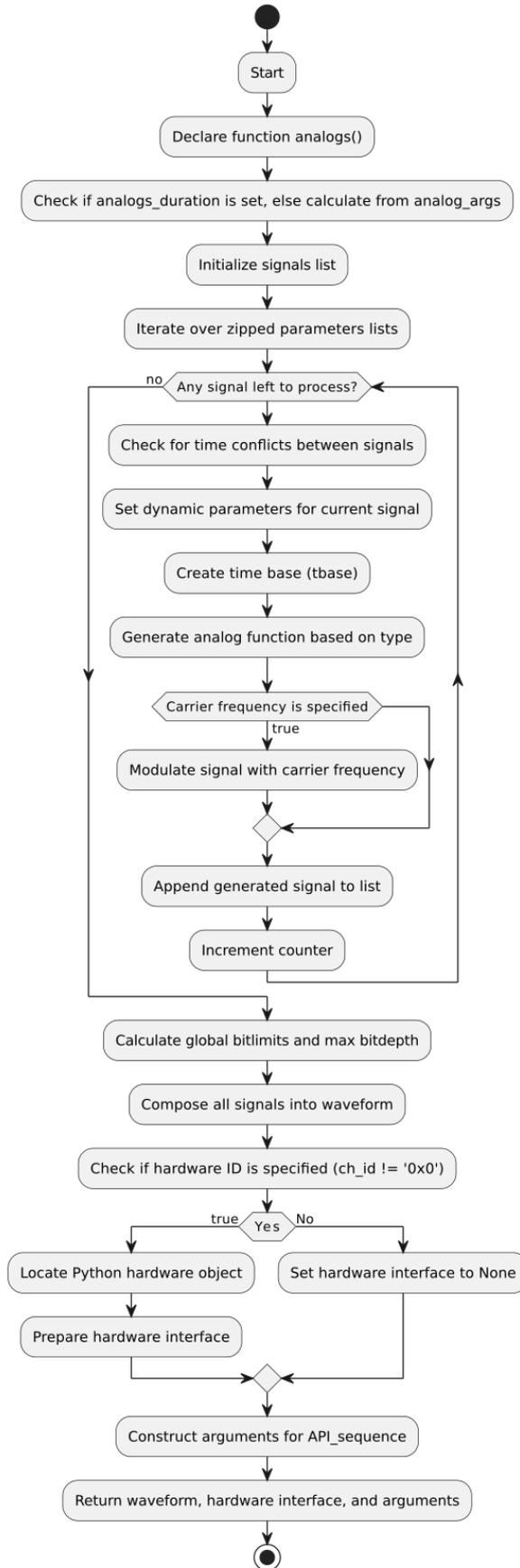


Figure F.11: Command analogs from API.

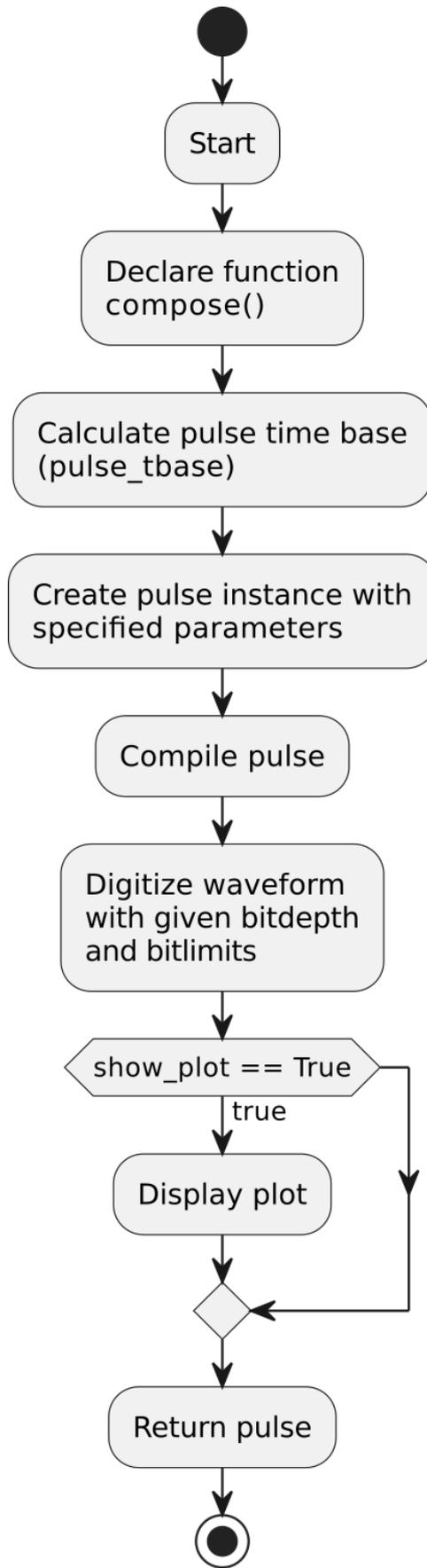


Figure F.12: Command `compose` from API.

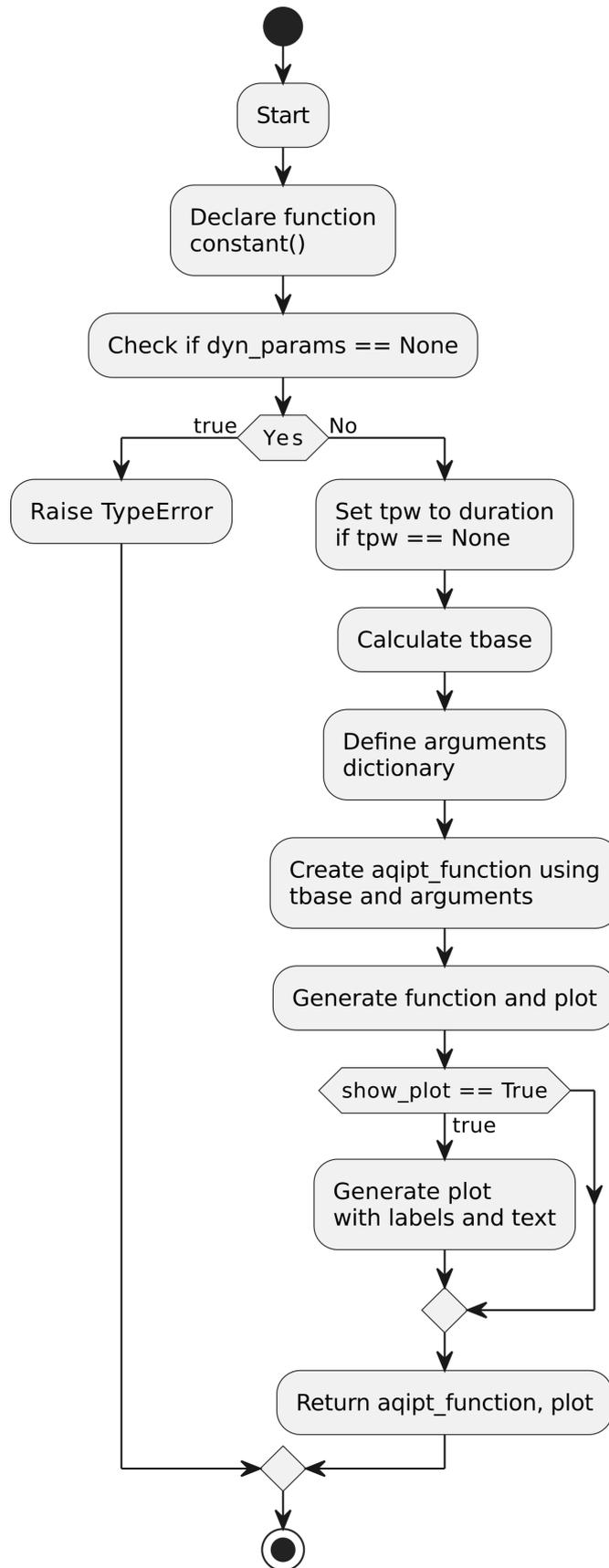


Figure F.13: Command `constant` from API.

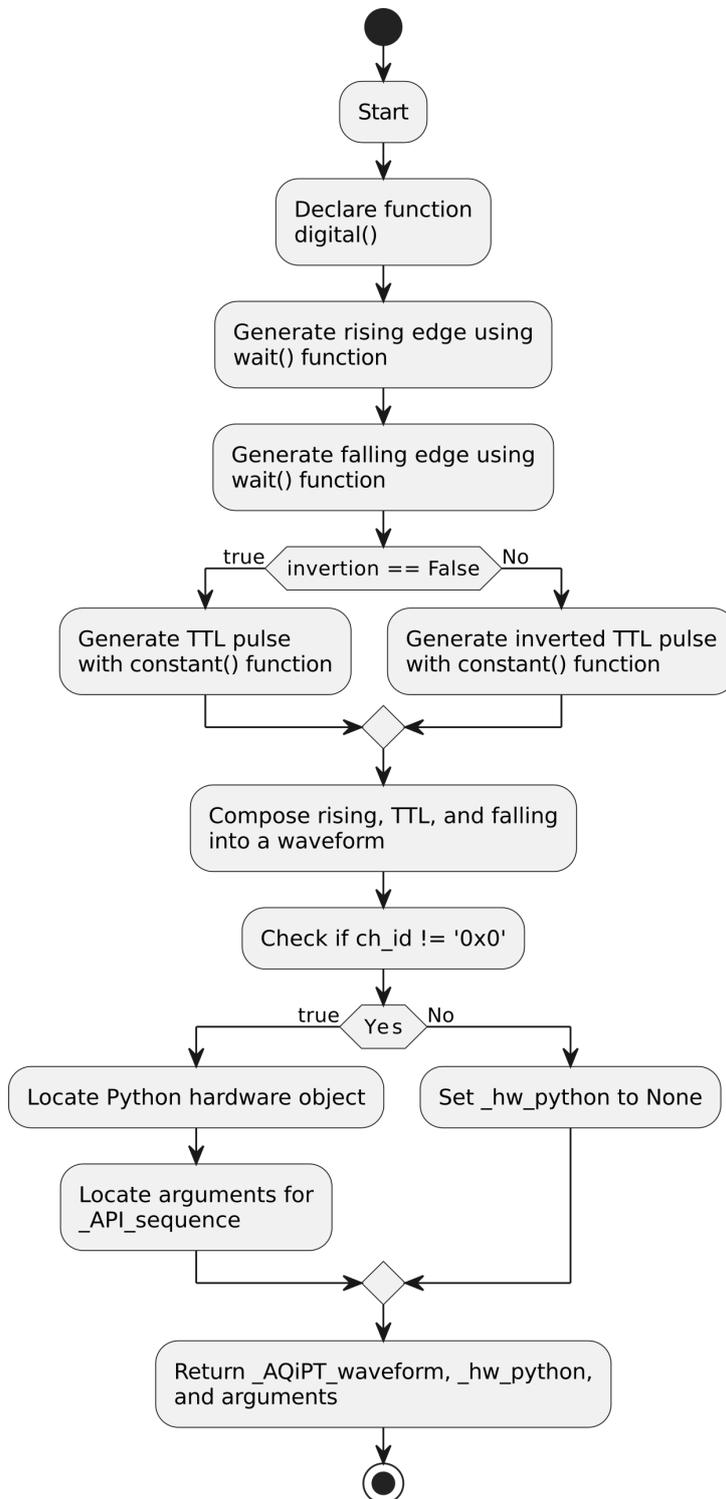


Figure F.14: Command `digital` from API.

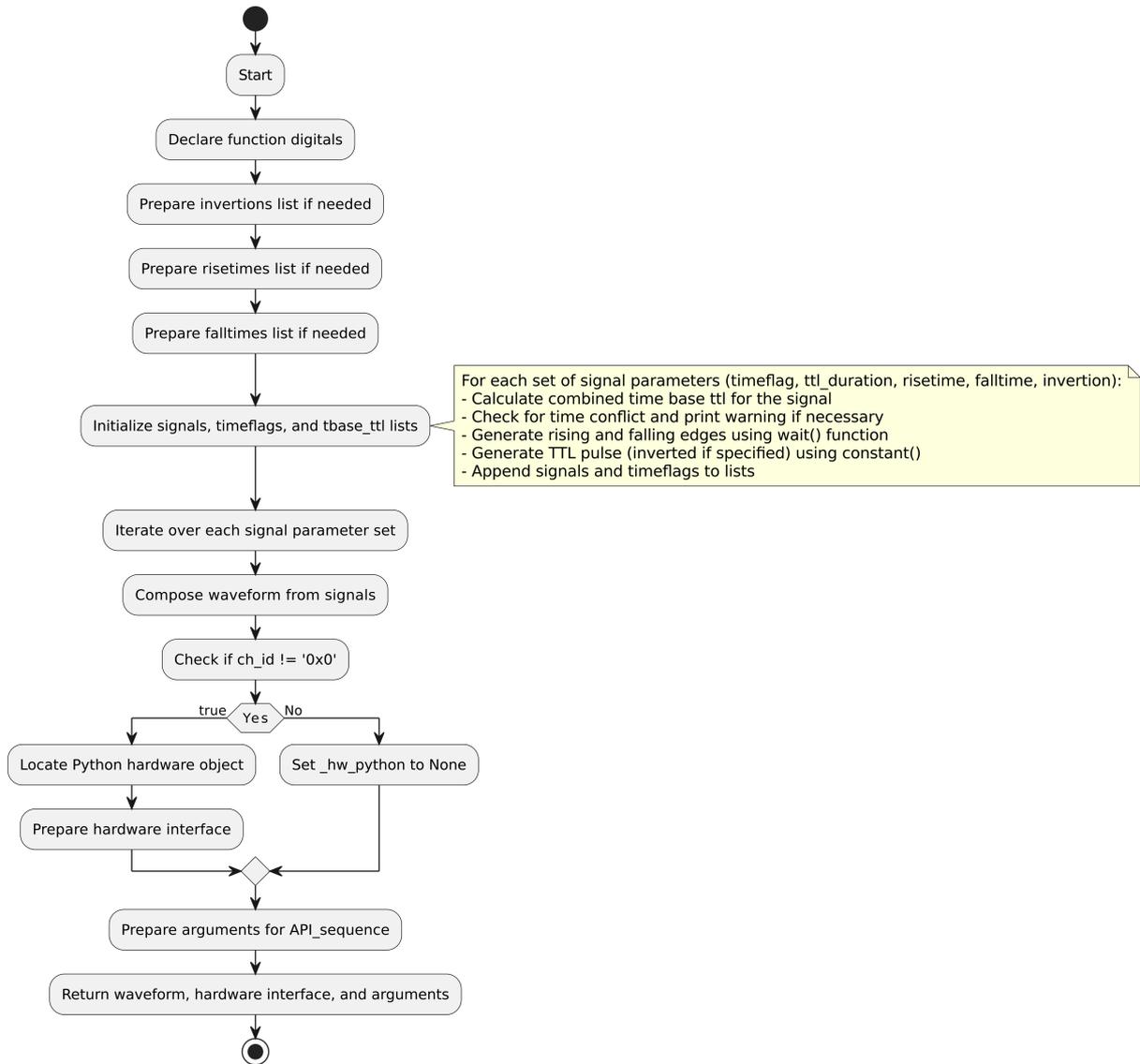


Figure F.15: Command digitals from API.

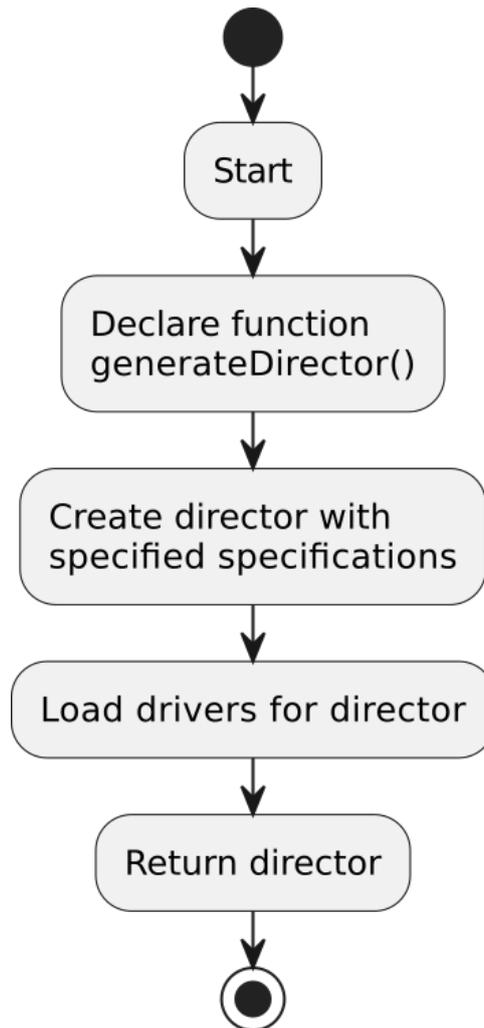


Figure F.16: Command `generateDirector` from API.

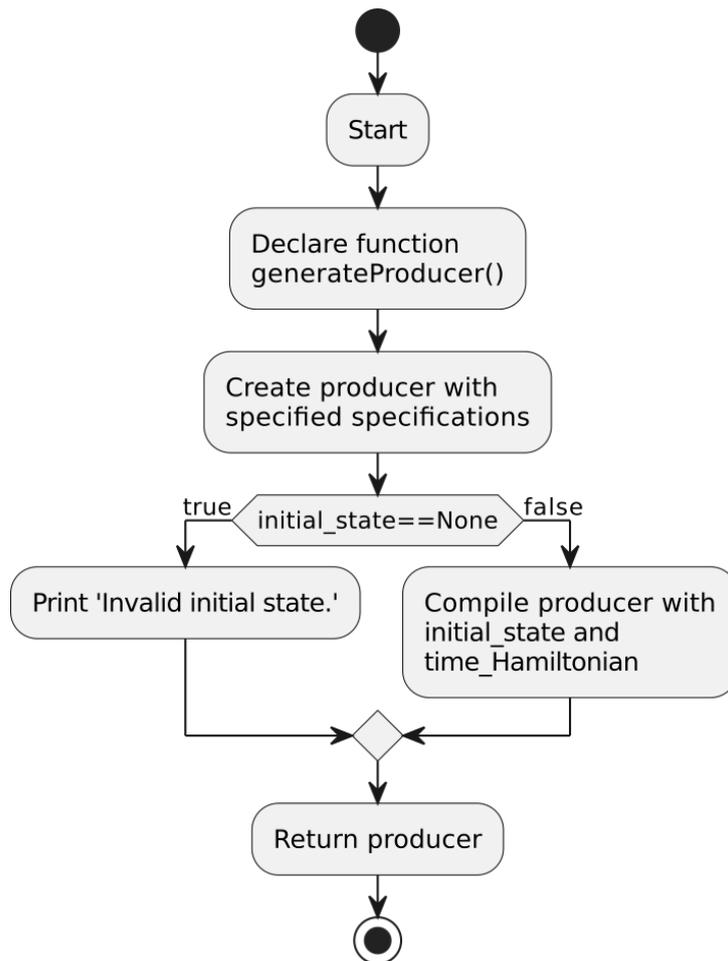


Figure F.17: Command `generateProducer` from API.

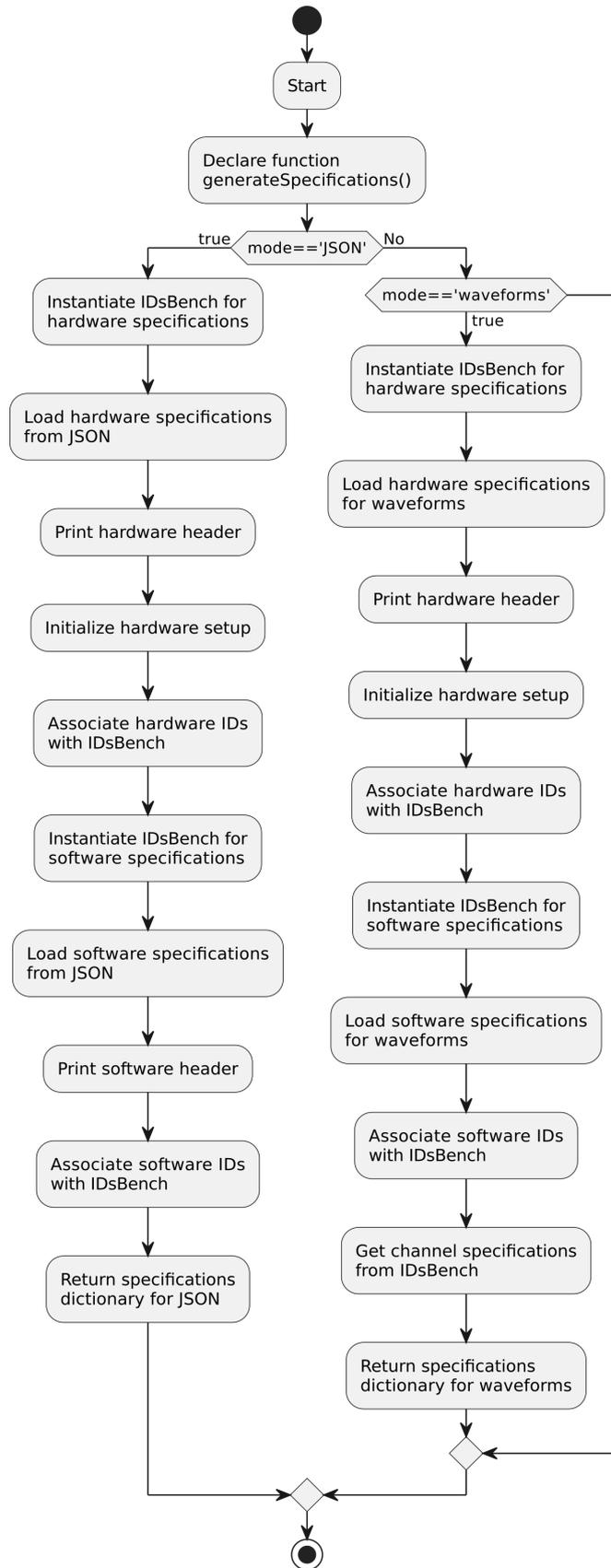


Figure F.18: Command `generateSpecifications` from API.

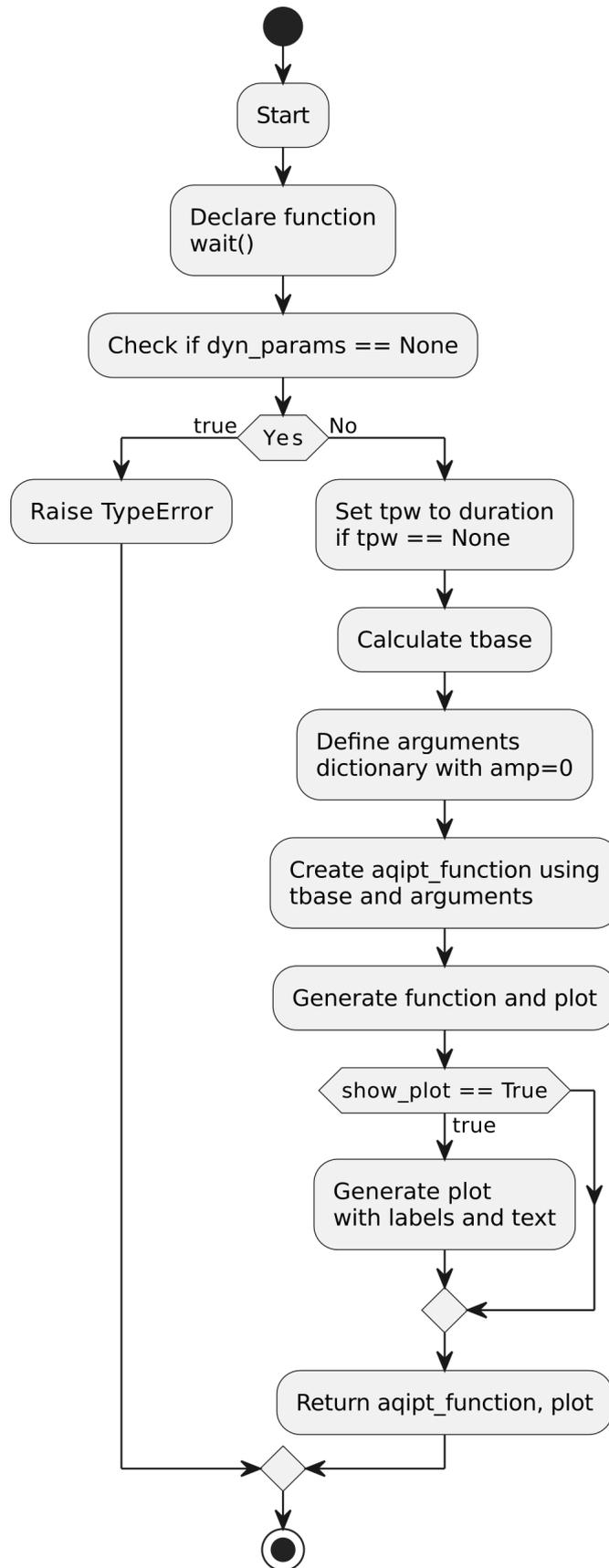


Figure F.19: Command `wait` from API.

The documentation of AQiPT is generated by Sphinx and available in the GitHub repository [Mor], see fig. F.20.

The screenshot shows the AQiPT documentation page. On the left is a dark blue sidebar with the AQiPT logo, a search bar labeled 'Search docs', and a 'Modules' section. The main content area is white and features a diagram of a quantum system with a central atom and various components like $\varepsilon(t)$, $\hat{H}(t)$, and $H^{\otimes n}$. Below the diagram is the title 'AQiPT: Atomic Quantum information Processing Toolbox' and a paragraph describing the tool. Further down are sections for 'Pre-requisites', 'Installation', and 'Development'.

Figure F.20: AQiPT documentation.

Bibliography

- [Alb18] Tameem Albash and Daniel A Lidar. “Adiabatic quantum computation”. In: *Reviews of Modern Physics* 90.1 (2018), p. 015002.
- [Álv23] Ángel Álvarez. *Desarrollo de software para la transpilación de operaciones lógicas nativas de qubits atómicos*. 2023.
- [Ana24] Shraddha Anand et al. “A dual-species Rydberg array”. In: *arXiv preprint arXiv:2401.10325* (2024).
- [And21] Loïc Anderegg et al. “Observation of microwave shielding of ultracold molecules”. In: *Science* 373.6556 (2021), pp. 779–782.
- [Aru19] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (2019), pp. 505–510.
- [Bal21] Harrison Ball et al. “Software tools for quantum control: Improving quantum computer performance through noise and error suppression”. In: *Quantum Science and Technology* 6.4 (2021), p. 044011.
- [Bar13] Rami Barends et al. “Coherent Josephson qubit suitable for scalable quantum integrated circuits”. In: *Physical review letters* 111.8 (2013), p. 080502.
- [Bar15] Daniel Barredo et al. “Coherent excitation transfer in a spin chain of three Rydberg atoms”. In: *Physical review letters* 114.11 (2015), p. 113002.
- [Bar18] Panagiotis Barkoutsos et al. “Quantum chemistry algorithms for efficient quantum computing”. In: *APS March Meeting Abstracts*. Vol. 2018. 2018, E33–007.
- [Bar22] Katrina Barnes et al. “Assembly and coherent control of a register of nuclear spin qubits”. In: *Nature Communications* 13.1 (2022), p. 2779.
- [Ber17a] Hannes Bernien et al. “Probing many-body dynamics on a 51-atom quantum simulator”. In: *Nature* 551.7682 (2017), pp. 579–584.
- [Ber17b] Hannes Bernien et al. “Probing many-body dynamics on a 51-atom quantum simulator”. In: *Nature* 551.7682 (2017), pp. 579–584.
- [Ber93] Ethan Bernstein and Umesh Vazirani. “Quantum complexity theory”. In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*. 1993, pp. 11–20.
- [Blu24] Dolev Bluvstein et al. “Logical quantum processor based on reconfigurable atom arrays”. In: *Nature* 626.7997 (2024), pp. 58–65.
- [Boh16] Justin G Bohnet et al. “Quantum spin dynamics and entanglement generation with hundreds of trapped ions”. In: *Science* 352.6291 (2016), pp. 1297–1301.

- [Bow19] William Bowden et al. “A pyramid MOT with integrated optical cavities as a cold atom platform for an optical lattice clock”. In: *Scientific reports* 9.1 (2019), p. 11704.
- [Bri09] Hans J Briegel et al. “Measurement-based quantum computation”. In: *Nature Physics* 5.1 (2009), pp. 19–26.
- [Bri12] Joseph W Britton et al. “Engineered two-dimensional Ising interactions in a trapped-ion quantum simulator with hundreds of spins”. In: *Nature* 484.7395 (2012), pp. 489–492.
- [Bur23] Guido Burkard et al. “Semiconductor spin qubits”. In: *Reviews of Modern Physics* 95.2 (2023), p. 025003.
- [Che22] Yeelai Chew et al. “Ultrafast energy exchange between two single Rydberg atoms on a nanosecond timescale”. In: *Nature Photonics* 16.10 (2022), pp. 724–729.
- [Che23] Liangyu Chen et al. “Transmon qubit readout fidelity at the threshold for quantum error correction without a quantum-limited amplifier”. In: *npj Quantum Information* 9.1 (2023), p. 26.
- [Chu36] Alonzo Church. “A note on the Entscheidungsproblem”. In: *The journal of symbolic logic* 1.1 (1936), pp. 40–41.
- [Cir12] J. Cirac and Peter Zoller. “Goals and opportunities in quantum simulation”. In: *Nature Physics* 8 (Apr. 2012), pp. 264–266. DOI: 10.1038/nphys2275.
- [Cir95] Juan I Cirac and Peter Zoller. “Quantum computations with cold trapped ions”. In: *Physical review letters* 74.20 (1995), p. 4091.
- [Con22] Iris Cong et al. “Hardware-efficient, fault-tolerant quantum computation with Rydberg atoms”. In: *Physical Review X* 12.2 (2022), p. 021049.
- [Cro19] Andrew W Cross et al. “Validating quantum computers using randomized model circuits”. In: *Physical Review A* 100.3 (2019), p. 032328.
- [Deu85] David Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.
- [DiV00] David P. DiVincenzo. “The Physical Implementation of Quantum Computation”. In: *Fortschritte der Physik* 48.9-11 (2000), pp. 771–783.
- [Eft21] Stavros Efthymiou et al. “Qibo: a framework for quantum simulation with hardware acceleration”. In: *Quantum Science and Technology* 7.1 (2021), p. 015018. DOI: 10.1088/2058-9565/ac39f5. URL: <https://doi.org/10.1088/2058-9565/ac39f5>.
- [Elt16] Amira M Eltony et al. “Technologies for trapped-ion quantum information systems: Progress toward scalability with hybrid systems”. In: *Quantum information processing* 15 (2016), pp. 5351–5383.
- [EuR] EuRyQa. *European Infrastructure for Rydberg Quantum Computing*. URL: euryqa.eu.
- [Eve23] Simon J Evered et al. “High-fidelity parallel entangling gates on a neutral-atom quantum computer”. In: *Nature* 622.7982 (2023), pp. 268–272.

- [Fey82] Richard P Feynman. “Simulating physics with computers”. In: *International journal of theoretical physics* 21.6/7 (1982), pp. 467–488.
- [Fey84] Richard P Feynman. “Quantum mechanical computers”. In: *Conference on Lasers and Electro-Optics*. Optica Publishing Group. 1984, TUAA2.
- [Fit17] Joseph F Fitzsimons and Elham Kashefi. “Unconditionally verifiable blind quantum computation”. In: *Physical Review A* 96.1 (2017), p. 012303.
- [Fow12] Austin G Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (2012), p. 032324.
- [Fu17] Xiang Fu et al. “An experimental microarchitecture for a superconducting quantum processor”. In: *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 2017, pp. 813–825.
- [Gaë10] A Gaëtan et al. “Analysis of the entanglement between two individual atoms using global Raman rotations”. In: *New Journal of Physics* 12.6 (2010), p. 065040.
- [Gam20] Filippo Maria Gambetta et al. “Engineering nonbinary Rydberg interactions via phonons in an optical lattice”. In: *Physical Review Letters* 124.4 (2020), p. 043402.
- [Ghe21] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. “Verification of quantum computation: an overview of existing approaches”. In: *Information Theory: Poincaré Seminar 2018*. Springer. 2021, pp. 113–209.
- [Gla17] Alexander W Glaetzle et al. “A coherent quantum annealer with Rydberg atoms”. In: *Nature communications* 8.1 (2017), p. 15813.
- [Got99] Daniel Gottesman and Isaac L Chuang. “Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations”. In: *Nature* 402.6760 (1999), pp. 390–393.
- [Gra19] TM Graham et al. “Rydberg-mediated entanglement in a two-dimensional neutral atom qubit array”. In: *Physical review letters* 123.23 (2019), p. 230501.
- [Gra22] TM Graham et al. “Multi-qubit entanglement and algorithms on a neutral-atom quantum computer”. In: *Nature* 604.7906 (2022), pp. 457–462.
- [Gra23a] TM Graham, E Oh, and M Saffman. “Multiscale architecture for fast optical addressing and control of large-scale qubit arrays”. In: *Applied Optics* 62.12 (2023), pp. 3242–3251.
- [Gra23b] TM Graham et al. “Midcircuit measurements on a single-species neutral alkali atom quantum processor”. In: *Physical Review X* 13.4 (2023), p. 041051.
- [Gri17] Michael Grieves and John Vickers. “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems”. In: *Transdisciplinary perspectives on complex systems: New findings and approaches* (2017), pp. 85–113.
- [Gro96] Lov K Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.
- [Gyg24] Flavien Gyger et al. “Continuous operation of large-scale atom arrays in optical lattices”. In: *arXiv preprint arXiv:2402.04994* (2024).

- [Hag12] JP Hague and C MacCormick. “Quantum simulation of electron–phonon interactions in strongly deformable materials”. In: *New Journal of Physics* 14.3 (2012), p. 033019.
- [Hog23] Craig W Hogle et al. “High-fidelity trapped-ion qubit operations with scalable photonic modulators”. In: *npj Quantum Information* 9.1 (2023), p. 74.
- [Höl24] Christian Hölzl et al. “Long-Lived Circular Rydberg Qubits of Alkaline-Earth Atoms in Optical Tweezers”. In: *arXiv preprint arXiv:2401.10625* (2024).
- [Isi23] Andrei Isichenko et al. “Photonic integrated beam delivery for a rubidium 3D magneto-optical trap”. In: *Nature communications* 14.1 (2023), p. 3080.
- [Jak00] Dieter Jaksch et al. “Fast quantum gates for neutral atoms”. In: *Physical Review Letters* 85.10 (2000), p. 2208.
- [Jan22] Sven Jandura and Guido Pupillo. “Time-optimal two-and three-qubit gates for Rydberg atoms”. In: *Quantum* 6 (2022), p. 712.
- [Jas48] Robert Jastrow. “On the rydberg-ritz formula in quantum mechanics”. In: *Physical Review* 73.1 (1948), p. 60.
- [Joh12] J Robert Johansson, Paul D Nation, and Franco Nori. “QuTiP: An open-source Python framework for the dynamics of open quantum systems”. In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772.
- [Kas20] Grzegorz Kaspowicz et al. “ArqX and sinara: Open software and hardware stacks for quantum physics”. In: *Quantum 2.0*. Optica Publishing Group, 2020, QTu8B–14.
- [Key20] Keysight. *SD1 3.x Software: for M31xxA Digitizers, M32xxA AWGs and M33xxA Combos*. 2020. URL: <https://www.keysight.com/us/en/lib/software-detail/instrument-firmware-software/sd1-3x-software-3120392.html>.
- [Kha20] Mohammadsadegh Khazali and Klaus Mølmer. “Fast multiqubit gates by adiabatic evolution in interacting excited-state manifolds of Rydberg atoms and superconducting circuits”. In: *Physical Review X* 10.2 (2020), p. 021054.
- [Kit97] A Yu Kitaev. “Quantum computations: algorithms and error correction”. In: *Russian Mathematical Surveys* 52.6 (1997), p. 1191.
- [Kob18] Takumi Kobayashi et al. “Uncertainty Evaluation of an 171 Yb Optical Lattice Clock at NMIJ”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 65.12 (2018), pp. 2449–2458.
- [Lan11] Andrew J Landahl, Jonas T Anderson, and Patrick R Rice. “Fault-tolerant quantum computing with color codes”. In: *arXiv preprint arXiv:1108.5738* (2011).
- [Lem24] Mariano Lemus et al. “Quantum Kolmogorov complexity and quantum correlations in deterministic-control quantum Turing machines”. In: *Quantum* 8 (2024), p. 1230.
- [Len93] Craig S Lent et al. “Quantum cellular automata”. In: *Nanotechnology* 4.1 (1993), p. 49.
- [Lev18] Harry Levine et al. “High-fidelity control and entanglement of Rydberg-atom qubits”. In: *Physical review letters* 121.12 (2018), p. 123603.

- [Lev19] Harry Levine et al. “Parallel implementation of high-fidelity multiqubit gates with neutral atoms”. In: *Physical review letters* 123.17 (2019), p. 170503.
- [Lia15] Yan Liang et al. “Shortcuts to adiabatic passage for multiqubit controlled-phase gate”. In: *Physical Review A* 91.3 (2015), p. 032304.
- [Lie18] Vincent Lienhard et al. “Observing the space-and time-dependent growth of correlations in dynamically tuned synthetic ising models with antiferromagnetic interactions”. In: *Physical Review X* 8.2 (2018), p. 021070.
- [Llo96] Seth Lloyd. “Universal quantum simulators”. In: *Science* 273.5278 (1996), pp. 1073–1078.
- [Luk18] Mikhail Lukin. “Probing many-body dynamics on a large-scale quantum simulator”. In: *APS March Meeting Abstracts*. Vol. 2018. 2018, P39–001.
- [Mac] QM Quantum Machines. *Quantum Machines Documentation*. URL: <https://docs.quantum-machines.co/1.1.7/>.
- [Mad20] Ivaylo S Madjarov et al. “High-fidelity entanglement and detection of alkaline-earth Rydberg atoms”. In: *Nature Physics* 16.8 (2020), pp. 857–861.
- [Man] HJ Manetsch et al. “A tweezer array with 6100 highly coherent atomic qubits (2024)”. In: *arXiv preprint arXiv:2403.12021* ().
- [Man24] Hannah J Manetsch et al. “A tweezer array with 6100 highly coherent atomic qubits”. In: *arXiv preprint arXiv:2403.12021* (2024).
- [Moh08] Masoud Mohseni, Ali T Rezakhani, and Daniel A Lidar. “Quantum-process tomography: Resource analysis of different strategies”. In: *Physical Review A* 77.3 (2008), p. 032322.
- [Mok20] Arezoo Mokhberi, Markus Hennrich, and Ferdinand Schmidt-Kaler. “Trapped Rydberg ions: A new platform for quantum information processing”. In: *Advances in Atomic, Molecular, and Optical Physics*. Vol. 69. Elsevier, 2020, pp. 233–306.
- [Mor] Manuel Morgado. *AQiPT*. URL: <https://github.com/manuelmorgado/AQiPT>.
- [Mor21] M Morgado and S Whitlock. “Quantum simulation and computing with Rydberg-interacting qubits”. In: *AVS Quantum Science* 3.2 (2021).
- [Nie01] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Vol. 2. Cambridge university press Cambridge, 2001.
- [Nie23] Jens Hedegaard Nielsen et al. *QCoDeS/Qcodes: QCoDeS 0.38.1*. Version v0.38.1. 2023. DOI: 10.5281/zenodo.7866908. URL: <https://doi.org/10.5281/zenodo.7866908>.
- [Nif20] Robert J Niffenegger et al. “Integrated multi-wavelength control of an ion qubit”. In: *Nature* 586.7830 (2020), pp. 538–542.
- [Nor24] MA Norcia et al. “Iterative assembly of $\{171\}$ Yb atom arrays in cavity-enhanced optical lattices”. In: *arXiv preprint arXiv:2401.16177* (2024).
- [Ock10] CF Ockeloen et al. “Detection of small atom numbers through image processing”. In: *Physical Review A* 82.6 (2010), p. 061606.

- [Orú19] Román Orús, Samuel Mugel, and Enrique Lizaso. “Quantum computing for finance: Overview and prospects”. In: *Reviews in Physics* 4 (2019), p. 100028.
- [Pag22] Alice Pagano et al. “Error budgeting for a controlled-phase gate with strontium-88 Rydberg atoms”. In: *Physical Review Research* 4.3 (2022), p. 033019.
- [Par08] E Paradis et al. “Observation of superfluorescent emissions from laser-cooled atoms”. In: *Physical Review A* 77.4 (2008), p. 043419.
- [Pau24] Lars Pause et al. “Supercharged two-dimensional tweezer array with more than 1000 atomic qubits”. In: *Optica* 11.2 (2024), pp. 222–226.
- [Per14] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature communications* 5.1 (2014), p. 4213.
- [Phi22] Stephan GJ Philips et al. “Universal control of a six-qubit quantum processor in silicon”. In: *Nature* 609.7929 (2022), pp. 919–924.
- [Pic18] CJ Picken et al. “Entanglement of neutral-atom qubits with long ground-Rydberg coherence times”. In: *Quantum Science and Technology* 4.1 (2018), p. 015011.
- [PIH] PIHeidelberg. *Logicbox*. URL: http://ew-dev.physi.uni-heidelberg.de/~rubio/LogicBox/index.php/Main_Page.
- [Pło18] Marcin Płodzień, Tomasz Sowiński, and Servaas Kokkelmans. “Simulating polaron biophysics with Rydberg atoms”. In: *Scientific Reports* 8.1 (2018), p. 9247.
- [Pre18] John Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.
- [Qis23] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.
- [Ras20] SE Rasmussen et al. “Single-step implementation of high-fidelity n-bit Toffoli gates”. In: *Physical Review A* 101.2 (2020), p. 022308.
- [Rav23] Brice Ravon et al. “Array of individual circular rydberg atoms trapped in optical tweezers”. In: *Physical Review Letters* 131.9 (2023), p. 093401.
- [Ray23] Jack Raymond et al. “Hybrid Quantum Annealing for Larger-than-QPU Lattice-structured Problems”. In: *ACM Transactions on Quantum Computing* 4.3 (2023). DOI: 10.1145/3579368. URL: <https://doi.org/10.1145/3579368>.
- [Roy24] Tania Roy. *A 2D route to 3D computer chips*. 2024.
- [Saf10] Mark Saffman, Thad G Walker, and Klaus Mølmer. “Quantum information with Rydberg atoms”. In: *Reviews of modern physics* 82.3 (2010), p. 2313.
- [Saf16] Mark Saffman. “Quantum computing with atomic qubits and Rydberg interactions: progress and challenges”. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 49.20 (2016), p. 202001.
- [Sch03] Ferdinand Schmidt-Kaler et al. “Realization of the Cirac–Zoller controlled-NOT quantum gate”. In: *Nature* 422.6930 (2003), pp. 408–411.
- [Sch18] VM Schäfer et al. “Fast quantum logic gates with trapped-ion qubits”. In: *Nature* 555.7694 (2018), pp. 75–78.

- [Sch20] Kai-Niklas Schymik et al. “Enhanced atom-by-atom assembly of arbitrary tweezer arrays”. In: *Physical Review A* 102.6 (2020), p. 063107.
- [Sch22] Lars Schreiber and Lotte Geck. “Sechs auf einen Streich Ebenen siliziumbasierte Quantencomputer den Weg zur Skalierbarkeit?” In: *Nature* 609.FZJ-2022-04162 (2022).
- [Sec16] Thomas Secker et al. “Controlled long-range interactions between Rydberg atoms and ions”. In: *Physical Review A* 94.1 (2016), p. 013420.
- [Sha23] Ariya Shajii et al. “Codon: A compiler for high-performance pythonic applications and dsls”. In: *Proceedings of the 32nd ACM SIGPLAN International Conference on Compiler Construction*. 2023, pp. 191–202.
- [She19] Cai-Peng Shen et al. “Construction of robust Rydberg controlled-phase gates”. In: *Optics Letters* 44.8 (2019), pp. 2036–2039.
- [She21] Sayali Shevate. “Preparation and Rydberg excitation of large arrays of microscopic atomic ensembles”. PhD thesis. Université de Strasbourg, 2021.
- [Shi18] Xiao-Feng Shi. “Deutsch, Toffoli, and CNOT gates via Rydberg blockade of neutral atoms”. In: *Physical Review Applied* 9.5 (2018), p. 051001.
- [Sho94] Peter W Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.
- [Šib17] Nikola Šibalić et al. “ARC: An open-source library for calculating properties of alkali Rydberg atoms”. In: *Computer Physics Communications* 220 (2017), pp. 319–331.
- [Sig21] Adrien Signoles et al. “Glassy dynamics in a disordered Heisenberg quantum spin system”. In: *Physical Review X* 11.1 (2021), p. 011011.
- [Sin23] Arpan Singh. *Experimental Characterization and Numerical Modelling of the Interfacial Heat Transfer Coefficient in Hot Stamping of Al-Si Coated 22MnB5 Steel*. 2023.
- [Ska20] Rasmus Vestergaard Skannrup, R Gerritsma, and SJJMF Kokkelmans. “Phonon-mediated spin-spin interactions between trapped Rydberg atoms”. In: *arXiv preprint arXiv:2008.13622* (2020).
- [Sta13] P. T. Starkey et al. “A scripted control system for autonomous hardware-timed experiments”. In: *Review of Scientific Instruments* 84.8 (2013), p. 085111. DOI: 10.1063/1.4817213. eprint: <https://doi.org/10.1063/1.4817213>. URL: <https://doi.org/10.1063/1.4817213>.
- [Sto06] Neil Storey. *Electronics: a systems approach*. Pearson Education, 2006.
- [Tur36] Alan Mathison Turing et al. “On computable numbers, with an application to the Entscheidungsproblem”. In: *J. of Math* 58.345-363 (1936), p. 5.
- [Vog09] Felix Vogt. “Creation of cold and dense ensembles of calcium atoms”. In: (2009).
- [Wac21] Andrew Wack et al. “Quality, speed, and scale: three key attributes to measure the performance of near-term quantum computers”. In: *arXiv preprint arXiv:2110.14108* 105 (2021), pp. 112–147.

- [Wan20] Yibo Wang et al. “Preparation of hundreds of microscopic atomic ensembles in optical tweezer arrays”. In: *npj Quantum Information* 6.1 (2020), p. 54.
- [Wan21] Pengfei Wang et al. “Single ion qubit with estimated coherence time exceeding one hour”. In: *Nature communications* 12.1 (2021), p. 233.
- [Wat18] TF Watson et al. “A programmable two-qubit quantum processor in silicon”. In: *nature* 555.7698 (2018), pp. 633–637.
- [Wei01] Yaakov S Weinstein et al. “Implementation of the quantum Fourier transform”. In: *Physical review letters* 86.9 (2001), p. 1889.
- [Wer07] J Werschnik and EKV Gross. “Quantum optimal control theory”. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 40.18 (2007), R175.
- [Whi23] S Whitlock and TD Kieu. “Quantum Factoring Algorithm using Grover Search”. In: *arXiv preprint arXiv:2312.10054* (2023).
- [Wil10] Tatjana Wilk et al. “Entanglement of two individual neutral atoms using Rydberg blockade”. In: *Physical review letters* 104.1 (2010), p. 010502.
- [WIN16] Tobias Martin WINTERMANTEL. *In-situ fluorescence imaging of ultracold potassium in an optical trap*. Thesis. 2016.
- [Win20] Tobias M Wintermantel et al. “Unitary and nonunitary quantum cellular automata with Rydberg arrays”. In: *Physical review letters* 124.7 (2020), p. 070503.
- [Win21a] TM Wintermantel et al. “Epidemic growth and Griffiths effects on an emergent network of excited atoms”. In: *Nature Communications* 12.1 (2021), p. 103.
- [Win21b] Tobias Martin Wintermantel. “Complex systems dynamics in laser excited ensembles of Rydberg atoms”. PhD thesis. Université de Strasbourg & Universität Heidelber, 2021.
- [Wüs11] S Wüster et al. “Excitation transport through Rydberg dressing”. In: *New Journal of Physics* 13.7 (2011), p. 073044.
- [Zen17] Yong Zeng et al. “Entangling two individual atoms of different isotopes via Rydberg blockade”. In: *Physical Review Letters* 119.16 (2017), p. 160502.
- [Zha20] Chi Zhang et al. “Submicrosecond entangling gate between trapped ions via Rydberg interaction”. In: *Nature* 580.7803 (2020), pp. 345–349.
- [Zha24] Bichen Zhang et al. “Scaled local gate controller for optically addressed qubits”. In: *Optica* 11.2 (2024), pp. 227–233.
- [Zür23] Zürich. *Bloqade.jl: Package for the quantum computation and quantum simulation based on the neutral-atom architecture*. 2023. URL: <https://github.com/QuEraComputing/Bloqade.jl/>.