MULT90063 Introduction to Quantum Computing

Lab Session 12

12.1 Introduction

Welcome to Lab 12 of MULT90063 Introduction to Quantum Computing. The purpose of this lab session is to get some experience with the HHL (Harrow, Hassidim, Lloyd) algorithm for solving linear equations.

In this lab we will set up one instance of solving a set of linear equations. First we will solve the system of equations using classical techniques (i.e. you solve them), and then we will implement the quantum circuit for solving this set of equations in the QUI. You may to do your algebraic calculations on a separate piece of paper, but where indicated write you answers here so we can easily check your progress.

12.2 Solving the set of linear equations classically

The set of linear equations which we will be solving in this lab is:

$$
\begin{aligned}
+15w + 9x + 5y - 3z &= 4 \\
+9w + 15x + 3y - 5z &= 4 \\
+5w + 3x + 15y - 9z &= 4 \\
-3w - 5x - 9y + 15z &= 4
\end{aligned}
$$

Exercise 12.2.1 Write this set of equations as a matrix equation, $A\vec{x} = \vec{b}$ where $\vec{x} = (w, x, y, z)^T$:

Exercise 12.2.2 Using any method of your choosing, solve the equations for w, x, y, z and write the solutions below:

Exercise 12.2.3 Find the eigenvalues, λ_i, and corresponding eigenvectors, $\vec{u}_i$ of A. Please normalize your eigenvectors so that the have a norm of 1, and make a valid quantum state. Write them out below:

Exercise 12.2.4 From the eigenvalues, λ_i, and eigenvectors, $\vec{u}_i$, find A^{-1}:

$$A^{-1} = \sum_i \frac{1}{\lambda_i} |u_i\rangle\langle u_i| = \sum_i \frac{1}{\lambda_i} \vec{u}_i (\vec{u}_i^*)^T$$

Exercise 12.2.5 Express b as a linear combination of the eigenvectors $\vec{u}_i$ of A, i.e. finding amplitudes β_i in the decomposition of the vector $\vec{b}$ in terms of the vectors $\vec{u}_i$:

$$\vec{b} = \sum_i \beta_i \vec{u}_i$$

Exercise 12.2.6 Solve the system of linear equations by equating $\vec{x} = A^{-1}\vec{b}$.

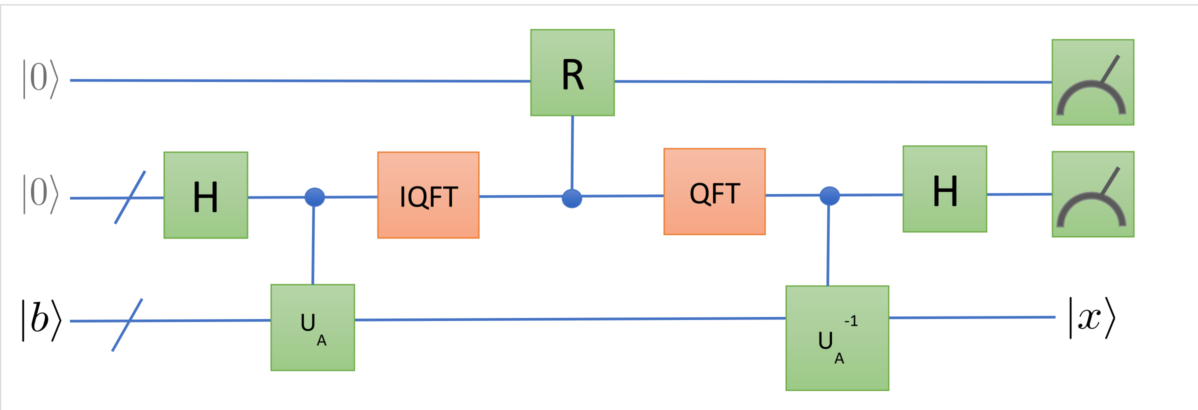
Exercise 12.2.7 Check that this answer can also be obtained by equating:

$$\vec{x} = \sum_i \frac{\beta_i}{\lambda_i} \vec{u}_i$$

12.3 Solving the system using HHL in the QUI

We will now program QUI solve the same set of linear equations in the HHL approach. You will need a total of 7 qubits, in order to solve the system of equations using HHL.

Recall from the lectures the basic HHL circuit is:



Here U represents a matrix of the form:

$$\exp\,(iAt)$$

The first qubit is an ancilla initialised to $|0\rangle$. The next 4 qubits form the "upper register" or "control register" which are initialised in an equal superposition. The "bottom register" of 2 qubits are initialised in the state corresponding to $\vec{b}/|\vec{b}| = \left(\hat{b}_0, \hat{b}_1, \hat{b}_2, \hat{b}_3\right)^T = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)^T$.

We now make the correspondence between this (nicely balanced) column 4-vector and the associated column 4-vector of *two* qubits (in a separable state in this instance)

$$|b\rangle = |q_1\rangle \otimes |q_2\rangle$$

where $|q_1\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|q_2\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. Make sure you can see how $|b\rangle$ represents $\vec{b}/|\vec{b}|$.

Exercise 12.3.1 Program the first step of the algorithm into the QUI, initializing the control/upper register and the lower register.

In the second step of the algorithm, we need to implement controlled rotations of the form,

$$\exp\,(iAt)$$

To see how to do this, we need to write out the matrix A as a linear combination of Pauli matrices.

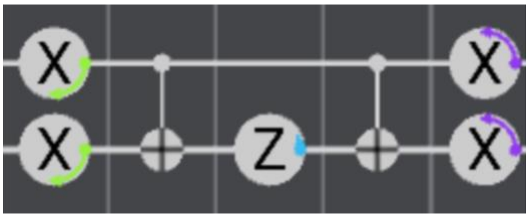
$$A = \begin{pmatrix} 15 & 9 & 5 & -3 \\ 9 & 15 & 3 & -5 \\ 5 & 3 & 15 & -9 \\ -3 & -5 & -9 & 15 \end{pmatrix} = 15\; I \otimes I + 9 Z \otimes X + 5 X \otimes Z + 3 Y \otimes Y$$

Note: Each term commutes with the other two (i.e. all inter-commute).

Exercise 12.3.2 Because each of these terms commute, we can consider each separately. i.e.

$$\begin{aligned} \exp(iAt) &= \exp(i[15\; I \otimes I + 9 Z \otimes X + 5 X \otimes Z + 3 Y \otimes Y]t) \\ &= \exp(i[15\; I \otimes I]t) \exp\,(i[3 Y \otimes Y]t) \exp\,(i[5 Z \otimes X]t)\, \exp\,(i[5 X \otimes Z]t) \end{aligned}$$

Verify, as far as you can using the QUI, that the following circuit makes a rotation by $\exp\,(i\theta Y \otimes Y)$:



Hint: refer back to Lecture 19 for the decomposition of exponentials of such operators, e.g.

$$\exp(i\theta Y \otimes Y) = \cos(\theta)\, I \otimes I + i \sin\,(\theta) Y \otimes Y$$

Use this to write out the effect of this sequence of gates on two qubit states (such as $|00\rangle$) which you can check in the QUI directly. Use a rotation angle of -0.1875π (and no global phase) in the z-rotation.

In the same manner, verify that this circuit (depending on what angles you put in for the X rotations) makes an operation, $\exp(i\theta Z \otimes X)$:

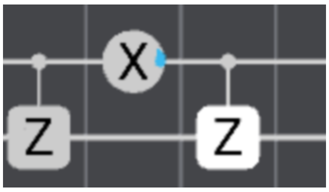
$$\exp(i\theta Z \otimes X) = \cos(\theta)\, I \otimes I + i \sin(\theta) Z \otimes X$$

Use an x-rotation angle of -0.3125π, and no global phase.

And that similarly that the following circuit makes an operation, $\exp(i\theta X \otimes Z)$:

$$\exp(i\theta X \otimes Z) = \cos(\theta)\, I \otimes I + i \sin(\theta) X \otimes Z$$

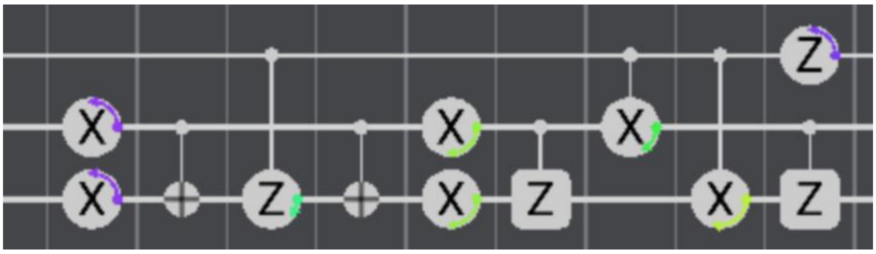
In this case, make the rotation angle -0.5625π, and no global phase.



Exercise 12.3.3 Implement the following circuit, combining all three rotations, which therefore applies a gate $\exp(iAt)$

Exercise 12.3.4 Implementing a controlled version of the circuit for $\exp(iA\theta)$.

We now would like to implement a controlled version of this gate. One way to do this is to control every single gate from the control register (lowest qubit of the "upper register"), however this is not required. A simpler way is to only control the two X, and one Z rotations. One extra z-rotation is required, corresponding to the identity term in the matrix, A. This term leads to a z-rotation on the *control* qubit.



Show (using the QUI or otherwise) that this circuit is equivalent to controlling every rotation.

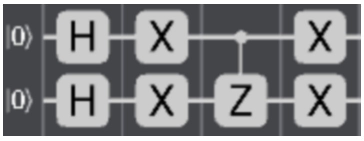
Finally, implement a the z-rotation on the control qubit of with an angle of 0.46875π (global phase of 0.234375π). This rotation comes from the identity term in A.

Exercise 12.3.5 Verifying your circuit using an eigenstate.

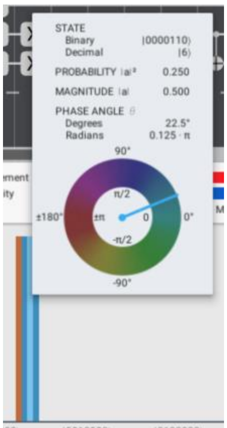
To verify that you have coded your circuit correctly, we will apply the circuit to an eigenstate of A. One such eigenstate is:

$$|u_1\rangle = \frac{1}{2}\begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

To verify, prepare this eigenstate (instead of b) using the circuit :



And verify that if the control qubit is initialized in this state, after the application of the circuit in 12.3.4 (and the control qubit set to 1) that the eigenstate receives a phase of $\frac{\pi}{8}$.

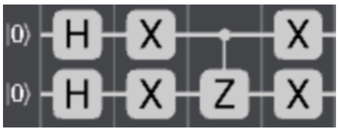


Exercise 12.3.6

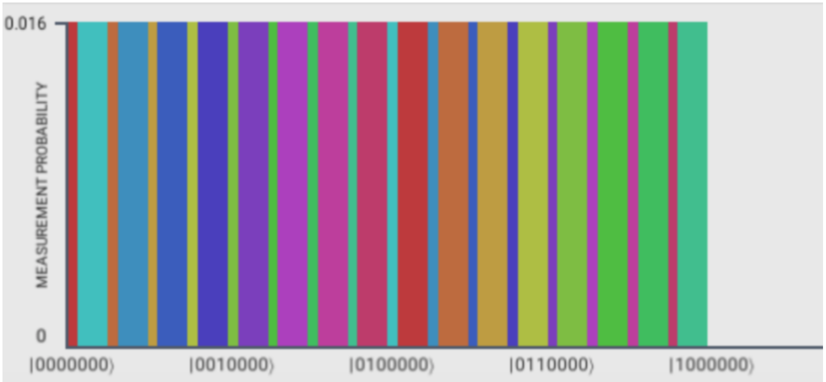
Extend your circuit, repeating the sequence for each of the four control qubits in the “upper register”, remembering to double the angle of each the rotation gates with each round/control qubit. Each round should be controlled from a different qubit in the control register. In terms of the original angle, the angles should be 1, 2, 4, and 8 times as large as the angles given in 12.3.4.

Exercise 12.3.7 Verify your circuit by applying to an eigenstate.

To verify your circuit, prepare the lower register to eigenstate of A in a similar way to 12.3.5, and Hadamard gates to the control/upper register.



Check that your circuit gives phases as shown below, with a phase proportional to the state, t, in the upper/control register:

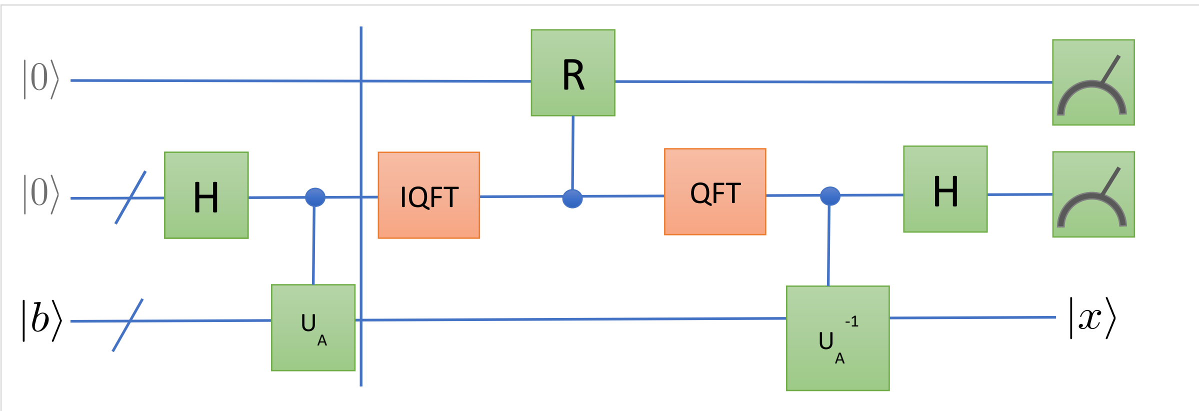


You should be able to verify that the phase is reported by QUI is

$$\frac{\pi}{8}t$$

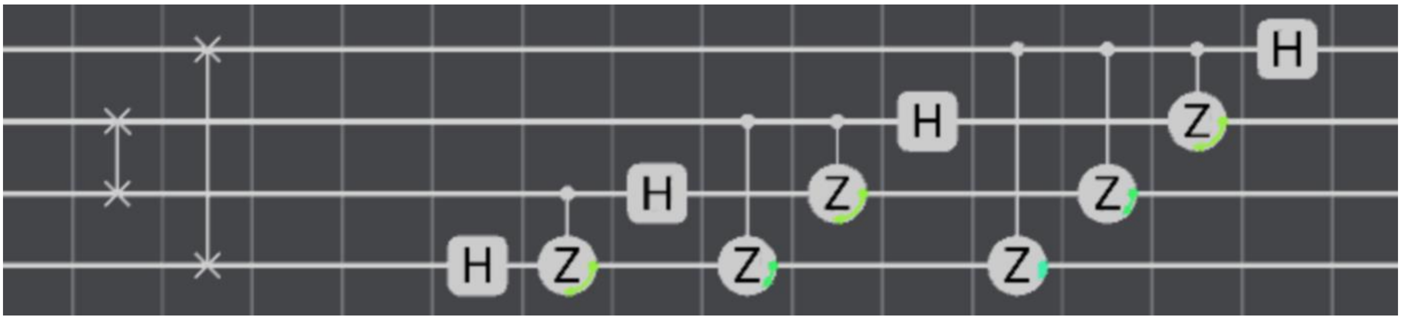
where t is the state of the control/upper register.

After this step has been implemented correctly, you have finished the exponentiation step of the algorithm:



Exercise 12.3.8 Finding the eigenvalues.

To find the eigenvalues, we need to implement a IQFT, pictured below. This IQFT is the same as those you have implemented previously.



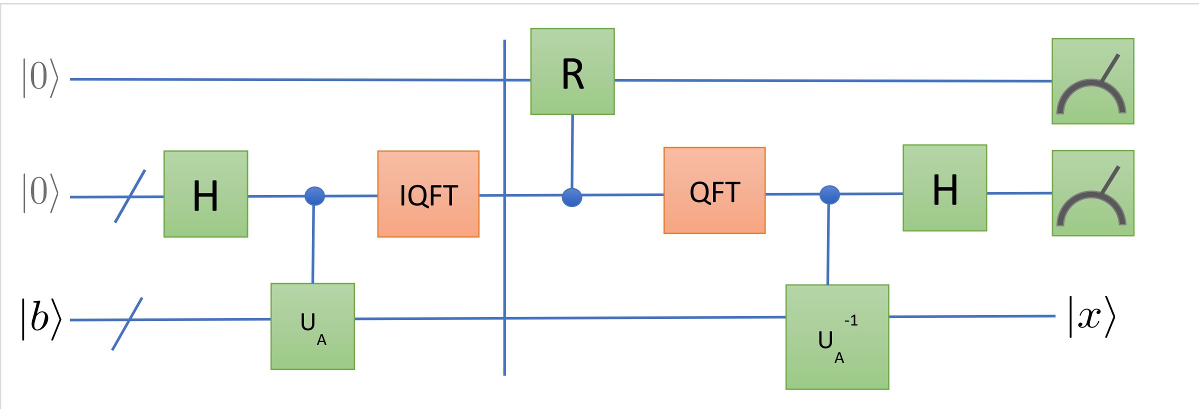
Go ahead and implement an IQFT on the control register.

Exercise 12.3.9 Check the eigenvalues.

After the application of the IQFT, the control/upper register should contain the eigenvalue corresponding to the input state (or a superposition of these values). For the eigenstate we have used in previous exercises, check that the control register is in the state, $|1\rangle$.

Prepare all four eigenstates as inputs to the circuit and check that after the IQFT that the control register contains the corresponding eigenvalue: 1, 2, 4 or 8.

After this step, you have implemented the circuit up to the end of the IQFT:



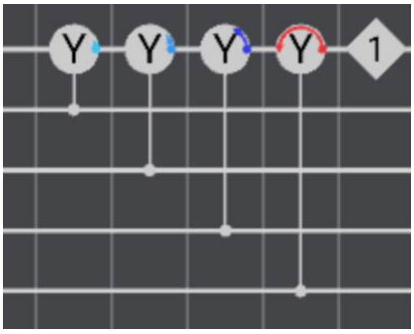
1.2.10 Invert the Eigenvalues

The next part of the circuit inverts the eigenvalues. In this operation, we will slightly cheat, and assume that we know they will be one of these four values: 1, 2, 4 or 8. We wish to implement operations, so that after applied to an initial $|0\rangle$ state, the amplitudes of the $|1\rangle$ state is, 1, $\frac{1}{2}$, $\frac{1}{4}$, and $\frac{1}{8}$ respectively.

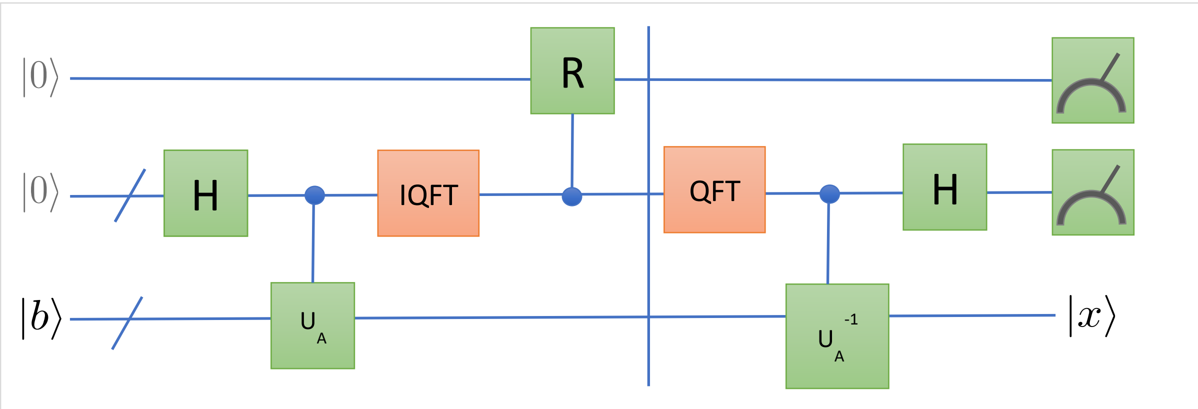
Applying to the initial $|0\rangle$ state, write out your angles which angle y-rotation gives each of these amplitudes:

Amplitude	**Angle of y-rotation**
$\frac{1}{1}$	π
$\frac{1}{2}$	
$\frac{1}{4}$	
$\frac{1}{8}$	

Implement these y-rotations in the QUI, controlled from the appropriate register in the control/upper register



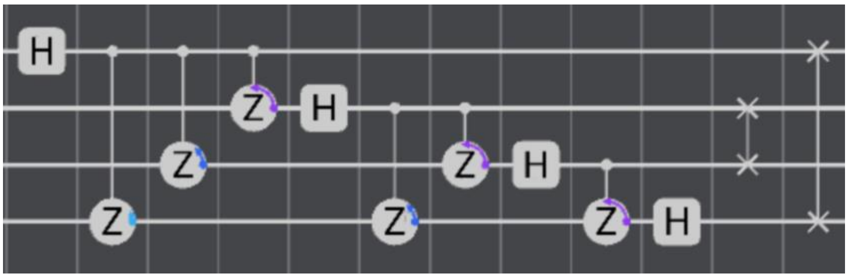
Finally, place a measurement after these rotations. On each run of the algorithm, we will only accept the result if this measurement is a "1". The collapses the state so the resulting amplitude is multiplied by $\frac{1}{\lambda}$.



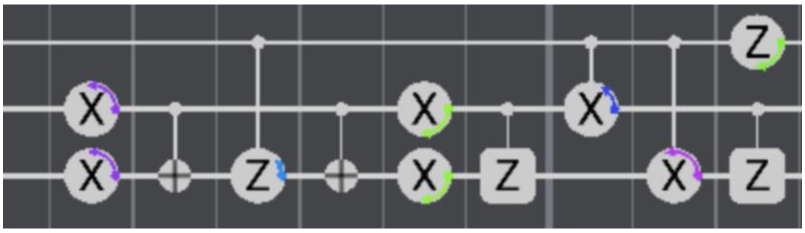
12.2.11 Uncalculate the Eigenvalues

We now need to "uncalculate" the IQFT and exponentiation. Do this by applying the inverse of each gate: reverse the sequence of gates, and change the sign of each rotation.

For the IQFT, this should be similar to the following circuit:



And a similar sequence for of the four exponentiation steps:



Finally, invert the initial Hadamard gates, as well as resetting the ancilla qubit from “1” to “0” using an x-gate:

After the application of these gates, provided a “1” is measured in the ancilla qubit, all of the control register should be in the “0” state.

12.2.12 Read the answer

Repeatedly run the algorithm until the measurement of the top register (the ancilla) is “1”. When it is, read the amplitudes of the resulting four states of the lower register. Are they proportional to, x, the solution to the linear equations?

If so, congratulations, you have demonstrated a version of HHL, one of the more difficult quantum algorithms, running in the QUI!

If not, debug your circuit! One way to do this is to verify that each of the four eigenstates are unchanged by the algorithm.

Your output should look similar to the amplitudes shown above.