A_III ② Quantum Fourier transform and Phase Estimation

Quantum computers can efficiently perform some tasks which (we believe) are not feasible on a classical computer.

Most remarkable example:

finding the prime factorization of an n-bit integer → QC can factor a number exponentially faster than the best known classical algorithms.

The key ingredient to quantum factoring and many other interesting pbs is that the Fourier Transform can be evaluated by an efficient quantum circuit (discovered by Peter Shor)

→ "Quantum Fourier Transform" (QFT)

↳ exploits quantum parallelism to achieve an exponential speedup of the classical Fourier Transform

* Classical (Discrete) Fourier Transform (DFT)

The usual discrete FT takes a vector of complex numbers $v = \begin{pmatrix} v_0 \\ \vdots \\ v_{N-1} \end{pmatrix}$ and transforms it into another vector $\tilde{v} = \begin{pmatrix} \tilde{v}_0 \\ \vdots \\ \tilde{v}_{N-1} \end{pmatrix}$

where $\tilde{v}_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} v_j \, e^{(2i\pi) kj/N}$

* Quantum Fourier Transform

QFT is the analogous of classical DFT:

Action on the computational-basis states of an n-qubit system:

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

where $N = 2^n$

Or more generally, the action on an arbitrary n-qubit state

$$|\Psi\rangle = \sum_{x=0}^{N-1} a_x |x\rangle \quad \text{is}$$

$$|\Psi\rangle = \sum_{x=0}^{N-1} a_x |x\rangle \xrightarrow{QFT} |\tilde{\Psi}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \left(\underbrace{\sum_{x=0}^{N-1} a_x e^{\frac{(2i\pi)xy}{N}}}_{\tilde{a}_y} \right) |y\rangle$$

→ The amplitudes $\tilde{a}$ (= $\begin{pmatrix} \tilde{a}_0 \\ \vdots \\ \tilde{a}_{N-1} \end{pmatrix}$) are the DFT of a (= $\begin{pmatrix} a_0 \\ \vdots \\ a_{N-1} \end{pmatrix}$).

The QFT is in fact a unitary transformation (and thus can be implemented on a QC)

$$|\tilde{\Psi}\rangle = U_{QFT} |\Psi\rangle$$

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

⇒ represented by an $N \times N$ matrix with matrix elements $\langle x | U_{QFT} | y \rangle = \frac{1}{\sqrt{N}} e^{(2i\pi)xy/N}$

The QFT is a change of basis btw the computational basis $|x\rangle$ and the Fourier basis $|\tilde{x}\rangle$.

Example : QFT of a single qubit

$(n = 1 \Rightarrow N = 2)$

$$|0\rangle \mapsto |\tilde{0}\rangle = \frac{1}{\sqrt{2}}\left[e^{(2i\pi)0\times 0/2}|0\rangle + e^{(2i\pi)0\times 1/2}|1\rangle\right]$$

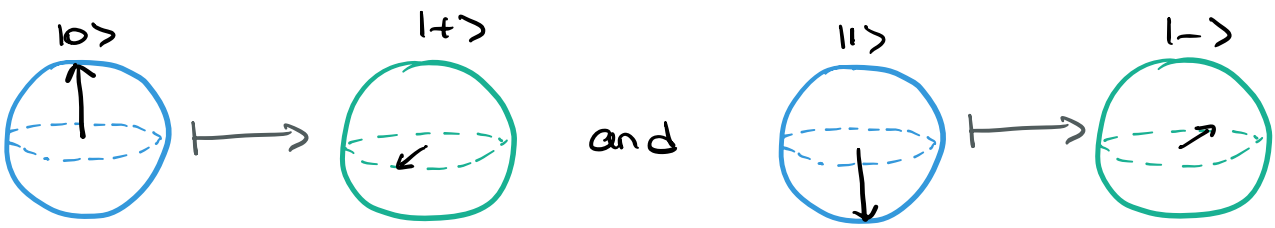
$$= \frac{1}{\sqrt{2}}\{|0\rangle + |1\rangle\} = |+\rangle .$$

$$|1\rangle \mapsto |\tilde{1}\rangle = \frac{1}{\sqrt{2}}\left[e^{(2i\pi)1\times 0/2}|0\rangle + e^{(2i\pi)1\times 1/2}|1\rangle\right]$$

$$= \frac{1}{\sqrt{2}}\{|0\rangle - |1\rangle\} = |-\rangle$$

$\Rightarrow$ in $N = 2$ dimensions, the QFT corresp. to the Hadamard operation.

"Fourier basis" $= \{|+\rangle, |-\rangle\}$.



The naive way of computing the QFT is to do a matrix-vector multiplication

$$\begin{pmatrix} & U_{QFT} & \end{pmatrix} \begin{pmatrix} |\psi\rangle \end{pmatrix} = \begin{pmatrix} |\tilde{\psi}\rangle \end{pmatrix}$$

In that case each entry of $|\tilde{\psi}\rangle$ takes $O(N)$ steps and there are N entries

$$\Rightarrow O(N^2) = O\left(2^{2n}\right) \text{ steps}$$

are needed to compute $|\tilde{\psi}\rangle$.

But there is a well known (classical) procedure that can reduce the number of operations:

Note that only certain terms of the product xy will contribute to

$$e^{(2\pi i)\, xy/N}$$

(others will give a phase $+1$)

Express $\begin{cases} x = x_{n-1}\, 2^{n-1} + x_{n-2}\, 2^{n-2} + \ldots + x_1 2 + x_0 \\ y = y_{n-1}\, 2^{n-1} + y_{n-2}\, 2^{n-2} + \ldots + y_1 2 + y_0 \end{cases}$

Then we have:

$$\frac{xy}{N} = \frac{xy}{2^n}$$

$$= y_{n-1}\left[\frac{2^{n-1}}{2^n}\right]\left(x_{n-1}2^{n-1} + x_{n-2}2^{n-2} \dots + x_1 2 + x_0\right) \quad \left[\frac{2^{n-1}}{2^n} = 2^{-1}\right]$$

$$+ y_{n-2}\left[\frac{2^{n-2}}{2^n}\right]\left(x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots + x_1 2 + x_0\right) \quad \left[\frac{2^{n-2}}{2^n} = 2^{-2}\right]$$

$$\vdots$$

$$+ y_1\left[\frac{2}{2^n}\right]\left(x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots + x_1 2 + x_0\right) \quad \left[\frac{2}{2^n} = 2^{-(n-1)}\right]$$

$$+ y_0\left[\frac{1}{2^n}\right]\left(x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots + x_1 2 + x_0\right) \quad \left[\frac{1}{2^n} = 2^{-n}\right]$$

$$= y_{n-1}\left(\underbrace{x_{n-1}2^{n-2}}_{\text{integer}} + \underbrace{x_{n-2}2^{n-3}}_{\text{integer}} + \dots + \underbrace{x_1}_{\text{integer}} + \frac{x_0}{2}\right) \quad \text{↳ not integ.}$$

$$+ y_{n-2}\left(\underbrace{x_{n-1}2^{n-3}}_{\text{int.}} + \underbrace{x_{n-2}2^{n-4}}_{\text{int.}} + \dots + \frac{x_1}{2} + \frac{x_0}{2^2}\right) \quad \frac{x_1}{2}: \text{not int.}, \ \frac{x_0}{2^2}: \text{not}$$

$$\vdots$$

$$+ y_1\left(\underbrace{x_{n-1}}_{\text{int}} + \frac{x_{n-2}}{2} + \dots + \frac{x_1}{2^{n-2}} + \frac{x_0}{2^{n-1}}\right)$$

$$+ y_0\left(\frac{x_{n-1}}{2} + \frac{x_{n-2}}{2^2} + \dots + \frac{x_1}{2^{n-1}} + \frac{x_0}{2^n}\right)$$

terms in $y_i x_j$ x integer will only contribute a phase $e^{2i\pi \times \text{integer}} = +1$ and thus can be discarded.

What survives is:

$$y_{n-1}\left(\frac{x_0}{2}\right) + y_{n-2}\left(\frac{x_1}{2} + \frac{x_0}{2^2}\right) + \cdots$$

($\cdot x_0$) ($\cdot x_1 x_0$)

$$\cdots + y_1\left(\frac{x_{n-2}}{2} + \cdots + \frac{x_1}{2^{n-2}} + \frac{x_0}{2^{n-1}}\right)$$

($\cdot x_{n-2} \cdots x_1 x_0$)

$$+ y_0\left(\frac{x_{n-1}}{2} + \frac{x_{n-2}}{2^2} + \cdots + \frac{x_1}{2^{n-1}} + \frac{x_0}{2^n}\right)$$

($\cdot x_{n-1} x_{n-2} \cdots x_1 x_0$)

$$= y_{n-1}(\cdot x_0) + y_{n-2}(\cdot x_1 x_0) \cdots$$

$$+ y_1(\cdot x_{n-2} \cdots x_1 x_0) + y_0(\cdot x_{n-1} \cdots x_1 x_0)$$

where the factors in parentheses are binary fractions.

Note: this trick is used classically in the Fast Fourier Transform to reduce the number of operations from $\Theta(N^2)$ to $\Theta(N \log N) = \Theta(n 2^n)$

With quantum parallelism, we can do much better and implement the QFT with $\Theta(n^2)$ elementary operations as we will now show.

We have

$$|\tilde{x}\rangle = U_{QFT}\,|x\rangle \qquad // \; |y_{n-1} \cdots y_1 y_0\rangle$$

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

$$= \frac{1}{\sqrt{N}} \left(\sum_{y_{n-1}=0,1} e^{(2i\pi) y_{n-1}(.x_0)} |y_{n-1}\rangle \right)$$

$$\times \left(\sum_{y_{n-2}} e^{(2i\pi) y_{n-2}(.x_1 x_0)} |y_{n-2}\rangle \right)$$

$$\times \cdots \times \left(\sum_{y_1} e^{(2i\pi) y_1 (.x_{n-2} \cdots x_1 x_0)} |y_1\rangle \right)$$

$$\times \left(\sum_{y_0} e^{(2i\pi) y_0 (.x_{n-1} \cdots x_1 x_0)} |y_0\rangle \right)$$

$$= \frac{1}{\sqrt{N}} \left[|0\rangle + e^{2i\pi (.x_0)} |1\rangle \right] \left[|0\rangle + e^{2i\pi (.x_1 x_0)} |1\rangle \right]$$

$$\times \cdots \times \left[|0\rangle + e^{2i\pi (.x_{n-2} \cdots x_1 x_0)} |1\rangle \right] \left[|0\rangle + e^{2i\pi (.x_{n-1} \cdots x_0)} |1\rangle \right]$$

$\Rightarrow$ we see that the Quantum Fourier transformed state $|\tilde{x}\rangle$ is actually a product state, i.e. it is not entangled (although did not look like a product state from the initial expression).

$\Rightarrow$ the QFT does not generate entanglement. (the QFT of a prod. state is a prod state).

This product state representation suggests that the QFT can be implemented by an efficient quantum circuit.

Let us build the circuit for $n = 3$ and generalize it after.

From the above derivation we have:

$$|\tilde{x}\rangle = |\tilde{x}_2\, \tilde{x}_1\, x_0\rangle$$

$$= \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi(.x_0)}|1\rangle\right]$$

$$\times \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi(.x_1x_0)}|1\rangle\right]$$

$$\times \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi(.x_2x_1x_0)}|1\rangle\right]$$

* $|\tilde{x}_2\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi(.x_0)}|1\rangle\right]$

$$= \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi x_0/2}|1\rangle\right]$$

$$= \begin{cases} \frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right] = |+\rangle & \text{if } x_0 = 0 \\ \frac{1}{\sqrt{2}}\left[|0\rangle - |1\rangle\right] = |-\rangle & \text{if } x_0 = 1 \end{cases}$$

$\Rightarrow |\tilde{x}_2\rangle$ can be obtained by applying a Hadamard gate.

$$* \quad |\tilde{x_1}\rangle = \frac{1}{\sqrt{2}} \left[|0\rangle + e^{2i\pi \left(\frac{x_1}{2} + \frac{x_0}{4} \right)} |1\rangle \right]$$

this state can be obtained by applying H to $|x_1\rangle$ and a controlled-rotation (with $|x_0\rangle$ control qubit):

$$|\tilde{x_1}\rangle = C_{x_0}(R_2) \; H \; |x_1\rangle$$

where we define the rotation

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{pmatrix}$$

$$\Rightarrow R_2 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$$

and $C_{x_0}(R_2)$ (x_0: control) | acts as $\mathbb{1}$ on $|0\rangle$

acts as $e^{\frac{i\pi}{2} x_0} = e^{2i\pi \frac{x_0}{4}}$ on $|1\rangle$

$$\Rightarrow |\tilde{x}_1\rangle = C_{x_0}(R_2)\, H|x_1\rangle$$

$$= C_{x_0}(R_2)\frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi\frac{x_1}{2}}|1\rangle\right]$$

$$= \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi\frac{x_1}{2}} e^{2i\pi\frac{x_0}{4}}|1\rangle\right]$$

$$= \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi(.x_1 x_0)}|1\rangle\right]$$

$\Rightarrow$ OK.

Note: the $C_{x_0}(R_2)$ should be applied before transforming $|x_0\rangle$ with H.

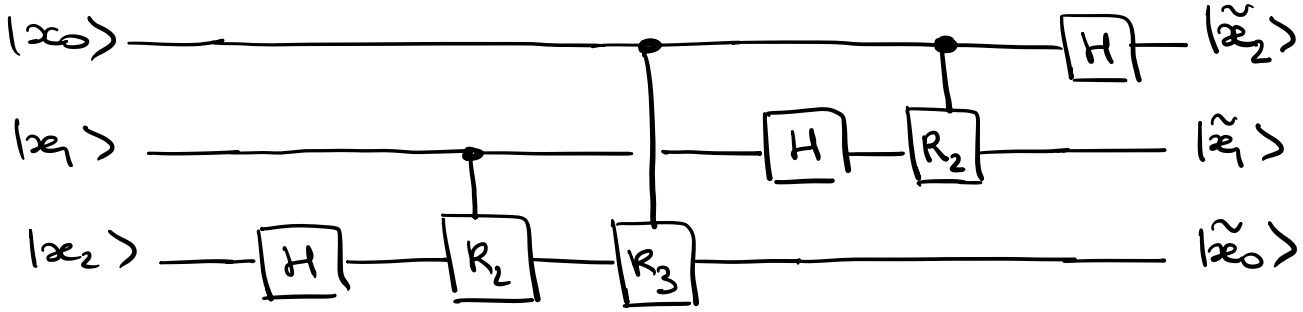
$$* \quad |\tilde{x}_0\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle + e^{2i\pi(.x_2 x_1 x_0)}|1\rangle\right]$$

$$= C_{x_0}(R_3)\, C_{x_1}(R_2)\, H\, |x_2\rangle$$

Note: Similarly $C_{x_0}(R_3)$ should be applied before acting on $|x_0\rangle$ and $C_{x_1}(R_2)$ before acting on $|x_1\rangle$.

$\Rightarrow$ First act on $|x_2\rangle$, then on $|x_1\rangle$, then on $|x_0\rangle$

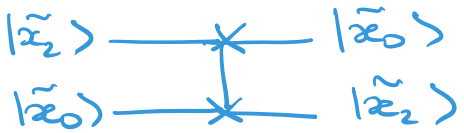
The corresponding circuit is :



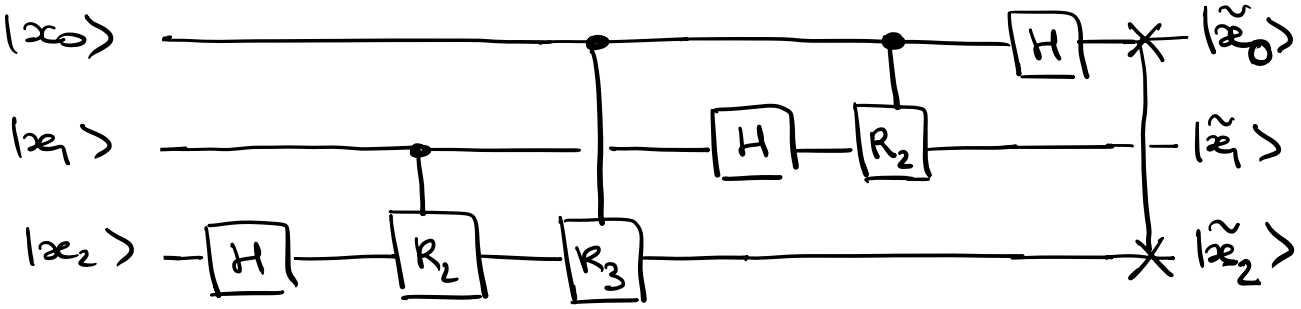
We note that the order of the output qubits are reversed.

=> we apply SWAP gates

Note : a SWAP gate exchanges two qubits :

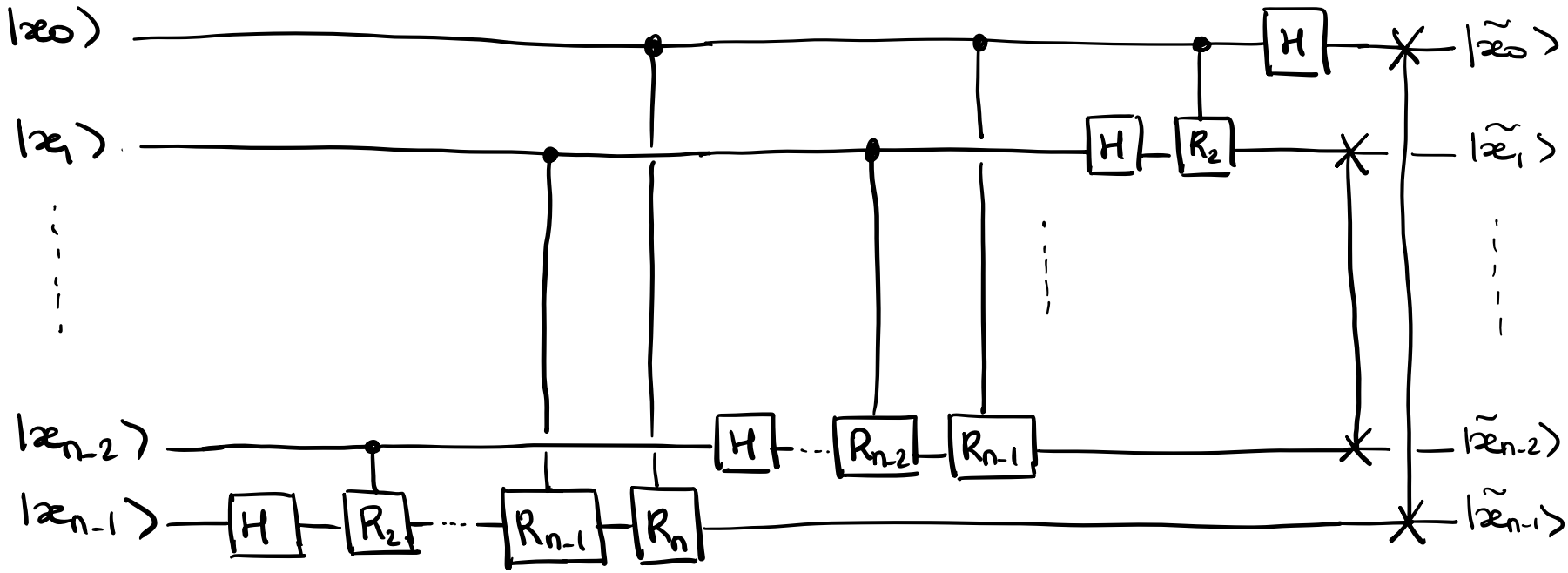


The final circuit is :



We used 3 H, 3 $C(R_k)$ and 1 SWAP gate.

=> Generalization to n qubits :



the controlled R_k is determined by

$k = d+1$ with $d =$ "distance" between the control qubit $|x_i\rangle$ and target $|x_j\rangle$ $\rightarrow$ $d = |i-j|$

How efficient is this circuit ?

→ we do H, and $(n-1)$ conditional rotations on the bottom qubit.

→ we do H and $(n-2)$ conditional rotations on the second to last qubit

⋮

→ H and 1 cond. rotation on the second qubit

→ H on the top qubit

$+ \frac{n}{2}$ SWAP gates

⇒ in total we have :

- n Hadamard gates
- $\underbrace{(n-1) + (n-2) + \dots + 1}$ controlled rotations

$$= \frac{n(n-1)}{2}$$

- $\frac{n}{2}$ SWAP (each can be implemented with three CNOT gates)

$$\Rightarrow \quad n + \frac{n(n-1)}{2} + \frac{3n}{2} = \Theta(n^2)$$
$$= \Theta((\log N)^2)$$

$\Rightarrow$ The QFT can be implemented with a polynomial-size circuit.

$\Rightarrow$ exponential speedup compared to the best classical algo (FFT) which require $\Theta(2^n \cdot n) = \Theta(N \log N)$ gates.

Note: in fact we can even reduce the quantum circuit complexity to $\sim n$ if we are willing to accept a small error - This is because the controlled-rotations $C(R_k)$ acting btw 2 "distant" qubits (for large k) will contribute only exponentially small phases.

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{pmatrix} \qquad k = 1 + d$$
↳ distance

if $d \gg 1 \quad = \quad R_k \sim \mathbb{1} + \begin{pmatrix} 1 & 0 \\ 0 & \underbrace{\varepsilon}_{\text{exponentially small}} \end{pmatrix}$