1 Quantum Parallelism

Quantum circuit can compute a Boolean function for all possible arguments in one-shot.

Example 1 *Let $f(a,b) = ab$. We can use the circuit shown on Fig. 1 to compute this function for all four possible choices of the arguments.*

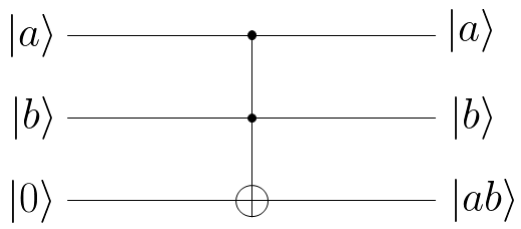


Figure 1: Simulation of AND gate

To achieve this goal we prepare first two qubits in the ***input joint state:***

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)|0\rangle$$
$$= \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle).$$

It is easy to see that at the output we obtain the following ***output joint state:***

$$\frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle).$$

One can see that the each term in the above expression has the form $|a\ b\ ab\rangle$. The problem is, however, that we do not have an access to the individual terms. We can make only measurements of these three qubits (we can also make measurement of any single qubit or any two qubits, but it does not look useful here). As a result of such measurement, we get

Classical Output	*Probability*
0 0 0	1/4
0 1 0	1/4
1 0 0	1/4
1 1 1	1/4

In general if we have a boolean function with n arguments $f(x_1, ..., x_n)$, we can construct a quantum circuit shown on Fig. 2.

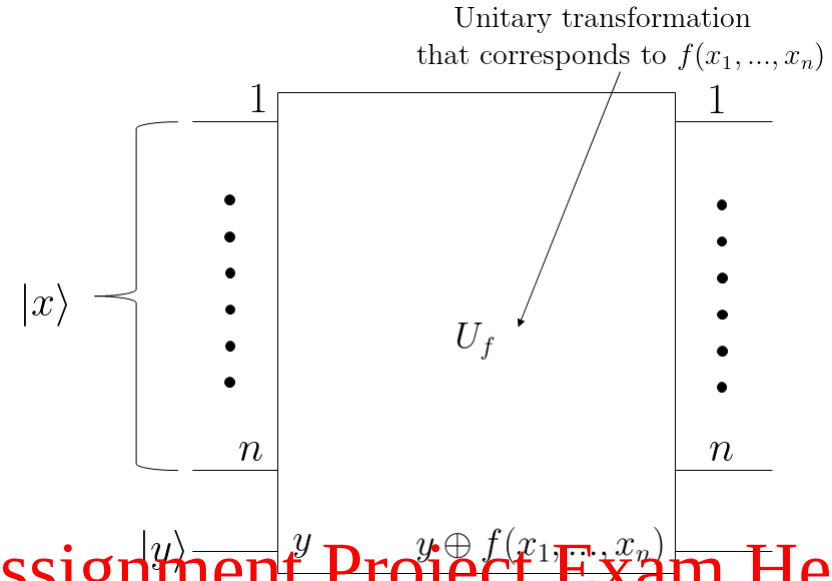


Figure 2: Quantum parallelism

If we prepare the first n qubits in input joint state:

$$\frac{1}{2^{n/2}} \sum_{x_1=0}^{1} ... \sum_{x_n=0}^{1} |x_1...x_n\rangle|y\rangle,$$

then at the output we will get **output joint state:**

$$\frac{1}{2^{n/2}} \sum_{x_1=0}^{1} ... \sum_{x_n=0}^{1} |x_1...x_n \; y \oplus f(x_1...x_n)\rangle.$$

So, if we use $y = 0$, then we obtain $|x_1...x_n \; f(x_1...x_n)\rangle$ for all possible arguments in one shot.

However, we still have the problem that we do not have an access to $f(x_1, ..., x_n)$ for needed $x_1, ..., x_n$. When we make a measurement, we will get at the classical outputs: $x_1, ..., x_n$, $f(x_1, ..., x_n)$ for random $x_1, ..., x_n$ with probability $1/2^n$.

2 Shor's Fast Factoring Algorithm

Assume we have an integer $n = p_1 p_2$ that is a product of two large primes p_1 and p_2. Our goal is to find these primes.

Shor's algorithm consists of three parts: Discrete Fourier Transform, Phase Estimation, and Order Finding. Order Finding leads to fast factoring. Below we consider all these parts.

2.1 Discrete Fourier Transform

Recall that complex numbers of the form $\omega = \mathbf{e}^{2\pi i/N}$ are called *N-th roots of unity*. According to Euler's formula we have

$$\mathbf{e}^{2\pi i/N} = \cos(2\pi/N) + i\sin(2\pi/N),\ i = \sqrt{-1}. \tag{1}$$

From this we have that $\mathbf{e}^{2\pi i} = 1$ and further

$$\mathbf{e}^{2\pi i\cdot m} = (\mathbf{e}^{2\pi i})^m = 1^m = 1,\ \text{for any interger } m. \tag{2}$$

DFT matrix F is the N× N unitary matrix

$$F = \frac{1}{\sqrt{N}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \vdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \vdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{N-1} & \omega^{(N-1)2} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \text{ where } \omega = \mathbf{e}^{2\pi i/N}.$$

So the j-th column of F is $\mathbf{f}_j = \begin{pmatrix} 1 & \omega^j & \omega^{j2} & \cdots & \omega^{j(N-1)} \end{pmatrix}^{\mathrm{T}}$. Note that the j-th row of F is $\mathbf{f}_j^T$. Note also that F is unitary since it is not difficult to check that $FF^\dagger = I_N$.

DFT of vector $\mathbf{x} \in \mathbb{C}^N$ is defined by $\mathbf{y} = F\mathbf{x}$.

Example 2 *Let $N = 2^2$. Then $\omega = \mathbf{e}^{2\pi i/4} = i$ and*

$$F = \frac{1}{2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}.$$

DFT of vector $\mathbf{x} \in \mathbb{C}^4$ is

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = F \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Remark 1 *Recall that in Dirac notation the state* $|j\rangle$ *means the* j*-th vector (in decimal representation) in an orthonormal basis of* $\mathbb{C}^{2^n}$*. Typically we use binary representation, but decimal one works equally well. If, for example,* $n = 2$ *then* $|2\rangle$ *means the same as* $|10\rangle$*, and we can write a generic quantum state as*

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle,$$

or equivalently as

$$\alpha_{00}|0\rangle + \alpha_{01}|1\rangle + \alpha_{10}|2\rangle + \alpha_{11}|3\rangle.$$

In what follows we assume that $N = 2^n$.

Let we have n qubits in a state $|j\rangle,\ j = 0, \ldots, N-1$¿ Let us further assume that we constructed a quantum circuit that conducts the unitary transform

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk}|k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \mathbf{e}^{2\pi ijk/N}|k\rangle,\ j = 0, \ldots, N-1. \quad (3)$$

Then if we submit to the input of this circuit n qubits in the state

$$\sum_{j=0}^{N-1} x_j|j\rangle, \quad (4)$$

then, according to (3), the circuit will transform this state into the output state

$$\sum_{j=0}^{N-1} x_j \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk}|k\rangle = \sum_{k=0}^{N-1} |k\rangle \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega^{jk} = \sum_{k=0}^{N-1} |k\rangle \underbrace{\mathbf{f}_k^T \mathbf{x}}_{y_k} = \sum_{k=0}^{N-1} y_k|k\rangle,$$

where coefficients x_j and y_k are connected by the DFT:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix} = F \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix}.$$

Thus, the circuit that conducts (3) will also conduct DFT for a general quantum state (4). Therefore we proceed with construction of a quantum

circuit that conducts the mapping:

$$|j\rangle \to \frac{1}{2^{n/2}} \sum_{k=0}^{N-1} \mathbf{e}^{2\pi ijk/2^n} |k\rangle = \textcircled{\alpha} \tag{5}$$

We represent integer k by its binary expansion:

$$k = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0, \ \ k_s = 0, 1.$$

(For example $5 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$.) Then

$$\begin{aligned}
\textcircled{\alpha} &= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \mathbf{e}^{2\pi ij(k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0)/2^n} |k_1 \cdots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \mathbf{e}^{2\pi ij(k_1/2 + k_2/4 + \cdots + k_n/2^n)} |k_1 \cdots k_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \mathbf{e}^{2\pi ijk_1/2} |k_1\rangle \otimes \mathbf{e}^{2\pi ijk_2/4} |k_2\rangle \otimes \cdots \otimes \mathbf{e}^{2\pi ijk_n/2^n} |k_n\rangle = \textcircled{\beta}
\end{aligned} \tag{6}$$

We note that the following sums of products are equal to the products of sums:

$$a_0 b_0 + a_0 b_1 + a_1 b_0 + a_1 b_1 = (a_0 + a_1)(b_0 + b_1),$$

and

$$\begin{aligned}
&a_0 b_0 c_0 + a_0 b_0 c_1 + a_0 b_1 c_0 + a_0 b_1 c_1 + a_1 b_0 c_0 + a_1 b_0 c_1 + a_1 b_1 c_0 + a_1 b_1 c_1 \\
=&(a_0 + a_1)(b_0 + b_1)(c_0 + c_1)
\end{aligned}$$

We note that (6) is also a sum of products with a_0, a_1, b_0, b_1, and so on, defined as

$$\begin{aligned}
k_1 = 0, \ a_0 &= \mathbf{e}^{2\pi ij \cdot 0/2} |0\rangle = |0\rangle \\
= 1, \ a_1 &= \mathbf{e}^{2\pi ij \cdot 1/2} |1\rangle \\
k_2 = 0, \ b_0 &= \mathbf{e}^{2\pi ij \cdot 0/4} |0\rangle = |0\rangle \\
= 1, \ b_1 &= \mathbf{e}^{2\pi ij \cdot 1/4} |1\rangle \\
&\vdots
\end{aligned}$$

Hence (6) can be written in the form:

$$\textcircled{\beta} = \frac{1}{2^{n/2}} \bigotimes_{l=1}^{n} (|0\rangle + \mathbf{e}^{2\pi i j \cdot 1/2^l}|1\rangle) = \textcircled{\gamma} \tag{7}$$

Now we represent j via its binary expansion:

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0, \ j_s = 0, 1,$$

and taking into account (2), we simplify the the second terms of the factors of (7) as follows:

$$\begin{aligned} l = 1: \ & \mathbf{e}^{2\pi i (j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_{n-1} \cdot 2 + j_n)/2} \\ & = \underbrace{\mathbf{e}^{2\pi i j_1 2^{(n-1)}/2}}_{1} \cdot \underbrace{\mathbf{e}^{2\pi i j_2 2^{(n-2)}/2}}_{1} \cdot \ldots \cdot \underbrace{\mathbf{e}^{2\pi i j_{n-1}}}_{1} \cdot \mathbf{e}^{2\pi i j_n/2} \\ & = \mathbf{e}^{2\pi i j_n/2} \\ l = 2: \ & \mathbf{e}^{2\pi i (j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_{n-1} \cdot 2 + j_n)/4} \\ & = \underbrace{\mathbf{e}^{2\pi i j_1 2^{n-1}/4}}_{1} \cdot \ldots \cdot \underbrace{\mathbf{e}^{2\pi i j_{n-2}}}_{1} \cdot \mathbf{e}^{2\pi i j_{n-1}/2} \cdot \mathbf{e}^{2\pi i j_n/4} \\ & = \mathbf{e}^{2\pi i j_{n-1}/2} \cdot \mathbf{e}^{2\pi i j_n/4} \\ & \vdots \quad \vdots \end{aligned}$$

Using these expresssions, we get

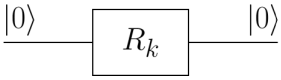
$$\begin{aligned} \textcircled{\gamma} = & \frac{1}{2^{n/2}} (|0\rangle + \mathbf{e}^{2\pi i j_n/2}|1\rangle) \otimes (|0\rangle + \mathbf{e}^{2\pi i (j_{n-1}/2 + j_n/4)}|1\rangle) \\ & \otimes \cdots \otimes (|0\rangle + \mathbf{e}^{2\pi i (j_1/2 + j_2/4 + \cdots + j_n/2^n)}|1\rangle). \end{aligned} \tag{8}$$

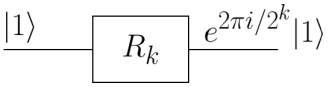
Summarizing the above arguments and derivations, we conclude that our goal is to construct a quantum circuit that conducts the transform

$$\begin{aligned} |j\rangle \ \rightarrow \ & \frac{1}{2^{n/2}} (|0\rangle + \mathbf{e}^{2\pi i j_n/2}|1\rangle) \otimes (|0\rangle + \mathbf{e}^{2\pi i (j_{n-1}/2 + j_n/4)}|1\rangle) \\ & \otimes \cdots \otimes (|0\rangle + \mathbf{e}^{2\pi i (j_1/2 + j_2/4 + \cdots + j_n/2^n)}|1\rangle) \end{aligned} \tag{9}$$

Now we will construct a circuit that would produce (8). We will use the following unitary rotations:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{e}^{2\pi i/2^k} \end{bmatrix}, \text{ which for } k = 2 \text{ becomes } R_2 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{e}^{2\pi i/4} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

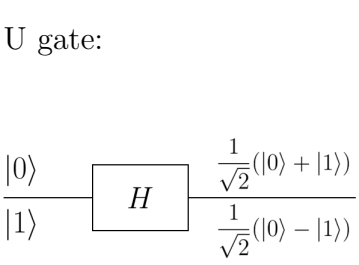




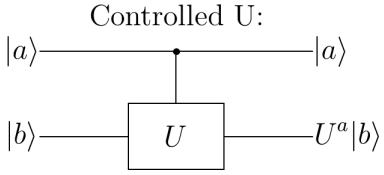
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{e}^{2\pi i/2^k} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \to \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{e}^{2\pi i/2^k} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mathbf{e}^{2\pi i/2^k} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mathbf{e}^{2\pi i/2^k}|1\rangle$$

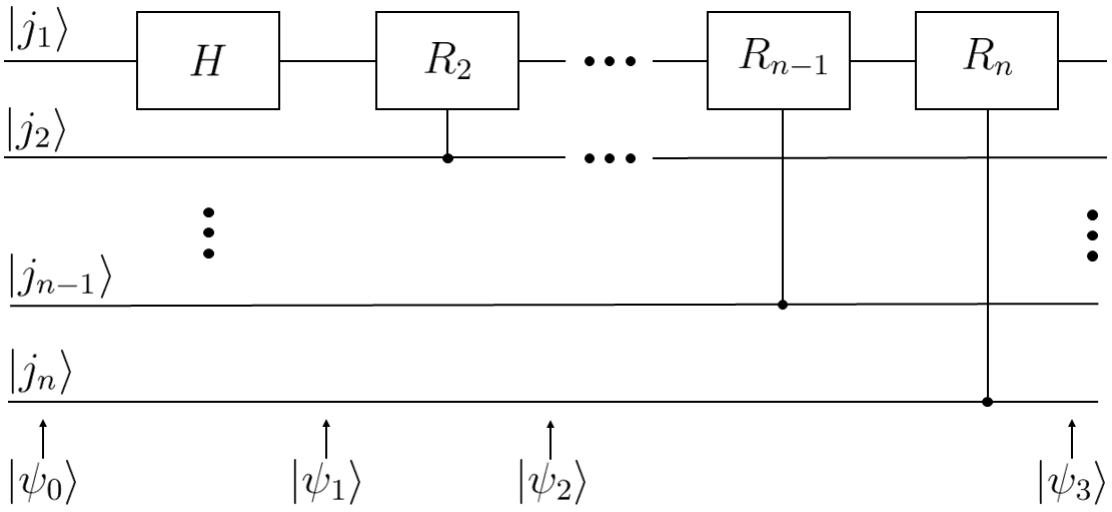
Remind the action of the Hadamard gate $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and Controlled U gate:



Controlled U:



The beginning of quantum circuit for DFT (the part for 1-st qubit):



In the very beginning we have $|\psi_0\rangle = |j_1 \dots j_n\rangle$. Next

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i j_1/2}|1\rangle)|j_2 j_3 \dots j_n\rangle$$

Indeed:

$$\text{If } j_1 = 0 : \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i \cdot 0/2}|1\rangle)|j_2 j_3 \dots j_n\rangle$$
$$= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|j_2 j_3 \dots j_n\rangle$$
$$\text{If } j_1 = 1 : \frac{1}{\sqrt{2}}(|0\rangle + \underbrace{\mathbf{e}^{2\pi i \cdot 1/2}}_{-1}1|1\rangle)|j_2 j_3 \dots j_n\rangle$$
$$= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|j_2 j_3 \dots j_n\rangle$$

Further

$$|\psi_2\rangle : j_2 = 0 : (|0\rangle + \mathbf{e}^{2\pi i j_1/2}|1\rangle)|0\rangle|j_3 \dots j_n\rangle$$
$$j_2 = 1 : (|0\rangle + \mathbf{e}^{2\pi i j_1/2} \cdot \mathbf{e}^{2\pi i \cdot 1/4}|1\rangle)|1\rangle|j_3 \dots j_n\rangle$$

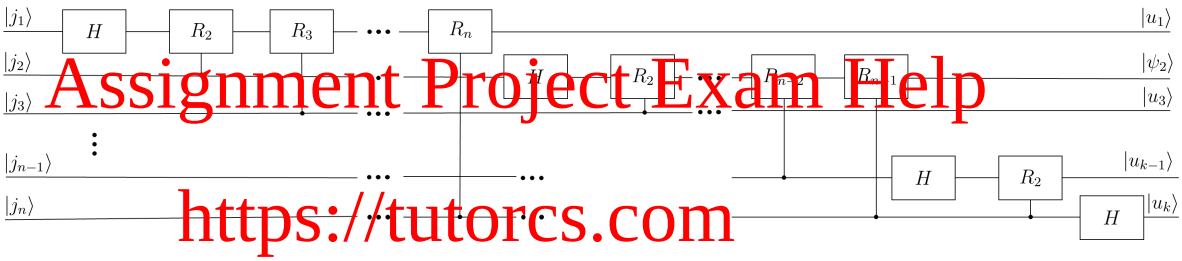
Hence

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i(j_1/2+j_2/4)}|1\rangle)|j_2\rangle|j_2\cdots j_n\rangle$$

and continuing in this way we obtain:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i(j_1/2+j_2/4+\cdots+j_{n-1}/2^{n-1}+j_n/2^n)}|1\rangle)|j_2\cdots j_n\rangle$$

We note that the state of the 1-st qubit in this expression is the same as the last factor in (8). Using the same type of gate connection and similar arguments, we obtain the circuit in which all n qubits will have the states corresponding to the factors of (8) taken in the reverse order:



The states of the qubits are

$$|u_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i(j_1/2+j_2/4+\cdots+j_{n-2}/2^{n-1}+j_n/2^n)}|1\rangle)$$

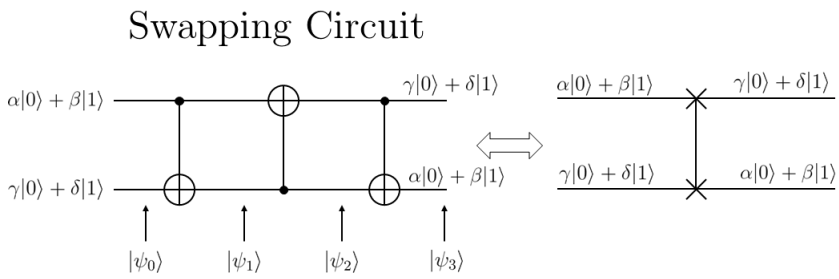
$$|u_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i(j_2/2+\cdots+j_n/2^{n-1})}|1\rangle)$$

$$\vdots$$

$$|u_{n-1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi i(j_{n-1}/2+j_n/2^4)}|1\rangle)$$

$$|u_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \mathbf{e}^{2\pi j_n/2}|1\rangle)$$

The joint state at the output of the circuit is $|u_1\ u_2 \ldots u_n\rangle$. Note that this state has the reverse order of multiplication compared with the needed state (8). To correct this we will use the swapping circuit.



2.1.1 Swapping Circuit

The following circuit swaps 1-st and 2-nd qubits.

$$
\begin{aligned}
|\psi_0\rangle &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle \\
|\psi_1\rangle &= \alpha\gamma|00\rangle + \alpha\delta|11\rangle + \beta\gamma|01\rangle + \beta\delta|10\rangle \\
|\psi_2\rangle &= \alpha\gamma|00\rangle + \alpha\delta|11\rangle + \beta\gamma|01\rangle + \beta\delta|10\rangle \\
|\psi_3\rangle &= \alpha\gamma|00\rangle + \alpha\delta|10\rangle + \beta\gamma|01\rangle + \beta\delta|11\rangle \\
&= (\gamma|0\rangle + \delta|1\rangle)(\alpha|0\rangle + \beta|1\rangle)
\end{aligned}
$$

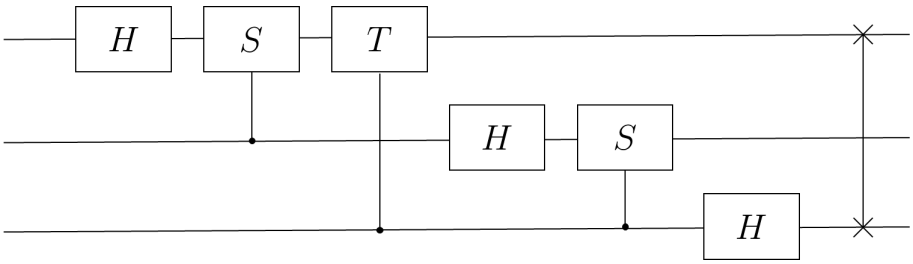
Example 3

$$
F = \frac{1}{\sqrt{8}} \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots
\end{bmatrix}
$$

$$
R_2 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{e}^{2\pi i/2^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{e}^{\pi i/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = S
$$

$$
R_3 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{e}^{2\pi i/8} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{e}^{\pi i/4} \end{bmatrix} = T
$$

The DFT circuit is shown on the next page.



2.1.2 Complexity of Quantum DFT

1. Qubit 1: H gate $+(n-1)$ rotations $\Rightarrow n$ gates
2. Qubit 2: H gate $+(n-2)$ rotations $\Rightarrow n-1$ gates

$\vdots$

n. Qubit n: H gate $\Rightarrow$ 1 gate

Total: $n(n+1)/2$ gates ($n \approx \log_2 N$)

The classical complexity is $\Theta(n2^n)$. **Exponential speed up!!**

Can we use quantum DFT to get all elements of vector $\mathbf{y} = F\mathbf{x}$? **No!**