Quantum Algorithms (4)

Hung-Wei Tseng

Quantum Fourier Transform (QFT)

QFT

- Transforms between two bases, the computational (Z) basis, and the Fourier basis.

$$|\text{State in Computational Basis}\rangle \xrightarrow{\text{QFT}} |\text{State in Fourier Basis}\rangle$$

$$\text{QFT}\,|x\rangle = |\widetilde{x}\rangle$$

Generalized QFT

- Given $|x\rangle = |x_1 \ldots x_n\rangle$

$$QFT_N|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle$$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i xy/2^n} |y\rangle$$ since $\omega_N^{xy} = e^{2\pi i \frac{xy}{N}}$ and $N = 2^n$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \left(\sum_{k=1}^{n} y_k/2^k\right)x} |y_1 \ldots y_n\rangle$$ rewriting in fractional binary notation $y = y_1 \ldots y_n, y/2^n = \sum_{k=1}^{n} y_k/2^k$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{2\pi i x y_k/2^k} |y_1 \ldots y_n\rangle$$ after expanding the exponential of a sum to a product of exponentials

$$= \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} \left(|0\rangle + e^{2\pi i x/2^k} |1\rangle\right)$$ after rearranging the sum and products, and expanding $\sum_{y=0}^{N-1} = \sum_{y_1=0}^{1} \sum_{y_2=0}^{1} \ldots \sum_{y_n=0}^{1}$

$$= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i}{2}x}|1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^2}x}|1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^{n-1}}x}|1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^n}x}|1\rangle\right)$$

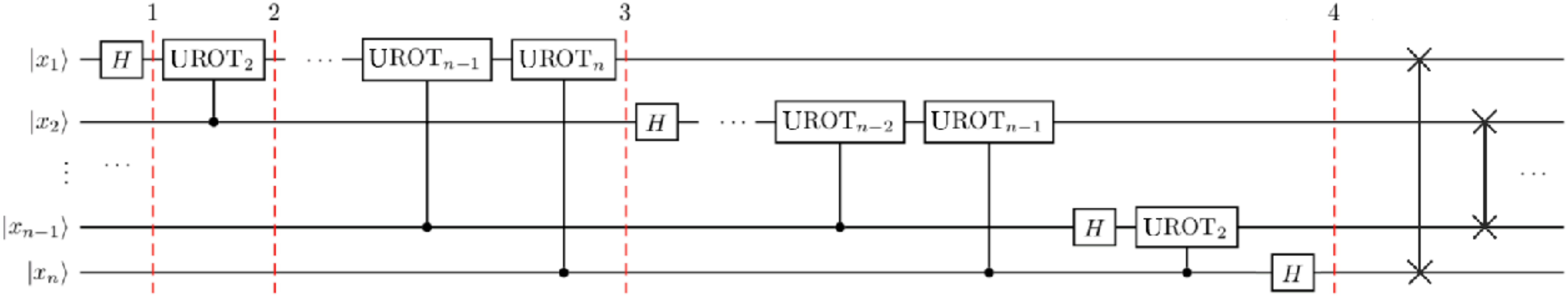
Implementing QFT

- $H|x_k\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + \exp\left(\frac{2\pi i}{2}x_k\right)|1\rangle\right)$

$CROT_k = \begin{bmatrix} I & 0 \\ 0 & UROT_k \end{bmatrix}$, where $UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi i}{2^k}\right) \end{bmatrix}$

$CROT_k|0x_j\rangle = |0x_j\rangle$ and $CROT_k|1x_j\rangle = \exp\left(\frac{2\pi i}{2^k}x_j\right)|1x_j\rangle$

The quantum circuit



Counting in quantum basis

$$x = 2^{n-1}x_1 + 2^{n-2}x_2 + \ldots + 2^1 x_{n-1} + 2^0 x_n$$

$$\Downarrow$$

$$\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(\frac{2\pi i}{2^n}x\right)|1\rangle\right] \otimes |x_2 x_3 \ldots x_n\rangle$$

Quantum Phase Estimation

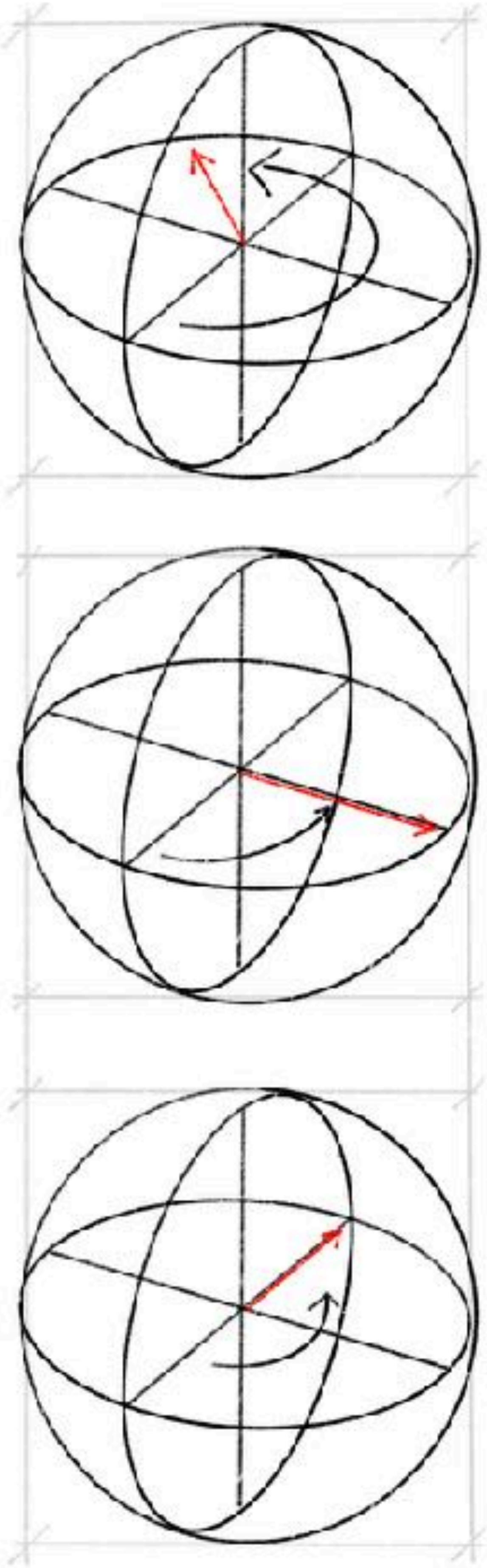
QPE

- Given a unitary operator U, the algorithm estimates θ in $U|\psi\rangle = e^{\boldsymbol{2\pi i\theta}}|\psi\rangle$.
  — $|\psi\rangle$ is an eigenvector and $e^{\boldsymbol{2\pi i\theta}}$ is the corresponding eigenvalue. Since U is unitary, all of its eigenvalues have a norm of 1.

- Or say we want to evaluate the $\dfrac{x}{2^n}$ in the transformed equation

$$\frac{1}{\sqrt{2}}\left[|0\rangle + \exp\left(2\pi i \times \frac{x}{2^n}\right)|1\rangle\right] \otimes |x_2x_3\ldots x_n\rangle$$

5 in the
fourier basis
(on 3 qubits)



$\frac{5}{8}$ Full turn

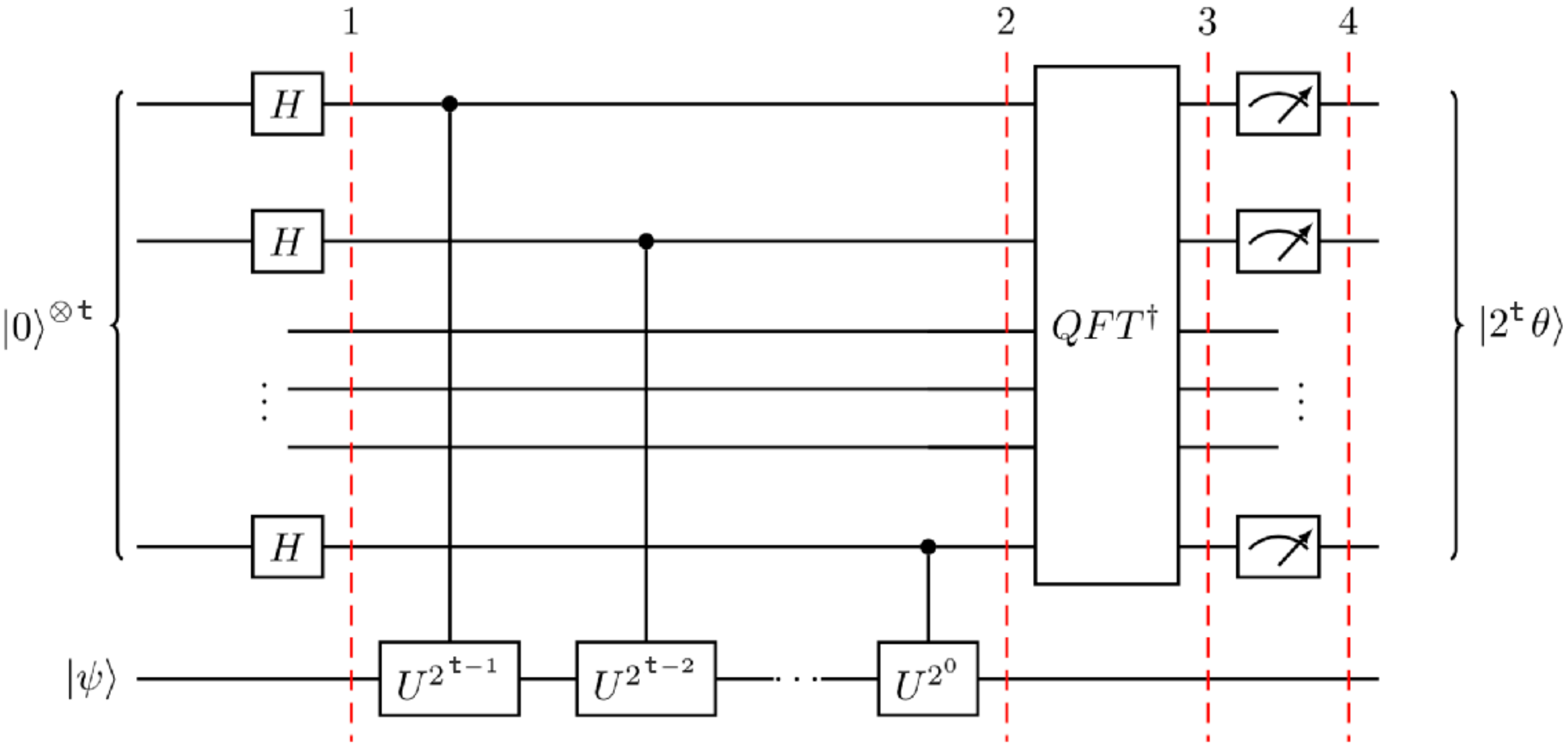
$\frac{10}{8}$ Full turn

$\frac{20}{8}$ Full turn

The idea

- Uses phase kickback to write the phase of U (in the Fourier basis) to the t qubits in the counting register.
  - In the Fourier basis the topmost qubit completes one full rotation when counting between 0 and 2^t.
  - To count to a number, x between 0 and 2^t, we rotate this qubit by x around the z-axis.
  - For the next qubit we rotate by $\frac{2x}{2^t}$, then $\frac{4x}{2^t}$ for the third qubit.
  - When we use a qubit to control the U-gate, the qubit will turn (due to kickback) proportionally to the phase $e^{2i\pi\theta}$.
  - Use successive CU-gates to repeat this rotation an appropriate number of times until we have encoded the phase theta as a number between 0 and 2^t in the Fourier basis.
  - Use QFT to convert this into the computational basis

Implementation of the idea



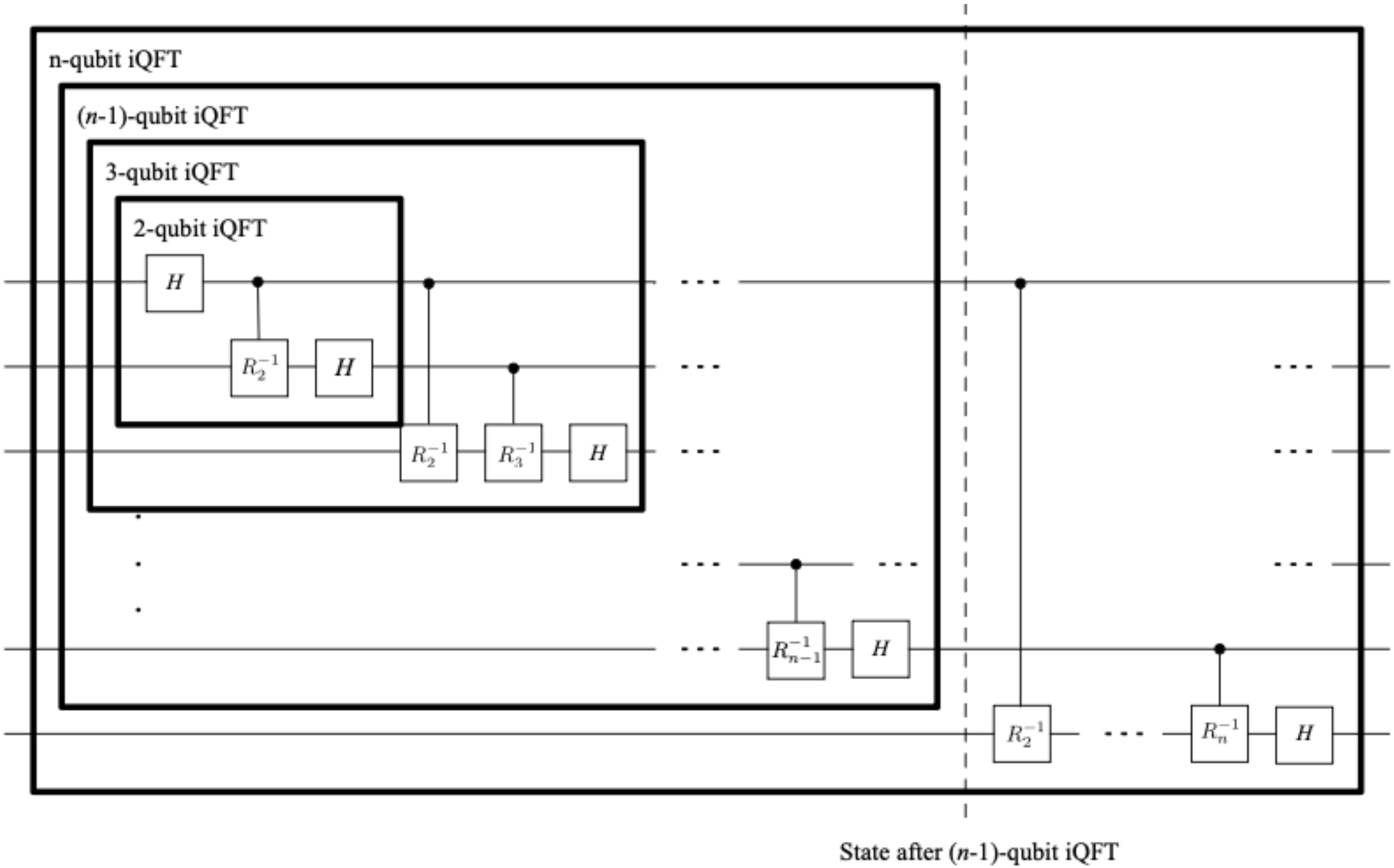


Figure 3.2: Inverse Quantum Fourier Transform (iQFT) circuit.

Implementation

- Setup: $|\psi\rangle$ is in one set of qubit registers. An additional set of n qubits form the counting register on which we will store the value $2^n\theta$: $|\psi_0\rangle = |0\rangle^{\otimes n}|\psi\rangle$
- Superposition: Apply a n-bit Hadamard gate operation $H^{\otimes n}$ on the counting register:

$$|\psi_1\rangle = \frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + |1\rangle\right)^{\otimes n}|\psi\rangle$$

- Controlled Unitary Operations — CU that applies the unitary operator U on the target register only if its corresponding control bit is $|1\rangle$.Since U is a unitary operator with eigenvector $|\psi\rangle$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, this means

$$U^{2^j}|\psi\rangle = U^{2^j-1}U|\psi\rangle = U^{2^j-1}e^{2\pi i\theta}|\psi\rangle = \cdots = e^{2\pi i 2^j\theta}|\psi\rangle$$

- Applying all the n controlled operations CU^{2^j} with $0 \leq j \leq n-1$, and using the relation

$$|0\rangle \otimes |\psi\rangle + |1\rangle \otimes e^{2\pi i\theta}|\psi\rangle = \left(|0\rangle + e^{2\pi i\theta}|1\rangle\right) \otimes |\psi\rangle$$

$$|\psi_2\rangle = \frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{\boldsymbol{2\pi i}\theta 2^{n-1}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{\boldsymbol{2\pi i}\theta 2^{1}}|1\rangle\right) \otimes \left(|0\rangle + e^{\boldsymbol{2\pi i}\theta 2^{0}}|1\rangle\right) \otimes |\psi\rangle$$

$$= \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{\boldsymbol{2\pi i}\theta k}|k\rangle \otimes |\psi\rangle$$

where k denotes the integer representation of n-bit binary numbers.

Implementation (2)

- Inverse Fourier Transform
  Recall that QFT maps an n-qubit input state $|x\rangle$ into an output as

$$QFT|x\rangle = \frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{\frac{2\pi i}{2}x}|1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^2}x}|1\rangle\right) \otimes \ldots \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^{n-1}}x}|1\rangle\right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^n}x}|1\rangle\right)$$

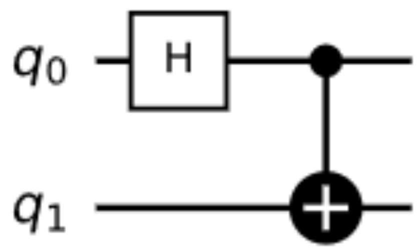
- Replacing x by $2^n\theta$ in the above expression gives exactly the expression of $\psi_2\rangle$. Therefore, to recover the state $|2^n\theta\rangle$, apply an inverse Fourier transform on the auxiliary register. Doing so, we find

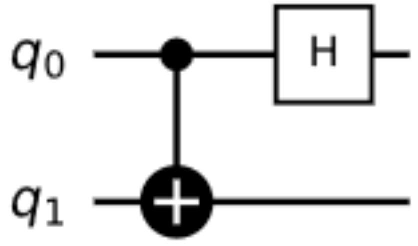
$$|\psi_3\rangle = \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{\boldsymbol{2\pi i}\theta k}|k\rangle \otimes |\psi\rangle \xrightarrow{\mathcal{QFT}_n^{-1}} \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{-\frac{2\pi i k}{2^n}(x-2^n\theta)}|x\rangle \otimes |\psi\rangle$$

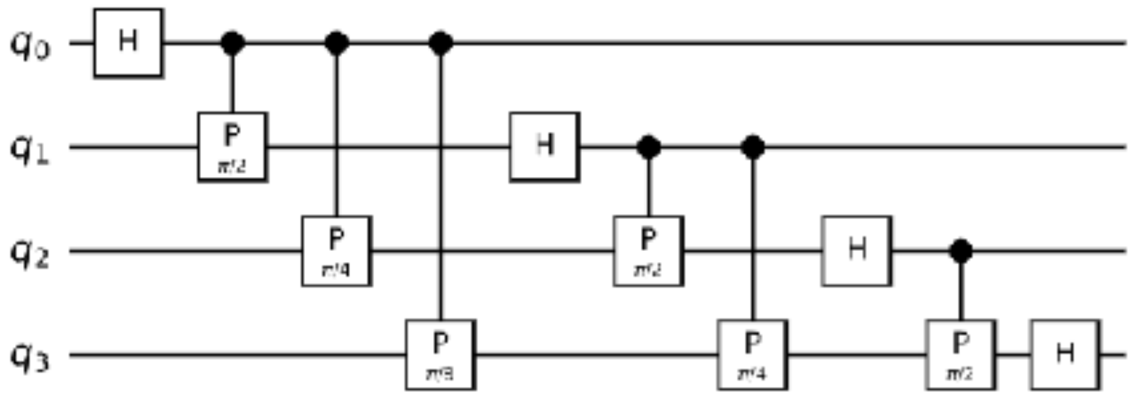
- Measurement: The above expression peaks near $x = 2^{n\theta}$. For the case when $2^{n\theta}$ is an integer, measuring in the computational basis gives the phase in the auxiliary register with high probability:
  $|\psi_4\rangle = |2^n\theta\rangle \otimes |\psi\rangle$
  For the case when $2^n\theta$ is not an integer, it can be shown that the above expression still peaks near $x = 2^n\theta$ with probability better than $4/\pi^2 \approx 40\,\%$.

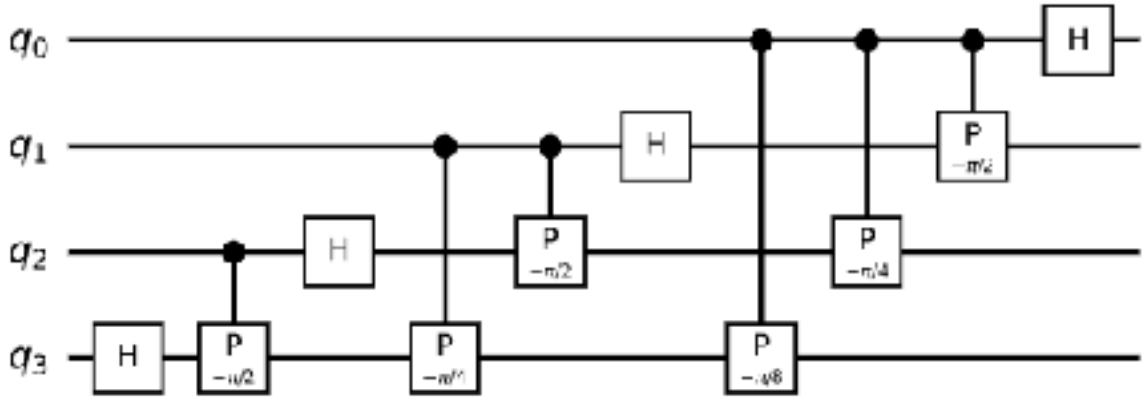
Inverse QFT

- Remember in linear algebra $(AB)^{-1} = B^{-1}A^{-1}$



's inverse is


- 

's inverse is


Example: T-Gate

- T-gate adds a phase of $e^{\frac{i\pi}{4}}$ to the state $|1\rangle$

$$T|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = e^{\frac{i\pi}{4}}|1\rangle$$

- Since QPE will give us θ where

$$T|1\rangle = e^{2i\pi\theta}|1\rangle$$

we expect to find

$$\theta = \frac{1}{8}$$

Shor’s Algorithm

Shor's algorithm

- Reducing the integer factorization problem into the period finding problem
  - $f(x) = a^x \bmod N = 1$ if a and N are positive integers, a is less than N, and they have no common factors.
  - Find the period, or order (r), the smallest (non-zero) integer makes f(x) = 1
  - Shor's solution was to use quantum phase estimation on the unitary operator to find r:
    $U|y\rangle \equiv |ay \bmod N\rangle$
  - The factors of N are $gcd(a^{\frac{r}{2}} \pm 1, N)$ and we are done.

Why $a^{\frac{r}{2}}$

- Since r is the period of $f(x) = a^x \bmod N$, $f(x) = a^r \bmod N = 1$
- If r is even

$$a^r \bmod N = 1$$
$$a^r - 1 \bmod N = 0$$
$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \bmod N = 0$$

- Let $a_0 = a^{\frac{r}{2}} - 1$ and $a_1 = a^{\frac{r}{2}} + 1$, this implies $N \mid (a^{\frac{r}{2}} - 1)a^{\frac{r}{2}} + 1$ or $N \mid a_0 a_1$
- Since r is the smallest integer such that $a^r \bmod N = 1$, N cannot divide a_0 — N has a non-trivial factor in common with either a_0 or a_1

How QPE can help find the period?

- For example, $a = 3$ and $N = 35$:

$$
\begin{aligned}
U|1\rangle &= |3\rangle \\
U^2|1\rangle &= |9\rangle \\
U^3|1\rangle &= |27\rangle \\
&\vdots \\
U^{(r-1)}|1\rangle &= |12\rangle \\
U^r|1\rangle &= |1\rangle
\end{aligned}
$$

- So a superposition of the states in this cycle ($|u_0\rangle$) would be an eigenstate of

$$
\begin{aligned}
U: |u_0\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |a^k \bmod N\rangle \\
|u_0\rangle &= \frac{1}{\sqrt{12}}(|1\rangle + |3\rangle + |9\rangle \ldots + |4\rangle + |12\rangle) \\
U|u_0\rangle &= \frac{1}{\sqrt{12}}(U|1\rangle + U|3\rangle + U|9\rangle \ldots + U|4\rangle + U|12\rangle) \\
&= \frac{1}{\sqrt{12}}(|3\rangle + |9\rangle + |27\rangle \ldots + |12\rangle + |1\rangle) \\
&= |u_0\rangle
\end{aligned}
$$

How QPE can help find the period? (2)

- Let's look at the case in which the phase of the kth state is proportional to k:

$$|u_1\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-\frac{2\pi ik}{r}} |a^k \bmod N\rangle$$

$$U|u_1\rangle = e^{\frac{2\pi i}{r}}|u_1\rangle$$

For a=3, N=35,

$$|u_1\rangle = \frac{1}{\sqrt{12}}(|1\rangle + e^{-\frac{2\pi i}{12}}|3\rangle + e^{-\frac{4\pi i}{12}}|9\rangle \ldots + e^{-\frac{20\pi i}{12}}|4\rangle + e^{-\frac{22\pi i}{12}}|12\rangle)$$

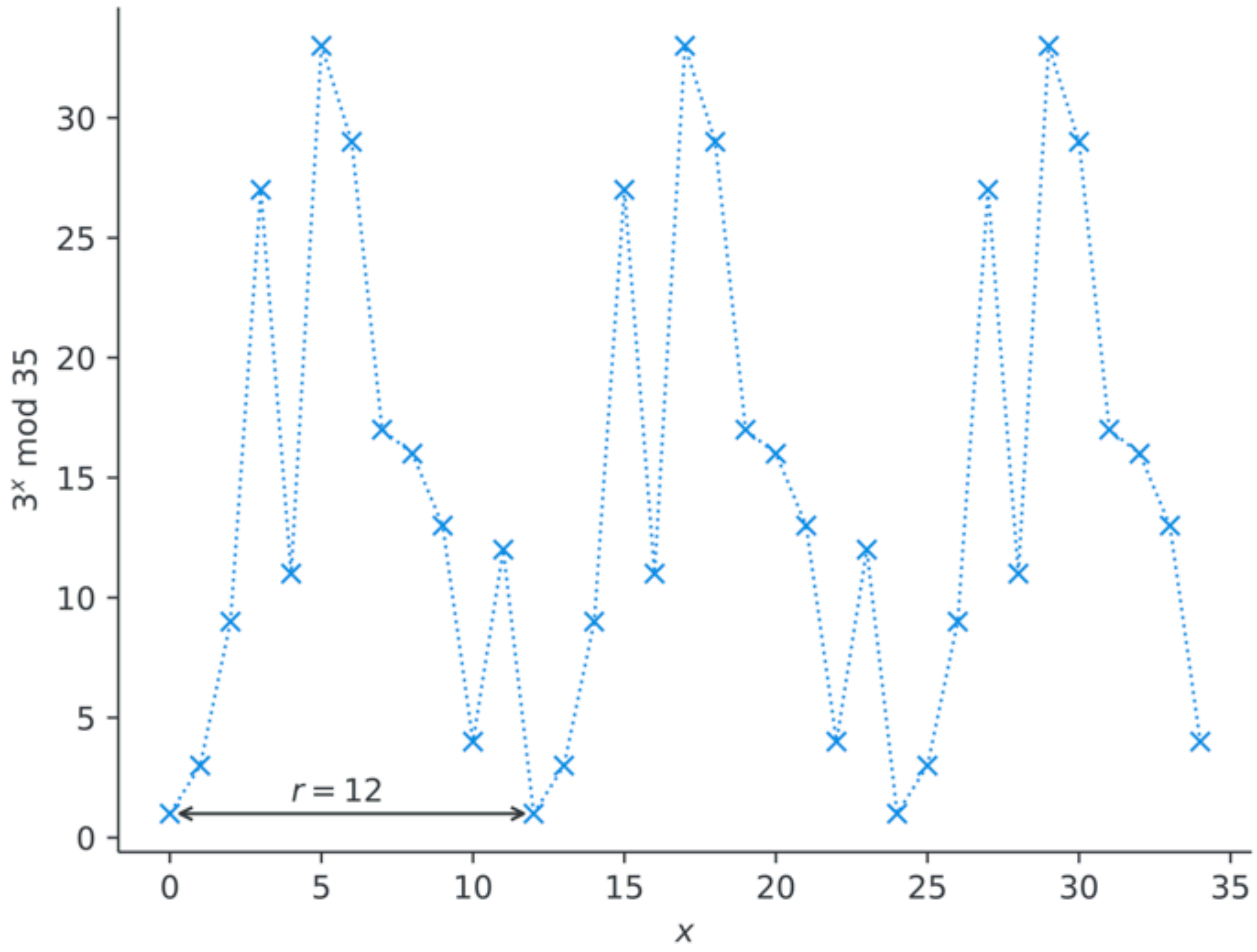
$$U|u_1\rangle = \frac{1}{\sqrt{12}}(|3\rangle + e^{-\frac{2\pi i}{12}}|9\rangle + e^{-\frac{4\pi i}{12}}|27\rangle \ldots + e^{-\frac{20\pi i}{12}}|12\rangle + e^{-\frac{22\pi i}{12}}|1\rangle)$$

$$U|u_1\rangle = e^{\frac{2\pi i}{12}} \cdot \frac{1}{\sqrt{12}}(e^{\frac{-2\pi i}{12}}|3\rangle + e^{-\frac{4\pi i}{12}}|9\rangle + e^{-\frac{6\pi i}{12}}|27\rangle \ldots + e^{-\frac{22\pi i}{12}}|12\rangle + e^{-\frac{24\pi i}{12}}|1\rangle)$$

$$U|u_1\rangle = e^{\frac{2\pi i}{12}}|u_1\rangle$$

We can see r = 12 appears in the denominator of the phase.

$3^6 = 729$, gcd(728, 35) = 7, gcd(730, 35)=5


30
25
20
15
10
5
0
3^x mod 35
r = 12
0
5
10
15
20
25
30
35
x