D.3 Shor

If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read our email,
Till we get crypto that's quantum, and daunt 'em.
— Jennifer and Peter Shor[6]

These lectures primarily follow Rieffel, Eleanor & Polak, Wolfgang, "An Introduction to Quantum Computing for Non-Physicists." (19 Jan. 2000) `http://arxiv.org/abs/quant-ph/9809016v2` [IQC].

¶1. **RSA:** The widely used RSA public-key cryptography system is based on the difficulty of factoring large numbers.

¶2. **Complexity:** The best classical algorithms are exponential in the size of the input, $m = \log M$.
Specifically, the best current (2006) algorithm (the *number field sieve algorithm*) runs in time $e^{\mathcal{O}(m^{1/3}\log^{2/3} m)}$.

¶3. Shor's algorithm is bounded error-probability quantum polynomial time (BQP), specifically, $\mathcal{O}(m^3)$.

¶4. **Period finding:** Shor's algorithm reduces factoring to finding the period of a function.

¶5. Shor's algorithm was invented in 1994, inspired by Simon's algorithm.

¶6. **QFT:** Like the classical Fourier transform, the Quantum Fourier Transform puts all the amplitude of the function into multiples of the frequency (reciprocal period).

¶7. Measuring the state yields the period with high probability.

[6] NC 216.

D.3.a Quantum Fourier transform

¶1. **Ciscoid basis:** Let f be a function defined on $[0, 2\pi)$.
You know that it can be represented in the *ciscoid* (sine and cosine) basis, $u_k(x) \stackrel{\text{def}}{=} \text{cis}(-kx) = e^{-ikx}$, where $k = 0, 1, 2, \ldots$ represents the overtone series (natural number multiples of the fundamental frequency). (The "$-$" sign is irrelevant, but will be convenient later.)

¶2. The Fourier coefficients are given by $\hat{f}_k = \langle u_k \mid f \rangle$.

¶3. **DFT:** For the *discrete Fourier transform* we suppose that f is represented by N samples, $f_j \stackrel{\text{def}}{=} f(x_j)$, where $x_j = 2\pi \frac{j}{N}$, with $j \in \mathbf{N} \stackrel{\text{def}}{=} \{0, 1, \ldots, N-1\}$.

¶4. **Discrete basis:** Likewise each of the basis functions is represented by N samples:

$$u_{kj} \stackrel{\text{def}}{=} \text{cis}(-kx_j) = e^{-2\pi ikj/N}, \ j \in \mathbf{N}.$$

¶5. **Roots of unity:** Notice that N samples of the fundamental period correspond to the N primitive N^{th}-roots of unity, that is, ω^j where $\omega = e^{2\pi i/N}$.
Hence, $u_{kj} = \omega^{-kj}$.

¶6. **Orthonormality:** It is easy to show that the $|u_k\rangle$ are orthogonal, and in fact that $|u_k\rangle/\sqrt{N}$ are ON.

¶7. Therefore, $|f\rangle$ can be represented by a Fourier series,

$$|f\rangle = \frac{1}{\sqrt{N}} \sum_{k \in \mathbf{N}} \hat{f}_k |u_k\rangle = \frac{1}{\sqrt{N}} \sum_{k \in \mathbf{N}} \langle u_k \mid f \rangle |u_k\rangle.$$

¶8. **Discrete Fourier transform:** Define the discrete Fourier transform of f, $|\hat{f}\rangle = \text{F}|f\rangle$, to be the vector of Fourier coefficients, $\hat{f}_k = \langle u_k \mid f \rangle$.

¶9. Determine F as follows:

$$\hat{f} = \begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \vdots \\ \hat{f}_{N-1} \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} \langle u_0 \mid f \rangle \\ \langle u_1 \mid f \rangle \\ \vdots \\ \langle u_{N-1} \mid f \rangle \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} \langle u_0 | \\ \langle u_1 | \\ \vdots \\ \langle u_{N-1} | \end{pmatrix} |f\rangle.$$

¶10. Therefore let

$$\mathrm{F} \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}} \begin{pmatrix} \langle u_0| \\ \langle u_1| \\ \vdots \\ \langle u_{N-1}| \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} & \cdots & \omega^{0\cdot(N-1)} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} & \cdots & \omega^{1\cdot(N-1)} \\ \omega^{2\cdot 0} & \omega^{2\cdot 1} & \cdots & \omega^{2\cdot(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{(N-1)\cdot 0} & \omega^{(N-1)\cdot 1} & \cdots & \omega^{(N-1)\cdot(N-1)} \end{pmatrix}.$$

That is, $\mathrm{F}_{kj} = \overline{u_{kj}}/\sqrt{N} = \omega^{kj}/\sqrt{N}$ for $k, j \in \mathbf{N}$.

¶11. Note that the "−" signs were eliminated by the conjugate transpose, $\langle u_k| = |u_k\rangle^\dagger$.

¶12. **Unitary:** F is unitary transformation (exercise).

¶13. **FFT:** The FFT reduces the number of operations required from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$.
It does this with a recursive algorithm that avoids recomputing values.
However, it is restricted to $N = 2^n$.

¶14. **QFT:** The QFT is even faster, $\mathcal{O}(\log^2 N)$, that is, $\mathcal{O}(n^2)$.
However, because the spectrum is encoded in the amplitudes of the state, we cannot get them all.
It too is restricted to $N = 2^n$.

¶15. The QFT transforms the amplitudes of a quantum state:

$$U_{\mathrm{QFT}} \sum_{j \in \mathbf{N}} f_j |j\rangle = \sum_{k \in \mathbf{N}} \hat{f}_k |k\rangle,$$

where $\hat{f} \stackrel{\text{def}}{=} \mathrm{F} f$.

¶16. Suppose f has period r, and suppose that $r \mid N$.
Then all all the amplitude of $\hat{f}$ should be at multiples of its fundamental frequency, N/r.

¶17. If $r \nmid N$, then the amplitude will be concentrated *near* multiples of N/r.
The approximation is improved by using larger n.

¶18. The QFT can be implemented with $n(n+1)/2$ gates of two types:
(1) One is H_j, the Hadamard transformation of the jth qubit.
(2) The other is a controlled phase-shift. Specifically $S_{j,k}$ uses qubit x_j to control whether it does a particular phase shift on the $|1\rangle$ component of qubit x_k.
That is, $S_{j,k}|x_j x_k\rangle \mapsto |x_j x'_k\rangle$ is defined by

$$S_{j,k} \stackrel{\text{def}}{=} |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| + e^{i\theta_{k-j}}|11\rangle\langle 11|,$$

where $\theta_{k-j} = \pi/2^{k-j}$.
That is, the phase shift depends on the indices j and k.

¶19. It can be shown that the QFT can be defined:[7]

$$U_{\text{QFT}} = \prod_{j=0}^{n-1} H_j \prod_{k=j+1}^{n-1} S_{j,k}.$$

This is $\mathcal{O}(n^2)$ gates.

[7] See [IQC] for this, with a detailed explanation in NC §5.1 (pp. 517–21).

D.3.b Shor's algorithm

¶1. Shor's algorithm depends on many results from number theory, which are outside of the scope of this course. Since this is not a course in cryptography or number theory, I will just illustrate the ideas.
Suppose we are factoring M (and $M = 21$ will be used for concrete examples).

¶2. **Step 1:** Pick a random number $a < M$. If a and M are not coprime (relatively prime), we are done.
(Euclid's algorithm is $\mathcal{O}(m^2) = \mathcal{O}(\log^2 M)$.)

¶3. *Example:* Suppose we pick $a = 11$, which is relatively prime with 21.

¶4. **Modular exponentiation:** Let $g(x) \stackrel{\text{def}}{=} a^x \pmod M$, for $x \in \mathbf{M} \stackrel{\text{def}}{=} \{0, 1, \ldots, M-1\}$.

¶5. This takes $\mathcal{O}(m^3)$ gates. It's the most complex part of the algorithm! (Reversible circuits typically use m^3 gates for m qubits.)

¶6. *Ex.:* In our case, $g(x) = 11^x \pmod{21}$, so

$$g(x) = \underbrace{1, 11, 16, 8, 4, 2,}_{\text{period}} 1, 11, 16, 8, 4, \ldots$$

¶7. In order to get a good QFT approximation, pick n such that $M^2 \leq 2^n < 2M^2$. Let $N = 2^n$.
Note that although the number of samples is $N = 2^n$, we need only n qubits (thanks to the tensor product).

¶8. *Ex.:* For $M = 21$ we pick $n = 9$ for $N = 512$ since $441 \leq 512 < 882$.

¶9. **Step 2 (quantum parallelism):** Apply U_g to the superposition

$$|\psi_0\rangle \stackrel{\text{def}}{=} H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x \in \mathbf{N}} |x\rangle$$

to get

$$|\psi_1\rangle \stackrel{\text{def}}{=} U_g|\psi_0\rangle|0\rangle^{\otimes m} = \frac{1}{\sqrt{N}} \sum_{x \in \mathbf{N}} |x, g(x)\rangle.$$

¶10. *Ex.:* Note that 14 qubits are required [$n = 9$ for x and $m \stackrel{\text{def}}{=} \lceil \lg M \rceil = 5$ for $g(x)$].

¶11. **Step 3 (measurement):** The function g has a period r, which we want to transfer to the amplitudes of the state so that we can apply the QFT.

¶12. This is accomplished by measuring (and discarding) the result register (as in Simon's algorithm).
Suppose the result register collapses into state g^*.
The input register will collapse into a superposition of all x such that $g(x) = g^*$. We can write it

$$|\psi_2\rangle \stackrel{\text{def}}{=} \frac{1}{Z} \sum_{x \in \mathbf{N}} f(x)|x, g^*\rangle = \left[\frac{1}{Z} \sum_{x \in \mathbf{N}} f(x)|x\rangle \right] |g^*\rangle,$$

where

$$f(x) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } g(x) = g^* \\ 0, & \text{otherwise} \end{cases},$$

and $Z \stackrel{\text{def}}{=} \sqrt{|\{x \mid g(x) = g^*\}|}$ is a normalization factor.

¶13. Note that the values x for which $f(x) \neq 0$ differ from each other by the period.
As in Simon's algorithm, if we could measure two such x, we would have useful information, but we can't.

¶14. Note: As it turns out, the preceding measurement of the result register can be avoided. This is in general true for "internal" measurement processes in quantum algorithms (Bernstein & Vazirani 1997).

¶15. *Ex.:* Suppose we measure the result register and get $g^* = 8$.
Fig. III.24 shows the corresponding f.

¶16. **Step 4 (QFT):** Apply the QFT to obtain,

$$\begin{aligned} |\psi_3\rangle &\stackrel{\text{def}}{=} U_{\text{QFT}} \left(\frac{1}{Z} \sum_{x \in \mathbf{N}} f(x)|x\rangle \right) \\ &= \frac{1}{Z} \sum_{\hat{x} \in \mathbf{N}} \hat{f}(\hat{x})|\hat{x}\rangle. \end{aligned}$$

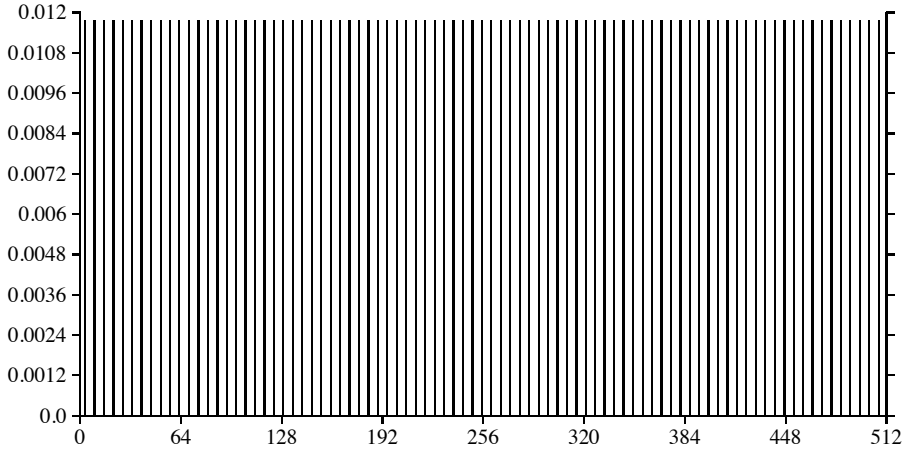

Figure III.24: Example probability distribution $|f(x)|^2$ for state $Z^{-1}\sum_{x\in\mathbf{N}} f(x)|x, 8\rangle$. In this example the period is $r = 6$ (e.g., at $x = 3, 9, 15, \ldots$). [fig. source: IQC]

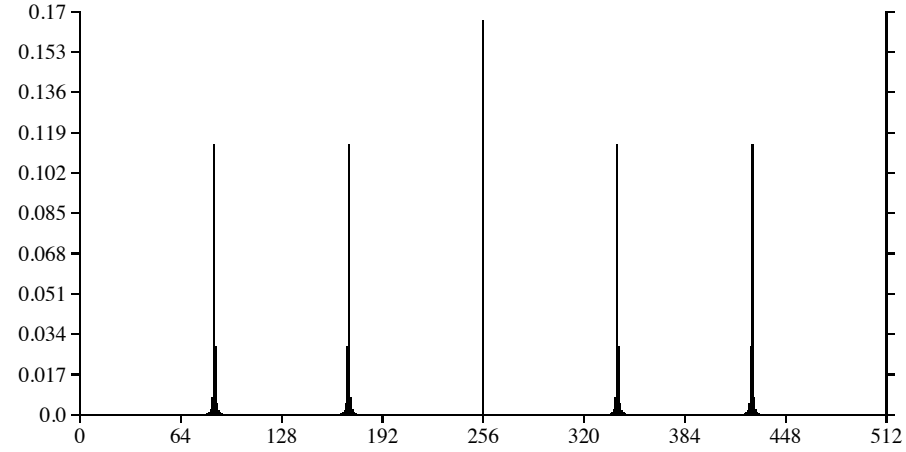

Figure III.25: Example probability distribution $|\hat{f}(\hat{x})|^2$ of the quantum Fourier transform of $f(x)$. The spectrum is concentrated near multiples of $N/6 = 512/6 = 85\ 1/3$, that is $85\ 1/3, 170\ 2/3, 256$, etc. [fig. source: IQC]

(The collapsed result register $|g^*\rangle$ has been omitted.)

¶17. If the period r divides $N = 2^n$, then $\hat{f}$ will be nonzero only at multiples of the fundamental frequency N/r.
That is, the nonzero components will be $|kN/r\rangle$.

¶18. If it doesn't divide, then the amplitude will be concentrated around these $|kN/r\rangle$.

¶19. *Ex.:* See Fig. III.24 and Fig. III.25 for examples of the probability distributions $|f(x)|^2$ and $|\hat{f}(\hat{x})|^2$.

¶20. **Step 5 (period extraction):** Measure the state in the computational basis.

¶21. **Period a power of 2:** If $r \mid N$, then the resulting state will be $v \stackrel{\text{def}}{=} |kN/r\rangle$ for some $k \in \mathbf{N}$.

¶22. Therefore $k/r = v/N$.

¶23. If k and r are relatively prime, as is likely, then reducing the fraction v/N to lowest terms will produce r in the denominator.
In this case the period is discovered.

¶24. **Period not a power of 2:** In this case, it's often possible to guess the period from a continued fraction expansion of v/N.[8]

¶25. *Ex.:* Suppose the measurement returns $v = 427$, which is not a power of two.
This is the result of the continued fraction expansion (see IQC):

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632
2	5	5	6	0.02352941
3	42	211	253	0.5

"which terminates with $6 = q_2 < M \leq q_3$. Thus, $q = 6$ is likely to be the period of f." [IQC]

[8]See Rieffel & Polak (App. B) for an explanation of this procedure and citations for why it works.

¶26. **Step 6 (finding a factor):** If the guess q is even, then $a^{q/2}+1$ and $a^{q/2}-1$ are likely to have common factors with M.
Use the Euclidean algorithm to check this.

¶27. **Reason:** If q is the period of $g(x) = a^x \pmod M$, then $a^q = 1 \pmod M$. This is because, if q is the period, then for all x, $g(x+q) = g(x)$, that is, $a^{q+x} = a^q a^x = a^x \pmod M$ for all x.

¶28. Therefore $a^q - 1 = 0 \pmod M$. Hence,

$$(a^{q/2}+1)(a^{q/2}-1) = 0 \pmod M.$$

Therefore, unless one of the factors is a multiple of M (and hence $= 0 \bmod M$), one of them has a nontrivial common factor with M.

¶29. *Ex.:* The continued fraction gave us a guess $q = 6$, so with $a = 11$ we should consider $11^3 + 1 = 1332$ and $11^3 - 1 = 1330$.
For $M = 21$ the Euclidean algorithm yields $\gcd(21, 1332) = 3$ and $\gcd(21, 1330) = 7$.
We've factored 21!

¶30. **Iteration:** There are several reasons that the preceding steps might not have succeeded:
(1) The value v projected from the spectrum might not be close enough to a multiple of N/r (¶24).
(2) In ¶23, k and r might not be relatively prime, so that the denominator is only a factor of the period, but not the period itself.
(3) In ¶28, one of the two factors turns out to be a multiple of M.
(4) In ¶26, q was odd.

¶31. In these cases, a few repetitions of the preceding steps yields a factor of M.

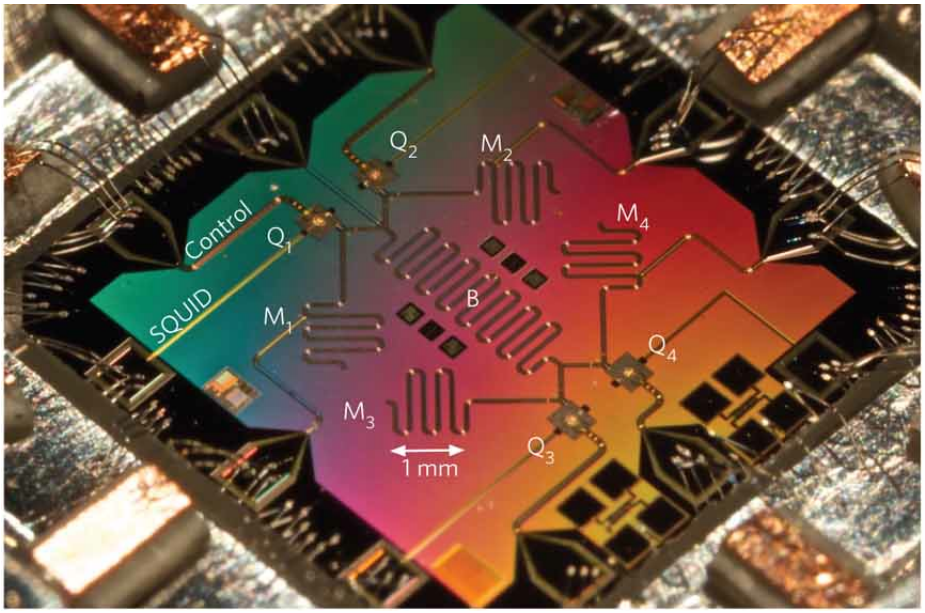

Figure III.26: Hardware implementation of Shor's algorithm developed at UCSB (2012). The M_j are quantum memory elements, B is a quantum "bus," and the Q_j are phase qubits that can be used to implement qubit operations between the bus and memory elements. [source: CPF]

D.3.c Recent progress

To read our E-mail, how mean
of the spies and their quantum machine;
be comforted though,
they do not yet know
how to factorize twelve or fifteen.
— Volker Strassen[9]

This lecture is based on Erik Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland & John M. Martinis, "Computing prime factors with a Josephson phase qubit quantum processor." *Nature Physics* **8**, 719–723 (2012) `doi:10.1038/nphys2385` [CPF].

[9] NC 216.

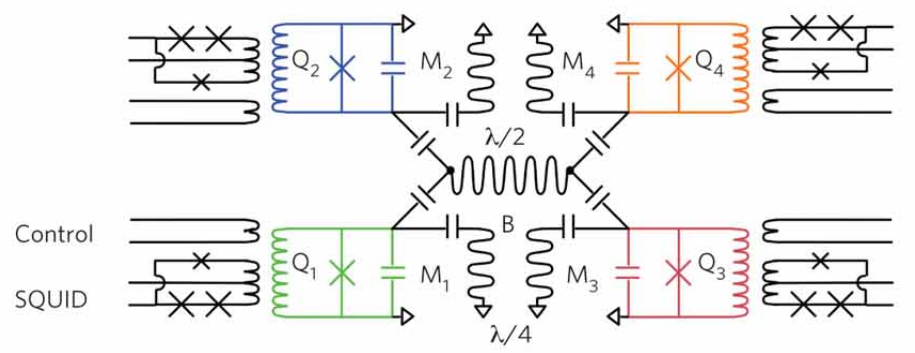

Figure III.27: Circuit of hardware implementation of Shor's algorithm developed at UCSB. [source: CPF]

¶1. In Aug. 2012 a group at UC Santa Barbara described a quantum implementation of Shor's algorithm that correctly factored 15 about 48% of the time (50% being the theoretical success rate).
(There have been NMR hardware factorizations of 15 since 2001, but there is some doubt if entanglement was involved.)

¶2. This is a 3-qubit compiled version of Shor's algorithm.
"Compiled" means that the implementation of modular exponentiation is for fixed M and a.

¶3. This case used fixed $a = 4$ as the coprime to $M = 15$.
In this case the correct period $r = 2$.

¶4. The device (Fig. III.26) has nine quantum devices, including four phase qubits and five superconducing co-planar waveguide (CPW) resonators.

¶5. The four CPWs (M_j) can be used as memory elements and fifth (B) can be used as a "bus" to mediate entangling operations.

¶6. In effect the qubits Q_j can be read and written.
Radiofrequency pulses in the bias coil can be used to adjust the qubit's frequency.
Gigahertz pulses can be used to manipulate and measure the qubit's state.
SQUIDs are used for one-shot readout of the qubits.

¶7. The qubits Q_j can be tuned into resonance with the bus B or memory elements M_j.

¶8. **Qubit gates:** The quantum processor can be used to implement the single-qubit gates X, Y, Z, H, and the two-qubit swap (iSWAP) and controlled-phase (C_ϕ) gates.

¶9. **Entanglement:** The entanglement protocol can be scaled to an arbitrary number of qubits.

¶10. **Relaxation and dephasing times:** about 200ns.