Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék

A kvantum Fourier-transzformáció és alkalmazása

Készítette:
Varga Balázs

2015. december 9.

Tartalomjegyzék

Kivonat

Dolgozatom célja a kvantum Fourier-transzformáció működésének és lehetséges alkalmazásainak részletes bemutatása, különös tekintettel napjaink egyik legelterjedtebb nyilvános kulcsú titkosítási eljárása, a Rivest–Shamir–Adleman-algoritmus (RSA) feltörésére.

Az első fejezetben röviden összefoglalom a klasszikus Fourier-transzformáció folytonos és diszkrét idejű változatának tulajdonságait, majd ebből kiindulva a második fejezetben bevezetem a kvantum Fourier-transzformáció fogalmát, és részletesen bemutatom a transzformációt megvalósító kvantumáramkör tervezésének lépéseit. Az algoritmus működőképességét számítógépes szimulációval is illusztrálom.

A harmadik fejezetben bemutatom a Fourier-transzformációra épülő egyik legnagyobb jelentőségű kvantumáramkör, a fázisbecslő ideális és valós körülmények közötti működését, valamint a megvalósítása során felmerülő mérnöki feladatokat.

A negyedik fejezetben az RSA-kód feltörésének lehetőségeivel foglalkozom. Röviden összefoglalom az RSA-eljárás működését, valamint klasszikus módszerekkel történő feltörésének lehetőségeit és matematikai akadályait, majd ezeket számítási példákkal is illusztrálom. Végül megmutatom, hogy egy kvantumszámítógépen implementált speciális fázisbecslő eljárás, a Shor-algoritmus segítségével a feltörés nem ütközik elvi akadályokba.

1. Klasszikus Fourier-transzformáció

A 18. században a kor három jeles matematikusa, d'Alembert, Bernoulli és Euler a megpengetett húr differenciálegyenletének megoldását kapták meg trigonometrikus sor alakjában, ám a megoldás helyességét nem mindannyian fogadták el. Eredményeiket azonban Jean-Baptiste Joseph Fourier francia matematikus és fizikus sikerrel használta fel hővezetési problémák analitikus megoldása során, és a ma Fourier-analízisként ismert matematikai módszer alapjainak lefektetését is az ő 1822-ben publikált értekezésének tulajdonítják.

1.1. Fourier-transzformáció folytonos időben

Az $x(t)$ abszolút integrálható függvény Fourier-transzformáltját (spektrumát) definiáljuk[1] a következőképpen:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{j2\pi ft}\, dt$$

Az inverz transzformáció pedig:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{-j2\pi ft}\, df$$

A folytonos Fourier-transzformáció a matematika és az alkalmazott tudományok számos területén használatos: leegyszerűsíti bizonyos differenciálegyenletek megoldását (és ezáltal a lineáris rendszerek analízisét), segítségével számítható egy valószínűségi eloszlás karakterisztikus függvénye, a kvantummechanikában a Fourier-transzformáció teremti meg a kapcsolatot egy részecske impulzusára és helyére vonatkozó hullámfüggvények között, valamint optikai lencsék képalkotása is vizsgálható Fourier-transzformációval. Mindazonáltal a kvantuminformatikai algoritmusok szempontjából a Fourier-transzformáció alábbiakban tárgyalt, diszkrét idejű változata (DFT) a nagyobb jelentőségű.

1.2. Diszkrét Fourier-transzformáció

Az N elemű $\mathbf{x}$ vektor diszkrét Fourier-transzformáltja az $\mathbf{y}$ vektor, ha

$$y_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} x_i e^{j\frac{2\pi}{N}ik}.$$

[1] Valójában több definíció is létezik. A legtöbb szakirodalom a kitevők előjelét fordítva adja meg, és a kifejezésekben konstans szorzók is szerepelhetnek. A jelen dokumentumban használt definíciók és jelölések az [1] irodalomban használtakkal konzisztensek. Megjegyzem azonban, hogy amíg ez a definíció a kvantuminformatika területén tipikusnak tekinthető, addig például a digitális jelfeldolgozás szakemberei általában más definícióhoz ragaszkodnak.

Vezessük be az $\omega = e^{j\frac{2\pi}{N}}$ jelölést az N-edik komplex egységgyökre! Ekkor az $\mathbf{Fx} = \mathbf{y}$ transzformáció operátorának mátrixa az alábbi formát ölti:

$$\mathbf{F} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

Könnyen ellenőrizhető, hogy az $\mathbf{F}$ operátor unitér, azaz inverze megegyezik az adjungáltjával: $\mathbf{x} = \mathbf{F}^{-1}\mathbf{y} = \mathbf{F}^{\dagger}\mathbf{y}$. Az inverz transzformáció (IDFT) képlete tehát:

$$x_i = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{-j\frac{2\pi}{N}ik}.$$

Mivel bizonyos feltételek betartása esetén a diszkrét Fourier-transzformált a folytonos spektrum mintavételes közelítését adja, a DFT jelentősége a mérnöki gyakorlatban óriási, a digitális jel- és képfeldolgozás minden területén megjelenik. Számunkra azonban a transzformáció unitér volta a legfontosabb – ez ugyanis azt is jelenti, hogy nincs elvi akadálya a DFT kvantumáramkörrel történő megvalósításának. A következő fejezetben erre példát is láthatunk.

Érdekesség, hogy bár logikailag nem feltétlenül így gondolnánk, de a ma diszkrét Fourier-transzformációként emlegetett eljárás időben előbb született meg, mint a folytonos Fourier-transzformáció. Sőt, Gauss 1805-ben csillagászati számításai során olyan módszert alkalmazott, amely nagyon hasonlított a DFT gyors elvégzésére (FFT) napjainkban leginkább elterjedt Cooley–Tukey-algoritmushoz.

2. Kvantum Fourier-transzformáció

2.1. A QFT bevezetése és megvalósítása kvantumáramkörrel

Legyen egy kvantummechanikai rendszer állapota $|\varphi\rangle$, amely a $|i\rangle$, $i = 0 \dots N-1$ bázisállapotok szuperpozíciójaként áll elő:

$$|\varphi\rangle = \sum_{i=0}^{N-1} \varphi_i |i\rangle$$

A $|\varphi\rangle$ szuperponált állapot – ebben a természetes bázisban – egyértelműen megadható a φ_i komplex valószínűségi amplitúdók vektoraként. Vegyük ezen vektor DFT-jét:

$$\psi_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \varphi_i e^{j\frac{2\pi}{N}ik}$$

Az így kapott együtthatókból mint valószínűségi amplitúdókból konstruált $|\psi\rangle$ kvantumállapotot nevezzük a $|\varphi\rangle$ állapot Fourier-transzformáltjának (QFT-jének):

$$\mathbf{F}|\varphi\rangle = |\psi\rangle = \sum_{k=0}^{N-1} \psi_k |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} \varphi_i e^{j\frac{2\pi}{N}ik} |k\rangle$$

Az $\mathbf{F}$ operátor unitér, azaz $|\varphi\rangle = \mathbf{F}^{-1}|\psi\rangle = \mathbf{F}^{\dagger}|\psi\rangle$. Ez alapján az inverz kvantum Fourier-transzformáció (IQFT) a klasszikus IDFT-hez hasonló módon számítható:

$$\varphi_i = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \psi_k e^{-j\frac{2\pi}{N}ik}$$

A továbbiakban a QFT-t elemi kapukból álló kvantumáramkörrel szeretnénk realizálni[2]. Ehhez fel kell tennünk, hogy a bemeneti $|\varphi\rangle$ állapot egy n qubites kvantumregiszter, azaz $N = 2^n$. Kihasználjuk továbbá a Hilbert-tér linearitását: elegendő olyan kvantumáramkört terveznünk, amely egy $|i\rangle$ bázisállapot transzformáltját állítja elő; ugyanez az áramkör a bázisállapotok tetszőleges lineáris kombinációja – azaz tetszőleges szuperponált $|\varphi\rangle$ állapot – esetén is működni fog. A $|i\rangle$ bázisállapot QFT-je:

$$\mathbf{F}|i\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}ik} |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(j\frac{2\pi}{2^n} i \sum_{l=1}^{n} k_l 2^{n-l} \right) |k\rangle$$

[2] Az itt bemutatott QFT-realizáció R. Griffiths és C-S. Niu, illetve R. Cleve, A. Ekert, C. Macchiavello és M. Mosca nevéhez fűződik. A Fourier-transzformáció kvantumos kiszámításáról szóló első publikációk 1994-ben jelentek meg.

Itt kihasználtuk, hogy a k természetes szám bináris alakja[3]: $k = \overline{k_1 k_2 \dots k_n} = k_1 2^{n-1} + k_2 2^{n-2} + \dots + k_n 2^0$, ahol $k_l \in \{0;1\}$. Hasonlóan, a $|k\rangle$ bázisállapot felírható n kvantumbit tenzorszorzataként: $|k\rangle = |k_1 k_2 \dots k_n\rangle = |k_1\rangle \otimes |k_2\rangle \otimes \dots \otimes |k_n\rangle$, ahol $|k_l\rangle \in \{|0\rangle; |1\rangle\}$. Ezek alapján:

$$\mathbf{F}|i\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \prod_{l=1}^{n} e^{j2\pi i k_l 2^{-l}} \bigotimes_{l=1}^{n} |k_l\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \bigotimes_{l=1}^{n} e^{j2\pi i k_l 2^{-l}} |k_l\rangle$$

Csoportosítsuk át a kifejezést úgy, hogy szétválasztjuk azokat az eseteket, amikor $k_l = 0$, illetve $k_l = 1$:

$$\mathbf{F}|i\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left(e^{j2\pi i \cdot 0 \cdot 2^{-l}} |0\rangle + e^{j2\pi i \cdot 1 \cdot 2^{-l}} |1\rangle \right) = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left(|0\rangle + e^{j2\pi i 2^{-l}} |1\rangle \right)$$

A tenzorszorzat egyes tényezőire vezessük be a $|\mu_l\rangle$ jelölést!

$$|\mu_l\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{j2\pi i 2^{-l}} |1\rangle \right) = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{j2\pi 0,\overline{i_{n-l+1} \cdots i_{n-1} i_n}} |1\rangle \right)$$

Az átalakításnál azt használtuk ki, hogy az i természetes szám bináris alakja $i = \overline{i_1 i_2 \dots i_n} = i_1 2^{n-1} + i_2 2^{n-2} + \dots + i_n 2^0$, ahol $i_m \in \{0;1\}$ az i szám m-edik bitje. Ennek a 2^l-edrésze: $i2^{-l} = \overline{i_1 i_2 \dots i_{n-l}\,, n_{n-l+1} \dots i_n} = \overline{i_1 i_2 \dots i_{n-l}} + \overline{0{,}n_{n-l+1} \dots i_n}$. Mivel ez a kifejezés a komplex exponenciális kitevőjében szerepel, az egészrész elhagyható – az ugyanis az egységvektor egész számú fordulatát írja le a komplex síkon, a szöget csak a törtrész fogja megváltoztatni.

A fenti megfontolásokkal megkaptuk a kvantum Fourier-transzformáció megvalósítás szempontjából érdekes, végső alakját:

$$\mathbf{F}|i\rangle = |\mu_1\rangle \otimes |\mu_2\rangle \otimes \dots \otimes |\mu_n\rangle =$$

$$= \left(\frac{|0\rangle + e^{j2\pi 0,\overline{i_n}} |1\rangle}{\sqrt{2}} \right) \otimes \left(\frac{|0\rangle + e^{j2\pi 0,\overline{i_{n-1} i_n}} |1\rangle}{\sqrt{2}} \right) \otimes \dots \otimes \left(\frac{|0\rangle + e^{j2\pi 0,\overline{i_1 i_2 \cdots i_n}} |1\rangle}{\sqrt{2}} \right)$$

Ez a formula a qubitek szintjén definiálja a bemenet és a Fourier-transzformált közti kapcsolatot. Például az első qubit:

$$|\mu_1\rangle = \begin{cases} \frac{|0\rangle + |1\rangle}{\sqrt{2}}, & \text{ha } i_n = 0 \\ \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{ha } i_n = 1, \end{cases}$$

tehát kimenet MSB-je a bemenet LSB-jéből egyetlen Hadamard-kapuval képezhető. Néz-

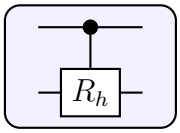
[3]Felülhúzással jelölöm, hogy a k_l számokat számjegyekként (most speciálisan bitekként), nem pedig egy szorzat tényezőiként kell olvasni.

zük most $|\mu_2\rangle$-t!

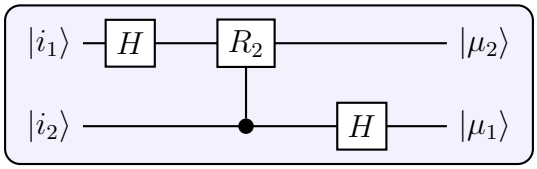
$$|\mu_2\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle e^{j2\pi 0,\overline{i_{n-1}}} \cdot \begin{cases} 1, & \text{ha } i_n = 0 \\ e^{j2\pi 2^{-2}}, & \text{ha } i_n = 1 \end{cases}$$

Ez $i_n = 0$ esetben ismét egy Hadamard-kapunak felel meg, azonban $i_n = 1$ esetben szükség van egy $2\pi 2^{-2}$ szögű fáziskapura is. Vezessük be az R_h vezérelt fáziskaput, amely úgy működik, hogy a vezérlőbemenet $|0\rangle$ volta esetén transzparens, míg $|1\rangle$ vezérlés esetén $2\pi 2^{-h}$ szöggel forgatja el a bemenet $|1\rangle$-hez tartozó komplex valószínűségi amplitúdójának fázisát! Ezen operátor mátrixa a $\{|0\rangle; |1\rangle\}^{\otimes 2}$ természetes bázisban:

$$\mathbf{R_h} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{j2\pi 2^{-h}} \end{bmatrix}$$



Az így definiált operátor mint áramköri elem segítségével már megrajzolhatjuk a két qubites kvantum Fourier-transzformátor sémáját:

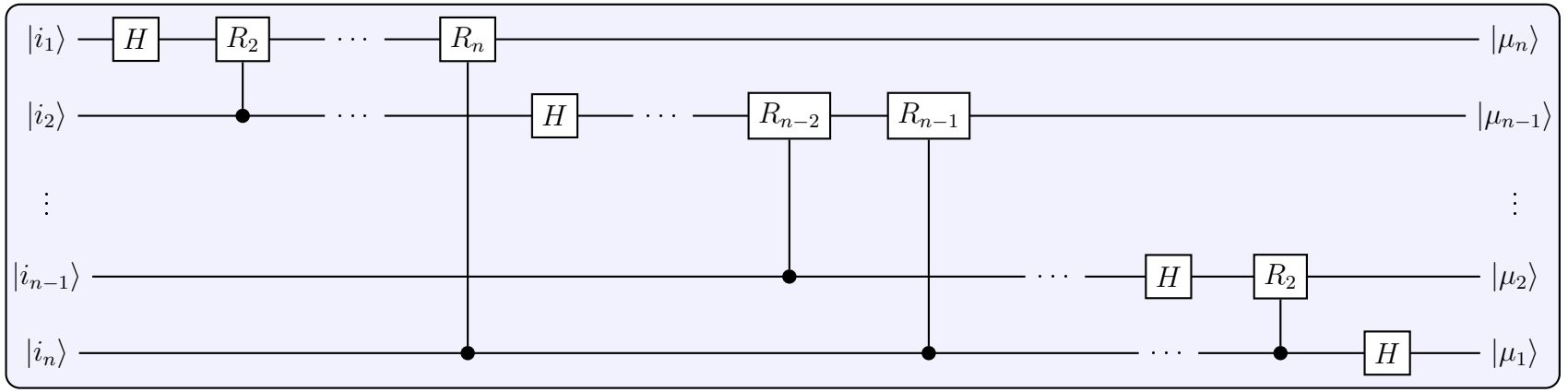


1. ábra. *Kétbites QFT áramkör*

Az általános, n qubites bemenettel működő QFT áramkör felépítése ugyanezen sémát követve rekurzívan megkapható az $n-1$ qubites változatból. A teljes áramkör rajza a 2. ábrán látható.

A fentiekben konstruáltunk egy kvantumáramkört, amely előállítja egy tetszőleges állapot QFT-jét. Van azonban egy nyilvánvaló hátránya: az algoritmus nem használható a hagyományos DFT gyorsabb kiszámítására, hiszen a Fourier-együtthatók komplex valószínűségi amplitúdókként jelennek meg a kimeneten, amelyeknek a fázisa makroszkopikusan mérhető jelentést nem hordoz. Ráadásul a számításigénnyel sem spóroltunk: a QFT $\mathrm{O}(n^2)$ komplexitású, míg a hagyományos FFT-nek csak $\mathrm{O}(n \log n)$ a lépésszáma[4]. A későbbiekben azonban látni fogjuk, hogy a QFT számos kvantumalgoritmus elengedhetetlen építőeleme.

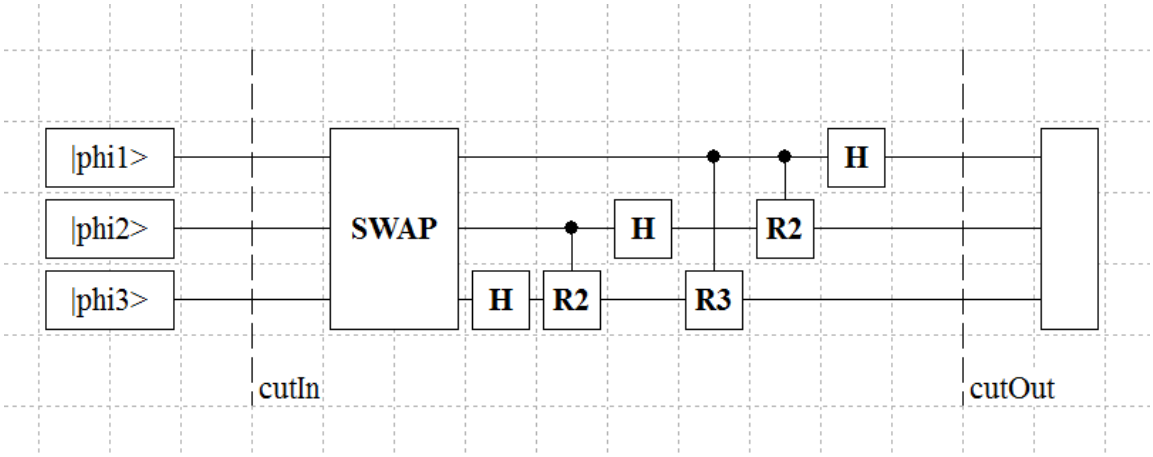
[4] A kettő között azonban van egy fontos különbség: a hagyományos információelméletben egy algoritmus számításigényén a végrehajtásához szükséges lépések számát értjük (amely arányos a futásidővel), egy kvantumalgoritmus számításigénye pedig a kvantumáramkör felépítéséhez szükséges elemi kapuk számát jelenti.



2. ábra. *A QFT teljes realizációja kvantumhálózattal*

2.2. Az algoritmus működésének szimulációja

Az alábbiakban egy számítógépes szimulációt láthatunk a három qubites QFT algoritmus működésére. A szimuláció a QCircuit nevű szoftver használatával készült, amelyet a BME egykori hallgatója, Pereszlényi Attila készített diplomamunkája keretében 2005-ben. A program igen sokoldalú, számos kvantumkaput és -csatornát tartalmaz, így kiválóan alkalmas különböző algoritmusok és kommunikációs protokollok vizsgálatára.



3. ábra. *A szimulátorban megépített QFT áramkör*

A megépített hálózat megfelel az előzőekben látott sémának. Az egyetlen új elem a SWAP kapu, amely csak kényelmi célokat szolgál: megfordítja a qubitek sorrendjét, hogy a kapott szimulációs eredmények jobban olvashatóak legyenek. A programban mátrixukkal definiálhatunk egyedi kapukat, a három qubites SWAP kapu esetében a megadandó 8×8-as mátrix az alábbi permutáló mátrix:

$$\mathbf{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A *cutIn* és *cutOut* megfigyelőpontok arra szolgálnak, hogy nyomon követhessük az egyes bemeneti és kimeneti qubitek állapotait. Természetesen ez csak szimulációs eszköz, makroszkopikusan az ilyen megfigyelés több akadályba is ütközne: egyrészt a komplex valószínűségi amplitúdókat nem is mérhetnénk meg, másrészt pedig bármilyen mérés befolyásolná a kvantumbitek állapotait.

Az áramkört a $|\varphi_1\rangle = 0{,}8 \cdot |0\rangle + 0{,}6 \cdot |1\rangle$, $|\varphi_2\rangle = 0{,}6 \cdot |0\rangle + 0{,}8 \cdot |1\rangle$ és $|\varphi_3\rangle = |1\rangle$ bemeneti állapotokkal inicializáltam, és a működés ellenőrzésére az alábbi egyszerű MATLAB szkriptet írtam:

```
phi1=[0.8 ; 0.6];
phi2=[0.6 ; 0.8];
phi3=[0.0 ; 1.0];
in=kron(kron(phi1,phi2),phi3) % Kronecker-féle tenzorszorzat
out=1/sqrt(8)*conj(fft(in)) % az eltérő definíciók miatt konjugálni kell!
```

A szimuláció eredményét, valamint a MATLAB szkript futásának kimenetét az alábbi táblázat mutatja. Látható, hogy a QFT áramkör helyesen működik, sikerrel előállította az elvárt eredményt.

QCircuit	MATLAB
Simulation results ========== CUT WATCH Name: cutIn 000:[0.0000 + 0.0000i] 001:[0.4800 + 0.0000i] 010:[0.0000 + 0.0000i] 011:[0.6400 + 0.0000i] 100:[0.0000 + 0.0000i] 101:[0.3600 + 0.0000i] 110:[0.0000 + 0.0000i] 111:[0.4800 + 0.0000i] CUT WATCH Name: cutOut 000:[0.6930 + 0.0000i] 001:[-0.0100 + 0.0700i] 010:[0.0000 - 0.0990i] 011:[0.0100 + 0.0700i] 100:[-0.6930 + 0.0000i] 101:[0.0100 - 0.0700i] 110:[0.0000 + 0.0990i] 111:[-0.0100 - 0.0700i]	in = 0 0.4800 0 0.6400 0 0.3600 0 0.4800 out = 0.6930 -0.0100 + 0.0700i 0 - 0.0990i 0.0100 + 0.0700i -0.6930 0.0100 - 0.0700i 0 + 0.0990i -0.0100 - 0.0700i

3. Kvantum-fázisbecslés

Ismeretes, hogy egy unitér mátrix minden sajátértéke rajta van a komplex egységkörön, vagyis egyetlen fázisszöggel jellemezhető. A fázisbecslés vagy sajátértékbecslés (QPE – *Quantum Phase Estimation*) feladata, hogy az $\mathbf{U}$ unitér transzformáció $|u\rangle$ sajátvektorához tartozó $e^{j\alpha_u}$ sajátértékének α_u fázisát előállítsa. Ha a transzformáció mátrixa adott, akkor természetesen fölösleges a feladathoz kvantumszámítógépet építenünk, azonban a gyakorlatban arról van szó, hogy az $\mathbf{U}$ operátort egy kvantumáramkör valósítja meg, és valamilyen feltételektől függően értékeli ki, tehát mátrixát nem ismerjük előre.

Praktikus megfontolások miatt a fázist $\alpha_u = 2\pi\kappa_u$ alakban írjuk fel, és a $\kappa_u \in [0;1[$ számot keressük. Egyelőre feltételezzük, hogy $\kappa_u = \frac{i}{2^n}$ alakú, ahol $i \in \{0, 1, \ldots, 2^n - 1\}$. Az előzőekben láttuk, hogy a $|i\rangle$ bázisállapot kvantum Fourier-transzformáltja:

$$\mathbf{F}|i\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}ik}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{j2\pi\frac{i}{2^n}k}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{j2\pi\kappa_u k}|k\rangle$$

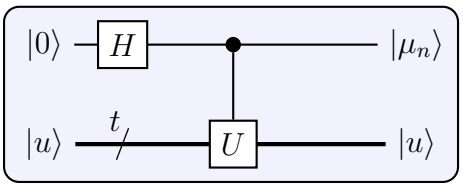
Ha ezt az állapotot elő tudnánk állítani, akkor egy IQFT végrehajtásával megkaphatnánk $|i\rangle$-t, amelyből κ_u számítható. Az előállítandó állapot l-edik qubitje:

$$|\mu_l\rangle = \frac{|0\rangle + e^{j2\pi i 2^{-l}}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{j2\pi 2^{n-l}\kappa_u}|1\rangle}{\sqrt{2}}$$

A legalacsonyabb helyiértékű qubit (LSB):

$$|\mu_n\rangle = \frac{|0\rangle + e^{j2\pi\kappa_u}|1\rangle}{\sqrt{2}}$$

Ezt az állapotot a 4. ábrán látható kvantumáramkörrel generálhatjuk. A hálózat tartalmaz egy vezérelt U kaput: ez úgy működik, hogy $|0\rangle$ vezérlőállapot esetén transzparens, $|1\rangle$ vezérlőállapot esetén pedig végrehajtja azt a t qubites $\mathbf{U}$ transzformációt, amelynek a sajátértékét keressük.



4. ábra. *A legalsó qubit előállítása*

Az áramkör működése nem triviális – ahogy az lenni szokott, amikor egy vezérelt kapu vezérlőbemenetére szuperponált állapotot teszünk[5]. A bemenet a $|0\rangle \otimes |u\rangle$ állapot,

[5]Ilyenkor gyakran találkozunk érdekes jelenségekkel. Szuperpozícióval vezérelt CNOT kapuval például összefonódott Bell-párokat állíthatunk elő, sőt teleportálhatunk is.

amelynek első qubitjéből a Hadamard-kapu a $|+\rangle$ állapotot képzi:

$$(\,\mathbf{H}|0\rangle\,)\otimes|u\rangle=\frac{|0\rangle+|1\rangle}{\sqrt{2}}\otimes|u\rangle=\frac{|0\rangle|u\rangle+|1\rangle|u\rangle}{\sqrt{2}},$$

ez jut a vezérelt U kapu bemenetére. A linearitás miatt a kapu működését az összeg két tagjára külön-külön vizsgálhatjuk. Amikor a felső qubit $|0\rangle$, akkor az U kapu ki van kapcsolva, amikor pedig $|1\rangle$, akkor az U kapu működik, és mivel a bemenetén a $|u\rangle$ sajátállapot van, $e^{j2\pi\kappa_u}$-val való szorzást valósít meg:

$$\mathbf{U}\left(\frac{|0\rangle|u\rangle+|1\rangle|u\rangle}{\sqrt{2}}\right)=\frac{|0\rangle|u\rangle+|1\rangle\otimes(\,\mathbf{U}|u\rangle\,)}{\sqrt{2}}=\frac{|0\rangle|u\rangle+|1\rangle\otimes(e^{j2\pi\kappa_u}|u\rangle)}{\sqrt{2}}=$$

$$=\frac{|0\rangle+e^{j2\pi\kappa_u}|1\rangle}{\sqrt{2}}\otimes|u\rangle=|\mu_n\rangle|u\rangle,$$

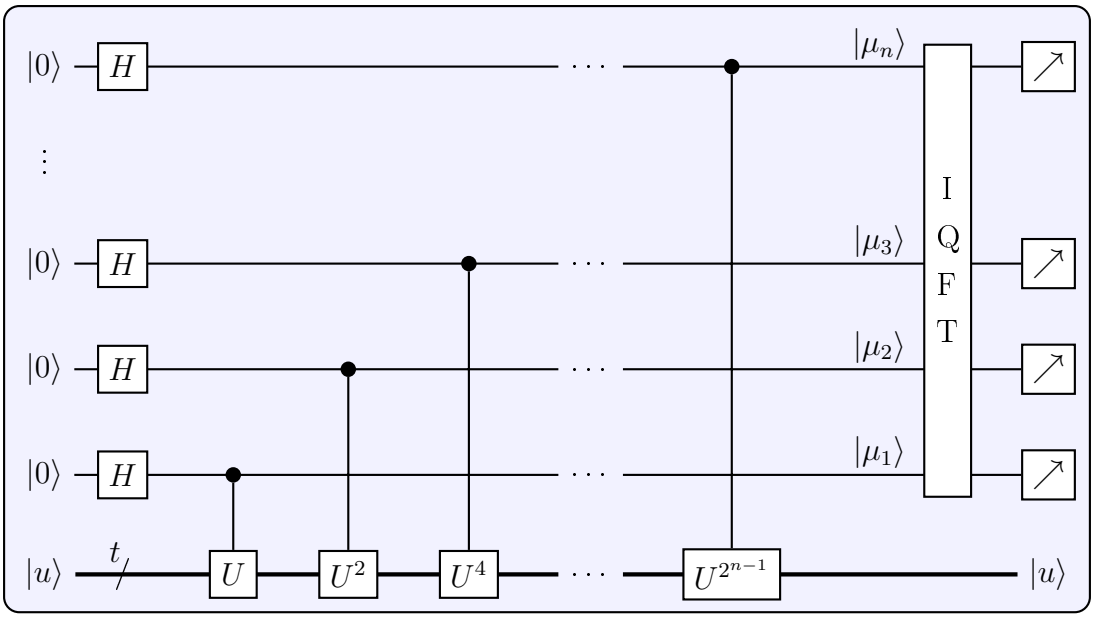
tehát a kimeneten valóban a kívánt állapot jelenik meg.

A többi $|\mu_l\rangle$ állapot előállítása hasonló módon történik, az egyetlen különbség, hogy az $\mathbf{U}$ operátort többször kell működtetni:

$$|\mu_l\rangle=\frac{|0\rangle+e^{j2\pi 2^{n-l}\kappa_u}|1\rangle}{\sqrt{2}}=\frac{|0\rangle+\left(e^{j2\pi\kappa_u}\right)^{2^{n-l}}|1\rangle}{\sqrt{2}}$$

$$\mathbf{U}^{2^{n-l}}|u\rangle=\underbrace{e^{j2\pi\kappa_u}e^{j2\pi\kappa_u}\dots e^{j2\pi\kappa_u}}_{2^{n-l}\text{ tényező}}|u\rangle=\left(e^{j2\pi\kappa_u}\right)^{2^{n-l}}|u\rangle$$

Ezekkel a megfontolásokkal már felépíthetjük az összes $|\mu_l\rangle$ kvantumbitet előállító, általános áramkört, amely az 5. ábrán látható.



5. ábra. *Általános sajátértékbecslő áramkör*

Az előállított $|\mu_l\rangle$ kvantumbitekből az inverz Fourier-transzformáció a $|i\rangle$ bázisállapotot képzi, amelyet projektív méréssel biztosan azonosítani is tudunk. Ezután a keresett fázistényező $\kappa_u = \frac{i}{2^n}$ módon számítható.

Ha a fázisra vonatkozó kezdeti feltételezésünket elvetjük, és bármilyen $\kappa_u \in [0; 1[$ számot megengedünk, akkor az áramkör nyilvánvalóan nem fog teljes bizonyossággal működni. Erre az esetre az [1] irodalomban minden részletre kiterjedő hibaanalízist olvashatunk, amelytől itt eltekintek, és az alábbiakban bizonyítás nélkül közlöm a fontosabb következményeket.

A fázistényezőt jobb híján továbbra is a $\kappa_u = \frac{m}{2^n}$ összefüggés alapján származtatjuk, ahol m a mérés eredménye. Most azonban a kapott κ_u hibával terhelt lesz, amely hiba két részből tevődik össze:

1. Az egyik hibakomponens tisztán klasszikus, kvantálási hiba jellegű. Ez abból adódik, hogy véges számú kvantumbittel csak véges számú különböző fázistényezőt tudunk megkülönböztetni.

2. A másik hibakomponens a kvantumos bizonytalanságból ered: az IQFT kimenetén nem az egyik bázisállapot fog előállni, hanem több bázisállapot szuperpozíciója, tehát a méréseink már nem teljesen determinisztikus eredményt fognak szolgáltatni.

Ha azt szeretnénk, hogy a κ_u fázistényező véges felbontásból eredő, klasszikus hibája legfeljebb 2^{-c} legyen, a kvantumos bizonytalanság miatti téves mérés valószínűsége pedig legfeljebb P_e, akkor a fázisbecslőnk felső regiszterének legalább

$$n = c - 1 + \left\lceil \log_2 \left(3 + \frac{1}{P_e} \right) \right\rceil$$

kvantumbitből kell állnia, de a mérést elegendő a regiszter alsó $c - 1$ qubitjén végezni. A szükséges bitszám meghatározása minden alkalmazás esetén egyedi, intuitív, mérnöki feladat – egyensúlyt kell találni az elvárt pontosság, valamint a megvalósítás költségei között.

Végezetül nézzük meg, hogy mi történik, ha az alsó, t qubites kvantumregisztert nem a $|u\rangle$ sajátállapottal inicializáljuk (mert mondjuk nem ismerjük), hanem egy tetszőleges $|\varphi\rangle$ állapottal! Mivel az $\mathbf{U}$ operátor unitér, sajátállapotai ortonormált bázist alkotnak, vagyis bármely tetszőleges $|\varphi\rangle$ állapot felírható a sajátállapotok szuperpozíciójaként. Ebből a linearitás miatt az következik, hogy ilyenkor a kimeneten az egyes sajátértékeket kapjuk meg a megfelelő valószínűségi eloszlás szerint. Ezt kihasználja a 4.3. pontban bemutatott Shor-algoritmus, de például a rendezetlen adatbázisokban történő keresésre szolgáló Grover-algoritmus is.

4. Az RSA-kód feltörése kvantumszámítógéppel

Az RSA-algoritmus napjaink egyik legelterjedtebb nyilvános kulcsú kriptográfiai eljárása, amelyet 1977-ben publikált Ron Rivest, Adi Shamir és Leonard Adleman. Számításelméleti szempontból az algoritmus érdekességét az adja, hogy a mai napig nem bizonyított, hogy biztonságos. Mindazonáltal jelenlegi ismereteink alapján úgy tűnik, a feltörés klasszikus módszerekkel olyan lassan menne, hogy meg sem érdemes próbálni. Látni fogjuk azonban, hogy kvantumszámítógéppel nincs elvi akadálya az RSA-titkosítás gyors visszafejtésének.

4.1. Az RSA-algoritmus

Az RSA-eljárás nyilvános kulcsú titkosítás, amely azt jelenti, hogy az üzenet titkosításához használt kulcs publikus, ellenben a kódolt üzenet visszafejtéséhez használt kulcs természetesen titkos. Nézzük meg, milyen lépéseket kell tennünk, ha ily módon titkosított üzenetekkel történő kommunikációt szeretnénk implementálni!

1. Választunk két különböző, nagy prímszámot, legyenek ezek p és q.
2. Képezzük az $N = p \cdot q$ szorzatot.
3. A szorzatra kiértékeljük az Euler-függvényt[6]: $\varphi(N) = (p-1)(q-1)$, majd választunk egy kicsi, páratlan a számot, melyre $\text{lnko}\{\varphi(N), a\} = 1$.
4. Kiszámítjuk az a szám modulo $\varphi(N)$ értelemben vett multiplikatív inverzét, jelöljük ezt b-vel: $a \cdot b \equiv 1 \pmod{\varphi(N)}$.
5. A nyilvános kulcs $K = \{a, N\}$, a titkos kulcs $L = \{b, N\}$.
6. A K kulcsot nyilvánosságra hozzuk, így aki az m üzenetet titkosítva szeretné nekünk elküldeni, az a $c = m^a \mod N$ kifejezés kiértékelésével megteheti.
7. Miután megkaptuk a c kódszöveget, az $m = c^b \mod N$ kifejezés kiértékelésével állíthatjuk belőle vissza az eredeti m üzenetet[7].

A kódolásra és a dekódolásra léteznek hatékony implementációk, de az algoritmus számításigénye ezekkel sem csekély. A gyakorlatban ezért sokszor előfordul, hogy maga az érdemi kommunikációs folyamat egy gyorsabb, szimmetrikus kulcsú algoritmussal titkorítva zajlik, és csupán az előzetes kulcsmegosztáshoz használnak RSA-t.

[6] Az Euler-féle φ-függvény egy adott pozitív egész számhoz a nála nem nagyobb relatív prímek számát adja meg.

[7] A dekódolási formula mögött az Euler–Fermat-tétel van. A visszafejtés helyességének bizonyítása elolvasható az [5] irodalomban.

Nézzünk egy számpéldát az algoritmus működésére! Természetesen a példában jóval kisebb számok szerepelnek, mint a gyakorlatban használatos RSA-kódolásokban. Összehasonlításképpen: napjainkban a prímszámokat úgy érdemes megválasztani, hogy az N szorzat legalább 2048 bites legyen, ez 617 decimális számjegynek felel meg.

1. Legyen a két prímszám $p = 53$ és $q = 47$.
2. A szorzatuk $N = 53 \cdot 47 = 2491$.
3. Az Euler-függvény $\varphi(2491) = 52 \cdot 46 = 2392$. Válasszuk ehhez az $a = 9$ relatív prímet!
4. A 9 modulo 2392 értelemben vett multiplikatív inverze $b = 1329$, mert $9 \cdot 1329 = 11961 = 5 \cdot 2392 + 1$.
5. A nyilvános kulcs $K = \{9; 2491\}$, a titkos kulcs $L = \{1329; 2491\}$.
6. Tegyük fel, hogy valaki az $m = 623$ üzenetet szeretné nekünk titkosítva elküldeni! Ekkor a $c = 623^9 \mod 2491 = 2288$ kódszöveget állítja elő a nyilvános kulcsunk ismeretében.
7. Végül a kódot visszafejtjük a titkos kulcsunkkal: $m = 2288^{1329} \mod 2491 = 623$.

4.2. Feltörés klasszikusan – rendkeresés

Az előző példa alapján az algoritmus nem tűnik túlságosan biztonságosnak. A nyilvános kulcsunk az $a = 9$ és $N = 2491$ számokból áll – aki rájön arra, hogy az N szám az 53 és 47 prímek szorzata, az a 4. pontban részletezett számítás végrehajtása után már meg is kapta a titkos kulcsunkat, amelynek birtokában a nekünk szánt kódolt üzeneteket vissza tudja fejteni.

Az algoritmus működőképességét azonban az biztosítja, hogy a gyakorlatban használt, többszáz jegyű kulcsok prímtényezőinek meghatározása reménytelenül lassan menne. Az N szám osztóinak megkeresése ugyanis FNP osztályú probléma: egyelőre nem találtak rá polinomidejű[8] algoritmust, és úgy sejtjük, hogy nem is létezik – bár ez nincs bizonyítva.

A prímtényezős felbontás meghatározására a legtriviálisabb módszer az, hogy a szám négyzetgyökénél kisebb összes prímszámmal megpróbáljuk elosztani. Ily módon megközelítőleg $4 \cdot 10^{134}$ évig tartana egy 1024 bites RSA-kulcs feltörése (összehasonlításképpen: az ősrobbanás óta mostanáig eltelt idő kb. $1{,}38 \cdot 10^{10}$ év). Ennél számos jobb algoritmus létezik, a leggyorsabbal ez az idő kb. 11,3 év lenne. Az alábbiakban egy olyan algoritmust mutatok be, amely ugyan a klasszikus módszerek közül messze nem a leggyorsabb, de a kvantumos megvalósítás szempontjából ennek lesz nagy jelentősége.

[8] A számításelméletben polinomidejűnek nevezünk egy algoritmust, ha lépésszáma a bemenet méretével legfeljebb polinomiálisan – tehát nem például exponenciálisan – növekszik.

Legyenek az x és N pozitív egész számok relatív prímek és legyen $x < N$. Ekkor az x szám *modulo N értelemben vett rendjének* nevezzük azt a legkisebb pozitív egész r számot, amelyre $x^r \equiv 1 \pmod{N}$. A következőkben egy számpéldán keresztül illusztrálom, hogy a rendkeresés felhasználható az RSA-kulcsok feltörésére.

1. Legyen a felbontandó szám az előző példában használt $N = 2491$.

2. Választunk egy 1 és $N - 1$ közötti egész számot, legyen ez most $x = 189$.

3. Megkeressük x modulo N értelemben vett rendjét. Ha a rend páratlanra adódik, akkor új x-et sorsolunk.

$$\begin{aligned} 189^1 &\equiv 189 \pmod{2491} \\ 189^2 &\equiv 847 \pmod{2491} \\ 189^3 &\equiv 659 \pmod{2491} \\ 189^4 &\equiv 1 \pmod{2491} \end{aligned}$$

4. Tehát x rendje $r = 4$. Képezzük az $y = x^{\frac{r}{2}} = 189^2 = 35721$ számot.

5. Kiszámítjuk $y - 1$ és $y + 1$ osztási maradékát N-nel. Ha valamelyik nulla, akkor új x-et sorsolunk.

$$k_- = (y - 1) \mod N = 35720 \mod 2491 = 846$$

$$k_+ = (y + 1) \mod N = 35722 \mod 2491 = 848$$

6. Képezzük az így kapott számok legnagyobb közös osztóját N-nel, ezek lesznek a keresett prímtényezők.

$$p = \text{lnko}\{k_+, N\} = \text{lnko}\{848; 2491\} = 53$$

$$q = \text{lnko}\{k_-, N\} = \text{lnko}\{846; 2491\} = 47$$

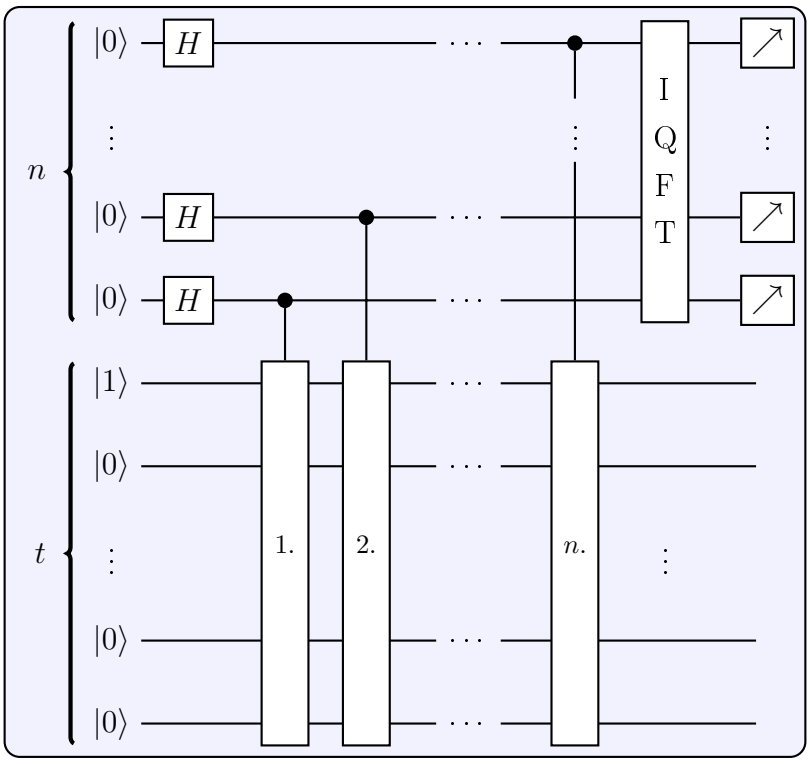
Valóban, $53 \cdot 47 = 2491$, tehát visszakaptuk a kiindulási prímszámokat, amelyek birtokában képezhető a titkos kulcs is. A legnagyobb közös osztó számítására létezik polinomidejű módszer (például az egyszerű euklideszi algoritmus is ilyen), így ha a rendkeresést gyorsan el tudnánk végezni, akkor a prímtényezőket is gyorsan megkapnánk[9]. A rendkeresés problémája klasszikus algoritmussal nem oldható meg polinomidőben, de amint a következő pontban látni fogjuk, kvantumalgoritmussal már igen.

[9]Megjegyzem, hogy az RSA-kód feltörésére más – szintén rendkeresésen alapuló – módszerek is léteznek. Köztük olyanok is, amelyek még a két prím meghatározását sem igénylik. Léteznek továbbá az informatikai rendszerek sebezhetőségeire építő feltörési kísérletek is, ezek tárgyalása azonban nem célja ennek a dolgozatnak.

4.3. Feltörés kvantumosan – Shor-algoritmus

Peter Shor, a Bell Laboratories kutatója – ma az MIT matematikaprofesszora – 1994-ben jött rá arra, hogy a rendkeresés klasszikusan FNP osztályú problémája kvantumszámítógép segítségével a BQP *(Bounded Error Quantum Polynomial Time)* problémaosztályba emelhető át, vagyis létezik olyan legfeljebb polinomiális komplexitású algoritmus, amely minden lehetséges bemenetre megoldást talál $\frac{1}{3}$-nál nem nagyobb hibavalószínűséggel.

Számos kvantumalgoritmus működése alapul a párhuzamosíthatóság elvén: a megoldandó feladatot egyszerre kiértékeljük az összes lehetséges bemenetre, majd a kimenetek közül annak a valószínűségi amplitúdóját erősítjük fel, amelyik a megoldást szolgáltatta. A Shor-algoritmus esetében ez azt jelenti, hogy kiszámítjuk $x^k \mod N$ értékét az összes lehetséges $k < N$ pozitív egész számra, és ezeket eltároljuk egy $t = \lceil \log_2 N \rceil$ qubites regiszterben. Egy másik regiszterben a k szám összes lehetséges értékét tároljuk, majd ezek közül annak az amplitúdóját erősítjük fel, amelyre $x^k \mod N = 1$. A 6. ábrán látható speciális fázisbecslő hálózat éppen ezt a működést valósítja meg.



6. ábra. *A Shor-algoritmus megvalósítása*

A felső regiszter szerepét könnyű azonosítani: a Hadamard-kapuk a klasszikus $|0\rangle$ állapotból a $|+\rangle$ egyenletes szuperpozíciót hozzák létre, tehát itt történik meg az összes lehetséges k érték előállítása. Az alsó sorban lévő számozott kapuk úgy működnek, hogy a felső regiszter l-edik vezetékével vezérelt kapu az $x^{2^{n-l}} \mod N$ műveletet hajtja végre, ha a vezérlőbemenet $|1\rangle$. Tehát az első az x értéket állítja elő, a második az x^2-et, és így tovább, modulo N értelemben. Számelméletből ismeretes, hogy a moduláris hatványozás

binárisan az alábbi módon végezhető:

$$x^k \mod N = \prod_{l=1}^{n} \left(x^{k_l 2^{n-l}} \mod N \right) =$$

$$= \left(x^{k_1 2^{n-1}} \mod N \right) \left(x^{k_2 2^{n-2}} \mod N \right) \ldots \left(x^{k_n 2^0} \mod N \right),$$

ahol $k_l \in \{0;1\}$ a k szám l-edik bitje, azaz $k = \overline{k_1 k_2 \ldots k_n} = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0$. Észrevehetjük, hogy a szorzat egyes tényezői éppen az imént definiált kapuknak felelnek meg, vagyis az alsó sor kapui együttesen az $x^k \mod N$ műveletet végzik el. Az áramkör tehát egy olyan fázisbecslőnek (ld. az 5. ábrát) felel meg, amelyben az $\mathbf{U}$ operátor egy x-szel való szorzást hajt végre modulo N értelemben:

$$\mathbf{U} : |q\rangle \to |(qx) \mod N\rangle$$

Megmutatható[10], hogy minden $b = 1, \ldots, r-1$ esetén a

$$|u_b\rangle = \sum_{s=0}^{r-1} \frac{e^{-j2\pi \frac{b}{r} s}}{\sqrt{r}} |x^s \mod N\rangle$$

állapot sajátvektora ennek az $\mathbf{U}$ operátornak, és a hozzá tartozó fázistényező $\kappa_b = \frac{b}{r}$. Mivel a keresett r rend megjelenik a fázistényezőben, a rendkeresés fázisbecsléssel történhet.

Felmerül azonban a kérdés, hogy milyen kezdeti állapottal inicializáljuk az alsó regisztert. Korábban láttuk, hogy a legjobb megoldás az volna, ha valamelyik sajátvektort tennénk ezekre a vezetékekre – azonban a sajátvektorok kifejezésében szerepel az r rend, amelyet éppen meg szeretnénk határozni. Nézzük meg, hogy mivel egyenlő a sajátvektorok egyenletes szuperpozíciója!

$$\sum_{b=0}^{r-1} \frac{1}{\sqrt{r}} |u_b\rangle = \sum_{b=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \frac{e^{-j2\pi \frac{b}{r} s}}{\sqrt{r}} |x^s \mod N\rangle =$$

$$\frac{1}{r} \sum_{s=0}^{r-1} \left(\sum_{b=0}^{r-1} e^{-j2\pi \frac{b}{r} s} \right) |x^s \mod N\rangle = |x^0 \mod N\rangle = |1\rangle$$

Azt kaptuk tehát, hogy a fázisbecslő alsó regiszterét a t qubites $|1\rangle$ állapottal inicializálhatjuk, ez látható a 6. ábrán is. Ekkor az m_b mérési eredményből $\kappa_b \approx \frac{m_b}{2^n}$ módon számíthatjuk a fázistényezőt, azt azonban nem tudjuk, hogy ez milyen b-hez tartozik, mivel b az egyenletes szuperpozícióval történő inicializálás miatt egyenletes valószínűségi eloszlást követ az $\{1; 2; \ldots; r-1\}$ halmazon. A következőkben azt nézzük meg, hogyan kapható meg mégis a rend a mérési eredményből.

[10] A bizonyítás elolvasható az [1] irodalomban.

Az $\frac{m_b}{2^n}$ szám *reguláris lánctört alakjának* nevezzük az

$$\frac{m_b}{2^n} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{q_4 + \dots}}}$$

emeletes törtet, a

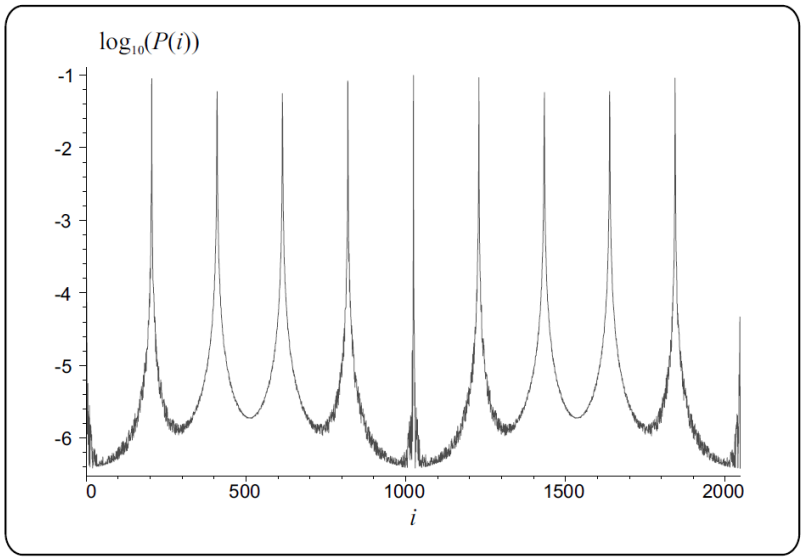
$$\zeta_1 = q_1, \quad \zeta_2 = q_1 + \frac{1}{q_2}, \quad \zeta_3 = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3}}, \quad \dots, \quad \zeta_l = \frac{m_b}{2^n}$$

számok pedig $\frac{m_b}{2^n}$ *lánctörtes közelítései*. A q_1, q_2, ... számok a lánctört *jegyei*. Igazolható, hogy amennyiben

$$\left| \frac{b}{r} - \frac{m_b}{2^n} \right| \leq \frac{1}{2r^2}$$

fennáll, akkor $\frac{b}{r}$ megegyezik $\frac{m_b}{2^n}$ valamelyik lánctörtes közelítésével. Ez a feltétel kielégíthető, ha a felső regisztert $n = \lceil \log_2(N^2) + \log_2(3 + P_e^{-1}) \rceil$ qubitesre választjuk. Nézzük meg az alábbi – az [1] irodalomból kölcsönzött – példán keresztül, hogyan használhatók fel ezen ismeretek a rend megkeresésére!

Tegyük fel, hogy egy kvantumszámítógépen megvalósítottuk a Shor-algoritmust $n = 11$ qubit szélességű felső regiszterrel! Az $N = 33$ kulcsot szeretnénk faktorizálni úgy, hogy az $x = 5$ szám rendjét keressük. A 7. ábrán látható, hogy az IQFT kimenetén milyen mérési eredményeket milyen valószínűséggel kaphatunk meg.



7. ábra. *Kimeneti statisztika*

A csúcsok helyei: $\{0; 205; 410; 614; 819; 1024; 1229; 1434; 1638; 1843\}$, ezek felelnek meg az egyes κ_b értékeknek. Megközelítőleg 78 % annak a valószínűsége, hogy egy mérés során meg is kapjuk az egyik értéket ezek közül. Tegyük fel, hogy a mérésünk eredménye $m_b = 614$. A $\frac{614}{2^{11}}$ számot lánctörtbe fejtve az alábbi jegyeket kapjuk:

$$q_1 = 0, \quad q_2 = 3, \quad q_3 = 2, \quad q_4 = 1, \quad q_5 = 50, \quad q_6 = 2.$$

Ezek alapján pedig a közelítések:

$$\zeta_1 = 0, \quad \zeta_2 = \frac{1}{3}, \quad \zeta_3 = \frac{2}{7}, \quad \zeta_4 = \frac{3}{10}, \quad \zeta_5 = \frac{152}{507}, \quad \zeta_6 = \frac{307}{1024} = \frac{m_b}{2^n}.$$

Ezek közül választjuk ki azt, amelyik a legközelebb van $\frac{614}{2048}$-hoz, de a nevezője még kisebb, mint N, ez $\zeta_4 = \frac{3}{10}$. Tehát x rendjére $r = 10$ adódott. Valóban, $5^{10} = 33 \cdot 295928 + 1$. Ebből N prímtényezői:

$$\begin{array}{rcccccc} p & = & \text{lnko}\{33;\ (5^5+1) \bmod 33\} & = & \text{lnko}\{33;\ 24\} & = & 3 \\ q & = & \text{lnko}\{33;\ (5^5-1) \bmod 33\} & = & \text{lnko}\{33;\ 22\} & = & 11 \end{array}$$

A fentiekben bemutatott módszerrel a korábban példaként hozott 1024 bites kulcsok elvben 10 ms alatt feltörhetőek. Sőt, egy tízezer bites kulcs faktorizálása sem tartana tovább egyetlen másodpercnél. Sajnos – illetve adatbiztonsági szempontból talán szerencsére – a gyakorlati megvalósítás itt még nem tart. 2001-ben az IBM kutatói egy hét qubites, mágneses magrezonancián alapuló kvantumáramkörrel sikeresen implementálták a Shor-algoritmust a $15 = 3 \cdot 5$ faktorizálásra, 2012-ben pedig a University of Bristol professzorainak a $21 = 3 \cdot 7$ felbontást is sikerült megvalósítaniuk egy optikai elvű kvantumáramkörrel – mindkét eredményről részletesen olvashatunk a [7] és [8] hivatkozásoban. Ezzel szemben klasszikus módszerekkel már 768 bites (232 decimális jegyű) RSA-kulcsot is sikerült feltörni. Fontos különbség azonban, hogy amíg a klasszikus feltörés hatékonyságának elvi, matematikai korlátai vannak, addig a kvantumos feltörés esetében ezek a korlátok csak gyakorlati, technikai jellegűek. Elképzelhető tehát, hogy megfelelő mérnöki megoldások alkalmazásával a közeljövőben rendkívül gyorsan feltörhetővé válnak a ma még biztonságosnak tekintett kulcsok is.

Irodalomjegyzék

[1] Sándor Imre, Ferenc Balázs. *Quantum Computing and Communications*. John Wiley & Sons Ltd., Chichester, 2005.

[2] Ivanyos Gábor. *Kvantumszámítógépes algoritmusok*. Debreceni Egyetem, 2011.

[3] Michael Loceff. *A Course in Quantum Computing*. Foothill College, 2015.

[4] Katona Gyula Y., Recski András, Szabó Csaba. *A számítástudomány alapjai*. Typotex, Budapest, 2006.

[5] Rónyai Lajos, Ivanyos Gábor, Szabó Réka. *Algoritmusok*. Typotex, Budapest, 2008.

[6] Juan Bermejo Vega. *Classical simulations of non-abelian quantum Fourier transforms*. Technische Universität München, 2011.

[7] Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, Isaac L. Chuang. *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*. Nature, 2001. vol. 414., 883-887. oldal, ISSN: 0028-0836.

[8] Enrique Martin-Lopez, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou, Jeremy L. O'Brien. *Experimental realization of Shor's quantum factoring algorithm using qubit recycling*. Nature Photonics, 2012. vol. 6., 773-776. oldal, ISSN: 1749-4885.

Rövidítések jegyzéke

DFT	Discrete Fourier Transform
IDFT	Inverse Discrete Fourier Transform
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
QFT	Quantum Fourier Transform
IQFT	Inverse Quantum Fourier Transform
LSB	Least Significant Bit
MSB	Most Significant Bit
QPE	Quantum Phase Estimation
RSA	Rivest–Shamir–Adleman-algoritmus
FNP	Function Nondeterministic Polynomial
BQP	Bounded Error Quantum Polynomial