SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 835

PROGRAMSKO OSTVARENJE SHOROVOG ALGORITMA U SIMULATORU KVANTNOG RAČUNALA

Velimir Kovačić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 835

PROGRAMSKO OSTVARENJE SHOROVOG ALGORITMA U SIMULATORU KVANTNOG RAČUNALA

Velimir Kovačić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 835

Pristupnik:	**Velimir Kovačić (0036533917)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Marin Golub

Zadatak: **Programsko ostvarenje Shorovog algoritma u simulatoru kvantnog računala**

Opis zadatka:

Opisati kvantna računala i način na koji se provodi kvantno računanje. Navesti na koji način kvantna računala predstavljaju problem za klasične kriptografske sustave. Opisti Shorov algoritam za rastavljanje brojeva na proste faktore. Ostvariti Shorov algoritam u programskom jeziku Phython koristeći slobodno dostupne knjižnice programa. Nadalje, ostvariti isti algoritam u programskom jeziku OpenQASM 2.0 te ga ispitati na simulatoru kvantnog računala. Razmotriti kako pripremljeni algoritam za izvođenje na simulatoru ostvariti na stvarnom slobodno dostupnom kvantnom računalu. Ispitati rad algoritma na proizvoljnom skupu kriptiranih poruka te analizirati dobivene rezultate.

Rok za predaju rada: 9. lipnja 2023.

SADRŽAJ

1. Uvod

Područje kvantnog računarstva vrlo je aktualno i vijesti o naprecima konstantno dolaze iz cijelog svijeta. Neki od najbitnijih u zadnje 2 godine su:

- 8. lipnja 2021. - U Japanu postignuta kvantna komunikacija putem optičkih vlakana na udaljenosti od 600 km[4]
- 8. listopada 2021. - U Kini predstavljeno novo 66-qubitno programabilno supravodljivo kvantno računalo[3]
- 16. studenog 2021. - Tvrtka IBM stvorila 127-qubitni kvantni procesor „IBM Eagle"[2]
- 22. lipnja 2022. - Prvi ikad ugrađeni kvantni strujni krug je predstavljen[6]
- 9. studenog 2022. - Tvrtka IBM predstavila je 433-qubitni kvantni procesor „Osprey"[5]

Kvantna računala pri računanju iskorištavaju pojave iz kvantne mehanike. Kao jedinicu informacije za razliku od klasičnog bita, koriste kvantni bit ili kraće qubit. Dok klasični bit može poprimiti vrijednosti 0 i 1, kvantni se bit prije mjerenja nalazi u stanju koje se može opisati kao kvantna superpozicija stanja 0 i 1, a tek u trenutku mjerenja poprima vrijednost 0 ili 1.

Klasično računalo s n bitova može prikazati brojeve od 0 do 2^n, a kvantno računalo s jednakim brojem qubitova može prikazati brojeve od 0 do 2^n istovremeno i nad svim brojevima zajedno izvršavati operacije. Dakle, jasno je da kvantna računala imaju prednost pred klasičnim u računarskim problemima koji se mogu učinkovitije rješavati uvođenjem paralelizma.

Kako bi se iskoristila ova poželjna svojstva kvantnih računala, razvijaju se kvantni algoritmi. Jedan od kvantnih algoritama od posebnog značaja je Shorov algoritam jer učinkovito rješava problem faktorizacije brojeva s velikim prostim faktorima. Problem na temelju kojeg je osigurana sigurnost jednog od najčešće korištenih asimetričnih kriptosustava: RSA. Srodni kvantni algoritmi predstavljaju opasnost i za kriptosustave utemeljene na eliptičkim krivuljama i diskretnim logaritmima.

Trenutno, kvantno računalo s dovoljnim brojem qubitova za probijanje programskih ostvarenja RSA, koja se koriste u praksi, ne postoji. No, to nije nikakva prepreka u teorijskom razmatranju Shorovog algoritma i njegovoj primjeni na manje brojeve, što je upravo tema ovog rada.

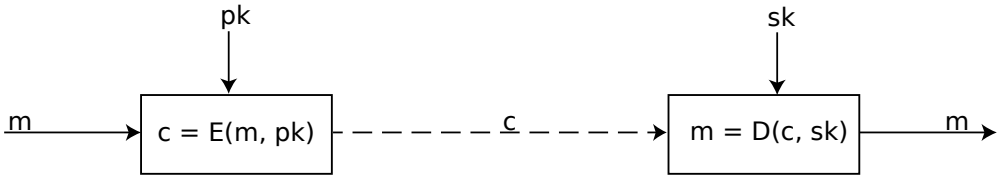
Razmotrit će se princip kriptosustava RSA, kvantno računanje, Shorov algoritam i njegovi dijelovi. Predstavit će se samostalno programsko ostvarenje Shorovog algoritma u simulatoru kvantnog računala koristeći slobodno dostupne knjižnice programa. Dodatno, bit će dane upute za pokretanje algoritma na stvarnom slobodno dostupnom kvantnom računalu.

2. Kriptosustav RSA

2.1. Asimetrična kriptografija

Asimetrična kriptografija proces je kriptiranja i dekriptiranja poruka parom ključeva. Javnim ključem pk (engl. *public key*) poruka m se kriptira, a privatni ključem sk (engl. *secret key*) dekriptira se.

Prednost koju asimetrična kriptografija ima nad simetričnom kriptografijom je izostanak potrebe za razmjenom jedinstvenog ključa za postupak kriptiranja i dekriptiranja. Sigurna razmjena simetričnog ključa je problematična.



Slika 2.1: Kriptiranje i dekriptiranje poruke

Asimetrična kriptografija ima 3 osnovna algoritma:

1. Generiranje ključeva: $G() = pk, sk$
2. Postupak kriptiranja $E(m, pk) = c$
3. Postupak dekriptiranja $D(c, sk) = m$

2.2. Matematička pozadina

2.2.1. Najveći zajednički djelitelj

Ako za prirodne brojeve a, k, d vrijedi jednakost $a = kd$, onda se kaže da je a djeljiv s d ili da d dijeli a, označava se $d|a$.

Prema osnovnom teoremu aritmetike, svaki prirodni broj ima jedinstven rastav na proste faktore. Neka je a prirodni broj, p_i i-ti prosti faktor u rastavu broja a, a α_i pripadna potencija:

$$a = \prod_i p_i^{\alpha_i} \tag{2.1}$$

Najveći zajednički djelitelj prirodnih brojeva a i b je najveći prirodni broj d, takav da vrijedi $d|a$ i $d|b$. Može se računati kao:

$$d = \mathrm{nzd}(a, b) = \prod_i p_i^{\min\{\alpha_i, \beta_i\}} \tag{2.2}$$

Ako dva broja nemaju zajedničkih djelitelja osim broja 1, to jest $\mathrm{nzd}(a, b) = 1$, kažemo da su a i b relativno prosti.

Drugi način računanja najvećeg zajedničkog djelitelja je Euklidovim algoritmom koji, u svom rekurzivnom obliku, glasi:

Algorithm 1 Euklidov algoritam

```
1: function NZD(a, b)
2:     if b == 0 then
3:         return a
4:     else
5:         return NZD(b, a%b)
```

Ponekad je korisno saznati cijele brojeve x i y takve da za prirodne brojeve a i b vrijedi:

$$ax + by = \mathrm{nzd}(a, b) \tag{2.3}$$

To se može postići proširenim Euklidovim algoritmom:

Algorithm 2 Prošireni Euklidov algoritam

```
1: function NZD_PROSIREN(a, b)
2:     (x, y, d, u, v, w) := (1, 0, a, 0, 1, b)
3:     while w > 0 do
4:         q := ⌊d/w⌋
5:         x := u
6:         y := v
7:         d := w
8:         u := x − qu
9:         v := y − qv
10:        w := d − qw
11:    return (x, y, d)
```

2.2.2. Kongruencije

Ako prirodni broj m dijeli razliku cijelih brojeva $a-b$, onda se kaže da je a kongruentan b modulo m, piše se:

$$a \equiv b \pmod{m} \tag{2.4}$$

Kongruencija a i b je relacija ekvivalencije, stoga vrijede svojstva refleksivnosti, simetričnosti i tranzitivnosti. Uz to, vrijede i sljedeća svojstva:

1. Ako $a \equiv b \pmod{m}$ i $c \equiv d \pmod{m}$, onda:

   $a + c \equiv b + d \pmod{m}$

   $a - c \equiv b - d \pmod{m}$

   $ac \equiv bd \pmod{m}$

2. Ako $a \equiv b \pmod{m}$ i $d|m$, onda:

   $a \equiv b \pmod{d}$

3. Ako $a \equiv b \pmod{m}$ i $c \neq 0$, onda:

   $ac \equiv bc \pmod{mc}$

4. Ako i samo ako $ax \equiv ay \pmod{m}$, onda:

   $x \equiv y \pmod{\frac{m}{\text{nzd}(a,m)}}$

Multiplikativni inverz modulo m broja a je broj a^{-1} za koji vrijedi:

$$a^{-1}a \equiv 1 \pmod{m} \tag{2.5}$$

Ako su a i m relativno prosti, onda se taj broj može pronaći proširenim Euklidovim algoritmom.

$$ax + my = \text{nzd}(a, m) \tag{2.6}$$
$$ax + my = 1 \tag{2.7}$$
$$ax = 1 + m(-y) \tag{2.8}$$
$$ax \equiv 1 \pmod{m} \tag{2.9}$$
$$x = a^{-1} \tag{2.10}$$

2.2.3. Reducirani sustav ostataka modulo m

Reducirani sustav ostataka modulo m je skup prirodnih brojeva relativno prostih s m, koji nisu međusobno kongruentni modulo m. Formalno:

$$\{r_i : r_i, r_j \in \mathbb{N} \wedge \text{nzd}(r_i, m) = 1 \wedge (i \neq j \wedge r_i \not\equiv r_j \pmod{m})\} \tag{2.11}$$

Broj elemenata u reduciranom sustavu ostataka modulo m označava se Eulerovim brojem φ.

$$\varphi(m) = \prod_i p_i^{\alpha_i}(1 - \frac{1}{p_i}) \tag{2.12}$$

Za umnožak relativno prostih brojeva, Eulerov broj φ je multiplikativna funkcija.

$$\varphi(pq) = \varphi(p)\varphi(q) \tag{2.13}$$

Za proste brojeve, Eulerov broj φ glasi:

$$\varphi(p) = p - 1 \tag{2.14}$$

Eulerov teorem govori da ako su a i m relativno prosti, onda vrijedi:

$$a^{\varphi(m)} \equiv 1 \pmod{m} \tag{2.15}$$

2.3. Funkcije kriptosustava

2.3.1. Generiranje ključeva

Odaberu se 2 nasumična velika prosta broja p i q, njihov umnožak je broj N. Zatim se odabere nasumičan broj e iz reduciranog sustava ostataka modulo $\varphi(N)$. Izračuna se d, multiplikativni inverz od e modulo $\varphi(N)$, proširenim Euklidovim algoritmom. Javni ključ pk je par brojeva (e, N), a privatni ključ sk je par brojeva (d, N). Iznimno je važno koristiti kriptografski siguran generator slučajnih brojeva i predati mu *seed* s visokom entropijom.

Algorithm 3 RSA generiranje ključeva

function G(seed)
    gnb := generatorSlučajnihVelikihProstihBrojeva(seed)
    p := gnb.iduci()
    q := gnb.iduci()
    $N := p * q$
    $\varphi(N) := (p-1) * (q-1)$
    rso := reduciraniSustavOstataka($\varphi(N)$)
    e := nasumičniOdabir(rso)
    d := NZD_PROSIREN($e, \varphi(N)$)[0]
    $pk = (e, N)$
    $sk = (d, N)$
    return (pk, sk)

2.3.2. Postupak kriptiranja

Postupak kriptiranja vrši se modularnim potenciranjem poruke m potencijom e modulo N. Pritom kako postupak kriptiranja ne bi bio podložan napadu odabranim izvornim tekstom, poruka m prvo se nadopunjava funkcijom Pad(). Za funkciju Pad() najčešće se koristi shema OEAP (engl. *Optimal Asymmetric Encryption Padding*.)

Algorithm 4 RSA postupak kriptiranja

function E(m, pk)
    $(e, N) := pk$
    m_p := Pad(m)
    $c := m_p^e \pmod N$
    return c

2.3.3. Postupak dekriptiranja

Postupak dekriptiranja obavlja se modularnim potenciranjem šifrata c potencijom d modulo N. Ako je korišteno nadopunjavanje funkcijom Pad(), mora postojati inverzna funkcija Unpad().

Algorithm 5 RSA postupak dekriptiranja

1: **function** D(m, sk)
2: $(d, N) := sk$
3: $m_p := c^d \pmod N$
4: $m := \text{Unpad}(m_p)$
5: **return** m

Postupak dekriptiranja vrijedi jer:

$$D(E(m, pk), sk) = \text{Unpad}((\text{Pad}(m)^e)^d) \pmod N \tag{2.16}$$
$$= \text{Unpad}(\text{Pad}(m)^{ed}) \pmod N \tag{2.17}$$
$$= \text{Unpad}(\text{Pad}(m)^{1+k\varphi(N)}) \pmod N \tag{2.18}$$
$$= \text{Unpad}(\text{Pad}(m)^1 \cdot \text{Pad}(m)^{k\varphi(N)}) \pmod N \tag{2.19}$$
$$= \text{Unpad}(\text{Pad}(m) \cdot 1) \pmod N \tag{2.20}$$
$$= m \pmod N \tag{2.21}$$

2.4. Sigurnost kriptosustava RSA

Sigurnost kriptosustava RSA temeljena je na tome da ne postoji učinkovit klasični algoritam za faktorizaciju brojeva s velikim prostim faktorima. Ključna je riječ „klasični", Shorov algoritam kvantni je algoritam za faktorizaciju brojeva s velikim prostim faktorima.

Napreci u kvantnom računarstvu mogli bi omogućiti praktičnu primjenu Shorovog algoritma za probijanje podataka zaštićenih postupkom kriptiranja RSA. Štoviše, danas se skupljaju velike količine podataka kriptiranih pomoću RSA kako bi se u budućnosti mogli dekriptirati. Protiv te prakse ne može se boriti povećanjem ključeva sukladno povećanju moći kvantnih računala.

3. Kvantna računala

3.1. Notacija bra-ket

Poznata je i kao Diracova notacija po svom stvoritelju Paulu Diracu. Sastoji se od 2 vrste objekata, objekta **bra** $\langle x|$ koji predstavlja vektor redak ili linearnu mapu i objekta **ket** $|y\rangle$ koji predstavlja vektor stupac.

$$\langle x| = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \tag{3.1}$$

$$|y\rangle = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{3.2}$$

Objekti bra i ket mogu se kombinirati na 2 načina, u objekt braket $\langle x|y\rangle$ koji predstavlja skalarni umnožak i objekt ket-bra $|y\rangle \langle x|$ koji predstavlja matricu dobivenu matričnim množenjem vektor stupca i vektor retka.

$$\langle x|y\rangle = x_1y_1 + x_2y_2 + \cdots + x_ny_n \tag{3.3}$$

$$|y\rangle \langle x| = \begin{pmatrix} x_1y_1 & x_2y_1 & \cdots & x_ny_1 \\ x_1y_2 & x_2y_2 & \cdots & x_ny_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1y_n & x_2y_n & \cdots & x_ny_n \end{pmatrix} \tag{3.4}$$

3.2. Hilbertov prostor

Hilbertov prostor vektorski je prostor nad poljem realnih ili kompleksnih brojeva, proširen operacijom skalarnog umnoška. U nastavku će se razmatrati Hilbertov prostor

nad poljem kompleksnih brojeva $\mathbb{C}$, a vektore i skalarne umnoške pisat će se u notaciji bra-ket. N-dimenzionalni Hilbertov prostor označava se simbolom $\mathcal{H}^{(N)}$. Za operaciju skalarnog umnoška vrijedi:

1. Simetrija uz konjugaciju:
   $\langle x|y\rangle = \langle y|x\rangle^*$
2. Linearnost prvog elementa:
   $\langle ax_1 + bx_2|y\rangle = a\langle x_1|y\rangle + b\langle x_2|y\rangle \qquad a, b \in \mathbb{C}$
3. Skalarni umnožak vektora samim sobom je pozitivno definitan:
   $\langle x|x\rangle \geq 0$

Norma vektora računa se kao:

$$\|x\| = \sqrt{\langle x|x\rangle} \tag{3.5}$$

Radi održavanja svojstava skalarnog umnoška Hilbertovog prostora, pri pretvorbi iz bra u ket i obratno potrebno je uz transponiranje provesti i kompleksnu konjugaciju. Transponiranje uz kompleksno konjugiranje zove se hermitska konjugacija i označava se simbolom † (engl. *dagger*).

$$|x\rangle = \langle x|^\dagger = (\langle x|^\intercal)^* \tag{3.6}$$

$$\langle x| = |x\rangle^\dagger = (|x\rangle^\intercal)^* \tag{3.7}$$

3.3. Stanje jednog qubita

Stanje qubita prikazuje se vektorom stanja $|\phi\rangle$, 2-dimenzionalnim vektorom u Hilbertovom prostoru $\mathcal{H}^{(2)}$ nad poljem kompleksnih brojeva $\mathbb{C}$. 2 su dimenzije radi analogije s klasičnim bitom.

$$|\phi\rangle = \begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix} \qquad \phi_1, \phi_2 \in \mathbb{C} \tag{3.8}$$

Vektor stanja mora biti normiran:

$$\|\phi\| = \sqrt{\langle\phi|\phi\rangle} = \sqrt{\phi_1^2 + \phi_2^2} = 1 \tag{3.9}$$

Bazna stanja u $\mathcal{H}^{(2)}$ koja odgovaraju stanjima klasičnog bita (0 i 1) su:

$$|0\rangle = \begin{bmatrix}1\\ 0\end{bmatrix} \tag{3.10}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.11}$$

Svako stanje može se zapisati kao superpozicija baznih stanja:

$$|\phi\rangle = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \phi_1 |0\rangle + \phi_2 |1\rangle \tag{3.12}$$

Vektor stanja može se zapisati u trigonometrijskom obliku, koji osigurava uvjet normiranosti:

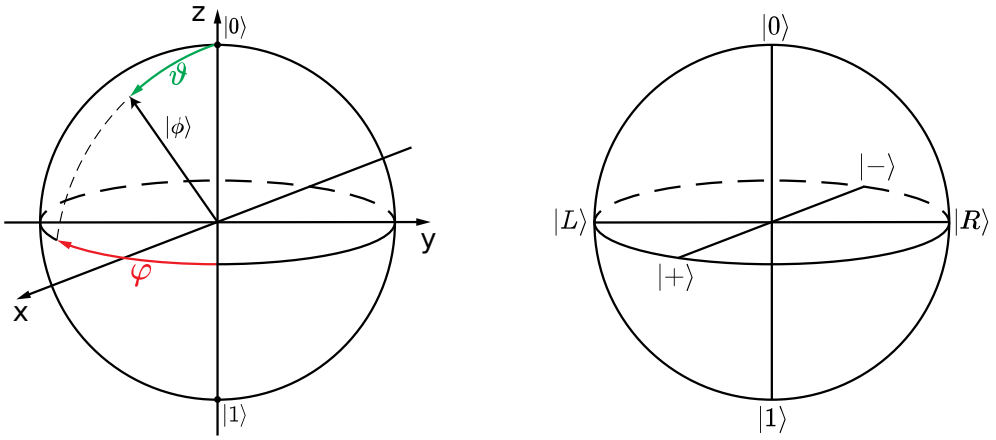
$$|\phi\rangle = e^{\frac{-i\varphi}{2}} \cos\frac{\vartheta}{2} |0\rangle + e^{\frac{i\varphi}{2}} \sin\frac{\vartheta}{2} |1\rangle \tag{3.13}$$

Gdje su φ i ϑ kutevi za koje vrijedi:

$$0 \leq \varphi < 2\pi \tag{3.14}$$

$$0 \leq \vartheta \leq \pi \tag{3.15}$$

Vektor stanja u ovom zapisu može se prikazati na Blochovoj sferi. Baza $|0\rangle$, $|1\rangle$ određena je točkama u kojima os z siječe sferu.



(a) Prikaz vektora $|\psi\rangle$ **(b)** Prikaz baza

Slika 3.1: Blochova sfera

Ponekad je umjesto baze $|0\rangle$, $|1\rangle$ pogodnije koristiti bazu $|+\rangle$, $|-\rangle$, određenu točkama u kojima os x siječe sferu.

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{3.16}$$

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{3.17}$$

Radi potpunosti spomenut će se i baza $|R\rangle$, $|L\rangle$, određena točkama u kojima os y siječe sferu, no ona se rijetko koristi.

$$|R\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) \tag{3.18}$$

$$|L\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle) \tag{3.19}$$

3.4. Stanje više qubitova

Kvantno računalo s n qubitova ima $N = 2^n$ baznih stanja, stanja koja bi se mogla prikazati na klasičnom računalu s jednakim brojem klasičnih bitova.

Stanje kvantnog računala s n qubitova prikazuje se vektorom stanja $|\phi\rangle$, N-dimenzionalnim vektorom u Hilbertovom prostoru $\mathcal{H}^{(N)}$ nad poljem kompleksnih brojeva $\mathbb{C}$. Vizualizacija vektora stanja u ovom prostoru otežana je jer ga nije moguće prikazati na Blochovoj sferi.

Stanje sustava s n qubitova računa se kao tenzorski umnožak pojedinih stanja qubitova:

$$|\phi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{bmatrix} \tag{3.20}$$

Bazna stanja, u nastavku, naizmjence će se označavati dekadskim i binarnim zapisom. Na primjer, na računalu s 3 qubita stanje je:

$$|3_{(10)}\rangle = |011_{(2)}\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.21}$$

3.4.1. Spregnutost i separabilnost

Stanje koje se može napisati kao tenzorski umnožak baznih stanja naziva se separabilno. Stanje $(|01_{(2)}\rangle + |11_{(2)}\rangle)$ je separabilno jer:

$$|01_{(2)}\rangle + |11_{(2)}\rangle = (|0\rangle + |1\rangle) \otimes |1\rangle \tag{3.22}$$

Stanje koje se ne može napisati kao tenzorski umnožak baznih stanja nazivamo spregnutim. Stanje $|00_{(2)}\rangle + |11_{(2)}\rangle$ je spregnuto.

U 4-dimenzionalnom Hilbertovom prostoru (2 qubita) postoji uvjet separabilnosti:

$$\alpha\,|00_{(2)}\rangle + \beta\,|01_{(2)}\rangle + \gamma\,|10_{(2)}\rangle + \delta\,|11_{(2)}\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \iff \alpha\delta = \beta\gamma \tag{3.23}$$

Uočava se da su separabilna stanja mali podskup svih stanja u pripadnom Hilbertovom prostoru.

3.5. Vjerojatnost očitanja

Vjerojatnost da se kvantni sustav s vektorom stanja $|\phi\rangle$ očita (izmjeri) u stanju $|\psi\rangle$ jednak je kvadratu modula amplitude vjerojatnosti.

$$p(\phi \to \psi) = |a(\phi \to \psi)|^2 \tag{3.24}$$

Amplituda vjerojatnosti očitanja sustava s vektorom stanja $|\phi\rangle$ u stanju $|\psi\rangle$ jednaka je:

$$a(\phi \to \psi) = \langle\psi|\phi\rangle \tag{3.25}$$

Primjer

Vjerojatnost očitanja stanja $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ u stanju $|0\rangle$.

$$p(|+\rangle \to |0\rangle) = |a(|+\rangle \to |0\rangle)|^2 \tag{3.26}$$

$$= |\,\langle 0|+\rangle\,|^2 \tag{3.27}$$

$$= |\frac{1}{\sqrt{2}}|^2 \tag{3.28}$$

$$= \frac{1}{2} \tag{3.29}$$

Kao što se moglo pretpostaviti, vjerojatnost da se stanje $|+\rangle$, koje je superpozicija stanja $|0\rangle$ i $|1\rangle$ u jednakom udjelu, očita u stanju $|0\rangle$ je 50%.

3.6. Kvantna logička vrata

3.6.1. Unitarni operator

Evolucija kvantnog stanja, pa tako i stanja kvantnog računala mora biti unitarna (reverzibilna), stoga se definira unitarni (invertibilni) operator U. Inverz U^{-1} unitarnog operatora jednak je transponiranom i kompleksno konjugiranom U.

$$U^{-1} = U^{\dagger} \tag{3.30}$$

3.6.2. Hermitski operator

Hermitski operator ostaje nepromijenjen transponiranjem i kompleksnim konjugiranjem.

$$U = U^{\dagger} \tag{3.31}$$

Ako je operator unitaran i hermitski onda je sam sebi inverz. Iz jednadžbi 3.30 i 3.31 dobiva se:

$$U = U^{-1} \tag{3.32}$$

3.6.3. Operator X

Klasična logička vrata poput AND i OR nisu reverzibilna, odnosno, na temelju njihovog izlaza nije moguće jednoznačno odrediti njihov ulaz. Klasična logička vrata NOT jesu reverzibilna i mogu se koristiti u kvantom logičkom krugu. Unitarni operator X predstavlja kvantna logička vrata NOT.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.33}$$

Utjecaj unitarnog operatora X na bazna stanja $|0\rangle$ i $|1\rangle$ je očekivan:

$$X\,|0\rangle = |1\rangle \tag{3.34}$$

$$X\,|1\rangle = |0\rangle \tag{3.35}$$

Unitarni operator X ujedno je i hermitski, stoga je sam sebi inverz.

$$X = X^{-1} \tag{3.36}$$

3.6.4. Operator H

Unitarni operator H predstavlja Hadamardova vrata.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.37}$$

Hadamardova vrata prebacuju bazna stanja $|0\rangle$ i $|1\rangle$ u bazna stanja $|+\rangle$ i $|-\rangle$ i obratno.

$$H|0\rangle = |+\rangle \tag{3.38}$$
$$H|1\rangle = |-\rangle \tag{3.39}$$
$$H|+\rangle = |0\rangle \tag{3.40}$$
$$H|-\rangle = |1\rangle \tag{3.41}$$

Unitarni operator H ujedno je i hermitski, stoga je sam sebi inverz.

$$H = H^{-1} \tag{3.42}$$

3.6.5. Operator R

Unitarni operator R operator je faznog pomaka, provodi rotaciju oko osi z Blochove sfere.

$$R[\varphi] = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \tag{3.43}$$

Stanja baze $|0\rangle$, $|1\rangle$ ne mijenja se jer ona sama leže na osi z. Vektor stanja $|0\rangle$ ostaje nepromijenjen, a na vektor stanja $|1\rangle$ primjenjuje fazni pomak.

$$R[\varphi]|0\rangle = |0\rangle \tag{3.44}$$
$$R[\varphi]|1\rangle = e^{i\varphi}|1\rangle \tag{3.45}$$

Važno je uočiti da fazni pomak vektora stanja ne mijenja stanje:

$$p(e^{i\varphi}|\phi\rangle \to e^{i\vartheta}|\phi\rangle) = |a(e^{i\varphi}|\phi\rangle \to e^{i\vartheta}|\phi\rangle)|^2 \tag{3.46}$$
$$= |(e^{i\vartheta}|\phi\rangle)^\dagger (e^{i\varphi}|\phi\rangle)|^2 \tag{3.47}$$
$$= |(e^{-i\vartheta}\langle\phi|)(e^{i\varphi}|\phi\rangle)|^2 \tag{3.48}$$
$$= |e^{-i\vartheta}e^{i\varphi}\langle\phi|\phi\rangle|^2 \tag{3.49}$$
$$= |e^{i(\varphi-\vartheta)}|^2 \tag{3.50}$$
$$= 1 \tag{3.51}$$

Vektore baze $|+\rangle$, $|-\rangle$ rotira oko ekvatora Blochove sfere za kut φ. Rotacija za kut π sama je sebi inverzna, a rotacija za kut 2π je operator identiteta I.

$$R[\pi] = R[\pi]^{-1} \tag{3.52}$$

$$R[2\pi] = I \tag{3.53}$$

3.6.6. Operator SWAP

Operator SWAP služi za zamjenu stanja 2 qubita.

$$\mathrm{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.54}$$

Djeluje očekivano:

$$\mathrm{SWAP}\,|00_2\rangle = |00_2\rangle \tag{3.55}$$

$$\mathrm{SWAP}\,|01_2\rangle = |10_2\rangle \tag{3.56}$$

$$\mathrm{SWAP}\,|10_2\rangle = |01_2\rangle \tag{3.57}$$

$$\mathrm{SWAP}\,|11_2\rangle = |11_2\rangle \tag{3.58}$$

Unitarni operator SWAP ujedno je i hermitski, stoga je sam sebi inverz.

$$\mathrm{SWAP} = \mathrm{SWAP}^{-1} \tag{3.59}$$

3.6.7. Upravljani operatori

Upravljani (kontrolirani) operatori su operatori čije izvođenje ovisi o stanju upravljačkog qubita. Takvi operatori označavaju se prefiksom „c“. Svaki unitarni operator može biti upravljan.

U slučaju sustava s 2 qubita, upravljani operator, čiji je upravljački qubit prvi, a upravljani qubit drugi, može se zapisati u sljedećem blok-matričnom zapisu:

$$cU = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} \tag{3.60}$$

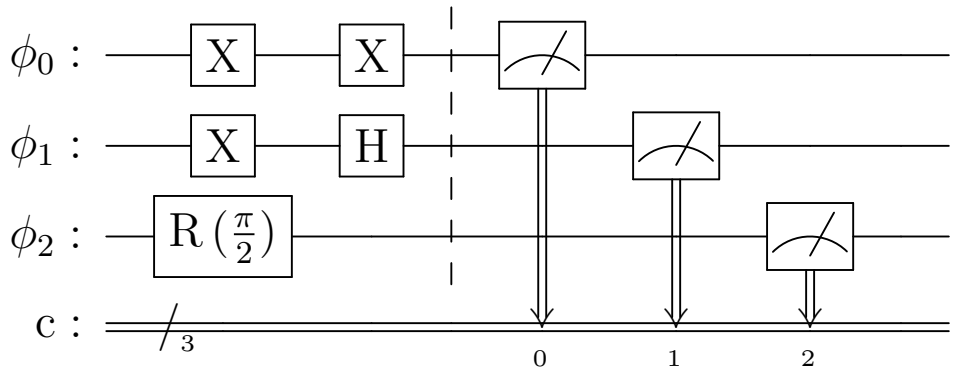
Primjer

Upravljani operator X:

$$cX = \begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.61}$$

3.7. Kvantni logički krug

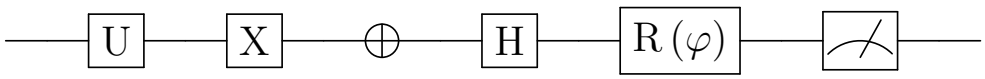
Kvantni logički krug sličan je klasičnom. S lijeve strane početna su stanja pojedinih qubitova, a linije predstavljaju tijek njihove evolucije. Uz qubitove na lijevoj strani nalaze se i klasični bitovi. Operatori se postavljaju na linije i djeluju na odgovarajuće qubitove. Na kraju se provodi mjerenje qubitova i izmjerena se stanja zapisuju u klasične bitove.



Slika 3.2: Primjer kvantnog logičkog kruga s 3 qubita

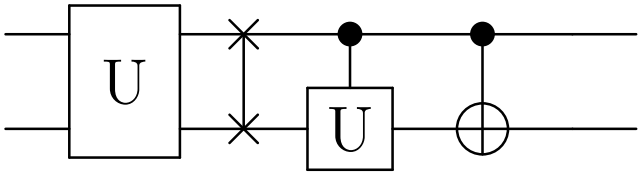
Koristi se konvencija koja nalaže da je najviši kvantni bit $|\phi_0\rangle$ ujedno i najmanje značajni. U ostatku teksta vodit će se tom konvencijom i označavat će ga se s ϕ_n (ako je u sustavu n qubitova.)

3.7.1. Oznake kvantnih logičkih krugova



Slika 3.3: Jednostavni unitarni operatori

Slika 3.3 prikazuje (s lijeva na desno) oznake za: unitarni operator U, operator X (1. prikaz), operator X (2. prikaz), operator H, operator $R[\varphi]$ i oznaku za mjerenje stanja qubita.



Slika 3.4: Unitarni operatori više qubitova

Slika 3.4 prikazuje (s lijeva na desno) oznake za: unitarni operator U, operator SWAP, operator cU i operator cX.

3.7.2. Računanje stanja u kvantnom logičkom krugu

Kao što se i početna stanja qubitova kvantnog logičkog kruga mogu tenzorskim umnoškom kombinirati u jedno stanje, tako se i unitarni operatori u istoj ravnini mogu kombinirati u jedan unitarni operator. Nedostatak operatora na nekoj liniji može se poistovjetiti s djelovanjem operatora identiteta I. To omogućuje simulaciju evolucije stanja kvantnog računala matričnim i tenzorskim množenjem.

U primjeru na slici 3.5 početno stanje glasi:

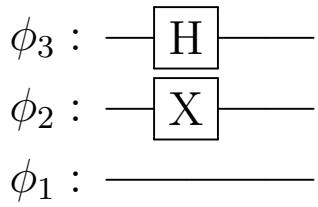
$$|\phi\rangle = |\phi_3\rangle \otimes |\phi_2\rangle \otimes |\phi_1\rangle \tag{3.62}$$

Kombinirani operator U kojim će se djelovati na stanje glasi:

$$U = H \otimes X \otimes I \tag{3.63}$$

Stanje sustava nakon operatora U se računa kao:

$$|\phi'\rangle = H\,|\phi\rangle = (H \otimes X \otimes I)(|\phi_3\rangle \otimes |\phi_2\rangle \otimes |\phi_1\rangle) \tag{3.64}$$



Slika 3.5: Kvantni logički krug s 3 qubita i 2 operatora u ravnini

3.8. Kvantni algoritmi

3.8.1. Kvantna Fourierova transformacija

Kvantna Fourierova transformacija služi za pretvorbu kvantnog stanja iz baze $|0\rangle$, $|1\rangle$ (računalne baze) u bazu $|+\rangle$, $|-\rangle$ (Fourierovu bazu). Označava se s QFT (engl. *Quantum Fourier transform*.)

$$\mathrm{QFT}\,|0\rangle = |+\rangle \tag{3.65}$$

$$\mathrm{QFT}\,|1\rangle = |-\rangle \tag{3.66}$$

Očigledno je da je u slučaju jednog qubita QFT jednak Hadamardovim vratima. U slučaju n bitova ($N = 2^n$ stanja), QFT se računa kao:

$$\mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle \tag{3.67}$$

$$= \frac{1}{\sqrt{N}} (e^0 |0\rangle + e^{\frac{2\pi i x}{N}} |1\rangle + \ldots + e^{\frac{2\pi i x (N-1)}{N}} |N-1\rangle) \tag{3.68}$$

Neka $|x\rangle$ predstavlja stanje u računalnoj bazi, onda $|\tilde{x}\rangle$ predstavlja odgovarajuće stanje u Fourierovoj bazi. Broj koji predstavlja stanje $y_{(10)}$ iz dekadskog brojevnog sustava može se napisati u binarnom kao niz znamenki:

$$y_{(2)} = y_1 y_2 y_3 \ldots y_n \tag{3.69}$$

U dekadskom brojevnom sustavu je:

$$y_{(10)} = \sum_{k=1}^{n} 2^{n-k} y_k \tag{3.70}$$

Jednadžbu QFT-a može se preurediti kako bi se lakše ostvarila u kvantnom logičkom krugu.

$$|\tilde{x}\rangle = \mathrm{QFT}\,|x\rangle \tag{3.71}$$
$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle \tag{3.72}$$
$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x \sum_{k=1}^{n} 2^{n-k} y_k}{N}} |y\rangle \tag{3.73}$$
$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x \sum_{k=1}^{n} 2^{-k} N y_k}{N}} |y\rangle \tag{3.74}$$
$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x \sum_{k=1}^{n} \frac{y_k}{2^k}} |y\rangle \tag{3.75}$$
$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{\frac{2\pi i x y_k}{2^k}} |y_1 y_2 y_3 \ldots y_n\rangle \tag{3.76}$$
$$= \frac{1}{\sqrt{N}} \sum_{y_1=0}^{1} \sum_{y_2=0}^{1} \ldots \sum_{y_n=0}^{1} \prod_{k=1}^{n} e^{\frac{2\pi i x y_k}{2^k}} |y_1 y_2 y_3 \ldots y_n\rangle \tag{3.77}$$
$$= \frac{1}{\sqrt{N}} (|0\rangle + e^{\frac{2\pi i x}{2^1}} |1\rangle) \otimes (|0\rangle + e^{\frac{2\pi i x}{2^2}} |1\rangle) \otimes \ldots \otimes (|0\rangle + e^{\frac{2\pi i x}{2^n}} |1\rangle) \tag{3.78}$$

Dakle, ako je u binarnom zapisu u računalnoj bazi $|x\rangle = |x_1 x_2 \ldots x_n\rangle = |x_n\rangle \otimes \ldots \otimes |x_2\rangle \otimes |x_1\rangle$, onda za k-ti qubit u Fourierovoj bazi vrijedi:

$$|\tilde{x}_k\rangle = (|0\rangle + e^{\frac{2\pi i x}{2^k}} |1\rangle) \tag{3.79}$$

Hadamardovim operatorom qubit iz stanja računalne baze može se dovesti u stanje Fourierove baze:

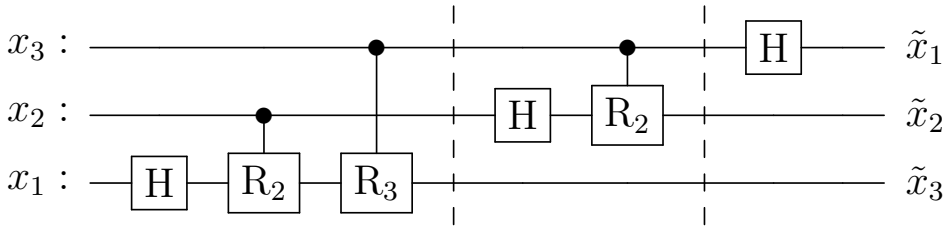
$$H\,|x\rangle = (|0\rangle + e^{\pi i x} |1\rangle) = (|0\rangle + e^{\frac{2\pi i x}{2}} |1\rangle) \tag{3.80}$$

Kako bi se moglo upravljati potencijom nazivnika eksponenta, uvodi se unitarni operator R_k:

$$R_k\,|0\rangle = 0 \tag{3.81}$$
$$R_k\,|1\rangle = e^{\frac{2\pi i}{2^k}} |1\rangle \tag{3.82}$$

R_k nije ništa drugo nego operator faznog pomaka $R[\varphi]$.

$$R_k = R\left[\frac{2\pi}{2^k}\right] \tag{3.83}$$



Slika 3.6: Kvantni logički krug za QFT s 3 qubita

Važno je uočiti da će na izlazu iz QFT-a qubitovi $|\tilde{x}\rangle$-a biti u obrnutom redoslijedu u odnosu na ulaz, to se može ispraviti primjenom operatora SWAP.

Na primjeru s $n = 3$ qubita pokazat će se da je krug ispravan, analogno će vrijediti za veći broj qubitova.

$$
\begin{aligned}
|\tilde{x}\rangle &= \mathrm{QFT}\,|x\rangle && (3.84)\\
&= \mathrm{QFT}\,|x_1x_2x_3\rangle && (3.85)\\
&= (H\,|x_3\rangle) \otimes (R_2(x_3)H\,|x_2\rangle) \otimes (R_3(x_3)R_2(x_2)H\,|x_1\rangle) && (3.86)\\
&= \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x_3}{2}}\,|x_3\rangle)\\
&\quad \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x_3}{2^2}} e^{\frac{2\pi i x_2}{2}}\,|1\rangle)\\
&\quad \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{2\pi i x_3}{2^3}} e^{\frac{2\pi i x_2}{2^2}} e^{\frac{2\pi i x_1}{2}}\,|1\rangle) && (3.87)\\
&= (|0\rangle + e^{\frac{2\pi i x}{2^1}}\,|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi i x}{2^2}}\,|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi i x}{2^3}}\,|1\rangle) && (3.88)\\
&= |\tilde{x}_1\rangle \otimes |\tilde{x}_2\rangle \otimes |\tilde{x}_3\rangle && (3.89)\\
&= |\tilde{x}_3\tilde{x}_2\tilde{x}_1\rangle && (3.90)
\end{aligned}
$$

3.8.2. Kvantna procjena faznog pomaka

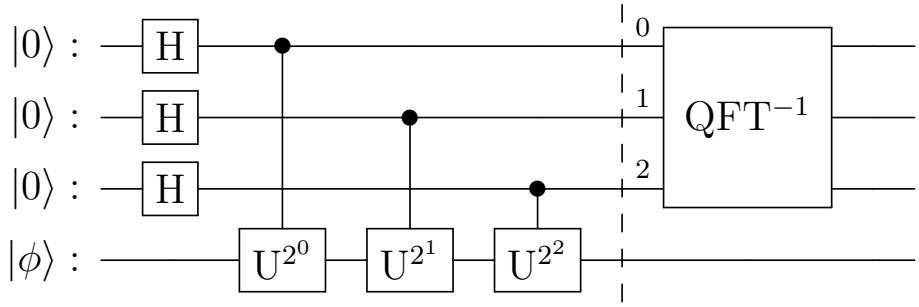
Neka je $|\phi\rangle$ vlastiti vektor unitarnog operatora U. Vrijedi:

$$U\,|\phi\rangle = \lambda\,|\phi\rangle \tag{3.91}$$

$$\lambda = e^{2\pi i\vartheta} \tag{3.92}$$

$e^{2\pi i\vartheta}$ je vlastita vrijednost, a ϑ pripadni fazni pomak. Kvantna procjena faznog pomaka QPE (engl. *Quantum phase estimation*) omogućava dobivanje vrijednosti faznog pomaka ϑ, a posljedično i vlastitu vrijednost operatora.

Kvantni logički krug za procjenu faznog pomaka izgleda ovako:



Slika 3.7: QPE s 3 qubita preciznosti

Uočava se pojava inverzne kvantne Fourierove transformacije. Sustav na slici 3.7 ima $n = 3$ qubita preciznosti. Poželjan je što veći broj n jer se njegovim povećanjem ujedno povećava i preciznost procjene faznog pomaka. Kvantni logički krug gradi se analogno za veće brojeve n. Neposredno prije primjene operatora inverzne kvantne Fourierove transformacije, stanje cijelog sustava $|\psi\rangle$ je:

$$\begin{aligned} |\psi\rangle &= \sqrt{\frac{1}{2^3}}(|0\rangle + e^{2\pi i 2^0 \vartheta}) \\ &\otimes (|0\rangle + e^{2\pi i 2^1 \vartheta}) \\ &\otimes (|0\rangle + e^{2\pi i 2^2 \vartheta}) \\ &\otimes |\phi\rangle \qquad (3.93) \\ &= |\tilde{\vartheta}\rangle \otimes |\phi\rangle \qquad (3.94) \end{aligned}$$

To je u korelaciji s kvantnom Fourierovom transformacijom jer bi transformiranje stanja $|\vartheta\rangle$ koje odgovara binarnom zapisu broja $2^n\vartheta$, gdje je $n = 3$, dalo:

$$\begin{aligned} |\tilde{\vartheta}\rangle &= \mathrm{QFT}\,|\vartheta\rangle \qquad (3.95) \\ &= \sqrt{\frac{1}{2^3}}(|0\rangle + e^{\frac{2\pi i 2^3 \vartheta}{2^1}}\,|1\rangle) \\ &\otimes (|0\rangle + e^{\frac{2\pi i 2^3 \vartheta}{2^2}}\,|1\rangle) \\ &\otimes (|0\rangle + e^{\frac{2\pi i 2^3 \vartheta}{2^3}}\,|1\rangle) \qquad (3.96) \end{aligned}$$

Što je ustvari isto stanje. Qubitovi su u obrnutom redoslijedu, no to se može ispraviti operatorima SWAP. Dakle, primjenom inverzne kvantne Fourierove transformacije može se iz stanja $|\psi\rangle$ dobiti stanje $|\vartheta\rangle \otimes |\phi\rangle$. Mjerenjem odgovarajućih qubitova dobiva se binarni broj blizak $2^n\vartheta$, koji se onda dijeli s 2^n da bi se dobila procjena faznog pomaka ϑ.

4. Shorov algoritam

Neka su p i q veliki prosti brojevi, a njihov umnožak je $N = pq$. Postavlja se pitanje, ako je poznat N, mogu li se na učinkovit način dobiti prosti faktori p i q?

Moguće ih je dobiti grubom silom. Pretražuje se skup prostih brojeva većih od 1, a manjih ili jednakih $\sqrt{N}$, ako za neki x iz tog skupa vrijedi $x|N$, onda je $p = x$, a $q = \frac{N}{x}$. To je rješenje, ali nije učinkovito.

Shorov algoritam omogućuje pronalaženje tih prostih faktora u polilogaritamskom vremenu uporabom kvantnog računala. Algoritam se sastoji od 2 dijela:

1. Klasični dio: pretvara problem faktorizacije broja u problem pronalaženja perioda funkcije

2. Kvantni dio: rješava problem pronalaženja perioda funkcije

4.1. Klasični dio algoritma

Na ulazu u klasični dio algoritma je broj N, a izlaz su prosti faktori p i q.

1. N mora biti neparan i ne smije biti potencija prostog broja. Ako je N paran, rješenje je $p = 2$ i $q = \frac{N}{2}$. Ako je N potencija prostog broja, također je pronađeno rješenje.

2. Slučajnim odabirom bira se broj a iz skupa $\{2, 3, \ldots N-1\}$, takav da je relativno prost s N. Ako nije relativno prost s N, rješenje je $p = \mathrm{nzd}(a, N)$ i $q = \frac{N}{\mathrm{nzd}(a,N)}$.

3. Stvara se funkcija modularnog potenciranja: $f(x) = a^x \pmod N$.

4. Funkcija $f(x)$ je periodična s periodom r. Vrijedi:

$$f(0) = 1 \qquad f(r) = 1 \qquad r \in \{2, 3, \ldots \varphi(N)\} \tag{4.1}$$

$$f(r) = a^r \pmod N \tag{4.2}$$
$$a^r \equiv 1 \pmod N \tag{4.3}$$
$$a^r - 1 \equiv 0 \pmod N \tag{4.4}$$
$$a^r - 1 = \lambda N \tag{4.5}$$
$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = \lambda N \tag{4.6}$$

5. Računa se period r funkcije kvantnim dijelom algoritma. Ako je period r neparan, vraća se na korak 2.

6. Računaju se prosti faktori p i q. Zbog 4.6 vrijedi:

$$N|(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \tag{4.7}$$

Poželjno je da N dijeli umnožak $(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)$, a da ne dijeli $(a^{\frac{r}{2}} - 1)$ i $(a^{\frac{r}{2}} + 1)$ zasebno. To znači da ni jedan od njih nije višekratnik broja N, nego svaki sadrži po jedan od njegova 2 prosta faktora. N sigurno ne dijeli $(a^{\frac{r}{2}} - 1)$ jer bi u suprotnom vrijedilo $a^{\frac{r}{2}} \equiv 1 \pmod N$ što je u kontradikciji s time da je period jednak r. Preostaje provjeriti za $(a^{\frac{r}{2}} + 1)$. Ako N dijeli $(a^{\frac{r}{2}} + 1)$, vraća se na korak 2. Inače, vrijedi:

$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = \lambda N \tag{4.8}$$
$$(a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) = \lambda_p \lambda_q pq \tag{4.9}$$

$$(a^{\frac{r}{2}} - 1) = \lambda_p p \tag{4.10}$$
$$(a^{\frac{r}{2}} + 1) = \lambda_q q \tag{4.11}$$

Proste faktore p i q mogu se izračunati na sljedeći način:

$$\mathrm{nzd}(a^{\frac{r}{2}} - 1, N) = \mathrm{nzd}(\lambda_p p, pq) = p \tag{4.12}$$
$$\mathrm{nzd}(a^{\frac{r}{2}} + 1, N) = \mathrm{nzd}(\lambda_q q, pq) = q \tag{4.13}$$

4.2. Kvantni dio algoritma

Ulaz u kvantni dio algoritma su parametri funkcije modularnog potenciranja a i N, a izlaz je period r te funkcije.

Pronalaženje perioda funkcije svodi se na kvantnu procjenu faznog pomaka. Procjenjuje se fazni pomak vlastite vrijednosti unitarnog operatora $U_{g_{a,N}}$ koji ostvaruje

funkciju modularnog množenja $g_{a,N}(x) = ax \pmod N$. Modularno potenciranje ostvaruje se kombinacijom operatora modularnog množenja i označavat će se kao unitarni operator $U_{f_{a,N}}$. $U_{f_{a,N}}$ ostvaruje funkciju $f_{a,N}(x) = a^x \pmod N$. Unitarni operator modularnog množenja definiran je kao:

$$U_{g_{a,N}} |x\rangle = \begin{cases} |ax \pmod N\rangle, & 0 \leq x < N \\ |x\rangle, & N \leq x < 2^n \end{cases} \tag{4.14}$$

n označava broj qubita na ulazu operatora, a 2^n je broj mogućih stanja koje sustav s n qubitova postiže. Stanja veća ili jednaka N preslikava kao operator identiteta. Ta stanja nisu relevantna jer neće nikada biti dovedena na ulaz operatora, ali je i za njih potrebno definirati preslikavanje radi unitarnosti operatora.

Operator $U_{f_{a,N}}$ stvara se na sljedeći način. Poznato je da za broj x u dekadskom sustavu vrijedi sljedeća pretvorba u binarni:

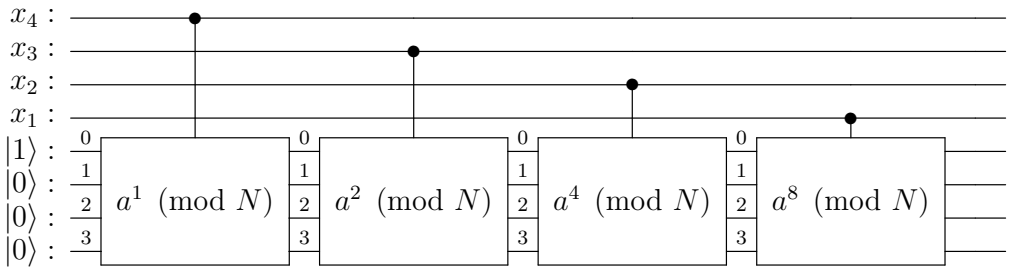
$$x_{(10)} = [x_1 x_2 \dots x_n]_{(2)} = 2^{n-1}x_1 + \dots + 2^1 x_2 + 2^0 x_n \tag{4.15}$$

Dakle funkcija $f_{a,N}(x) = a^x \pmod N$ može se napisati kao:

$$f_{a,N}(x) = a^{2^{n-1}x_1 + \dots + 2^1 x_2 + 2^0 x_n} \pmod N \tag{4.16}$$

$$= a^{2^{n-1}x_1} \dots a^{2^1 x_2} a^{2^0 x_n} \pmod N \tag{4.17}$$

$$= a^{2^{n-1}x_1} \dots a^{2^1 x_2} a^{2^0 x_n} \pmod N \tag{4.18}$$



Slika 4.1: Primjer modularnog potenciranja ostvarenog operatorima modularnog množenja za $n = 4$

Kvantni logički kruga koji ostvaruje operator $U_{f_{a,N}}$ imat će na ulazu stanje $|x\rangle \otimes |1\rangle$, a na izlazu stanje $|x\rangle \otimes |a^x \pmod N\rangle$. U primjeru na slici 4.1 vidi se da svaki qubit od $|x\rangle$ upravlja jednim od operatora modularnog množenja, što u konačnici odgovara izrazu 4.18.

Odabire se broj registara n, najmanji broj takav da $2^n \geq N$. Takav izbor n omogućava da se u binarnom zapisu predstave svi brojevi do N. Neka se kvantni registar w, čije je stanje $|w\rangle$ sastoji od n qubita. Neka se kvantni registar x, čije je stanje $|x\rangle$ sastoji od $2n$ qubita, što se ispostavilo dovoljnim za procjenu faznog pomaka. Neka je c klasični registar u kojeg se zapisuje očitano stanje registra x na kraju izvođenja kvantnog logičkog kruga.

Stanja $|x\rangle$ i $|w\rangle$ u početku su jednaka $|0\rangle$.

$$|\phi\rangle = |x\rangle \otimes |w\rangle = |0\rangle \tag{4.19}$$

U prvom se koraku nad stanjem $|x\rangle$ primjeni niz Hadamardovih vrata čime stanje $|x\rangle$ postaje superpozicija svih baznih stanja. Stanje $|w\rangle$ pretvori se NOT vratima u stanje $|1\rangle$. Stanje $|1\rangle$ zapravo je superpozicija prvih r rezultata funkcije modularnog potenciranja (cijeli ciklus).

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{i=0}^{r-1} |a^i \pmod N\rangle \tag{4.20}$$

Očigledno je da je $|1\rangle$ vlastiti vektor operatora modularnog potenciranja $U_{f_{a,N}}$ jer potenciranje broja potencijom 1 ne mijenja broj.

Radi preglednosti, neka je $|a\rangle \otimes |b\rangle = |a\rangle\,|b\rangle$. Rezultat prvog koraka je stanje sustava:

$$|\phi'\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle + |2\rangle + \ldots + |2^{2n-1}\rangle) \otimes |1\rangle \tag{4.21}$$

$$= \frac{1}{\sqrt{2^n}}(|0\rangle\,|1\rangle + |1\rangle\,|1\rangle + |2\rangle\,|1\rangle + \ldots + |2^{2n-1}\rangle\,|1\rangle) \tag{4.22}$$

Nakon prolaska kroz operator modularnog potenciranja $U_{f_{a,N}}$ stanje je:

$$|\phi''\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle \otimes |a^0 \pmod N\rangle \tag{4.23}$$

$$+ |1\rangle \otimes |a^1 \pmod N\rangle \tag{4.24}$$

$$+ |2\rangle \otimes |a^2 \pmod N\rangle \tag{4.25}$$

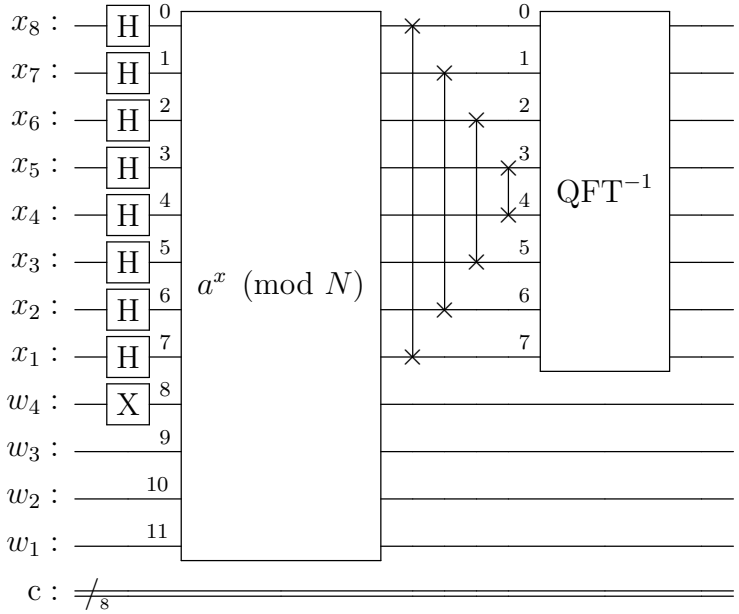
$$\vdots \tag{4.26}$$

$$+ |2^{2n-1}\rangle \otimes |a^{2^{2n-1}} \pmod N\rangle) \tag{4.27}$$

Prije ulaska u inverznu kvantnu Fourierovu transformaciju, potrebno je operatorima SWAP zamijeniti redoslijed qubitova registra x. Ovisno o programskom ostvarenju operatora $\mathrm{QFT}^\dagger$-a moguće je zamjenu obaviti na izlazu.

Konačni izlaz bit će bazno stanje koje predstavlja broj $2^n\vartheta$. Vrijedi $\vartheta = \frac{\lambda}{r}$. Samo je potrebno naći r i λ koji odgovaraju danom razlomku i vrijednost r je period koji vraća kvantni dio algoritma.

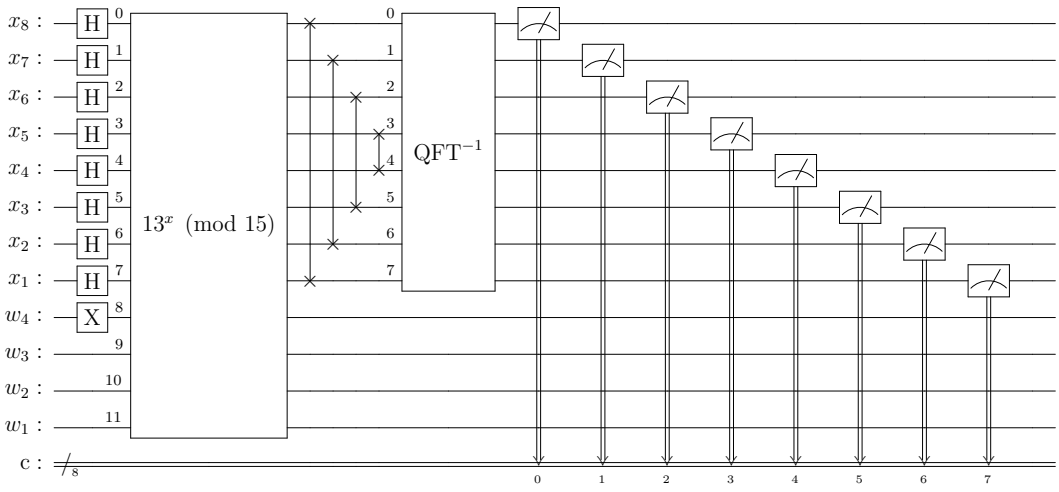
$$|\phi'''\rangle = |2^n\vartheta\rangle = |2^n\frac{\lambda}{r}\rangle \tag{4.28}$$



Slika 4.2: Kvantni logički krug Shorovog algoritma za $n = 4$ (bez mjerenja stanja na izlazu)

4.3. Primjer provedbe algoritma

Neka je $N = 15$. Očigledno je da su prosti faktori 5 i 3. N nije paran i nije potencija prostog broja. Odabire se $a = 13$, relativno prost s N jer $\mathrm{nzd}(13, 15) = 1$. Funkcija modularnog potenciranja je $f(x) = 13^x \pmod{15}$. Traži se period funkcije. Zatim se odabire $n = 4$ jer je to najmanji broj za koji vrijedi $2^n \geq N$. Veličina kvantnog registra w je n, a kvantnog registra x i klasičnog registra c je $2n$. Stvara se sljedeći kvantni logički krug:



Slika 4.3: Primjer kvantnog logičkog kruga za $a = 13$, $N = 15$

U početku je stanje sustava:

$$|\phi\rangle = |0\rangle \tag{4.29}$$

Nakon primjene Hadamardovih i NOT vrata stanje postaje:

$$|\phi'\rangle = \frac{1}{\sqrt{2^8}} \left(\sum_{i=0}^{2^8-1} |i\rangle \right) \otimes |1\rangle \tag{4.30}$$

$$= \frac{1}{16}(|0\rangle\,|1\rangle + |1\rangle\,|1\rangle + |2\rangle\,|1\rangle + \ldots + |255\rangle\,|1\rangle) \tag{4.31}$$

Nakon primjene operatora modularnog potenciranja, stanje je:

$$|\phi''\rangle = \frac{1}{16}(|0\rangle \otimes |13^0 \pmod{15}\rangle \tag{4.32}$$

$$+ |1\rangle \otimes |13^1 \pmod{15}\rangle \tag{4.33}$$

$$+ |2\rangle \otimes |13^2 \pmod{15}\rangle \tag{4.34}$$

$$\vdots$$

$$+ |255\rangle \otimes |13^{255} \pmod{15}\rangle) \tag{4.35}$$

Izračunaju li se potencije vidi se da je to stanje:

$$|\phi''\rangle = \frac{1}{16}(|0\rangle\,|1\rangle + |1\rangle\,|13\rangle + |2\rangle\,|4\rangle + |3\rangle\,|7\rangle \tag{4.36}$$

$$+ |4\rangle\,|1\rangle + |5\rangle\,|13\rangle + |6\rangle\,|4\rangle + |7\rangle\,|7\rangle \tag{4.37}$$

$$\vdots$$

$$+ |252\rangle\,|1\rangle + |253\rangle\,|13\rangle + |254\rangle\,|4\rangle + |255\rangle\,|7\rangle) \tag{4.38}$$

Radi jednostavnosti i bez smanjenja općenitosti može se pretpostaviti da je izmjeren registar $w = 13$ i time kolapsiran broj baznih stanja u superpoziciji stanja sustava. Stanje je sada:

$$|\phi''\rangle = \frac{1}{8}(|1\rangle + |5\rangle + |9\rangle + \ldots + |253\rangle) \tag{4.39}$$

Preostaje samo primijeniti inverznu kvantnu Fourierovu transformaciju $\mathrm{QFT}^\dagger$:

$$|\phi'''\rangle = \mathrm{QFT}^\dagger |\phi''\rangle \tag{4.40}$$

$$= \frac{1}{128}\sum_{k=0}^{255}(e^{\frac{-2\pi ik}{255}} + e^{\frac{-2\pi i5k}{255}} + e^{\frac{-2\pi i9k}{255}} + \ldots + e^{\frac{-2\pi i253k}{255}})\,|k\rangle \tag{4.41}$$

$$= \frac{1}{128}(64\,|0\rangle - 64i\,|64\rangle - 64\,|128\rangle + 64i\,|192\rangle) \tag{4.42}$$

$$= \frac{1}{2}(|0\rangle - i\,|64\rangle - |128\rangle + i\,|192\rangle) \tag{4.43}$$

S jednakom vjerojatnošću izmjerit će se jedno od stanja $|0\rangle$, $|64\rangle$, $|128\rangle$ i $|192\rangle$. Rezultati za svaki mogu se vidjeti u tablici:

Stanje $\|\phi\rangle$	Fazni pomak ϑ	Razlomak $\frac{\lambda}{r}$	Period r
$\|0\rangle$	0	$\frac{0}{r}$	-
$\|64\rangle$	0.25	$\frac{1}{4}$	4
$\|128\rangle$	0.5	$\frac{1}{2}$	2
$\|192\rangle$	0.75	$\frac{3}{4}$	4

Iz 2 od 4 moguća stanja može se deducirati točan period $r = 4$. Uočava se da 4 može biti nazivnik i ostala 2 razlomka. Potvrda da je doista $r = 4$:

$$13^0 \equiv 1 \pmod{15} \tag{4.44}$$

$$13^1 \equiv 13 \pmod{15} \tag{4.45}$$

$$13^2 \equiv 4 \pmod{15} \tag{4.46}$$

$$13^3 \equiv 7 \pmod{15} \tag{4.47}$$

$$13^4 \equiv 1 \pmod{15} \tag{4.48}$$

Mora se provjeriti dijeli li N broj $a^{\frac{r}{2}} + 1$. Ne dijeli, $15 \nmid 170$. Sada se računa:

$$\mathrm{nzd}(13^{\frac{4}{2}} - 1, 15) = \mathrm{nzd}(168, 15) = 3 \tag{4.49}$$

$$\mathrm{nzd}(13^{\frac{4}{2}} + 1, 15) = \mathrm{nzd}(170, 15) = 5 \tag{4.50}$$

Izračunato je da su 3 i 5 prosti faktori broja 15.

5. Programsko ostvarenje Shorovog algoritma

5.1. Knjižnica programa Qiskit

Qiskit javno je dostupna knjižnica programa otvorenog koda u programskom jeziku Python, koja služi za rad s kvantnim logičkim krugovima. Omogućava izradu vlastitih kvantnih logičkih krugova i kvantnih algoritama, njihovo simuliranje i njihovo pokretanje na udaljenom poslužitelju.

```
from qiskit import *

qreg_q = QuantumRegister(2, 'q')
creg_c = ClassicalRegister(2, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.h(qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
```

Slika 5.1: Primjer koda u Pythonu koji koristi knjižnicu programa Qiskit

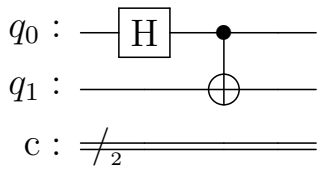
5.2. Programski jezik OpenQASM

OpenQASM imperativni je programski jezik otvorenog koda, koji služi za opisivanje kvantnih logičkih krugova. Koristan je kada je potrebno prenijeti opis kvantnog logičkog kruga. Nedostatak je što je izrada složenijih krugova vremenski zahtjevna jer se svaki operator mora ručno deklarirati.

```
qreg q[2];
creg c[2];

h q[0];
cx q[0], q[1];
```

Slika 5.2: Primjer koda u OpenQASM-u za isti logički krug kao na slici 5.1



Slika 5.3: Kvantni logički krug opisan slikama 5.1 i 5.2

5.3. Struktura programa

Programsko ostvarenje Shorovog algoritma napisano je u programskom jeziku Python (verzija 3) uz korištenje knjižnice programa Qiskit. Sastoji se od 3 datoteke: `shor_klasicni.py`, `shor_kvantni.py` i `main.py`.

Datoteka `main.py` traži korisnički unos broja N za rastav na proste faktore i onda poziva klasični dio Shorovog algoritma koji se nalazi u datoteci `shor_klasicni.py`. Klasični dio Shorovog algoritma izvodi se u `shor_klasicni.py`, a kada je potrebno pronaći period funkcije modularnog potenciranja, poziva se kvantni dio algoritma koji se nalazi u datoteci `shor_kvantni.py`. U kvantnom dijelu algoritma stvara se kvantni logički krug za pronalazak perioda funkcije. Opis kruga simulira se lokalno ili se šalje udaljenom poslužitelju na izvođenje i čekaju se rezultati. Dobiveni se rezultati onda vraćaju klasičnom dijelu algoritma.

5.3.1. Klasični dio algoritma

`shor_klasicni.py` sadrži dvije funkcije.

1. nzd(a, b)

Opis:	Programsko ostvarenje Euklidovog algoritma
Ulaz:	Brojevi a i b
Izlaz:	Najveći zajednički djelitelj od a i b

2. shor_klasicni (N)

Opis:	Programsko ostvarenje klasičnog dijela Shorovog algoritma
Ulaz:	Broj N
Izlaz:	Prosti faktori p i q od N

5.3.2. Kvantni dio algoritma

`shor_kvantni.py` sadrži 7 funkcija:

1. brzo_potenciranje_2 (a, N, e)

Opis:	Brzo računanje izraza $a^{2^e} \pmod N$
Ulaz:	Brojevi a, e i N
Izlaz:	Rezultat operacije $a^{2^e} \pmod N$

2. f(a, N, x)

Opis:	Računanje funkcije modularnog množenja
Ulaz:	Brojevi a, x i N
Izlaz:	Rezultat operacije $ax \pmod N$

3. stavarnje_mod_operatora (a, N, p, e, broj_qbit)

Opis:	Stvaranje unitarnog operatora modularnog množenja
Ulaz:	Brojevi a, p, e i N i broj qubita operatora
Izlaz:	Unitarni operator koji ostvaruje funkciju $f(x) = xa^p \pmod N$

4. dodavanje_potenciranja (a, N, qreg_x, qreg_w, qkrug)

Opis:	Dodavanje unitarnog operatora modularnog potenciranja u kvantni logički krug
Ulaz:	Brojevi a i N, kvantni registri x i w i kvantni logički krug
Izlaz:	Nema izlaza, dodaje unitarni operator ostvaruje funkciju $f(x) = xa^p \pmod N$ u kvantni logički krug

5. dodavanje_qft_dagger(qreg_x, qkrug)

Opis:	Dodavanje inverzne kvantne Fourierove transformacije u kvantni logički krug
Ulaz:	Kvantni registar x i kvantni logički krug
Izlaz:	Nema izlaza, dodaje unitarni operator $\mathrm{QFT}^\dagger$ u kvantni logički krug

6. stvaranje_qkruga(a, N, broj_qbit , k)

Opis:	Stvaranje kvantnog logičkog kruga za procjenu perioda funkcije
Ulaz:	Brojevi a, N i k i broj qubita u registru w
Izlaz:	Kvantni logički krug za procjenu perioda funkcije

7. pronalazak_perioda(a, N)

Opis:	Pokretanje kvantnog logičkog kruga za pronalazak perioda funkcije i analiza rezultata
Ulaz:	Brojevi a i N
Izlaz:	Lista mogućih vrijednosti perioda r

5.4. Detalji programskog ostvarenja

Program uglavnom prati prethodni opis Shorovog algoritma, no razlikuje se po tome da se svaki kvantni logički krug izvršava više puta. Moguće je dobiti pogrešan period iz kvantnog dijela algoritma što bi rezultiralo ponovnim biranjem broja a i izgradnjom

novog kvantnog logičkog kruga. Izgradnja kruga vremenski je zahtjevna, dok višestruko izvođenje već izgrađenog kruga nije. Stoga, kako bi se minimizirala vjerojatnost dobivanja pogrešnog perioda, simulacija kvantnog logičkog kruga izvodi se više puta i vraćaju se svi dobiveni periodi. Provjera svih mogućih perioda otvara mogućnost da se s pogrešnim periodom dobije točno rješenje, što se ponekad događa.

Ovisno o korisnikovim mogućnostima, postoje 2 opcije simuliranja kvantnog logičkog kruga. Lokalno simuliranje u Qiskitovom simulatoru AER i simuliranje na proizvoljnom IBM-ovom udaljenom poslužitelju.

Primarni nedostatak programa je način stvaranja operatora modularnog množenja. Na temelju definicije operatora stvara se matrica operatora. Matrica operatora, uporabom funkcija knjižnice programa Qiskit, automatski se pretvara u niz osnovnih unitarnih operatora. Bez iznimke, taj je niz operatora vrlo dugačak i neoptimiziran, što rezultira vrlo velikim kvantnim logičkim krugovima. Drugi je nedostatak eksponencijalni rast složenosti povećanjem broja qubitova. Za simuliranje 1 qubita više, dvostruko se povećava dimenzija vektora stanja i dimenzije matrica.

U datoteci `shor_kvantni.py` korisnik može namjestiti 4 parametra:

1. `API_KLJUC`

   Korisnik koji ima korisnički račun na platformi IBMQ, može koristiti svoj API ključ kako bi se simulacija kvantnog logičkog kruga izvodila na udaljenom IBM-ovom simulatoru. Ako korisnik ostavi varijablu kao prazan niz znakova `""`, simulacija će se izvoditi lokalno.

2. `SHOTS`

   Koristi se za upravljanjem broja izvođenja simulacije kvantnog logičkog kruga.

3. `K`

   Namješta se preciznost mjerenja faznog pomaka. Registar x je K puta veći od registra w. Što je broj K veći, to je mjerenje preciznije, ali i cjelokupni program sporiji. Preporučeno je postaviti varijablu na najmanje 2.

4. `IBM_BACKEND`

   Služi za odabir IBM-ovog poslužitelja na kojem će se kvantni logički krug simulirati ili izvršavati. Primjerice „`ibm_qasm_simulator`” ili „`ibm_perth`”.

5.5. Upute za pokretanje

Potrebno je na računalu imati instaliranu potporu za programski jezik Python (izrađeno je u verziji 3.11.3) i dodatne knjižnice programa: `qiskit` i `qiskit_ibm_provider`. One se mogu preuzeti iz naredbenog retka:

```
pip install qiskit
pip install qiskit_ibm_provider
```

Za pokretanje samog programa, potrebno je, iz direktorija u kojem se nalaze sve 3 datoteke programa, u naredbenom retku pokrenuti naredbu:

```
python main.py
```

5.6. Primjer izvođenja

1. Korisnika se prvo traži da upiše broj N koji će algoritam faktorizirati (npr. 15).

   ```
   Unesite broj za rastav na proste faktore: 15
   ```

2. Odabire se broj a i provjerava je li relativno prost s N.

   ```
   Odabiremo nasumicni broj a = 13
   ```

   Ako a nije relativno prost s N, ispisuje se rješenje (npr. $a = 10$).

   ```
   Pogodili smo a s jednim istim prostim faktorom kao 15,
           ne moramo dalje racunati
   Rjesenja su 5 i 3
   ```

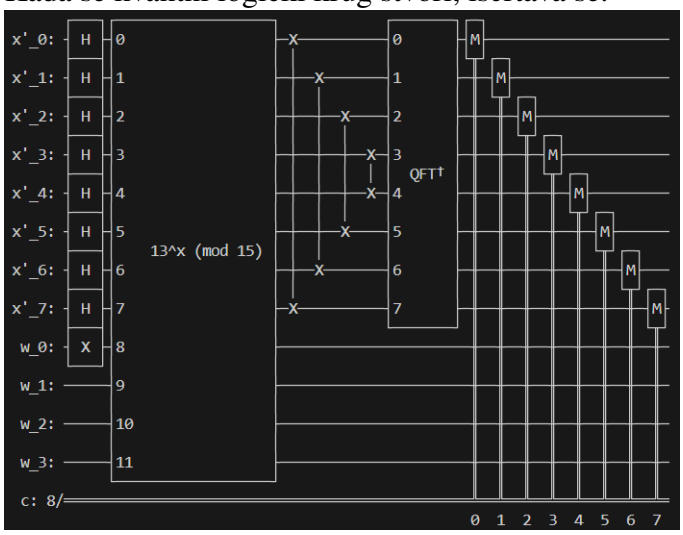
3. Parametri funkcije modularnog potenciranja a i N predaju se kvantnom dijelu algoritma.

   ```
   Formiramo funkciju f(x) = 13^x (mod 15)
           i trazimo njen period
   ```

4. Stvara se kvantni logički krug s danim parametrima.

   ```
   Stvaranje kvantnog logickog kruga
   ```

5. Kada se kvantni logički krug stvori, iscrtava se.



6. Kvantni logički krug priprema se za izvođenje na simulatoru.

```
Priprema simulacije
```

7. Kvantni logički krug izvodi se na simulatoru.

```
Pokretanje simulacije
Kraj simulacije
```

8. Računaju se mogući periodi iz izmjerenih faznih pomaka i poredaju se po učestalosti.

```
Izmjereni fazni pomaci:
        theta = 3/4
        theta = 1/2
        theta = 1/4
Moguci periodi i broj ponavljanja:
        r = 4   (47 ponavljanja)
        r = 2   (29 ponavljanja)
```

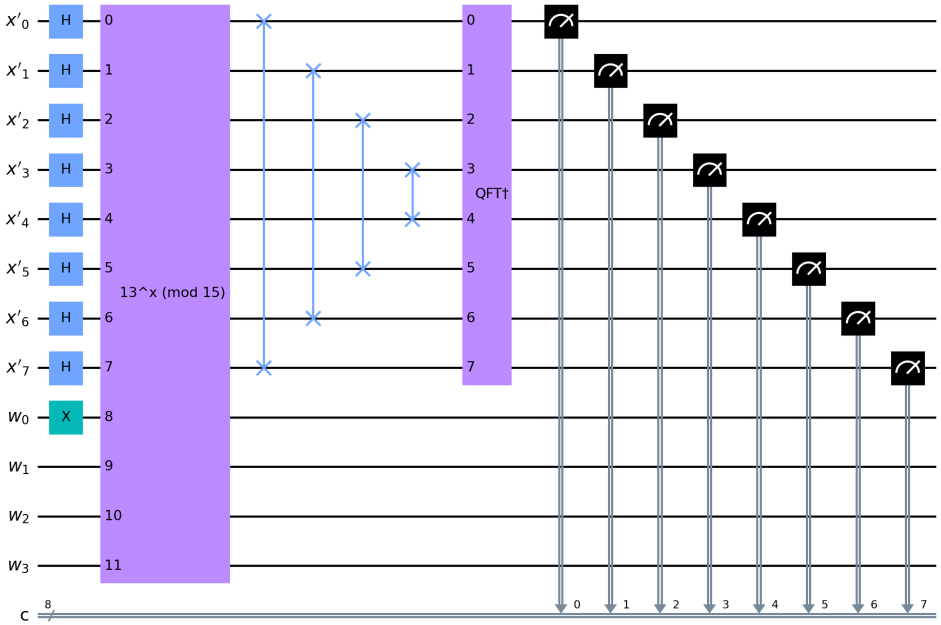
9. Po redu od najučestalijeg, za svaki se period r provjerava zadovoljava li kriterije $2 \mid r$ i $N \nmid (a^{\frac{r}{2}} + 1)$. Ako ne, provjera se sljedeći period r. Ako ni jedan ne zadovoljava kriterije, vraća se na korak 2.

```
r = 4 zadovoljava kriterije
```

10. Ako je pronađen zadovoljavajuć period r, računaju se prosti faktori od N.

```
Racuanmo nzd(a^(r/2) + 1, N) = nzd(170, 15) = 5
Racuanmo nzd(a^(r/2) - 1, N) = (168, 15) = 3
Rjesenja su 5 i 3
Vrijeme izvodenja 1.54 s
```

Tijekom izvođenja, program generira dijagram `shor_krug.png` i OpenQASM2.0 kod `shor_qasm.txt` korištenog kvantnog logičkog kruga. Datoteke smjesti u isti direktorij u kojem se nalazi izvorni kod programa.



Slika 5.4: Kvantni logički krug koji program generira (iz primjera)

Ako se na ulaz programa dovede N koji je prost broj, program će za svaki a iz skupa $\{2, 3, \ldots, N-1\}$ stvoriti kvantni logički krug, simulirati ga i dobiti nezadovoljavajuć rezultat. Program će stati s izvođenjem kad prođe sve brojeve iz skupa.

Ako se pak na ulaz dovede N koji je umnožak više od 2 prosta broja, program će podijeliti broj na umnožak 2 broja veća od 1, bez jamstva da je jedan od njih prost.

5.7. Vremenska analiza izvođenja simulacije

Na temelju nekoliko primjera, analizirat će se brzina izvođenja programa. Koristi se lokalni simulator AER i parametri SHOTS = 100 i K = 2. Gledaju se slučajevi kada se ne mora ponavljati odabir a nakon pronalaženja perioda r. Ako je veličina kvantnog registra w jednaka n, onda je ukupan broja qubitova u sustavu jednak $3n$.

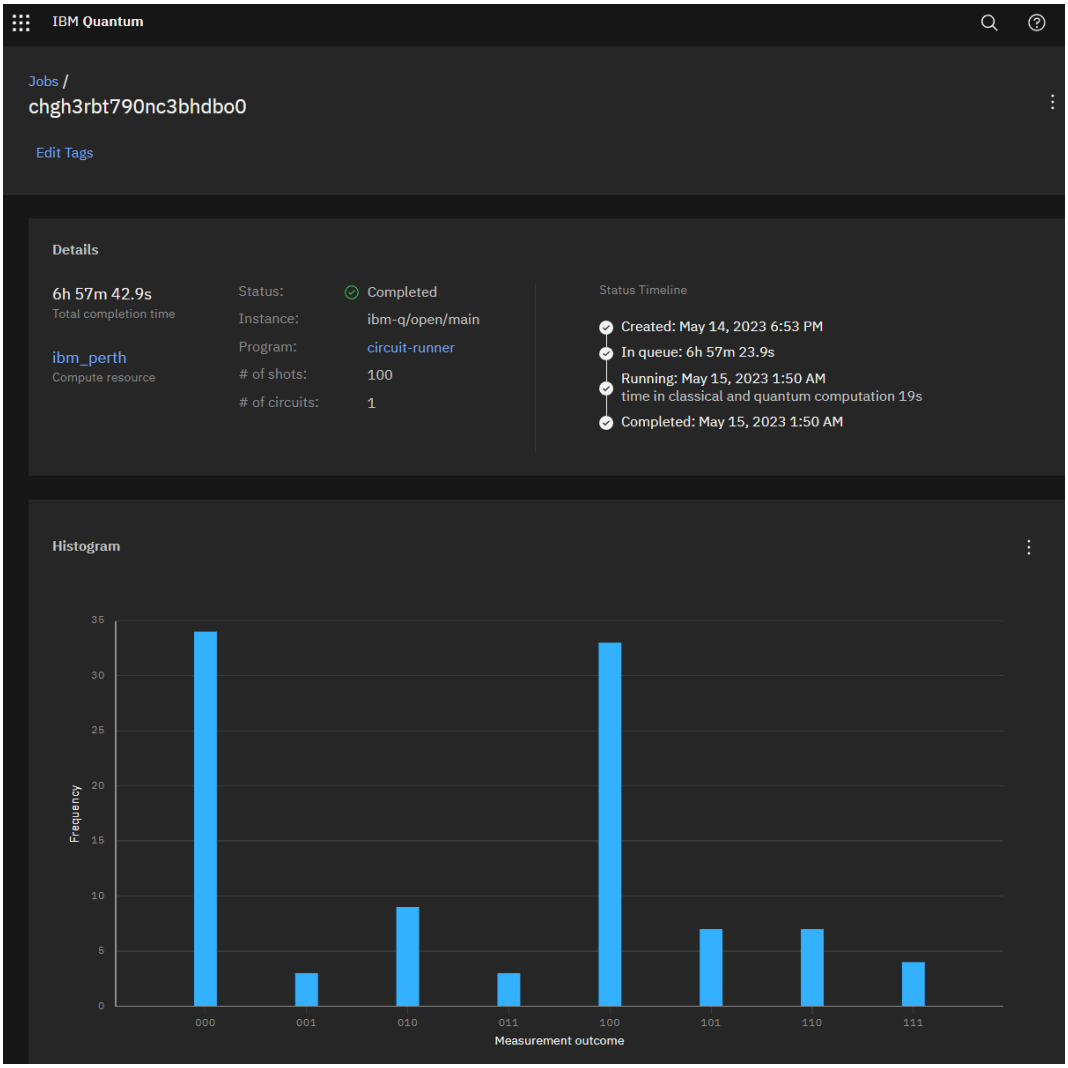
Ulaz N	Prosti faktori p i q	Broj qubitova n registra w	Vrijeme izvođenja
6	2, 3	3	1.10 s
10	2, 5	4	1.53 s
14	2, 7	4	2.88 s
15	3, 5	4	1.31 s
21	3, 7	5	12.28 s
22	2, 11	5	12.42 s
26	2, 13	5	12.58 s
33	3, 11	6	69.22 s
55	5, 11	6	78.73 s
65	5, 13	7	379.14 s
119	7, 17	7	455.97 s

Iz rezultata očigledno je da broj qubitova n najviše utječe na vrijeme izvođenja programa. To je očekivano jer za povećanje broja n za 1, broj qubitova u sustavu povećava se za 3, a dimenzije vektora i matrica, u sustavu kojim se simulira izvođenje, povećavaju se čak 8 puta.

5.8. Pokretanje na stvarnom kvantnom računalu

Tvrtka IBM korisnicima svoje platforme IBMQ omogućava besplatno korištenje kvantnih procesora s 5 i 7 qubitova. Ako bi se koristio kvantni logički krug s parametrom K = 2, ne bi se mogao faktorizirati ni broj 6 (potrebno je 9 qubitova). Stoga je potrebno koristiti manji K. Mogu se upotrijebiti kombinacije veličina registara x i w tako da su u zbroju manje ili jednake 7. To će, nažalost, dovesti do manje preciznog mjerenja faznog pomaka.

Kako bi se program mogao pokrenuti na kvantnom računalu od 7 qubitova, potrebno je postaviti parametar K na 1, dodati API ključ i promijeniti udaljeni poslužitelj s IBM-ovog simulatora na neko od IBM-ovih kvantnih računala. Sada se može pokrenuti program za faktorizaciju broja $N = 6$ s funkcijskim parametrom $a = 5$. Preko stranice IBMQ može se vidjeti status zadanog posla.



Slika 5.5: Sučelje stranice IBMQ za pregledavanje dovršenog posla

Po završetku posla, rezultati su dostavljeni programu, a vidljivi su i na stranici IBMQ. Budući da je red čekanja za pokretanje trajao 6 sati i 57 minuta, program je u međuvremenu bio ugašen, no neovisno o tome rezultati se mogu ručno analizirati.

Najčešće izmjeren broj (osim 0 koji ne nosi nikakvu informaciju) bio je $2^n\vartheta = 100_{(2)} = 4_{(10)}$. Kako je $n = 3$ qubitova, fazni pomak iznosi $\vartheta = \frac{4}{2^3} = \frac{1}{2}$, a to odgovara periodu $r = 2$. Potvrda da je to doista period funkcije $f(x) = 5^x \pmod 6$:

$$f(0) = 1 \pmod 6 \tag{5.1}$$

$$f(1) = 5 \pmod 6 \tag{5.2}$$

$$f(2) = 1 \pmod 6 \tag{5.3}$$

6. Zaključak

Shorov algoritam uspješno ispunjava svoju zadaću i relativno je jednostavan za programski ostvariti. Glavni nedostatak dolazi izvan samog algoritma, a to je nedostatak dovoljno moćnih kvantnih računala. Nedostatak samog programskog ostvarenja nalazi se u neučinkovitoj izradi operatora modularnog potenciranja, što usporava stvaranje krugova i izvođenje simulacija.

Prema nekim procjenama, za razbijanje kriptosustava RSA, potrebno je računalo s barem milijun qubitova[7]. Jasno je da ako je trenutno najveći broj qubitova u kvantnom računalu 433, kriptosustav RSA neće biti probijen u skorije vrijeme.

Unatoč tome, teorijsko razmatranje Shorovog algoritma i drugih trenutno nedostižnih aspekata kvantnog računarstva otišlo je vrlo daleko i ne čeka da tehnologija sustigne. Ovaj rad primjer je toga, program može primiti broj bilo koje veličine, no na stvarnom kvantnom računalu ne može faktorizirati broj veći od 6.

LITERATURA

[1] IBM quantum, 2021. URL https://quantum-computing.ibm.com/.

[2] IBM unveils breakthrough 127-qubit quantum processor. 2021. URL https://newsroom.ibm.com/2021-11-16-IBM-Unveils-Breakthrough-127-Qubit-Quantum-Processor.

[3] China has quantum computers that are 1 million times more powerful than google's. 2021. URL https://techhq.com/2021/10/china-has-quantum-computers-that-are-a-million-times-more-powerful-than-googles/.

[4] Toshiba announces breakthrough in long distance quantum communication. 2021. URL https://www.toshiba.eu/pages/eu/Cambridge-Research-Laboratory/toshiba-announces-breakthrough-in-long-distance-quantum-communication.

[5] IBM unveils 400 qubit-plus quantum processor and next-generation ibm quantum system two. 2022. URL https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two.

[6] A huge step forward in quantum computing was just announced: The first-ever quantum circuit. 2022. URL https://www.sciencealert.com/a-huge-step-forward-in-quantum-computing-was-just-announced-the-first-ever-quantum-circuit.

[7] Are quantum computers about to break online privacy? 2023. URL https://www.nature.com/articles/d41586-023-00017-0.

[8] Saša Ilijić. Kvantna računala. URL http://sail.zpf.fer.hr/labs/kvarac/slides/.

[9] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023. URL https://qiskit.org/learn/.

Programsko ostvarenje Shorovog algoritma u simulatoru kvantnog računala

Sažetak

Shorov algoritam kvantni je algoritam, koji u polilogaritamskom vremenu rješava problem faktorizacije brojeva čiji su faktori veliki prosti brojevi. U ovom radu dan je pregled kriptosustava RSA, kvantnog računarstva, kvantnih algoritama i Shorovog algoritma. Konačno, opisuje samostalno programsko ostvarenje Shorovog algoritma u simulatoru kvantnog računala koristeći slobodno dostupne knjižnice programa.

Ključne riječi: Shorov algoritam, kvantni algoritam, kvantno računalo, simulator kvantnog računala, RSA, kriptografija, asimetrična kriptografija, faktorizacija

Shor's algorithm implementation on a quantum computer simulator

Abstract

Shor's algorithm is a quantum algorithm that can solve the problem of finding large prime factors of integers in polylogarithmic time. This paper gives an overview of the RSA encryption scheme, quantum computing, quantum algorithms and Shor's algorithm. Finally, it describes a self-made implementation of Shor's algorithm in a quantum computer simulator using publicly available program libraries.

Keywords: Shor's algorithm, quantum algorithm, quantum computer, quantum computer simulator, RSA, cryptography, asymmetric cryptography, factorization