Algorytm Shor'a

Konrad Rzońca

15 lutego 2018

Spis treści

1 Wstęp

Aby zrozumieć i skutecznie omówić działanie algorytmu Shor'a na przykładzie układu elektronicznego zacznę od omówienia działania komputera kwantowego, używanych w nim bramek i idei stojącej za algorytmem.

1.1 Komputer kwantowy

1.1.1 Qubity

W taki sam sposób jak tradycyjne komputery operują na bitach, komputery kwantowe operują na qubitach, są to zazwyczaj pojedyńcze elektrony lub protony.[5]

Spin cząstki w polu magnetycznym, w momencie pomiaru, może przyjmować tylko dwa stany - zgodnie z polem w którym się znajduje czyli spin-up state, lub przeciwnie - spin-downstate. Jednak w większości przypadków znajdują się one w stanie superpozycji - ich stan jest niemożliwy do stwierdzenia, aż do dokonania pomiarów, możemy go jedynie określać z pewnym prawdopodobieństwem. [9]

Dodatkową własnością takich cząstek, jest to, że mogą być ze sobą w stanie związania (quantum entanglement), jest to zjawisko jeszcze niewyjaśnione przez fizykę, pomiar jednej ze związanych ze sobą cząstek determinuje stan tej drugiej - zależnie od splątania będzie miała ona stan ten sam lub przeciwny do cząstki którą mierzymy [5]

1.1.2 Stan qubita

Stan każdego qubita w układzie określić możemy jako $a|0\rangle + b|1\rangle, a, b \in C$, gdzie $a^2 + b^2 = 1$. W praktyce oznacza to, że prawdopodobieństwo, że po pomiarze qubit przyjmie stan 0 z prawdopodobieństwem a^2, stan 1 b^2.

1.1.3 Stan komputera kwantowego

Stan całej maszyny określa się jako przestrzeń Hilberta w której $|0\rangle$ oraz $|1\rangle$ są jej wektorami bazowymi. [3] Aktualny stan maszyny jest punktem w 2^n wymiarowej przestrzeni wektorowej. Kombinacje stanu dwóch systemów kwantowych możemy określić jako tensor z argumentów podsystemów, zatem stan układu dwoje qubitów: $|A\rangle = a_0|0\rangle + a_1|1\rangle$ oraz $|B\rangle = b_0|0\rangle + b_1|1\rangle$ możemy rozpisać jako:

$$|A\rangle|B\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{pmatrix}$$

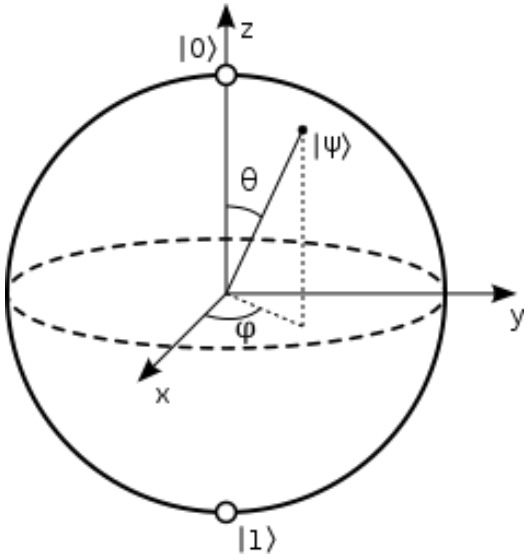
Stan takiego układu możemy zapisać jako $|A, B\rangle$.

Późniejsze operacje na qubitach zazwyczaj są opisywane jako macierze $2^n \times 2^n$, przykładowo reprezentacja macierzowa M =

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ i & i & 0 & 0 \\ 0 & 1 & i & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

oznaczałaby, że
$M|0,0\rangle = |0,1\rangle$, czyli prawdopodobieństwo układu 0,0 byłoby po przejściu równe prawdobodobieństwu układu 0,1 przed przejściem
$M|0,1\rangle = i|0,0\rangle + i|0,1\rangle$ itd.
Zauważmy, że bramka M nie byłaby fizycznie możliwa do zrealizowania, ponieważ $M|0,0\rangle^2 + M|0,1\rangle^2 + M|1,0\rangle^2 + M|1,1\rangle^2$!= 1
(chociażby dlatego, że dla układu wejściowego spełniającego warunek z kwadratami $M|1,1\rangle^2$ musiałoby się równać 1 a $M|0,0\rangle^2 + M|0,1\rangle^2 + M|1,0\rangle^2 > 0$)

1.1.4 Sfera Bloch'a



Rysunek 1: Sfera Bloch'a

Jednym z bardziej popularnych sposobów przedstawiania stanu qubita jest tzw. sfera Bloch'a. Rezprezentuje ona stan pure pojedyńczego stanu kwantowego drugiego poziomu który możemy zapisać w postaci

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Stan ten, tak jak było to opisane w poprzednich podpunktach, musi spełniać warunek $a^2+b^2 = 1$. Rozważmy liczbę zespoloną z która będzie reprezentacją stanu naszego qubita

$$z = x + iy, x, y \in R$$

Liczba z musi spełniać nasz pierwotny warunek zatem $|z|^2 = 1$, rozpisując ją dalej uzyskujemy $z * z = (x+iy)^2 = x^2+y^2$, jest to równanie koła w centrum w (0,0).[4]

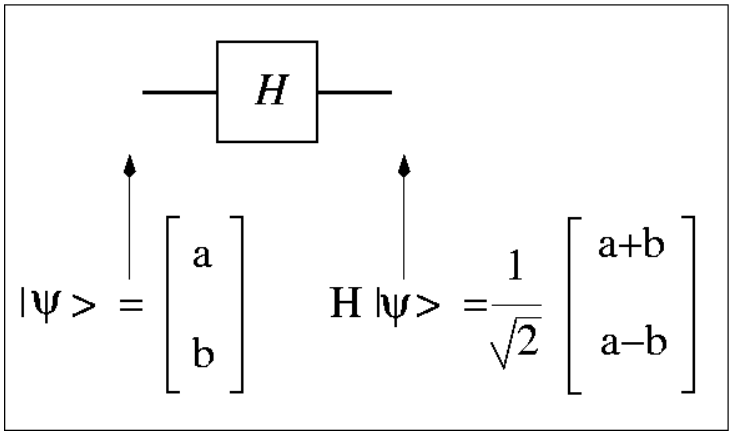
Przechodząc na współrzędne sferyczne i biorąc pod uwagę, ze promień sfery którą rozważamy wynosi 1 uzyskujemy $z = r(\cos\theta + i\sin\theta) = re^{i\theta} = e^{i\theta}$ Pozostawia nas to zatem z jednym stopniem swobody, reprezentacja fizyczna z-eta to po prostu stan naszego spinu, dla z = 1 spin będzie górny (spin-up - $|0\rangle$, czyli a = 1), natomiast kiedy z = -1 to spin będzie dolny.

Wracając do naszych parametrów a, b je reprezentujemy kolejno jako $r_a e^{i\phi_a}$ i $r_b e^{i\phi_b}$, nie potrzebujemy się wgłębiać bardziej w matematykę, co jest istotne to intuicja związana z tym układem która jest potem przydatna w rozumieniu bramek Pauliego.

1.2 Podstawowe bramki

1.2.1 Hadamard gate

Podstawową intuicją stojącą za bramkami Hadamarda jest to, że pozwalają nam przejść w stan super pozycji dwóch stanów, oraz z niego wyjść. Bramki Hadamarda wykorzystuje się głównie, aby generować losowe stany wyjścia, sytuacja jednak staje się ciekawsza kiedy połączymy je z innymi bramkami, możemy wtedy wykonywać obliczenia na wszystkich możliwych wejściach jednocześnie, następnym ciekawym faktem jest, że po dwuktrotnym przejściu przez bramkę Hadamarda otrzymamy identyczność. [8]



[1]

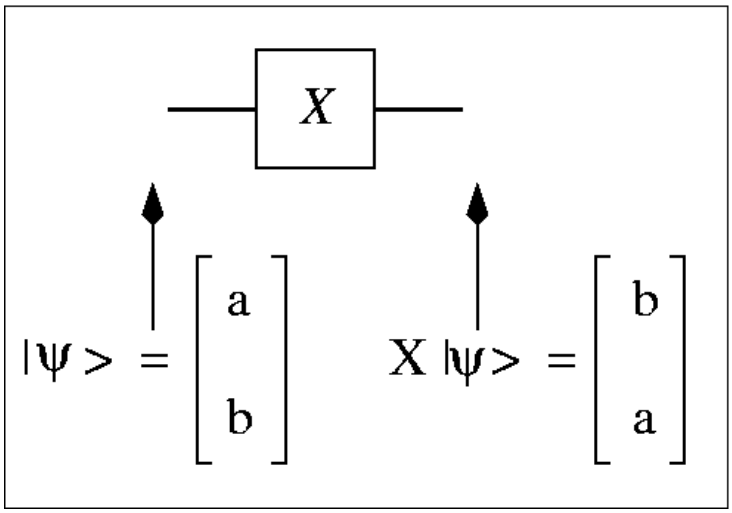
$$|0\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|1\rangle \rightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Bramkę Hadamarda możemy przedstawić również w reprezentacji macierzowej

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

1.2.2 Pauli-X gate czyli kwantowa bramka NOT

Bramka X jest odpowiednikiem kwantowym klasycznej bramki NOT - zmienia stan qubita na przeciwny. Obraca ona nasz układ o π rad w okół osi x w układzie sfery Bloch'a[10]



[1]

Bramka X przedstawiona w reprezentacji macierzowej to:

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

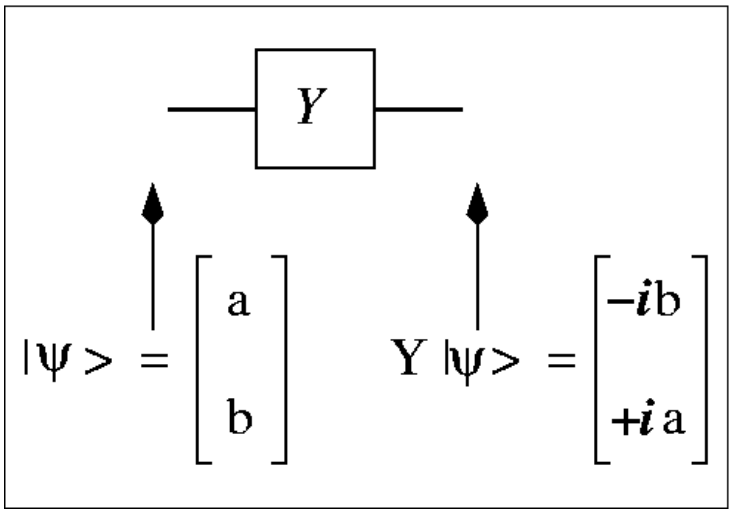
natomiast w układach bramki NOT często są przedstawiane również jako


[10]

Mapuje ona wartość $|0\rangle$ do $|1\rangle$ i na odwrót. X(X(A)) = A

1.2.3 Pauli-Y gate

Ta bramka nie ma odpowiednika w układach klasycznych, tak jak bramka X obraca układ na osi x o π radianów, tak bramka Y robi to dla osi y[10]



[1]

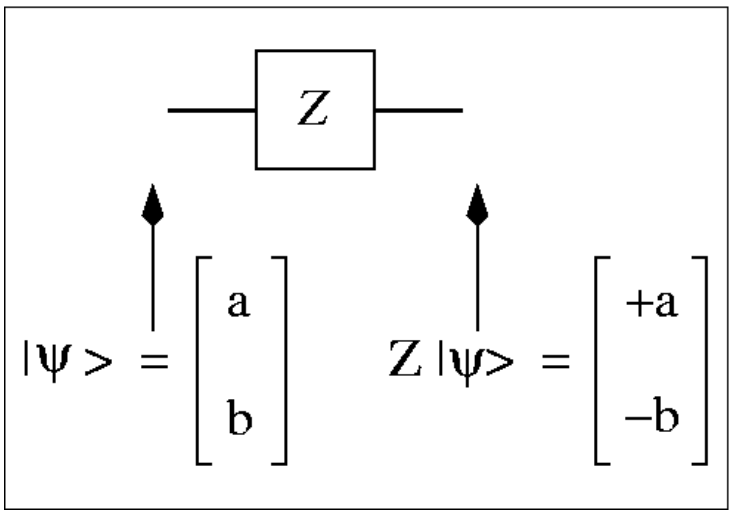
Bramka Y przedstawiona w reprezentacji macierzowej to:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Możemy łatwo obliczyć, że Y(Y(A)) = A, co jest zresztą bardzo intuicyjne - obrót w okół którejkolwiek z osi układu współrzędnych o łączne 2π radianów wraca nasz układ do pozycji startowej

1.2.4 Pauli-Z gate

Analogicznie, bramka Z obraca układ o π radianów na osi z.[10]



[1]

Z(Z(A)) = A
Bramka Z przedstawiona w reprezentacji macierzowej to:

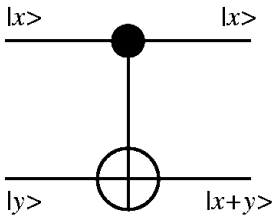
$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Z(Z(A)) = A

1.2.5 CNOT gate

Bramka CNOT (controled NOT) do działania potrzebuje przynajmniej dwóch qubitów, domyślnie zmienia ona stan qubita na przeciwny kiedy qubit kontrolny jest równy 1[10]

Schemat bramki CNOT kontrolowanej przez qubit to



[1]

Schemat i działanie wszystkich bramek kontrolowanych jest analogiczny: są oznaczane jak bramka kontrolowana dopięta do qubita kontrolującego jej działanie. Nie będę w związku z tym omawiał wersji kontrolowanych wszystkich przedstawionych bramek.

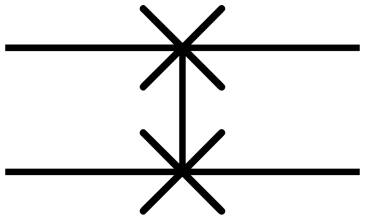
Powyższa bramka CNOT przedstawiona w reprezentacji macierzowej to:

$$CNOTx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$CNOTx|0,0\rangle = |0,0\rangle$
$CNOTx|0,1\rangle = |0,1\rangle$
$CNOTx|1,0\rangle = |1,1\rangle$
$CNOTx|1,1\rangle = |1,0\rangle$
Widzimy zatem, że dla qubita kontrolnego x równego 1 bramka wykonuje operację NOT na wejściu y, w systemie binarnym możemy po prostu zapisać, że y = x + y
Pamiętajmy, że współczynniki przy poszczególnych qubitach opisane w 1.1.2 to nie ich wartość, ale prawdopodobieństwo, że ją przybiorą, w trakcie pomiaru qubit zawsze jest równy 0 lub 1 - stan wyjściowy bramki przybiera wartość jednoznaczną.

1.2.6 S gate

Bramka S nazywana również SWAP służy do zamiany stanu dwóch bitów.[10]



[10]

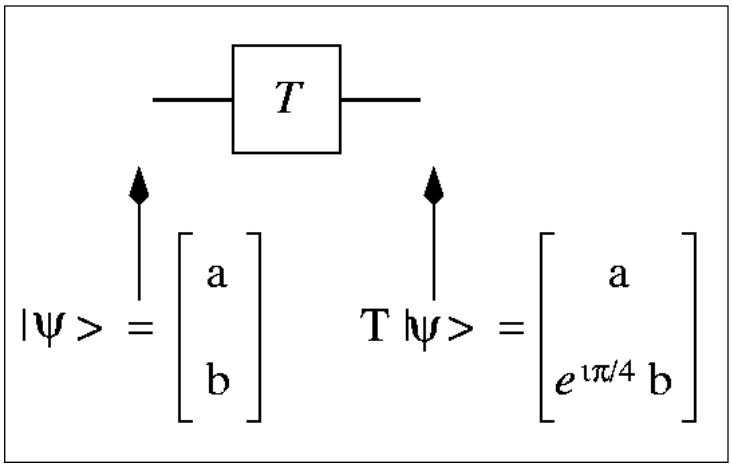
Bramka S przedstawiona w reprezentacji macierzowej to:

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

$SWAP|0,0\rangle = |0,0\rangle$
$SWAP|0,1\rangle = |1,0\rangle$
$SWAP|1,0\rangle = |0,1\rangle$
$SWAP|1,1\rangle = |1,1\rangle$
Łatwo wywnioskować, że S(S(A)) = A

1.2.7 T gate

Bramka T to jak bramka Z ale obracamy nie o π a $\frac{\pi}{4}$



[10]

Bramka T przedstawiona w reprezentacji macierzowej to:

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

$T|0\rangle = |0\rangle$
$T|1\rangle = e^{i\pi/4}|1\rangle$

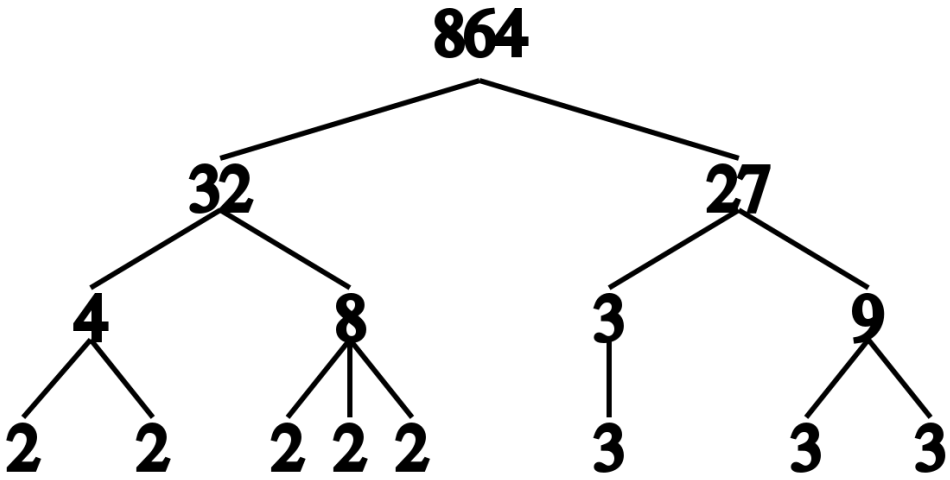
T(T(T(T(T(T(T(A)))))))) = A - po 8 krotnym użyciu bramki T qubit wraca do stanu pierwotnego: $\frac{\pi}{4} * 8 = 2\pi$, spotkałem się również w podanych źródłach z bramką S(A) = T(T(A)), zatem SWAP będziemy zawsze oznaczać jako SWAP.

Jeżeli chodzi o intuicje stojącą za tym, że całkowite prawdopodobieństwo będzie nadal równe jeden wróćmy do naszej reprezentacji układu w sferze Bloch'a b = $r_b e^{i\phi_b}$, po pomnożeniu przez $e^{i\pi/4}$, $(|b'|)^2 = (|r_b e^{i\phi_b} * e^{i\pi/4}|)^2 = b^2 * |e^{i\pi/2}| = b^2$ zatem równanie jest zachowane, bardziej ogólnie obracając w okół z zawsze zachowamy wartość modułu.

2 Opis ogólnego działania algorytmu Shor'a

2.1 Zasada działania algorytmu Shor'a

Algorytm Shor'a jest algorytmem służącym do szukania rozkładu liczby naturalnej na dwie liczby pierwsze, wiemy, że każda liczba ma dokładnie jeden taki rozkład.[7]



Rysunek 2: rozkład 864

Dzięki narzędziom matematycznym możemy ograniczyć problem szukania rozkładu liczby to problemu szukania okresu pewnego ciągu elementów, weźmy sobie np. ciąg którego n-ty wyraz to$x(n) = 2^n$

$$2, 4, 8, 16, 32, 64, 128, 256...$$

Teraz, weźmy zamiast tego ciąg $x(n) = 2^n \mod 15$, czyli reszty z dzielenia przez 15 każdego z wyrazów wyższego ciągu, co otrzymamy to

$$2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1...$$

Jest to ciąg okresowy o okresie równym 4. Euler odkrył, że jeżeli, weźmiemy dwie liczby pierwsze p i q, $m = p * q$, oraz jakieś x niepodzielne ani przez q ani p (w naszym wyższym przykładzie 2) to ciąg $a_n = x^n \mod m$

$$x \mod m, x^2 \mod m, x^3 \mod m, ...$$

posiada okres który jest dzielnikiem (p - 1)(q - 1). Dla naszego przykładu p = 3, q = 5, m = 15, x = 2, zatem zgadza się, 4 dzieli 2*4 = 8. Daje nam to jeden z dzielników (p-1)(q-1), po wypróbowaniu wystarczająco wielu x-ów uzyskujemy cały iloczyn co z kolei daje nam potem możliwość wiliczenia samych p i q.

Jest to niewykonywalne w tradycyjnym systemie komputerowym ponieważ sam okres może wynosić prawie m, a jest tylko jednym z kroków w działaniu naszego algorytmu, przy dużych liczbach jak te używane do szyfowania RSA ma to bardzo dużą złożoność, jednak układy kwantowe pozwalają nam ją obejść.

Sam algorytm Shor'a działa w następujących krokach:

1. Bierzemy losową liczbę naturalną k taką, że k $<$ m. Obliczamy (ze złożonością wielomianową np. algorytmem Euklidesa) największy wspólny dzielnik m i k (gcd(m,k)), jeżeli gcd(m,k) $\neq 1$, to znaleźliśmy dzielnik m i jest nim k, kończymy. jeżeli jednak gcd $= 1$ kontynuujemy dalej.

2. Znajdujemy okres P

$$k \bmod m, k^2 \bmod m, k^3 \bmod m, ...$$

3. Dla P nieparzystego wracamy na początek, dla parzystego kontynuujemy

4. $(k^{P/2}-1)(k^{P/2}+1) = k^P - 1$, teraz zauważmy, że tak jak w przykładzie gdzie k $= 2$, $k^P \bmod m = 1$, wracając jeszcze raz do przykładu

$$2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1...$$

   $2^4 \bmod 15 = 16 \bmod 15 = 1$, ostatni wyraz w pojedyńczym okresie tego ciągu to zawsze jeden dla dowolnych k,m naturalnych, będzie on miał numer P stąd ten rachunek.
   Dalej, wiemy już, że $(k^P - 1) \bmod m = 0$, jeżeli $(k^{P/2} + 1) \bmod m = 0$ to wracamy do początku, inaczej kontynuujemy

5. Liczymy d $= gcd(k^{P/2} - 1, m)$, dla $(k^{P/2} + 1) \bmod m \neq 0$ potrafimy wykazać, że d jest nietrywialnym $(\neq 1, m)$dzielnikiem m-a.
   Zwracamy d. W ten sposób możemy wyznaczyć dzielniki m-a, jeżeli to dwie liczby pierwsze tak jak w RSA to je znajdziemy.

2.2 Implementacja algorytmu Shor'a na układzie kwantowym

Kwantowy algorytm shor'a do działania potrzebuje dwóch rejestrów kwantowych. Na początku pierwszy rejestr musi znaleźć liczbę $q = 2^s, s \in N$ która spełni warunek $n^2 \leq q < 2n^2$, gdzie n to rozkładana liczba, zauważmy, że zawsze istnieje taka liczba. Następnie należy wykonać następujące kroki[3]

1. *Inicjalizacja*: Umieszczamy pierwszy rejestr w superpozycji

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle$$

   Zauważmy, że $0, 1, \sqrt{2}, \sqrt{3}...\sqrt{q-1}$ to wszytskie możliwe dzielniki pierwsze n-a co wynika z warunku jaki narzuciliśmy na q.

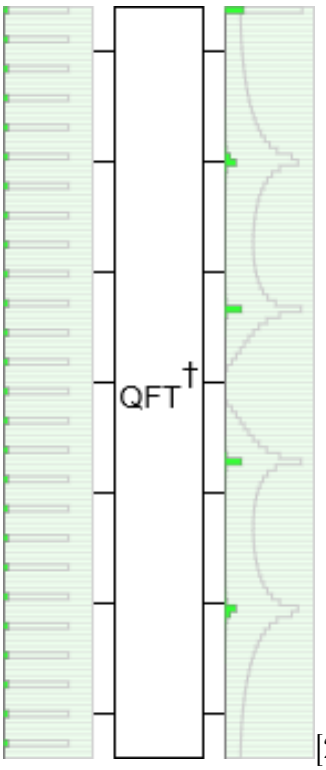
2. *Obliczenia*: Wybieramy losowe x i liczymy w drugim rejestrze x^a

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|x^a\rangle$$

   Teraz przykład, stan maszyny kwantowej

$$\frac{1}{\sqrt{7}} \sum_{k=0}^{4} |5k\rangle$$

   możemy zapisać jako $\frac{1}{\sqrt{7}}|00000\rangle + \frac{1}{\sqrt{7}}|00101\rangle + \frac{1}{\sqrt{7}}|01010\rangle + \frac{1}{\sqrt{7}}|01111\rangle + \frac{1}{\sqrt{7}}|11001\rangle$, przypisuje on $\frac{1}{\sqrt{7}}$ decymalnie stanom 0,5,10,15,20. Inne stany nie będą posiadały wartości, zatem posiada on okres równy 5. Po transformacie Fouriera uzyskalibyśmy zatem 5 lokalnych maksimów w rozkładzie stanu tego układu.[2]



[2]

Na tym obrazku maksima reprezenotwane są przez zielone znaczniki. ransformatę Fouriera będziemy wykonywać w kroku następnym na pierwszym rejestrze.

Wracamy do naszego układu

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |x^a\rangle$$

3. *Transformata Fouriera*: Za pomocą bramek Hadamarda i bramek zmiany fazy jesteśmy w stanie przeprowadzać transformaty Fouriera w naszych układach, po dokonaniu jej na pierwszym rejestrze stan naszej maszyny możemy wyrazić jako:

$$\frac{1}{q} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} (e^{2\pi i ac/q}) |c\rangle |x^a\rangle$$

4. *Obserwacje*: W naszej sumie znajdzie się okres którego szukamy, mianowicie jeżeli weźmiemy jakieś x^k, i weźmiemy te części sumy po a w których $x^a \equiv x^k$ mod n, do dla danego c prawdopodobieństwo, że maszyna znajdzie się w stanie $|c\rangle|x^k\rangle$ możemy określić jako

$$\left| \frac{1}{q} \sum_{a:\ x^a \equiv x^k}^{q-1} \left(e^{2\pi i a c/q}\right) \right|^2$$

k wybieramy takie, że $0 \leq$ k $<$ r, a r to najmniejsza liczba naturalna spełniająca równanie $x^r \equiv 1$ mod n

5. *Continued Fraction Expansion*: Jeżeli istnieje takie d, że

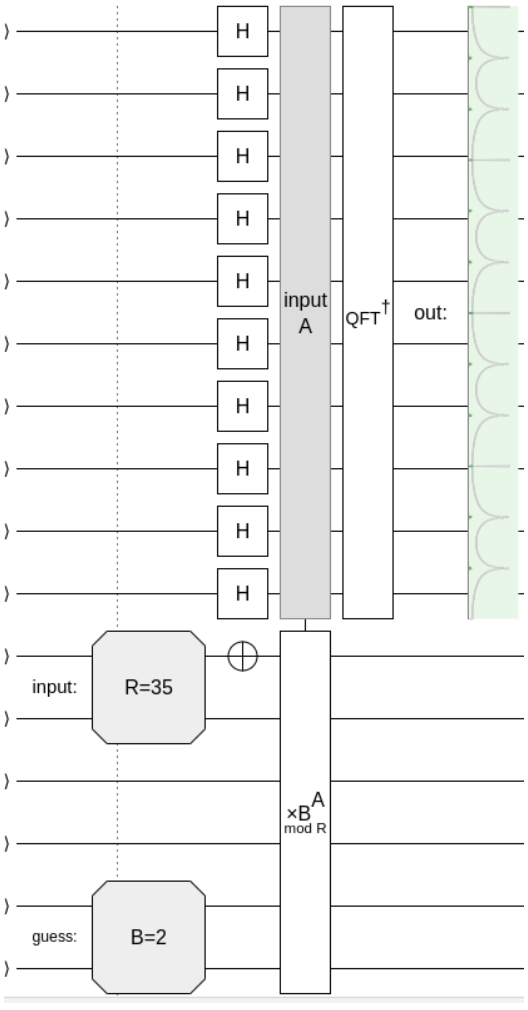
$$\frac{-r}{2} \leq dq - rc \leq \frac{r}{2}$$

to prawdopodobieństwo wystąpienia $|c, x^k\rangle$ jest większe niż $1/3r^2$, następnie d/r może być otrzymane przez zaokrąglenie c/q. r to prawdopodobny okres Q.

3 Działanie algorytmu Shor'a na przykładzie rozkładu liczby 21 w konkretnym obwodzie kwantowym za pomocą narzędzia Quirk

Posługiwać się będę krokami z rozdziału 2.1 i narzędziem Quirk, a konkretnie zaproponowanym w nim obwodzie to liczenia okresu P. Jest to jedynie obwód symulacyjny i wiele dość skomplikowanych układów upraszcza, jednak pomaga on przedstawić samą ideę.

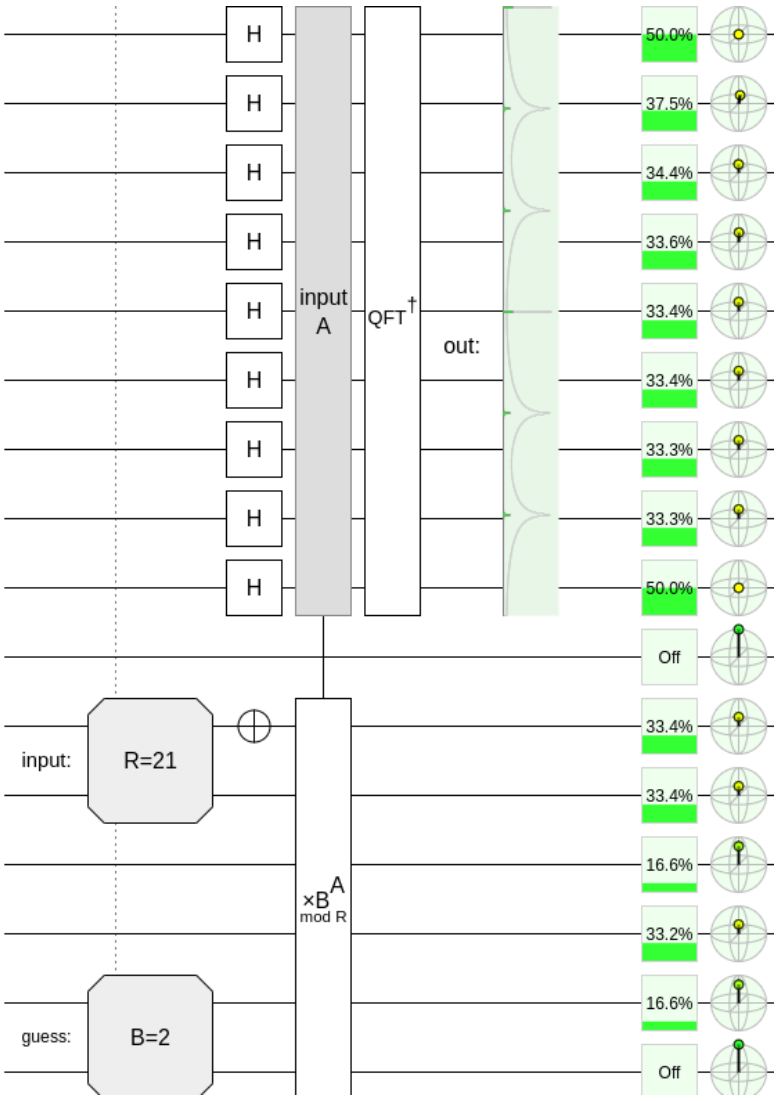
Schemat proponowany przez Quirka wygląda następująco



[6]

Najpierw inicjalizujemy 10 stanów nieoznaczonych za pomocą bramek Hadamarda, $21^2 = 441$, $441 < 512 < 882$, potrzebujemy zatem 9 bramek więc 10 wyrzucam z układu. Następnie liczymy $k^r mod N$, gdzie k to wybrana w kroku pierwszym liczba, N to liczba której rozkładu szukamy, a r to będą wszystkie liczby z zakresu od 0 do 512, na raz. Następnie za pomocą transformaty Fouriera zczytujemy okres, przejdźmy do praktyki.
21 wybrałem losowo i robię to pierwszy raz na quirku.

1. Za k biorę 2, spełnia warunki
2. Okres jaki wskazał Quirk to 6, obrazek z pomiaru na następnej stronie
3. 6 jest parzyste, kontynuujemy
4. 9 mod 21 to nie zero więc przechodzimy do ostatniego kroku
5. Liczymy d = gcd(7, 21) = 7, zgadza się ! 21/7 = 3, zatem algorytm działa


H
H
H
H
H
H
H
H
H
input
A
QFT†
out:
50.0%
37.5%
34.4%
33.6%
33.4%
33.4%
33.3%
33.3%
50.0%
Off
input:
R=21
33.4%
33.4%
16.6%
33.2%
16.6%
Off
×B^A
mod R
guess:
B=2

Bibliografia

[1] *agh jeanbel*. URL: http://people.math.gatech.edu/~jeanbel/4803/Lectures/Gates/qgates.html.

[2] *Algorithmic Assertions*. URL: http://algassert.com/post/1718.

[3] *Comment on Demonstrations of Shor's Algorithm in the Past Decades by Zhengjun Cao1, Zhenfu Cao, Lihua Liu*. URL: https://eprint.iacr.org/2015/1207.pdf.

[4] *Ian Glendinning for EUROPEAN CENTRE FOR PARALLEL COMPUTING AT VIENNA*. URL: http://www.vcpc.univie.ac.at/~ian/hotlist/qc/talks/bloch-sphere.pdf.

[5] *qubit by Margaret Rouse*. URL: http://whatis.techtarget.com/definition/qubit.

[6] *Quirk!* URL: http://algassert.com/quirk.

[7] *Stephanie Blanda - Shor's Quantum Factoring Algorithm*. URL: https://blogs.ams.org/mathgradblog/2014/04/30/shors-algorithm-breaking-rsa-encryption/.

[8] *The Hadamard gate by Michael Nielsen*. URL: https://www.youtube.com/watch?v=x6gOp_o7Bi8.

[9] *Veritasum*. URL: https://www.youtube.com/watch?v=v1_-LsQLwkA.

[10] *Wikipedia the free encyklopedia*. URL: https://en.wikipedia.org/wiki/Quantum_gate.