



**POLITECHNIKA
BYDGOSKA**
Wydział Telekomunikacji,
Informatyki i Elektrotechniki

PRACA MAGISTERSKA

na kierunku Informatyka Stosowana II st.

Badanie możliwości lub zastosowań obliczeń kwantowych w placówkach dydaktycznych lub badawczych

Exploring the possibilities or applications of quantum computing in teaching or research settings

Piotr Muchowski
121716

Promotor: dr hab. inż. Rafał Długosz prof. PBŚ

Bydgoszcz, Czerwiec 2024

Metryka pracy magisterskiej

Dane ogólne

Nazwa Uczelni	Politechnika Bydgoska im. Jana i Jędrzeja Śniadeckich
Wydział	Telekomunikacji, Informatyki i Elektrotechniki
Kierunek	Informatyka stosowana, II stopień
Tryb studiów	stacjonarne
Dane autora	Piotr Muchowski, 121716
Dane promotora	dr hab. inż. Rafał Długosz prof. PBŚ

Dane dotyczące pracy dyplomowej

Język pracy	język polski [PL]
Tytuł pracy	Badanie możliwości lub zastosowań obliczeń kwantowych w placówkach dydaktycznych lub badawczych
Opis pracy	Celem pracy będzie zbadanie możliwości zastosowania obliczeń kwantowych w przykładowych algorytmach z obszaru przetwarzania sygnałów.
Typ pracy	magisterska
Streszczenie	Obliczenia kwantowe, to obliczenia wykorzystujące efekty kwantowe, których badaniem zajmuje się fizyka kwantowa. Obliczenia kwantowe stanowią nad-zbiór obliczeń klasycznych. Oznacza to, że rozwijanie tej dziedziny może pozwolić na nowe rodzaje obliczeń. Ta praca poświęcona jest poszukiwaniu zastosowań obliczeń kwantowych w przykładowych dziedzinach takich jak cyberbezpieczeństwo, sztuczna inteligencja, symulacje procesów czy edukacja.
Słowa kluczowe	QPU, informatyka, badania, chmura, liczby, kodowanie, matematyka, symulacja

Diploma thesis record

General information

University name	Bydgoszcz University of Science and Technology
Faculty	Telecommunications, Computer Science and Electrical Engineering
Field of study	Computer science, 2nd level
Mode of study	full-time
Author's information	Piotr Muchowski, 121716
Supervisor's information	Ph.D. engineer, prof. PBŚ, Rafał Długosz

Data regarding the diploma thesis

Language of thesis	Polish [PL]
Title of thesis	Exploring the possibilities or applications of quantum computing in teaching or research settings
Description	The thesis focuses on applications of quantum computing in the area of signal processing algorithms.
Type of thesis	Master's
Abstract	Quantum computing is computing using quantum effects, which are studied by quantum physics. Quantum computing is a superset of classical computation. Consequently, development in this field could enable new types of computations. This work is devoted to finding applications of quantum computing in exemplary fields such as cybersecurity, artificial intelligence, process simulations and education.
Keywords	QPU, computer science, research, cloud, numerics, coding, math, simulation

Spis treści

Metryka pracy magisterskiej.....	2
Diploma thesis record.....	3
Spis treści.....	4
Wstęp.....	5
Metodologia.....	7
Przegląd obecnego stanu obliczeń i technologii kwantowych.....	8
Rodzaje bramek kwantowych.....	36
Prymitywy kwantowe.....	42
Algorytmy kwantowe.....	51
Praktyczne przykłady korzystania z technologii kwantowych.....	55
Wnioski.....	67
Pomysły na przyszłe zastosowania.....	68
Materiały dydaktyczne.....	72
Instrukcja do ćwiczeń.....	73
Spis literatury.....	83
Spis rysunków i wzorów.....	87
Dodatek: Zapis matematyczny obliczeń kwantowych.....	89

Wstęp

Obliczenia kwantowe, to obliczenia wykorzystujące efekty kwantowe, których badaniem zajmuje się fizyka kwantowa. Obliczenia kwantowe stanowią nad-zbiór obliczeń klasycznych. Oznacza to, że rozwijanie tej dziedziny może pozwolić na nowe rodzaje obliczeń. Ta praca poświęcona jest poszukiwaniu zastosowań obliczeń kwantowych w przykładowych dziedzinach takich jak cyberbezpieczeństwo, sztuczna inteligencja, symulacje procesów czy edukacja i badania naukowe.

W cyberbezpieczeństwie, obliczenia kwantowe mogą zarówno zagrozić poufności informacji, zabezpieczanej dotychczasowymi metodami, jak też dać nowe możliwości jej ochrony. Jeżeli dobrze zrozumiemy zjawiska kwantowe, to możliwe będzie na przykład chronienie informacji w taki sposób, aby kwantowa wiadomość uległa samozniszczeniu w przypadku nieuprawnionego dostępu, a odczytana wiadomość była całkowicie losowa, co w efekcie blokuje wyciek informacji i umożliwia łatwe wykrycie próby ingerencji w transmisję danych.

Rozwój sztucznej inteligencji zmierza w kierunku modeli z coraz większą ilością parametrów, które wymagają do działania dużej ilości zasobów komputerowych, nie mówiąc o dużym zużyciu energii. W przyszłości może się to okazać problemem hamującym dalszy rozwój sztucznej inteligencji. Dzisiejsze komputery kwantowe przypominają dawne urządzenia z XX-wieku, które wykorzystywały bardzo niewielką ilość złącz półprzewodnikowych, jednak przez kilkadziesiąt lat udało się je rozwinąć. Gdyby komputery kwantowe powtórzyły ten sukces, to mogłyby stać się alternatywną platformą dla rozwijania sztucznej inteligencji. Komputery kwantowe są interesujące dla zastosowań związanych ze sztuczną inteligencją także dlatego, że generują określone stany wyjściowe z określonym prawdopodobieństwem, podobnie jak sieć neuronowa klasyfikująca dane wejściowe do jednej z N wyjściowych kategorii. Taka zasada działania komputerów kwantowych może uprościć implementację sztucznej inteligencji na tego typu urządzeniach albo uczynić ją znacznie wydajniejszą w porównaniu z wersją elektroniczną.

Symulacja procesów zachodzących w świecie fizycznym z użyciem klasycznych komputerów polega na opisaniu tych procesów za pomocą liczb, równań i modeli. Następnie stosowane są odpowiednie algorytmy przekształcające dane, dokonujące obliczeń numerycznych z większą lub mniejszą dokładnością, oraz przesyłanie, przechowywanie i łączenie danych. Często odbywa się to sekwencyjnie, element po elemencie z całego zbioru danych. Ponadto przy złożonych modelach zawierających wiele parametrów pojawiają się dodatkowe problemy. Na końcu wyniki obliczeń są interpretowane, przy czym ze względu na kwantyzację związaną ze skończoną ilością bitów, wyniki są zawsze przybliżone. Jest to dość skomplikowane. W przypadku symulacji procesów, odbywających się poprzez bezpośrednie interakcje częstek takich jak fotony, możemy o obliczeniach myśleć bardziej jak o stworzeniu środowiska w którym badamy zachodzące procesy. Ze względu na pominięcie konwersji danych mogą one być dokładniejsze albo szybsze.

Zastosowania obliczeń kwantowych w edukacji to przede wszystkim poszukiwanie nowych rodzajów obliczeń, możliwych do wykonania na sprzęcie kwantowym, badanie właściwości,

opracowywanie algorytmów kwantowych implementujących istniejące lub nowe operacje na danych, oraz wszystko co nie mieści się w pozostałych kategoriach, w tym działalność naukowa.

Podsumowując, technologia kwantowa jest warta poznania i rozwijania. Stąd wybrany przez mnie temat pracy magisterskiej. Postawione pytanie badawcze to “Czy i jak można wykorzystać obliczenia kwantowe w placówkach badawczych i dydaktycznych?”

Metodologia

Tematem pracy jest znalezienie odpowiedzi na pytanie czy i w jaki sposób technologie kwantowe mogą być interesujące dziedziną badań w placówkach dydaktycznych i naukowych. Aby odpowiedzieć na to pytanie, konieczne jest najpierw poznanie i zaprezentowanie aktualnego stanu technologii w 2024 roku. Aby to zrobić, dokonano przeglądu publikacji naukowych w bazach danych dostępnych na Politechnice Bydgoskiej takich jak IEEE. Ponieważ dziedzina ta rozwija się szybko, dlatego brano pod uwagę publikacje z roku 2022 i późniejsze, z pewnymi nielicznymi wyjątkami. Skupiono się na poznaniu aktualnego stanu obliczeń kwantowych, komunikacji kwantowej i dziedzin powiązanych. Ta praca zawiera streszczenie najważniejszej wiedzy uzyskanej z tych publikacji, pomagając w zrozumieniu dalszej części badań opisanych w kolejnych rozdziałach. W sumie przeanalizowano około 70 dokumentów naukowych oraz pewną liczbę pomocniczych źródeł takich jak witryny internetowe.

W dalszej części pracy skupiono się na obliczeniach kwantowych w modelu bramkowym, prezentując podstawowe struktury od najniższej warstwy abstrakcji do najwyższej. Wymieniono zatem podstawowe bramki używane w obliczeniach, następnie zaprezentowano popularne prymitywy, po nich zaś omówiono algorytmy korzystające z opisanych prymitywów do realizacji bardziej złożonych operacji takich jak optymalizacja czy faktoryzacja liczb.

W kolejnej części opisano i zaprezentowano wybrane algorytmy, po czym zbadano możliwości uruchomienia ich na symulatorach i rzeczywistych komputerach kwantowych dostępnych publicznie lub dzięki dostępowi do zasobów akademickich. Ponadto podjęto próbę stworzenia własnego algorytmu realizującego rozwiązanie wybranego problemu.

Podsumowując pracę, streszczono obserwacje, spostrzeżenia i wyniki badań, będące podstawą do wyciągnięcia wniosków i oszacowania perspektyw rozwoju technologii, a także zasugerowania pomysłów na wykorzystanie obliczeń kwantowych w przyszłości.

Uzupełnieniem pracy jest propozycja ćwiczeń laboratoryjnych do wykonania przez uczniów i studentów w ramach poznawania technologii kwantowych. Skupiono się na tym aby ćwiczenia były łatwe i nie wymagały głębokiego zrozumienia tematu, a stanowiły raczej inspirację do własnych przemyśleń i pobudzenia twórczej postawy wobec zmieniającej się technologii.

Przegląd obecnego stanu obliczeń i technologii kwantowych

Obecny stan rozwoju technologii kwantowej na początku 2024 roku to era NISQ (Noisy intermediate-scale quantum era). Oznacza ona dostępność prostych procesorów QPU złożonych z mniej niż 1000 kubitów, podatnych na szумy i zakłócenia. Choć istnieją już algorytmy korekcji pozwalające na emulowanie wirtualnych bezszumnych kubitów, to zbyt mała ilość fizycznych kubitów ogranicza ich stosowanie. W erze NISQ, tradycyjne algorytmy szyfrowania takie jak RSA, są uważane za bezpieczne, jednak przestaną być w momencie oddania do użytku komputerów kwantowych nowej generacji, posiadających $>10^4$ kubitów i niezawodną korekcję błędów. Będzie to oznaczało koniec ery NISQ i rozpoczęcie erę Post-kwantową. Prawdopodobnie nastąpi to w ciągu kilku, maksymalnie kilkunastu lat.

Kwantowe technologie to nie tylko obliczenia kwantowe. Możemy wyodrębnić technologiczne gałęzie i powiązane dziedziny, jednocześnie wyjaśniając czym są:

Obliczenia kwantowe (quantum computing, QC) - obszar obejmujący algorytmy i protokoły wykorzystujące zjawiska kwantowe takie jak superpozycja czy splątanie do przetwarzania informacji lub rozwiązywania problemów. W porównaniu do obliczeń opartych o logikę binarną, stanowią bardziej ogólną formę obliczeń. Nowe rodzaje obliczeń możliwe do przeprowadzenia w dziedzinie kwantowej są przedmiotem badań.

Komunikacja kwantowa (quantum networks, QN) - dziedzina zajmująca się przesyłaniem informacji kwantowej na odległość, łączeniem komputerów kwantowych w klastry, protokołami komunikacyjnymi czy badaniem zjawisk występujących na łączach. Do komunikacji wykorzystuje się światłowody, a na większe odległości łącza satelitarne. Takie sieci już istnieją.

Kwantowa dystrybucja kluczy (QKD) - zajmuje się metodami uzgadniania kluczy wykorzystywanych na przykład w szybkich i bezpiecznych sieciach łączących infrastrukturę 5G. Ze względu na właściwości kwantowej informacji i ograniczenia obecnej technologii, metody te zapewniają wysoki poziom bezpieczeństwa wymienianych kluczy i stosunkowo niewielką przepustowość ograniczającą częstotliwość rotacji kluczy.

Algorytmy kwantowe - złożone z bramek kwantowych operacje na informacji kwantowej realizujące na przykład wyszukiwanie, transformację Fouriera albo faktoryzację liczb.

Bramki kwantowe - standardowe operacje na stanach kwantowych jednego lub więcej kubitów, opisane za pomocą algebry liniowej.

Symulatory obwodów kwantowych - oprogramowanie działające na tradycyjnych komputerach, symulujące obwody złożone z bramek kwantowych. Pozwala na badanie i testowanie algorytmów kwantowych złożonych z nie więcej niż 20-30 kubitów.

Fotonika - generowanie, detekcja i manipulacja pojedynczymi fotonami, wykorzystywane na przykład do przenoszenia informacji kwantowej.

Wyżarzanie kwantowe - wykorzystująca kwantowe efekty, metoda poszukiwania optymalnych rozwiązań problemów które są zbyt złożone dla tradycyjnych algorytmów obliczeniowych (na przykład z powodu zbyt dużej ilości jednocześnie poszukiwanych parametrów)

Kryptografia - zajmuje się metodami zabezpieczania poufności informacji. W odniesieniu do technologii kwantowych, zajmuje się też badaniem nowych metod z wykorzystaniem tych technologii, a także badaniem bezpieczeństwa dotychczasowych metod wobec dostępności technologii kwantowych.

Metody kodowania informacji - sposoby zapisu informacji w sposób zapewniający takie cechy jak efektywność, odporność na błędy i zakłócenia, trwałość, skalowalność czy niezawodność.

Informatyka kwantowa - zajmuje się integracją technologii kwantowej z tradycyjną informatyką, standaryzacją interfejsów i protokołów komunikacyjnych, projektowaniem sprzętu wykorzystującego technologie kwantowe, wdrażaniem technologii kwantowych w przyszłych systemach.

5G i 6G - Zestaw technologii będących podstawą działania dzisiejszego i przyszłego społeczeństwa. Technologie kwantowe stanowią ważny filar umożliwiający przyszły i znaczący rozwój tych technologii.

QInternet - ogólnoswiatowa sieć nowej generacji oparta o technologie kwantowe, umożliwiająca wymianę informacji, w tym informacji kwantowej.

Symulacje kwantowe - Symulacje procesów, cząstek chemicznych lub innych modeli z użyciem narzędzi wykorzystujących do działania technologie kwantowe.

Metaverse - Świat abstrakcyjny będący połączeniem świata rzeczywistego oraz pewnej ilości światów wirtualnych, będący miejscem doświadczenia, interakcji i życia dla przyszłych społeczeństw.

Obliczenia kwantowe tolerujące błędy - odpowiedź na problem zaszumienia fizycznych kubitów, prowadzący do niedokładnych wyników. Polegają na stworzeniu koncepcji wirtualnych, bez-szumnych kubitów, symulowanych poprzez pewną (kilukrotnie większą) ilość rzeczywistych kubitów oraz algorytmu korekcji błędów.

Kwantowe uczenie maszynowe - Technologia znana z tradycyjnej informatyki, wykorzystująca obliczenia kwantowe, kwantowe sieci neuronowe oraz algorytmy kwantowe. Bazując na technologii kwantowej, może wyłapać więcej wzorców w danych.

Cyberbezpieczeństwo - dziedzina zajmująca się ochroną informacji, procesów czy systemów. Powiązanie z technologiami kwantowymi skupia się głównie na identyfikowaniu i przeciwdziałaniu zagrożeniom jakie technologie kwantowe mogą wnieść do

dotychczasowego świata, a także na badaniu i wykorzystywaniu nowych technologii kwantowych do poprawy bezpieczeństwa.

Przyjrzyjmy się teraz bliżej wymienionym dziedzinom.

Obliczenia kwantowe

W publikacji [1] przedstawiono historię rozwoju obliczeń kwantowych i dziedzin powiązanych. W latach 1990-2000 były to głównie obliczenia kwantowe i kwantowe maszyny Turinga. Prace były teoretyczne, a eksperymentowano z pojedynczymi kubitami. W latach 2000-2010 nastąpił gwałtowny wzrost ilości publikacji, pojawiły się też nowe zagadnienia takie jak informacja kwantowa i algorytmy, oraz powstały pierwsze języki programowania obliczeń kwantowych. Ponadto wśród ówczesnych popularnych słów kluczowych możemy znaleźć takie terminy jak *quantum memory*, *adiabatic QC*, *topological QC*, *quantum walk*, *and quantum wire*. Świadczą one o przejściu od teorii do prototypów urządzeń i rozwiązywaniu problemów z implementacją obliczeń kwantowych, oraz powstaniu pierwszych wielo-kubitowych urządzeń i algorytmów. W latach 2010-2020 obserwujemy dalszy wzrost aktywności w dziedzinie obliczeń kwantowych. Najważniejszymi zagadnieniami łączącymi badania są algorytmy i informacja kwantowa. Zauważamy też przejście od sprzętu w stronę oprogramowania, co oznacza że rozwój odbywa się na coraz wyższych warstwach abstrakcji. Powstaje ekosystem technologii, zasobów czy wiedzy koniecznych do zorganizowanego rozwoju technologii. W tym czasie powstają pierwsze publicznie dostępne komercyjne lub edukacyjne usługi obliczeń kwantowych w chmurze, czy framework programistyczne oraz kod programów kwantowych. We wspomnianych latach często pojawiają się takie zagadnienia jak bramki, splątanie kwantowe, *quantum dot*, optyka kwantowa, komunikacja kwantowa, kody korekcyjne, kwantowe przetwarzanie, kwantowe obwody i mechanika kwantowa. Nowymi gałęziami są też kwantowe wyżarzanie, kwantowe przetwarzanie i reprezentowanie obrazów, kwantowe uczenie maszynowe, *blind QC*, *delegated QC*, *quantum electrodynamics*, *quantum sensors*, *quantum internet*.

2011-2020		
Label	Degree	Centrality
Quantum Image Processing	52	
Quantum Machine Learning	31	
Blind Quantum Computing	23	
Delegated Quantum Computing	16	
Circuit Quantum Electrodynamics	14	
Quantum Image Representation	13	
Quantum Internet	13	
Quantum Processing Unit	13	
Quantum Sensor	13	
Noisy Intermediate-scale Quantum	12	

(Rys. 1) Kierunki rozwoju technologii kwantowych w latach '10, źródło [1]

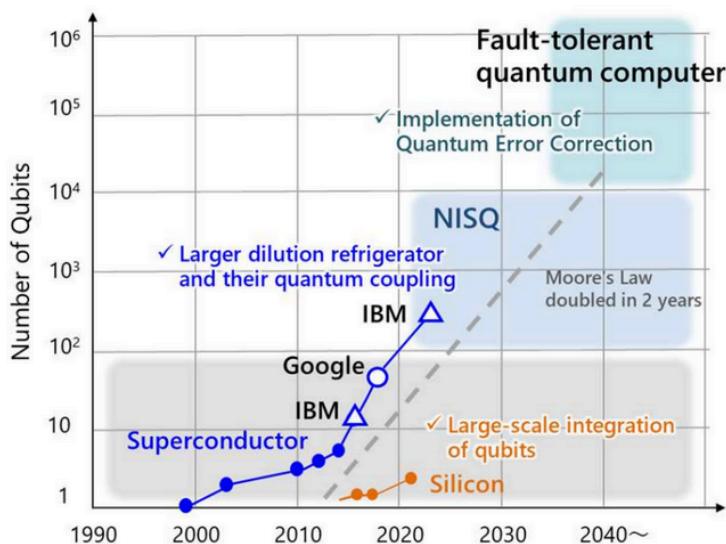
Według [1], technologia kwantowa pozwala tworzyć nowe rozwiązania chroniące prywatność. Można na przykład kierować do bazy danych zapytania w taki sposób aby uzyskać interesującą odpowiedź bez ujawniania treści zapytania albo informacji będącej granicami poszukiwanego zakresu danych.

Podsumowując [1], początkowo opracowywano prototypy, natomiast ostatnie lata, to prace na tym aby już istniejące rozwiązania działały lepiej. Według autorów [1], główne obszary które się rozwiną w najbliższych latach to informacja kwantowa, wykrywanie kwantowe oraz obrazowanie, komunikacja kwantowa i kryptografia oraz oczywiście obliczenia kwantowe.

Praca naukowa [2] skupia się na trzech obszarach dotyczących aktualnego rozwoju obliczeń kwantowych. Tematem pracy jest struktura rynku, odpowiedzialność za przyszłe zmiany spowodowane dostępnością obliczeń kwantowych oraz rozwój technologiczny obwodów półprzewodnikowych.

Pełne wdrożenie obliczeń kwantowych do życia codziennego to długotrwały proces. Choć wiele zainwestowano już w tą technologię, to społeczeństwo odczuje korzyści dopiero po jakimś czasie. Aby wypełnić tą lukę, badacze [2] opracowali w 2015 roku, inspirowane kwantowo wyżarzanie CMOS. Polega ono na zamianie kombinatorycznego problemu optymalizacyjnego na model Isinga, tak jak to się dzieje w wyżarzaniu kwantowym. Dalsza praca z modelem odbywa za pomocą tradycyjnego sprzętu ASIC, FPGA lub GPU, wykorzystując obliczenia równoległe, co według autorów przynosi obiecujące rezultaty.

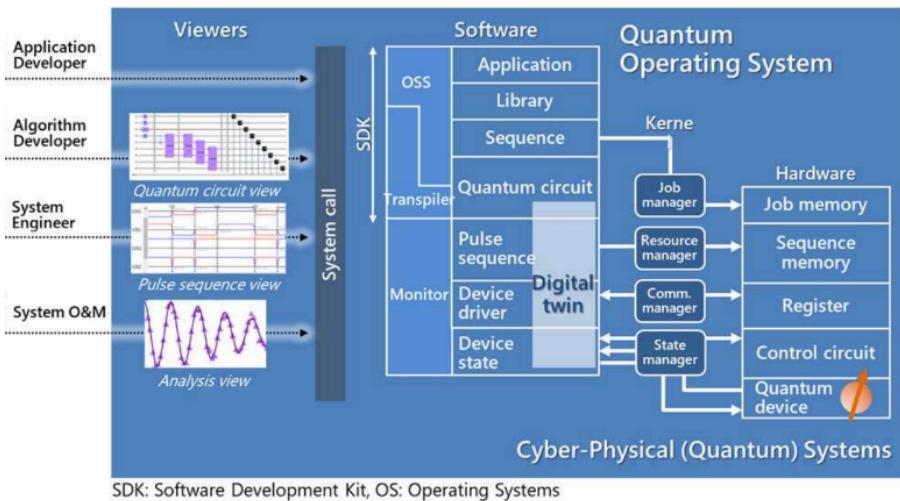
Obliczenia kwantowe i komputery kwantowe potrafią dokładnie symulować zachowanie cząstek. Pozwala to na badania nad nowymi substancjami, lekami czy enzymami bez dotychczasowych ograniczeń w ich maksymalnym rozmiarze. To jeden z wielu przykładów możliwości jakie dają obliczenia kwantowe. Z drugiej strony część społeczeństwa traktuje komputery kwantowe tylko jako "lepszą wersję" zwykłych komputerów, która robi to samo tylko szybciej. W rzeczywistości, epoka post-kwantowa otworzy przed ludzkością nowy rozdział w rozwoju, pozwalając realizować pomysły i marzenia, które dotychczas uważano za niemożliwe.



(Rys. 2) Postęp technologiczny i epoki komputerów kwantowych, źródło [2]

Dzisiejszym problemem jest skonstruowanie komputera kwantowego tolerującego błędy (ang. *FTQC*), posiadającego odpowiednio dużą liczbę kubitów o odpowiednio wysokiej stabilności, wyposażonego w skuteczne algorytmy korekcji błędów. Próbowano już z nadprzewodnikami, uwięzionymi jonami, *quantum dots*, atomami i fotonami, ale celu jeszcze nie osiągnięto. Trwają prace zarówno nad podniesieniem jakości, czyli stabilności kubitów, oraz ilości połączonych kubitów na jednym chipie, czyli zdolności do rozwiązywania bardziej złożonych problemów czy uruchamiania bardziej skomplikowanych algorytmów. Dodatkową komplikacją jest jeszcze konieczność dodania mechanizmów korekcji błędów. Badacze [2], w odróżnieniu od pozostałych, skupiają się najpierw na zwiększeniu ilości kubitów wykorzystując technologie scalonych układów krzemowych. Na początku tworzą dużą macierz z kubitami, a następnie pozwalają kubitom przemieszczać się z każdą operacją. W ten sposób, odpowiednio kierując ruchem, można ominąć ograniczenia ilości odczytywalnych kubitów na chipie, jednocześnie redukując przesłuchy będące głównym problemem w operacjach na kubitach. Opracowano też proces QCMOS, co pozwala umieszczać obwody kontrolne na tym samym chipie co macierz kubitów.

Quantum Operating System



(Rys. 3) Architektura kwantowego systemu operacyjnego, źródło [2]

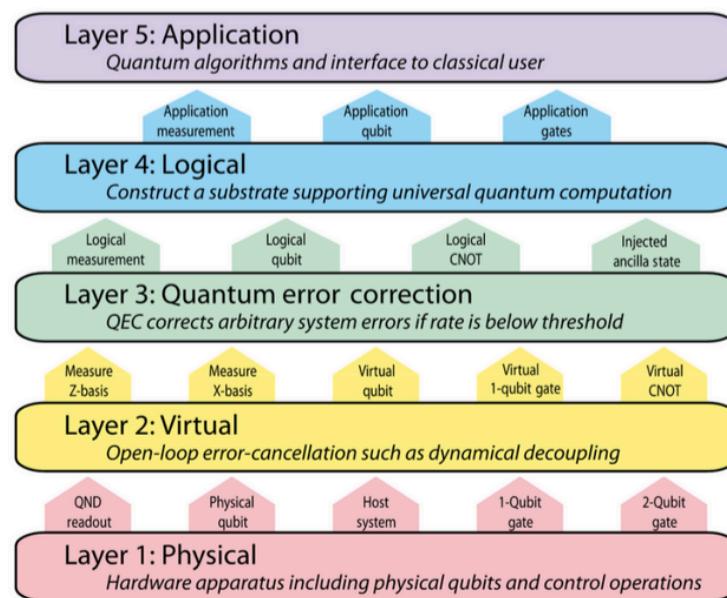
Kolejną nowością jest, już teraz, rozpoczęcie definiowania architektury i systemu operacyjnego dla komputerów kwantowych, oraz oddzielenie jej od fizycznej implementacji która będzie się mogła łatwo zmieniać. Ważne jest też postrzeganie komputera kwantowego jako cyber-fizycznego systemu kontrolującego dużą ilość kubitów i peryferii. Innymi słowy, [2], idą już w wyższe logiczne warstwy abstrakcji, mimo niedoskonałości warstwy fizycznej. Zamierzają w 2027 roku oddać do testów krzemowy komputer kwantowy dużej skali działający w chmurze.

Z [3] dowiadujemy się więcej o erze NISQ, wspomnianej na początku tego rozdziału. W tej erze dostępne są średniej skali zaszumione komputery kwantowe. Opracowano też takie algorytmy jak algorytm Shor'a służący do faktoryzacji liczb, czy algorytm wyszukiwania Grovera. Znane są już zastosowania NISQ w optymalizacji, kryptografii, uczeniu maszynowym i inżynierii materiałowej.

Obliczenia kwantowe różnią się od obliczeń klasycznych pewnymi kluczowymi właściwościami. Po pierwsze, kubity mogą przyjąć jednocześnie stan 0 i 1 w dowolnej wzajemnej proporcji, co określa się stanem superpozycji. Gdy wartość kubitu jest mierzona, superpozycja znika i otrzymujemy zwykły bit informacji, o wartości 0 lub 1, z prawdopodobieństwem rezultatu określonym kwadratem tzw. amplitudy prawdopodobieństwa danego stanu, będącego składnikiem superpozycji. Dokładniej wyjaśniono to w rozdziale poświęconym rodzajom bramek kwantowych. Kolejną właściwością obliczeń kwantowych jest splatanie. Oznacza ono oddziaływanie kubitów na siebie. Odczytanie jednego ze splatanych kubitów natychmiast zmienia stan drugiego. Algorytmy kwantowe wykorzystują te efekty do uzyskania odpowiedzi na rozwiązywany problem. W odróżnieniu od klasycznych obliczeń, tutaj odpowiedź nie jest deterministyczna, a probabilistyczna. Oznacza to, że nie możemy przewidzieć wyniku obliczeń, a jedynie określić prawdopodobieństwo wystąpienia określonych stanów wyjściowych.

Zaszumienie oznacza, że komputery kwantowe ery NISQ, są podatne na takie zjawiska jak dekoherencja wywołana niepożdanym oddziaływaniem środowiska na kubity, prowadząca do utraty informacji. Inne przyczyny utraty dokładności obliczeń to szумy termiczne, pola

elektromagnetyczne, czy skazy materiałowe przy produkcji komponentów. Utrzymanie kubitów we właściwym stanie przez cały czas trwania obliczeń jest trudne. Pewnym rozwiązaniem jest podzielenie problemu na małe podproblemy i współpraca podzespołów kwantowych i klasycznych. Ponadto możemy stosować model warstwowy, który pozwala stosować wysokopoziomowe algorytmy tworzone przez programistów i tłumaczyć abstrakcyjne operacje na fizyczne niskopoziomowe działania na kubitach. Jedną z warstw jest warstwa korekcji błędów, przez co algorytm kwantowy operuje na logicznych, idealnych kubitach, zamiast na zaszumionych, fizycznych. Warstwowy stos pozwala też oddzielić tworzenie algorytmów kwantowych, od operacji fizycznych. Innymi słowy, użytkownik czy programista nie musi być ekspertem w dziedzinie fizyki kwantowej, czy znać szczegółów technologicznych implementacji, a tworzony kod może być uruchamiany na komputerach o różnej zasadzie działania. Jest to ważne ze względu na ciągły i szybki rozwój warstwy sprzętowej.



(Rys. 4) Warstwowy model obliczeń kwantowych, źródło [3]

Algorytmy korekcji błędów (QEC) pomagają utrzymać informację w niezmienionym stanie. Jest to konieczne, ponieważ obliczenia kwantowe polegają na precyzyjnych modyfikacjach stanów kwantowych. Możemy przechowywać informację w wielu kopiiach, oraz zarezerwować dodatkowe kubity dla kontroli prawidłowości informacji i naprawy powstających błędów. Możemy stosować zero-szumną ekstrapolację albo destylację magicznych stanów. Inna metoda to akceptacja błędów i stosowanie takich algorytmów dla których sporadyczne błędy nie dyskwalifikują wyniku obliczeń. Podsumowując, korekcja błędów to kluczowa gałąź, bez której postęp technologiczny nie będzie łatwy.

Można wyodrębnić pewne rodzaje platform do obliczeń kwantowych. Kwantowe procesory bazujące na bramkach, programowalne przez użytkownika, realizują obliczenia przez precyzyjnie regulowane bramki operujące na kubitach. Takie systemy buduje IBM i Rigetti. Nadają się do badania algorytmów i kodów korekcyjnych. Wyżarzanie to inny rodzaj obliczeń kwantowych, nadający się szczególnie do rozwiązywania problemów optymalizacyjnych. Działają na zasadzie poszukiwania idealnego rozwiązania bazując na stanach

energetycznych i entropii. Dalej mamy analogowe symulatory kwantowe. Działają na zasadzie symulacji zjawisk kwantowych, wykorzystując takowe. Jest to de facto bardziej rzeczywisty eksperyment niż symulacja. Mogą być używane przede wszystkim w badaniach naukowych albo symulacjach związków chemicznych. Kolejna klasa urządzeń to sprzęt oparty o rozwiązańa fotoniczne. Tutaj kubity reprezentowane są przez fotony, co naturalnie predestynuje ten rodzaj technologii nie tylko do obliczeń, ale również do komunikacji. Wreszcie ważnym elementem ekosystemu jest dostęp chmurowy, inaczej możliwość korzystania z obliczeń kwantowych w formie usługi świadczonej zdalnie. Pozwala prowadzić badania naukowcom, których ośrodek nie posiada własnego sprzętu, porównywać algorytmy na różnych platformach, czy lepiej wykorzystywać kosztowne instalacje. Jest też ważnym elementem pozwalającym technologiom kwantowym opuścić laboratoria i znaleźć zastosowania w życiu codziennym. Takie usługi zapewniają dostawcy tradycyjnych usług chmurowych, takich jak na przykład Microsoft czy IBM.

Przechodząc do zastosowań obliczeń kwantowych, wyróżniamy cyberbezpieczeństwo i kryptografię. Tutaj obliczenia kwantowe to miecz obosieczny, gdyż dając możliwości tworzenia nowych technologii zabezpieczających dane, jednocześnie generują zagrożenia dla starych algorytmów kryptograficznych, które powstały w innych czasach. W zasadzie dbanie o aktualność zabezpieczeń to podstawowa praktyka w świecie cyberbezpieczeństwa, jednak kwantowa rewolucja technologiczna wymusza znaczne zintensyfikowanie działań w tym zakresie, co aktualnie się dzieje.

Dalej mamy optymalizację i jej szerokie zastosowania, dzięki dotychczasowym licznym implementacjom z wykorzystaniem klasycznych komputerów. Obliczenia kwantowe mogą znacznie poprawić wydajność tych algorytmów, lub przesunąć granice postępu w tych dziedzinach. Przykładem mogą być kombinatoryczne problemy związane z optymalną trasą przejazdu.

W dziedzinie sztucznej inteligencji, efekty kwantowe pozwalają na szybsze trenowanie modeli i prawdopodobnie inne nieznane jeszcze ulepszenia. Ponadto wiele modeli AI operuje na prawdopodobieństwie, a to fundamentalna cecha obliczeń kwantowych, co powinno wzbudzić zainteresowanie naukowców z obu dziedzin.

Symulacje kwantowe cząstek chemicznych oraz inżynieria materiałowa stanowią obecnie obszary o ogromnym zapotrzebowaniu na moc obliczeniową, przekraczającą możliwości klasycznych komputerów. Obliczenia kwantowe mogą uczynić badania efektywniejszymi. Te symulacje często także badają zjawiska występujące na poziomie kwantowym. Zamiast opisywać problem matematycznie i wykonywać skomplikowane obliczenia na wielu zmiennych, można przetłumaczyć problem symulacji cząstek, na eksperyment możliwy do realizacji w procesorze kwantowym. Badając zjawiska kwantowe występujące w komputerze kwantowym, do czego mamy już bogaty zestaw narzędzi, technologii i algorytmów, możemy dowiedzieć się czegoś o fizycznych cząstkach, które symulujemy. Innymi słowy zamiast wielu obliczeń, możemy zamodelować cząstki w komputerze. Istnieją analogowe symulatory kwantowe, które nie służą do obliczeń a właśnie do takich eksperymentów.

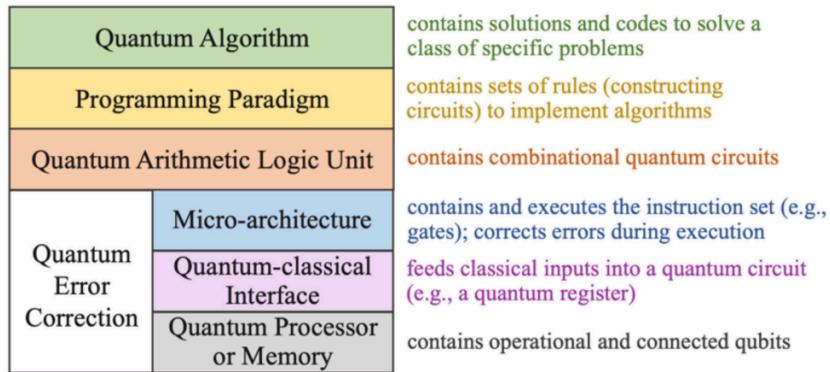
Obliczenia kwantowe mogą też pomóc podejmować decyzje, także przy niepełnej wiedzy, na przykład odpowiadać na pytania, których odpowiedź zależy od czynników o nieznanych wartościach, biorąc pod uwagę wszystkie możliwe kombinacje jednocześnie, dzięki paralelizmowi kwantowemu.

Dokument [4], m.in. wprowadza więcej teorii, na której opierają się obliczenia kwantowe. Można wymienić takie kluczowe efekty kwantowe jak superpozycja, interferencja, splątanie czy dekoherencja. Ważną właściwością stanów kwantowych jest niemożliwość ich sklonowania (kopiowania).

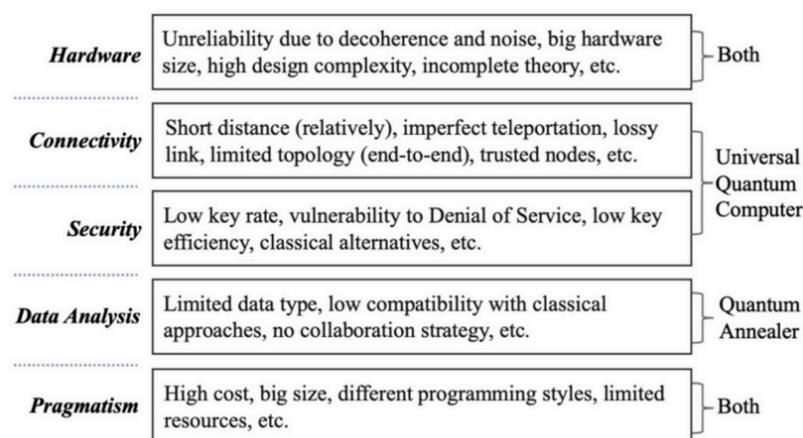
Same stany kwantowe mogą być reprezentowane przez różne rodzaje nośników. Możemy wyróżnić kubity fotoniczne, gdzie stan reprezentowany jest przez polaryzację, częstotliwość lub tzw mod przestrzenny. Efekty kwantowe występują też w nadprzewodnikach. Elektrony na orbitach atomów posiadają spiny, które można mierzyć i także podlegają prawom fizyki kwantowej, podobnie jak jony czy atomy.

Obecne komputery kwantowe wykorzystują jedną z wymienionych wyżej technologii. Możliwe, że w przyszłości zostaną odkryte nowe. Ciekawym pomysłem jest łączenie różnych nośników w jednym komputerze. Można używać stacjonarnych kubitów jako podstawowych komponentów, łącząc je w większe zestawy przy użyciu fotonów. Mogłoby to pozwolić na zwiększenie całkowitej ilości kubitów w systemie, tworząc duże sieci z małych komputerów kwantowych, na wzór klasycznych chmur. W odróżnieniu od tych ostatnich, gdzie moc obliczeniowa rośnie liniowo z ilością urządzeń w chmurze, tutaj, wykorzystując efekty kwantowe możemy stworzyć jeden połączony superkomputer, gdzie całkowita ilość kubitów rośnie liniowo ale moc obliczeniowa wykładniczo [5]. Innym pomysłem wykorzystania różnych technologii reprezentacji kubitów może być komputer złożony ze wszystkich dostępnych rodzajów kubitów, gdzie oprogramowanie działające na tym komputerze będzie optymalnie dobierać rodzaj kubitów do rozwiązywanego problemu, rozbijając wysokopoziomowy algorytm na podproblemy i zarządzając przydziałem zasobów tak, aby najefektywniej wykorzystać zalety różnych technologii. Fotoniczne i nadprzewodzące kubity są łatwe do przenoszenia ale bardziej podatne na szumy. Kubity oparte na atomach, jonach i spinie elektronu są stabilniejsze, ale trudno łączyć je w duże zestawy.

Podsumowując, komputery kwantowe dzielą się na cyfrowe, oparte o bramki, oraz analogowe oparte na wyżarzaniu. Badane są możliwości stworzenia komputerów cyfrowo analogowych, oraz hybrydowych urządzeń klasyczno-kwantowych. Przykładem analogowych komputerów są urządzenia firmy D-Wave. Cyfrowe urządzenia dostarczają natomiast Google, Microsoft czy IBM.



(Rys. 5) Architektura komputera kwantowego, źródło [4]



(Rys.6) Problemy obliczeń kwantowych, źródło [4]

Główym problemem w obecnych komputerach kwantowych jest szum i niepożądana interakcja z otoczeniem. Powoduje on zanikanie informacji kwantowej, podobnie jak tarcie w mechanice powoduje ustanie ruchu. Z jednej strony powinniśmy zatem maksymalnie odizolować procesor od środowiska zewnętrznego, ale z drugiej strony jakiś kontakt powinien pozostać aby móc ingerować w stany kubitów i je mierzyć. Ponieważ suma kwadratów amplitud wszystkich stanów układu kwantowego jest równa 1, to każda zmiana dowolnego stanu przenosi się na pozostałe. Podobnie jest z fazą. W rezultacie stany stają się niedokładne, a błąd szybko kumuluje się w trakcie obliczeń, co prowadzi do losowych, niezgodnych z teorią wyników. Widać to gdy porównamy wyniki rzeczywistych obliczeń kwantowych obarczonych szumem, z symulacjami które szumu nie uwzględniają. Algorytmy eliminacji tych błędów zwykle wymagają sporej nadmiarowości, co przekłada się na wysokie wymagania odnośnie sprzętu dotyczące ilości wykorzystywanych kubitów. Oczyszczanie stanów wielo-kubitowych dobrze działa tylko na stanach dobrze znanych, takich jak baza Bella, trudniejsze okazuje się oczyszczanie kubitów splątanych, znajdujących się w nietypowych stanach.

Inny problem to rozmiary, czy zużycie energii. Chociaż sam procesor kwantowy jest wielkości monety, to sprzęt towarzyszący, utrzymujący temperaturę bliską zera bezwzględnego jest duży. Według [4], obecny koszt jednego kubitu to 10 tyś \$. Aby obliczenia kwantowe stały się powszechnie, potrzebujemy komputerów złożonych z milionów kubitów i znacznego obniżenia kosztów. Nie dysponujemy jeszcze technologiami, które

pozwoliły by budować tanie i złożone komputery kwantowe pracujące w temperaturze pokojowej.

Następne wyzwanie wiąże się z naturą stanów kwantowych. Nie można ich kopiować czy klonować. Jeżeli informację utracimy, trudno ją odzyskać. To wymusza inny styl programowania, czy zarządzania przepływem danych. Sposoby zwiększenia niezawodności takie jak nadmiarowość, stosowane z powodzeniem w klasycznych obliczeniach, nie zdają egzaminu w przypadku obliczeń kwantowych. Nie ułatwia to integracji systemów klasycznych i kwantowych. W przypadku klasycznego tworzenia oprogramowania, na przykład w języku python, możliwe jest powtarzanie nieudanych instrukcji, zatrzymywanie programu, podglądanie jego stanu, jego zmiennych, czy wprowadzanie ulepszeń w trakcie działania programu. W kwantowych symulatorach jest to również możliwe, jednak obecne symulatory mają ograniczone możliwości skalowania. Natomiast obliczenia kwantowe w ogóle nie pozwalają debugować kodu w sposób znany i stosowany dotychczas. Oznacza to konieczność opracowania nowych koncepcji i praktyk związanych z tworzeniem oprogramowania do komputerów kwantowych.

Innym nie mniej kluczowym problemem jest niekompletność wiedzy o zjawiskach kwantowych. Budujemy prototypy i przeprowadzamy eksperymenty z obliczeniami, ponieważ to działa i daje interesujące wyniki, a nawet poznaliśmy już sprytne sztuczki pozwalające rozwiązywać pewne problemy szybciej i efektywniej. Jednakże, splątanie kwantowe jest pełne niejasności i w zasadzie snujemy hipotezy w jaki sposób ono działa. Jeżeli weźmiemy teraz pod uwagę, że traktujemy kryptografię kwantową jako ultra bezpieczną technologię, mimo iż nie potrafimy dokładnie zrozumieć wszystkich procesów z których ona korzysta, to może się w przyszłości okazać, że dalsze odkrycia naukowe ujawnią luki w bezpieczeństwie protokołów i systemów kwantowych.

Kody korekcyjne stosowane w obliczeniach kwantowych to przede wszystkim kody powtórzeniowe. Informacja w kilku kopiach jest mniej podatna na utratę. W razie niepożądanych zmian, można przeprowadzić głosowanie większościowe. Okazuje się, że aby zabezpieczyć 1 logiczny kubit, należy użyć co najmniej 5 fizycznych kubitów. Ponadto, te dodatkowe kubity są mocno skorelowane ze sobą, zatem będą się zachowywać podobnie. Można więc mierzyć dodatkowe kubity, aby wykryć ewentualne symptomy błędów. Aby jeszcze bardziej zmniejszyć prawdopodobieństwo błędów, można zwiększać ilość dodatkowych kubitów, albo tworzyć wielopoziomowe kody korekcyjne. Z drugiej strony, ze względu na brak możliwości klonowania informacji, nie jest to łatwe. Dodatkowo pojawia się problem z bezpieczeństwem informacji, gdyż można przechwytywać wybrane kubity podglądając ich zawartość, a jednocześnie nie zakłócając działającego algorytmu.

Name	Description	Ref.
Shor code	Shor was the first to discover a method of formulating multiple physical qubits to represent one logical qubit. The Shor code encodes one logical qubit with nine physical qubits, which can correct arbitrary errors in a single qubit.	[175]
Steane code	Steane accomplished the same thing as the Shor code with seven physical qubits.	[180]
5-qubit codes	This class of codes can do the same with five physical qubits. It has been shown that five is the minimum.	[171]
CSS codes	CSS codes are the generalization of the Steane code, named after the authors. They are particular types of stabilizer codes.	[180] [170]
Stabilizer codes	All methods that use multiple physical qubits to represent one logical qubit are called Stabilizer Codes. This includes the above four codes.	[181]
Bacon-Shor codes	Bacon-Shor codes are square-lattice-based 2-dimensional codes with two parameters of the lattice.	[182]
Surface code and Color code	These topology-based stabilizer codes have the potential for large systems of well-protected logical qubits.	[166] [183]
Bosonic codes	Bosonic codes are hardware-efficient alternatives to stabilizer codes. They use multi-photon states of superconducting cavities to encode information.	[184]

(Rys. 7) Różne kody korekcyjne stosowane w komputerach kwantowych, źródło [4]

Aby zapisać informację do wielu kubitów jednocześnie, powinny one być wcześniej ze sobą splątane. Z drugiej strony, splątane kubity oddziałują na siebie, więc odczytując jeden z nich wpływamy na stan pozostałych, co jest przeciwieństwem redundancji. Także ewentualne zakłócenia propoagują się między kubitami. Wreszcie, kubity nie reprezentują 2 stanów binarnych. Kwantowe kody korekcyjne są więc trudniejsze do konstruowania od ich klasycznych odpowiedników, jednak udowodniono że stworzenie komputera kwantowego odpornego na błędy jest teoretycznie możliwe.

Z dokumentu [4] (wyd. drugi kwartał 2023) dowiadujemy się, że firma IBM opracowuje komputer oparty na bramkach, których w 2023 roku ma być 1000, a w 2025 aż 4000. Z kolei D-Wave to komputery typu annealer, gdzie kubitów jest zwykle więcej, ponieważ ich podatność na szum jest mniejsza. Podsumowując, trwa zacięta walka o ilość kubitów i ich stabilność, osobno w dwóch wymienionych przed chwilą architekturach. Ponadto D-Wave myśli o połączeniu obu architektur w jedną.

	Quantum Services
<i>AI</i> :	CQC, QbitLogic, QC Ware, Qindom, Xanadu, etc.
<i>Cloud Computing</i> :	Amazon, D-Wave, Google, Microsoft, IonQ, Xanadu, etc.
<i>Encryption</i> :	EYL, ID Quantique, MagiQ, QNu Labs, TAQBit Labs, etc.
	Quantum Software
<i>Quantum Solutions</i> :	1QBit, Anyon, Benchmark, QC Ware, QuSoft, Zapata, etc.
<i>Development Toolkits</i> :	Alibaba, Amazon, Google, IBM, Microsoft, QCI, Xanadu, etc.
<i>Operating Systems</i> :	Q-CTRL, CQC, etc.
	Quantum Hardware
<i>Universal</i> :	Amazon, Google, Honeywell, IBM, Intel, IonQ, Microsoft, Rigetti, etc.
<i>Annealers</i> :	D-Wave, etc.
<i>Chips</i> :	BraneCell, IonQ, Optalysys, QCI, Qutools, Rigetti, Turing, Xanadu, etc.

(Rys. 8) Dostawcy technologii, źródło [4]

Na rysunku powyżej przedstawiono dostawców rozwiązań technologicznych i usług (stan na 2023r.). Na kolejnym rysunku znajduje się zestawienie typów komputerów kwantowych. 2 pierwsze były już często wymieniane. Ostatnie pozycja to nowatorska technologia użycia enionów, czyli kwazicząstek, których istnienie potwierdzono doświadczalnie w 2020 roku.

Name	Description
Logic Gate Model (Universal Quantum Computing)	It is the most prevalent model for building a quantum computer using universal quantum logic gates analogous to classical logic gates. It is also referred to as a digital model.
Quantum annealing	It is ideal for solving optimization problems. It is an analog model.
Digital-analog model	It merges digital and analog operations. Taking advantage of both sides, it aims to be universal, scalable, and error-corrected.
Adiabatic model	It is based on quantum annealing and the adiabatic theorem. It is an alternative for optimization problems and is polynomial-time equivalent to the gate model.
Topological model	It models the two main properties of an exotic type of particle known as anyons: fusion and braiding. In fusion, two anyons are brought together. They either annihilate or become a fermion. Braiding means that the moving anyons' trajectories affect the fusion results. These properties result in built-in protection similar to quantum error correction, so qubits based on anyons are much less noisy.

(Rys. 9) Modele komputerów kwantowych, źródło [4]

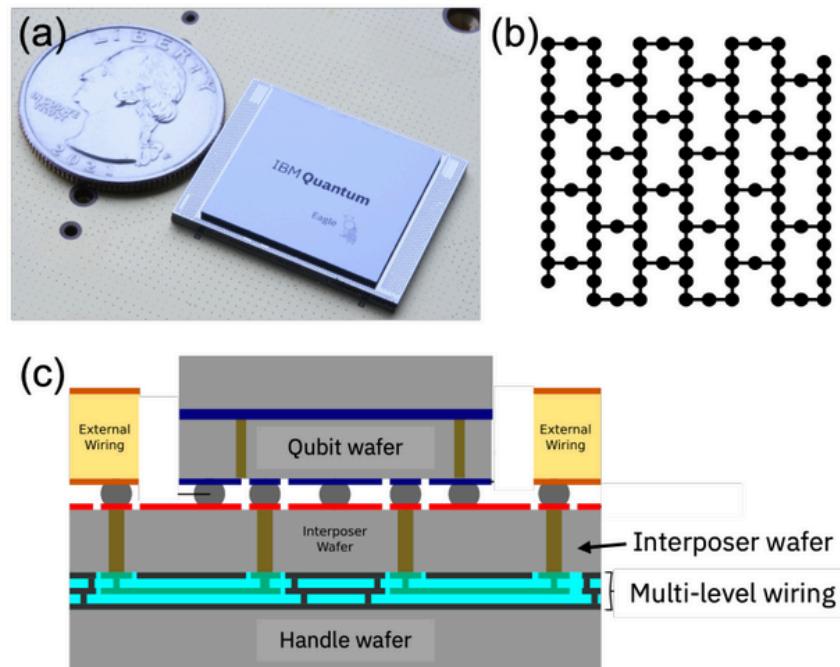
Pamiętając, że obecnie nie istnieją środowiska programistyczne na komputerach kwantowych, korzystamy z symulatorów komputerów kwantowych na klasycznych komputerach, gdzie każda operacja może być podejrzana, odwrócona czy poprawiana, bez niszczenia informacji kwantowej. Z drugiej strony należy pamiętać, że każdy kolejny kubit w symulacji to 2 razy większe zapotrzebowanie symulatora na klasyczną moc obliczeniową, zatem wraz ze wzrostem złożoności algorytmów, osiągniemy kres możliwości stosowania symulacji. Popularne obecnie narzędzia przedstawiono na rysunku.

Name	Description
Google Cirq	It is a Python-based framework for programming quantum circuits. Its simulator also simulates noises.
Google TensorFlow Quantum	It provides ways to use quantum computing inside TensorFlow for hybrid quantum-classical machine learning.
IBM Qiskit	It is a Python-based library to develop quantum programs. It provides convenient ways to test programs in IBM's real quantum computers.
Microsoft Quantum Development Kit	It provides a quantum programming language called Q# and IDE for program visualization and analysis. It provides convenient ways to run programs on the Azure Quantum workspace.
Xanadu Strawberry Fields	It is a Python-based library to program for photonic quantum computing. It provides convenient ways to make remote execution on Xanadu's quantum hardware.

(Rys. 10) Środowiska do tworzenia oprogramowania dla komputerów kwantowych, źródło [4]

W [6] opisano komputery oparte o nadprzewodnictwo. Wykorzystują krzemowe złącze Josephson'a działające jak nieliniowa indukcyjność. Cały kubit to obwód rezonansowy, którym można sterować za pomocą nanosekundowych impulsów mikrofalowych o częstotliwości 5GHz. Schłodzone w temperaturze 15mK, kubity osiągają znaczną szybkość operacji bramkowych. Pozwala to zwiększyć stosunek czasu koherencji (jak długo informacja kwantowa żyje w nienaruszonym stanie) do czasu operacji bramkowej, a zatem można korzystać z głębszych obwodów kwantowych (czyli bardziej złożonych aplikacji). Użycie nadprzewodzących kubitów pozwala wykorzystać obecne technologie produkcji półprzewodników, dając duże pole do poprawy parametrów i wysoką skalowalność.

Bazując na powyższych osiągnięciach, firma IBM opracowała kompleksowe rozwiązanie gotowe do używania w centrach danych, poczynając od 5 kubitowego komputera w chmurze w 2016 r. po maszyny 127 kubitowe w 2022r. Dodatkowo opracowano framework Qiskit, aby uprościć naukowcom korzystanie z tej technologii.



(Rys. 11) 127 kubitowy procesor Eagle, źródło [6]

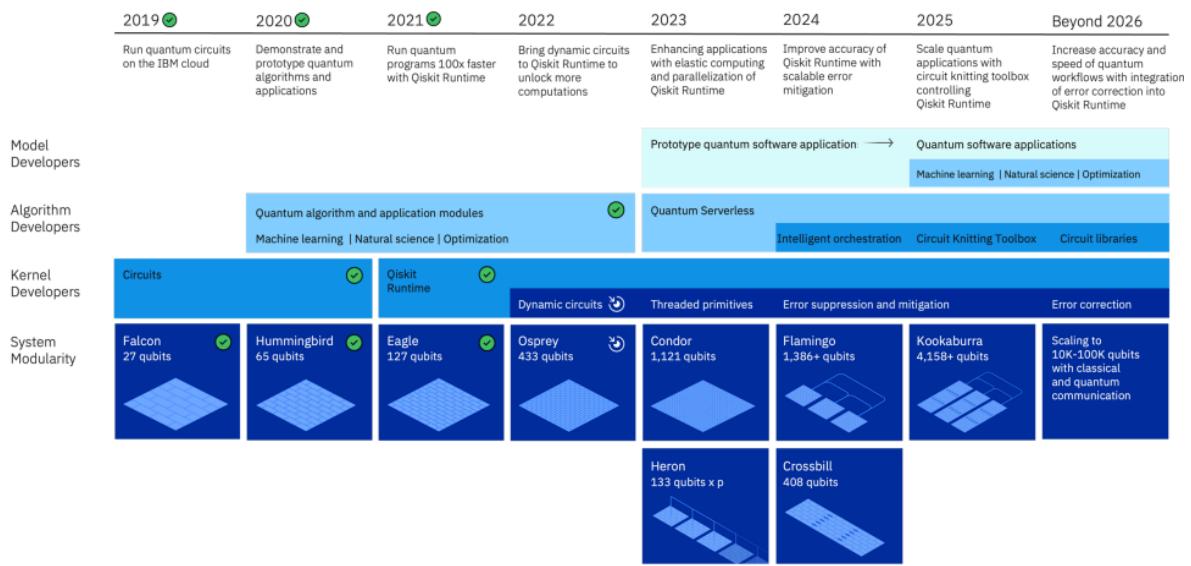
Komputery kwantowe charakteryzują się 3 kluczowymi parametrami determinującymi ich stopień zaawansowania, a przez to przydatność do konkretnych zastosowań.

Skala - czyli ilość kubitów, określająca jak dużo informacji może być reprezentowane w czasie obliczeń. Od skali zależy wielkość problemów obliczeniowych, możliwych do rozwiązania. Przy dużej skali trudniej uzyskać zadowalający czas koherencji. Mimo to każdego roku udaje się podwoić ilość kubitów. Osiąga się to stosując architekturę kratową, czy dostrajając częstotliwości pracy poszczególnych kubitów. Kolejny problem to odczyt tak dużej ilości kubitów oraz interfejsy wejścia/wyjścia, które nie zakłócają sąsiednich kubitów. IBM rozwiązał ten problem stosując ekranowanie, oraz łącząc chipy różnego typu w jeden. Ponadto opracował nowe przewody mikrofalowe, zajmujące mniej miejsca. Rozwiązaniem przyszłościowym, pozwalającym na wzrost skali, może być architektura modułowa, czyli łączenie chipów w jeden większy procesor. Połączenia mogą być zarówno klasyczne jak i kwantowe. Podsumowując, IBM chce wykorzystać jednocześnie klasyczne zrównoleglenie, kwantowe zrównoleglenie przy użyciu linków kwantowych, oraz tworzenie wielo-chipowych procesorów za pomocą kwantowych połączeń brzegowych. Ma to pozwolić w 2025 roku osiągnąć 4158 kubitów.

Development Roadmap

Executed by IBM
On target

IBM Quantum



(Rys. 12) Plan rozwoju procesorów kwantowych IBM [6]

Jakość - wyrażona za pomocą QV (quantum volume) wielkość określająca jak duży obwód kwantowy może być zaimplementowany na danym procesorze. Pod uwagę brana jest ilość jednocześnie zaangażowanych kubitów, jak i ilość operacji bramkowych na nich przeprowadzonych (głębokość). Parametr całosciowo oddaje wydajność urządzenia, uwzględniając wszystkie błędy, przesłuchy, szумy, koherencję, interfejsy wejścia/wyjścia, a nawet oprogramowanie optymalizujące proces komplikacji. Jest to dobry parametr umożliwiający porównywanie urządzeń różnego typu.

Szybkość - wyrażona przez parametr CLOPS (operacje w obwodzie na sekundę), określa ile operacji na kubitach można przeprowadzić w jednostce czasu. Parametr uwzględnia czasy działania wszystkich komponentów, wliczając elektronikę pomocniczą, operacje wejścia/wyjścia czy nawet komplikację kodu. Jak się okazuje, w technologii nadprzewodnictwa, czasy właściwych operacji na kubitach to mały ułamek całkowitego czasu działania. W przypadku kubitów jonowych, opóźnienia wywołane operacjami na bramkach są już o rzęd wielkości większe. Zwykle najczęściej czasu zabiera komplikacja i transfer danych. Typowa szybkości to 1-10k w 2022 r.

Qiskit to narzędzie, które efektywnie łączy obliczenia klasyczne z kwantowymi, dając duże przyspieszenie algorytmów, rzędu x120 (dla VQE). Szczegóły opisano w [6], strona 29. W przyszłości ma być możliwa realizacja dynamicznych obwodów kwantowych, to jest takich, które re-konfigurują się w trakcie działania programu, według wyników uzyskiwanych już w trakcie obliczeń. Pozwoli to na zmniejszenie głębokości algorytmów lub przyda się w nowych rodzajach kodów korekcji błędów. Dalszy postęp to tworzenie bibliotek czy architektury bez-serwerowej, pozwalające na korzystanie z obliczeń kwantowych bez konieczności znajomości zagadnień sprzętowych. Obiecujące wyniki dają techniki *circuit knitting*, *embedding*, *entanglement forging* i *circuit cutting*, będące różnymi sztuczkami, jak lepiej wykorzystać zasoby. Technika *circuit cutting* polega na dzieleniu problemu na małe obwody kwantowe, oraz używaniu klasycznych obliczeń GPU i CPU jako uzupełnienia przy przetwarzaniu danych. *Entanglement forging* polega na podzieleniu problemu na 2 słabo

skorelowane podproblemy. Niewielka korelacja pozwala na przetworzenie ich oddzielnie, a następnie użycie klasycznych obliczeń do symulacji brakujących relacji i uzyskania finalnego rozwiązania. *Embedding* polega na użyciu obliczeń kwantowych tylko do kluczowych aspektów problemu a symulacji klasycznej do pozostały mniej krytycznej części, wykorzystując teorię HF i DFT (*Hartree-Fock, Density functional theory*). Taki odchudzony problem można rozwiązać na prostszych, dostępnych już dziś a nie w przyszłości komputerach kwantowych.

Podsumowując, konieczny jest dalszy rozwój stosu technologicznego złożonego ze sprzętu, oprogramowania nisko- i wysokopoziomowego, algorytmów korekcyjnych, czy technologii towarzyszących. Prognozy dotyczące rozwoju są optymistyczne, ponieważ nie brakuje pomysłów rozwiązywania pojawiających się trudności, a ponadto wciąż trwają badania nad naturą mechaniki kwantowej, co pozwala przypuszczać że jest wiele do odkrycia w tej dziedzinie, co z pewnością przełoży się na przełomowe rozwiązania poprawiające osiągi we wszystkich 3 wymienionych wcześniej metrykach.

Publikacja [7] opisuje obliczenia kwantowe od strony informatyki. Do opisu obwodów kwantowych służą kwantowe języki programowania takie jak Python czy QASM, oraz frameworki bazujące na nich takie jak na przykład Qiskit. Dzięki tym językom i frameworkom możemy wykonywać obliczenia kwantowe w sposób wygodny dla użytkownika, pozwalając na ukrycie szczegółów technicznych poprzez tworzenie bibliotek z algorytmami.

Obwód kwantowy przed uruchomieniem musi być poddany komplikacji, aby zamienić go w postać, która opisuje jakie operacje należy wykonać na danych kwantowych, wykorzystując standardowy zestaw operacji kwantowych, czyli bramek. Kompilacja jest jednym z etapów tłumaczenia programu napisanego przez dewelopera do postaci możliwej do uruchomienia na sprzęcie kwantowym. Kompilator pozwala zatem na ujednolicenie programów przygotowanych w różnych środowiskach programistycznych do jednego standardowego modelu. Kompilator nie zajmuje się jednak szczegółami sprzętu kwantowego, takimi jak zasumienie, czy podatność na błędy, gdyż operuje na idealnych bramkach. Zadaniem kompilatora jest umożliwienie elastycznego definiowania obwodów.

Kolejny krok to synteza obwodu kwantowego dla konkretnego sprzętu, z wykorzystaniem obsługiwanych przez ten sprzęt bramek. Niektóre standardowe operacje mogą być realizowane na różne sposoby. Ponadto niektóre z bramek są trudniejsze w realizacji, lub bardziej podatne na szумy, dlatego w tym miejscu możliwa jest optymalizacja obwodu z uwzględnieniem tych ograniczeń. Ważną sprawą jest zmiana wszystkich operacji nieodwracalnych na ich odwracalne odpowiedniki, ponieważ chodzi o poprawę efektywności energetycznej obliczeń. Operacja nieodwracalna wiąże się ze zniszczeniem części przetwarzanej informacji, stąd jej nieodwracalność. Zgodnie z prawem Rolfa Landauera, zniszczenie informacji wymaga zużycia energii. To jest szkodliwe, bo zwiększa wydzielanie ciepła w opartym o nadprzewodnictwo procesorze kwantowym, który powinien pracować w temperaturze bliskiej zeru absolutnemu. Ponadto użycie nieodwracalnych operacji nie pozwala na cofnięcie obliczeń, a jest to wartościowa i często wykorzystywana sztuczka w pisaniu programów kwantowych. Podsumowując, tak jak wycieki pamięci w klasycznym programowaniu są niepożądane, podobnie w obliczeniach kwantowych odradza się stosowanie operacji nieodwracalnych, gdyż "wycieka" nam informacja którą przetwarzamy. W obliczeniach kwantowych często chodzi bardziej o przekształcenie danych wejściowych w

wyjściowe wyniki, zatem najlepiej jeżeli wykorzystamy bezstratnie całą informację wejściową.

Po syntezie następuje mapowanie logicznych kubitów na kubity fizyczne, 1 do wielu lub 1 do 1 w zależności czy stosowana jest korekcja błędów czy nie. Ten proces jest mocno powiązany z architekturą komputera kwantowego, topologią połączeń między kubitami, czy innymi charakterystykami sprzętu. Na tym kroku pewne części obwodu mogą zostać przeprojektowane. Tutaj także możliwa jest optymalizacja wydajności poprzez uwzględnienie sprzętowych i technologicznych parametrów QPU. W tym miejscu warto zwrócić uwagę na to, że różne rodzaje problemów mogą być lepiej lub gorzej rozwiązywane przez daną architekturę QPU. Dlatego warto rozważyć korzystanie z wielu różnych QPU jednocześnie, dzieląc problem na podproblemy i wysyłając je do odpowiednich dostawców. Jeżeli obwód po wszystkich etapach przekształcenia okazuje się bardziej złożony niż oryginalny obwód, to warto spróbować użyć innego QPU lub innej technologii obliczeń kwantowych. Niewykluczone, że w przyszłości rozwój komputerów kwantowych doprowadzi do powstania urządzeń wykorzystujących jednocześnie różne technologie (jony, nadprzewodnictwo, fotony, eniony), będących wielofunkcyjnymi QPU radzącymi sobie przyzwoicie z każdym programem. Alternatywną gałęzią będą zapewne urządzenia bardziej specjalizowane, osiągające lepszą wydajność dla pewnej klasy problemów, czy wreszcie wysoce zoptymalizowany sprzęt o wybitnej wydajności dla jednego konkretnego problemu. Zatem choć kusząca może być dla dewelopera, zaczerpnięta z klasycznego świata IT pokusa wygody programowania wysokopoziomowego, to obliczenia kwantowe mogą wymagać stałego uwzględniania architektury niskopoziomowej, a nawet stosowania wielu architektur jednocześnie.

Kolejne poruszane w [7] zagadnienie to symulatory obliczeń kwantowych. Firma Nvidia zaproponowała środowisko cuQuantum, które może wykorzystywać GPU do symulacji, ponieważ obliczenia są intensywne, ale dość proste i powtarzalne. To głównie operacje na macierzach i tensorach. Inna metoda, dająca dobre rezultaty to użycie FPGA, które cechują się dużą wydajnością, a zużywają mniej energii. Symulatory oparte o FPGA dzielą się na uniwersalne symulatory ogólnego przeznaczenia, oraz produkty specjalizowane pod konkretne algorytmy kwantowe. Te ostatnie osiągają wyższą wydajność dla wybranych algorytmów takich jak na przykład kwantowa transformata Fouriera.

Host Processor

Q) How to develop user-friendly quantum software?



Control Processor Plane

Q) How to improve error mitigation?

Q) How to implement high-performance error correction?



Control and Measurement Plane

Q) How to prevent decoherence?

Q) How to increase gate fidelity?



Quantum Data Plane

Q) How to efficiently control the qubit?

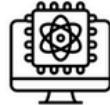


Fig. 2. Open research question for each plane in a quantum computer

(Rys. 13) źródło [7]

Na rysunku 13 przedstawiono typową warstwową architekturę komputera kwantowego opartego na bramkach. Na każdej z warstw pojawiają się kluczowe problemy do rozwiązania. Odpowiadają one w przybliżeniu wspomnianym wcześniej etapom komplikacji programu kwantowego do postaci możliwej do wykonania na fizycznych kubitach. Na najniższej, nie wspominanej wcześniej warstwie, ciekawym problemem jest hipotetyczny QRAM, czyli kwantowa pamięć operacyjna dla procesora kwantowego. Stworzenie takiego komponentu pozwoliłoby rozszerzyć możliwości obliczeń kwantowych. QRAM miałby przechowywać informację kwantową w komórkach jedno- lub wielo-kubitowych, co oznaczałoby, że jedna n-kubitowa komórka pamięci może przechowywać do 2^n wartości w superpozycji. Co więcej, także adres komórki mógłby być rejestrem kubitowym, co oznaczałoby, że możemy odczytywać jednocześnie wiele komórek, uzyskując na wyjściu superpozycję zawartości wybranych komórek. Oczywiście QRAM musiałby działać zgodnie z zasadami mechaniki kwantowej, w szczególności niemożliwe byłoby skopiowanie informacji. Nie powinien używać też operacji nieodwracalnych, ponieważ następowałaby utrata informacji. Nasuwają się tutaj pytania, co by było gdyby chcieć zapisać coś jednocześnie do kilku komórek, a potem odczytywać je pojedynczo? Albo czy gdybyśmy odczytali kilka komórek na raz, a na wyjściu danych otrzymali superpozycję, czy nie stracimy części informacji, a jeśli tak to gdzie znajdzie się pozostała część? Czy splątanie kwantowe nie powiąże jakoś danych w QRAM pomiędzy poszczególnymi komórkami i nie spowoduje jakichś ubocznych efektów gdy przeprowadzimy działania na jednej z komórek? Wreszcie czy stosowanie architektury opartej na adresacji poszczególnych komórek ma sens albo jest optymalną metodą uzyskiwania dostępu, skoro mamy takie mechanizmy jak splątanie czy teleportacja? Może jeszcze lepszą metodą byłoby przechowywanie informacji poprzez dynamiczną mapę splatań i innych oddziaływań między poszczególnymi kubitami? Być może w ciągu kolejnych dziesięcioleci poznamy odpowiedzi na te pytania.

Materiałów na temat obliczeń kwantowych i rozwiązań technologicznych znalazłem więcej. Najważniejsze z nich, zostały streszczone i skomentowane powyżej. Z materiałów dodatkowych, można wymienić szczegółowy opis efektywnej kontroli QPU opartego o nadprzewodnictwo [8], pozwalający przyspieszyć obliczenia. QubiC 2.0 [9] to inny system kontroli QPU, bazujący na Xilinx Radio Frequency System-on-Chip (FPGA), z otwartym kodem źródłowym, rozwijany przez Lawrence Berkeley National Laboratory. Odpowiednio dobierając typy bramek oraz parametry impulsów kontrolnych, oraz stosując adaptacyjny algorytm redukcji szumów QuPAD [10] można znacznie przyspieszyć (270 razy) algorytm uczenia maszynowego taki jak VQE (*variational quantum algorithm*) oraz poprawić dokładność klasyfikacji o kilkadziesiąt procent. Całkiem osobna gałąź wiedzy, dotycząca komputerów kwantowych opartych o jony, a ścisłe ich elektroniki kontrolnej, została przedstawiona w [11]. Komputery jonowe, jako kubity, wykorzystują naładowane cząstki, uwiezione w miejscu, w elektromagnetycznych studniach. Stan jonów może być modyfikowany i odczytywany za pomocą impulsów laserowych, oraz detektorów fotonów. Publikacja zbiera i przedstawia informacje o rozwiązaniach technologicznych stosowanych do kontrolowania kubitów jonowych. Podobnie [14], opisuje szczegółowo działanie kubitów z użyciem jonów, skupiając się tym razem na właściwych cząstkach. Abstrahując od technologii jonowych, integracja CPU, GPU i QPU opisana w [12] pozwoli na efektywniejsze wykorzystanie zalet każdej technologii. Będzie to przydatne w nowych aplikacjach takich jak Metaverse. Dokument [13] poświęcono szczegółowo opisowi kubitów nadprzewodzących, wraz z teorią i obliczaniem parametrów pracy. W [15] opisano przetworniki cyfrowo-analogowe i analogowo-cyfrowe, jako przykład mieszanej elektroniki typu Cryo-CMOS, używanej w komputerach kwantowych. Wreszcie ostatnią, ale bardzo ważną sprawą w obliczeniach kwantowych, szczególnie w środowiskach chmurowych lub współużytkowanych, jest zapewnienie odpowiednich standardów ochrony przetwarzanych danych, własności intelektualnej czy prywatności użytkowników [16].

Tak oto podsumowaliśmy stan rozwoju obliczeń kwantowych - kluczowej technologii kwantowej. Przejdźmy więc do dziedzin powiązanych.

Komunikacja kwantowa

Jak wspomniano na początku tego rozdziału, dziedzina ta zajmuje się wymianą informacji, w tym informacji kwantowej. Choć pierwsze pojedyncze prototypy procesów kwantowych służyły badaniom zjawisk kwantowych i eksperymentalnym obliczeniom, to oczywiste jest, że aby obliczenia były użyteczne, powinna istnieć możliwość udostępniania wyników poza QPU oraz odbierania informacji, w tym informacji kwantowej z zewnątrz. Ponadto komunikacja kwantowa pozwala łączyć małe QPU w większe zestawy, pomnażając ich moc obliczeniową, co może okazać się kluczowe do stworzenia dużych komputerów kwantowych w przyszłości. Wreszcie komunikacja kwantowa ze względu na specyficzną naturę warstwy fizycznej, uważana jest za bezpieczniejszą alternatywę dla klasycznie szyfrowanej komunikacji internetowej. Najpowszechniejszym zastosowaniem komunikacji kwantowej są obecnie bezprzewodowe, satelitarne i światłowodowe sieci dystrybucji kluczy (QKD), wykorzystujące bazujące na zjawiskach kwantowych protokoły uzgadniania współdzielonych sekretów, mogących być następnie użytych jako klucze do szyfrów symetrycznych w klasycznej komunikacji. W przyszłości komunikacja kwantowa pozwoli stworzyć całkowicie

kwantowy internet (QInternet). Zanim to nastąpi, przyjrzyjmy się zebranym przeze mnie materiałom.

W pracy [17], naukowcy wspominają o komunikacji kwantowej jako ultra bezpiecznej technologii komunikacji z łodziami podwodnymi. Osiągnięte przepustowości to 170kb/s na 100m.

Sieciom 5G poświęcono publikację [18], opisując wysiłki w kierunku zabezpieczania transmisji z wykorzystaniem kwantowej dystrybucji kluczy (QKD) i kryptografii post-kwantowej. QKD wykorzystuje takie protokoły jak BB84, do uzgodnienia pewnych losowych sekretów. Protokół BB84 pozwala generować w sposób ciągły losowe wartości, które są znane tylko stronom biorącym udział w protokole. Próba przechwycenia wymienianych kubitów, zostanie łatwo wykryta, zatem pomyślne wygenerowanie kluczy oznacza, że są one rzeczywiście poufne. Tak uzgodnione klucze stanowią dodatkowe zabezpieczenie do już istniejącej klasycznej architektury bezpieczeństwa opartej o tradycyjne protokoły kryptograficzne. Protokół BB84 polega na mierzeniu każdego przesyłanego kubitu za pomocą losowo wybranej bazy, zarówno u nadawcy jak i odbiorcy. Po wykonaniu serii pomiarów i odrzuceniu tych, w których bazy się nie zgadzały, otrzymujemy po obu stronach te same zmierzzone wartości kubitu. Jeżeli pomiędzy nadawcą i odbiorcą dokonano przechwycenia danych lub innej ingerencji to wartości nie będą się zgadzały w 100% ponieważ każda ingerencja nieodwracalnie zaburza kubity. Aby zweryfikować bezpieczeństwo połączenia, strony powinny przekazać sobie część wygenerowanego klucza dla porównania. Jeżeli nie stwierdzono różnic, to pozostała część klucza może być użyta do szyfrowania danych. Ograniczeniem obecnej technologii QKD jest stosunkowo niewielka ilość uzyskanej w ten sposób losowej informacji. Ogranicza to częstotliwość rotacji kluczy szyfrujących.

Publikacja [19] opisuje sieci front-haul, będące łączami modułów nadawczo-odbiorczych 5G z jednostkami przetwarzającymi sygnał w formie zdigitalizowanej. Badacze stworzyli środowisko pomiarowe, w którym badają wydajność szyfrowania z wykorzystaniem kwantowej dystrybucji kluczy. Osiągnęli prędkości standardowe rzędu 100 Gbps, oraz niskie czasy rekonfiguracji kluczy, co oznacza że ten sam klucz był użyty do zaszyfrowania tylko 10 Gb danych. Jest to przykład praktycznego wykorzystania zjawisk kwantowych w rozwiązywaniu problemów życia codziennego.

Wracając do [4] w kontekście komunikacji kwantowej, możemy przyjrzeć się sieciom kwantowym bliżej. Komunikacja kwantowa wykorzystuje zwykle protokół teleportacji, polegający na wykorzystaniu splatania kwantowego do przesłania stanu lokalnego kubitu do kubitu odległego. Splatanie kwantowe można uzyskać na różne sposoby, na przykład wywołując interakcje między kubitami. Należy zwrócić uwagę na to, że protokół teleportacji wymaga transmisji klasycznej informacji w celu umożliwienia prawidłowego odbioru danych, co oznacza że komunikacja może następować najwyższej z prędkością światła. Samo oddziaływanie między splatanymi kubitami odbywa się jednak natychmiast.

Komunikacja kwantowa boryka się z trudnościami technicznymi takimi jak szумy, niestabilność, dekoherencja czy ograniczenia w odległości i przepustowości. Teoretycznie kubit może przechowywać informację kwantową w nieskończoność, o ile jest całkowicie odizolowany od otoczenia. Komunikacja jest przeciwieństwem izolacji, stąd nieuniknione

interakcje z otoczeniem zaburzają stan kubitów. Kubit może przypadkowo splątać się z otoczeniem i przez to nie spełniać pierwotnego przeznaczenia.

Łącza bezprzewodowe wydają się być stabilniejsze od światłowodowych, ponieważ odkryto okna transmisyjne w paśmie widzialnym o niskim tłumieniu i zakłóceniach. Np fala 650-670 nm posiada niewielki rozrzut dyfrakcyjny, a 770 nm nadaje się do dużych przepustowości.

Kolejnym problemem są ograniczenia spowodowane brakiem możliwości kopiowania informacji. W klasycznych systemach łączności, dane odebrane nieprawidłowo mogą być przesłane ponownie, aby uzupełnić braki u odbiorcy. W przypadku informacji kwantowej, jest ona niszczona u nadawcy, a pojawia się u odbiorcy. Jeżeli teleportacja się nie powiedzie, to informacja ginie bezpowrotnie. Można temu zaradzić poprzez użycie kodowania z redundancją, czy innych metod umożliwiających tolerowanie błędów. Jednak nie jest to rozwiązanie idealne. Jednym z problemów jest bezpieczeństwo informacji kwantowej poddanej redundancji. Gdy informacja istnieje w jednym egzemplarzu i jest nie-klonowalna, to nie da się jej przechwycić niezauważenie. Gdy stosujemy nadmiarowość na łączu, to możliwe staje się przechwycenie wybranych kubitów i odtworzenie informacji, a jednocześnie sieć działa dalej, gdyż toleruje sporadyczne błędy transmisji.

Parametr *Fidelity* określa jak dużo informacji jest zachowanej po teleportacji. Wartości przekraczające 90% są uważane za dobry rezultat. Dla porównania standardy takie jak ethernet cechują się o wiele rzędów lepszą wiernością.

Sieć kwantowa działa najlepiej w modelu komunikacji 1 do 1. Trudno jest realizować komunikację 1 do wielu, z powodu nie-klonowalności informacji. Poza tym sieć kwantowa wygląda podobnie do klasycznej sieci. Można wyodrębnić zbliżone topologie połączeń. Węzły powinny podejmować decyzję jak trasować dane. Powinny też uwzględniać probabilistyczny charakter połączeń. Wreszcie zakłada się, że elementy sieci kwantowej są zaufane. Nie jest to możliwe w klasycznym internecie, który jest w rękach wielu niezależnych podmiotów.

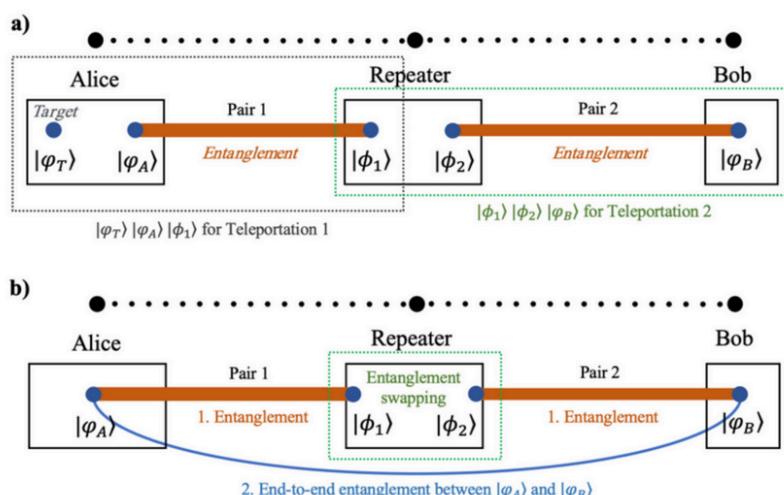
Połączenia w sieciach kwantowych można realizować z wykorzystaniem istniejących światłowodów oraz źródeł emitujących pojedyncze fotony lub pary splątanych fotonów. Można używać światłowodów jedno i wielomodowych. Zaobserwowano że QKD działa lepiej (szybciej) w światłowodach wielomodowych, ponieważ pozwalają na transmisję w wielowymiarowych trybach przestrzennych. Niestety są one bardziej podatne na szумy, a co za tym idzie ogranicza to odległości transmisji. Próbuje się opracowywać kody korekcyjne które pozwolą rekonstruować informację, co pozwoli na poprawę osiągów.

Osobną kategorią są łączza bezprzewodowe, czyli optyczna transmisja w wolnej przestrzeni. Osiągane są o wiele większe odległości, gdyż wpływ atmosfery jest niewielki. Przy braku atmosfery w przestrzeni kosmicznej, zasięg ulega dalszej poprawie. Z drugiej strony, łączza w otwartej przestrzeni są podatne na naturalne czynniki, na przykład deszcz, albo na świetlny szum tła w dzień.

Type	Distance	Fidelity/QBER	Description	Year				
Optical fiber (QKD)	>830 km	3.79%	It uses an optimized four-phase twin-field protocol with a high-quality setup to implement twin-field QKD.	2022	Optical fiber (Entanglement)	50 km	0.86 ± 0.03	It uses the sources of ion-photon entanglement via cavity-QED techniques and a single photon entanglement.
Optical fiber (QKD)	>511 km	0.43%	It uses a sending-or-not-sending protocol with quantum and classical communication in the fiber trunk to implement twin-field QKD.	2021	Optical fiber (Entanglement)	192 km	0.85 ± 0.02	It creates remote entanglement based on polarization-entangled photon pairs in submarine cables.
Optical fiber (QKD)	2 km	$\approx 4.9\%$	It uses multicore fiber to increase the key rate generation (6.3 Mbit/s) with enhanced error tolerance.	2021	Optical fiber (Entanglement)	100 km	0.93	It uses non-degenerate down-conversion by polarization-entangled photon pairs to distribute entangled pairs.
Optical fiber (QKD)	421 km	$\approx 3\% - 6\%$	It uses QKD-optimized superconducting single-photon detectors and ultra-low-loss fibers.	2018	Optical fiber (Teleportation)	>100 km	0.837 ± 0.02	It uses four high-detection-efficiency superconducting nanowire single-photon detectors for quantum teleportation.
Optical fiber (Entanglement)	20 km	$\geq 0.785 \pm 0.009$	It uses polarization-preserving quantum frequency conversion to create entanglement.	2020	Free space (Teleportation)	1,400km	0.80 ± 0.01	It is claimed to be the first quantum teleportation from a ground observatory to a low Earth orbit satellite.
					Free space (Entanglement)	1,200 km	$\geq 0.87 \pm 0.09$	It is based on the observation of the survival of 2-photon entanglement and a violation of Bell inequality.

(Rys. 14) Zestawienie połączeń kwantowych różnego typu, źródło [4]

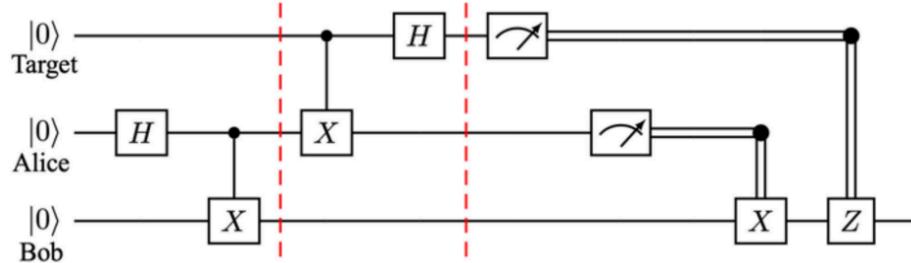
Co ciekawe, stan splatania może być regenerowany przez przekaźniki, a dzięki teleportacji kubitów, splatane kubity można przenosić do odległych węzłów co pozwala na tworzenie splatania typu end-to-end. Można zatem obejść ograniczenia odległości. Tworzenie splatania end-to-end można realizować na 2 sposoby: a) przez wielokrotną teleportację, b) przez zastąpienie 2 splatań jednym (*entanglement swapping*). Przedstawiono je na rysunku poniżej.



(Rys. 15) Tworzenie splatania end-to-end za pomocą splatań między sąsiednimi węzłami, [4]

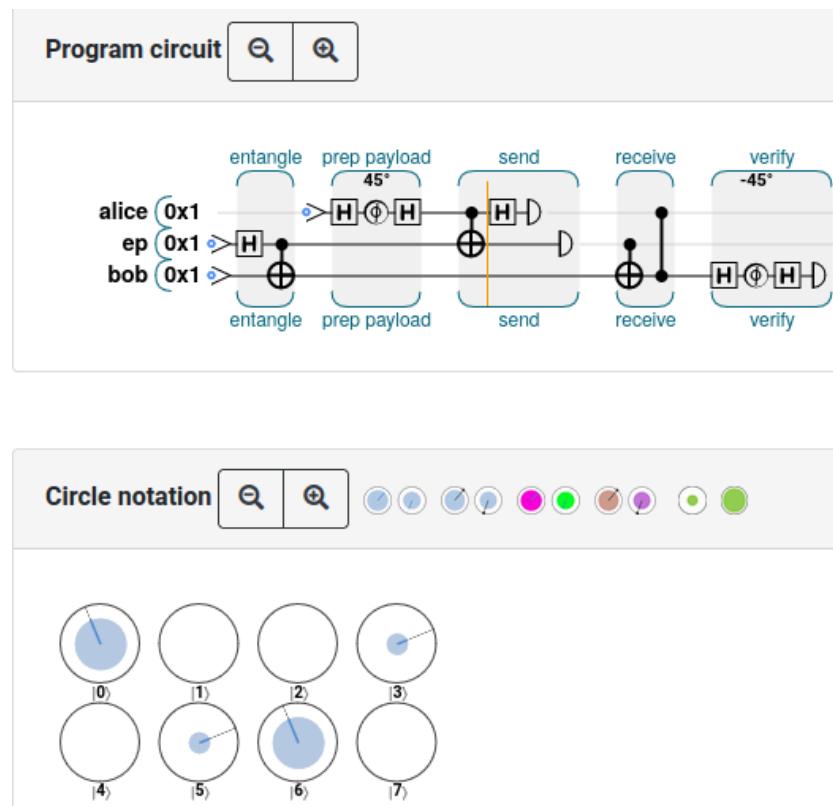
Zanim opiszemy szczegółowo jak działają te protokoły, poznajmy protokół teleportacji jako taki. Pojedyncza teleportacja wykorzystuje 3 kubity. 1 i 2 znajdują się u nadawcy, natomiast 3 u odbiorcy. ponadto 2 i 3 są ze sobą już wcześniej splatane. Aby dokonać teleportacji kubitu 1, nadawca wykonuje pewne operacje na kubitach 1 i 2. następnie przesyła 2 bity

klasycznej informacji do odbiorcy. Odbiorca wykorzystuje tą informację aby dokonać końcowego przetwarzania stanu kubitu 3. Po teleportacji kubit 3 zawiera identyczny stan jak wcześniej zawierał 1, natomiast informacja w 1 zostanie zniszczona. Stąd nazwa teleportacja.



(Rys. 16) Obwód teleportacji, źródło [4]

Na (Rys. 16) kubit 1 to Target, 2 - Alice, 3 - Bob. Na początku, nadawca i odbiorca uzyskują parę splatanych kubitów, 2 i 3, będących w superpozycji. Taka para może być użyta jeden raz. Można taką parę uzyskać z użyciem bramek H i X, następnie jeden z kubitów przesyłać do Boba (na przykład jako foton) a drugi zatrzymać do dalszych operacji. Alternatywnie, można użyć źródła emitującego już splatane pary, jedną cząstkę złapać a drugą przesyłać bezpośrednio do odbiorcy. W każdym przypadku, splatany kubit w trakcie przesyłania nie zawiera jeszcze żadnej informacji. W trakcie teleportacji, zostanie z kubitami 2 i 3 splatany także kubit 1, co widać na (Rys. 17), w miejscu programu zaznaczonym pomarańczową pionową kreską. W ten sposób możliwe staje się przeniesienie informacji z 1 do 3.



(Rys. 17) Dowód na splatanie kubitów K1, K2 i K3, źródło [20]

Koła oznaczają rozkład prawdopodobieństwa odczytania określonego stanu 3 kubitów. Kubit K1 na początku był w superpozycji (stan 0 był bardziej prawdopodobny niż 1). Po splątaniu kubitów widzimy zależność kubitu K3, od K1 i K2. Zależność jest następująca: Jeżeli kubit K2 zostanie odczytany jako 0, to K3 po odczytcie będzie mieć taki sam stan jak K1. Jeżeli kubit K2 zostanie odczytany jako 1, to K3 po odczytcie będzie mieć przeciwny stan niż K1. Innymi słowy, stan każdego kubitu jest determinowany stanem pozostałych. Znając stan 2 dowolnych, możemy wyliczyć stan trzeciego.

Rysunek 17 pochodzi z symulatora [20]. Odwołania do [20] w dalszej części pracy, w zależności od kontekstu, oznaczają sam symulator, przykłady dostępne w tym symulatorze, przykłady zmodyfikowane na potrzeby tej pracy lub wyniki uruchomienia własnego kodu.

Protokół teleportacji zaczyna się miejscu zaznaczonym pierwszą przerywaną linią (patrz Rys. 16). Nadawca wykonuje operację CNOT (kubit 2 jest negowany w zależności od stanu kubitu 1), a następnie używa bramki Hadamarda na kubicie 1. Na koniec stan kubitów jest mierzony, a wyniki w postaci 2 klasycznych bitów są przesyłane do odbiorcy. Ponieważ wszystkie 3 kubity są splątane, dlatego operacje na kubitach u nadawcy zmieniają stan kubitu u odbiorcy, mimo odległości między nimi. W ten sposób informacja z kubitu 1 przenosi się do kubitu 3. W miejscu zaznaczonym drugą przerywaną linią, kubit 3 znajduje się już w jednym z 4 możliwych stanów, z których łatwo można otrzymać stan docelowy. Jeden z tych stanów to stan identyczny z początkowym stanem teleportowanego kubitu 1. Pozostałe 3 stany są modyfikacjami teleportowanego stanu. Odbiorca nie wie w którym stanie znajduje się jego kubit, a może się tego dowiedzieć z 2 bitów klasycznej informacji otrzymanych od nadawcy. Następnie używa odpowiednich bramek warunkowych X i Z, kontrolowanych tymi bitami, aby doprowadzić stan swojego kubitu do właściwego stanu. X i Z to warunkowe bramki kwantowe sterowane klasycznymi bitami informacji. Jest to szczególny przypadek warunkowych bramek sterowanych kubitami kontrolnymi, dla zdeterminowanej wartości kontrolnej (brak superpozycji, wejście kontrolne może być tylko w 2 różnych stanach). Pozwala to wykonać 4 różne przekształcenia: nic nie rób, tylko negacja, tylko zmiana fazy oraz negacja ze zmianą fazy. W ten sposób protokół teleportacji pozwala przenosić (teleportować) stany kwantowe z jednego kubitu do innego, także na duże odległości. Wystarczy aby nadawca i odbiorca mieli 2 splątane kubity. Nie jest konieczne posiadanie bezpośredniego połączenia między nadawcą i odbiorcą. Ze względu na konieczność przesłania bitów klasycznych, nie możemy przesyłać informacji szybciej niż światło.

Jak splątać odległe kubity? Najprościej użyć źródła, wytwarzającego pary splątanych fotonów lub innych cząstek. Następnie jedną z nich trzeba wysłać do partnera komunikacji a drugą zatrzymać. Cząstki zachowują stan splątania mimo rozdzielenia, czy przemieszczenia ich, i o ile nie nastąpi przypadkowa dekoherencja spowodowana niedoskonałością technologii, mogą być użyte w różnych protokołach i algorytmach. Wysłanie kubitu może odbyć się bezpośrednim światłowodem, lub łączem bezprzewodowym w otwartej przestrzeni. Co jednak jeżeli nie mamy bezpośredniego połączenia z odbiorcą, ale mamy dostęp do QInternetu albo innej sieci pośredniczącej w wymianie informacji kwantowej?

Rozwiązaniem są właśnie protokoły wspomniane wcześniej, przedstawione na (Rys. 15). Najpierw wyjaśnijmy przypadek a) czyli wielokrotną teleportację. Chociaż Alicja i Bob nie mają bezpośredniego połączenia, to każde z nich jest połączone z przekaźnikiem, który pełni analogiczną funkcję do routera w sieci IP.

Na początku Alicja i przekaźnik tworzą parę splatanych kubitów φ_A i Φ_1 . Teleportowany kubit oznaczono jako φ_T . Ukończenie pierwszej teleportacji spowoduje, że informacja z φ_T trafi do Φ_1 . W tym momencie nasz kubit przebył już jeden hop w sieci i znalazł się w miejscu początkowym do przeprowadzenia drugiej teleportacji. Przekaźnik i Bob tworzą własną parę splatanych kubitów Φ_2 i φ_B . Teleportowany kubit to wspomniany już Φ_1 (zawierający to co wcześniej φ_T). Po ukończeniu drugiej teleportacji Φ_1 trafi do φ_B . Zatem stan φ_T trafi ostatecznie do φ_B . Jeśli φ_T był na początku splatany z innym kubitem posiadanym przez Alicję, to po obu teleportacjach Alicja i Bob będą mieć parę end-to-end splatanych kubitów. Splatanie powinno “podażać” za teleportowanym kubitem. Alternatywnie Alicja może Bobowi przesyłać zwykły, niesplatany kubit jako informację.

Jeśli celem jest uzyskanie splatania end-to-end, możemy posłużyć się też protokołem b). Tak jak w poprzednim przypadku, najpierw uzyskuje się osobne splatanie między każdą parą węzłów sieci. Następnie węzeł pośredniczący przeprowadza operację *entanglement swapping*, która z dwóch splatań, tworzy jedno splatanie end-to-end.

Posiadanie splatanej pary kubitów w odległych systemach pozwala na wiele interesujących operacji. Można dzięki temu na przykład programować interakcje między kubitami w różnych QPU. To tak, jak byśmy stworzyli duży wirtualny QPU z wielu fizycznych QPU. Ponadto przeprowadzanie obliczeń kwantowych to przetwarzanie informacji kwantowej zawartej w kubitach, a żeby informację przetwarzać, trzeba ją też mówić przechowywać, transmitować i udostępniać. Zatem sieć Internet nie jest odpowiednim nośnikiem, ponieważ pozwala na transmisję jedynie informacji klasycznej. Dlatego komunikacja kwantowa jest taka ważna.

Stworzenie QInternetu, czyli odmiany Internetu do wymiany informacji kwantowej, wymaga opracowania wszystkich elementów sieciowych, bazując na fizyce kwantowej i uwzględniając naturę tej informacji. Jednym z elementów są routery, kierujące ruchem, z wykorzystaniem dostępnych połączeń. Powinny one posiadać pamięć kwantową, do przechowywania informacji o trasach, oraz mieć możliwość buforowania informacji kwantowej. Ze względu na niedeterministyczną naturę połączeń, powinny one uwzględniać właściwości poszczególnych linków przy podejmowaniu decyzji o następnym węźle dla danych. Sieci kwantowe mogą mieć różne topologie, chociaż ze względu na brak możliwości klonowania informacji, trudno jest realizować transmisję 1 do wielu. Z drugiej strony efekty kwantowe pozwalają na osiągnięcie nowej jakości w ochronie poufności informacji. Wymaga to opracowania nie tylko rozwiązań sprzętowych, ale i oprogramowania zarządzającego siecią oraz nowych protokołów komunikacyjnych. Jednym z takich protokołów jest protokół zaproponowany przez indyjskich i amerykańskich naukowców [21].

Protokół QTTP (*Quantum Text Teleportation Protocol*) bazuje na protokole teleportacji opisany wcześniej oraz kodowaniu Huffmana, w celu kompresji danych. Protokół polega na zakodowaniu każdego bitu informacji tekstowej w kubicie, a następnie teleportowaniu go do odbiorcy. Odbiorca odzyskuje oryginalną treść, finalizując teleportację przez warunkowe użycie bramek X i Z, a następnie mierząc stan swojego kubitu. Bezpieczeństwo transmisji opiera się na wysłaniu informacji tylko w jednej kopii. Jeżeli kubit zostanie przechwycony, to odbiorca nie otrzyma informacji, a błędy transmisji są łatwe do zauważenia. Użycie kodowania Huffmana pozwala na zmniejszenie ilości przesyłanych danych. Ta sama grupa

naukowców stworzyła też analogiczne protokoły do teleportacji obrazów, dźwięku i plików wideo, [22] wykorzystując tą samą metodę, wspomnianą wcześniej w [21].

Bardziej zaawansowane protokoły transmisji danych w sieciach kwantowych QDN, odpowiednie do rozproszonych obliczeń kwantowych, opisano w [23]. Sieć QDN to specyficzna odmiana Internetu kwantowego, odpowiednia do implementacji w centrach danych. Zaproponowano 2 rodziny protokołów pozwalające na niezawodną transmisję danych kubitowych: bazujący na teleportacji protokół Tele-QDN oraz na metodzie tell-and-go TAG-QDN. W pierwszej metodzie tworzy się splatanie end-to-end i przesyła dane za pomocą protokołu teleportacji. W drugiej metodzie informacja jest enkodowana bezpośrednio w fotonach, które są przesyłane bezpośrednimi linkami światłowodowymi do miejsca przeznaczenia, z użyciem przekaźnika lub bez niego. Ponadto protokoły QDN zapewniają kontrolę przepływu danych, w celu zarządzania natłokiem informacji. Jest to konieczne ponieważ ewentualne przeciążenia w sieci mogłyby prowadzić do nieodwracalnej utraty informacji, a ponadto pamięć kubitowa jest zasobem mocno ograniczonym i należy nim zarządzać oszczędnie. Z tego powodu także ilość jednocześnie transportowanych sesji jest kontrolowana. Innymi słowy chodzi o jak najpłynniejszy ruch kubitów, unikanie wąskich gardeł, monitorowanie wykorzystania pamięci kubitowej, czy sprawiedliwe zarządzanie podziałem przepustowości. Od strony kwantowej, ciekawą nowością jest metoda kodowania pojedynczego kubitu użytkownika w wielu kubitach transportowych, używając kodowania nadmiarowego (*secret sharing*), pozwalającego na odtworzenie oryginalnej informacji mimo jej częściowej utraty. Pozwala to na osiągnięcie niezawodności w dostarczaniu kubitów.

Kodowanie działa następująco: Przymijmy użycie schematu (2,3). Oznacza on zakodowanie każdego kubitu użytkownika w 3 kubitach transportowych. Oryginalny kubit można odtworzyć dysponując dowolnymi dwoma. Chcemy przesłać kubit A od nadawcy do odbiorcy. Zakodujemy kubit A jako zestaw $A_1 A_2 A_3$ i przesyłamy je kolejno do odbiorcy. Odbiorca rekonstruuje z nich oryginalny kubit A. Wystarczą 2 pierwsze z nich aby było to możliwe. Jeżeli wystąpi błąd w komunikacji i kubit A_1 nie dotrze do odbiorcy, to nadawca odtwarza z pozostałych posiadanych kubitów A_2 i A_3 oryginalny kubit A, następnie generuje nowe kubity $A_1 A_2$ i A_3 i próbuje przesłać kubit A_1 . Jeżeli się to uda, to nadawca przesyła kubit A_2 . Pomyślne przesłanie kubitu A_2 oznacza sukces transmisji, gdyż odbiorca ma już wymagane 2 kubity do rekonstrukcji A. Jeżeli jednak A_2 przepadnie, to nadawca posiada już tylko A_3 i nie może odtworzyć pierwotnego A. Także odbiorca nie może go odtworzyć bo ma tylko A_1 . Nadawca nie może wysłać A_3 za pomocą niepewnego łączka, bo jeśli także A_3 zginie, to kubit A będzie nieodwracalnie utracony, a protokół musi zapewnić niezawodną transmisję. Aby rozwiązać ten dylemat, nadawca może zakodować A_3 kodem nadmiarowym. Powstaną kubity $A_{31} A_{32} A_{33}$. Jeżeli uda się dostarczyć A_{31} i A_{32} , to odbiorca będzie mógł odtworzyć A_3 . Mając A_1 otrzymany wcześniej i zrekonstruowany A_3 , odbiorca może odtworzyć A, co zapewnia sukces transmisji. Podsumowując, protokół tworzy warstwę oferującą niezawodny transport kubitów. Protokół działa dobrze dla stabilnych łącz, ze sporadycznymi błędami. Jeżeli błędów jest więcej to powoduje to częstsze zagniezdżanie się algorytmu, co obniża efektywną przepustowość oraz zwiększa zapotrzebowanie na pamięć kubitową do przechowywania wszystkich kubitów pośrednich. Zaletą jest natomiast niezawodność algorytmu. Ewentualne niedokładności mogą powstać z powodu niedoskonałości operacji na kubitach, dodatkowo kumulujące się jeżeli algorytm musi naprawiać więcej błędów.

W opracowywanych obecnie standardach sieci 6G, będących ewolucyjno-revolucyjnym rozszerzeniem obecnych sieci 5G, przewiduje się wykorzystanie obliczeń i komunikacji kwantowej [24]. Przykładowe technologie wymieniono na (Rys. 18).

Stages of quantum-enabled 6G systems	New functions/services	Enabling technology
Near-term (2–3 years)	Secure communications for backhaul	Fiber QKD
Middle-term (3–5 years)	Secure satellite communications; satellite-assisted secure communications	Satellite QKD
	Optimal wireless resource management	Quantum computing
Long-term (5–10 years)	Data computing with privacy preservation	Blind quantum computing
	Real-time wireless AI	Quantum machine learning

(Rys. 18) Wykorzystanie technologii kwantowej w 6G, źródło [24]

Częścią sieci 6G będzie też segment satelitarny złożony z satelitów rozmieszczonych na niskich, średnich i geostacjonarnych orbitach. Możliwość bezpośredniej optycznej kwantowej komunikacji została już praktycznie zrealizowana. Sieci satelitarne są i będą wykorzystywane do dystrybucji kluczy. Mogą być też użyte do oferowania użytkownikom obliczeń kwantowych jako usługi, oraz pośredniczyć w komunikacji. Sieć 6G ma też przewidywać i integrować *edge computing*, czyli obliczenia (w tym obliczenia kwantowe) wykonywane na węzłach końcowych. Edge computing w odróżnieniu od tradycyjnych obliczeń opartych na chmurze, pozwala na przykład na wstępna czy końcową obróbkę informacji. Jeżeli w budynku znajduje się 10 tysięcy kamer monitoringu wysokiej rozdzielczości pracujących ciągle, to generują one duży ruch do chmury. Jednak większość tej informacji nie jest użyteczna. Edge computing w formie AI, pozwala na odfiltrowanie interesującej treści co przekłada się na zmniejszenie wolumenu przesyłanych danych, a przez to znaczną redukcję zużycia energii. Dostępność w przyszłości QPU pracujących w temperaturze pokojowej, pozwoli na implementację *edge computing* także w wersji kwantowej.

Komunikacja kwantowa jest częścią cyberbezpieczeństwa opartego o determinizm w architekturze zerowego zaufania [25]. Przyszłe sieci mają działać w przewidywalny sposób, zapewniając gwarantowane zasoby użytkownikom. Bezpieczeństwo ma też zapewniać szyfrowanie odporne na technologie kwantowe.

Istnieje wiele protokołów wymiany kluczy, większość wspomniano w [26]. Opisano tam też obecne osiągnięcia w parametrach sieci QKD, oraz wymieniono standardy opisujące protokoły i użyte technologie komunikacyjne.

Rodzaje bramek kwantowych

Obliczenia kwantowe w QPU opartym na bramkach, wykorzystują pewien zestaw operacji na pojedynczych kubitach lub ich grupach. Bramki te dokonują transformacji stanu kubitu do nowego stanu, co można opisać za pomocą algebry liniowej i macierzy. Taki opis znajduje się na końcu tej pracy. Wszystkie operacje z wyjątkiem operacji mierzenia stanu kubitu są odwracalne, tak jak odwracalne są zdarzenia w świecie kwantowym. Operacje te działają także w przypadku gdy kubity wejściowe są w superpozycji (reprezentują jednocześnie kilka różnych stanów) co można interpretować jako obliczenia równoległe na wszystkich możliwych przypadkach. Jedyną nieodwracalną operacją jest READ, która niszczy информацию kwantową i zwraca jeden bit klasycznej informacji z każdego kubitu.

Stan kubitu $|\psi\rangle$ można opisać jako superpozycję 2 stanów, 0 i 1. Dopóki nie odczytamy kubitu, oba stany mogą współistnieć jednocześnie.

$$|\psi\rangle = \alpha|0\rangle + e^{i\phi}\beta|1\rangle$$

gdzie spełniona jest zależność:

$$\alpha^2 + \beta^2 = 1$$

α i β to tak zwane amplitudy prawdopodobieństwa, a ich kwadraty określają prawdopodobieństwo odczytu z kubitu danego stanu. Kąt Φ to faza względna, która nie ma wpływu na prawdopodobieństwo odczytu określonego stanu, ale ma znaczenie przy innych operacjach. W szczególnym przypadku, jeżeli α lub β wynosi 0, to kubit jest w stanie zdeterminowanym, to znaczy, że stan kubitu można przewidzieć z całkowitą pewnością. Pojedynczy kubit posiada 2 takie stany, zaś rejestr N kubitów posiada 2^N takich stanów odpowiadających wszystkim binarnym kombinacjom zer i jedynek.

Znając te podstawy można poznać najpopularniejsze bramki wymienione poniżej. Każda bramka może być matematycznie opisana jako operacja macierzowa bramki i stanu kubitu lub kubitów. Jako wynik, otrzymamy nowy stan kubitu lub kubitów. Zatem może istnieć macierz odwrotna, która przywróci stan pierwotny (wycofa zmiany wprowadzone przez poprzednią bramkę).

Bramka I, jest bramką identycznościową, nie zmienia stanu kubitu.

Bramka X albo NOT, dokonuje negacji stanu jednego kubitu. Zamienia miejscami α i β , oraz odwraca fazę Φ na kąt przeciwny. Podwójne użycie tej bramki odwraca stan z powrotem do stanu początkowego.

Odmianą tej bramki jest bramka CNOT działająca na 2 kubitach. Jeden z nich to kubit kontrolny, natomiast na drugim wykonywana jest operacja negacji. Jeżeli kubit kontrolny jest w superpozycji, to bramka jednocześnie neguje i nie neguje drugi kubit, tworząc na wyjściu superpozycję obu wyników.

Jest też bramka CCNOT, znana jako bramka Toffoliego. Różni się od CNOT tym, że ma dwa, a nie jedno wejście kontrolne. Działa (neguje kubit) gdy wejścia kontrolne są w stanie $|11\rangle$ lub w superpozycji gdzie ten stan jest możliwy.

Kolejną ważną bramką jest bramka Hadamarda, oznaczana jako H lub HAD. Umożliwia ona wprowadzenie kubitu w stan superpozycji. Jest też częścią iteracji Grovera. Stan superpozycji, to stan kubitu lub grupy kubitów, który nie jest jednoznacznie zdeterminowany (odczyt może dać różne wyniki). Bramka H użyta na kubicie w stanie $|0\rangle$, powoduje zmianę tego zdeterminowanego stanu na "sumę" (superpozycję) stanów $|0\rangle$ i $|1\rangle$ w równych proporcjach. Użyta drugi raz, przywraca stan poprzedni, zatem użyta podwójnie, nie zmienia stanu kubitu.

Spośród bramek operujących na fazie można wymienić PHASE nazywana też ROTZ. Ta bramka zmienia fazę względową kubitu o dowolny kąt. Aby odwrócić jej działanie, wystarczy użyć jej ponownie zmieniając kąt w przeciwną stronę.

Istnieją też bramki ROTX i ROTY, zmieniające kąt na sferze Blocha w pozostałych 2 płaszczyznach, prostopadłych do ROTZ. (Sfera Blocha to graficzna reprezentacja stanu kubitu jako punktu na jej powierzchni).

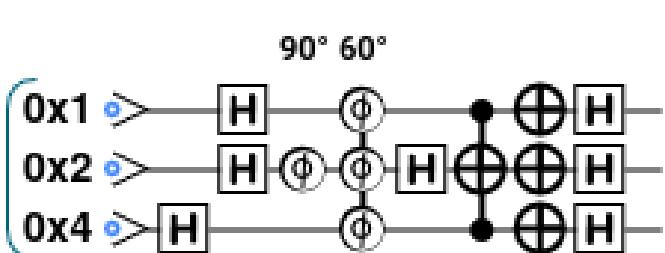
Istnieje operacja CPHASE, która zmienia fazę względową o zadany kąt w rejestrze kubitowym, dla wszystkich stanów w których użyte kubity mają wartość $|1\rangle$.

Operacje SWAP i CSWAP pozwalają wzajemnie zamienić stany 2 kubitów. Ta druga robi to warunkowo w zależności od kubitu kontrolnego.

Na koniec istnieje operacja, która ma różne nazwy, a ustawia kubit lub grupę kubitów w pożądanym stanie początkowym, takim jak $|0\rangle$ czy $|1\rangle$.

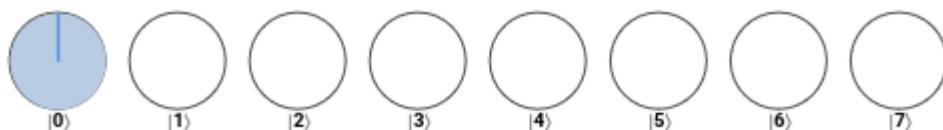
Warto zwrócić uwagę na to, że nie wszystkie QPU obsługują wszystkie typy bramek, ze względu na szczegóły fizycznej realizacji obliczeń. Z drugiej strony pewne bramki, np swap mogą być zastąpione serią prostszych bramek. Także niektóre z bramek są trudne w implementacji fizycznej, lub ich parametry są gorsze niż realizacja alternatywna. Istnieje więc pole do optymalizacji programów kwantowych pod konkretny sprzęt.

```
qc.reset(3);
qc.write(0);
qc.had(4);
qc.had(3);
qc.phase(90, 2);
qc.cphase(60);
qc.had(2);
qc.cnot(2, 5);
qc.not();
qc.had();
```

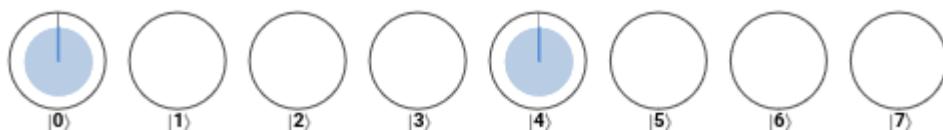


(Rys. 19) Przykładowy kod instrukcji kwantowych i stworzony obwód, źródło [20]

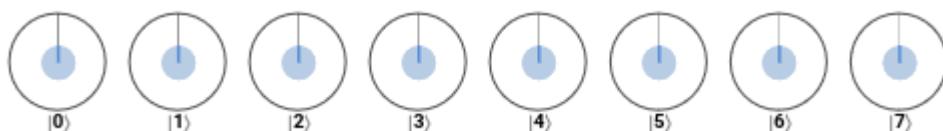
Powyższy kod to JavaScript/QC Engine. Jest przeznaczony do opisywania obwodów kwantowych w symulatorze [20]. Elementy oznaczają operacje na jednym lub kilku kubitach. Stan rejestru jest wizualizowany za pomocą wykresu kołowego. Stopień wypełnienia koła to amplituda prawdopodobieństwa określonego stanu rejestru. Kreski oznaczają fazy relatywne poszczególnych stanów. Dodatnie kąty to obrót w lewo. Prześledźmy jak zmienia się stan rejestru złożonego z 3 kubitów w każdym kroku. Ponieważ mamy 3 kubity, zatem możemy odczytać $2^3=8$ różnych kombinacji zer i jedynek. Dlatego wykres zawiera 8 kół.



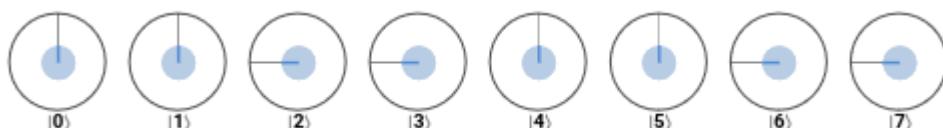
Rejestr zainicjalizowany stanem $|000\rangle$



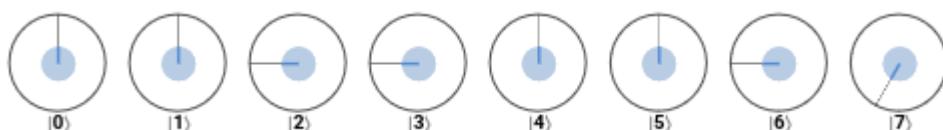
Hadamard na kubicie 0x4 - rejestr w superpozycji stanów $|100\rangle$ i $|000\rangle$



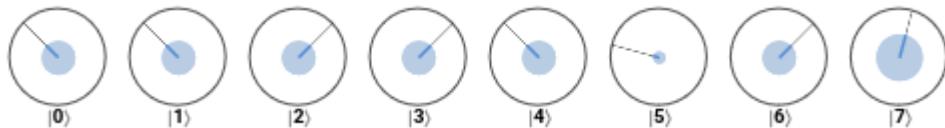
Hadamard na 0x3 (2 pozostałe kubity) - teraz rejestr jest w każdym możliwym stanie



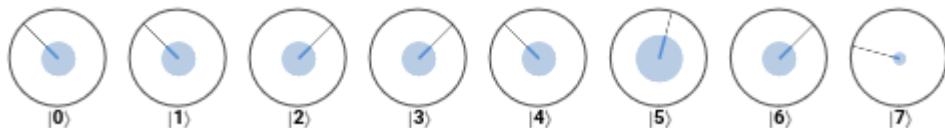
Kubit 0x2: PHASE(90) - Wszystkie stany w których środkowy kubit przyjmuje wartość 1 otrzymują relatywną rotację fazy o 90 stopni. Jest tak, ponieważ relatywna rotacja fazy nie działa na stan $|0\rangle$, a działa na $|1\rangle$. W tym konkretnym przypadku oznacza to stany pasujące do wzorca $|*1*$ (gwiazdka to 0 lub 1).



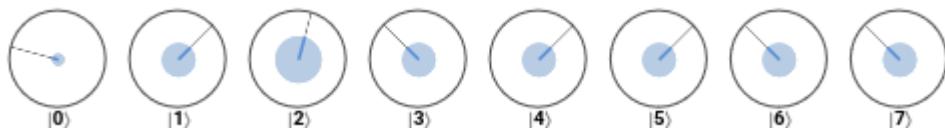
CPHASE na 3 kubitach, warunkowy obrót o kolejne 60 stopni tylko dla stanu $|111\rangle$



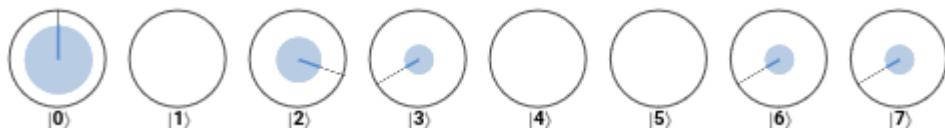
Operacja HAD na kubicie 2. Widać z wykresu, że kubity uległy splątaniu. Splątanie często powstaje przy operacjach warunkowych. Najprościej je poznać po utracie symetryczności czy regularności, albo po korelacji między wartością kubitów prowadzącej do rozkładu przypominającego XOR lub XNOR tak jak na rysunku 17.



Operacja CCNOT neguje kubit 2, stany $|101\rangle$ i $|111\rangle$ zamieniają się miejscami. (tylko stany w których kubity kontrolne mają wartość 1)



Negacja całego rejestru odwraca kolejność kół



Operacja Hadamarda zmienia stan rejestru. Kubity pozostają splątane. Teraz jeżeli któryś z kubitów po odczycie ma stan 0, to jest duże prawdopodobieństwo, że pozostałe kubity też będą odczytane jako 0. Jest tak ponieważ stan $|000\rangle$ ma wysoką amplitudę prawdopodobieństwa. Amplituda prawdopodobieństwa to promień wewnętrznego koła, a prawdopodobieństwo odczytania z kubitów danego stanu wyrażone jest przez pole wewnętrznego koła.

(Rys. 20-28) Wykresy kołowe stanu, symulator obliczeń kwantowych, źródło [20]

To prosta demonstracja jak działają bramki. Wykres kołowy pozwala na przybliżenie czytelnikowi co dzieje się ze stanem kubitów. Jest to uproszczony zapis. Alternatywnie, można posługiwać się zapisem macierzowym z liczbami zespolonymi. Taki zapis jest bardziej dokładny, ale każdą operację na N kubitach, trzeba wyrazić macierzą kwadratową o rozmiarze $2^N \times 2^N$. Macierzowy opis kubitów i bramek jest zawarty na końcu tej pracy. Oprócz prostych bramek, istnieją też bardziej zaawansowane operacje. Cechuje je wszystkie odwracalność, to znaczy że żadna informacja nie jest tracona, ilość kubitów przyjmowanych jest równa ilości kubitów zwracanych. Jedną z nich jest wspomniana wcześniej bramka Toffoliego. Pozostałe 3-kubitowe bramki wymieniono w tabeli. A, B i C to stany kubitów na wejściu, a P, Q i R to odpowiednio stany tych kubitów na wyjściu.

Gate	P	Q	R
Fredkin [27]	A	$\bar{A}B+AC$	$AB+\bar{A}C$
SFV [20]	$A \oplus B$	$B \oplus C$	$\bar{A}B+BC+\bar{A}C$
TS-3 [33]	A	B	$A \oplus B \oplus C$
RM [34]	$A \oplus BC$	$\bar{A}B+AC$	$\bar{A}C+AB$
RG [35]	$A \oplus B$	$AB+BC+AC$	$\bar{B}C+A(\bar{B} \oplus C)$
RG1 [21]	\bar{B}	$A\bar{B}+BC$	$A \oplus C$
G1 [36]	\bar{B}	$\bar{A}\bar{B}+AB$	$\bar{A}\bar{B}C+A\bar{C}+B\bar{C}$
RG2 [21]	$\bar{A}\bar{B} \oplus C$	$\bar{A} \oplus \bar{B}$	A
URLG [37]	$AB+BC+AC$	$\bar{A}B+\bar{B}\bar{C}+\bar{A}\bar{C}$	$A\bar{B}+\bar{B}\bar{C}+\bar{A}\bar{C}$
MPG [38]	A	$A \oplus B$	$A\bar{B} \oplus C$
SAM [38]	\bar{A}	$\bar{A}B \oplus A\bar{C}$	$\bar{A}C \oplus AB$
R [34]	$A \oplus B$	A	$\bar{C} \oplus (AB)$
QCA1 [39]	$AB+BC+AC$	$AB+\bar{B}\bar{C}+A\bar{C}$	$\bar{A}B+BC+\bar{A}C$
QCA2 [39]	$AB+BC+AC$	$AB+\bar{B}\bar{C}+AC$	$\bar{A}B+\bar{B}\bar{C}+\bar{A}\bar{C}$
RQG [40]	$AB+BC+AC$	$\bar{A}B+BC+\bar{A}C$	$A \oplus C$
Peres [41]	A	$A \oplus B$	$AB \oplus C$
URG [42]	$(A+B) \oplus C$	B	$AB \oplus C$
SSG [43]	$B \oplus C$	$A \oplus B$	$AB+BC+AC$
MNFT [44]	$(A \oplus B)C \oplus A$	$A \oplus B$	$(A \oplus B)C \oplus A \oplus C$
SRK [45]	\bar{A}	$A \oplus B \oplus C$	$\bar{A}C \oplus AB$
SSG-1 [46]	$A \oplus B$	$\bar{B} \oplus C$	$AB+BC+AC$
NNG [47]	C	$AB+\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}$	$AB+A\bar{C}+\bar{A}\bar{B}C$
IMG [48]	B	$\bar{A}C+A(BC+\bar{B}\bar{C})$	\bar{A}
TR [49]	A	$A \oplus B$	$A\bar{B} \oplus C$
RUG [50]	$AB+BC+AC$	$AB+\bar{A}\bar{C}$	$B \oplus C$

(Rys. 29) 3-kubitowe bramki odwracalne [29]

Znaczek plusa w kółku oznacza logiczną operację EXCLUSIVE OR, natomiast kropka w kółku oznacza operację EXCLUSIVE OR, której wartość wyjściowa została dodatkowo zanegowana. Kreska góra również oznacza zanegowanie wartości wyrażenia pod nią. Jak można zauważyć, bramki wymienione w tabeli wykorzystują logikę Boola. Mogą być więc zaimplementowane także w logice binarnej. Kwantowe bramki natomiast nie muszą ograniczać się do stanów 0 i 1, lecz mogą operować na superpozycji stanów wejściowych, zwracając superpozycję odpowiednich (wyrażonych wzorami z tabeli) stanów wyjściowych.

NG [51]	A	$\overline{AB} \oplus C$	$\overline{A} \overline{C} \oplus \overline{B}$
DG [51]	A	$\overline{A \oplus B}$	$\overline{AB} \oplus C$
RMG [30]	$A \oplus BC$	$\overline{AB} + AC$	$\overline{AC} + AB$
Toffoli [12]	A	B	$AB \oplus C$
MCL [52]	$\overline{B} \overline{C}$	$\overline{A} \overline{B}$	A
GI [36]	\overline{B}	$\overline{A} \overline{B} + AB$	$\overline{A} \overline{BC} + \overline{AC} + B\overline{C}$
BJN [53]	A	B	$(A+B) \oplus C$
HAS [54]	A	$A \oplus B \oplus C$	$AB + \overline{AC}$
New Gate [55]	A	$AB \oplus C$	$\overline{A} \overline{C} \oplus \overline{B}$
ORG-I [56]	$AB + (A \oplus B)C$	$A \oplus B$	$A\overline{B} + (A \oplus \overline{B})$
ORG-II [56]	$A\overline{B} + BC$	$\overline{AB} + \overline{BC}$	$AB + \overline{BC}$
MF [57]	A	$\overline{AB} + A\overline{C}$	$AB + \overline{AC}$
BG-1 [58]	$A \oplus B$	$B \oplus C$	C
GB-1 [58]	$A \oplus B \oplus C$	$B \oplus C$	C
NG-R2 [59]	A	$A \oplus B$	$(A+B) \oplus C$
DG [60]	A	$\overline{A \oplus B}$	$\overline{AB} \oplus C$
RSG [61]	$A \oplus B$	$AB \oplus C$	$\overline{AB} \oplus C$
TKS [62]	$A\overline{C} + BC$	$A \oplus B \oplus C$	$AC + B\overline{C}$
NCT [63]	A	B	$\overline{AB} \oplus C$
NRLG [64]	A	$A \odot B$	$\overline{AB} \oplus C$
S ₁ G [65]	$\overline{AB} \oplus C$	$A \odot B$	$\overline{AB} \odot C$
S ₂ G [65]	$AB \oplus C$	$A \oplus B$	$AB \oplus C$
FRSG-1[66]	A	$A \odot B$	$AB \oplus C$
JTF1[66]	A	$A \oplus B$	$A \oplus B \oplus C$
MG[67]	A	$A \oplus B \oplus C$	$\overline{AC} \oplus AB$
PRG[68]	$A \oplus B \oplus C$	$AC + \overline{B} \overline{C}$	$AC + B\overline{C}$
NFT[69]	A	$\overline{BC} \oplus \overline{AC}$	$BC \oplus A\overline{C}$

(Rys. 29 c.d.) 3-kubitowe bramki odwracalne [29]

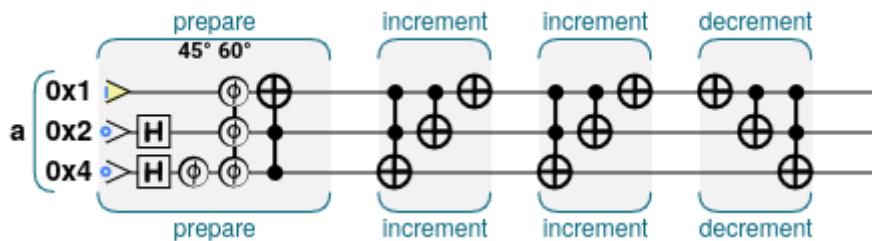
Aby bramka była odwracalna wystarczy że dla każdego zestawu danych wejściowych (wartości poszczególnych kubitów) istnieje dokładnie jeden zestaw danych wyjściowych. Można traktować bramkę jako funkcję przyjmującą argumenty wejściowe i zwracającą parametry wyjściowe. Dla tej funkcji istnieje funkcja odwrotna. Abstrahując od funkcji i bramek, ze względu na właściwości mechaniki kwantowej, wszystkie obliczenia kwantowe są odwracalne (oprócz READ). Wystarczy wykonać obwód kwantowy w odwrotnej kolejności aby cofnąć obliczenia do pierwotnego stanu.

Prymitywy kwantowe

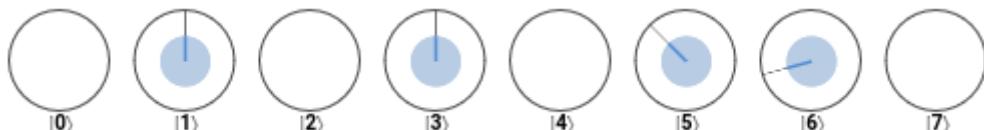
Skoro znamy już proste bramki to można poznać bardziej złożone operacje takie jak kwantowa transformata Fouriera, kwantowe szacowanie fazy czy wzmacnianie amplitudy z użyciem algorytmu Grovera. Istnieją też prymitywy pozwalające na proste operacje arytmetyczne, takie jak inkrementacja/dekrementacja. Informację można też kodować w fazie względnej. Prymitywy kwantowe to warstwa pośrednia między podstawowymi bramkami a złożonymi aplikacjami [27].

Inkrementacja w obliczeniach kwantowych w odróżnieniu od wersji klasycznej dotyczy jednocześnie wszystkich możliwych stanów rejestru. Najszybciej wyjaśni to (Rys. 30-34).

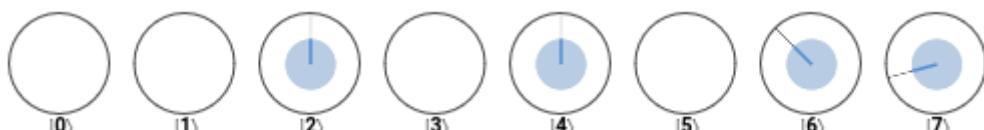
Zakodowaliśmy w sekcji prepare, w 3-kubitowym rejestrze, 4 liczby za pomocą amplitud prawdopodobieństwa.



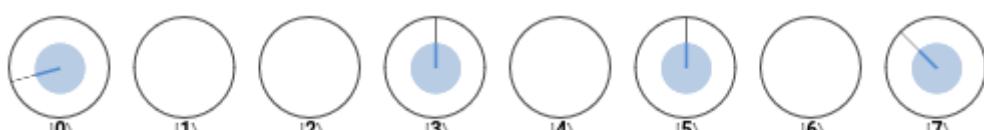
(Rys. 30) Obwód testowy [20]



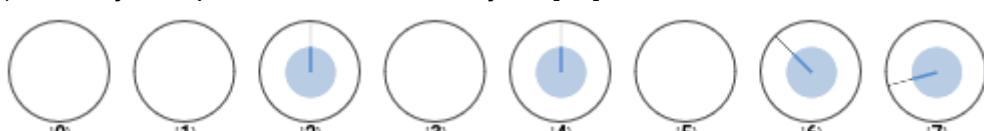
(Rys. 31) Przygotowany stan początkowy rejestru [20]



(Rys. 32) Stan rejestrzu po pojedynczej inkrementacji [20]



(Rys. 33) Stan rejestrzu po dwóch inkrementacjach [20]



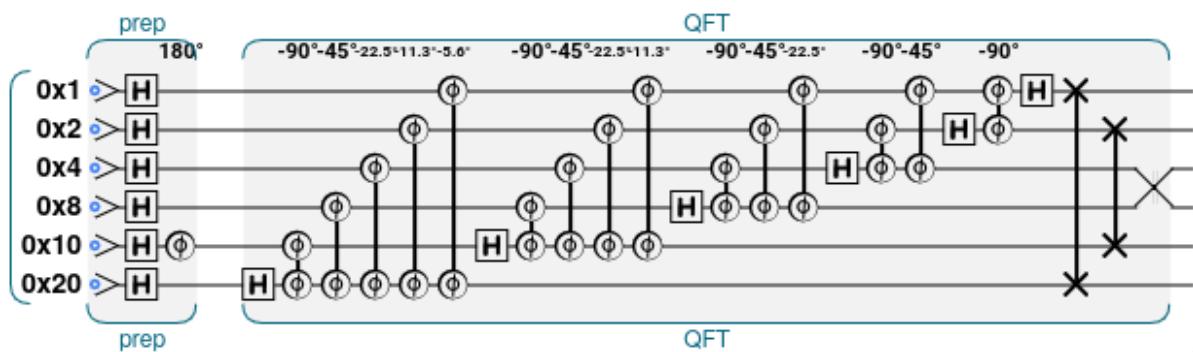
(Rys. 34) Stan rejestrzu po dekrementacji [20]

Ten przykład pokazuje jednocześnie kilka faktów. Po pierwsze, kwantowa inkrementacja inkrementuje w superpozycji każdy występujący stan. Patrząc na wykres kołowy, zawartości

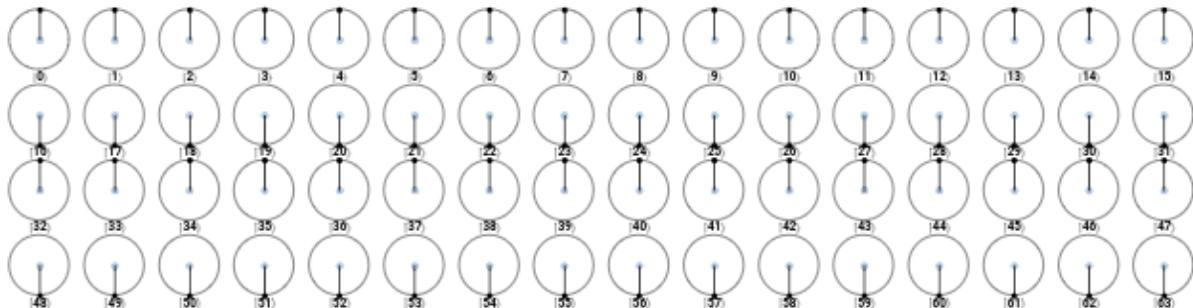
kół są przesuwane o 1 miejsce w prawo. Ostatnie koło z prawej strony zostaje przeniesione na początek. Stan początkowy został tak przygotowany aby było to łatwo zauważalne. Po drugie, wiemy że dekrementacja jest odwrotnością inkrementacji, czyli przywraca stan do poprzedniego stanu (porównaj Rys. 32 i 34). Zatem dekrementacja cofa efekt inkrementacji. Stąd wniosek że instrukcje realizujące dekrementację powinny być takie same jak do inkrementacji, ale zapisane w odwrotnej kolejności. Patrząc na zapis obwodu, widzimy że rzeczywiście tak jest. Obliczenia kwantowe są odwracalne, wystarczy wykonać program od tyłu.

Kolejna istotna sprawa to prawidłowe używanie tak zwanych kubitów skreżowych [27]. Założymy że chcemy obliczyć wartość bezwzględną, to znaczy jeżeli rejestr może reprezentować liczby zarówno dodatnie jak i ujemne (w kodzie U2), po operacji otrzymamy tylko liczby dodatnie. W obliczeniach kwantowych jest to problem, gdyż tracimy informacje o znaku, a zatem nie możemy odwrócić obliczeń. Użycie dodatkowego kubitu, w którym przechowamy znak oryginalnej liczby pozwoli spełnić wymog odwracalności obwodu. Taki kubit staje się jednak spłaty z rejestrów na których wykonano operację obliczania wartości bezwzględnej. Jeżeli będziemy chcieli użyć go później do innych operacji, to wpłynie to na stan rejestrów z wynikiem. Aby tego uniknąć należy rozwiązać kubit skreżowy przez cofnięcie obliczeń wartości bezwzględnej po użyciu wyniku.

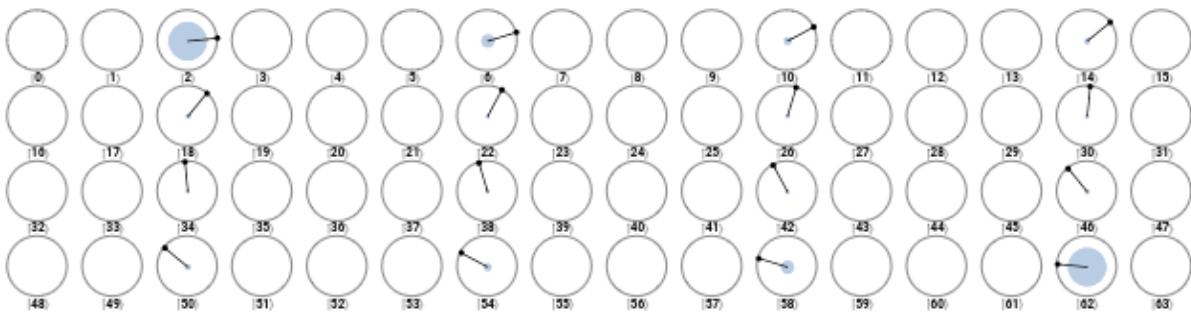
Kwantowa transformata Fouriera (QFT) to ciekawy prymityw wzorujący się na dyskretnej transformacie Fouriera (DFT). Przeprowadzenie transformaty na rejestrze zawierającym informację o próbkach jakiegoś sygnału zakodowanych w stanach kwantowych, pozwala otrzymać widmo sygnału w dziedzinie częstotliwości. QFT jest szybsze niż klasyczne DFT dla rejestrów o długości powyżej 22 kubitów. QFT pozwala wyszukiwać okresowość, a to przydaje się m.in. przy faktoryzacji liczb w algorytmie Shora.



(Rys. 35) Obwód transformaty Fouriera [20]



(Rys. 36) Przygotowany sygnał fali prostokątnej zakodowany w modułach i fazach [20]



(Rys. 37) Widmo sygnału zakodowane w modułach i fazach [20]

```

qc.reset(6);
qc.label('prep');
qc.write(0);
qc.had();
qc.phase(180, 16);

qc.label('');
qc.nop();

qc.label('QFT');
qc.QFT();

```

(Rys. 38) Kod QCEngine [20]

Na rysunku 35 przedstawiono obwód QFT oraz wstępne przygotowanie sygnału prostokątnego. Numery kubitów oznaczono w pozycyjnej notacji heksadecymalnej (liczba hex oznacza pozycję bitu w rejestrze). Na rysunku 36 widać przygotowanie fali prostokątnej o 2 cyklach, zakodowanej w stanach rejestru będącego w superpozycji. Dla stanów w których bit 0x10 jest równy 1, wartość sygnału prostokątnego jest ujemna. Wybrałem rozmiar rejestru 6 kubitów. 6-kubitowy rejestr może przechowywać $2^6=64$ próbek sygnału. Amplituda prawdopodobieństwa wyrażona promieniem wewnętrznego koła wyraża moduł próbki, natomiast faza relatywna koduje kąt liczby zespolonej. (W tym przypadku sygnał ma tylko wartości rzeczywiste, kąt 0° oznacza wartość dodatnią sygnału, a 180° ujemną. Amplituda jest zatem stała, zmienia się tylko faza na przeciwną). QFT zapewnia przejście od opisu sygnału w dziedzinie czasu, do opisu w dziedzinie częstotliwości. Po QFT widać że sygnał składa się głównie ze składowej 2 oraz -2. Rzeczywiście w rejestrze mieścią się 2 cykle fali prostokątnej. Widać też nieparzyste składowe harmoniczne, charakterystyczne dla symetrycznego sygnału prostokątnego.

Możemy też zrobić odwrotność QFT, aby wygenerować sygnał składający się z wybranych częstotliwości. Kod przygotowuje 6-kubitowy rejestr w stanie opisującym widmo częstotliwości jako złożone z pierwszej i trzeciej harmonicznej, w plusie jak i w minusie, dzięki czemu uzyskamy sygnał w czasie, w dziedzinie liczb rzeczywistych, o amplitudzie prawdopodobieństwa określającej moduł próbki, a fazie określającej jej znak. Ze względu na ograniczenia symulatora, dodano funkcję uzupełniającą odczyty prawdopodobieństwa o znak. Wartość odczytanego prawdopodobieństwa to kwadrat amplitudy prawdopodobieństwa, więc otrzymane wartości próbek określają chwilową moc sygnału. Kod został opisany komentarzami.

```

// inicjalizacja rejestr
var num_qubits = 6;
qc.reset(num_qubits);
var qin = qint.new(num_qubits, 'qin');

qc.label('write freq');
qin.write(0);

// tworzenie 4 składowych w dziedzinie częstotliwości
qin.had(34);
// ustawienie składowych w odpowiednich miejscach
qin.not(1);
qin.cnot(31,32);
qin.cnot(2,33);
qin.cnot(1,32);
qc.label('');
qc.nop();

// właściwe QFT (odwrócone)
qc.label('invQFT');
qin.invQFT()

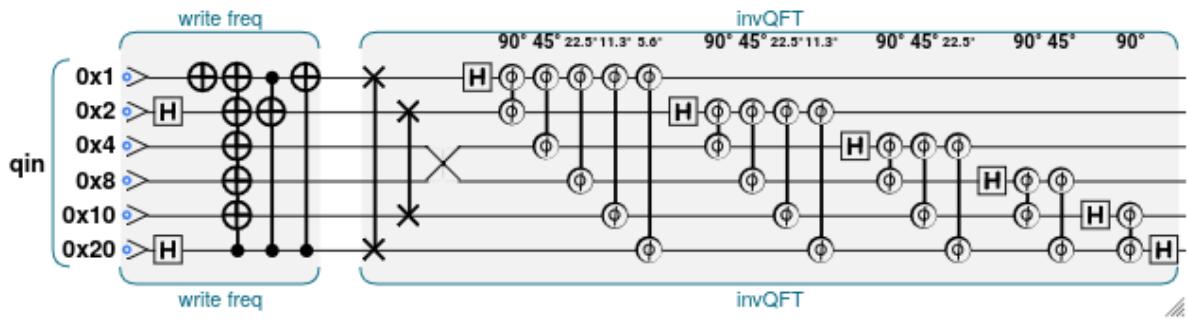
// funkcje pomocnicze niezwiązane z przetwarzaniem sygnału
function checkSign(number) {
    if ((number >= 8 && number <= 15) || (number >= 24 &&
number <= 39) || (number >= 48 && number <= 55)) {
        return "-";
    } else {
        return "";
    }
}

for (var i = 0; i < 2**num_qubits; ++i)
{
    qc.print(i+";"+checkSign(i)+qin.peekProbability(i)+"\n");
}

```

(Rys. 39) Kod QCEngine, opisujący użycie odwrotnej transformaty Fouriera [20]

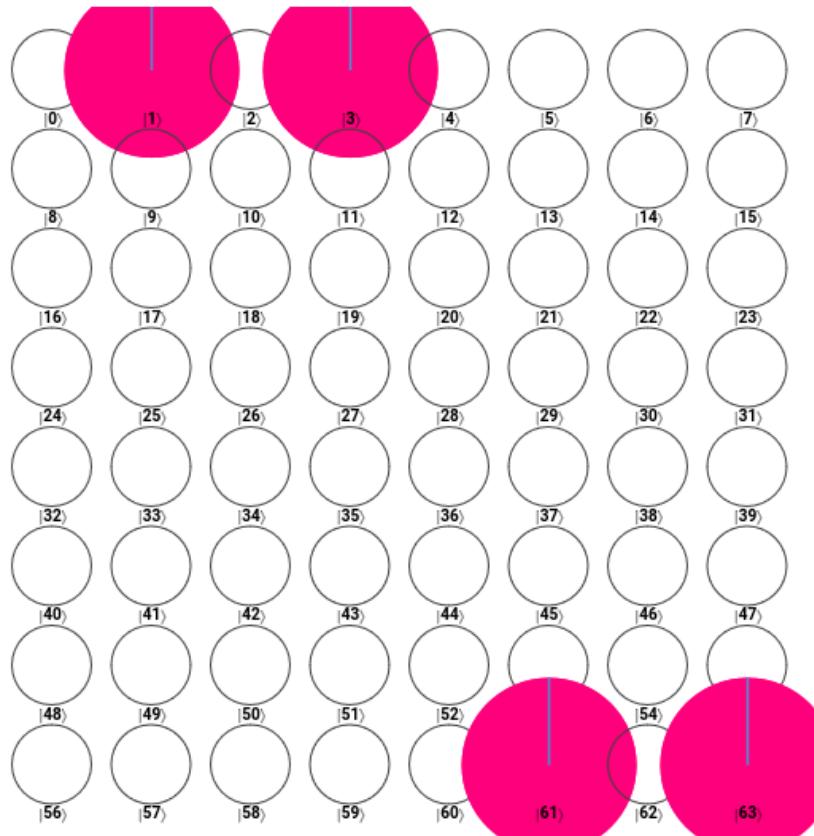
Obwód tworzy rejestr w odpowiedniej superpozycji 4 stanów, o tej samej amplitudzie każdy. Następnie wykonywane jest odwrotne QFT, będące wcześniej opisanym QFT ale zapisanym od tyłu. Na końcu programu, rejestr kubitowy zawiera próbki sygnału zakodowane amplitudowo i fazowo.



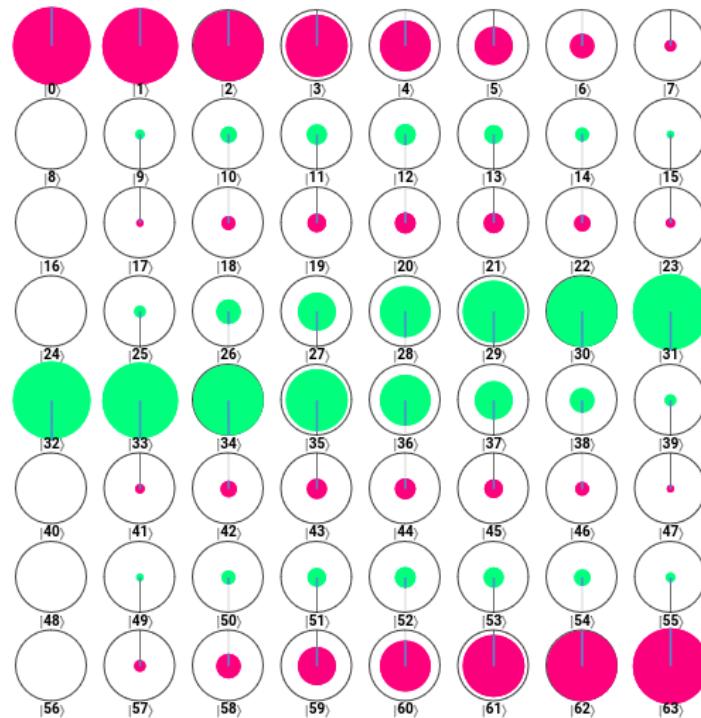
(Rys. 40) Odwrotna transformata Fouriera - tworzenie przebiegu w czasie [20]

Aby przebieg był lepiej widoczny, amplitudy zostały dodatkowo wzmacnione, przez co stopień wypełnienia koła nie jest skalibrowany do rzeczywistej amplitudy prawdopodobieństwa. Jest to celowy zabieg, gdyż działanie wykresu kołowego zademonstrowano już wcześniej. Ponadto wprowadzono kolory wyrażające znak wartości sygnału.

Różowy - faza 0° - sygnał dodatni
 Zielony - faza 180° - sygnał ujemny



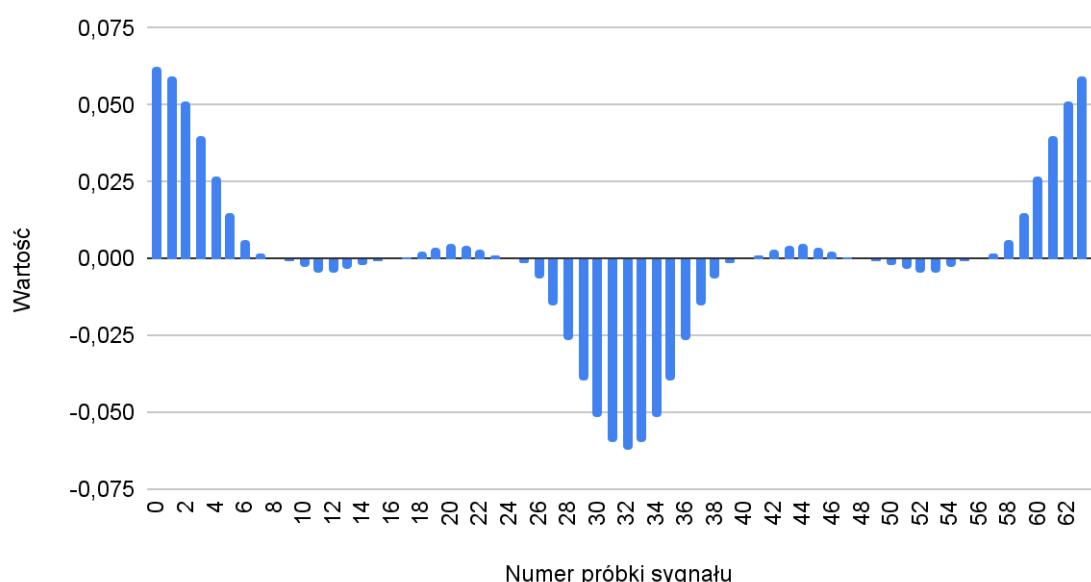
(Rys. 41) Przygotowane widmo sygnału [20]



(Rys. 42) Przebieg sygnału w czasie [20]

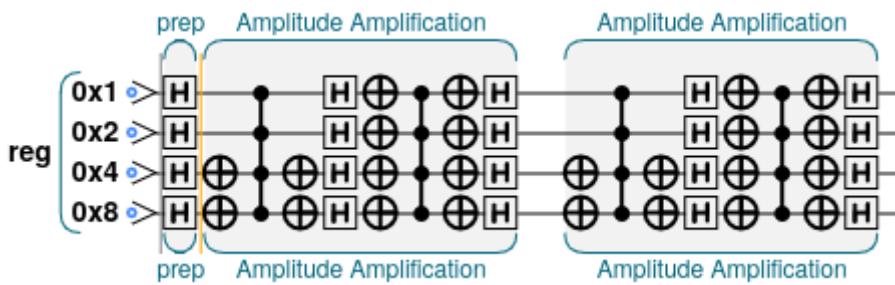
Aby lepiej jeszcze zwizualizować przebieg w czasie, wykorzystano dane liczbowe wygenerowane przez program, do zbudowania tradycyjnego wykresu $A^2(t)$. A to amplituda prawdopodobieństwa stanu opisującego N-tą próbkę otrzymanego sygnału, określająca amplitudę sygnału w tej próbce. A^2 to chwilowa moc sygnału. Wartość tej mocy przedstawia wykres. Gdyby cała moc sygnału była skupiona w pojedynczej próbce, to otrzymalibyśmy wartość równą 1. Możemy to sprawdzić, dodając wszystkie wartości bezwzględne do siebie.

Wartość a Numer próbki sygnału

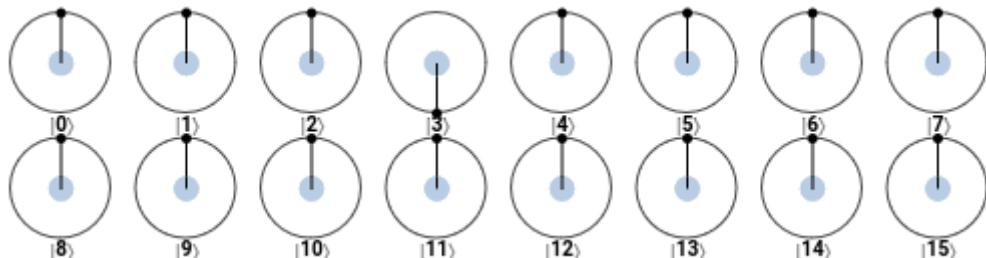


(Rys. 43) Wykres z danych liczbowych (źródło: opracowanie własne)

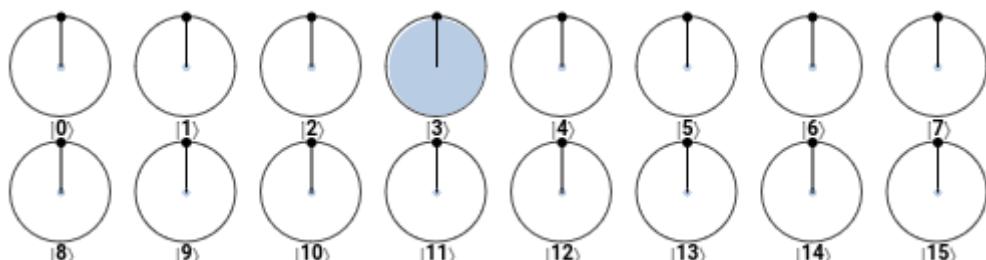
Wzmocnienie amplitudy (AA) jest innym prymitywem, którego działanie jest warte przyjrzenia się bliżej. Ten prymityw służy do zamiany faz względnych, które nie są odczytywalne, do rozkładu prawdopodobieństw stanów, który możemy poznać przez (wielokrotny) odczyt kubitów. Iteracja Grovera składa się z kroku oznaczania jednego ze stanów za pomocą odwrócenia fazy, a następnie wykonywana jest operacja "lustro", która zwiększa amplitudę oznaczonego stanu, zmniejszając jednocześnie pozostałe amplitudy i zerując oznaczenia. Optymalna ilość iteracji zależy od wielkości rejestru oraz ilości oznaczonych w ten sposób stanów. Zbyt duża ilość iteracji powoduje pogarszanie (osłabianie) pożądanych amplitud. W rzeczywistości, wielkość wzmocnienia wahą się cyklicznie od 0 do 100% w funkcji ilości iteracji. Częstotliwość tych oscylacji zależy jak już wspomniano także od ilości oznaczonych stanów. A pomiaru częstotliwości umiemy już dokonać za pomocą transformaty Fouriera. Zatem używając AA i QFT możemy szacować ilość oznaczonych stanów. Nazywa się to szacowaniem sumy.



(Rys. 44) Obwód wykorzystujący wzmocnienie amplitudy [20]



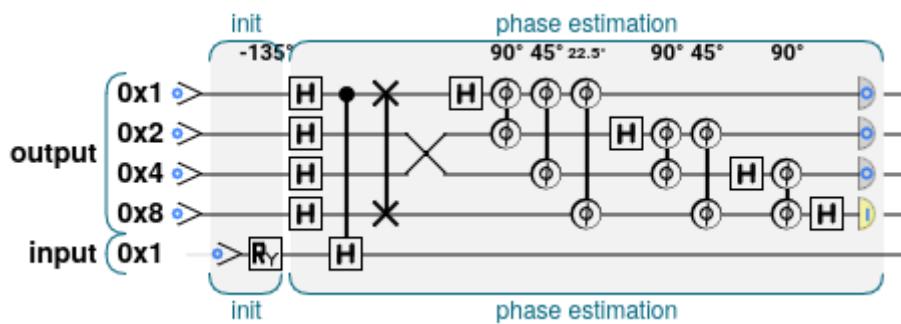
(Rys. 45) Oznaczyliśmy $|3\rangle$ [20]



(Rys. 46) Stan $|3\rangle$ posiada maksymalną amplitudę [20]

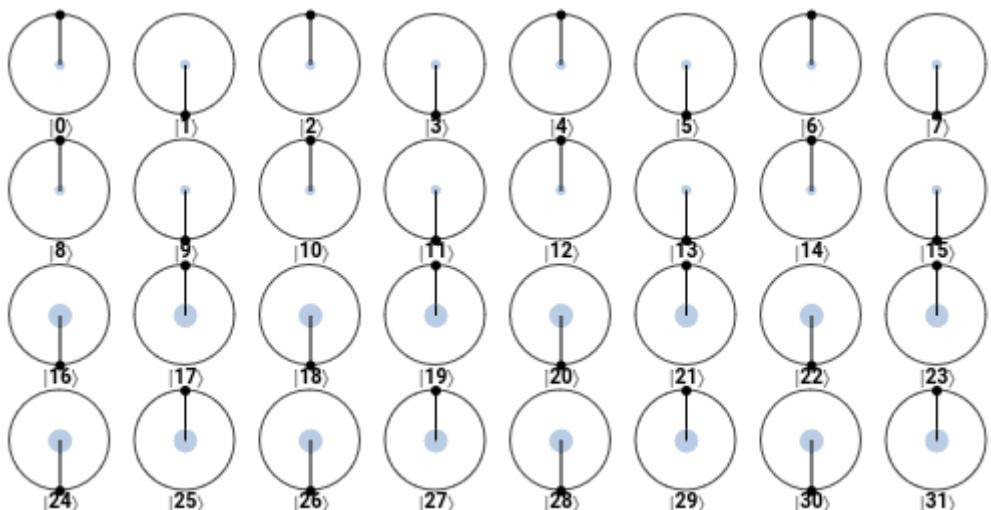
Szacowanie fazy to inny rodzaj prymitywu, użyteczny przy uczeniu maszynowym. Każda operacja, taka jak na przykład bramka Hadamarda, posiada zestaw stanów własnych, które

nie są przez nią modyfikowane w odróżnieniu od wszystkich pozostałych. Z perspektywy obliczeń na macierzach, są to jej wektory własne. Ponadto każdemu stanowi własnemu możemy przypisać fazę własną, czyli to jak bramka zmienia fazę globalną. Fazy globalne w odróżnieniu od faz relatywnych nie są widoczne bezpośrednio w wynikach obliczeń. Szacowanie fazy pozwala jednak wydobyć i tą informację, poprzez połączenie kilku prymitywów. Aby to zrobić, wykorzystuje się kontrolowaną wersję badanej operacji działającej na jednym z jej stanów własnych. Do kontrolowania operacji wykorzystuje się dodatkowy rejestr będący w superpozycji. Chodzi o to aby badaną operację uruchomić tyle razy, ile wynosi stan rejestrów kontrolnych. Każde uruchomienie operacji powoduje zmianę fazy globalnej na rejestrze na którym operacja działa, oraz odbicie fazowe na rejestrze kontrolnym. W rezultacie rejestr kontrolny będzie zawierać fazy zmieniające się w sposób cykliczny w funkcji ilości uruchomień. A cykliczność zmian fazy potrafimy zmierzyć za pomocą znanej nam już odwrotnej transformaty Fouriera. W skrócie, warunkowość badanej operacji pozwala poznać jej fazę globalną w funkcji ilości uruchomień, gdyż faza globalna badanej operacji, staje się fazą relatywną w rejestrze kontrolnym. QFT tylko przelicza różnicę faz na każde dodatkowe uruchomienie, na postać wyrażoną w formie amplitudy prawdopodobieństwa, możliwą do łatwego odczytania. Rozdzielcość pomiaru fazy własnej zależy od rozmiaru rejestrów kontrolnych.

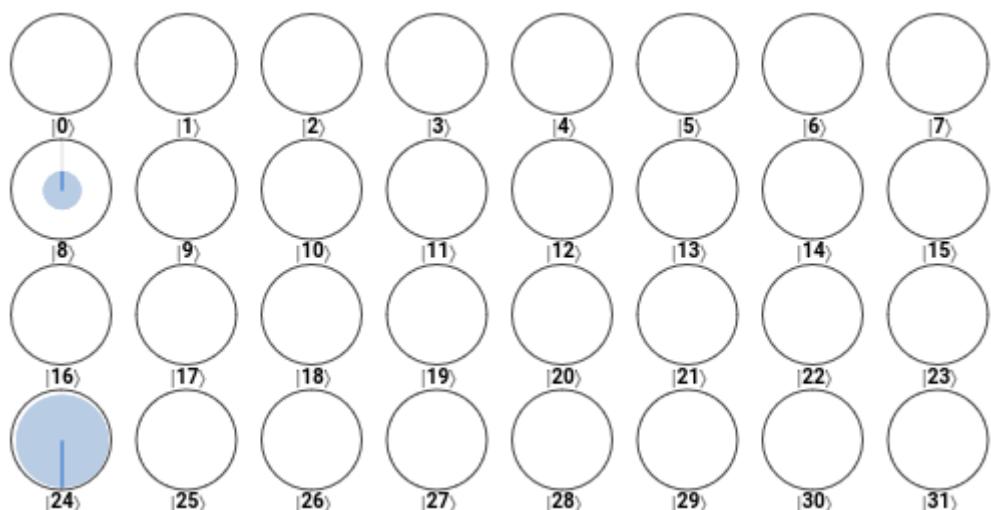


(Rys. 47) Obwód szacowania fazy globalnej dla operacji Hadamarda [20]

W tym obwodzie, wprowadzamy kubit w jeden ze stanów własnych operacji Hadamarda. Następnie wprowadzamy rejestr kontrolny w superpozycję. Dalej mamy warunkową operację Hadamarda. Operacja ta wykonana parzystą ilość razy nie wnosi nic. Natomiast wykonana nieparzystą ilość razy ma efekt taki sam jak wykonana pojedynczo. Dlatego wystarczy pojedyncza bramka H sterowana najmłodszym kubitem rejestrów kontrolnych. Reszta obwodu to odwrotna transformata Fouriera, zakończona odczytem rejestrów kontrolnych.



(Rys. 48) Wnoszona faza własna w funkcji ilości uruchomień operacji H [20]



(Rys. 49) Wyznaczona faza własna [20]

Obwód udzielił odpowiedzi $|8\rangle$, co należy tłumaczyć jako połowę zakresu 4-kubitowego rejestru kontrolnego, zatem faza własna bramki Hadamarda dla badanego stanu własnego wynosi $360/2 = 180$ stopni. Tak jest w rzeczywistości. Notabene, 2 szeregowo użyte bramki Hadamarda, dają fazę globalną 360 stopni, czyli nie zmieniają jej. Na (Rys. 49) widzimy dodatkowe koło, gdyż wykres pokazuje stan wszystkich kubitów, łącznie z wejściowym dla badanej operacji. Najstarszy kubit należy zignorować, dlatego $|8\rangle$ i $|24\rangle$ są tym samym stanem rejestru kontrolnego.

Tak oto opisane zostały najważniejsze prymitywy, wykorzystywane w bardziej złożonych aplikacjach. Pomogło to zademonstrować możliwości, specyfikę oraz sposób myślenia potrzebne do budowania aplikacji.

Algorytmy kwantowe

Prymitywy przedstawione wcześniej mogą być łączone i używane do rozwiązywania bardziej złożonych problemów. Jednym z nich jest badanie wyrażeń logicznych i poszukiwanie warunków spełnialności. Dla zwykłego komputera oznaczałoby to obliczenie tabeli prawdy dla każdej kombinacji wejść, co przy znacznej ich ilości jest problemem. Okazuje się, że komputer kwantowy może w superpozycji dokonać pomiaru wszystkich przypadków jednocześnie. W tym celu należy wyrazić po kolej wiązki warunki logiczne za pomocą bramek, a następnie obrócić fazę tych stanów rejestru, dla których wyrażenie logiczne jest prawdziwe. Na koniec, za pomocą iteracji Grovera można zamienić różnice fazowe na amplitudy prawdopodobieństw, które mogą zostać odczytane [27]. Jeżeli wyrażenie ma tylko jedno rozwiązanie, to zostanie ono znalezione ze 100% prawdopodobieństwem. Jeżeli jest więcej rozwiązań, to zostanie zwrócone losowo jedno z nich. Gdyby natomiast wyrażenie logiczne było zawsze fałszywe, to QPU zwróci losowe rozwiązanie, które jednak nie będzie prawdziwe. Dlatego jeżeli zachodzi podejrzenie, że wyrażenie logiczne może nie mieć wogół rozwiązania, należy wynik z QPU zweryfikować klasycznie aby zidentyfikować taki przypadek.

Obliczenia kwantowe mogą przydać się też w obróbce grafiki. Kwantowy supersampling pozwala na dokładniejsze skalowanie grafiki, z mniejszą ilością szumu niż analogiczny algorytm klasyczny.

Jednak najbardziej znanym algorymem dla komputerów kwantowych jest algorytm Petera Shora. Jest tak ponieważ pozwala on na szybką faktoryzację wielkich liczb, a to zagroża bezpieczeństwu infrastruktury klucza publicznego opartej o takie algorytmy jak RSA, DSA, ECDSA, ElGamal czy DH. Przyjrzyjmy się zatem mu bliżej.

Faktoryzacja liczb staje się łatwa jeżeli potrafimy szybko i łatwo odnaleźć okres po którym powtarzają się wartości funkcji

$$f(x) = a^x \bmod(N)$$

(Wzór 50) Faktoryzacja Shora [27]

gdzie:

N - faktoryzowana liczba złożona z 2 liczb pierwszych

a - liczba ko-pierwsza do N, czyli liczba pierwsza, która nie jest składnikiem N

mod - operator zwracający resztę z dzielenia przez N

Spróbujmy więc w Pythonie rozłożyć na czynniki liczbę N=205, wybierając liczbę a=2, gdyż jak widać nie jest ona czynnikiem N. Oto prosty kod Pythona, liczący f(x).

```
for i in range(30):
    print(f"iteracja {i}  reszta {(2**i)%205}")

iteracja 0  reszta 1
```

```

iteracja 1    reszta  2
iteracja 2    reszta  4
iteracja 3    reszta  8
iteracja 4    reszta 16
iteracja 5    reszta 32
iteracja 6    reszta 64
iteracja 7    reszta 128
iteracja 8    reszta 51
iteracja 9    reszta 102
iteracja 10   reszta 204
iteracja 11   reszta 203
iteracja 12   reszta 201
iteracja 13   reszta 197
iteracja 14   reszta 189
iteracja 15   reszta 173
iteracja 16   reszta 141
iteracja 17   reszta 77
iteracja 18   reszta 154
iteracja 19   reszta 103
iteracja 20   reszta 1
iteracja 21   reszta 2
iteracja 22   reszta 4
iteracja 23   reszta 8
iteracja 24   reszta 16
iteracja 25   reszta 32
iteracja 26   reszta 64
iteracja 27   reszta 128
iteracja 28   reszta 51
iteracja 29   reszta 102

```

(Rys. 51) Poszukiwanie okresowości dla $a=2$

Wartości powtarzają się gdy x zmieni się o $p=20$.

Teraz wyliczamy 2 liczby pomocnicze według wzoru:

$$k_1, k_2 = a^{p/2} \pm 1$$

(Wzór 52) Liczby k_1 i k_2

$$k_1=1025, k_2=1023$$

Kolejny krok to wyliczenie największego wspólnego dzielnika z (N, k_1) oraz (N, k_2) . Skorzystamy z kalkulatora [28]. Otrzymamy odpowiednio 205 oraz 1, co jest poprawnym wynikiem, ale nie jest on pozytyczny. Należy wybrać inną liczbę a .

Wybierzmy $a=3$, która również nie jest czynnikiem $N=205$

```

for i in range(30):
    print(f"iteracja {i}    reszta {(3**i)%205}")

```

```

iteracja 0 reszta 1
iteracja 1 reszta 3
iteracja 2 reszta 9
iteracja 3 reszta 27
iteracja 4 reszta 81
iteracja 5 reszta 38
iteracja 6 reszta 114
iteracja 7 reszta 137
iteracja 8 reszta 1
iteracja 9 reszta 3
iteracja 10 reszta 9
iteracja 11 reszta 27
iteracja 12 reszta 81
iteracja 13 reszta 38
iteracja 14 reszta 114
iteracja 15 reszta 137
iteracja 16 reszta 1
iteracja 17 reszta 3
iteracja 18 reszta 9
iteracja 19 reszta 27
iteracja 20 reszta 81
iteracja 21 reszta 38
iteracja 22 reszta 114
iteracja 23 reszta 137
iteracja 24 reszta 1
iteracja 25 reszta 3
iteracja 26 reszta 9
iteracja 27 reszta 27
iteracja 28 reszta 81
iteracja 29 reszta 38

```

(Rys. 53) Poszukiwanie okresowości dla $a=3$

Tutaj okres wynosi $p=8$, a stąd wynikają $k_1=82$, $k_2=80$

Wyliczając ponownie NWD [28], otrzymujemy $a=41$ i $b=5$, które to liczby są poszukiwanymi czynnikami liczby N , gdyż $ab=N$, co łatwo sprawdzić.

Skoro nasz algorytm działa, to pozostaje do rozwiązyania problem jak szybko zbadać funkcję $f(x)$, aby poznać jej okres. Okazuje się że komputer kwantowy zrobi to szybko przy użyciu kwantowego mnożenia oraz QFT, traktując wartość funkcji jak sygnał wejściowy i wyliczając widmo tego sygnału w dziedzinie częstotliwości. Algorytm Shora jest dość zawiły, jednak ogólne jego działanie polega na obliczaniu wyniku według wzoru 50, dla superpozycji wszystkich x . Następnie zawartość rejestrów jest przekształcana w taki sposób aby ujawnić jak często wyniki się powtarzają. Ze względu na specyfikę kwantowej wersji tych obliczeń, przy dostatecznie dużych liczbach ujawnia się przewaga obliczeń kwantowych.

Ciekawymi algorytmami wartymi wspomnienia są obliczenia związane z uczeniem maszynowym. Algorytm HHL będący ich ważnym składnikiem, służy do rozwiązywania układów równań w postaci $Ax=b$, poprzez znalezienie odwrotności macierzy A . HHL zwraca wektor x w formie rejestru kwantowego. Kluczowym krokiem do łatwego odwrócenia macierzy A , jest jej dekompozycja do wartości i wektorów własnych oraz zmiana oryginalnej

bazy na bazę złożoną z wektorów własnych. To jest możliwe dzięki poznanemu wcześniej prymitywowi szacowania fazy. Taka macierz, staje się macierzą diagonalną, gdzie wystarczy znaleźć odwrotności liczb na głównej przekątnej. Odbywa się to za pomocą operacji ROTY na splątanym kubicie skreżowym. Algorytm na koniec wzmacnia amplitudę rozwiązania, oraz cofa obliczenia, rozplątując rejestr zawierający wynik z pozostałymi kubitami.

Innym algorymem jest QPCA, czyli kwantowe poszukiwanie głównych składowych. W klasycznej implementacji wykorzystuje obliczanie macierzy kowariancji między znormalizowanymi cechami, a następnie rozkładanie otrzymanej macierzy w celu otrzymania wektorów i wartości własnych. Te wektory własne określają kierunki składowych PCA, a wartości własne mówią o wielkości wariancji wzdłuż wyznaczonych kierunków, umożliwiając odrzucenie kierunków o niewielkiej wariancji i pozwalając na redukcję wymiarowości danych bez utraty wiedzy w nich ukrytej.

Kwantowa wersja PCA polega na zdefiniowaniu rejestru QPU opisującego wyznaczoną macierz kowariancji, przekształceniu jego stanu w operator gęstości, a następnie użyciu prymitywu szacowania fazy w celu wyznaczenia wektorów i wartości własnych pozwalających na zdefiniowanie wielowymiarowych punktów danych w nowej bazie. Ponieważ szacowanie fazy wymaga wcześniejszej znajomości jednego ze stanów własnych, a jest to właśnie dana której szukamy, dlatego macierz kowariancji należy przedstawić jako operator gęstości. Wynikiem QPCA jest jeden z wektorów własnych macierzy kowariancji oraz odpowiadająca mu wartość własna. Innymi słowy, QPCA zwraca losowo wybrany główny kierunek składowy oraz przypisaną mu wariancję, przy czym z największym prawdopodobieństwem są zwracane najbardziej znaczące kierunki. Zatem uruchamiając QPCA wielokrotnie, możemy z dużą pewnością poznać główne składowe badanego zbioru danych uczących. QPCA działa dobrze na danych w których wynik klasyfikacji zależy głównie od niewielkiego ułamka wszystkich cech, a właśnie w takich zbiorach PCA przynosi największą redukcję wymiarowości. Co ważne QPCA dostarcza wynik w kwantowej postaci, więc dobrze nadaje się on do dalszego kwantowego przetwarzania, a nie bezpośredniego odczytu.

Ostatnim algorymem jest QSVM. Algorytm ten pozwala na wytrenowanie na danych w celu otrzymania pewnych stanów kodujących parametry hiperpłaszczyzn klasyfikacji. Następnie za pomocą testów SWAP można porównać podobieństwo klasyfikowanego punktu do otrzymanego stanu i podjąć decyzję o przynależności punktu do klasy. Algorytm polega na przekształceniu problemu trenowania do obliczeń macierzowych, w których QPU jest dobre i szybkie. Dane treningowe dostarczane są jako ich superpozycja.

Więcej algorytmów kwantowych można znaleźć w zestawieniu [31].

Praktyczne przykłady korzystania z technologii kwantowych

Jak wykazano wcześniej, obliczenia kwantowe i powiązane technologie sprzętowe obecnie wciąż rozwijają się. Co roku powstają prototypy o parametrach przewyższających osiągi urządzeń poprzednich generacji. W środowisku naukowym, na poziomie ogólnosławowym nie brak wiedzy, środków czy pomysłów na dalsze działania w tym kierunku. Ten rozdział jest jednak poświęcony znalezieniu odpowiedzi na pytanie, jak uczeń czy student może uzyskać dostęp do tego fascynującego świata, oraz czy i w jaki sposób może uzyskać dostęp do sprzętu i symulatorów kwantowych.

Pierwszym wartościowym zasobem, który warto wymienić jest wiedza naukowa jak działają technologie kwantowe. Mimo już 30 letniej historii rozwoju tej technologii, obliczenia kwantowe to wciąż temat nowy, dość mało znany. Dlatego warto choćby побieżnie przestudiować najważniejsze zasady, pojęcia, prymitywy czy bramki. Choć matematyczny opis może wydawać się ciężki i odpychający, to podstawy działania kubitów nie są aż tak złożone jak mogłyby się wydawać. Ponadto do samego wypróbowania gotowych algorytmów lub eksperymentowania z modyfikowaniem ich, nie jest konieczne całkowite zrozumienie wszystkich szczegółów. Z drugiej strony, posiadanie dużej wiedzy początkowej, pozwala lepiej orientować się w technologii, zadawać lepsze pytania, szybciej znajdować odpowiedzi i rozumieć przynajmniej ogólnie to, co widzimy w symulatorze. Za skupieniem się na teorii przemawia też słaba dostępność realnego sprzętu, który w większości jest w fazie prototypów niekoniecznie nadających się do zastosowań w życiu codziennym. Wreszcie dostęp do wiedzy naukowej zaspokaja ciekawość i pozwala lepiej poznać ekscytujący świat technologii kwantowych, dając przyjemność w poznawaniu nowych obszarów, a gdy technologia rozwinię się wystarczająco, pozwoli na płynne przejście od teorii do zastosowań praktycznych.

Wiedzę naukową znajdziemy w bazach takich jak IEEE Access albo podobnych źródłach. Będąc uczniem lub studentem powinniśmy w ramach organizacji taki dostęp posiadać. Materiałów na temat obliczeń kwantowych jest tak wiele, że nie jest możliwe przejrzenie wszystkich. Należy precyźniej określić obszar zainteresowań. Czy będą to konkretne algorytmy, czy raczej interesują nas nowości z zakresu rozwoju sprzętu, korzystając z filtrów można zawęzić wyniki do ilości którą jesteśmy w stanie przetworzyć. Warto korzystać z publikacji wydanych w ostatnich 1-2 latach, gdyż informacja szybko się staje nieaktualna. Ponadto przeszukiwanie bazy pozwala sortować materiały według ilości cytowań i innych metryk jakości materiałów. Początek tej pracy polegający na streszczeniu aktualnego stanu technologii kwantowej został opracowany na podstawie znalezionych publikacji [1-4,6-19,21-26,29].

Kolejnym źródłem są publikacje książkowe [27], w razie braku wersji papierowej, także na platformach elektronicznych. Pomagają one w przystępny sposób uporządkować różne zagadnienia. Przydaje się to zwłaszcza na początku, gdy wiele terminów czy koncepcji jest nowych i niezrozumiałych.

Innym zasobem edukacyjnym jest symulator kwantowy, bez którego trudno wyobrazić sobie praktyczną naukę czy wypróbowanie zdobytej wiedzy w praktyce. Jeden z takich symulatorów [20], prosty obsłuze i niewymagający żadnego środowiska poza przeglądarką internetową pozwala oswoić się z suchą teorią w bardziej praktyczny sposób. Był on już wykorzystywany w opisanych wcześniej przykładach, między innymi w wykresach kołowych stanu rejestrów. Posiada on zestaw programów które można uruchomić, modyfikować czy tworzyć własne w języku Javascript/QCEngine wzorując się na dostępnych przykładach. Przy korzystaniu z symulatorów, należy jedynie pamiętać o ograniczeniach możliwości symulacji. Dla znacznej ilości kubitów symulacja będzie działać wolno, jednak proste przykłady można prześledzić na małych rejestrach złożonych z kilku kubitów.

Run Program Ex 4-1: Basic telep... ▾ QC Engine ▾ 🔍 🔍

```

1 // Programming Quantum Computers
2 // by Eric Johnston, Nic Harrigan and Mercedes Gimeno-Segovia
3 // O'Reilly Media
4
5 // To run this online, go to http://oreilly-qc.github.io?p=4-1
6
7 // This sample demonstrates basic teleportation.
8
9 qc.reset(3);
10 var alice = qint.new(1, 'alice');
11 var ep   = qint.new(1, 'ep');
12 var bob  = qint.new(1, 'bob');
13 var al = 0;
14 var a2 = 0;
15
16 // This will work with entangle() and alice_prep() in either order.
17 // Try swapping them to verify this.
18 entangle();
19 alice_prep();
20 alice_send();
21 bob_receive();
22 bob_verify();
23
24
25 function entangle()

```

Source code on Github

Program circuit 🔍 🔍

Circle notation 🔍 🔍

(Rys. 54) Okno symulatora kwantowego w przeglądarce internetowej Firefox [20]

Gdyby to było za mało, istnieje Qiskit [30], czyli biblioteka w Pythonie służąca nie tylko do symulacji, ale także do tworzenia oprogramowania kwantowego działającego na realnym sprzęcie po wybraniu odpowiedniego backendu. Qiskit został opracowany przez IBM, a sama firma oferuje też dostęp do swoich komputerów kwantowych w chmurze, pozwalając używać Qiskit do rzeczywistych obliczeń. Dostępne są QPU ze 127 kubitami.

Qiskit można uruchomić na Google Collab [32], platformie do eksperymentów naukowych i badań, także działającej w przeglądarce, wykorzystującej Pythona. Różnica w stosunku do poprzedniego symulatora [20] polega na używaniu chmury Google do obliczeń, zamiast lokalnego silnika Javascript. Pozwala to na dostęp do większej mocy obliczeniowej, 12GB RAM i około 100GB przestrzeni dyskowej w darmowym planie. Dane te są jednak ulotne gdyż backend jest usuwany po wykonaniu obliczeń i dłuższym braku aktywności. Jednak wyniki zostaną zapisane jako notatnik typu Jupyter Notebook, popularny wśród naukowców.

```

Untitled0.ipynb ☆
Plik Edytuj Widok Wstaw Środowisko wykonawcze Narzędzia Pomoc Wszystkie zmiany zostały zapisane
+ Kod + Tekst
19s
!pip install qiskit
!pip install qiskit_aer

# Importujemy potrzebne moduły z Qiskit
from qiskit import QuantumCircuit, transpile, assemble
from qiskit_aer import Aer
from qiskit.visualization import plot_histogram

# Tworzymy obwód kwantowy z jednym qubitem
qc = QuantumCircuit(1, 1)

# Dodajemy bramkę Hadamarda (H) na qubicie
qc.h(0)

# Wykonujemy pomiar na qubicie
qc.measure(0, 0)

# Wyświetlamy obwód
qc.draw()

# Transpilujemy obwód do backendu Aer
qc = transpile(qc, Aer.get_backend('qasm_simulator'))

# Uruchamiamy symulację
job = assemble(qc)

# Pobieramy wyniki
simulator = Aer.get_backend('qasm_simulator')
result = simulator.run(qc).result()

# Wyświetlamy wyniki

```

0 s ukończono o 10:40

(Rys. 55) Google Collab z programem w Qiskit do generowania liczb losowych [32]



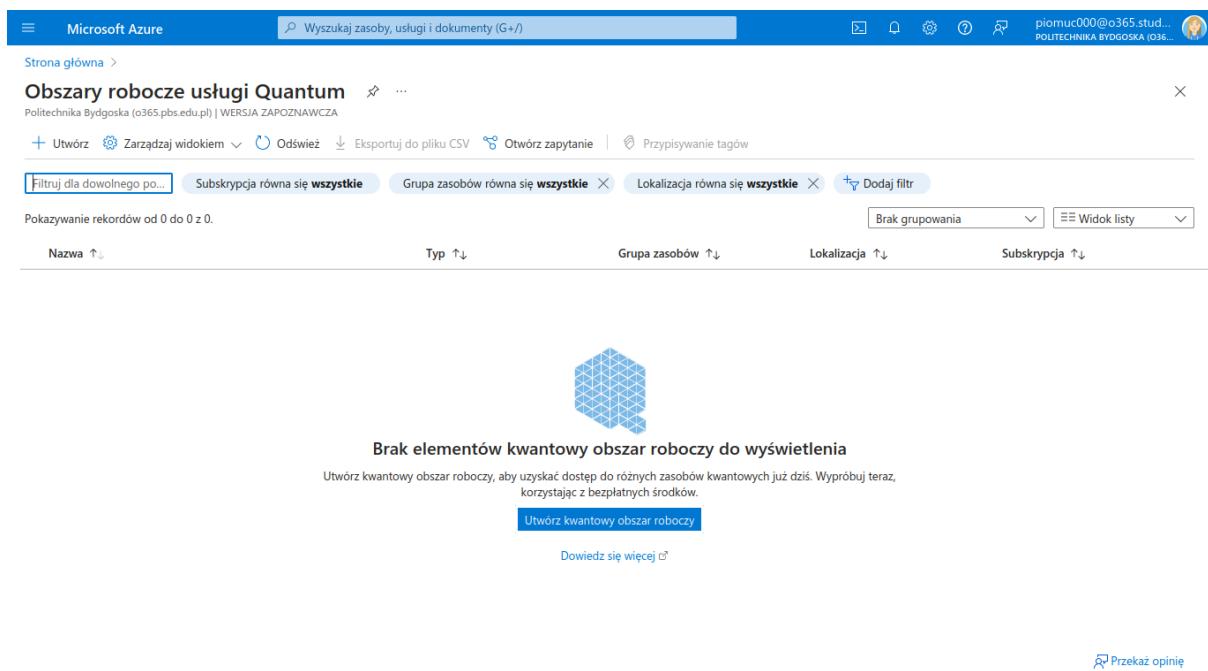
(Rys. 56) Rozkład zwracanego losowego bitu [32]

Ponadto symulatory kwantowe są też często częścią środowisk badawczych w chmurze. Służą one do przetestowania kodu i koncepcji, zanim program zostanie wysłany do sprzętowego QPU. Większość, jeśli nie wszyscy dostawcy wymienieni w [4] i (Rys. 8) posiadają takie symulatory, co jest wygodne, gdyż czas oczekiwania na wykonanie kodu na QPU jest dość długi, rzędu godzin, a ponadto QPU czasem doświadczają przerw konserwacyjno-kalibracyjno-technologicznych. Dostęp do symulatora jest relatywnie dużo łatwiejszy, gdyż symulacja opiera się w 100% o klasyczne obliczenia.

Podsumowując, o ile symulacje nie stanowią problemu, to dostęp do fizycznego sprzętu kwantowego już tak. Przede wszystkim dlatego, że jest to technologia bardzo droga. Ponadto sprzęt jest stale rozwijany, zatem technologia dostępna publicznie jest spóźniona w stosunku do osiągnięć naukowych o 1-2 lata. Kolejny problem to zaszumienie obliczeń, objawiające się widocznymi różnicami między wynikami symulacji a rezultatami obliczeń na fizycznym QPU. Z drugiej strony, dostawcy na brak użytkowników nie mogą narzekać, a często dają możliwość darmowego dostępu do QPU pozwalając użytkownikowi na wykonanie pewnej, zwykle dość niewielkiej ilości obliczeń. Tego typu dostęp jest więc atrakcyjny dla uczniów, studentów i jednostek edukacyjnych.

Aby zweryfikować jakość i dostępność publicznych GPU, przyjrzyjmy się dostępnym możliwościom zaczynając od usług zapewnianych przez jednostkę edukacyjną. Od pandemii COVID-19 wzrosło zainteresowanie platformami chmurowymi w edukacji, co oznacza że większość szkół i uczelni posiada już jakąś usługę tego typu. Politechnika Bydgoska korzysta z chmury Microsoft. Zasadniczo PBŚ nie oferuje dostępu do QPU, a głównym powodem posiadania subskrypcji jest dostęp do oprogramowania Office i Teams dla pracowników i studentów. Niemniej jednak PBŚ umożliwia zaaplikowanie o zarządzane konto Microsoft, umożliwiające użytkownikowi identyfikowanie się jako osoba ucząca się, co też należy uczynić.

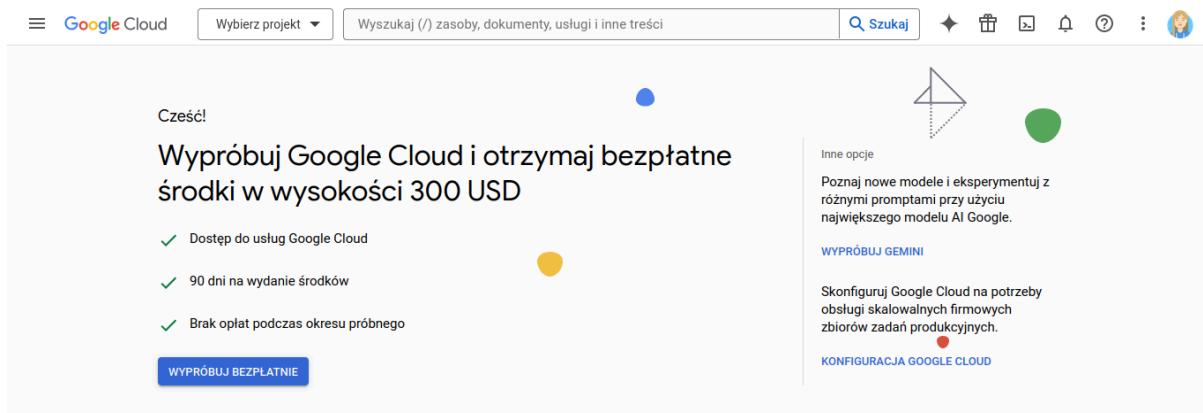
Kolejnym krokiem jest uzyskanie dostępu do platformy Azure. Tutaj możliwość dostępu jest mniejsza, gdyż zasadniczo usługi chmurowe są płatne, jednak Microsoft daje organizacjom edukacyjnym ograniczone czasowo i kwotowo jednorazowe pule darmowych środków per użytkownika na korzystanie ze swojej infrastruktury. Autor tej pracy korzystał na Azure m.in. z maszyn wirtualnych, baz danych czy środowiska do uczenia maszynowego, zanim przeszedł do obliczeń kwantowych. Same obliczenia kwantowe mogą być realizowane w ramach kwantowego obszaru roboczego, czyli jednej z usług na platformie Azure, o której należy zaaplikować osobno. (Rys. 57) Mając już edukacyjne konto Microsoft, oraz dostęp do Azure, tworzenie kwantowego obszaru roboczego jest łatwe. Microsoft daje uczniom i studentom osobną pulę bezpłatnych środków na obliczenia kwantowe, jednak korzystanie z obszaru kwantowego wymaga, aby również subskrypcja na cały Azure była aktywna. Tego warunku autorowi niniejszej pracy nie udało się spełnić, gdyż całe środki z Azure zostały wykorzystane na "zwykłe usługi chmurowe" i subskrypcja uległa dezaktywacji, blokując dostęp do chmury Azure, w tym do kwantowego obszaru roboczego, zanim autor zdążył wykonać jakiekolwiek obliczenia. W późniejszym czasie udało się uzyskać nową subskrypcję Azure, jednak była to wersja skromniejsza, nie obejmująca kwantowych obszarów roboczych.



(Rys. 57) Uzyskanie własnego obszaru roboczego Quantum [33]

Podsumowując, Microsoft oferuje obliczenia kwantowe w chmurze, co pozwala na łatwą integrację obliczeń kwantowych jako komponentu w większej aplikacji, albo samodzielne korzystanie z obliczeń w obszarze roboczym. Ponadto zapewnia łatwy dostęp do różnych dostawców technologii kwantowej, a liczba dostępnych kwantowych backendów dalej powiększa się. Na uwagę zasługuje też spory wybór obsługiwanych języków programowania. Z drugiej strony, przejście przez wszystkie subskrypcje i plany, może być trudne, szczególnie bez oficjalnego wsparcia administratora IT organizacji edukacyjnej, czy jawnego uwzględnienia Azure i usług kwantowych w ramach kont uczniów i studentów. Ponadto dostęp do usług Microsoft jest obecnie w fazie zapoznawczej, mocno ograniczonej czasowo i ilościowo.

Innym popularnym dostawcą chmury jest Google. Na tej chmurze także można uruchomić obliczenia kwantowe. Ze względu na brak edukacyjnego konta Google w PBŚ, autor nie miał możliwości zweryfikować dostępności tych usług, ale spróbował użyć samodzielnie utworzonego niezarządzanego konta. Google pozwala na bezpłatny 90-dniowy okres próbny, dając też pewną ilość darmowych środków do wykorzystania na dowolne usługi chmurowe (a zatem także obliczenia kwantowe, o ile nie są rozliczane w sposób odmienny). Niestety do weryfikacji konta potrzebny jest numer karty kredytowej, a tą w Polsce można posiadać z reguły od 18 roku życia, choć są pewne możliwości dla młodszych osób. Konieczność posiadania konta bankowego i karty, to poważne utrudnienie dla części osób, chcących poznawać technologie kwantowe, tym bardziej że karta nawet nie służy do płatności, a jedynie potwierdzenia, że konto nie jest zakładane przez robota. Na szczęście konta zarządzane i udostępniane przez szkołę czy uczelnię nie wymagają weryfikacji przez użytkownika końcowego. Google pozwala na korzystanie z obliczeń kwantowych u mniejszej ilości dostawców niż Microsoft. Ponadto nie wszystkie zasoby kwantowe są dostępne publicznie. Do programowania wykorzystuje się narzędzie Cirq. Ponieważ obliczenia kwantowe są częścią platformy chmurowej, można przypuszczać, że podobnie jak w Azure, można wykorzystać QC jako część większej aplikacji.



Popularne materiały dla początkujących

Filtruj według

Internet, rozwiązania mobilne, gry, pamięć masowa | Kontenery, maszyny wirtualne, hybrydowe/multi, przenoszenie zadań | Dane, AI/ML, SAP
Mapy, interfejsy API | Ogólne

Gotowe szablony rozwiązań

- Wdroż trzydziemiarową aplikację internetową
- Wdrażaj zarządzane maszyny wirtualne z równoważeniem...
- Utwórz hurtownię danych za pomocą BigQuery

(Rys. 58) Konsola Google Cloud do zarządzania zasobami w chmurze [34]

IBM udostępnia w momencie pisania tej pracy 4 QPU oraz 5 symulatorów różnego typu. Pierwszym krokiem jest założenie konta. Mając konto możemy pobrać klucz do API umożliwiający tworzenie tak zwanych jobów, czyli zadań obliczeniowych do symulatora lub QPU.

IBM Quantum Platform							
Dashboard Compute resources Jobs							
Instance resources All systems All simulators							
You have access to the following resources with instance ibm-q/open/main.							
<input type="text"/> Search by system or simulator name						Your systems & simulators (9)	
Name	Qubits	EPLG	CLOPS	Status	Total pending jobs	Processor type	Features
ibm_sherbrooke	127	1.7%	5K	● Online	17	Eagle r3	
ibm_brisbane	127	1.9%	5K	● Online	1028	Eagle r3	
ibm_osaka	127	2.8%	5K	● Online	1	Eagle r3	
ibm_kyoto	127	3.6%	5K	● Online	20	Eagle r3	
simulator_stabilizer	5000	-	-	● Online	3	Clifford simulator	
simulator_mps	100	-	-	● Online	3	Matrix Product State	
simulator_extended_stabilizer	63	-	-	● Online	3	Extended Clifford (e.g. Clifford+T)	
ibmq_qasm_simulator	32	-	-	● Online	3	General, context-aware	
simulator_statevector	32	-	-	● Online	3	Schrödinger wavefunction	

(Rys. 59) Dostępne zasoby IBM [35]

Spróbujmy zatem utworzyć prosty obwód i porównać wyniki symulacji z wynikami na sprzęcie. Naszym obwodem będzie 4-kubitowy rejestr, który zainicjalizujemy w superpozycji

$|0\rangle + |1\rangle + |4\rangle + |7\rangle$, a następnie dokonamy kwantowej inkrementacji o 3 i zbadamy rozkład prawdopodobieństwa odczytania poszczególnych stanów. Przygotujmy zatem eksperyment, wyjaśniając poszczególne kroki.

```
# instalujemy wymagane zależności
!pip3 install qiskit qiskit-aer qiskit-ibm-runtime pylatexenc > /dev/null

# importujemy pakiety
from qiskit import QuantumCircuit
from qiskit_ibm_runtime import QiskitRuntimeService
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
from qiskit.transpiler.preset_passmanagers import generate_preset_pass_manager

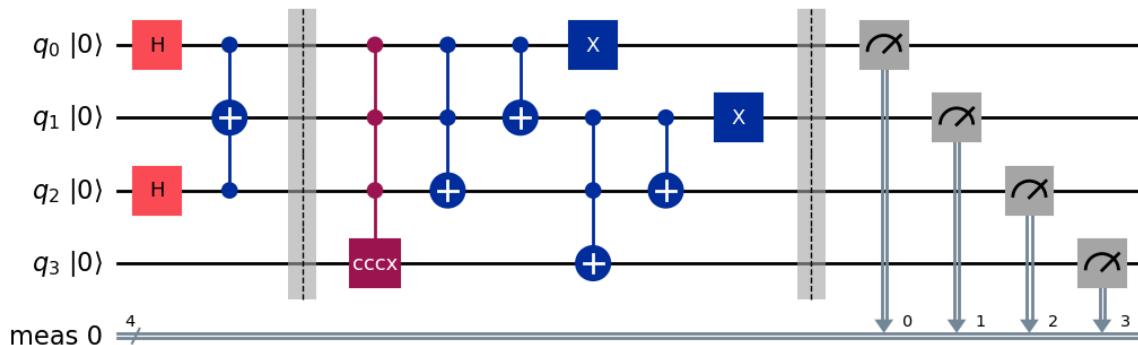
# tworzenie brakującej bramki CCCNOT
cccxnot = QuantumCircuit(1, name="cccx")
cccxnot.x(0)
cccx_gate = cccxnot.to_gate().control(3)

# tworzymy obwód eksperymentu
obwod = QuantumCircuit(4)
obwod.h([0,2])
obwod.ccx(0,2,1)
obwod.barrier()
obwod.append(cccx_gate, [0,1,2,3], [])
obwod.ccx(0,1,2)
obwod.cx(0,1)
obwod.x(0)
obwod.ccx(1,2,3)
obwod.cx(1,2)
obwod.x(1)
obwod.measure_all()

# rysowanie obwodu
obwod.draw(output="mpl", initial_state=True, idle_wires=False)
```

(Rys. 60) Część pierwsza kodu eksperymentu [32]

Pierwszą interesującą częścią programu jest tworzenie prostego obwodu złożonego z pojedynczej bramki NOT, którą natychmiast konwertujemy do wersji warunkowej, gdyż jest ona nam potrzebna w obwodzie eksperymentu. Następnie tworzymy obwód właściwy złożony z 4 kubitów. Na 2 kubitach działa bramka Hadamarda wprowadzająca superpozycję wartości rejestru $|0\rangle + |1\rangle + |4\rangle + |5\rangle$ gdzie każda wartość ma tę samą amplitudę. CCX to bramka Toffoliego, czyli warunkowe NOT o 2 wejściach kontrolnych. Bramka działa na drugim kubicie negując jego stan zawsze gdy pierwszy i trzeci kubit mają stan "1". Stanie się tak gdy rejestr będzie w stanie 5, 7, 13 lub 15. Jak widać 5 występuje w superpozycji, zatem zanegowanie drugiego kubitu zmieni $|5\rangle$ w $|7\rangle$. Ostatecznie rejestr znajdzie się w stanie $|0\rangle + |1\rangle + |4\rangle + |7\rangle$. Wybrałem ten stan, ponieważ pozwala on łatwo zademonstrować inkrementację. Inkrementacja jest realizowana przez kolejne bramki, w tym wytworzoną wcześniej 3-wejściową bramkę Toffoliego cccx_gate. Cały obwód przedstawia rysunek.



(Rys. 61) Obwód eksperymentu [30, 32]

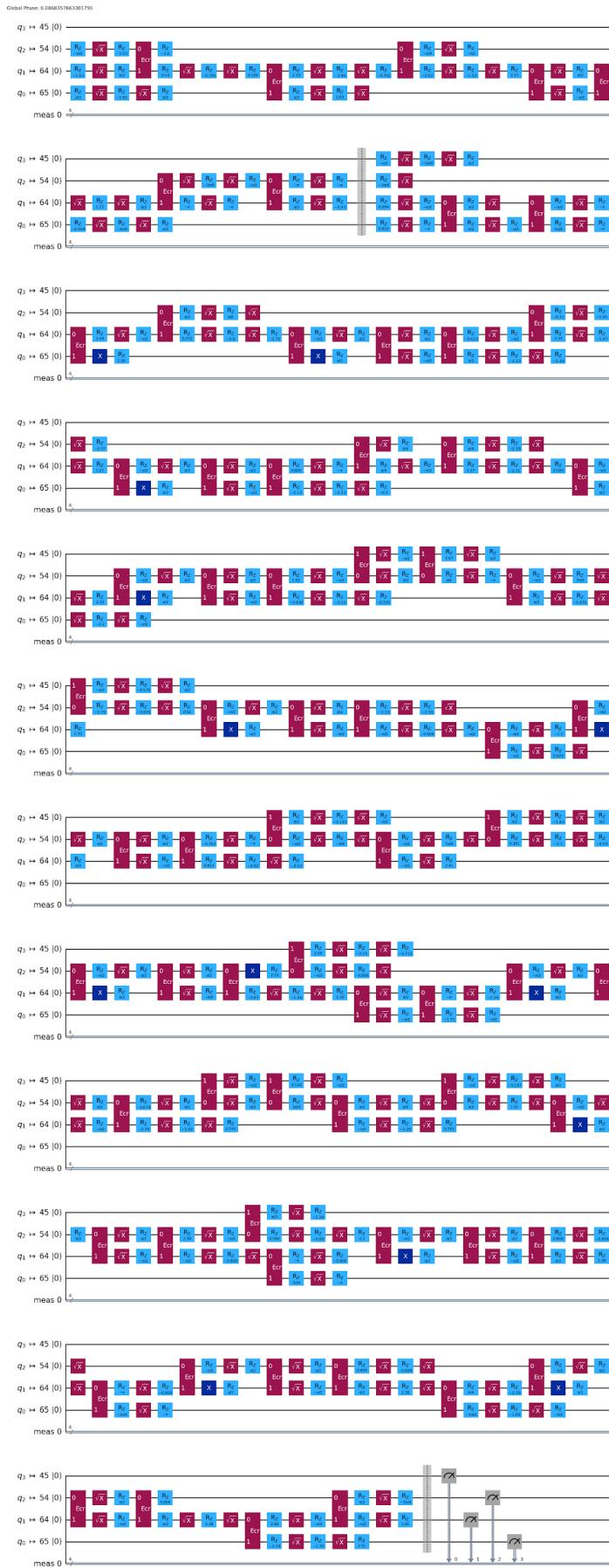
Dalsza część kodu realizuje tworzenie obiektu do komunikacji z centrum obliczeń kwantowych firmy IBM, oraz bardzo ważny krok - zamianę instrukcji wysokopoziomowych na bramki dostępne w danym QPU. QPU dostępne w centrum IBM posiadają instrukcje ECR, ID, RZ, SX, X. Nie ma natomiast bramki Hadamarda czy CCX. Dlatego ten krok jest konieczny aby program w ogóle mógł być uruchomiony.

```
# Tworzymy obiekty do komunikacji z QPU
usluga_qc = QiskitRuntimeService(channel="ibm_quantum", instance="ibm-q/open/main")
wybrany_qpu = usluga_qc.backend(name="ibm_kyoto")

# transpilacja obwodu do postaci akceptowanej przez wybrane QPU
ppm = generate_preset_pass_manager(backend=wybrany_qpu, optimization_level=3)
obwod = ppm.run(obwod)

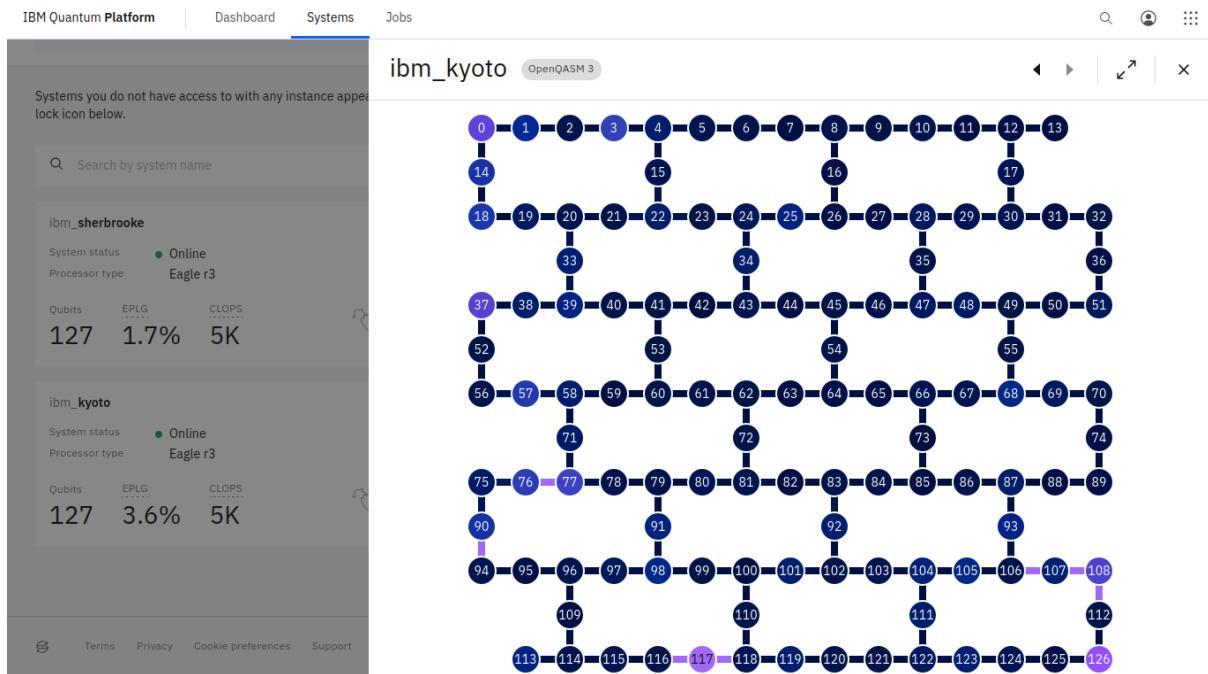
# rysowanie obwodu dla docelowej architektury QPU
obwod.draw(output="mpl", initial_state=True, idle_wires=False)
```

(Rys. 62) Dalsza część kodu eksperymentu [32]



(Rys. 63) Przekształcony obwód dla QPU [30]

Niewątpliwie mniejsza ilość dostępnych bramek zwiększa ilość instrukcji niskopoziomowych. Ponadto transpilator przypisał zdefiniowane kubity do konkretnych miejsc w procesorze. (Rys. 64) przedstawia mapę procesora, z uwzględnieniem danych kalibracji, gdzie jaśniejsze elementy oznaczają kubity z większą podatnością na błędy.



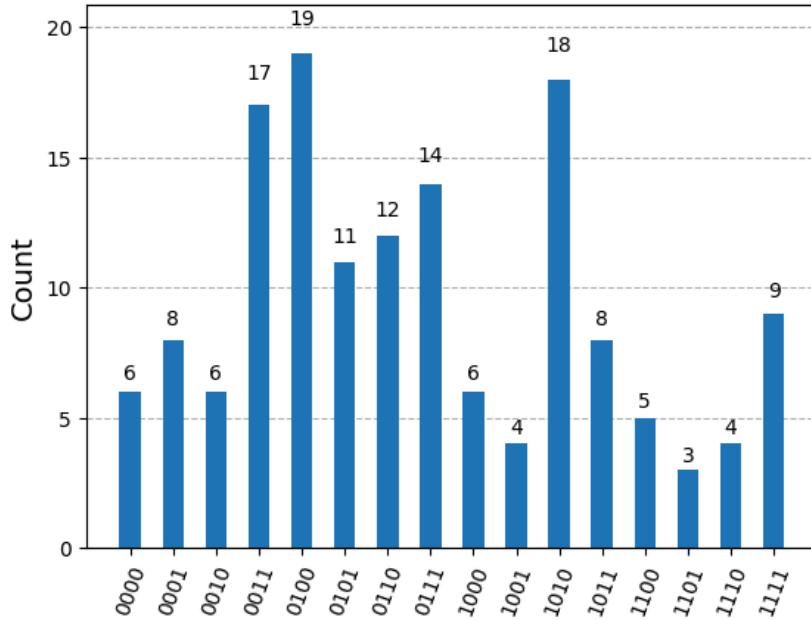
(Rys. 64) Mapa użytego QPU [35]

```
# Tworzenie zadania
job = wybrany_qpu.run([obwod], shots=150)

# pobieranie wyników
wyniki = job.result()
plot_histogram(wyniki.get_counts())
```

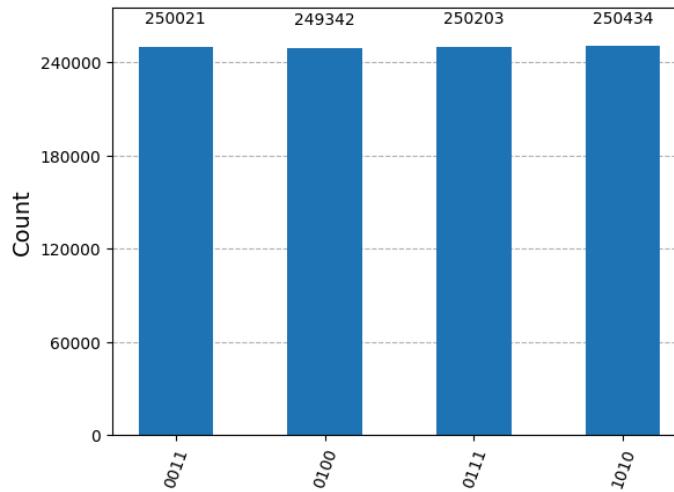
(Rys. 65) Kod wywołujący program na QPU i pobierający wyniki gdy będą dostępne [32]

Histogram poniżej przedstawia rozkład odczytanych stanów 4-kubitowego rejestru, na którym przeprowadzaliśmy inkrementację.



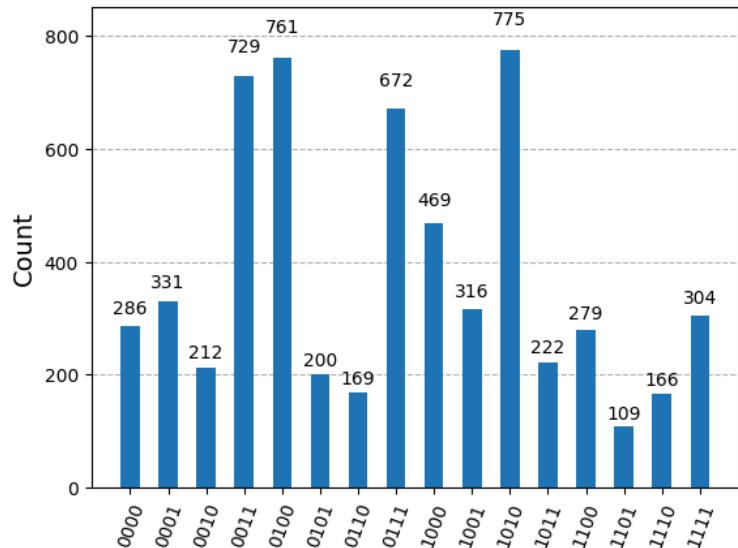
(Rys. 66) Otrzymane wyniki z QPU ibm_kyoto [35]

Na (Rys. 66) widać, że stany $|3\rangle$, $|4\rangle$, $|7\rangle$ i $|10\rangle$ występuły najczęściej. Oznacza to, że obwód działa poprawnie, chociaż widać duże zaszumienie obliczeń. Zobaczmy jak wyglądałby rozkład na idealnym symulatorze. Aby to zrobić, wystarczy zastąpić obiekt interakcji z centrum danych, obiektem AerSimulator, a także zwiększyć ilość shotów, czyli iteracji uruchomienia obwodu kwantowego.



(Rys. 67) Wyniki symulacji w Qiskit AerSimulator [30]

Teraz nie ma wątpliwości, jaki wynik byśmy otrzymali, gdyby nasze QPU było lepszej jakości. Możemy jednak uruchomić obwód jeszcze raz na innym QPU, o niższym współczynniku błędu, oraz zwiększyć ilość shotów, aby próba była większa. Zobaczmy jak wpłynie to na wyniki.



(Rys. 68) Otrzymane wyniki z QPU ibm_sherbrooke [35]

Rzeczywiście, stany $|3\rangle$, $|4\rangle$, $|7\rangle$ i $|10\rangle$ są teraz wyraźniej widoczne.

Wnioski

Podsumowując, badając praktyczne możliwości realizacji obliczeń kwantowych, udało się skorzystać z darmowego planu IBM Quantum i wykonać prosty obwód wykonujący inkrementację. Porównując symulatory z rzeczywistymi QPU, możemy wymienić wady i zalety obu możliwości. Zaletą symulatora jest jego dostępność oraz możliwość rozpoczęcia obliczeń natychmiast, bez długiego oczekiwania w kolejce. Wadą natomiast jest brak rzeczywistej kwantowości, co oznacza, że nie uzyskuje się kwantowego przyspieszenia. Ponadto symulator ma ograniczoną moc obliczeniową i nie może zasymulować obwodu z dużą ilością kubitów. Symulatory sprawdzą się więc do testowania i debugowania prostych algorytmów. Zaletą użycia QPU jest to, że obliczenia odbywają się rzeczywiście kwantowo i przedżej czy później każdy algorytm będzie musiał być uruchomiony na prawdziwym sprzęcie. Do wad należy trudny dostęp, długi czas oczekiwania na wyniki i zaszumienie obliczeń wynikające z obecnych niedoskonałości technologii. W pracy nad obwodami kwantowymi, konieczne jest użycie obu rozwiązań ponieważ wzajemnie uzupełniają się.

Innym wnioskiem dotyczącym zastosowania obliczeń kwantowych w placówkach dydaktycznych i naukowych jest stwierdzony brak lub niewielkie wsparcie macierzystych organizacji w dostępie do QPU. Wszystkie wypróbowane rozwiązania były wersjami zapoznawczymi lub ograniczonymi czasowo albo ilościowo. Gdyby obliczenia kwantowe miały stać się przedmiotem badań i nauczania, to należałoby rozważyć zakup odpowiednich subskrypcji w celu zagwarantowania stałego i wystarczającego dostępu do QPU. W trakcie badań dostępności QPU okazało się, że najłatwiejszy dostęp oferowała firma niezwiązana z dostarczaniem platform edukacyjnych dla szkół i uczelni.

Ciekawym spostrzeżeniem jest również doświadczenie tego jak szybko rozwija się technologia kwantowa. W trakcie pisania tego rozdziału, pojawiła się nowa wersja biblioteki Qiskit [30], natomiast w przeciągu kilku dni od założenia konta w IBM Quantum [35], do uruchomienia pierwszego zadania obliczeniowego, symulatory (Rys. 59) zostały usunięte z instancji "ibm-q/open/main" i pojawiła się informacja aby korzystać z lokalnych symulatorów. Ponadto obserwując liczbę kubitów w dostępnych QPU, można zauważyć że ich liczba podwaja się każdego roku. Można oszacować, że próg 10 000 kubitów, oznaczający erę post-kwantową, zostanie osiągnięty na początku lat 30-tych obecnego stulecia.

Obecnie, zastosowania komputerów kwantowych to przede wszystkim badania naukowe i edukacja. Istnieją już pierwsze aplikacje związane z technologiami kwantowymi, takie jak sieci dystrybucji kluczy QKD. W najbliższym czasie należy oczekwać kolejnych rozwiązań osiągających wystarczającą dojrzałość technologiczną do zastosowania w życiu codziennym. W chwili obecnej jest najlepszy czas na to, aby z technologiami kwantowymi się zaznajamiać, poznawać ich możliwości i ograniczenia, a także gromadzić wiedzę i umiejętności do wykorzystania w niedalekiej przyszłości.

Zatem na pytanie badawcze, postawione w tej pracy, czy obliczenia kwantowe mają zastosowanie w placówkach dydaktycznych lub naukowych, można odpowiedzieć jednoznacznie twierdząco.

Pomysły na przyszłe zastosowania

Próbując znaleźć zastosowania obliczeń i technologii kwantowych, należałoby najpierw zaznajomić się z już dokonanymi osiągnięciami nauki, co zostało wykonane w początkowej części tej pracy. Choć temat nie został wyczerpany, a rozwój dalej trwa, to z zebranych informacji wyłania się obiecujący obraz obliczeń kwantowych mogących w przyszłości przesunąć granice tego co uważa się za wykonalne. Szybkie i pojemne bazy danych wykorzystujące algorytm Grovera, bezpieczna komunikacja już od "zerowej warstwy ISO/OSI", technologie chroniące prywatność, modele AI zawierające wielką ilość zmiennych zakodowanych w rejestrach kubitowych, czy inne pomysły, które dopiero się pojawią gdy QPU staną się niezbędnym komponentem infrastruktury informatycznej. To wszystko czeka na odkrycie i wdrożenie. Aby jednak dać kilka przykładów już teraz, można rozważyć następujące zagadnienia.

Obecnie ilość całej wyprodukowanej informacji cyfrowej na świecie to około 2^{79} bitów. Dokładna ilość ani rząd nie są istotne. Chodzi o to, że można w dużym uproszczeniu stwierdzić, iż rejestr N-kubitowy, może przechowywać 2^N bitów informacji. A jeszcze ogólniej, że parametry obliczeniowe rosną wykładniczo ze wzrostem liczby kubitów. Zatem teoretycznie, dostępne już dziś QPU zawierające 127 kubitów, mogą przechowywać całą wiedzę ludzkości. Gdyby móc tworzyć modele AI wytrenowane na tej ilości danych, to jest możliwe, że mogły by znacznie wyprzedzić te używane obecnie. W ogóle sama koncepcja przetwarzania tak wielkiej ilości informacji w jednym miejscu jest już interesująca sama w sobie.

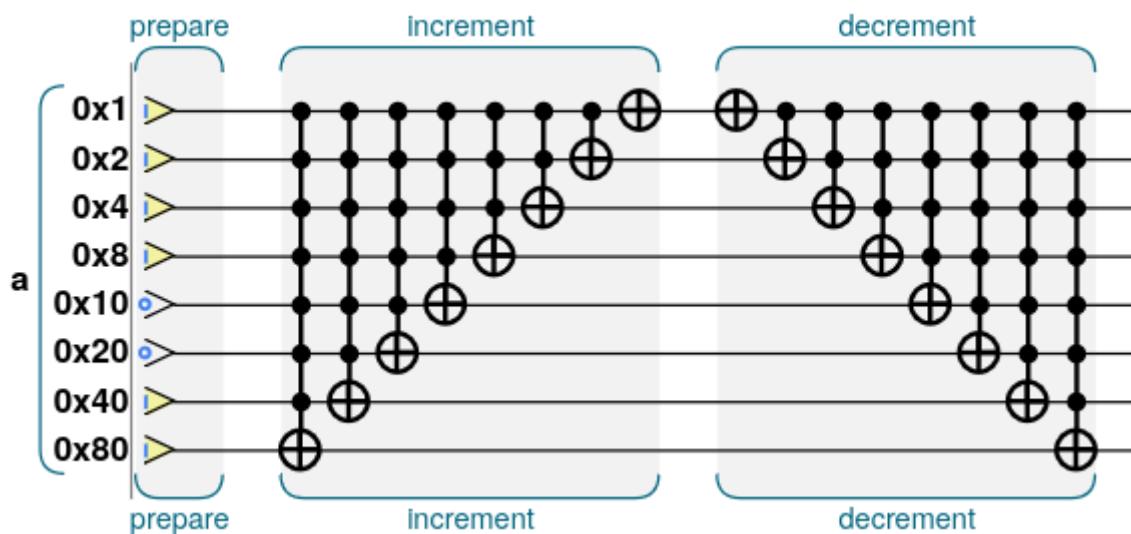
Skoro można przetwarzać bez problemu tak wielkie ilości danych, to różne symulacje, np symulacje pogody, nie muszą ograniczać się w rozdzielczości modelu, a zatem wyniki mogą być dokładniejsze. Istnieją też dziedziny badań które dostarczają ogromnych ilości danych pomiarowych, takie jak na przykład eksperymenty w Europejskim centrum badań jądrowych CERN. W 2012 roku naukowcy radzili sobie z tym problemem, filtrując dane i zapisując tylko interesujące zdarzenia, co oznacza, że w LHC mogły wystąpić procesy fizyczne, które uszły uwadze badaczy tylko dlatego, że nie zostały zarejestrowane z powodu braku przestrzeni w centrum danych. Gdyby skonstruować komputer kwantowy, w którym informacja może być akumulowana przez znacznie dłuższy czas, to można by kierować tam dane pomiarowe, przetwarzać je, czy budować modele uczące się na danych i poszukiwać nowych wzorców.

Skoro jesteśmy przy temacie zaawansowanej aparatury badawczej, to sam komputer kwantowy jest dość precyzyjnym instrumentem pomiarowym, któremu pomysłowi naukowcy mogą znaleźć alternatywne zastosowanie, niekoniecznie związane z informatyką. Wiadomo, że niejeden wynalazek powstał jako produkt uboczny przy okazji badań nad czymś innym. Prace nad technologiami kwantowymi, inspirują też oryginalne rozwiązania problemów niezwiązanych z kwantowością. Przykładem są klasyczne algorytmy inspirowane obliczeniami kwantowymi [2].

W temacie ochrony informacji i poufnej komunikacji, efekty kwantowe mogą pomóc stworzyć algorytm który tak przekształca dane, że do ich odczytania konieczna jest znajomość

pewnych losowo wybranych parametrów. Co więcej, ze względu na charakterystykę informacji kwantowej, siłowe odgadnięcie poprawnych wartości może być niemożliwe, gdyż będzie tylko jedna próba na odczyt informacji, a w momencie odczytu, jej kwantowa postać zostanie zniszczona. Użycie niewłaściwych parametrów nie pozwoli w ogóle poznać treści wiadomości. Zamiast tego otrzyma się bezwartościowe wartości losowe. Byłby to odpowiednik systemu klasycznego, który dopuszcza tylko jedną próbę uwierzytelnienia, a w przypadku niepoprawnej weryfikacji wymazuje przechowywane dane. Takie zabezpieczenie komunikacji "na zerowej warstwie ISO/OSI" byłoby bezpieczniejsze niż implementacja programowa, a nawet sprzętowe moduły kryptograficzne. Z drugiej strony, stworzenie takiego algorytmu wymaga uwagi i staranności, aby rzeczywiście można było matematycznie udowodnić jego bezpieczeństwo. Ponadto jest jeszcze niepewność związana z samą mechaniką kwantową, która wciąż jest badana i odkrywana, a co za tym idzie, nie można w 100% stwierdzić, że coś jest bezpieczne, jeżeli nie znamy wszystkich szczegółów. Można natomiast oczekwać, że dodanie kryptografii kwantowej jako dodatkowej warstwy zabezpieczeń utrudni nieuprawniony dostęp i w taki sposób należy ją stosować.

Przyglądając się obliczeniom kwantowym, można zauważyc, że QPU nie tyle liczy, ale bardziej przekształca dane wejściowe w wyniki, używając pewnych reguł matematycznych albo logicznych, wyrażonych operacjami kwantowymi. Czym jest inkrementacja w binarnym systemie pozycyjnym? Najprościej, polega ona na znalezieniu N najmłodszych bitów które mają wartość "1", oraz negacji N+1 najmłodszych bitów (czyli negujemy wszystkie "1" i dodatkowo o 1 bit więcej). Przyjrzyjmy się teraz obwodowi inkrementacji.



(Rys. 69) Obwód inkrementacji i dekrementacji [20]

W tym obwodzie, warunkowe bramki NOT aktywują się tylko gdy wszystkie ich wejścia są w stanie 1. Ten warunek nie jest spełniony dla pierwszych 3 bramek, ale jest spełniony dla czwartej i każdej kolejnej (ostatnia bramka NOT jest bezwarunkowa). Bit 0x10 stanie się "1". Konsekwentnie, kolejne bramki też zostaną aktywowane, gdyż korzystają jedynie z podzbioru wejść bramki czwartej. Zatem bity 0x8, 0x4 i 0x2 zostaną przestawione na "0". Ostatnia bramka jest bezwarunkowa, to znaczy negacja odbywa się zawsze, bit 0x1 też zostanie przestawiony na "0". Podsumowując, stan 11001111 zmieni się w 11010000.

Nawet jeżeli liczba N najmłodszych bitów które mają wartość "1" wynosi 0, (tak jak dla stanu który otrzymaliśmy), to $N+1=0+1=1$, zatem jeden najmłodszy bit zawsze musi być zanegowany. Bity są negowane kolejno od najstarszego do najmłodszego, gdyż negacja zależy od wartości wszystkich młodszych bitów, dlatego nie mogą one być do tego czasu modyfikowane. Ostatni najmłodszy bit jest negowany bezwarunkowo, zatem może się to odbyć na końcu. Wracając do początkowej definicji inkrementacji, tożsame jest stwierdzenie, że bit jest negowany tylko gdy wszystkie młodsze bity mają wartość "1", a gdy jest to najmłodszy bit - zawsze. Widzimy, że obwód kwantowy dokładnie odpowiada naszej definicji.

To ważne spostrzeżenie może zainspirować do konstruowania obwodów kwantowych jako struktur opisujących równania matematyczne albo reguły logiczne. Taki obwód wyraża zasadę inkrementacji w binarnym systemie pozycyjnym jako taką, zatem tak jak wzór matematyczny, będzie ona działać niezależnie od wartości rejestru wejściowego, a nawet w przypadku gdy ten rejestr będzie w superpozycji kilku różnych wartości. Możemy zatem sobie wyobrazić przyszłe języki programowania dla komputerów kwantowych, w których programista opisuje problem za pomocą wzorów matematycznych, które są kompilowane do macierzy opisujących operacje kwantowe i stanów rejestrów. Te niskopoziomowe struktury zamieniają dane wejściowe w wyniki.

Kwantowe obliczenia są odwracalne. Zatem dla każdego działającego obwodu możemy znaleźć obwód odwrotny, który zamienia wyniki w dane wejściowe i zrobi to w tym samym czasie co obwód pierwotny. W klasycznych obliczeniach, nie zawsze tak jest. Taki odwrócony obwód może przydać się przy poszukiwaniu właściwych parametrów do osiągnięcia określonych rezultatów, np w uczeniu maszynowym.

Kwantowe obliczenia mogą się przydać w systemach przetwarzających dane na żywo, na przykład w samochodach autonomicznych. W takich systemach należy szybko dokonać złożonych obliczeń, w których klasyczne algorytmy mogą nie być wystarczająco efektywne. Ponadto dane nie muszą być przechowywane długo, wystarczy użyć wyników do wpłynięcia na jakąś decyzję. Ponadto probabilistyczny charakter wyników, czy sporadyczne błędy obliczeń nie są problemem, gdyż obliczenia będą przeprowadzane cyklicznie, zatem wyniki będą uśrednianie.

Kwantowe obliczenia można użyć w algorytmach rojowych, do modelowania zachowania częstek. Zazwyczaj w obliczeniach kwantowych obwód jest uruchamiany wielokrotnie, aby uzyskać rozkład wartości wyjściowych. W przypadku algorytmów rojowych, wystarczający może być zaledwie jeden lub kilka wyników na każdą częstkę. Częstek jest wiele, zatem wiedza uzyskana z obliczeń kwantowych i tak się skumuluje w całym roju, a błędne zachowanie pojedynczej częstki nie będzie problemem, gdyż bardziej liczy się trend całej grupy. Tym bardziej, że te błędne wyniki niekoniecznie muszą być zupełnie przypadkowe, bo skoro się pojawiły, to znaczy, że w rozkładzie miały pewne prawdopodobieństwo wystąpienia, a to już jakaś informacja. Poza tym częstki w roju także korzystają z losowości, więc przynajmniej będzie to losowość wysokiej jakości, bo uzyskana kwantowo. W ten sposób można wykorzystać także wady sprzętu, aby zrobić coś pozytywnego.

Ostatni pomysł to konwersja obecnych algorytmów do postaci kwantowej, a także przegląd dotychczasowych wynalazków, czy nie było już takiego rozwiązania, które zostało odrzucone

z powodu braku odpowiedniej technologii lub uznane za zbyt wymagające jeśli chodzi o zasoby obliczeniowe. Technologia kwantowa zniesie niektóre z ograniczeń, zatem sensowne jest ponowne rozważenie ich zastosowalności.

Materiały dydaktyczne

Kolejny rozdział jest inny niż reszta pracy, gdyż nie zawiera opisu działań badawczych przeprowadzonych przez autora. Zawiera natomiast przykłady tematów i materiały, które w 2024 roku mogą być punktem startowym dla każdego, kogo interesują lub dopiero zainteresują obliczenia i technologie kwantowe. Ponadto zawiera wskazówki jak poruszać się po nieznanych obszarach aby sprawiało to radość i przynosiło efekty w postaci przyrostu wiedzy i zrealizowanych pomysłów. Ponieważ dziedzina się rozwija, a główni użytkownicy sprzętu i oprogramowania kwantowego to naukowcy i inżynierowie, o fizykach nie wspominając, dlatego łatwo można natrafić na materiały zawierające niezrozumiałe wzory, terminy czy koncepcje. Na szczęście nie trzeba wszystkiego rozumieć. Oto instrukcja.

Instrukcja do ćwiczeń

Bity i kubity

Na początek obejrzyj film <https://www.youtube.com/watch?v=JhHMJCUmq28>

W trakcie oglądania, zapisuj nazwy których nie znasz.

Poszukaj odpowiedzi na pytania:

- Czym kubit różni się od bitu?
- Co dzieje się ze stanem kubitu po dokonaniu obserwacji (pomiaru)?
- Ile informacji może przechowywać zestaw 4 kubitów?
a ile dziesięciu? dwudziestu? stu? tysiąca?

Poszukaj publicznych komputerów kwantowych w chmurze i zobacz ile mają kubitów?

Jeśli natrafiasz na nazwy i pojęcia których nie rozumiesz, koniecznie zapisuj je.

Jeśli przychodzą Ci do głowy pytania, to tym bardziej je zapisuj.

W wolnym czasie poszukaj wyjaśnień tych pojęć, nie musisz ich rozumieć dokładnie, wystarczy ogólne wyjaśnienie. Jeśli możesz, korzystaj z naukowych baz danych i publikacji. Jeśli nie, zapytaj o taki dostęp w swojej szkole lub uczelni.

Obejrzyj film <https://www.youtube.com/watch?v=OWJCfOvochA&t=1s>

Superpozycja

Dowiedz się, co to znaczy, że kubit może przebywać w superpozycji. Poszukaj jak zapisuje się matematycznie stan kubitu w superpozycji (tak z grubsza). Co znaczy $|0\rangle$ i $|1\rangle$?

Otwórz w przeglądarce symulator obwodów kwantowych <https://oreilly-qc.github.io/>

Uruchom przykład 2-1 przedstawiający 1 kubit, bramkę Hadamarda i odczyt wartości kubitu.

Uruchom obwód wielokrotnie.

- Jak często pojawia się “0” i “1”?
- Co to jest bramka Hadamarda, jak ona działa?

Przyjrzyj się wykresowi kołowemu, poszukaj co oznaczają poszczególne elementy.

Prześledź działanie obwodu krok po kroku, jak poszczególne operacje modyfikują stan kubitu.

W jakim stanie jest kubit tuż przed odczytem? A co dzieje się po odczycie?

Zmień początkowy stan kubitu na “1”. Zobacz co zmienia się w przebiegu programu i wykresie kołowym? Poczytaj sobie o fazie względnej i globalnej.

Usuń bramkę H i uruchom kilka razy obwód. Dlaczego zawsze otrzymujemy ten sam stan?

Jeśli masz jakieś pytania, zapisuj je, a potem poszukaj odpowiedzi. Jeżeli znalezione źródła pochodzą od społeczności, to poszukaj większej ilości źródeł w celu weryfikacji informacji.

Rejestr wielo-kubitowy

Uruchom w symulatorze przykład 2-2. Zmień rozmiar rejestru z 8 na 4 kubity. Dlaczego teraz jest 16 kół, z czego to wynika? Jak to ma się do ilości informacji którą rejestr może

przechowywać? Uruchom obwód kilkadziesiąt razy i zapisz jakie wartości przyjmują poszczególne kubity po odczycie i jak to się ma do wykresu kołowego.

Wybieraj na schemacie kolejno poszczególne momenty i zobacz jak zmienia się stan rejestru.

Zmień qc.had() na qc.had(N), gdzie N to potęga liczby 2.

<https://www.memorizer.pl/tekst/18104/potegi-liczby-2/> Jak zmienił się obwód? Uruchom obwód kilkanaście razy i zapisz wszystkie wyniki. Jakią wartością się pojawiają? Zmień N na inną potęgę liczby 2 i powtórz eksperyment. Co się zmieniło? Dlaczego tak jest? W miejscu N wpisz liczbę będącą sumą 2 dowolnych potęg liczby 2 (na przykład 12). Jak zmieni się schemat obwodu? W jakim stanie jest rejestr tuż przed odczytem? Jakią wartością pojawiają się w wynikach? Dlaczego tak jest?

Poszukaj odpowiedzi na wszystkie pytania i wątpliwości które Ci się nasunęły.

2 bramki Hadamarda

Otwórz symulator i przykład 2-1.

Przerób obwód tak, żeby była parzysta ilość połączonych bramek H, jedna za drugą. Porównaj stan wejściowy, ze stanem po 1 bramce, pod 2-ch, itd. Co zauważasz?

Usuń lub dodaj 2 bramki H do szeregu. Czy coś się zmieniło w wynikach?

Zapoznaj się z pojęciem odwracalności obliczeń kwantowych. Dowiedz się jaka bramka odwraca działanie H. Poszukaj jakie jeszcze są bramki kwantowe, jak działają i co jest ich odwrotnością? Jak tworzy się całe obwody odwracające obliczenia kwantowe?

Jeśli masz chęci i znasz macierze to poszukaj macierzowych opisów bramek jedno-kubitowych i jak taka macierz przekształca stan wejściowy w stan wyjściowy. Poszukaj opisu macierzowego zastosowania na kubicie kolejno 2 bramek H.

https://pl.wikipedia.org/wiki/Bramka_Hadamarda

Bramka NOT i pierwiastek z NOT

Poszukaj jak działa klasyczna bramka NOT, znajdź tabelę prawdy dla tej bramki. Co daje połączenie 2 bramek NOT jedna za drugą?

Uruchom obwód z przykładu 2-3. Prześledź krok po kroku jak poszczególne operacje zmieniają stan kubitu. Zmodyfikuj bramkę RNOT, używając mniejszego kąta i nazwij ją po swojemu. Połącz szeregowo tyle bramek aby suma zastosowanych kątów wynosiła razem minus 180 stopni. Zobacz po kolei jak zmienia się stan kubitu i czy na końcu jest zawsze $|0\rangle$.

Poszukaj schematu zastępczego bramki NOT złożonego z H i PHASE. Załaduj w symulatorze jeszcze raz przykład 2-3. Czy zauważasz podobieństwo RNOT do NOT? Dlaczego ktoś mógł chcieć nazwać tą bramkę pierwiastkiem z NOT?

Porównaj stan kubitu na wykresie kołowym w miejscu przed drugą bramką H ze stanem po trzeciej bramce H? Czy jest taki sam? Co się stanie jeśli z obwodu usuniemy 2 i 3 bramkę H? Czy wpływa to na wynik na końcu obwodu? Co się stanie jeżeli zastąpisz 2x PHASE(-90) przez jedną PHASE(-180)? Czy taki obwód będzie działać (i wyglądać) jak pojedyncza bramka NOT? Co będzie jak zamiast ujemnych kątów użyjemy dodatnie?

Pobaw się jeszcze bramkami, kątami, wypróbowuj swoje pomysły, obserwuj co jak wpływa na stan kubitu. Jak czegoś nie rozumiesz, zapisz to i później poszukaj odpowiedzi.

Splątanie kubitów

Uruchom w symulatorze obwód 3-2. Na początek usuń (zakomentuj "//") bramkę H i pobaw się samą bramką CNOT (warunkowe NOT). Wypróbuje qc.write(N) gdzie N od 0 do 3. Wyjaśnij powstające stany wyjściowe. Prześledź cały obwód krok po kroku. Dlaczego po bramce CNOT stany $|1\rangle$ i $|3\rangle$ zamieniają się miejscami? Jeśli jest to niejasne, to w instrukcji cnot, w miejscu a wpisz b, a w miejscu b wpisz a. Jak zmieniło to wygląd i działanie bramki CNOT? Który kubit jest teraz kontrolny? Które stany zamieniają się teraz miejscami? Dlaczego?

Zresetuj przykład do początkowego stanu i zobacz jak działa obwód z bramką H. Poszukaj co to jest splątanie kwantowe. Prześledź obwód krok po kroku, zwłaszcza oddzielny odczyt obu kubitów, śledząc też wykres kołowy. Dlaczego odczyt obu kubitów wywołuje zawsze identyczny wynik?

Wymyśl jak splątać kubity tak "żeby się nie lubiły". To znaczy aby zawsze uzyskiwać przeciwne stany. Jeśli nie masz pomysłu, wypróbuje różne stany początkowe N.

Stopniowe określanie stanu rejestru

Wyczyść okno kodu i uruchom ten kod. Prześledź jak zmienia się stan rejestru, odczytując na schemacie obwodu po jednym kubicie na raz.

```
qc.reset(4); // allocate some qubits  
qc.write(0); // write the value zero  
qc.had(); // place them all into superposition of 0 and 1  
qc.read(1);  
qc.read(2);  
qc.read(4);  
qc.read(8);
```

Obserwuj na wykresie kołowym jak odczyt kubitów eliminuje pewne stany, pozostawiając na końcu tylko jeden stan wyrażony odczytanymi wartościami kubitów. Które stany są eliminowane, a które pozostają? Zmień kod, aby kubity były odczytywane w odwrotnej kolejności. Do qc.had() wpisz taką liczbę aby jedna z bramek H zniknęła. Zobacz teraz jak odczyt kolejnych kubitów zmienia stan rejestru. (określonego też rozkładem prawdopodobieństwa poszczególnych stanów). Czy teraz odczyt któregoś z kubitów zawsze zmienia stan rejestru? Jeśli nie, to które kubity są inne?

Jeśli chcesz, możesz potem jeszcze raz wrócić do ćwiczenia "splątanie kubitów", aby lepiej zrozumieć jak na wykresie kołowym działa pojedyncze odczytywanie kubitów.

Odbicie fazowe

Uruchom w symulatorze przykład 3-3. Prześledź krok po kroku jak działa obwód. Na które stany działają poszczególne instrukcje warunkowej rotacji fazy? Jaka jest reprezentacja binarna tych stanów?

Zakomentuj instrukcję had, następnie wypróbuje różne wartości w reg1.write. Jak zmienia się faza? Do reg2.write wpisz 0. Czy faza dalej się zmienia? Wpisz do reg1 i reg2 zera. Odkomentuj i zmień linię zawierającą had na qc.had(). Jak teraz wygląda wykres kołowy? Dowiedz się więcej o warunkowej rotacji fazy. Pobaw się i pozmieniaj różne rzeczy w obwodzie według własnego uznania. Jak znajdziesz coś ciekawego to poszukaj wyjaśnienia dlaczego tak jest.

Test swap

Wybierz przykład 3-4. Prześledź obwód, jak zmieniają się stany. Upewnij się, że dla dowolnych dwóch identycznych stanów na wejściach, wynikiem testu będzie zawsze "1". Zmień stany na wejściu na dwa różne. Prześledź jeszcze raz obwód, oszacuj prawdopodobieństwo pojawienia się "0" na wyjściu testu na podstawie wykresu kołowego tuż przed odczytem, a potem sprawdź to praktycznie.

Możesz wybierać na przemian moment przed i po odczycie. Cofnięcie pomarańczowego markera nie spowoduje cofnięcia operacji odczytu, gdyż jest ona nieodwracalna, ale uruchomi cały obwód ponownie aż do wybranego momentu. Przyjrzyj się wykresowi kołowemu i wyjaśnij dlaczego w ogóle może pojawić się "0" na wyjściu testu. porównaj z przypadkiem gdy stany wejściowe są identyczne.

Wpisz na wejścia dwie "1". Do jednego z rejestrów wejściowych dodaj rotację fazy o 30-180 stopni. Prześledź obwód, zobacz czy obwód potrafi wykryć, że stany się różnią fazą. Ustaw oba wejścia z powrotem na "0". dodaj do jednego z wejść (bezpośrednio za zapisem stanu) obwód podobny do NOT, ale z kątem 30 stopni. Przyjrzyj się jak wygląda wykres kołowy na każdym kroku. Zobacz czy obwód wykryje różnicę i jak często się to będzie działo. Zmień kąt na 60 stopni i zobacz jeszcze raz. Zastanów się, dlaczego po wykonaniu SWAP, w niektórych przypadkach test może dać "0"? Jeśli nie wiesz, przyjrzyj się uważnie jak zmienia się stan rejestru w każdym kroku, oraz które koła zamieniają się miejscami. Poszukaj wyjaśnienia testu swap i do czego może być stosowany. Zobacz też https://en.wikipedia.org/wiki/Swap_test

Teleportacja kubitu

Na początek poczytaj o protokole teleportacji, jak działa algorytm, jakie informacje są przesyłane, co się po kolei dzieje.

Na przykład tu <https://www.sciencedirect.com/topics/engineering/quantum-teleportation>

Uruchom przykład 4-1. Jeżeli nie rozumiesz jak działa kwantowa teleportacja, oto krótkie wyjaśnienie:

- trzeba przesłać (teleportować) informację z kubitu "alice", do kubitu "bob"
- kliknij na schemacie w koniec "prep payload"
- Informacja wygląda tak jak w $|0\rangle$ i $|1\rangle$, zapamiętaj ją
- kliknij w koniec "send"
- Informacja została przesłana, ale może być w jednej z 4 alternatywnych wersji
- kliknij w koniec "receive"
- Informacja została doprowadzona do właściwej formy na podstawie 2 bitów od nadawcy

- weryfikacja jest odwróconym obwodem który doprowadza odebrany bit do stanu $|0\rangle$, co można sprawdzić, bo widać to na wykresie kołowym

Jeżeli wciąż nie rozumiesz to nie zniechęcaj się. Jeszcze dokładniej prześledź krok po kroku jak poszczególne bramki zmieniają stany kubitów. Wykres kołowy przedstawia stan wszystkich 3 kubitów, są one splątane, więc nie da się ich stanu analizować oddzielnie. Możesz też zakomentować część kodu, np funkcję entangle, co pomoże analizować mniejsze fragmenty. Przypomnij też sobie poprzednie ćwiczenia.

Inkrementacja i dekrementacja

Uruchom przykład 5-1. Zawiera on bramki CNOT ale z większą ilością wejść. Dowiedz się jak taka bramka działa i kiedy. Zobacz co inkrementacja robi z rejestrów będącymi w superpozycji. Dlaczego dekrementacja wygląda podobnie, tylko lustrzanie? Pozmienią ją w kodzie jaką liczbę dodawać do rejestrów z 1 na większe liczby, np 2, 5, 8, 15. Dlaczego dodawanie 15 wygląda jak dekrementacja o 1? Dlaczego inkrementacja o 2 wygląda jak o 1, tylko bez najmłodszego kubitu? Zwiększy rejestr do 8 kubitów. Pobaw się większymi liczbami. Spróbuj wyjaśnić zasadę działania inkrementacji w ogólności w pozycyjnym systemie binarnym. Czym jest inkrementacja w binarnym systemie pozycyjnym? (Wskazówka: Najprościej, polega ona na znalezieniu N najmłodszych bitów które mają wartość "1", oraz negacji N+1 najmłodszych bitów (czyli negujemy wszystkie "1" i dodatkowo o 1 bit więcej)). Zastanów się czy to stwierdzenie jest rzeczywiście prawdziwe. Porównaj je z działaniem obwodu wyrażonego schematem z przykładu.

Możesz poszukać albo wymyślić jak zrobić obwód, który dodaje do rejestrów liczbę zapisaną w innym rejestrze. Poszukaj też odpowiedzi na pytania i wątpliwości.

Dodawanie liczb

Jeśli w poprzednim ćwiczeniu próbowałeś stworzyć obwód dodawania, to teraz możesz porównać go z przykładem 5-2, będącym tematem obecnego ćwiczenia. Rozciągnij wykres kołowy aby zobaczyć wszystkie 64 koła. Porównaj stan przed i po dodaniu. Zobacz które koła się przesuwają, i o ile. Zakomentuj H i PHASE, następnie rozważaj przypadki osobno, bez superpozycji stanów. (Przy okazji możesz później dowiedzieć się dlaczego PHASE działa tylko na stan "1"). Ustaw rejestr b na "0" i dodaj bramki H na każdym kubicie b. Porównaj stan przed i po dodaniu. Ustaw a na "0" i dodaj bramki H na 2 najmłodszych kubitach a. Porównaj stan przed i po dodaniu. Zastąp instrukcje dotyczące a, następującym kodem:

```
qc.label('prepare');
a.write(0);
a.had(3);
a.phase(45,1);
a.phase(90,2);
```

Zobacz efekt na wykresie kołowym. Instrukcje dotyczące b, zastąp kodem:

```
b.write(0);
```

```
b.had();
b.phase(60,1);
b.phase(30,2);
b.had();
```

Zobacz jak teraz wygląda wykres kołowy. Dla rejestru a, pozostaw tylko instrukcję a.write(0), a resztę zakomentuj. Zobacz jak wygląda wykres przed i po dodaniu. Pobaw się jeszcze, wprowadzając różne modyfikacje i sprawdzając efekty. Czy potrafisz już interpretować stan rejestru na podstawie wykresu kołowego? Jeśli nie, to wykonaj jeszcze raz kroki z tego ćwiczenia i poszukaj teorii o stanach rejestru, czy interpretacji wykresów kołowych. Jeśli rozumiesz wszystko, spróbuj trudniejszego przykładu 5-3.

Znajdź błąd w obwodzie kwantowym

Przykład 5-4 zawiera obwód dokonujący inkrementacji b, tylko wtedy gdy $a < 3$. Błąd polega na tym, że nie jest tak dla każdego a. Trzeba znaleźć dla jakiej wartości a, obwód działa inaczej niż podaje jego oficjalny opis.

Na początek, zauważ, że wykres kołowy zawiera 8 kolumn i 8 wierszy. Tak się szczegółowo składa, że numer kolumny odpowiada wartości binarnej rejestru a, natomiast numer wiersza wartości b. Zatem zapełnione koła leżące na przecięciu wierszy i kolumn, określają wszystkie możliwe kombinacje a i b. Jeżeli koło przesuwa się w prawo, to oznacza to inkrementację a, bez zmiany b. Jeżeli w dół, to b rośnie przy niezmienionym a.

Przejdźmy zatem do obwodu. Prześledź go krok po kroku. Zobacz co się dzieje, jak badany jest warunek $a < 3$. Dlaczego lewe koła się obniżyły a prawe nie?

Popraw w 17 linii (1, 5) na (1, 7). Następnie między 19 i 20 linię wstaw

```
a.cnot(2,4);
```

Dlaczego koło określające a=7 obniżyło się, chociaż $7 > 3$?

Popraw w 17 linii (1, 7) na (0-7). W linii 18 zmień 1 na 0. Zakomentuj linię 20.

Linię 19 popraw na

```
a.hadamard();
```

Uruchom obwód i korzystając z wykresu kołowego sprawdź dla jakich "a" następuje inkrementacja "b"? (patrz wskazówka na początku). Czy to jest zgodne z opisem obwodu? Jak poprawić obwód aby działał zgodnie z opisem? Gdybyś miał z tym problem to wypróbowuj to (należy dodać na końcu, ale możesz wstawić gdziekolwiek i sprawdzić co się stanie)

```
qc.label("");
qc.nop();
qc.label('korekcja');
qc.cnot(8,7);
qc.cnot(16,15);
qc.cnot(32,31);
```

Wyjaśnij jak działa korekcja. Jak można by przerobić korekcję gdybyśmy chcieli skorygować stan inny niż 7? (wskazówka: bramki NOT przed i po)

Zobacz czy można inkrementować b o inną liczbę?

Będziesz też musiał poprawić korekcję odpowiednio (wzoruj się na 4 bloku, tylko dekrementuj).

Zapamiętaj: Ustawiając rejestr "a" w każdym możliwym stanie, wykonaliśmy obliczenia w superpozycji do zbadania warunkowości inkrementacji "b", w zależności od wszystkich wartości "a". Następnie w ten sam sposób zbadaliśmy poprawiony obwód. Jest to kwantowa równoległość obliczeń niedostępna na zwykłym komputerze.

Flip fazy i iteracja wyszukiwania Grovera

Zobacz przykład 5-5. Przyjrzyj się jak działa. Ten obwód odwraca fazę o 180 stopni (koło staje się zielone), ale robi to tylko przy spełnieniu pewnych warunków. Faza będzie odwrócona tylko gdy 3 kubity od dołu będą mieć "1". Przeanalizuj obwód dokładnie i zapamiętaj jak działa.

Otwórz przykład 6-2, Każda iteracja zawiera podobny flip fazy, a bramki NOT pozwalają wybrać dowolne koło. Jednak zasadniczym elementem jest funkcja Grover(). Zobacz jak Grover potrafi wyszukać stan z odwróconą fazą. Przejrzyj cały obwód i znajdź optymalną ilość iteracji aby uzyskać największe prawdopodobieństwo odczytu oznaczonej wartości. Poszukaj informacji o algorytmie Grovera.

Zmień liczbę kubitów na większą, zwiększą także liczbę iteracji. Potestuj obwód.

Poszukaj w literaturze wzoru na optymalną liczbę iteracji.

Zobacz też przykład 6-3, z oznaczaniem kilku wartości.

Dowiedz się, jaki jest efekt odczytywania rejestrów po wyszukiwaniu Grovera. W obwodach 6-2 i 6-3 dobierz optymalną liczbę iteracji (wyświetla się w output), a potem dodaj na końcu krok odczytu wszystkich kubitów. Zobacz jakie wartości się pojawiają gdy:

- jest jedna oznaczona wartość
- jest kilka oznaczonych wartości

Uruchamiaj obwód wielokrotnie i zapisuj otrzymane stany. Podsumuj ile było i jakich stanów.

Kwantowe badanie wyrażeń logicznych

Zbadaj, czy wyrażenie $Y = (A \cup \overline{B}) \cap C$ jest prawdziwe dla jakiejś kombinacji A, B i C. Logika ta została już zaimplementowana w przykładzie 10-1. Przyjrzyj się jak zbudowany jest obwód. Dlaczego ostatnie działanie AND jest w formie fazowej? Dlaczego cofamy obliczenia? Uruchom przykład i zobacz które stany mają odwróconą fazę. Wyznacz jakie są wartości kubitów A, B i C w tych stanach, podstaw wartości do wzoru i sprawdź czy dla tych stanów wychodzi $Y = "1"$, a dla wszystkich pozostałych $Y = "0"$.

Przerób funkcję pAND żeby wyglądała tak:

```
function phase_and(qubits)
{
    qc.cphase(45, qubits);
}
```

Zobacz jak wygląda wykres kołowy, a potem zresetuj obwód do oryginalnego stanu. Napisz funkcję $(a \text{ XOR } b)$ i użyj jej w miejsce poprzedniej, tak aby zaimplementować $Y = (A \oplus B) \cap C$. Jeżeli wszystko będzie dobrze policzone, to fazy $|5\rangle$ i $|6\rangle$ powinny zostać odwrócone. Zrób wyszukiwanie Grovera na tym lub na oryginalnym obwodzie. Ważne: W symulatorze jest błąd i qc.Grover(7) może nie działać. Na szczęście można zrobić go samemu.

```
qc.had(7);
qc.not(7);
qc.cphase(180,7);
qc.not(7);
qc.had(7);
qc.read();
```

Sprawdź czy odpowiednie stany są znajdowane. Jeżeli obwód często zwraca zły wynik to znaczy, że ilość iteracji Grovera jest nieoptymalna. Usuń qc.read(), następnie na końcu dodaj jeszcze raz zawartość linii 20-35 i na koniec dodaj powyższy kod Grovera. Dodaj tyle powtórzeń ile potrzeba, aby amplitudy nieoznaczonych stanów były jak najmniejsze. Okazuje się że bramki NOT i CNOT nie zakłócą stanów fazowych. Jedyną operacją fazową jest pAND. Dodaj za dużo iteracji i zobacz jak Grover “psuje” amplitudy oznaczonych stanów.

A czy wiesz co się stanie, jeżeli uruchomisz wyszukiwanie Grovera kiedy żadna faza nie została odwrócona?

Pozostałe obwody

Wybierz któryś z obwodów i zobacz jak działa. Poczytaj jak działa transformata Fouriera i zobacz jakieś obwody, w których jest używana. Poszukaj wiedzy na tematy związane z obwodami, algorytmami czy pytaniami które masz.

Qiskit

Poszukaj informacji o programowaniu w języku Python. Załącz sobie konto Google. Otwórz Google Collab i stwórz nowy notatnik. Napisz jakiś program w Pythonie i uruchom go.

Poszukaj informacji o Qiskit. Zwróć uwagę na wersję, gdyż ta biblioteka zmienia się bardzo szybko, a dokumentacja (zwłaszcza nieoficjalna) niekoniecznie nadąża za zmianami.

Uruchom w notatniku kod:

```
!pip3 install qiskit qiskit-aer pylatexenc
```

```
from qiskit import QuantumCircuit
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
```

```
obwod = QuantumCircuit(4)
```

```
obwod.h([0,2])
obwod.ccx(0,2,1)
obwod.barrier()
obwod.ccx(1,2,3)
obwod.cx(1,2)
obwod.x(1)
obwod.measure_all()
obwod.draw(output="mpl", initial_state=True, idle_wires=False)
```

```
# KONIECZNIE w kolejnej komórce
qpu = AerSimulator()
job = qpu.run([obwod], shots=6000)
wyniki = job.result()
plot_histogram(wyniki.get_counts())
```

Co robi ten kod? Co robi symulowany obwód kwantowy? Ile razy symulator uruchomił obwód? Jakie stany zostały odczytane?

Poszukaj innych programów na QPU i spróbuj je uruchomić, lub wymyśl własny algorytm, lub spróbuj uruchomić przykłady z symulatora używanego wcześniej. Poszukaj odpowiedzi na pytania, które Ci przyjdą do głowy.

Dostęp do komputera kwantowego

Na stronie IBM, poszukaj instrukcji jak połączyć się z ich komputerem. Utwórz darmowe konto w IBM Quantum, korzystając z wcześniej utworzonego konta Google. Możesz też wybrać innego dostawcę. Kod Python jest podobny jak do AerSimulator, ale backend będzie inny. Oto różnice:

- trzeba zaimportować moduł pythona dostawcy obliczeń kwantowych i drugi do transpilacji
- obiekt qpu należy uzyskać według instrukcji od dostawcy, używając indywidualnego tokenu
- należy dokonać transpilacji obwodu na hardware dostawcy
- na wynik obliczeń trzeba poczekać, zatem potrzebny jest dodatkowy kod umożliwiający pobranie id zadania po wysłaniu, a wyników po zakończeniu obliczeń

Poza tym jest podobnie jak na symulacji, zatem warto testować obwody na symulatorze.

Pobierz sobie indywidualny token do API usługi IBM Quantum lub innej.

Szczegółowa instrukcja obsługi QPU może się niedługo zmienić, obecnie przydatne mogą okazać się rysunki 60, 62 i 65.

Zobacz jaki sprzęt kwantowy jest dostępny u wybranego dostawcy, ile posiada kubitów, jaką ma szybkość obliczeń, poziom zaszumienia, czy obsługiwane instrukcje.

Przy wysyłaniu zadań zwróć uwagę na długość kolejki albo czas oczekiwania.

Dobieraj rozsądnie ilość shotów aby zmieścić się w darmowym limicie obliczeń.

Poszukaj algorytmów do uruchomienia na fizycznym QPU

Poszukaj co jest aktualnie dostępne w bibliotece Qiskit. Przykładowe algorytmy to:

- Faktoryzacja Shora (tylko w starych wersjach)
- Rozwiązywanie układów równań liniowych
- Transformata Fouriera
- Uczenie maszynowe (QSVM)
- Kwantowy supersampling
- Algorytmy wyszukujące
- Rozwiązywanie obwodów logicznych

Poszukaj innych projektów rozszerzających możliwości Qiskit, na przykład moduły do korzystania z innych dostawców, algorytmy radzenia sobie z szumem.

Zobacz czym różni się prymityw Estimator od Sampler.

Dowiedź się o optymalizacji obwodów, albo o tzw. PulseProgramming.

Wypróbuj też inne toolkiti niż Qiskit, np Cirq.

Poszukaj rozwiązań wykorzystujących korekcję szumu, wirtualne/logiczne kubity, czy inne ciekawostki jakie Ci przyjdą do głowy. Baw się dobrze. To już wszystkie ćwiczenia. Dalej możesz poznawać świat kwantowy samodzielnie.

Jak lubisz Minecrafta to zobacz QiskitBlocks <https://github.com/JavaFXpert/QiskitBlocks>

Spis literatury

- [1] Mario Coccia, Saeed Roshani, and Melika Mosleh, "Evolution of Quantum Computing: Theoretical and Innovation Management Implications for Emerging Quantum Industry" IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 71, pp. 2270-2280, 2024
- [2] Hiroyuki Mizuno (hiroyuki.mizuno.vp@hitachi.com), "Quantum Computing from Hype to Game Changer", JSAP 2023 Symposium on VLSI Technology and Circuits Digest of Technical Papers 978-4-86348-806-9, Center for Exploratory Research, R&D Group, Hitachi, Ltd., Tokyo, Japan, ©2023
- [3] Swati Arya¹, Syed Anas Ansar², Shruti Aggarwal³, "Quantum Odyssey: Traversing the NISQ Era's Quantum Terrain", Proceedings of the 2023 3rd International Conference on Technological Advancements in Computational Sciences, IEEE Conference ID: 59847 1st – 3rd Nov. 2023
- [4] Zebo Yang, Maede Zolanvari, Raj Jain, "A Survey of Important Issues in Quantum Computing and Communications", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 25, NO. 2, SECOND QUARTER 2023
- [5] Microsoft, "What is a Qubit?", Microsoft Azure, (dostęp: 21-03-2024)
<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-qubit>
- [6] Heike Riel, "Quantum Computing Technology and Roadmap", IBM Research, Rüschlikon, Switzerland, ESSDERC 2022 - IEEE 52nd European Solid-State Device Research Conference (ESSDERC) | 978-1-6654-8497-8/22/\$31.00 ©2022 IEEE | DOI: 10.1109/ESSDERC55479.2022.9947181, 2022
- [7] Beatrice Branchini, Davide Conficconi, Francesco Peverelli, Donatella Sciuto, Marco D. Santambrogio, "A Bird's Eye View on Quantum Computing: Current and Future Trends", Dipartimento di Elettronica, Informatica e Bioingegneria, Politecnico di Milano, Milan, Italy, IEEE EUROCON 2023 - 20th International Conference on Smart Technologies 978-1-6654-6397-3/23/\$31.00, DOI: 10.1109/EUROCON56442.2023.10198957, 2023
- [8] Xiaorang Guo, Kun Qin, Martin Schulz, "HiSEP-Q: A Highly Scalable and Efficient Quantum Control Processor for Superconducting Qubits", School of Computation, Information and Technology, Technical University of Munich; Leibniz Supercomputing Centre Garching, Germany, 2023 IEEE 41st International Conference on Computer Design (ICCD) | 979-8-3503-4291-8/23/\$31.00 ©2023 IEEE | DOI: 10.1109/ICCD58817.2023.00023, 2023
- [9] Gang Huang, Yilun Xu, Neelay Fruitwala, Abhi D. Rajagopala, Kasra Nowrouzi, Ravi K. Naik, David Santiago, Irfan Siddiqi, "QubiC 2.0: A Flexible Advanced Full Stack Quantum Bit Control System", Lawrence Berkeley National Laboratory, Berkeley, CA, USA; University of California at Berkeley, Berkeley, CA, USA, 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3503-4323-6/23/\$31.00 ©2023 IEEE | DOI: 10.1109/QCE57702.2023.10227, 2023

- [10] Zhirui Hu, Robert Wolle, Mingzhen Tian, Qiang Guan, Travis Humble, Weiwen Jiang, "Toward Consistent High-fidelity Quantum Learning on Unstable Devices via Efficient In-situ Calibration", Department of Electrical and Computer Engineering, George Mason University, VA, USA, Quantum Science and Engineering Center, George Mason University, VA, USA, Department of Physics & Astronomy, George Mason University, VA, USA, Department of Computer Science, Kent State University, OH, USA, Oak Ridge National Laboratory, TN, USA, 2023 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3503-4323-6/23/\$31.00 ©2023 IEEE | DOI: 10.1109/QCE57702.2023.00099, 2023
- [11] STEFANIE CASTILLO, "The Electronic Control System of a Trapped-Ion Quantum Processor: A Systematic Literature Review", Innsbruck Power Electronics Laboratory, Institute of Mechatronics, University of Innsbruck, 6020 Innsbruck, Austria, Digital Object Identifier 10.1109/ACCESS.2023.3289936, current version 6 July 2023
- [12] Bharat S Rawal, "Quantum Integrated (C+G+Q)PU Split Architecture", Benedict College, Columbia, SC, USA, 2023 International Wireless Communications and Mobile Computing (IWCMC) | 979-8-3503-3339-8/23/\$31.00 ©2023 IEEE | DOI: 10.1109/IWCMC58020.2023.10183269, 2023
- [13] Eisuke Abe, "Superconducting route to quantum computing", RIKEN Center for Quantum Computing, Wako, Saitama 351-0198, Japan, 2023
- [14] WEN-HAN PNG, TING HSU, TZE-WEI LIU, GUIN-DAR LIN, MING-SHIEN CHANG, "Quantum Computing With Trapped Ions, An overview", 30 | IEEE NANOTECHNOLOGY MAGAZINE | AUGUST 2022 1932-4510/22, Digital Object Identifier 10.1109/MNANO.2022.3175384, 2022
- [15] Stavroula Kapoulea, Meraj Ahmad, Martin Weides, Hadi Heidari, "Cryo-CMOS Mixed-Signal Circuits for Scalable Quantum Computing: Challenges and Future Steps", James Watt School of Engineering, University of Glasgow, Glagsow G12 8QQ, UK, 2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/\$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10182164, 2023
- [16] Daniel Volya, Tao Zhang, Nashmin Alam, Mark Tehranipoor, Prabhat Mishra, "Towards Secure Classical-Quantum Systems", Department of Computer & Information Science & Engineering; Department of Electrical & Computer Engineering, University of Florida, Gainesville, Florida, USA, 2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) | 979-8-3503-0062-8/23/\$31.00 ©2023 IEEE | DOI: 10.1109/HOST55118.2023.10133344, 2023
- [17] ZIHAN QU, MENGQIN LAI, "A Review on Electromagnetic, Acoustic, and New Emerging Technologies for Submarine Communication", School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; Sichuan Inspection and Testing Center for Dental Devices and Materials, Jiangyou, Ziyang 641300, China, IEEE Access, Digital Object Identifier 10.1109/ACCESS.2024.3353623, 2024

[18] Miralem Mehic, Libor Michalek, Emir Dervisevic, Patrik Burdiak, Matej Plakalovic, Jan Rozhon, Nerman Mahovac, Filip Richter, Enio Kaljic, Filip Lauterbach, Pamela Njemcevic, Almir Maric, Mirza Hamza, Peppino Fazio, Miroslav Voznak, "Quantum Cryptography in 5G Networks: A Comprehensive Overview", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 26, NO. 1, FIRST QUARTER 2024

[19] Ekin Arabul, Romerson D. Oliveira, Amin Emami, Stavros Typos, Constantinos Vrontos, Rui Wang, Reza Nejabati, Dimitra Simeonidou, "100 Gbps quantum-secured and O-RAN-enabled programmable optical transport network for 5G fronthaul", High Performance Network Group, University of Bristol, Woodland Road, Bristol, UK, Vol. 15, No. 8 / August 2023 / Journal of Optical Communications and Networking C223, 2023

[20] O'Reilly, Symulator obliczeń kwantowych QC Engine, przykład 4-1, prosta teleportacja, (dostęp: 10-04-2024), <https://oreilly-qc.github.io/>

[21] Mekala Karthik, Jitesh Lalwani, Babita Jajodia, "Quantum Text Teleportation Protocol for Secure Text Transfer by using Quantum Teleportation and Huffman Coding", Artificial Brain Tech Inc, 2055 Limestone RD, STE 200-C, Wilmington, Delaware, USA 19808; Artificial Brain Technology (OPC) Private Limited, Pune, India 411057; Department of Electronics and Communication Engineering, Indian Institute of Information Technology Guwahati, India, 2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT) | 978-1-6654-5361-5/22/\$31.00 ©2022 IEEE | DOI: 10.1109/TQCEBT54229.2022.10041500, Pune Lavasa Campus, India. Oct 14-15, 2022

[22] Mekala Karthik, Jitesh Lalwani, Babita Jajodia, "Quantum Image Teleportation Protocol (QITP) and Quantum Audio Teleportation Protocol (QATP) by using Quantum Teleportation and Huffman Coding", Artificial Brain Tech Inc, 2055 Limestone RD, STE 200-C, Wilmington, Delaware, USA 19808; Artificial Brain Technology (OPC) Private Limited, Pune, India 411057; Department of Electronics and Communication Engineering, Indian Institute of Information Technology Guwahati, India, 2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT) | 978-1-6654-5361-5/22/\$31.00 ©2022 IEEE | DOI: 10.1109/TQCEBT54229.2022.10041599, Pune Lavasa Campus, India. Oct 14-15, 2022

[23] Yangming Zhao, Chunming Qiao, "Distributed Transport Protocols for Quantum Data Networks", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 31, NO. 6, DECEMBER 2023, Digital Object Identifier 10.1109/TNET.2023.3262547, 2023

[24] Chonggang Wang, Akbar Rahman, "Quantum-Enabled 6G Wireless Networks: Opportunities and Challenges", 1536-1284/22/\$25.00 © 2022 IEEE IEEE Wireless Communications, Digital Object Identifier: 10.1109/MWC.006.00340, February 2022

[25] TED H. SZYMANSKI, "The "Cyber Security via Determinism" Paradigm for a Quantum Safe Zero Trust Deterministic Internet of Things (IoT)", Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada, Digital Object Identifier 10.1109/ACCESS.2022.3169137, May 4, 2022

- [26] Yuan Cao, Yongli Zhao, Qin Wang, Jie Zhang, Soon Xin Ng, Lajos Hanzo, “The Evolution of Quantum Key Distribution Networks: On the Road to the Qinternet”, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 24, NO. 2, SECOND QUARTER 2022, Digital Object Identifier 10.1109/COMST.2022.3144219, 2022
- [27] Nicholas Harrigan, Eric R. Johnston, Mercedes Gimeno-Segovia, “Komputer kwantowy. Programowanie, algorytmy, kod”, O'Reilly, Helion, ISBN: 9788328367784, December 2020
- [28] Matemaks.pl, Matematyka maksymalnie prosta, Algorytm NWD, (dostęp: 8-05-2024), <https://www.matemaks.pl/algorytm-euklidesa.html>
- [29] SYED FARAH NAZ, AMBIKA PRASAD SHAH, “Reversible Gates: A Paradigm Shift in Computing”, IC-ResQ Lab., Department of Electrical Engineering, Indian Institute of Technology Jammu, Jammu and Kashmir 181221, India, Digital Object Identifier 10.1109/OJCAS.2023.3305557, 2023
- [30] IBM, Qiskit toolkit, (dostęp: 11-05-2024), <https://www.ibm.com/quantum/qiskit>
- [31] Quantum Algorithm Zoo, (dostęp: 13-05-2024), <https://quantumalgorithmzoo.org/>
- [32] Collab, Google, (dostęp: 13-05-2024), <https://colab.google/>
- [33] Microsoft Azure, (dostęp: 13-05-2024), <https://portal.azure.com/#home>
- [34] Google Cloud Console, (dostęp: 13-05-2024),
<https://console.cloud.google.com/welcome/new>
- [35] IBM Quantum, (dostęp: 13-05-2024), <https://quantum.ibm.com/>

Spis rysunków i wzorów

- (Rys. 1) Kierunki rozwoju technologii kwantowych w latach '10, źródło [1]
- (Rys. 2) Postęp technologiczny i epoki komputerów kwantowych, źródło [2]
- (Rys. 3) Architektura kwantowego systemu operacyjnego, źródło [2]
- (Rys. 4) Warstwowy model obliczeń kwantowych, źródło [3]
- (Rys. 5) Architektura komputera kwantowego, źródło [4]
- (Rys. 6) Problemy obliczeń kwantowych, źródło [4]
- (Rys. 7) Różne kody korekcyjne stosowane w komputerach kwantowych, źródło [4]
- (Rys. 8) Dostawcy technologii, źródło [4]
- (Rys. 9) Modele komputerów kwantowych, źródło [4]
- (Rys. 10) Środowiska do tworzenia oprogramowania dla komputerów kwantowych, źródło [4]
- (Rys. 11) 127 kubitowy procesor Eagle, źródło [6]
- (Rys. 12) Plan rozwoju procesorów kwantowych IBM [6]
- (Rys. 13) źródło [7]
- (Rys. 14) Zestawienie połączeń kwantowych różnego typu, źródło [4]
- (Rys. 15) Tworzenie splatania end-to-end za pomocą splatów między sąsiednimi węzłami, [4]
- (Rys. 16) Obwód teleportacji, źródło [4]
- (Rys. 17) Dowód na splatanie kubitów 1,2 i 3, źródło [20]
- (Rys. 18) Wykorzystanie technologii kwantowej w 6G, źródło [24]
- (Rys. 19) Przykładowy kod instrukcji kwantowych i stworzony obwód, źródło [20]
- (Rys. 20-28) Wykresy kołowe stanu, symulator obliczeń kwantowych, źródło [20]
- (Rys. 29) 3-kubitowe bramki odwracalne [29]
- (Rys. 30) Obwód testowy [20]
- (Rys. 31) Przygotowany stan początkowy rejestru [20]
- (Rys. 32) Stan rejestru po pojedynczej inkrementacji [20]
- (Rys. 33) Stan rejestru po dwóch inkrementacjach [20]
- (Rys. 34) Stan rejestru po dekrementacji [20]
- (Rys. 35) Obwód transformaty Fouriera [20]
- (Rys. 36) Przygotowany sygnał fali prostokątnej zakodowany w modułach i fazach [20]
- (Rys. 37) Widmo sygnału zakodowane w modułach i fazach [20]
- (Rys. 38) Kod QCEngine [20]
- (Rys. 39) Kod QCEngine, opisujący użycie odwrotnej transformaty Fouriera [20]
- (Rys. 40) Odwrotna transformata Fouriera - tworzenie przebiegu w czasie [20]
- (Rys. 41) Przygotowane widmo sygnału [20]
- (Rys. 42) Przebieg sygnału w czasie [20]
- (Rys. 43) Wykres z danych liczbowych (źródło: opracowanie własne)
- (Rys. 44) Obwód wykorzystujący wzmacnianie amplitudy [20]
- (Rys. 45) Oznaczylismy $|3\rangle$ [20]
- (Rys. 46) Stan $|3\rangle$ posiada maksymalną amplitudę [20]
- (Rys. 47) Obwód szacowania fazy globalnej dla operacji Hadamarda [20]
- (Rys. 48) Wnoszona faza własna w funkcji ilości uruchomień operacji H [20]
- (Rys. 49) Wyznaczona faza własna [20]
- (Wzór 50) Faktoryzacja Shora [27]
- (Rys. 51) Poszukiwanie okresowości dla $a=2$
- (Wzór 52) Liczby k_1 i k_2

- (Rys. 53) Poszukiwanie okresowości dla a=3
- (Rys. 54) Okno symulatora kwantowego w przeglądarce internetowej Firefox [20]
- (Rys. 55) Google Collab z programem w Qiskit do generowania liczb losowych [32]
- (Rys. 56) Rozkład zwracanego losowego bitu [32]
- (Rys. 57) Uzyskanie własnego obszaru roboczego Quantum [33]
- (Rys. 58) Konsola Google Cloud do zarządzania zasobami w chmurze [34]
- (Rys. 59) Dostępne zasoby IBM [35]
- (Rys. 60) Część pierwsza kodu eksperymentu [32]
- (Rys. 61) Obwód eksperymentu [30, 32]
- (Rys. 62) Dalsza część kodu eksperymentu [32]
- (Rys. 63) Przekształcony obwód dla QPU [30]
- (Rys. 64) Mapa użytego QPU [35]
- (Rys. 65) Kod wywołujący program na QPU i pobierający wyniki gdy będą dostępne [32]
- (Rys. 66) Otrzymane wyniki z QPU ibm_kyoto [35]
- (Rys. 67) Wyniki symulacji w Qiskit AerSimulator [30]
- (Rys. 68) Otrzymane wyniki z QPU ibm_sherbrooke [35]
- (Rys. 69) Obwód inkrementacji i dekrementacji [20]

Dodatek: Zapis matematyczny obliczeń kwantowych

Podstawowe bramki kwantowe i ich macierzowa reprezentacja

Operator	Gate(s)	Matrix
Pauli-X (X)		\oplus
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Źródło: Wikipedia

Zapis wektorowy wybranych stanów rejestru 2-kubitowego

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$|01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$|10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$|11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Operacja na kubicie wyraża się równaniem macierzowym

$$[\text{macierz operacji kwantowej}] \times [\text{stan kubit/-ów przed}] = [\text{stan kubit/-ów po}]$$

Użyjmy dla przykładu CNOT na stanie $|10\rangle$ (lewy kubit (1) to kubit kontrolny)

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

Widzimy że prawy kubit został zanegowany, gdyż rejestr przeszedł ze stanu $|10\rangle$ do $|11\rangle$

W ogólności, dla superpozycji wszystkich stanów rejestru, wyrażonej amplitudami prawdopodobieństw a b c d , będącymi liczbami zespolonymi spełniającymi warunek

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$$

możemy operacje CNOT opisać następująco

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a \\ b \\ d \\ c \end{bmatrix}$$

Porównując to do wykresu kołowego w symulatorze [20], odpowiada to zamianie miejscami kół stanu $|2\rangle$ i $|3\rangle$.

Ogólny zapis wektorowy stanu rejestru 2-kubitowego w bazie standardowej.

$$|\Psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

$$|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$$

Operacja kwantowa, przekształcająca stan początkowy rejestru Ψ_1 w stan Ψ_2 , opisana jest równaniem

$$U|\Psi_1\rangle = |\Psi_2\rangle$$

gdzie U to unitarna macierz opisująca operację kwantową (bramki kwantowe muszą być unitarne)

Zobaczmy, czy możemy łączyć bramki jedna po drugiej, opisując każdą bramkę osobną macierzą. Zróbmy bramkę NOT w wersji podanej w tabeli, oraz bramkę NOT złożoną z 2 bramek H i jednej PHASE 180.

Niech stan początkowy kubitu to

$$|\Psi_1\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$|\Psi_2\rangle = X|\Psi_1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$$

Widzimy że stan kubitu został zanegowany.

Teraz spróbujmy opisać macierzowo alternatywny zapis bramki NOT jako H, PHASE 180, H, lub krócej HZH. W tym miejscu trzeba dodać, że macierze opisujące działania będą używane w kolejności od prawej do lewej (co w tym konkretnym przypadku akurat jest niezauważalne, z uwagi na symetryczność ciągu operacji).

$$|\Psi_2\rangle = HZH|\Psi_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$|\Psi_2\rangle = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a+b \\ a-b \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a+b \\ b-a \end{bmatrix} = \frac{1}{2} \begin{bmatrix} a+b+b-a \\ a+b-b+a \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2b \\ 2a \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}$$

Otrzymaliśmy identyczny wynik.

Przedstawmy teraz dowód na odwracalność obliczeń kwantowych, złożonych z jednej lub więcej operacji kwantowych.

Niech dowolna operacja kwantowa będzie wyrażona unitarną macierzą U . Jeżeli obliczenia składają się z większej ilości operacji, to możemy wyznaczyć pojedynczą równoważną operację $U = FEDCBA$, gdzie A, B, C, D, E i F to kolejne operacje składowe.

Wiadomo z definicji, że każda macierz unitarna U posiada macierz odwrotną U^{-1} gdzie

$$UU^{-1} = I = U^{-1}U$$

Mogliśmy więc napisać

$$|\Psi\rangle = I|\Psi\rangle = UU^{-1}|\Psi\rangle = U^{-1}U|\Psi\rangle$$

gdzie I , to macierz identycznościowa (jednostkowa), nie zmieniająca stanu kubitu Ψ .

Z równania widać, że nie zmienia stanu kubitu także wykonanie kolejno operacji U^{-1} i U w dowolnej kolejności. To oznacza, że dla operacji kwantowej U , istnieje operacja kwantowa U^{-1} cofająca efekt tej pierwszej, co wynika też z definicji macierzy unitarnej.

Zapiszmy to jeszcze prościej. Wykonajmy obliczenia U , przekształcając stan kubitu.

$$|\Psi'\rangle = U|\Psi\rangle$$

gdzie $|\Psi'\rangle$ to stan końcowy po obliczeniach

pomnóżmy teraz obie strony równania przez U^{-1}

$$U^{-1}|\Psi'\rangle = U^{-1}U|\Psi\rangle$$

Widac, że prawa strona da się sprowadzić do samego $|\Psi\rangle$. Zapiszmy, odwracając strony.

$$|\Psi\rangle = U^{-1}|\Psi'\rangle$$

Otrzymaliśmy zatem 2 wzory, pozwalające dowolnie przekształcać stan kubitu między stanem $|\Psi\rangle$ oraz $|\Psi'\rangle$. Istnienie macierzy U^{-1} wynika z definicji macierzy unitarnej, a ten przykład pokazuje jak jej użyć do cofnięcia obliczeń ze stanu końcowego $|\Psi'\rangle$ do stanu początkowego $|\Psi\rangle$.

W przypadku, gdy U składa się z kilku operacji kwantowych, na przykład $U=FEDCBA$, to należy każdą składową operację kwantową odwrócić osobno, zaczynając od ostatniej, pamiętając o zapisie działań na macierzach od prawej do lewej.

$$U^{-1} = A^{-1}B^{-1}C^{-1}D^{-1}E^{-1}F^{-1}$$

Podsumowując, mamy 2 wzory:

wzór na wykonanie obliczeń złożonych z wielu operacji

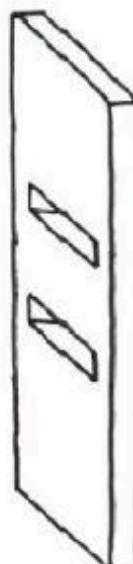
$$|\Psi' \rangle = FEDCBA|\Psi \rangle$$

oraz wzór na cofanie tych obliczeń

$$|\Psi \rangle = A^{-1}B^{-1}C^{-1}D^{-1}E^{-1}F^{-1}|\Psi' \rangle$$

W ten sposób pokazaliśmy, że pod względem złożoności obliczeniowej są one identyczne. Ma to ważne konsekwencje. Jeśli znamy interesujący algorytm, to możemy też łatwo otrzymać algorytm odwrotny. Dysponując otrzymanymi wynikami obliczeń, możemy odzyskać użyte oryginalne parametry wejściowe. Jak pokazano we wcześniejszych rozdziałach, istnieje na przykład kwantowa transformata Fouriera, oraz odwrotna kwantowa transformata Fouriera. Cofanie obliczeń usuwa też wszystkie efekty uboczne, takie jak niepożądane splątanie kubitów, dlatego jest często stosowanym wzorcem projektowym.

Koniec



kwejk.pl