

# Wprowadzenie do programowania kwantowego

Jarosław Miszczak  
IITiS PAN

Instytut Badań Systemowych  
Warszawa, 1 grudnia 2018

Cele wykładu

Wprowadzenie

Obliczenia kwantowe

Procedury kwantowe

Programowanie kwantowe

Manipulacja bramkami kwantowymi

Programowanie QRAM

Programowanie wysokopoziomowe



## Cele wykładu

- ▶ Poznanie zasad kwantowego przetwarzania informacji.



## Cele wykładu

- ▶ Poznanie zasad kwantowego przetwarzania informacji.
- ▶ Zrozumienie różnicy między obliczeniami klasycznymi a kwantowymi.



## Cele wykładu

- ▶ Poznanie zasad kwantowego przetwarzania informacji.
- ▶ Zrozumienie różnicy między obliczeniami klasycznymi a kwantowymi.
- ▶ Wprowadzenie różnych podejść do programowania kwantowego.



## Cele wykładu

- ▶ Poznanie zasad kwantowego przetwarzania informacji.
- ▶ Zrozumienie różnicy między obliczeniami klasycznymi a kwantowymi.
- ▶ Wprowadzenie różnych podejść do programowania kwantowego.
- ▶ Napisanie kodu, który można uruchomić na procesorze kwantowym.

# ➤ Wprowadzenie

## Obliczenia

Jednym z podstawowych (nieformalnych) założeń współczesnej informatyki jest hipoteza Churcha-Turinga, która określa rodzinę funkcji obliczalnych.

### Hipoteza Churcha-Turinga

Dowolne rozsądne obliczenia, mogą być wykonane przez maszynę Turinga.

*Every effectively calculable function is a computable function*

Hipoteza ta

- ▶ nie dotyczy kosztów wykonywanych obliczeń,
- ▶ zawiera ukryte założenie, że **rozsądne** obliczenia są wykonywane mechanicznie zgodnie z zasadami fizyki klasycznej.

# ➤ Wprowadzenie

## Obliczenia

- ▶ Wszystko wskazuje na to, iż mechanika klasyczna nie jest podstawową teorią opisującą układy fizyczne.
- ▶ W chwili obecnej uznaje się, iż taką teorią jest mechanika kwantowa.

## Hipoteza Churcha-Turinga-Deutscha

Każdy proces fizyczny może być symulowany przez uniwersalny komputer.



# ➤ Wprowadzenie

## Motywacja

### Prawo Moore'a (1965)

Moc obliczeniowa podwaja się co dwa lata dzięki zwiększeniu **gęstości tranzystorów**.

### Prawo Wirtha-Gatesa

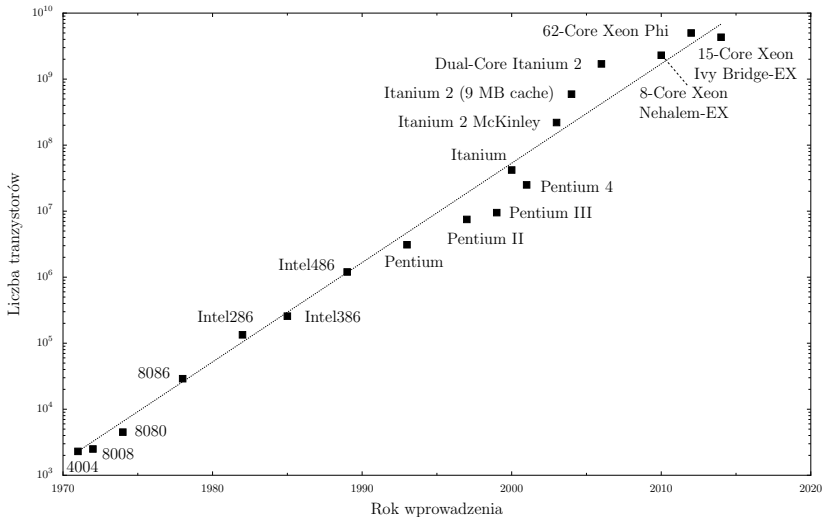
Wzrost mocy obliczeniowej jest kompensowany wzrostem **złożoności oprogramowania**.

*Software gets slower faster than hardware gets faster.*



# Wprowadzenie

## Motywacja



(Dane: [http://en.wikipedia.org/wiki/Transistor\\_count](http://en.wikipedia.org/wiki/Transistor_count))

# ✚ Wprowadzenie

## Motywacja



Osborne Executive (1982) oraz iPhone (2007)

(Źródło: [http://en.wikipedia.org/wiki/Moore's\\_law](http://en.wikipedia.org/wiki/Moore's_law))

# ✚ Wprowadzenie

## Motywacja



# ➤ Wprowadzenie

## Motywacja



Komandor podporucznik Data, 2366

(Źródło: <http://memory-alpha.wikia.com/wiki/Data>)

# ✦ Wprowadzenie

## Motywacja

- ▶ Zgodnie z informacjami w odcinku 9 drugiego sezonu serialu *Star Trek: The Next Generation* komandor porucznik Data dysponował mocą obliczeniową 60 teraflopów. W roku 1989, kiedy był emitowany ten odcinek, był zatem 60 000 szybszy od najszybszego superkomputera (wówczas Cray Y-MP pracujący z częstotliwością 167 MHz z wydajnością do 333 megaflopów).
- ▶ Obecnie (listopad 2018) najszybszy superkomputer świata (Summit o mocy 143 petaflopów) jest niemal 2500 razy szybszy niż Data.

# ✚ Wprowadzenie

## Motywacja

- ▶ Richard Feynmann, 1982 – symulacja układów fizycznych przez komputery klasyczne jest nieefektywna.

(Richard P. Feynman, *Simulating Physics with Computers*, International Journal of Theoretical Physics, Vol 21, Nos. 6/7, 1982)

# ➤ Wprowadzenie

## Motywacja

- ▶ Czy maszyna Turinga może wykonać symulacje układów kwantowych efektywnie?
- ▶ Czy stan układu kwantowego może być wykorzystany do przechowywania (i przesyłania) dowolnie dużej ilości informacji?



# ➤ Wprowadzenie

## Motywacja

- ▶ W mechanice kwantowej pomiar powoduje, że system ulega zniszczeniu. Czy może być to wykorzystane do zabezpieczenia danych?

# ✦ Wprowadzenie

## Motywacja

- ▶ S.J. Wiesner (1970): niepodrabialne pieniądze – podstawa zakazu klonowania i kryptografii kwantowej. S.J. Wiesner, *Conjugate Coding*, SIGACT News, Vol. 15, pp. 78-88 (1983).
- ▶ A. Ekert (1991): kryptografia kwantowa (Artur Ekert, *Quantum cryptography based on Bell's theorem*. Physical Review Letters, 67: 661–663.)

# Obliczenia kwantowe

# ➤ Obliczenia kwantowe

## Zasada liniowości

Pierwszą z zasad przy wprowadzaniu opisu układów w języku mechaniki kwantowej jest **zasada liniowości**.

### Liniowość

Dwa stany układu kwantowego można dodać i otrzymany w ten sposób obiekt (czyli kombinacja liniowa) jest również poprawnym stanem układu.

Kombinacja liniowe stanów nazywane są **superpozycjami stanów**.

# ➤ Obliczenia kwantowe

## Zasada liniowości

- ▶ Superpozycja **nie oznacza**, że układ jest w każdym ze stanów składowych.
- ▶ Po wykonaniu pomiaru, układ będzie w jednym ze stanów które chcemy zmierzyć.
- ▶ Superpozycja może mieć współczynniki zespolone. Prawdopodobieństwa wyników pomiarów są uzyskiwane jako kwadraty modułów tych współczynników.

## ➤ Obliczenia kwantowe

### Zasada liniowości

Aby wykonywać obliczenia ( $\equiv$  operować na danych), konieczne jest wprowadzenie stosownej reprezentacji danych.

### Zasada “zerowego” kwantowania

Przyjmijmy, że  $0 \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  oraz  $1 \mapsto \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .

Ponieważ  $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$  i  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  to wektory, więc kwantowy bit może być w stanie

$$x_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + x_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad x_0, x_1 \in \mathbb{C},$$

np.  $\frac{1}{2} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{i}{2} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .

## ➤ Obliczenia kwantowe

### Notacja Diraca

Notacja Diraca polega na skrótowym zapisaniu reprezentacji danych.

#### Notacja Diraca

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Ponieważ  $|0\rangle$  i  $|1\rangle$  to wektory, więc kwantowy bit może być w stanie

$$|a\rangle = x_0|0\rangle + x_1|1\rangle, \quad x_0, x_1 \in \mathbb{C},$$

np.  $\frac{1}{2}|0\rangle + \frac{i}{2}|1\rangle$ .

# ➤ Obliczenia kwantowe

## Obwody kwantowe

### Odwracalność

Drugą zasadą jest **odwracalność operacji** wykonywanych na stanach (czyli ewolucji układu).

Z fizycznego punktu widzenia wynika ona z zasady zachowania energii – formalizm wektorów stanu zakłada, że nie dochodzi do wpływu informacji z układu (lub równoważnie, że układ jest całkowicie odizolowany).

### Bramki kwantowe

Przekłada się to na opis ewolucji układu za pomocą **macierzy unitarnych**, nazywanych **bramkami kwantowymi**.



# ➤ Obliczenia kwantowe

## Obwody kwantowe

- ▶ Charles Bennett, 1973 – uniwersalna maszyna Turinga może być zrealizowana w sposób odwracalny.  
(C. H. Bennett, *Logical reversibility of computation*, IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532 (1973).)

# ➤ Obliczenia kwantowe

## Obwody kwantowe

Przykłady bramek uniwersalnych dla klasycznych obliczeń

- ▶ Bramka Fredkina (kontrolowany SWAP)
- ▶ Bramka Toffoliego (podwójnie kontrolowany NOT)

Bramki te są **trójbitowe**.

## CNOT

W przypadku obliczeń kwantowych do konstrukcji zbioru zupełnego wystarczy dwukubitowa bramka CNOT.

## ➤ Obliczenia kwantowe

### Obwody kwantowe

#### Zasada “zerowego” kwantowania operacji

Dla zadanej operacji, kwantowym odpowiednikiem jest takie przekształcenie wektorów, które działa odpowiednio na bazie  $\{|0\rangle, |1\rangle\}$ .

Przykładowo dla operacji negacji kwantowy odpowiednik to

$$Not = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Możliwe jest natomiast wprowadzenie operacji, które nie mają odpowiedników klasycznych, np. operacja Hadamarda

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

która wprowadza superpozycję 0 i 1.

## ➤ Obliczenia kwantowe

### Obwody kwantowe

Kolejny nieklasyczny przykład to **pierwiastek kwadratowy z negacji**  $\sqrt{Not}$ , który spełnia własność

$$\sqrt{Not}\sqrt{Not}|x\rangle = Not|x\rangle$$

dla dowolnego wejścia  $|x\rangle$ .

Oczywiście nic nie stoi na przeszkodzie, żeby zdefiniować pierwiastek dowolnego stopnia z dowolnej macierzy.

# ❖ Obliczenia kwantowe

## Obwody kwantowe

Co warto zapamiętać?

- ▶ W modelu kwantowym zarówno przestrzeń dozwolonych stanów, jak i struktura operacji, są **bogatsze** niż w przypadku modelu klasycznego.
- ▶ Sugeruje to, iż potencjalnie mogą służyć one do wykonywania obliczeń w sposób **bardziej wydajny**.

# Procedury kwantowe

# ➤ Procedury kwantowe

## Algorytm Grovera

- ▶ Algorytm Grovera pozwala na wyszukanie elementu w zbiorze.
- ▶ Wykorzystuje on superpozycję do zapisania informacji o przeszukiwanym zbiorze.
- ▶ Algorytm Grovera jest probabilistyczny – nie zawsze dostajemy dobry wyniki!

## ➤ Procedury kwantowe

### Algorytm Grovera

- ▶ Dysponujemy zbiorem  $N$  elementów, z których jeden,  $\omega$ , spełnia pewien warunek (np. jest liczbą pierwszą).
- ▶ Do konstrukcji algorytmu kwantowego potrzebujemy operacji unitarnej, która odpowiada funkcji sprawdzającej ten warunek,

$$U_{\omega}|x\rangle = -|x\rangle$$

dla  $x = \omega$  spełniającego nasz warunek oraz

$$U_{\omega}|x\rangle = |x\rangle$$

dla  $x \neq \omega$ .



# ➤ Procedury kwantowe

## Algorytm Grovera

### Kroki algorytmu

- ▶ Przygotuj superpozycję  $|s\rangle$  wszystkich przeszukiwanych stanów.
- ▶ Zastosuj operator  $U_\omega$ .
- ▶ Zastosuj operator  $U_s = 2|s\rangle\langle s| - \mathbb{I}$
- ▶ Powtórz dwa poprzednie kroki około  $\frac{\pi}{4}\sqrt{N}$  razy.

# ➤ Procedury kwantowe

## Algorytm Grovera

Zobaczmy jak to działa dla jednej iteracji...

# ➤ Procedury kwantowe

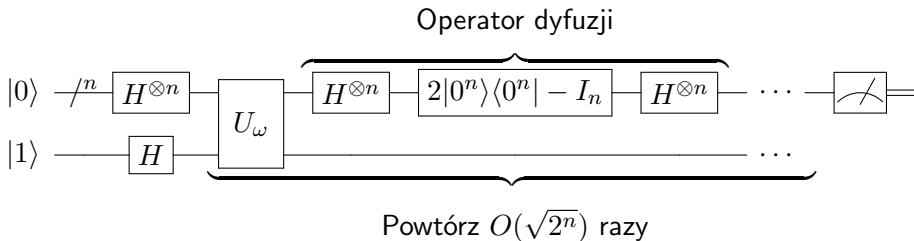
## Algorytm Grovera

- ▶ Proporcja (a właściwie amplituda) stanu  $|\omega\rangle$  wzrasta po pierwszej iteracji do  $\frac{2}{\sqrt{N}}$ .
- ▶ Dla  $N = 4$  wystarczy jedna iteracja.

# ➤ Procedury kwantowe

## Algorytm Grovera

Obwód dla  $N$  elementów zapisanych na  $n$  qubitach.



# ➤ Procedury kwantowe

## Algorytm Grovera

- ▶ Algorytm Grovera można uogólnić dla kilku zaznaczonych elementów.
- ▶ Ponieważ prawdopodobieństwo uzyskania dobrej odpowiedzi rośnie z każdym krokiem, można rozważyć kiedy warto przerwać algorytm.

# ➤ Procedury kwantowe

## Teleportacja

- ▶ Protokoły kwantowe są budowane do *kodowania* i *przesyłania* informacji.
- ▶ Oprócz superpozycji stanów, ważna jest tu również możliwość operowania na układach złożonych.

# ➤ Procedury kwantowe

## Teleportacja



Transporter działający na statku klasy Galaxy, 2364

## ➤ Procedury kwantowe

### Teleportacja

- ▶ Teleportacja pozwala na przesłanie stanu kwantowego poprzez wysłanie dwóch bitów.
- ▶ Układ, którego stan jest teleportowany, jest niszczone w wyniku działania procedury teleportacji.

### Uwaga!

Teleportacja służy do przesyłania **stanu** układu!



# ➤ Procedury kwantowe

## Teleportacja

- ▶ Teleportacja polega przesłaniu stanu między dwoma punktami – lub osobami, np. Alicją i Bobem.
- ▶ Zakładamy, że Alicja chce przekazać Bobowi (nieznany) stan  $|\psi\rangle_c = a|0\rangle + b|1\rangle$ .
- ▶ Protokół wymaga, żeby współdzielili jeden ze stanów *splątanych*:

$$|\Phi^+\rangle_{AB} = 1/\sqrt{2}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B),$$

$$|\Phi^-\rangle_{AB} = 1/\sqrt{2}(|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B),$$

$$|\Psi^+\rangle_{AB} = 1/\sqrt{2}(|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B),$$

$$|\Psi^-\rangle_{AB} = 1/\sqrt{2}(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B).$$

## ➤ Procedury kwantowe

### Teleportacja

Przyjmijmy, że Alicja i Bob współdzielą stan

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Zatem stan całego układu to

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes (\alpha|0\rangle_B + \beta|1\rangle_B)$$

## ➤ Procedury kwantowe

### Teleportacja

Stan całości można zapisać jako

$$\begin{aligned} |\Phi^+\rangle_{AB} \otimes |\psi\rangle_C = & \frac{1}{2} |\Phi^+\rangle_{AC} \otimes (\alpha|0\rangle_B + \beta|1\rangle_B) \\ & + \frac{1}{2} |\Phi^-\rangle_{AC} \otimes (\alpha|0\rangle_B - \beta|1\rangle_B) \\ & + \frac{1}{2} |\Psi^+\rangle_{AC} \otimes (\beta|0\rangle_B + \alpha|1\rangle_B) \\ & + \frac{1}{2} |\Psi^-\rangle_{AC} \otimes (\beta|0\rangle_B - \alpha|1\rangle_B) \end{aligned}$$

# ➤ Procedury kwantowe

## Teleportacja

- ▶ Do zakończenia protokołu Alicja musi wysłać Bobowi informację o stanie jej podukładu.
- ▶ Możliwych stanów jest  $2^2$ , czyli potrzebne są dwa bity.
- ▶ W zależności od otrzymanych danych, Bob wykonuje jedną z operacji.

- ▶  $|\Phi^-\rangle \mapsto \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

- ▶  $|\Psi^-\rangle \mapsto \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

- ▶  $|\Psi^+\rangle \mapsto \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

## ➤ Procedury kwantowe

### Co dalej?

- ▶ Algorytm Grovera (i inne algorytmy kwantowe) są tylko *podprocuderami* do tworzenia użytecznych algorytmów bazujących na mechanice kwantowej.
- ▶ Podobnie jest z teleportacją w zakresie komunikacji kwantowej – może być ona podstawą do zabezpieczania transmisji danych.

# Programowanie kwantowe



# Programowanie kwantowe

*Programowanie kwantowe to proces, który prowadzi od sformułowania problemu obliczeniowego do kodu wykonywalnego na komputerze kwantowym*

## ➤ Programowanie kwantowe

- ▶ *Jedynym sposobem nauki nowego języka programowania kwantowego jest pisanie programów.*





## Programowanie kwantowe

- ▶ *Jedynym sposobem nauki nowego języka programowania kwantowego jest pisanie programów.*
- ▶ *Proces przygotowania programów na komputer kwantowy jest szczególnie atrakcyjny, ponieważ nie tylko może być opłacalny ekonomicznie i naukowo, ale może również być estetycznym doświadczeniem, podobnie jak komponowanie poezji lub muzyki.*



## Programowanie kwantowe

- ▶ *Jedynym sposobem nauki nowego języka programowania kwantowego jest pisanie programów.*
- ▶ *Proces przygotowania programów na komputer kwantowy jest szczególnie atrakcyjny, ponieważ nie tylko może być opłacalny ekonomicznie i naukowo, ale może również być estetycznym doświadczeniem, podobnie jak komponowanie poezji lub muzyki.*
- ▶ *Tylko współczesny komputer kwantowy uczynił programowanie kwantowe zarówno trudnym, jak i istotnym.*

## Programowanie kwantowe

- ▶ *Jedynym sposobem nauki nowego języka programowania kwantowego jest pisanie programów.*
- ▶ *Proces przygotowania programów na komputer kwantowy jest szczególnie atrakcyjny, ponieważ nie tylko może być opłacalny ekonomicznie i naukowo, ale może również być estetycznym doświadczeniem, podobnie jak komponowanie poezji lub muzyki.*
- ▶ *Tylko współczesny komputer kwantowy uczynił programowanie kwantowe zarówno trudnym, jak i istotnym.*
- ▶ *Projektowanie kwantowego języka programowania jest jak spacer po parku. Parku Jurajskim.*

# ➤ Programowanie kwantowe

Co daje nam programowanie kwantowe?

- ▶ Możliwość wykorzystania istniejących komputerów kwantowych.

# ➤ Programowanie kwantowe

Co daje nam programowanie kwantowe?

- ▶ Możliwość wykorzystania istniejących komputerów kwantowych.
- ▶ Zabawa z mechaniką kwantową.

# ✚ Programowanie kwantowe

Co daje nam programowanie kwantowe?

- ▶ Możliwość wykorzystania istniejących komputerów kwantowych.
- ▶ Zabawa z mechaniką kwantową.
- ▶ Nowe elementy nie istniejące w zwykłych językach programowania...

# ✚ Programowanie kwantowe

Co daje nam programowanie kwantowe?

- ▶ Możliwość wykorzystania istniejących komputerów kwantowych.
- ▶ Zabawa z mechaniką kwantową.
- ▶ Nowe elementy nie istniejące w zwykłych językach programowania...
- ▶ ...które mogą być wykorzystane do opisu świata fizycznego.

# ✚ Programowanie kwantowe

## Różne podejścia do programowania kwantowego

- ▶ Poziom 0: Bezpośrednia manipulacja bramkami kwantowymi.



# ✚ Programowanie kwantowe

## Różne podejścia do programowania kwantowego

- ▶ Poziom 0: Bezpośrednia manipulacja bramkami kwantowymi.
- ▶ Poziom 1: Model QRAM.

# ✚ Programowanie kwantowe

## Różne podejścia do programowania kwantowego

- ▶ Poziom 0: Bezpośrednia manipulacja bramkami kwantowymi.
- ▶ Poziom 1: Model QRAM.
- ▶ Poziom 2: Programowanie wysokopoziomowe.

# ✦ Programowanie kwantowe

## Różne podejścia do programowania kwantowego

- ▶ Poziom 0: Bezpośrednia manipulacja bramkami kwantowymi.
- ▶ Poziom 1: Model QRAM.
- ▶ Poziom 2: Programowanie wysokopoziomowe.
- ▶ Poziom 2': Programowanie wysokospecjalizowane.

# Manipulacja bramkami kwantowymi

# ➤ Manipulacja bramkami kwantowymi

## Level 0

Bezpośrednie operowanie na obwodach.

- ▶ Manipulacja obwodami w środowisku graficznym.
- ▶ Operacje na wektorach i macierzach.

# Manipulacja bramkami kwantowymi

## Manipulacja obwodami w środowisku graficznym

The screenshot displays the IBM Quantum Experience graphical circuit editor. At the top, a dark blue header shows two active user sessions: 'IBM Q 5.1 [ibmqx4]' and 'IBM Q 5 [ibmqx2]'. Below this, the main workspace is titled 'FourQubitsTest' with an 'Add a description' button. A toolbar includes 'New', 'Save', and 'Save as' buttons. A status bar indicates the backend is 'ibmqx4', with 'My Units: 18' and 'Experiment Units: 3'. The circuit editor shows five qubits (q[0] to q[4]) and a classical register 'c'. The circuit includes Hadamard (H) gates on q[1] and q[3], CNOT gates with targets on q[0] and q[2], and measurement operations on q[0] and q[2]. A right-hand panel contains a library of quantum gates (Z, H, S, S†, T, T†, CNOT) and barrier operations. A progress bar at the bottom indicates the simulation status.

<https://quantumexperience.ng.bluemix.net/qx/editor>

# Manipulacja bramkami kwantowymi

## Manipulacja obwodami w środowisku graficznym

QUIDE - Quantum Integrated Development Environment

Build Circuit Run In Console

Teleportation.cs

```
11 public static void Main() {
12     QuantumComputer comp = QuantumComputer.GetInstance();
13
14     // nontrivial state to be teleported
15     var x_initStates = new Dictionary<ulong, Complex>() {
16         {0, new complex(-0.6, 0)},
17         {1, new complex(0, 0.8)}
18     };
19     Register x = comp.NewRegister(x_initStates, 1);
```

Circuit Designer

To Code Clear Ungroup Group

Select Composite Gate...

$|x_0\rangle = |*\rangle$   
 $|y_0\rangle = |0\rangle$   
 $|z_0\rangle = |0\rangle$

Output

Output from register: root

States Visibility

☐ Show All ☒ Show Only Non-Zero

Probability Scale

☒ Absolute ☐ Relative

Value	Qubits	Probability	Amplitude
$ 0\rangle$	$ 000\rangle$	P = 0.18	-0.42 + 0.00i
$ 3\rangle$	$ 011\rangle$	P = 0.18	-0.42 + 0.00i
$ 5\rangle$	$ 101\rangle$	P = 0.32	0.00 + 0.57i
$ 6\rangle$	$ 110\rangle$	P = 0.32	0.00 + 0.57i

Properties

Bloch sphere visualization

<http://www.quide.eu/> and <https://bitbucket.org/quide/>

# Manipulacja bramkami kwantowymi

## Manipulacja obwodami w środowisku graficznym

Quantum Inspire

Quick Guide [FAQ](#) [Exit](#)

Editor Results Teleportation Saved [Run](#)

```
1 version 1.0
2
3 qubits 3
4 error_model depolarizing_channel, 0.1 # error model used in simulation
5
6 # start writing your code here
7
8 # initialize the state
9 prep_z q[0]
10 T q[0]
11
12 # initialize shared registers
13 prep_z q[1:2]
14
15 # entangled state preparation
16 h q[1]
17 CNOT q[1], q[2]
18
```

q[0] -  $|0\rangle$  -  $T$  -  $H$  -  $Z$

q[1] -  $|0\rangle$  -  $H$  -  $\bullet$  -  $\bullet$  -  $X$  -  $Y$

q[2] -  $|0\rangle$  -  $\bullet$  -  $\bullet$  -  $X$  -  $Y$

Pauli-X gate  
Pauli-Z gate  
Hadamard gate  
Identity gate  
90 degree rotations  
Rx Gate  
Ry Gate  
Rz Gate

<https://www.quantum-inspire.com/>



## ➤ Manipulacja bramkami kwantowymi

### Manipulacja obwodami w środowisku graficznym

- ▶ Bezpośrednia manipulacja bramkami kwantowymi, pomiar.

## ➤ Manipulacja bramkami kwantowymi

### Manipulacja obwodami w środowisku graficznym

- ▶ Bezpośrednia manipulacja bramkami kwantowymi, pomiar.
- ▶ Integracja z tekstowym formatem opisu obwodów: QASM (IBM, QuTech) or C# (QulIDE).

## ➤ Manipulacja bramkami kwantowymi

### Manipulacja obwodami w środowisku graficznym

- ▶ Bezpośrednia manipulacja bramkami kwantowymi, pomiar.
- ▶ Integracja z tekstowym formatem opisu obwodów: QASM (IBM, QuTech) or C# (QulIDE).
- ▶ Przydatny do testowania małych obwodów.

## ➤ Manipulacja bramkami kwantowymi

### Manipulacja obwodami w środowisku graficznym

- ▶ Bezpośrednia manipulacja bramkami kwantowymi, pomiar.
- ▶ Integracja z tekstowym formatem opisu obwodów: QASM (IBM, QuTech) or C# (QulIDE).
- ▶ Przydatny do testowania małych obwodów.
- ▶ Mniej dla prawdziwych algorytmów.

## ❖ Manipulacja bramkami kwantowymi

### Manipulacja obwodami w środowisku graficznym

- ▶ Bezpośrednia manipulacja bramkami kwantowymi, pomiar.
- ▶ Integracja z tekstowym formatem opisu obwodów: QASM (IBM, QuTech) or C# (QulIDE).
- ▶ Przydatny do testowania małych obwodów.
- ▶ Mniej dla prawdziwych algorytmów.
- ▶ Musisz żyć z ograniczeniami topologii połączeń (IBM Q).

## ➤ Manipulacja bramkami kwantowymi

### Manipulacja obwodami w środowisku graficznym

- ▶ Bezpośrednia manipulacja bramkami kwantowymi, pomiar.
- ▶ Integracja z tekstowym formatem opisu obwodów: QASM (IBM, QuTech) or C# (QulIDE).
- ▶ Przydatny do testowania małych obwodów.
- ▶ Mniej dla prawdziwych algorytmów.
- ▶ Musisz żyć z ograniczeniami topologii połączeń (IBM Q).
- ▶ Ale możesz użyć prawdziwego komputera kwantowego!

# ➤ Manipulacja bramkami kwantowymi

## Manipulacja obwodami w środowisku graficznym

### Zalety

- ▶ Bezpośrednie przełożenie na obwody kwantowe.

### Wady

- ▶ Czasochłonne wprowadzanie kodu.
- ▶ Brak możliwości wykonywania wsadowego oraz generowania kodu.



# Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

Jak już wiecie

- ▶ stany kwantowe to wektory (przynajmniej chcielibyśmy żeby nimi były)





# Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

Jak już wiecie

- ▶ stany kwantowe to wektory (przynajmniej chcielibyśmy żeby nimi były)
- ▶ ...bramki kwantowe to po prostu macierze (co najwyżej  $4 \times 4$ , i nie interesuje nas żadna dekoherencja)...



# Manipulacja bramkami kwantowymi

## Alternatywne podejście: manipulacja macierzami i wektorami

Jak już wiecie

- ▶ stany kwantowe to wektory (przynajmniej chcielibyśmy żeby nimi były)
- ▶ ...bramki kwantowe to po prostu macierze (co najwyżej  $4 \times 4$ , i nie interesuje nas żadna dekoherencja)...
- ▶ ...no i tylko ostatni krok w obliczeniach jest dziwny.



# Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

Jak już wiecie

- ▶ stany kwantowe to wektory (przynajmniej chcielibyśmy żeby nimi były)
- ▶ ...bramki kwantowe to po prostu macierze (co najwyżej  $4 \times 4$ , i nie interesuje nas żadna dekoherencja)...
- ▶ ...no i tylko ostatni krok w obliczeniach jest dziwny.
- ▶ *Prawdziwi Programiści piszą programy kwantowe w FORTRANIE.*



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

Na przykład, w języku Julia...



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

Na przykład, w języku Julia...

Przykłady w WolfLang są dostępne na GitHub-ie



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!
- ▶ Ponieważ...

## ➤ Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!
- ▶ Ponieważ...
  - ▶ ...znasz już język,





## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!
- ▶ Ponieważ...
  - ▶ ...znasz już język,
  - ▶ ...bardzo łatwo jest wdrożyć klasyczną kontrolę,



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!
- ▶ Ponieważ...
  - ▶ ...znasz już język,
  - ▶ ...bardzo łatwo jest wdrożyć klasyczną kontrolę,
  - ▶ ...stosunkowo łatwo jest uwzględnić efekty dekoherencji.



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!
- ▶ Ponieważ...
  - ▶ ...znasz już język,
  - ▶ ...bardzo łatwo jest wdrożyć klasyczną kontrolę,
  - ▶ ...stosunkowo łatwo jest uwzględnić efekty dekoherencji.
- ▶ Brakujący element: zarządzanie pamięcią.



## Manipulacja bramkami kwantowymi

Alternatywne podejście: manipulacja macierzami i wektorami

- ▶ Właściwie to jest prawie tak dobre, jak to tylko możliwe!
- ▶ Ponieważ...
  - ▶ ...znasz już język,
  - ▶ ...bardzo łatwo jest wdrożyć klasyczną kontrolę,
  - ▶ ...stosunkowo łatwo jest uwzględnić efekty dekoherencji.
- ▶ Brakujący element: zarządzanie pamięcią.
- ▶ W większości przypadków nie potrzebujesz całej mocy / bibliotek / itd. dostarczanych z językiem-hostem.

# ➤ Manipulacja bramkami kwantowymi

## Co dalej?

- ▶ IBM Q Experience:  
<https://quantumexperience.ng.bluemix.net/qx/editor>

# ➤ Manipulacja bramkami kwantowymi

## Co dalej?

- ▶ IBM Q Experience:  
<https://quantumexperience.ng.bluemix.net/qx/editor>
- ▶ Pakiety/biblioteki do manipulacji na macierzach:

# ✚ Manipulacja bramkami kwantowymi

## Co dalej?

- ▶ IBM Q Experience:  
<https://quantumexperience.ng.bluemix.net/qx/editor>
- ▶ Pakiety/biblioteki do manipulacji na macierzach:
  - ▶ quantum-octave (Octave/Matlab):  
<https://github.com/ZKSI/quantum-octave>

# ➤ Manipulacja bramkami kwantowymi

## Co dalej?

- ▶ IBM Q Experience:  
<https://quantumexperience.ng.bluemix.net/qx/editor>
- ▶ Pakiety/biblioteki do manipulacji na macierzach:
  - ▶ quantum-octave (Octave/Matlab):  
<https://github.com/ZKSI/quantum-octave>
  - ▶ QuTiP (Python library): <http://qutip.org/>



# ➤ Manipulacja bramkami kwantowymi

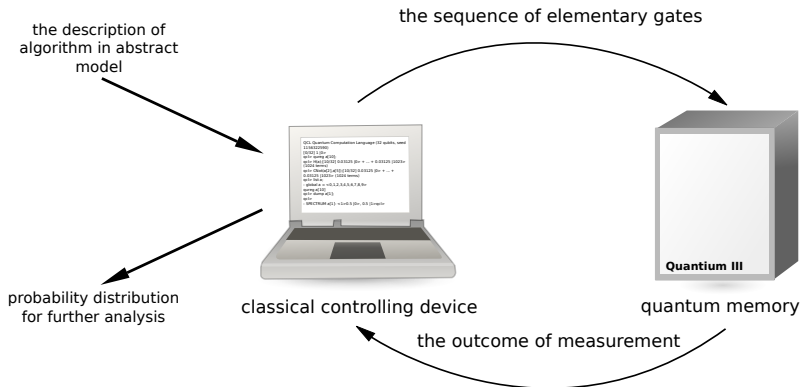
## Co dalej?

- ▶ IBM Q Experience:  
<https://quantumexperience.ng.bluemix.net/qx/editor>
- ▶ Pakiety/biblioteki do manipulacji na macierzach:
  - ▶ quantum-octave (Octave/Matlab):  
<https://github.com/ZKSI/quantum-octave>
  - ▶ QuTiP (Python library): <http://qutip.org/>
- ▶ Więcej na Quantiki  
<https://quantiki.org/wiki/list-qc-simulators>

# Programowanie QRAM

## Co to jest QRAM?

QRAM  $\equiv$  Quantum Random Access Machine



# ➤ Programowanie QRAM

## Co to jest QRAM?

- ▶ Procesor kwantowy jest traktowany jako wyspecjalizowany co-procesor.
- ▶ Wykonywane są na nim tylko te instrukcje, które wykorzystują kwantowość procesu.
- ▶ W tym modelu mieści się również wykorzystanie specjalistycznych procesorów kwantowych, które nie są uniwersalne (np. procesor D-Wave Systems służący do optymalizacji)

# ✚ Programowanie QRAM

## Zalety modelu QRAM

- ▶ abstrakcja danych

# ✚ Programowanie QRAM

## Zalety modelu QRAM

- ▶ abstrakcja danych  $\equiv$  alokacja pamięci kwantowej

# ✦ Programowanie QRAM

## Zalety modelu QRAM

- ▶ abstrakcja danych  $\equiv$  alokacja pamięci kwantowej
- ▶ złożone operacje kwantowe

# ✦ Programowanie QRAM

## Zalety modelu QRAM

- ▶ abstrakcja danych  $\equiv$  alokacja pamięci kwantowej
- ▶ złożone operacje kwantowe  $\equiv$  funkcje opakowujące sekwencje bramek kwantowych lub podstawowych procedur kwantowych



# ✦ Programowanie QRAM

## Zalety modelu QRAM

- ▶ abstrakcja danych  $\equiv$  alokacja pamięci kwantowej
- ▶ złożone operacje kwantowe  $\equiv$  funkcje opakowujące sekwencje bramek kwantowych lub podstawowych procedur kwantowych
- ▶ klasyczna kontrola operacji kwantowych

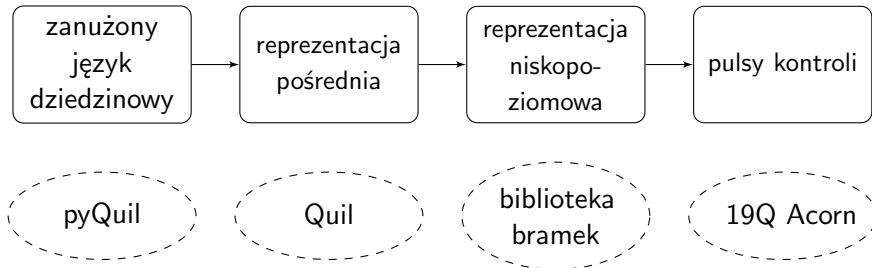
# ✦ Programowanie QRAM

## Zalety modelu QRAM

- ▶ abstrakcja danych  $\equiv$  alokacja pamięci kwantowej
- ▶ złożone operacje kwantowe  $\equiv$  funkcje opakowujące sekwencje bramek kwantowych lub podstawowych procedur kwantowych
- ▶ klasyczna kontrola operacji kwantowych  $\equiv$  pętle, instrukcje warunkowe etc. zmieszane z kodem kwantowym

# ✦ Programowanie QRAM

## Architektura oprogramowania



# ✦ Programowanie QRAM

## Kwantowe oprogramowanie pośrednie

Obecnie standardowe podejście dostarczania usług obliczeń kwantowych

- ▶ język dziedzinowy osadzony w języku uniwersalnym → C/Python/Wolfram/Haskell z biblioteką funkcji

# ✦ Programowanie QRAM

## Kwantowe oprogramowanie pośrednie

Obecnie standardowe podejście dostarczania usług obliczeń kwantowych

- ▶ język dziedzinowy osadzony w języku uniwersalnym → C/Python/Wolfram/Haskell z biblioteką funkcji
- ▶ abstrakcja danych → alokacja rejestrów klasycznych i kwantowych opertych na  $ku(b|d)itach$

# ✦ Programowanie QRAM

## Kwantowe oprogramowanie pośrednie

Obecnie standardowe podejście dostarczania usług obliczeń kwantowych

- ▶ język dziedzinowy osadzony w języku uniwersalnym → C/Python/Wolfram/Haskell z biblioteką funkcji
- ▶ abstrakcja danych → alokacja rejestrów klasycznych i kwantowych opartych na  $ku(b|d)$ itach
- ▶ funkcje kwantowe → niestandardowe bramki zdefiniowane przez macierze lub złożone instrukcje

# ✦ Programowanie QRAM

## Kwantowe oprogramowanie pośrednie

Obecnie standardowe podejście dostarczania usług obliczeń kwantowych

- ▶ język dziedzinowy osadzony w języku uniwersalnym → C/Python/Wolfram/Haskell z biblioteką funkcji
- ▶ abstrakcja danych → alokacja rejestrów klasycznych i kwantowych opartych na  $ku(b|d)$ itach
- ▶ funkcje kwantowe → niestandardowe bramki zdefiniowane przez macierze lub złożone instrukcje
- ▶ klasyczna kontrola pamięci kwantowej → poprzez wykorzystanie języka-hosta.

# ✦ Programowanie QRAM

## ...i jego zalety...

- ▶ łatwy do nauczenia i używania,



# ➤ Programowanie QRAM

...i jego zalety...

- ▶ łatwy do nauczenia i używania,
- ▶ auto-magiczne zarządzanie pamięcią kwantową (ponieważ kubity są po prostu tablicami),

# ✚ Programowanie QRAM

...i jego zalety...

- ▶ łatwy do nauczenia i używania,
- ▶ auto-magiczne zarządzanie pamięcią kwantową (ponieważ kubity są po prostu tablicami),
- ▶ integracja z klasyczną maszyną

# ➤ Programowanie QRAM

...i jego wady

- ▶ bardzo podobny do kodu niskiego poziomu (ale można zbudować bibliotekę bramek)

# ✚ Programowanie QRAM

## ...i jego wady

- ▶ bardzo podobny do kodu niskiego poziomu (ale można zbudować bibliotekę bramek)
- ▶ brak wyrazistości  $\equiv$  działasz tylko na podstawowych składnikach

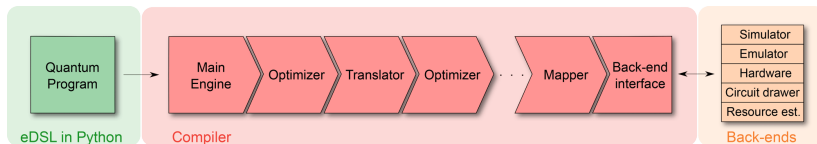
# ✦ Programowanie QRAM

## Przykład 1: ProjectQ

- ▶ biblioteka Pythona opracowana przez ETH (<https://projectq.ch/>)
- ▶ oferuje różne platformy docelowe
  - ▶ sprzęt (IBM Q Experience)
  - ▶ symulator (`use_hardware=False`)
  - ▶ licznik zasobów (resource counter)
  - ▶ reprezentacja graficzna obwodu

# ✚ Programowanie QRAM

## Przykład 1: ProjectQ



# ✚ Programowanie QRAM

## Przykład 1: ProjectQ

### Zalety

- ▶ Naturalna (dla fizyków) składnia do wykonywania bramek kwantowych.
- ▶ Instrukcje meta dla kwantowo kontrolowanych operacji kwantowych i obsługa odliczania.
- ▶ Bardzo łatwa konstrukcja kontrolowanych operacji.

# ➦ Programowanie QRAM

## Przykład 1: ProjectQ

Kilka przykładów...

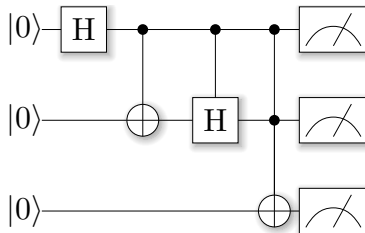


# ✚ Programowanie QRAM

## Przykład 1: ProjectQ

### Metainstrukcja Control

Wykonanie kodu opiera się na stanie rejestru kwantowego.

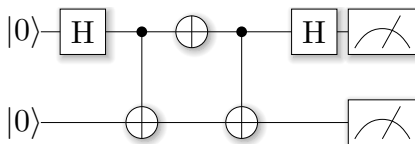


# ➤ Programowanie QRAM

## Przykład 1: ProjectQ

### Metainstrukcja Dagger

Odwrotne wykonanie kodu (odwracalność obliczeń)



# ✦ Programowanie QRAM

## Przykład 2: Rigetti Forest i pyQuil

- ▶ Quil  $\equiv$  assembler kwantowy podobny do QASM.

# ✦ Programowanie QRAM

## Przykład 2: Rigetti Forest i pyQuil

- ▶ Quil  $\equiv$  assembler kwantowy podobny do QASM.
- ▶ pyQuil  $\equiv$  biblioteka języka Python do manipulacji programami w Quil-u.

## ✦ Programowanie QRAM

### Przykład 2: Rigetti Forest i pyQuil

- ▶ Quil  $\equiv$  asembler kwantowy podobny do QASM.
- ▶ pyQuil  $\equiv$  biblioteka języka Python do manipulacji programami w Quil-u.
- ▶ Dostęp do maszyny kwantowej 19Q-Acorn firmy Rigetti!

# ✦ Programowanie wysokopoziomowe

## Języki dziedzinowe

### Poziom 2

Język specyficzny dla domeny (domenowy) z abstrakcją danych i funkcji.

- ▶ QCL (<http://tph.tuwien.ac.at/~oemer/qcl.html>)
- ▶ LanQ (<http://lanq.sourceforge.net/>)
- ▶ QPL and cQPL (<https://arxiv.org/abs/quant-ph/0511145>)
- ▶ Scaffold (<https://github.com/epiqc/ScaffCC>)

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

- ▶ Pierwsze wydanie w 1998, ostatnie w 2014 (<http://tph.tuwien.ac.at/~oemer/qcl.html>).

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

- ▶ Pierwsze wydanie w 1998, ostatnie w 2014 (<http://tph.tuwien.ac.at/~oemer/qcl.html>).
- ▶ Niezależny od architektury język programowania dla komputerów kwantowych.



# ✦ Programowanie wysokopoziomowe

## Przykład 1: QCL – nacisk na obliczenia kwantowe

### Cechy

- ▶ Składnia dla obliczeń odwracalnych (uncomputing).

# ✦ Programowanie wysokopoziomowe

## Przykład 1: QCL – nacisk na obliczenia kwantowe

### Cechy

- ▶ Składnia dla obliczeń odwracalnych (uncomputing).
- ▶ Różne typy pamięci kwantowej dla lepszej optymalizacji.

# ✦ Programowanie wysokopoziomowe

## Przykład 1: QCL – nacisk na obliczenia kwantowe

### Cechy

- ▶ Składnia dla obliczeń odwracalnych (uncomputing).
- ▶ Różne typy pamięci kwantowej dla lepszej optymalizacji.
- ▶ Warunki kwantowe — wykonanie kontrolowane kwantowo (uogólnienie bramek kontrolowanych).

# ✦ Programowanie wysokopoziomowe

## Przykład 1: QCL – nacisk na obliczenia kwantowe

### Cechy

- ▶ Składnia dla obliczeń odwracalnych (uncomputing).
- ▶ Różne typy pamięci kwantowej dla lepszej optymalizacji.
- ▶ Warunki kwantowe — wykonanie kontrolowane kwantowo (uogólnienie bramek kontrolowanych).
- ▶ Zróżnicowane typy struktur kontrolnych (związane z zarządzaniem pamięcią): operatory kwantowe, funkcje kwantowe, procedury.

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Zaawansowane zarządzanie pamięcią kwantową za pomocą typów:

- ▶ `qureg` — podstawowy typ na alokację rejestrów kwantowych,

## ❖ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Zaawansowane zarządzanie pamięcią kwantową za pomocą typów:

- ▶ `qureg` — podstawowy typ na alokację rejestrów kwantowych,
- ▶ `quconst` — stały rejestr kwantowy,

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Zaawansowane zarządzanie pamięcią kwantową za pomocą typów:

- ▶ `qureg` — podstawowy typ na alokację rejestrów kwantowych,
- ▶ `quconst` — stały rejestr kwantowy,
- ▶ `quvoid` — pusty rejestr, wykorzystywany do zapisania wyników,

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Zaawansowane zarządzanie pamięcią kwantową za pomocą typów:

- ▶ `qureg` — podstawowy typ na alokację rejestrów kwantowych,
- ▶ `quconst` — stały rejestr kwantowy,
- ▶ `quvoid` — pusty rejestr, wykorzystywany do zapisania wyników,
- ▶ `quscra`ch — rejestr pomocniczy, musi być pusty przed i po wywołaniu.



## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Typy funkcji kwantowych:

- ▶ procedure — obliczenia kwantowe z kontrolą klasyczną,

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Typy funkcji kwantowych:

- ▶ procedure — obliczenia kwantowe z kontrolą klasyczną,
- ▶ qufunct — **odwracalna** forma funkcji,

## ✦ Programowanie wysokopoziomowe

### Przykład 1: QCL – nacisk na obliczenia kwantowe

Typy funkcji kwantowych:

- ▶ procedure — obliczenia kwantowe z kontrolą klasyczną,
- ▶ qufunct — **odwracalna** forma funkcji,
- ▶ operator — złożona operacja kwantowa.

# ✦ Programowanie wysokopoziomowe

## Przykład 1: QCL – nacisk na obliczenia kwantowe

Some examples...

## ✦ Programowanie wysokopoziomowe

### Przykład 2: cQPL – nacisk na komunikację kwantową

- ▶ QPL (specyfikacja języka) oraz cQPL (implementacja QPL, <https://arxiv.org/abs/quant-ph/0511145>)

## ➤ Programowanie wysokopoziomowe

### Przykład 2: cQPL – nacisk na komunikację kwantową

- ▶ QPL (specyfikacja języka) oraz cQPL (implementacja QPL, <https://arxiv.org/abs/quant-ph/0511145>)
- ▶ Paradygmat funkcyjny.

## ✦ Programowanie wysokopoziomowe

### Przykład 2: cQPL – nacisk na komunikację kwantową

- ▶ QPL (specyfikacja języka) oraz cQPL (implementacja QPL, <https://arxiv.org/abs/quant-ph/0511145>)
- ▶ Paradygmat funkcyjny.
- ▶ Brak publicznie dostępnej implementacji.

## ✦ Programowanie wysokopoziomowe

### Przykład 2: cQPL – nacisk na komunikację kwantową

- ▶ QPL (specyfikacja języka) oraz cQPL (implementacja QPL, <https://arxiv.org/abs/quant-ph/0511145>)
- ▶ Paradygmat funkcyjny.
- ▶ Brak publicznie dostępnej implementacji.
- ▶ Dedykowana składnia do tworzenia kwantowych kanałów komunikacyjnych poprzez współdzielenie (splątanych) kubitów.





Pytania?

# Pytania?

<https://github.com/jmischczak/qprog-tutorial>