

Ryszard WINIARCZYK, Piotr GAWRON  
Instytut Informatyki Teoretycznej i Stosowanej PAN

## SYMULACJA KOMPUTERÓW KWANTOWYCH\*

**Streszczenie.** W artykule przedstawiamy trzy symulatory komputerów kwantowych: Quantum Computer Language, Quantum Computer Emulator oraz Quantum Circuit Simulator. Analizujemy potrzebę symulacji i przedstawiamy różne podejścia do tego zagadnienia.

## SIMULATION OF QUANTUM COMPUTERS

**Summary.** The article presents three quantum computer simulators: Quantum Computer Language, Quantum Computer Emulator and Quantum Circuit Simulator. Different approaches to the question of quantum computer simulation are described.

### 1. Wstęp

Badania nad opracowaniem komputerów kwantowych prowadzone są w ostatnich latach bardzo intensywnie, gdyż z tego typu rozwiązaniami wiąże się duże nadzieje. W literaturze pojawiają się liczne publikacje przedstawiające wyniki prac teoretycznych, jednakże badania doświadczalne są jeszcze na etapie realizacji pojedynczych prostych układów bramek logicznych. Nie można oczekiwać, że w ciągu najbliższego dziesięciolecia zostanie zbudowany nie laboratoryjny komputer kwantowy. W związku z tymi przesłankami symulacja komputerów kwantowych na maszynach klasycznych staje się niezbędna. Daje ona możliwości testowania istniejących i tworzenia nowych algorytmów oraz umożliwia wspomaganie dydaktyki. Ten ostatni aspekt symulacji obliczeń kwantowych może się okazać istotny, gdy komputery kwantowe zostaną wprowadzone do użytku i wystąpi zapotrzebowanie na informatyków zdolnych do programowania w nowym środowisku.

---

\* Praca wykonana w ramach grantu KBN 7T11C 017 21

Ponieważ dziedzina ta rozwija się bardzo szybko, nie zostały jeszcze ustalone żadne standardy, ani schematy działania. Próby ustanowienia takich są podejmowane – powstały propozycje sposobów prezentacji algorytmów kwantowych oraz nowych języków programowania [1] [2]. Obecnie pojawia się wiele różnorodnych propozycji symulatorów [3]. Prezentują różne podejścia do problemu, różne modele oraz poziomy abstrakcji. W tej pracy przedstawimy ich przegląd bazując na algorytmie wyszukiwania Grovera [4] [5] [6].

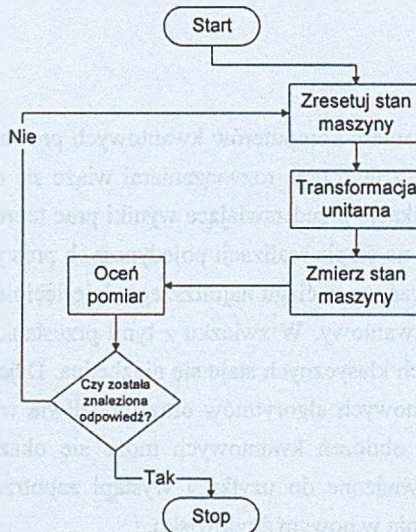
## 2. Różne podejście do symulacji

### 2.1. QCL – środowisko programowania komputera kwantowego

QCL [1] jest środowiskiem programowania komputerów kwantowych. Tak jak w klasycznych językach programowanie daje on możliwość opisu ciągu operacji elementarnych oraz zagnieżdżone poziomy abstrakcji.

QCL jest programem Unix-owym przeznaczonym do pracy zarówno w trybie wsadowym, jak i interaktywnym, jako interpreter.

Algorytm kwantowy w najprostszym przypadku możemy przedstawić za pomocą ciągu operacji unitarnych poprzedzonych zerowaniem, a zakończonych pomiarem. Taki przebieg ma np. symulacja układu kwantowego na komputerze kwantowym. Dla typowych zadań informatyki obliczenia kwantowe przyjmują postać algorytmów probabilistycznych. Rys. 1 przedstawia zarys podstawowego nieklasycznego algorytmu probabilistycznego.



Rys. 1. Schemat blokowy elementarnego algorytmu kwantowego

Fig. 1. Flow chart of elementary quantum algorithm



QCL, jak każdy z proponowanych dotychczas kwantowych języków programowania, oparty jest na konstrukcji hybrydowej; program zachowuje się z punktu widzenia użytkownika w sposób klasyczny: pobiera dane wyjściowe, przetwarza dane i zwraca wyniki. Różnica polega na tym, że przetwarzanie odbywa się również na maszynie kwantowej. Kontrola sterowania, czyli operacji na maszynie kwantowej, należy do komputera klasycznego.

QCL zawiera wszystkie cechy klasycznego języka programowania. „Kwantowość” tego języka wyrażana jest przez: zmienne typu rejestr kwantowy, operacje nieunitarne: zerowanie i pomiar, operacje unitarne: ogólny operator unitarny i operator pseudoklasyczny. QCL pozwala na odwracanie operacji unitarnych i może dokonywać zarządzania kwantową przestrzenią tymczasową, czyli istnieją wbudowane mechanizmy, które pozwalają na alokowanie i zwalnianie lokalnych rejestrów kwantowych wewnątrz funkcji.

Ponieważ sam język jako taki nie byłby użyteczny (z braku fizycznych implementacji komputera kwantowego), twórca QCL'a zintegrował go z symulatorem.

Przedstawimy teraz program zapisujący algorytm Grovera w jego najprostszej postaci (uwzględniającej wyszukiwanie tylko jednego klucza):

Funkcja kwantowa, która zwraca w rejestrze  $f$  wynik porównania zawartości rejestru kwantowego  $x$  i klucza  $n$ ,  $f := |1\rangle$  jeżeli  $x=n$  lub  $f := |0\rangle$  jeżeli  $x \neq n$

```
qufunct query(quireg x,quvoid f,int n) {
  int i;
  for i=0 to #x-1 {      // x -> NOT (x XOR n)
    if not bit(n,i) { Not(x[i]); }
  }
  CNot(f,x);           // flip f if x=1111..
  for i=0 to #x-1 {    // x <- NOT (x XOR n)
    if not bit(n,i) { !Not(x[i]); }
  }
}
```

Operator inwersji wokół średniej

```
operator diffuse(quireg q) {
  Mix(q);              // transformacja Hadamarda
  Not(q);              // odwróć q
  CPhase(pi,q);       // obróć pi jeżeli q=1111..
  !Not(q);             // wycofaj inwersję
  !Mix(q);            // wycofaj transformację Hadamarda
}
```

Operator inwersji amplitudy elementu szukanego

```
operator search(quireg q,int n) {
  int i;
  qureg f[1];
  for i=1 to ceil(sqrt(2^#q)) {
```

```

    query(q, f, n);
    CPhase(pi, f);
    !query(q, f, n);
    diffuse(q);
  }
}

```

Główna procedura programu, pobiera wartość zmiennej  $n$  (wyszukiwany klucz).

```

procedure grover(int n) {
  int l=floor(log(n,2))+1;      // ilość qubitów
  int m=ceil(pi/8*sqrt(2^l));  // ilość iteracji
  int x;
  int i;
  qureg q[l];
  qureg f[l];
  print l,"qubits, using",m,"iterations";
  {
    reset;
    Mix(q);                    // przygotuj superpozycję
    for i= 1 to m {           // główna pętla
      query(q, f, n);         // oblicz C(q)
      CPhase(pi, f);         // zanegu |n>
      !query(q, f, n);       // wycofaj obliczenie C(q)
      diffuse(q);            // operator dyfuzji
    }
    measure q, x;             // pomiar
    print "measured", x;
  } until x==n;
}

```

A tak wygląda wykonanie programu na symulatorze w trybie interaktywnym. Linia rozpoczęta przez „@” oznacza instrukcję elementarną, a linia rozpoczęta przez „%” oznacza wynik działania instrukcji.

```

qcl> grover(3) //Wyszukiwanie elementu o nr. 3, któremu odpowiada stan |10>
: 2 qubits, using 1 iterations
@ RESET
@ Mix(qureg q=|.10>) //Wstępne utworzenie superpozycji stanów
% 0.5 |000> + 0.5 |010> + 0.5 |001> + 0.5 |011>
@ CNot(qureg q=|. .>, quconst c=|.10>) //Początek operacji inwersji wokół średniej
% 0.5 |000> + 0.5 |010> + 0.5 |001> + 0.5 |111>
@ CPhase(real phi=3.141593, qureg q=|. .>)
% 0.5 |000> + 0.5 |010> + 0.5 |001> - 0.5 |111>
@ !CNot(qureg q=|. .>, quconst c=|.10>) //Koniec operacji inwersji wokół średniej
% 0.5 |000> + 0.5 |010> + 0.5 |001> - 0.5 |011>
@ Mix(qureg q=|.10>) //Początek operacji inwersji amplitudy elementu szukanego
% 0.5 |000> + 0.5 |010> + 0.5 |001> - 0.5 |011>
@ Not(qureg q=|.10>)
% 0.5 |011> + 0.5 |001> + 0.5 |010> - 0.5 |000>
@ CPhase(real phi=3.141593, qureg q=|.10>)
% -0.5 |011> + 0.5 |001> + 0.5 |010> - 0.5 |000>
@ !Not(qureg q=|.10>)
% -0.5 |000> + 0.5 |010> + 0.5 |001> - 0.5 |011>
@ !Mix(qureg q=|.10>) //Koniec operacji inwersji amplitudy elementu szukanego
% -1 |011>

```



```

@ MEASURE 1 |11>> //Pomiar
: measured 3
[0/3] -1 |011>

```

## 2.2. QCE – symulator modeli komputerów kwantowych

Ogólnie rzecz biorąc, warstwa sprzętowa komputera kwantowego jest modelowana przez spiny atomów (qubity), które ewoluują w czasie zgodnie z równaniem Shrödingera zależnym od czasu:

$$i \frac{\partial}{\partial t} |\Phi(t)\rangle = H(t) |\Phi(t)\rangle$$

gdzie

$$|\Phi(t)\rangle = a(0,0,\dots,0;t) |0,0,\dots,0\rangle + a(1,0,\dots,0;t) |1,0,\dots,0\rangle + \dots + a(1,1,\dots,1;t) |1,1,\dots,1\rangle$$

opisuje stan całego komputera kwantowego w chwili czasu  $t$ . Współczynniki zespolone  $a(0,0,\dots,0;t), \dots, a(1,1,\dots,1;t)$  całkowicie opisują stan układu kwantowego. Hamiltonian  $H(t)$  zależny od czasu przyjmuje formę:

$$H(t) = - \sum_{j,k=\alpha=x,y,z}^L J_{j,k,\alpha}(t) S_j^\alpha S_k^\alpha - \sum_{j=1}^L \sum_{\alpha=x,y,z} (h_{j,\alpha,0}(t) + h_{j,\alpha,1}(t) \sin(f_{j,\alpha}t + \phi_{j,\alpha})) S_j^\alpha, \quad (1)$$

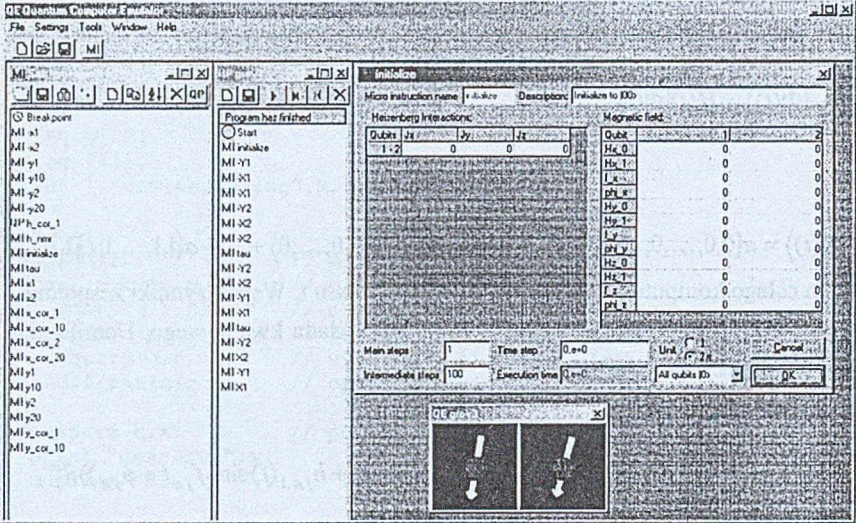
gdzie pierwsza suma przebiega po wszystkich parach  $P$  spinów (qubitów),  $S_j^\alpha$  opisuje  $\alpha$ -ty składnik spinu-1/2 operatora reprezentującego  $j$ -ty qubit,  $J_{j,k,\alpha}(t)$  determinuje siłę interakcji pomiędzy qubitami oznaczonymi  $j$  i  $k$ ,  $h_{j,\alpha,0}(t)$  i  $h_{j,\alpha,1}(t)$  oznaczają odpowiednio statyczne (magnetyczne) i okresowe pole działające na  $j$ -ty spin. Częstotliwość i faza zmiennego pola elektromagnetycznego jest oznaczona  $f_{j,\alpha}$  i  $\phi_{j,\alpha}$ . Liczba qubitów to  $L$ , a wymiar przestrzeni Hilberta  $D=2^L$ .

Hamiltonian (1) jest wystarczająco ogólny, by opisać najważniejsze cechy fizycznych modeli komputera kwantowego. Twórcy QCE pozostawili interakcje pomiędzy różnymi składowymi spinów, ponieważ nie została jeszcze wybrana technologia, w której będą konstruowane komputery kwantowe.

Algorytm kwantowy składa się z szeregu operacji elementarnych zwanych przez twórców QCE mikroinstrukcjami, reprezentującymi najmniejsze jednostki operacji, które może wykonać procesor kwantowy. Działanie mikroinstrukcji na stanie  $|\Psi\rangle$  procesora kwantowego jest opisane poprzez czas jej działania oraz wszystkie  $J$ -ty i  $h$  z równania (1), które pozostają stałe dla danej mikroinstrukcji [7].



QCE jest programem graficznym działającym w środowisku Windows z przyjaznym interfejsem użytkownika. W celu przeprowadzenia symulacji trzeba wybrać zbiór mikroinstrukcji, z których chcemy korzystać (do programu dołączone są liczne biblioteki), a następnie wprowadzić program. Przebieg symulacji jest na bieżąco wizualizowany; może on być również zapamiętany w pliku celem dalszej obróbki danych.



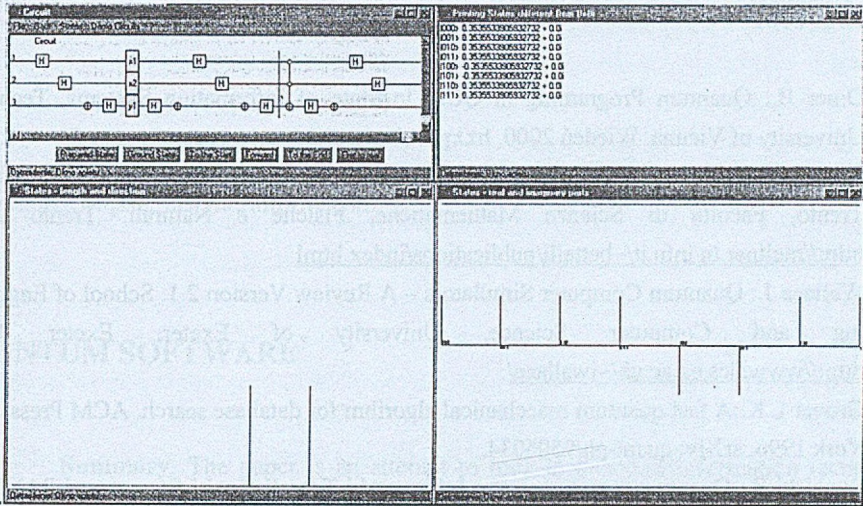
Rys. 2. Ekran symulatora QCE  
Fig. 2. View on QCE simulator

### 2.3. QuaSi2 – symulator edukacyjny

QuaSi2 jest symulatorem graficznym napisanym w Javie. Jest przeznaczony do symulacji algorytmów kwantowych przedstawionych w postaci bramek kwantowych. Program udostępnia cztery okna: układ bramek kwantowych, stan komputera kwantowego w postaci: liczbowej, wykresu Realis i Imaginaris oraz wykresu fazy i długości dla stanów bazowych.

Cechy QuaSi2 dają podstawowe możliwości symulowania obwodów kwantowych. Możliwość wizualizacji procesu obliczeniowego jest bardzo pomocna przy analizie na najniższym poziomie algorytmów kwantowych.





Rys. 3. Ekran symulatora QuaSi 2  
Fig. 3. View on QuaSi 2 simulator

### 3. Wnioski

Różnice pomiędzy pokazanymi tu symulatorami są bardzo wyraźne; wynika to z różnego podejścia ich twórców do problemu symulacji komputerów kwantowych. Zagadnienie symulacji komputera kwantowego może być rozpatrywane na różnych poziomach:

- stanu spinu w cząsteczce (QCE),
- obwodów bramek kwantowych (QuaSi2),
- języka opisu algorytmów kwantowych (QCL).

Zróznicowane poziomy abstrakcji opisu systemów fizycznych dają możliwość prowadzenia szerokich badań, mających na celu bądź to analizę fizycznych implementacji komputera kwantowego, bądź badanie obwodów kwantowych jako najprostszego modelu informatycznego, bądź przygotowanie języka programowania dla przyszłych fizycznych implementacji. Na każdym z tych poziomów dokonuje się odmiennych rodzajów badań.

## LITERATURA

1. Omer B.: Quantum Programming in QCL. Institute of Information Systems, Technical University of Vienna. Wiedeń 2000. <http://tph.tuwien.ac.at/~oemer>
2. Betteli S.: Toward an architecture for quantum programming. Università degli Studi di Trento, Facoltà di Scienze Matematiche, Fisiche e Naturali. Trento 2002. <http://meitner.tn.infn.it/~bettelli/publications/index.html>
3. Wallace J.: Quantum Computer Simulators – A Review Version 2.1. School of Engineering and Computer Science, University of Exeter. Exeter 1999. <http://www.dcs.ex.ac.uk/~jwallace/>
4. Grover L.K.: A fast quantum mechanical algorithm for database search. ACM Press, New York 1996. arXiv: quant-ph/9506034.
5. Bugajski S.: Quantum search. Instytut Informatyki Teoretycznej i Stosowanej PAN. Gliwice 2001.
6. Węgrzyn S., Klamka J.: Kwantowe systemy informatyki. Instytut Informatyki Teoretycznej i Stosowanej PAN. Gliwice.
7. De Readt H., Hams A. H., Michielsen K.: Quantum Computer Emulator. ArXiv:quant-ph/9911041. 2000. <http://rugth30.phys.rug.nl/compphys0/qce.htm>
8. QuaSi - Quantum Circuit Simulator, <http://iaks-www.ira.uka.de/home/matteck/QuaSi/aboutquasi.html>

Recenzent: Prof. dr hab. inż. Jerzy Klamka

Wpłynęło do Redakcji 24 kwietnia 2002 r.

**Abstract**

The Article presents three different quantum computer simulators: Quantum Computer Language, Quantum Computer and Quantum Circuit Simulator. Basic concepts of quantum computing and analyzes different levels of abstraction in quantum programming on real and simulated quantum machines. The programming principles and using QCL simulator is described on example of Grover's algorithm implementation.