

terminin aşağıdakı tərfi ən çox istifadə olunur: məlumat ətrafdakı dünya haqqında şüurlu bilikdir, saxlama, çevrilmə, ötürülmə və istifadə obyektidir.

INFORMATICS AND ITS ROLE IN MODERN WORLD

Kubishev İlgar Nadir

Summary

The science of methods and processes for collecting, storing, processing, transferring, analyzing and evaluating information using computer technologies that provide the ability to use it for decision-making is informatics. You can also give this definition: computer science is the science of some actions on information. In computer science, the following definition of this term is most often used: information is a conscious knowledge about the world around it, which is an object of storage, transformation, transmission and use.

Rəyçi: dosent Əkrəm Aslanov

İnformatika kafedrasının 16 noyabr 2021-ci il tarixli iclasın 03 sayılı protokolu.

Daxil olma tarixi 18 noyabr 2021-ci il

UOT:372.0:002

KVANT KOMPUTERLƏRİ ÜÇÜN PROQRAMLAŞDIRMA DİLLƏRİ

Cəbiyeva Şəbnəm Şahlar qızı

Azərbaycan Texniki Universiteti, Bakı, Azərbaycan

Zcshebnem@gmail.com

Xülasə: *Bütün bunlardan hansı nəticəyə gəlmək olar, kvant kompüterləri prinsipə yeni sistemdir. O, işlədiyi təmələ görə adi kompüterlərdən fərqlənir.*

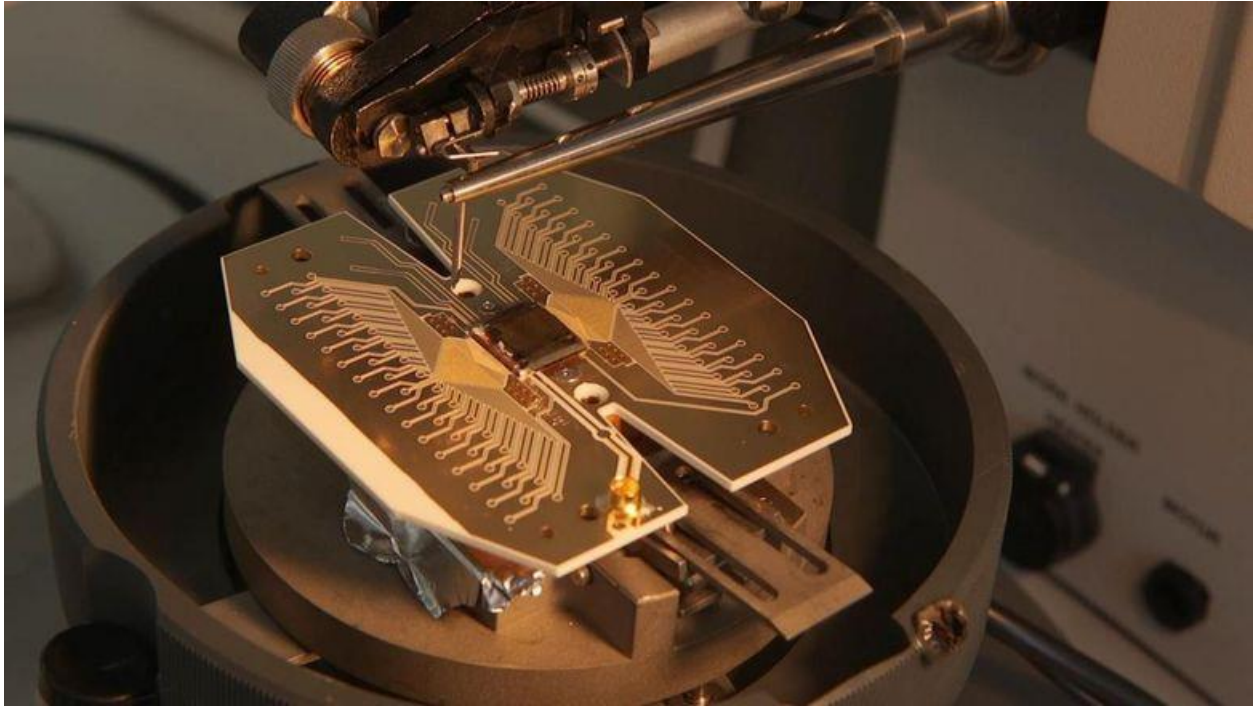
Onlar həqiqətən müqayisə edilə bilmərik. Bu, abak / hesab taxtasını və müasir kompüterləri müqayisə etmək kimidir!

Həmçinin, nə vaxtsa mağazaya gedib kiçik kvant prosessorlu kompüter ala biləcəyinizə dair böyük şübhələr var. Amma onlara ehtiyacınız yoxdur. Orta istifadəçi üçün kvant kompüterləri müasir məlumat mərkəzləri, yəni uzaqda yerləşən və sadəcə olaraq həyatımızı yaxşılaşdıran və ya heç olmasa fərqli edən görünməz köməkçilərimizə çevriləcək!

Açar sözlər: Kvant kompüterləri, Kvant hesablanması, proqram, proqramlaşdırma dilləri, alqoritm

Keywords: quantum computers, quantum computing, programming languages, program, algorithm

Ключевые слова: квантовый компьютер, квантовое вычисление, программа, языки программирования, алгоритм



İon kvant kompüterinin nüvəsinin prototipi. Ion Quantum Technology Group, Sussex Universiteti

Kvant kompüterləri haqqında xəbərlər zaman zaman mediada çıxır. İnsanların addım-addım bu növ kompüterlərin yaradılışına necə yaxınlaşdığını eşidirsiniz, baxmayaraq ki, əksəriyyət üçün kvant hesablamanın inkişafı qərribə və sirli bir sənət olaraq qalır.

Xoşbəxtlikdən, bu problemi həll etmək üçün geniş bir kütlənin diqqətini çəkən böyük layihələr ortaya çıxır. Məsələn, bir neçə il əvvəl IBM hər kəsə 5 qubitlik prosessorla işləyən bir kompüterə qoşulmağı mümkün etdi. Layihə üçün 70 min insan qeydiyyatdan keçdi.

Lakin, kvant hesablama sənayesi hələ başlanğıc mərhələsindədir. Bir çox prototip yaradılsa da, onlar hələdə adi dizüstü kompüterdən fərqli heç nə edə bilmirlər. Lazımi avadanlıq hələ yoxdur, amma öyrənməli olduğumuz başqa bir şey var - kvant proqramlaşdırma dilləri.

Kvant hesablama dünyasına səyahət hər kəs üçün əlçatandır və normal proqramlaşdırmada hər hansı bir adi dili öyrənmək kimidir.

Kvant hesablama

Kvant hesablama dünyasına dalmaq üçün evdə kvant kompüter saxlamağa ehtiyac yoxdur. Həqiqi kvant cihazlarına çıxış təmin edən layihələr var, ancaq sadə kvant proqramları adi bir dizüstü kompüterdə asanlıqla simulyasiya edilə bilər.

Proqramlara girməzdən əvvəl, kvant hesablama ilə bağlı bəzi əsas şeyləri xatırlayaq. Brian Hayesin "Kvant Kompüterinizi Proqramlaşdırma" məqaləsi bizə kömək edəcək.

Tipik bir kompüter bitlərə əsaslanır: yalnız iki mümkün dəyəri olan dəyişənlərdən bəhs edirik. Onları tez -tez 0 və 1 adlandırırıq, baxmayaraq ki, Boole cəbri kontekstində onlara "Doğru" və "Yanlış" deyə bilərik.

NOT, AND və OR kimi sadə boolean əməliyyatlar bitlərdə yerinə yetirilə bilər. Bir az daha mürəkkəb olan hər hansı bir dəyişən (int və ya float kimi) yalnız bir çox bitlərin toplusudur.

Kvant kompüterləri kvant bitlərinə və ya qubitlərə əsaslanır. Həm də 0 və 1 adlandırılacaq, lakin iki mümkün dəyər var. Amma kvant mexanikasının qanunları superpozisiya vəziyyətləri dediyimiz digər dəyərləri də qəbul edir.

Superpozisiya vəziyyətləri 0 və 1 -in arasında mövcud olan dəyərlərdir. Qubitəin əks qütblərdə 0 və 1 olduğu bir küərə olaraq düşünə bilərik. səthdəki bütün mümkün nöqtələrdir. Superpozisiya vəziyyətləri səthdəki bütün mümkün nöqtələrdir.

Məsələ ondadır ki, bir qubit aralıq dəyərə malik ola bilməz, məsələn 0,63; bir kubit vəziyyəti ölçüldükdə nəticə həmişə 0 və ya 1 olur. Amma hesablama zamanı qubit vəziyyətlərin qarışığı kimi hərəkət edə bilər - məsələn, 63% sıfır və 37% bir.

Qubit davranışının başqa bir əsas aspekti dalğa fizikasında yaxşı tanınan bir fenomen olan müdaxilədir. İki dalğa üst-üstə düşdükdə, onlar bir-birini gücləndirə bilər və ya bir-birini ləğv edə bilər (dalğalar fazadan kənardırsa). Riyazi olaraq hər hansı bir nöqtədə birləşmiş dalğaların intensivliyi fərdi dalğa amplitüdlərinin cəminin kvadratı ilə müəyyən edilir. İki amplitüd eyni işarəyə malik olduqda, müdaxilə konstruktivdir; bir amplituda müsbət, digəri mənfi olduqda, nəticədə yaranan dağıdıcı müdaxilə bir dalğadan daha az intensivlik verir.

Dalğalar kimi, qubit amplitudası 0 və 1 ola bilər, hansı ki, müsbət və ya mənfi ola bilər. Amplitüdlərin əlamətlərindən asılı olaraq, kvant müdaxiləsi bir qubitəin vəziyyətini ölçərkən müəyyən bir vəziyyətin müşahidə olunma ehtimalını artırır və ya azaldır.

Kvant kompüterləri üçün bütün maraqlı alqoritmlərdə müdaxilə böyük rol oynayır, yəni belə bir maşının klassik kompüterini üstələməsinə imkan verə biləcək bütün alqoritmlərdə. Kvant sisteminin təkamülünün təşkilinin ümumi ideyası ondan ibarətdir ki, yanlış cavablar dağıdıcı müdaxilə ilə sıxışdırılır, düzgün cavablar isə konstruktiv müdaxilə ilə gücləndirilir. Beləliklə, alqoritmlər kvant sistemlərinə xas olan paralellik formasından istifadə edir.

Kvant qəribəliyinin ən son aspektlərindən biri dolaşıqlıqdır. Siz kvant registrində bir qubit dəyişib, digərlərini isə dəyişməz saxlaya bilməzsiniz. İlk öncə, kvant sistemi ilə hesablanan hər bir funksiya tamamilə geri çevrilə bilər olmalıdır. Əgər maşın B nəticəsini çıxarmaq üçün A girişini alırsa, onda B varsa, A-nı bərpa etmək mümkün olmalıdır.

Hər bir funksiya eyni sayda giriş və çıxışa malik olmalıdır. Bir zərbə ilə bu qayda hesabın çoxunu qadağan edir. Adi alqoritmi geri dönməzdir. Məsələn, 7 əldə etmək üçün 3 və 4 əlavə edə bilərsiniz, lakin orijinal 3 və 4-ü bərpa etmək üçün 7-ni ayıra bilməzsiniz.

Kvant hesablamasındakı başqa bir qadağa qubit sürətidir (bu prinsip klonlanmayan teorem adlanır). Həmçinin, siz hesablamaların ortasında ixtiyari olaraq kubitləri yenidən quraşdırıb və ya yenidən yükləyə bilməzsiniz. Bunu etməyə cəhd kvant superpozisiyasını məhv edərdi.

Birlikdə götürdükdə, qubit əməliyyatlarına qoyulan məhdudiyyətlər hər hansı bir kvant proqramının boru arxitekturasına malik olmasını nəzərdə tutur - məlumat birbaşa bir ucdan digərinə axır. Proqram strukturunda sikl ola bilməz, hardakı idarəetmə əvvəlki nöqtəyə qaytarıla bilməz.

Bu çətinliklərin cavabı kvant proqramlaşdırma dillərində tapılır. Əslində, kvant kompüterləri hibrid cihazlardır: qismən kvant və qismən klassik kompüterlər. Adi kompüter elementlərinin istifadəsi xarici proqramlarla qarşılıqlı əlaqə yaratmaq üçün giriş və çıxışları emal etmək üçün lazımdır. Beləliklə, klassik kod və kvant kodu bir proqramda birləşdirilə bilər.

Open Quantum Assembly Language (OpenQASM)

Open QASM mənbə kodu Quantum Experience kvant hesablama platforması ilə istifadə üçün IBM Quantum Information Software Kit (QISKit) hissəsi kimi buraxılmışdır. OpenQASM, elektron sxemlərin quruluşunu və davranışını təsvir etmək üçün istifadə olunan xüsusi proqramlaşdırma dilləri (məsələn, Verilog) ilə oxşarlıqları vardır.

QASM proqramları əslində həmişə eyni şəkildə başlayır: biz bizə lazım olan bütün bitləri müəyyən edirik - həm kvant, həm də normal. Aşağıda OpenQASM mənbə koduna bir nümunə verilmişdir. Proqram iki dörd bitlik nömrə əlavə edir.

```

OPENQASM 2.0;
include "qelib1.inc";
gate majority a,b,c
{
    cx c,b;
    cx c,a;
    ccx a,b,c;
}
gate unmaj a,b,c
{
    ccx a,b,c;
    cx c,a;
    cx a,b;
}
qreg cin[1];
qreg a[4];
qreg b[4];
qreg cout[1];
creg ans[5];
// устанавливаем входящие значения
x a[0]; // a = 0001
x b;    // b = 1111
// добавляем a к b, сохраняем результаты в b
majority cin[0],b[0],a[0];
majority a[0],b[1],a[1];
majority a[1],b[2],a[2];
majority a[2],b[3],a[3];
cx a[3],cout[0];
unmaj a[2],b[3],a[3];
unmaj a[1],b[2],a[2];
unmaj a[0],b[1],a[1];
unmaj cin[0],b[0],a[0];
measure b[0] -> ans[0];
measure b[1] -> ans[1];
measure b[2] -> ans[2];
measure b[3] -> ans[3];
measure cout[0] -> ans[4];

```

Q#

Yüksək səviyyəli proqramlaşdırma dili Q # kvant fizikası üzrə dərin biliyə ehtiyacı aradan qaldırır. Dil dərsləri ilə maraqlananlar üçün vektor və matris riyaziyyatını əhatə edən kvant hesablamasının əsas anlayışları, kubitlər, Dirak qeydi, Pauli prinsipi və kvant sxemləri haqqında məlumat verilir.

```

operation Teleport(msg : Qubit, there : Qubit) : () {
    body {
        using (register = Qubit[1]) {
            let here = register[0];
            H(here);
            CNOT(here, there);
            CNOT(msg, here);
            H(msg);
            if (M(msg) == One) { Z(there); }
            if (M(here) == One) { X(there); }
        }
    }
}

```

Q # digər proqramlaşdırma dillərindən fərqli görünür və bir qədər C # dilinə bənzəyir.

Quantum Development Kit ətraflı quraşdırma təlimatları və giriş kurikulumları ilə pulsuz olaraq mövcuddur. Q #, 32-qubit kvant prosessorunu simulyasiya edən Visual Studio kvant simulyatorunda tərtib edir. Simulyator 40 kubit-ə qədər simulyasiya edə bilər.

LIQI (Language-Integrated Quantum Operations)

Microsoft Research-də Quantum Architectures and Computation Group tərəfindən yaradılmış LIQI platformasına proqramlaşdırma dili, optimallaşdırma və planlaşdırma alqoritmləri və bir neçə kvant simulyatoru daxildir. LIQI .NET Framework ailəsindən yüksək səviyyəli F # proqramı kimi yazılmış kvant alqoritmini kvant kompüterləri üçün aşağı səviyyəli təlimatlara çevirmək üçün istifadə edilə bilər.

Quantum Computation Language (QCL)
QCL və ya Quantum Computation Language, 1998-ci ildə Bernhard Omer tərəfindən yaradılmışdır. Dilin inkişafı bu gün də davam edir: klassik kompüterdə kvant proqramlarını işlətməyə imkan verən emulyator var. Əlbəttə ki, emulyator kvant paralelliyi üçün sürətlənmə təmin edə bilməz; digər tərəfdən, o, proqramçıya bəzi faydalı funksiyaları təklif edir, məsələn, kubitlərin daxili vəziyyətini yoxlamaq əməlləri (bunu real kvant aparatında etmək olduqca çətindir).

QCL və ya Quantum Computation Language, 1998-ci ildə Bernhard Omer tərəfindən yaradılmışdır. Dilin inkişafı bu gün də davam edir: klassik kompüterdə kvant proqramlarını işlətməyə imkan verən emulyator var. Əlbəttə ki, emulyator kvant paralelliyi üçün sürətlənmə təmin edə bilməz; digər tərəfdən, o, proqramçıya bəzi faydalı funksiyaları təklif edir, məsələn, kubitlərin daxili vəziyyətini yoxlamaq əməlləri (bunu real kvant aparatında etmək olduqca çətindir).

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

QCL, dəyişən dəyərləri təyin etmək və sıfırlamaq üçün birbaşa əmrlərə etibar etdikləri üçün bəzən "imperativ" dillər kimi təsvir edilən C və Java-dan sintaksisi götürür. Bu cür təlimatlar adətən kvant hesablamalarında qadağandır, buna görə də QCL proqramının əsas hissələri yalnız klassik aparat üzərində işləyir. Kvant sistemi kubitlərin hesablanması üçün uyğun formatda verilə bilən suallara cavab vermək üçün "oracle" rolunu oynayır. Oracle üçün hər bir sorğu tələb olunan baza arxitekturasına malik olmalıdır, lakin o, xarici klassik kontekstdə birləşdirilə bilər.

```
operator dft(quireg q) {
  const n=#q;
  int i; int j;
  for i=1 to n {
    for j=1 to i-1 {
      V(pi/2^(i-j),q[n-i] & q[n-j]);
    }
    H(q[n-i]);
  }
  flip(q);
}
```

Diskret Furje çevrilməsi Şorun faktorizasiya alqoritmində həlledici addımdır. Şor alqoritmində faktorlara bölünəcək ədəd dalğalı, dövri signal kimi qəbul edilir. Əgər N , u və v əmsallarına malikdirsə, o zaman N , u təkrar v və ya v təkrar u təkrarlarından ibarətdir. Şor alqoritmi bu cür təkrarların müddətini tapmaq üçün kvant paralelliyindən istifadə edir, baxmayaraq ki, proses yuxarıdakı nümunədə göründüyü qədər sadə və sadə deyil.

Digər Yanaşmalar



Rəngli kvadratlar beş IBM kvant bitinə nə edəcəyini bildirir. Kvadratları cəkarək Siz Öz kvant hesablamalarınızı yarada bilərsiniz.

IBM Quantum Experience layihəsi hər kəsə real kvant kompüterində eksperimental proqram işlətmək imkanı verir. IBM proqramlaşdırma dili ilə işləmək proqramla musiqi hazırlamağa bənzəyir. Proqramçı sadəcə olaraq proqram yazmaq üçün kvant obyektlərini müəyyən bir sahəyə sürükləyib buraxa bilər.

Quantum Computing Playground brauzer pəncərəsində kvant kompüterini işləməyi simulyasiya etməyə imkan verən WebGL Chrome təcrübəsidir. Səzləmə və 3D kvant vizuallaşdırma funksiyaları ilə öz Qscript dili var. Kvant hesablama platforması 22 kubit-ə qədər kvant registrlərini səmərəli şəkildə simulyasiya edə bilər.

Python QISKit SDK-ya kvant proqramlaşdırmasının məqsədlərini göstərmək üçün IBM Q mühəndislərinin təqdim etdiyi bir neçə alət daxildir. Xüsusilə, SDK mürəkkəb təcrübələr

üçün çoxlu tapşırıqları necə yerinə yetirə biləcəyinizi göstərir. Adından da göründüyü kimi, QISKit tərtibatçılara Python istifadə edərək kvant kompüterini araşdırmaq imkanı verir.

Onlarla kvant proqramlaşdırma dilləri (və simulyatorlar) var, lakin onların hamısı virtual maşında işləyir. IBM Q, ehtimal ki, real kvant kompüterinə çıxış təklif edən yeganə layihədir. Bununla belə, "kvant proqramlaşdırma" ilə məşğul olmağa başlamaq üçün əsl qabaqcıl cihaza çıxış əldə etmək heç də vacib deyil. Artıq təkcə perspektivli kvant alqoritmlərinin işini öyrənmək deyil, həm də oyunlar kimi işləyən proqramlar yaratmaq mümkündür. Amma bu, tamamilə başqa hekayədir.

Mənbələr

- [\[Even You Can Help Build a Quantum Computer\]](#)
- [\[Quipper: The First High-Level Scalable Programming Language for Quantum Computers\]](#)
- [\[Quantum Programming Language\]](#)
- [\[How to program a quantum computer\]](#)
- [\[Structured Quantum Programming\]](#)
- [\[QCL — A Programming Language for Quantum Computers\]](#)

ЯЗЫКИ ПРОГРАММИРОВАНИЯ ДЛЯ КВАНТОВЫХ КОМПЬЮТЕРОВ

Джабиева Шабнам Шахлар кызы

Резюме

Из всего этого можно сделать вывод, что квантовый компьютер - принципиально новая система. Он отличается от обычных компьютеров основанием, на котором работает.

Их действительно нельзя сравнивать. Это все равно, что сравнивать обычные счета и современные компьютеры!

Есть также серьезные сомнения в том, что когда-нибудь вы сможете пойти в магазин и купить компьютер с маленьким квантовым процессором. Но они тебе не нужны. Для обычного пользователя квантовые компьютеры станут современными центрами обработки данных, то есть нашими невидимыми помощниками, которые находятся далеко, и просто улучшат нашу жизнь или, по крайней мере, сделают ее другой!

PROGRAMMING LANGUAGES FOR QUANTUM COMPUTERS

Djabiyeva Shabnam Shahlar

Summary

From all this, we can conclude that a quantum computer is a fundamentally new system. It differs from conventional computers in the basis on which it operates.

They really can't be compared. It's like comparing abacus and modern computers!

There are also serious doubts that someday you will be able to go to the store and buy a computer with a small quantum processor. But you don't need them. For the average user, quantum computers will become modern data centers, that is, our invisible helpers who are far away, and will simply improve our life, or at least make it different!

Rəyçi: "Mühəndis riyaziyyatı və süni intellekt" kafedrasının t.e.n., dos. Qəhrəmanova T.İ.

"Mühəndis riyaziyyatı və süni intellekt" kafedrasının 05 noyabr 2021-ci il tarixli iclasın 05 sayılı protokolu

Daxil olma tarixi 18 noyabr 2021-ci il