

Квантови алгоритми : концепция и пример

Николай М. Николов

План :

1. Исторически бележки	2
2. Моделът на логическите изчислителни вериги ("класика")	5
3. Първо понятие за квантов алгоритъм	9
4. Ключови страни на новото понятие	12
5. Илюстрация на физическа реализация	14
6. Пример : намиране на период	15

Част, която е маркирана отпред със зелен маркер е извън материала в курса.

1. Исторически бележки

През около първата третина на XX век започва по-интензивното уточняване на понятието за алгоритъм. Предложени са няколко схеми (подхода) и през 1936* този процес в известен смисъл завършва с формулирането на тезиса на Черч (-Тюринг).

* - 1936 е също и годината на "квантовата лешка" - завършващ стадий в аксиоматизирането на квантова теория

Концепцията за квантов алгоритъм води началото си от 80-те години, с работите на Deutsch и Feynman, но още не е достигнала етапа на "тезиса на Черч"

Началото:

Deutsch D., Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer (1985)

[http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/\[1985-D\]\[cr\]_Quantum_Theory,_the_Church-Turing_Principle_and_the_Universal_Quantum_Computer-By_Deutsch_D-deutsch85.pdf](http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/[1985-D][cr]_Quantum_Theory,_the_Church-Turing_Principle_and_the_Universal_Quantum_Computer-By_Deutsch_D-deutsch85.pdf)

Proc. R. Soc. Lond. A **400**, 97–117 (1985)

Printed in Great Britain

Quantum theory, the Church–Turing principle and
the universal quantum computer

BY D. DEUTSCH

Department of Astrophysics, South Parks Road, Oxford OX1 3RQ, U.K.

(Communicated by R. Penrose, F.R.S. – Received 13 July 1984)

Feynman R.P., Simulating physics with computers (1982)

[http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/\[1982-F\]\[cr\]_Simulating_physics_with_computers-By_Feynman_R.P-feynman1982.pdf](http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/[1982-F][cr]_Simulating_physics_with_computers-By_Feynman_R.P-feynman1982.pdf)

International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Тази статия е забележителна
за мен с две особености:

- В нея се дискутира въпроса за симулация на физически процес на компютър и се досега до извода, че докато за класическата механика (където се решават крайни системи от ОДУ) това може да се извърши ефективно (т.е., "бързо") на класически компютър, то за квантовата механика се изчислява първо пространственото разпределение на вероятността с ЧДУ и това води до неефективност на симулацията

488

Feynman

infinite number of possible values, it'd have to be digitized. You might be able to get away with a theory by redescribing things without an electric field, but supposing for a moment that you've discovered that you can't do that and you want to describe it with an electric field, then you would have to say that, for example, when fields are smaller than a certain amount, they aren't there at all, or something. And those are very interesting problems, but unfortunately they're not good problems for classical physics because if you take the example of a star a hundred light years away, and it makes a wave which comes to us, and it gets weaker, and weaker, and weaker, and weaker, the electric field's going down, down, down, how low can we measure? You put a counter out there and you find "clunk," and nothing happens for a while, "clunk," and nothing happens for a while. It's not discretized at all, you never can measure such a tiny field, you don't find a tiny field, you don't have to imitate such a tiny field, because the world that you're trying to imitate, the physical world, is not the classical world, and it behaves differently. So the particular example of discretizing the electric field, is a problem which I would not see, as a physicist, as fundamentally difficult, because it will just mean that your field has gotten so small that I had better be using quantum mechanics anyway, and so you've got the wrong equations, and so you did the wrong problem! That's how I would answer that. Because you see, if you would imagine that the electric field is coming out of some 'ones' or something, the lowest you could get would be a full one, but that's what we see, you get a full photon. All these things suggest that it's really true, somehow, that the physical world is representable in a discretized way, because every time you get into a bind like this, you discover that the experiment does just what's necessary to escape the trouble that would come if the electric field went to zero, or you'd never be able to see a star beyond a certain distance, because the field would have gotten below the number of digits that your world can carry.

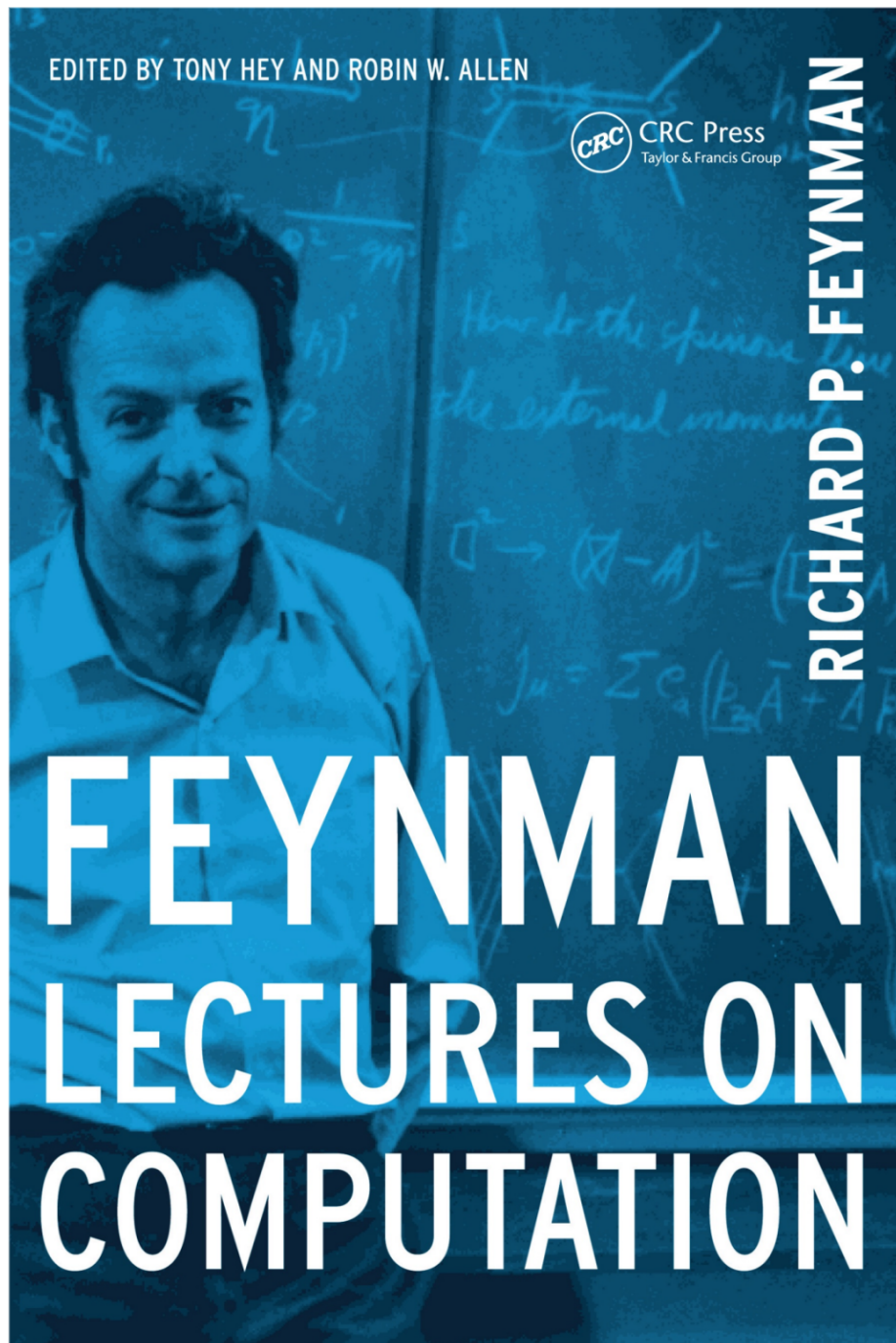
- няма цитирана литература ...

ОДУ = Обикновени Диференциални Уравнения
ЧДУ = Частни Диференциални Уравнения

Забележка: може да е полезно за пошук и информации да направят справка с посмъртно-издадената книга на Файнман по класическата теория на алгоритмите, за да придобият представа, как тази област се възприема от физиците.

Feynman, R. P. , Lectures on Computation (1996-2018)

[http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/\[1996-2018-F\]_Lectures%20on%20Computation-By_Richard_P._Feynman.pdf](http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/[1996-2018-F]_Lectures%20on%20Computation-By_Richard_P._Feynman.pdf)



(Boolean circuits)

2. Моделът на логическите изчислителни вериги ("класика")

[1996-2018-F], p.21-22

$$S := A + B \bmod 2$$

$$C := A + B - S \text{ в } \mathbb{Z} - \text{"carry"} \text{ ("на ум")}$$

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 2.1 A "Truth Table" for Binary Addition

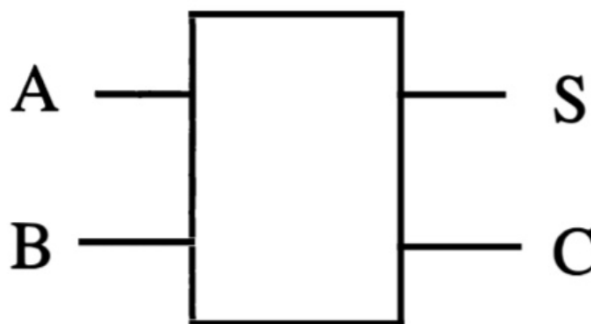


Fig. 2.2 A Black Box Adder

Забележка: горното съответствие $(A, B) \mapsto (S, C)$ не е биекция

Някои означения от курсовете по "уифрова електроника"
("digital electronics")

[1996-2018-F], p.22-26

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



AND

Fig. 2.3 The AND Gate

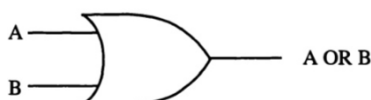
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



XOR

Fig. 2.4 The XOR Gate

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



OR

Fig. 2.5 The OR Gate

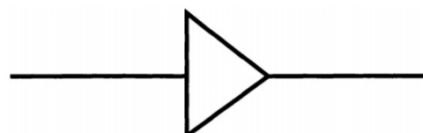


Fig. 2.6 The Identity

изключващо или
= събиране mod 2 ($:= \oplus$)

A	NOT A
0	1
1	0

Fig. 2.7 The NOT Gate

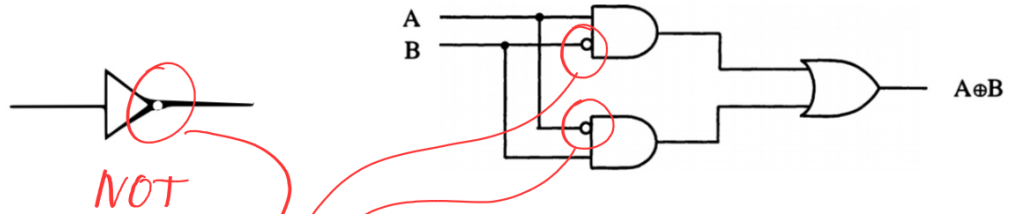


Fig. 2.9 XOR expressed in ANDs and ORs

физическа (инженерна)
реализация:

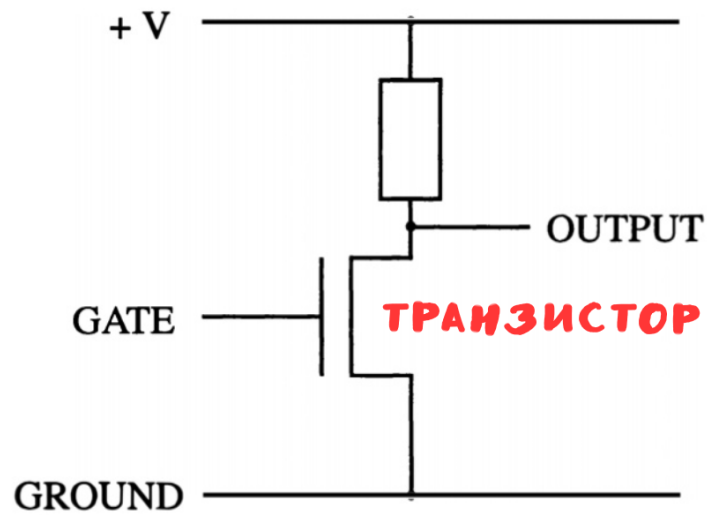


Fig. 2.12 The Transistor Inverter, or NOT Gate

Допълнителна забележка: гейт на Toffoli



$$\begin{cases} y_1 = x_1 + x_2 x_3 \pmod{2} \\ y_2 = x_2 \\ y_3 = x_3 \end{cases}$$

- въз. еднозн. и обр. $(\mathbb{Z}/2\mathbb{Z})^{\times 3}$
 $\{0, 1\}$

Значение: control bit: 0 $\rightarrow$ x_2 AND x_3
garbage bits

control bit: 0 $\rightarrow$ x_2 XOR x_3
garbage bits

control bit: 1 $\rightarrow$ NOT x_2
control bit: 0 $\rightarrow$ garbage bits

възпроизвежда
базисните логически
гейтове

остатъци при делене на 2

Точното понятие е "uniform circuit family" (съгласувана редица от логически изчислителни вериги) :

- редица от ориентирани графи Γ_n , който вертекси са декорирани от някакво крайно множество логически клетки $f_1, \dots, f_k$

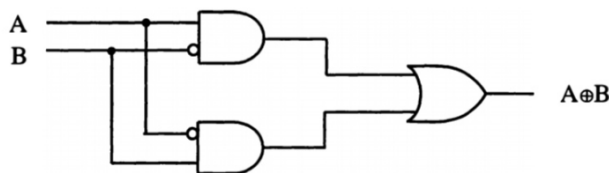


$$\forall f_j: \{0,1\}^{x_{r_j}} \rightarrow \{0,1\}^{x_{s_j}}$$

декартова степен

Например: Γ_n (за някое n) е :

[1996-2018-F], p.26



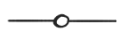
- от по-горе

Fig. 2.9 XOR expressed in ANDs and ORs

Тук базисните вертеки включват



,



,



$$\{0,1\}^{x_2} \xrightarrow{AND} \{0,1\}, \quad \{0,1\} \xrightarrow{NOT} \{0,1\}, \quad \{0,1\} \xrightarrow{\Delta} \{0,1\}^{x_2}$$

репликация
(диагоналното
изображение)

- Тогава на $\forall$ граф Γ_n се съставя функция

$$\{0,1\}^{x_{R_n}} \xrightarrow{\Gamma_n} \{0,1\}^{x_{S_n}}$$

броя входни линии

броя изходни линии

според композицията, която се определя от графа.

- Редицата се нарича съгласувана ("uniform"), ако

$$R_1 < R_2 < \dots < R_n < \dots, \quad S'_1 < S'_2 < \dots < S'_n < \dots$$

и за $m < n$, то

$$\Gamma_m(x_1, \dots, x_{R_m}) = (y_1, \dots, y_{S'_m})$$

$$\Rightarrow \Gamma_n(0, \dots, 0, x_1, \dots, x_{R_m}) = (0, \dots, 0, y_1, \dots, y_{S'_m})$$

В допълнение: нас ще ни интересуват съгласувани редици $\{\Gamma_n\}$, които са алгоритмично породени.

Така, горното понятие всъщност не е всъщност определение (или модел) на алгоритъм. То обаче се явява полезно при анализа на понятията за сложност на алгоритъм.

- источник за неспециалисти: https://en.m.wikipedia.org/wiki/Circuit_complexity

За специалисти: има връзката между "uniform circuit family" и машина на Тюринг?

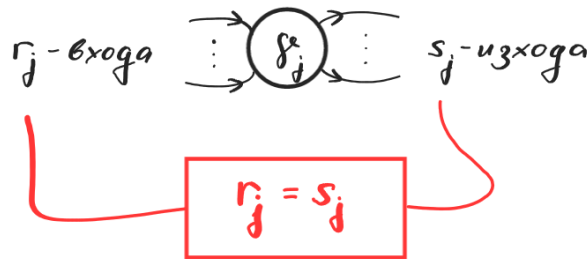
Горният модел на изчисления по-скоро отговаря на електронен калкулатор ("елка"), отколкото на "полен компютър".

От този модел произлиза и основният модел на квантов компютър, който по-скоро е "квантова елка".

3. Първо понятие за квантов алгоритъм: съпоставяна редица от квантови изчислителни вериги

- Отново, както и по-горе в с. 2, стартираме с алгоритмично породена редица от насочени графи, чийто вертекси са избрани измежду крайно (или, по-общо, алгоритмично избримо) множество от вертекси.

ИМА ВАЖНО ОГРАНИЧЕНИЕ:



броя на входовете = броя на изходите

$\Rightarrow$ това е така и за целите графи Γ_n : $R_n = S_n$

- интерпретация на верига Γ , като изображение: ^(с.е., граф)

На $\forall g_j$ съпоставяме унитарен оператор ($\equiv$ матрица)

$$g_j : \underbrace{(\mathbb{C}^2)^{\otimes \Gamma_j}}_{\cong \mathbb{C}^{2^{\Gamma_j}}} \longrightarrow \underbrace{(\mathbb{C}^2)^{\otimes \Gamma_j}}_{\cong \mathbb{C}^{2^{\Gamma_j}}} \quad \text{тензорна степен}$$

$\Rightarrow$ при интерпретацията на веригата, като композиция:

$$\Gamma_n : (\mathbb{C}^2)^{\otimes R_n} \longrightarrow (\mathbb{C}^2)^{\otimes R_n} \quad \text{- унитарно}$$

За целта, следваме следните конвенции от линейната алгебра:

— стандартния базис в $\mathbb{C}^2$ е $\{|0\rangle, |1\rangle\}$; той е орто-нормиран.

— стандартния базис в $(\mathbb{C}^2)^{\otimes n} \equiv \mathbb{C}^{2^n}$

е $|k_{n-1}\rangle \otimes \dots \otimes |k_0\rangle \equiv |\ell\rangle \equiv |k_{n-1} \dots k_0\rangle$, за $\ell = k_{n-1}2^{n-1} + \dots + k_12^1 + k_02^0$
(отново, орто-нормиран) — двоично разлагане.

нататък се свиква "изчислителни базиси" (computational basis).

— Нека $g: (\mathbb{C}^2)^{\otimes r} \longrightarrow (\mathbb{C}^2)^{\otimes r}$ е линейно с матрица:

$$g(|k_1\rangle \otimes \dots \otimes |k_r\rangle) =: \sum_{k'_1, \dots, k'_r = 0,1} g_{k_1, \dots, k_r; k'_1, \dots, k'_r} |k'_1\rangle \otimes \dots \otimes |k'_r\rangle$$

всичко ще ги изпускаме както по-горе

Тогава за всяко влагане $1 \leq t_1 < t_2 < \dots < t_r \leq n$ ($t_a \in \mathbb{N}$) полагаме $1 \leq t'_1 < t'_2 < \dots < t'_r \leq n$ ($t'_a \in \mathbb{N}$)

$(g)_{t_1, \dots, t_r; t'_1, \dots, t'_r}: (\mathbb{C}^2)^{\otimes n} \longrightarrow (\mathbb{C}^2)^{\otimes n}$ — линейно изображение, определено от:

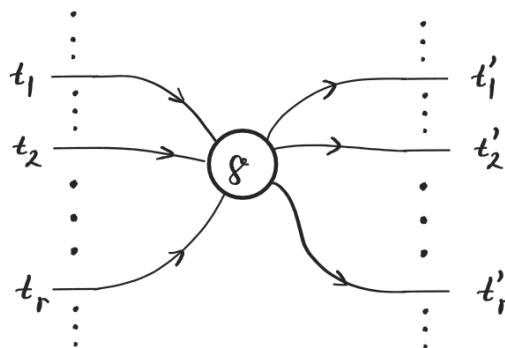
$$(g)_{t_1, \dots, t_r; t'_1, \dots, t'_r} |\dots k_{t_1} \dots k_{t_2} \dots k_{t_r} \dots\rangle$$

$$:= \sum_{k'_{t_1}, \dots, k'_{t_r} = 0,1} g_{k_{t_1}, \dots, k_{t_r}; k'_{t_1}, \dots, k'_{t_r}} |\dots k'_{t'_1} \dots k'_{t'_2} \dots k'_{t'_r} \dots\rangle$$

с уговорката, че в многотогизата не "пишем нищо".

Унитарно е, ако и g е унитарно

Графичен израз:



— Вертексите могат да се разширяват допълнително с действие на пермутации: g^σ

- Условие за "целостна класичност" (мой израз!)

$\forall n \quad \Gamma_n \{ \text{изчислителен базис на } (\mathbb{C}^2)^{\otimes R_n} \} = \{ \text{изчислителен базис на } (\mathbb{C}^2)^{\otimes R_n} \}$

т.е., $\Gamma_n (|x_1 \dots x_{R_n}\rangle) = |y_1 \dots y_{R_n}\rangle$

и $(x_1, \dots, x_{R_n}) \mapsto (y_1, \dots, y_{R_n})$ е целого изчислително съответствие

$$\{0, 1\}^{x R_n} \rightarrow \{0, 1\}^{x R_n}$$

По-слабо, условие: разглежда се двойна фамилия

$$\Gamma_{n,h} : (\mathbb{C}^2)^{\otimes R_n} \rightarrow (\mathbb{C}^2)^{\otimes R_n}, \text{ където } R_n \text{ не зависи от } h=1, 2, \dots$$

$$\Gamma_{n,h} (|x_1 \dots x_{R_n}\rangle) = |y_1 \dots y_{R_n}\rangle + \Theta_h$$

където $\|\Theta_h\|^2 = 1 - (\text{Probability for } (y_1, \dots, y_{R_n}))^2 \xrightarrow{h \rightarrow \infty} 0$

и това увеличава изчислителното време.

В обобщения случай, квантовият алгоритъм е вероятностен, с възможност за грешка, но в този случай се счита, че има "бърз" класически алгоритъм за проверка на отговора.

- Условие за съгласуваност: както и в класическия случай искаме получението

съответствие $(x_1, \dots, x_{R_n}) \mapsto (y_1, \dots, y_{R_n})$

$$\{0, 1\}^{x R_n} \rightarrow \{0, 1\}^{x R_n} \quad \text{да съгласувани.}$$

- Допълнителни отслабвания: - частична дефиниционна област, при условие, че

има "бърза" класическа проверка за допустимост на входните данни

- може да има допълнителни битовете (входни-контролни и изходни-дагбаде).

- може недетерминистичен изход: $\sum \Psi_{x_1, \dots, x_{R_n}}^{\leftarrow \in \mathbb{C}} |x_1 \dots x_{R_n}\rangle$

тогава $|\Psi_{x_1, \dots, x_{R_n}}|^2 = \text{вероятността за изход } (x_1, \dots, x_{R_n})$.

- може системно неединно измерване по с.нар. "проекционен постулат" на ф.Н.

4. Ключови страни на новото понятие - "квантов алгоритъм"

- В него е заложено понятието за класически алгоритъм. Оттук се прави извода, че класа на "квантово изчислимите функции" не е по-голям от класическия клас (?) разбира се, при условие, че базисните квантови гейтове (вертекс) са класически изчислими матрици.
- Квантовите изчисления са винаги обратими, заедно с базисните гейтове. Това е ограничение още на класическо ниво !!! По-такъв начин квантовите изчисления са подобрение (ускоряване) на, евентуално, предварително вложени (задавени) класически алгоритми. Затова, не е ясно дали като цяло ще има подобрение (поне за мен !?)
- Теоремата на Тоголи все пак ни казва, че изискването за обратимост дори на класическо ниво не стеснява класа на изчислимите функции: с цената на добавяне на допълнителни битовете (входни - контролни, изходни - дагъве) всяка редица от класически изчислителни вериги може да се възпроизведе от редица от класически обратими такива.

- Отворен (за мен) въпрос е дали всяка биекция (пермутация)

$S_{2^n} \ni \sigma : \{0, 1\}^{2^n} \rightarrow \{0, 1\}^{2^n}$ може да се представи, като композиция по верига от $\sigma' \in S_{2^m}$ за $m < n$.

Но това е в посока на т.нар. "обратими изчисления" (класически)
/ reversible computation /

Една от практическите ползи на тази област е в борбата със затоплянето (в т.е. и "глобалното").

- По-определение $\hat{M}(|x_1 \dots x_n\rangle) := |y_1 \dots y_n\rangle$,
ако $M(x_1, \dots, x_n) = (y_1, \dots, y_n)$ ($x_j, y_k \in \{0, 1\}$)

-ще го наричам "квантуван класически гейт" (мой израз).

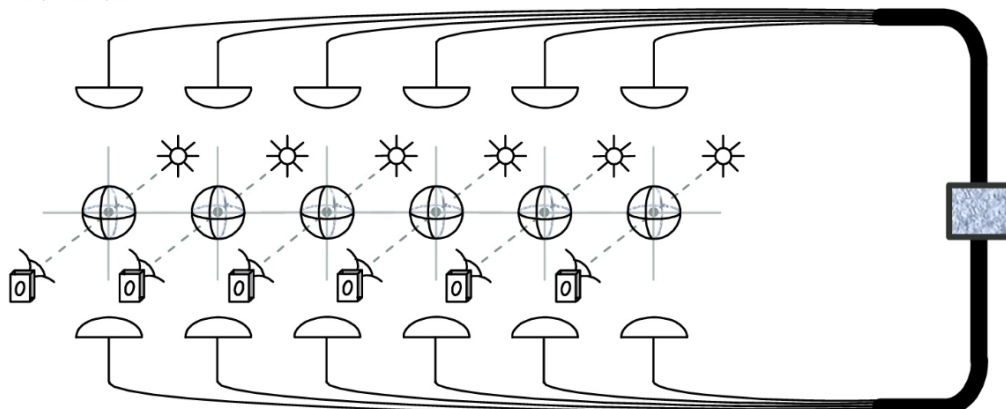
- Наивно, причината за оскъпването съществено намаляване на броя елементи в квантовите изчислителни вериги, т.е., ускоряването на изчислението, е в това, че групата на класическите обратими изчисления върху н бита, S_{2^n} , се потапя в безкрайната група $U(2^n)$

В литературата има понятие универсална система от квантови гейтове : това са $g_j \in U(2^{r_j})$ т.е. при всевъзможните композиции по вериги те дават точно подмножество в $U(2^n)$ за $\forall n \in \mathbb{N}$

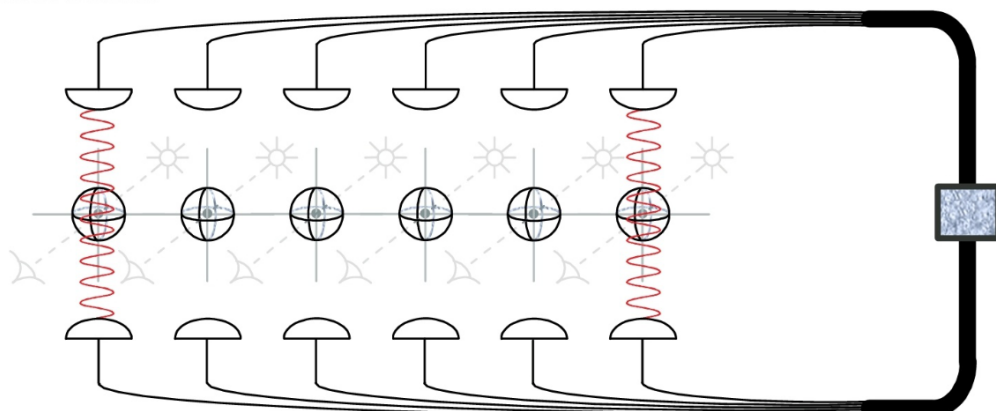
Това обаче изглежда (поке от пръв поглед) твърде силно, по-късно в крайна сметка ние искаме да апроксимирате само $S_{2^n} \subseteq U(2^n)$.

5. Илюстрация на физическа реализация

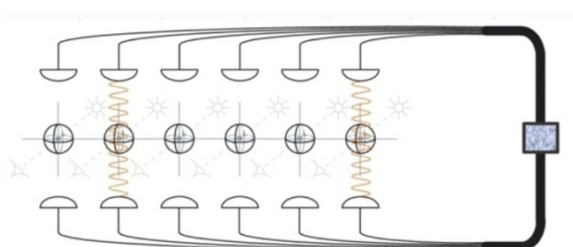
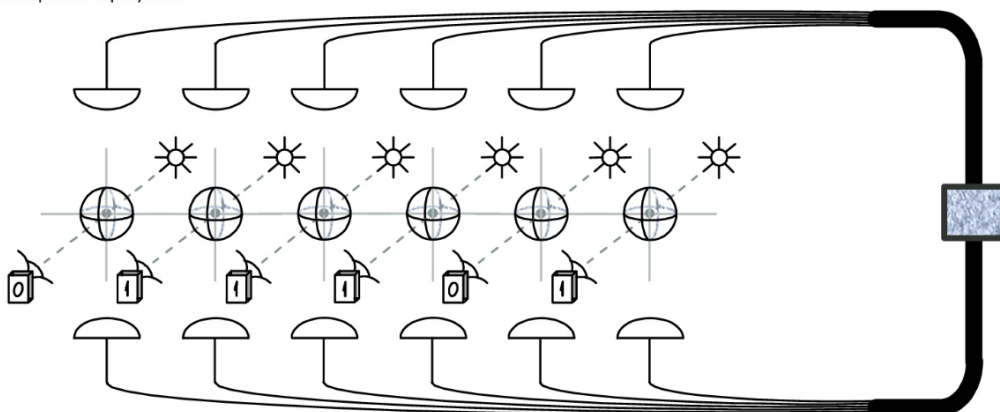
1) Инициализация



2) Квантово изчисление

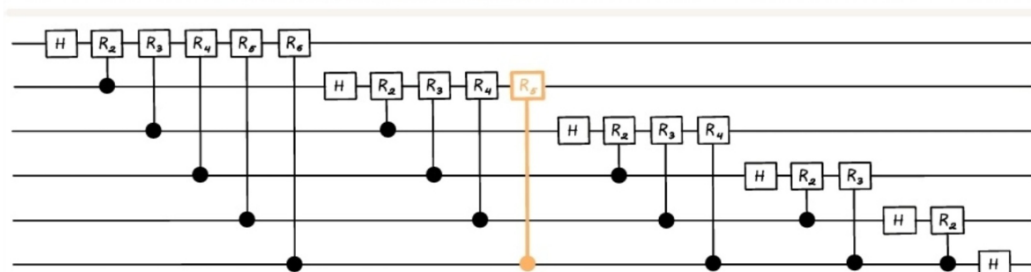


3) Измерване на резултата



Гравидните квантови трансформации, които задават елементарните стъпки, се наричат "квантови операции" (quantum gates).

Тяхната последователност образува "квантова верига" (quantum circuit).



6. Пример: намиране на период : дадена е функция $f: \mathbb{N} \rightarrow \mathbb{N}$, която е периодична, $f(X+P) = f(X)$ и алгоритмично изчислима; $P = \text{период} = ?$
 Например, $f(X) = Y^X \bmod Z$ (оттук следва намирането на прост делител)

а) Мошви и цели .

► Всяко изчисление върху n -бита, изсяга $N = 2^n$ - възможности .

► Хармон на физичите :

Изчислителният процес ще наричаме "бърз", ако времето му има полиномиален ръс спрямо $n = \log N$

Ще казваме, че е "бавен", ако ръсът на времето е полиномиален, но спрямо $N = 2^n$, т.е., то n е експоненциален .

► Сред бързите изчисления са събирането, умножаването, деленето, степенуването .

Предполагаме, че алгоритъмът за функция $f(X)$ по-горе е "бърз".

► Известните класически алгоритми за намиране на прост делител и периоди са бавни, понеже са посвещават "бързи проверки" на $N = 2^n$ възможности .

► Квантовите алгоритми за намиране на прост делители или на периоди не са детерминистични и също изискват на финала "бързи проверки".
 Те обаче генерират специални вероятностни разпределения върху множеството за търсене, така че за $\forall p \in (0, 1)$ да $\exists$ определен полиномиален ръс $T(n)$, така че с вероятност p да получим резултата за време $< T(n)$

► Източници :

[1997-S] PETER W. SHOR, POLYNOMIAL-TIME ALGORITHMS FOR PRIME FACTORIZATION AND DISCRETE LOGARITHMS ON A QUANTUM COMPUTER, SIAMJ.COMPUT. Vol. 26, No. 5, pp. 1484-1509, October 1997

[http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/\[1997-S\]\[cr\]_POLYNOMIAL-TIME_ALGORITHMS_FOR_PRIME_FACTORIZATION_AND_DISCRETE_LOGARITHMS_ON_A_QUANTUM_COMPUTER-By_PETER_W._SHOR-shor1997.pdf](http://theo.inrne.bas.bg/~mitov/QuInfLit/Seminar071022/[1997-S][cr]_POLYNOMIAL-TIME_ALGORITHMS_FOR_PRIME_FACTORIZATION_AND_DISCRETE_LOGARITHMS_ON_A_QUANTUM_COMPUTER-By_PETER_W._SHOR-shor1997.pdf)

[2000-2010-NC] Nielsen M.A., Chuang I.L., Quantum Computation and Quantum Information, 10th Anniversary Edition Chapt. 5

[http://theo.inrne.bas.bg/~mitov/QuInfLit/Basic_Refs/\[2000-2010-NC\]_Nielsen%20M.A.,%20Chuang%20I.L.,%20Quantum%20Computation%20and%20Quantum%20Information,%2010th%20Anniversary%20Edition.pdf](http://theo.inrne.bas.bg/~mitov/QuInfLit/Basic_Refs/[2000-2010-NC]_Nielsen%20M.A.,%20Chuang%20I.L.,%20Quantum%20Computation%20and%20Quantum%20Information,%2010th%20Anniversary%20Edition.pdf)

б) Базисни гейтове

$$1 = e^0 \quad y=0 \quad y=1 \quad \boxed{e^{\frac{2\pi i}{2} x} = (-1)^x}$$

$$\bullet \quad \text{---} \boxed{H} \text{---} \quad H|x\rangle = 2^{-1/2} (|0\rangle + (-1)^x |1\rangle) = 2^{-1/2} \sum_{y=0,1} \exp\left(2\pi i \frac{xy}{2}\right) |y\rangle$$

$$2^{-1/2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\bullet \quad \text{---} \boxed{R_a} \text{---} \quad R_a |xy\rangle = \exp\left(2\pi i \frac{xy}{2^a}\right) |xy\rangle$$

$$\left. \begin{array}{c} \text{---} \boxed{R_a} \text{---} \\ \text{---} \bullet \text{---} \end{array} \right\} \text{алтернативно означение (тип "условна операция")}$$

- Ще считаме, че имаме "бърза", обратима класическа верига

$$\begin{array}{c} X \\ V \end{array} \text{---} \boxed{O_f} \text{---} \begin{array}{c} X \\ V \end{array} \oplus^n f(X) \quad \text{собиране бит по бит mod 2} \quad \text{за } X, Y \leq 2^n.$$

Ще я пренесем "квантово" със същото означение

По-такъв начин, най-общо ще построим квантов алгоритъм, който "вход" е класически алгоритъм.

За по-конкретно, когато $f(X) = Y^X \text{ mod } Z$, O_f ще придобие вида

$$\begin{array}{c} X \\ Y \\ Z \\ V \\ \text{(control bits) } W \end{array} \text{---} \boxed{U_n} \text{---} \begin{array}{c} X \\ Y \\ Z \\ V \oplus^n (Y^X \text{ mod } Z) \\ T \text{ (garbage)} \end{array}$$

Трябва да се внимава с остатъчните помощни битове (garbage), които накрая не се четат: нужно е техните свойства да не се променят при промяна на "оперативните данни" - иначе техните вектори няма да се "офакторизират".

Упражнение 1:

$$|0 \dots 0\rangle \left\{ \begin{array}{l} |0\rangle \xrightarrow{H} 2^{-1/2} (|0\rangle + |1\rangle) \\ |0\rangle \xrightarrow{H} 2^{-1/2} (|0\rangle + |1\rangle) \\ \vdots \\ |0\rangle \xrightarrow{H} 2^{-1/2} (|0\rangle + |1\rangle) \end{array} \right\} 2^{-n/2} (|0\rangle + |1\rangle) \otimes \dots \otimes (|0\rangle + |1\rangle)$$

$H^{\otimes n}$

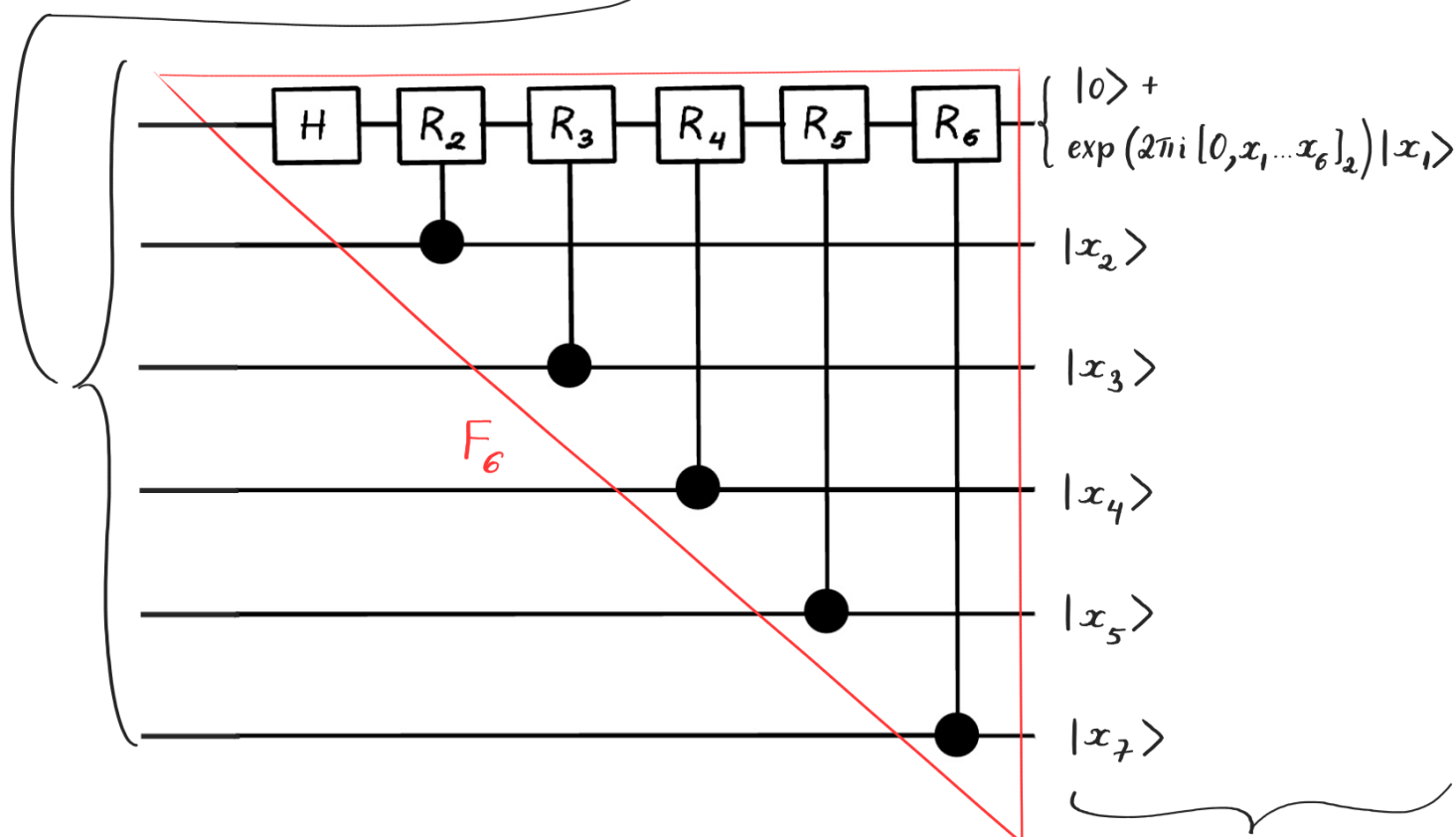
$$= 2^{-n/2} \sum_{x_1, \dots, x_n = 0, 1} |x_1 \dots x_n\rangle$$

“първа основна теорема на квантовата информатика” (:)

Важна терминология:

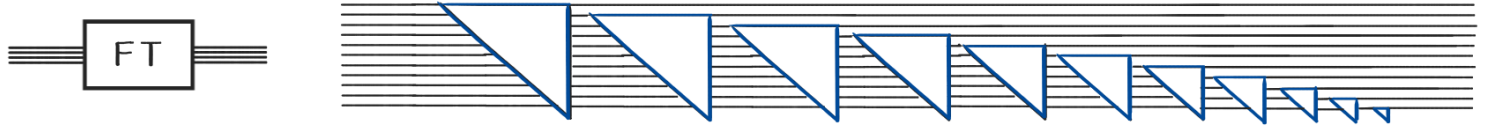
“разпространение по линейност” =: “квантов паралелизъм” (:)

Упражнение 2: $|x_1 \dots x_5\rangle$



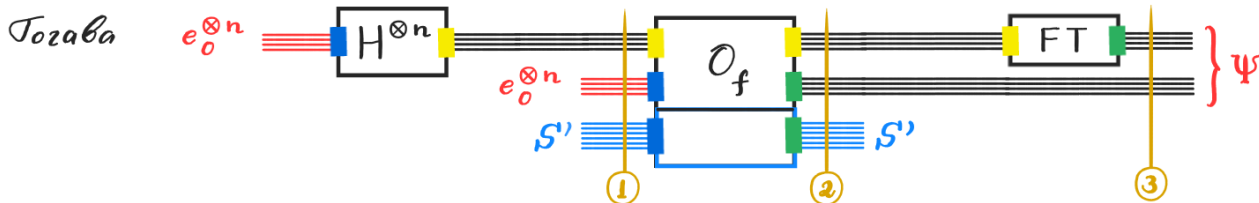
$$2^{-1/2} (|0\rangle + \exp(2\pi i [0, x_1 \dots x_6]_2) |x_1\rangle) \otimes |x_2 \dots x_5\rangle$$

Означаваме:



Изчислителната сложност на FT е $n + (n-1) + \dots + 1 = \frac{1}{2} n(n+1)$

- полиномиална



легенда:

≡ - нулеви входни битовете

■ - входни битовете за цялата верига

■ - междинни битовете на веригата

■ - изходни битовете за цялата верига

≡ - допълнителни помощни битовете

③ - междинно изчисление

↘ - стандартен ред на четене

Пресмятаня

$$1) \text{FT}(|x_1 \dots x_n\rangle)$$

$$= 2^{-n/2} \bigotimes_{k=0}^{n-1} \left(|0\rangle + \exp\left(2\pi i \frac{[x_1 \dots x_n]_2}{2^n} 2^k\right) |1\rangle \right)$$

$$= 2^{-n/2} \bigotimes_{k=0}^{n-1} \left(\sum_{y_k=0,1} \exp\left(2\pi i \frac{[x_1 \dots x_n]_2}{2^n} y_k 2^k\right) |y_k\rangle \right)$$

$$= 2^{-n/2} \sum_{y_1, \dots, y_n=0,1} \exp\left(\frac{2\pi i}{2^n} [x_1 \dots x_n]_2 [y_n \dots y_1]_2\right) |y_1 \dots y_n\rangle$$

да означим $|X\rangle := |x_1 \dots x_n\rangle$, ако $X = [x_1 \dots x_n]_2$

$$\bar{X} := [x_n \dots x_1]_2$$

$$\text{Тогава } \text{FT}(|X\rangle) = 2^{-n/2} \sum_{Y=0}^{2^n-1} \exp\left(\frac{2\pi i}{2^n} X \bar{Y}\right) |Y\rangle$$

2) Пресмятане на Ψ

Ще извършим анализа на алгоритма при условие, че $f(X)$ е биекция върху периода си

$$f: \{0, 1, \dots, P-1\} \cong \{0, 1, \dots, P-1\}$$

Тогава:

$$\textcircled{1} = (H^{\otimes n} |0\rangle) \otimes |0\rangle \otimes |S\rangle$$

фиксиран, но неотчетен

$$\xrightarrow{O_f} = 2^{-n/2} \sum_{X=0}^{2^n-1} |X\rangle \otimes |0\rangle \otimes |S\rangle$$

$$\textcircled{2} = 2^{-n/2} \sum_{X=0}^{2^n-1} O_f(|X\rangle \otimes |0\rangle) \otimes |S\rangle$$

$$= 2^{-n/2} \sum_{X=0}^{2^n-1} |X\rangle \otimes |f(X)\rangle \otimes |S\rangle$$

$$= 2^{-n/2} \sum_{C=0}^{P-1} \sum_{L=0}^{K-1} |X_C + LP\rangle \otimes |C\rangle \otimes |S\rangle + \text{remainder}$$

остатък при делене на X с P , като сме положили

$f(X_C) = C$ и съответствието $X_C \leftrightarrow C$ е биекция,

тъй като f е биекция върху периода си.

К е остатък при делене на 2^n на P , т.е., колко пъти периода влиза в 2^n

Положаваме: $R := 2^n - KP$ - остатък.

От тук идва остатъчната сума:

пренебрегваме remainder

$$\textcircled{3} \approx 2^{-n/2} \sum_{C=0}^{P-1} \sum_{L=0}^{K-1} FT(|X_C + LP\rangle) \otimes |C\rangle \otimes |S\rangle$$

$$= 2^{-n/2} \sum_{C=0}^{P-1} \sum_{L=0}^{K-1} 2^{-n/2} \sum_{Y=0}^{2^n-1} \exp\left(\frac{2\pi i}{2^n} (X_C + LP) \bar{Y}\right) |Y\rangle \otimes |C\rangle \otimes |S\rangle$$

$$= 2^{-n} \sum_{C=0}^{P-1} \sum_{Y=0}^{2^n-1} \left[\exp\left(\frac{2\pi i}{2^n} X_C \bar{Y}\right) \sum_{L=0}^{K-1} \exp\left(\frac{2\pi i}{2^n} P \bar{Y}\right)^L \right] |Y\rangle \otimes |C\rangle \otimes |S\rangle$$

$\Psi_{Y,C,S}$

$$= 2^{-n} \sum_{C=0}^{P-1} \sum_{Y=0}^{F'-1} \Psi_{Y,C,S} |Y\rangle \otimes |C\rangle \otimes |S\rangle = \Psi - \text{remainder}$$

Вероятност за изход $(Y, C, S) : \|\psi_{Y, C, S}\|^2 = 2^{-2n} \frac{\sin^2 K'_c \delta(Y, C, S)}{\sin^2 \delta(Y, C, S)}$

Вероятност за грешка от **remainder**: $\|\text{remainder}\|^2 \approx \frac{R}{2^n}$ ($R = \text{остаток } 2^n / P$)

Анализ на най-вероятния резултат

$$\|\psi_{Y, C, S}\|^2 = 2^{-2n} \frac{\sin^2 K'_c \delta(Y, C, S)}{\sin^2 \delta(Y, C, S)} := 2^{-2n} \left| \sum_{L=0}^{K-1} \exp\left(\frac{2\pi i}{2^n} P \bar{Y}\right)^L \right|^2$$

$$= 2^{-2n} \left| \frac{\exp\left(2\pi i \frac{P}{2^n/\bar{Y}}\right)^K - 1}{\exp\left(2\pi i \frac{P}{2^n/\bar{Y}}\right) - 1} \right|^2 \Rightarrow \delta(Y, C, S) = \pi \frac{P}{2^n/\bar{Y}}$$

≈ цяло число

$$\Rightarrow \max \|\psi_{Y, C, S}\|^2 \sim \frac{1}{P} \quad \text{и се достига когато } 2^n/\bar{Y} \approx \text{делител на } P$$