Квантови алгоритми : концепция и пример

Николай М. Николов

План :

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 Чърч (-Тюринг).

* - 1936 е също и годината на "квантовата лошка" - 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 Чърч"

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

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

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

Simulating Physics with Computers

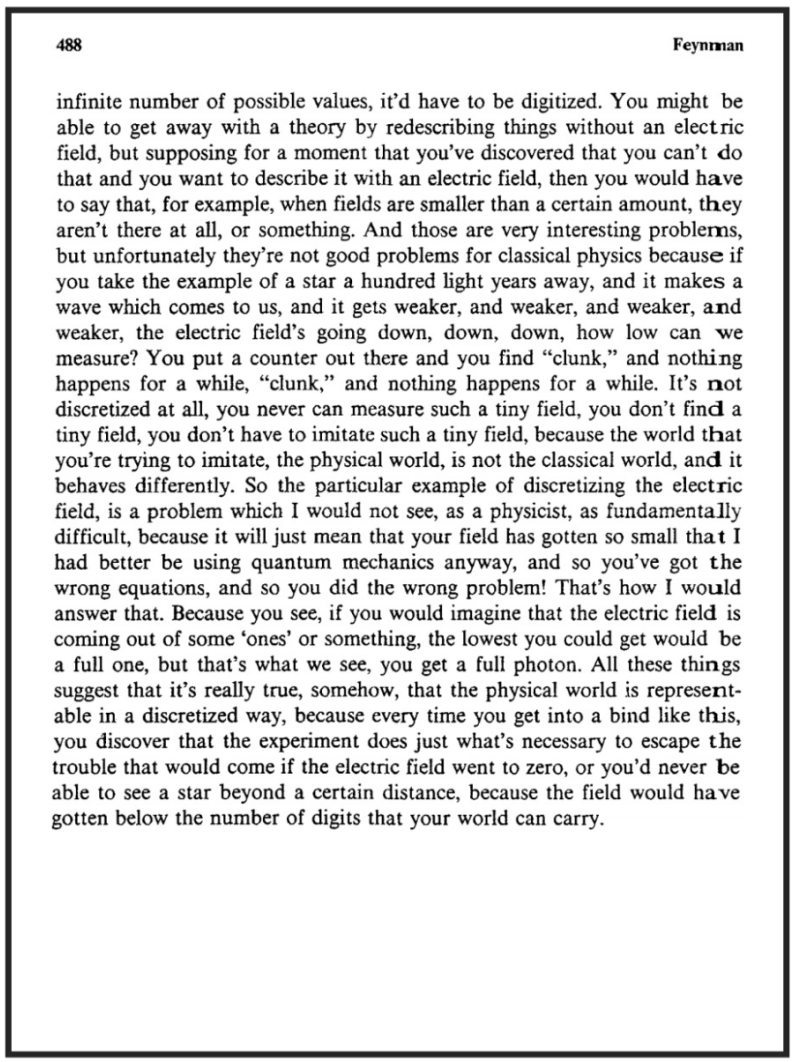
Richard P. Feynman

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 физични процеси на компютър и се достига 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



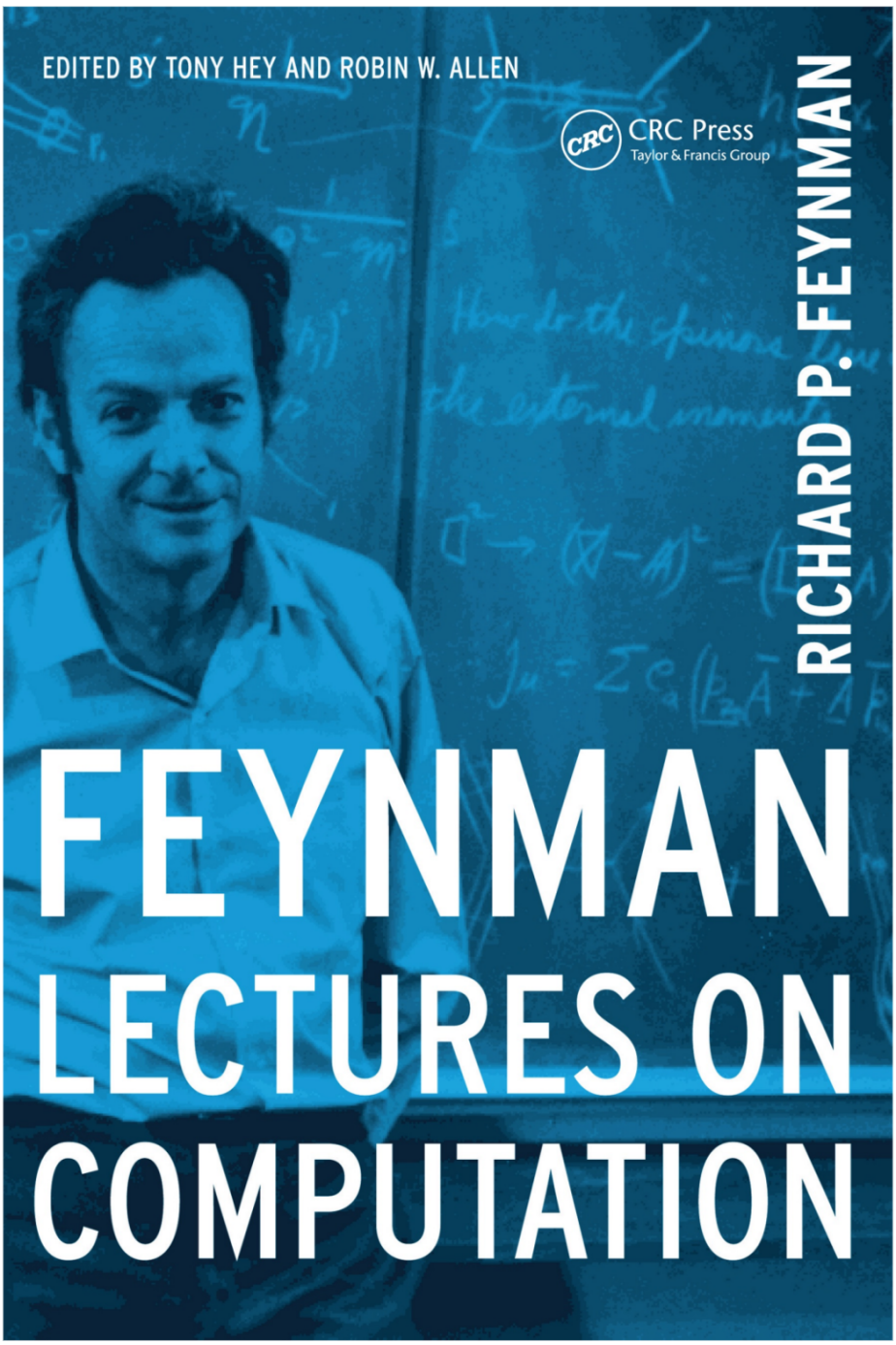
488 Feynman

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 ...

Забележка: може да е полезно за лошуи и информатиуи 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



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

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

$S := A + B \bmod 2$

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

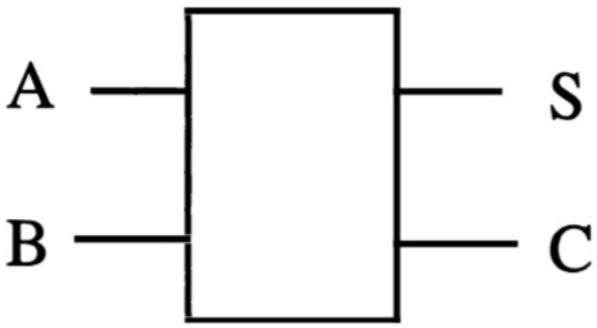


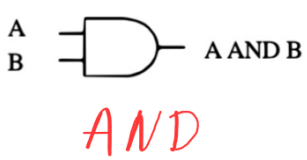
Fig. 2.2 A Black Box Adder

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

Някои означения от курсовете по "цифрова 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

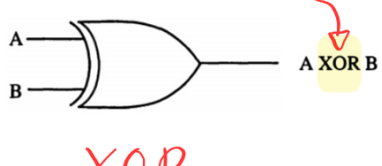


AND

Fig. 2.3 The AND Gate

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

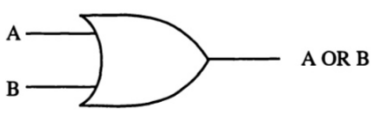
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



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

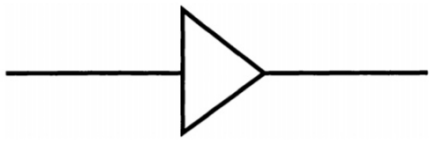
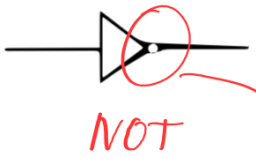


Fig. 2.6 The Identity

A	NOT A
0	1
1	0

Fig. 2.7 The NOT Gate



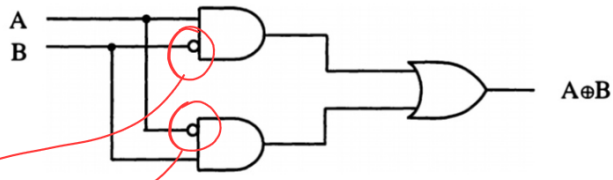


Fig. 2.9 XOR expressed in ANDs and ORs

"0" - указва NOT

Физическа (инженерна) реализация:

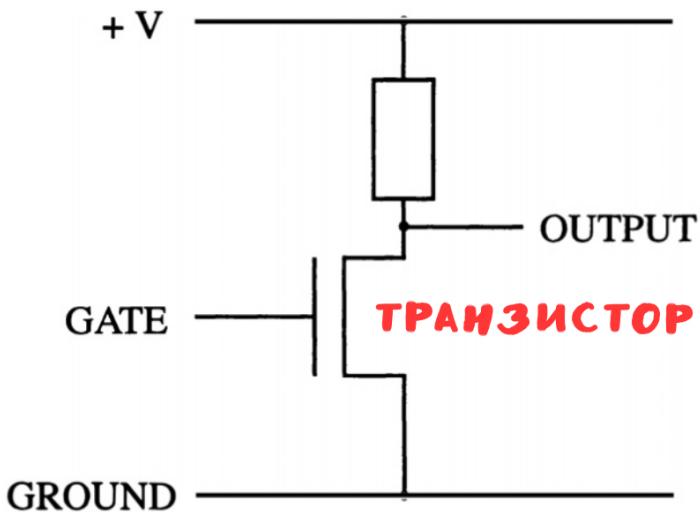
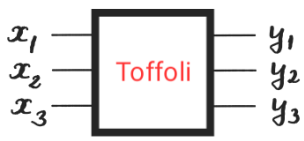


Fig. 2.12 The Transistor Inverter, or NOT Gate

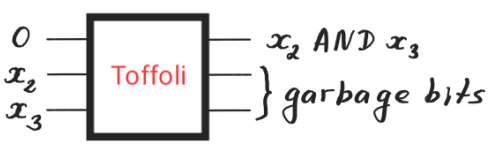
Допълнителна забележка: гейт на Toffoli

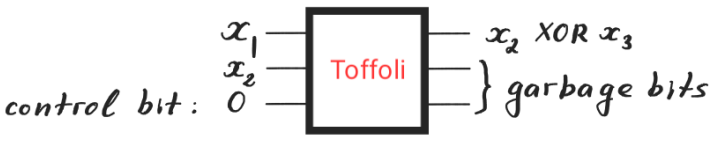


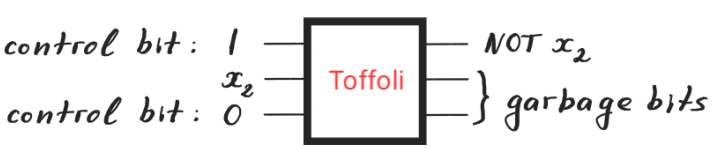
$$\begin{cases} y_1 = x_1 + x_2 x_3 \pmod 2 \\ y_2 = x_2 \\ y_3 = x_3 \end{cases}$$

- вз. еднозн. и обр. $(\mathbb{Z}/2\mathbb{Z})^{\times 3}$

Значение:







} възпроизвеждат базисните логически гейтове

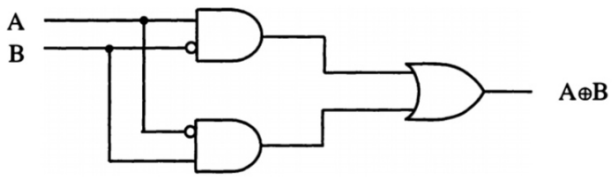
Точното понятие е "uniform circuit family" (съгласувана редица от логически изчислителни вериги):

- редица от ориентирани графи Γ_n, чиито върхове са декорирани от някакво крайно множество логически гейтове $g_1, \dots, g_k$

$$\forall g_j : \{0,1\}^{\times r_j} \longrightarrow \{0,1\}^{\times s_j}$$

Например: Γ_n (за някое n) е:

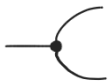
[1996-2018-F], p.26



- от по-горе

Fig. 2.9 XOR expressed in ANDs and ORs

тук базисните върхове включват


,
, 

$$\{0,1\}^{\times 2} \xrightarrow{AND} \{0,1\}, \quad \{0,1\} \xrightarrow{NOT} \{0,1\}, \quad \{0,1\} \xrightarrow{\Delta} \{0,1\}^{\times 2}$$

репликация (диагонално изображение)

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

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

R_n – броя входни линии; S_n – броя изходни линии

според композицията, която се определя от графа.

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

$$R_1 < R_2 < \cdots < R_n < \cdots \,, \quad S_1 < S_2 < \cdots < S_n < \cdots$$

и за $m < n$, то

$$\Gamma_m(x_1, \ldots, x_{R_m}) = (y_1, \ldots, y_{S_m})$$

$$\Rightarrow \quad \Gamma_n(0, \ldots, 0, x_1, \ldots, x_{R_m}) = (0, \ldots 0, y_1, \ldots, y_{S_m})$$

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.

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

Въпрос към специалисти : каква е връзката между "uniform circuit family" и машина на Тюринг ?

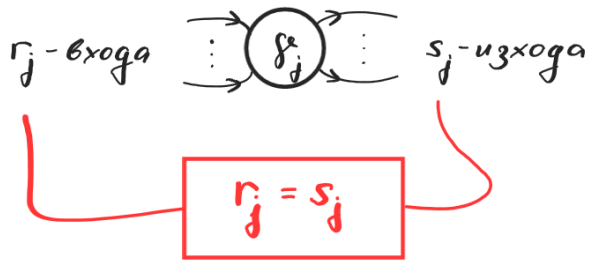
За мен обаче, горния модел на изчисления по-скоро отговаря на електронен калкулатор ("елка"), отколкото на "пълен компютър".

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

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

- Отново, както и по-горе в т.2, стартираме с алгоритмично породена редица от насочени графи, чиито върхове са избрани измежду крайно (или, по-общо, алгоритмично изброимо) множество от върхове.

ИМА ВАЖНО ОГРАНИЧЕНИЕ:



броя на входовете = броя на изходите

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

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

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

$$g_j : \underbrace{(\mathbb{C}^2)^{\otimes r_j}}_{\cong \mathbb{C}^{2^{r_j}}} \longrightarrow \underbrace{(\mathbb{C}^2)^{\otimes r_j}}_{\cong \mathbb{C}^{2^{r_j}}}$$

$\Rightarrow$ 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 $\{e_0, e_1\}$; той е орто-нормиран.
- стандартния базис в $(\mathbb{C}^2)^{\otimes n} \equiv \mathbb{C}^{2^n}$ е $e_{k_{n-1}} \otimes \cdots \otimes e_{k_0} \equiv e_\ell$, за $\ell = k_{n-1} 2^{n-1} + \cdots + k_1 2^1 + k_0 2^0$ - двоично разлагане.

(отново, орто-нормиран)

наричат се също "изчислителни базиси" (computational basis).

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

$$g(e_{k_1} \otimes \cdots \otimes e_{k_r}) =: \sum_{k'_1, \ldots, k'_r = 0,1} g_{k_1, \ldots, k_r ; k'_1, \ldots, k'_r} \; e_{k_1} \otimes \cdots \otimes e_{k_r}$$

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

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

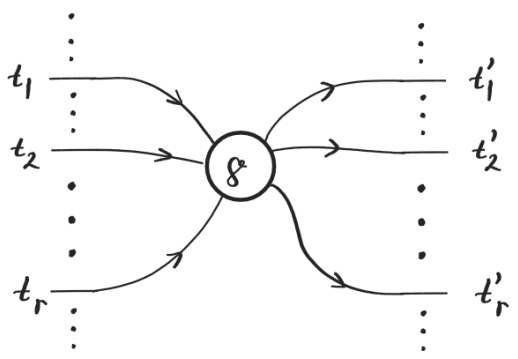
$$(g)_{t_1, \ldots, t_r ; t'_1, \ldots, t'_r} \left(\cdots \otimes e_{k_{t_1}} \otimes \cdots \otimes e_{k_{t_2}} \otimes \cdots \otimes e_{k_{t_r}} \otimes \cdots \right)$$

$$:= \sum_{k'_{t_1}, \ldots, k'_{t_r} = 0,1} g_{k_{t_1}, \ldots, k_{t_r} ; k'_{t_1}, \ldots, k'_{t_r}} \cdots \otimes e_{k'_{t_1}} \otimes \cdots \otimes e_{k'_{t_2}} \otimes \cdots \otimes e_{k'_{t_r}} \otimes \cdots$$

с уговорката, че в многоточията не "пипаме нищо".

Унитарно е, ако и g е унитарно

Графичен израз:



- Веригите могат да се разширят допълнително с действия на пермутации: g^σ

• Условие за "цялостна класичност" (мой израз!)

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

т.е., $\Gamma_n(e_{x_1}\otimes\cdots\otimes e_{x_{R_n}}) = e_{y_1}\otimes\cdots\otimes e_{y_{R_n}}$

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

$$\{0,1\}^{\times R_n} \longrightarrow \{0,1\}^{\times R_n}$$

По-слабо, условие: разглежда се двойна фамилия

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

$$\Gamma_{n,h}(e_{x_1}\otimes\cdots\otimes e_{x_{R_n}}) = e_{y_1}\otimes\cdots\otimes e_{y_{R_n}} + \mathcal{O}_h$$

където $\|\mathcal{O}_h\|^2 = 1 - (\text{Probability for } (y_1,\dots,y_{R_n}))^2 \xrightarrow[h\to\infty]{} 0$

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

$$\{0,1\}^{\times R_n} \longrightarrow \{0,1\}^{\times R_n}$$ да съгласувани.

• Допълнителни отслабвания: -частична дефиниционна област, при условие, че има "бърза" класическа проверка за допустимост на входните данни

- може да има допълнителни битове (входни-контролни и изходни-garbage).

- може недетерминистичен изход: $\sum \Psi_{x_1,\dots,x_{R_n}} e_{x_1}\otimes\cdots\otimes e_{x_{R_n}}$ ($\Psi \in \mathbb{C}$)

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

- може частично междинно измерване по ст.нар. "проекционен постулат" на ф.М.

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 (вертекси) 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 (забавени) класически алгоритми.
Затова, не е ясно дали като цяло ще има подобрение (поне за мен !?)

- Теоремата на Тофоли все пак ни казва, че изискването за обратимост дори на класическо ниво не стеснява класа на изчислимите функции: с цената на добавяне на допълнителни битове (входни - контролни, изходни - дъвадже) 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)
$S_{2^n} \ni \sigma : \{0,1\}^{\times n} \longrightarrow \{0,1\}^{\times n}$ 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 $\hat{M}(e_{x_1} \otimes \cdots \otimes e_{x_n}) := e_{y_1} \otimes \cdots \otimes e_{y_n}$,

  ако $M(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ $(x_j, y_k \in \{0,1\})$

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

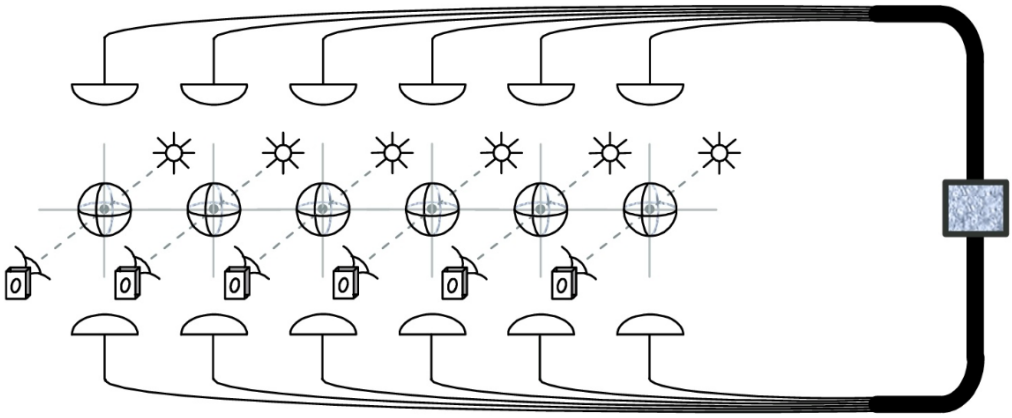
- Наивно, причината за очакваното съществено намаляване на броя елементи в квантовите изчислителни вериги, т.е., ускоряването на изчислението, е в това, че групата на класическите обратими изчисления върху n бита, S_{2^n}, се потапя в безкрайната група $U(2^n)$

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

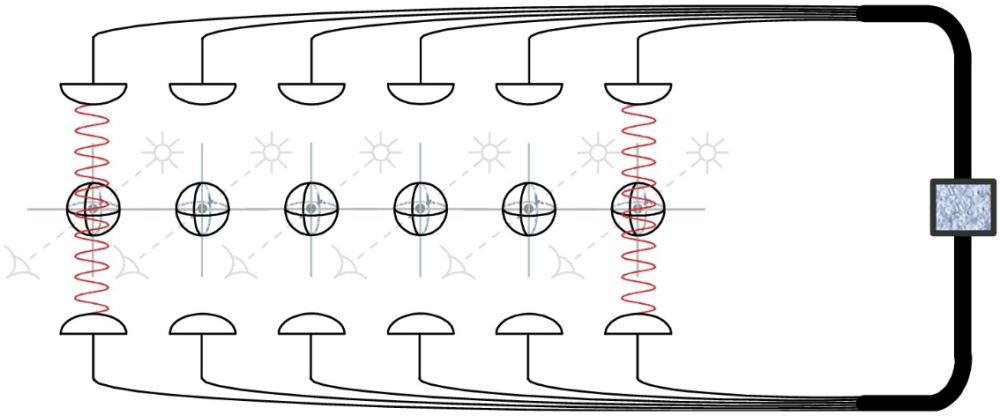
  Това обаче изглежда (поне от пръв поглед) твърде силно, понеже в крайна сметка ние искаме да апроксимираме само $S_{2^n} \subseteq U(2^n)$.

5. Илюстрация на физична реализация

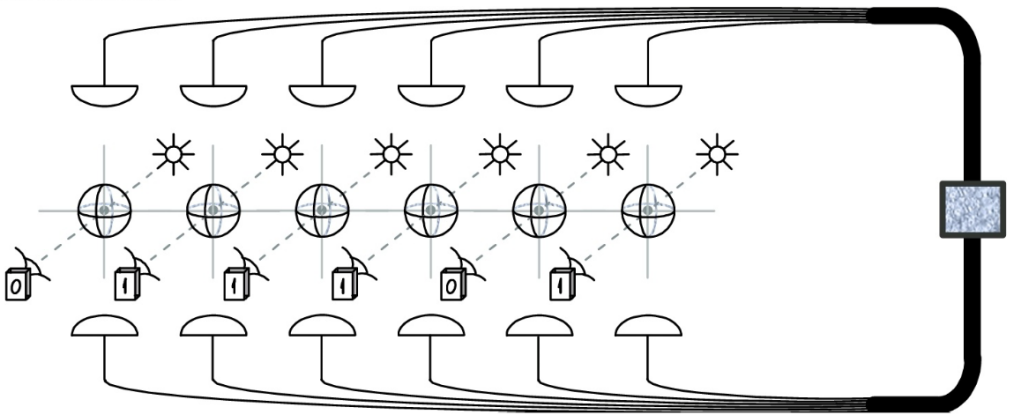
1) Инициализация

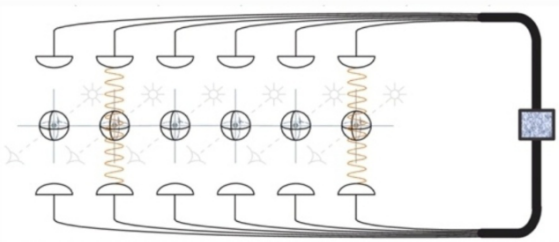


2) Квантово изчисление



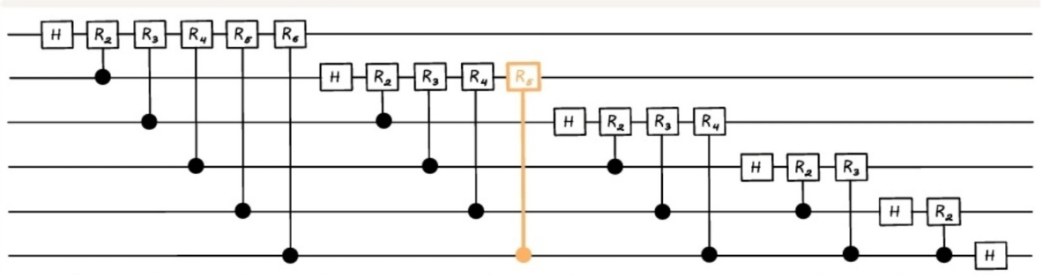
3) Измерване на резултата





Градивните 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 $f: \mathbb{N} \to \mathbb{N}$, която е периодична, $f(X+P) = f(X)$ и алгоритмично изчислима; P = период = ?
Например, $f(X) = Y^X \bmod Z$ (отпук следва намирането на прости делители)

а) Мотиви и цели.

- Всяко изчисление върху n-бита, зависи $N = 2^n$ - възможности.
- Жаргон на физиците:
  Изчислителния процес ще наричаме "бърз", ако времето му има полиномиален ръст спрямо $n = \log N$
  Ще казваме, че е "бавен", ако ръста на времето е полиномален, но спрямо $N = 2^n$, т.е., по n е експоненциален.
- Сред бързите изчисления са събирането, умножаването, деленето, степенуването.
  Предполагаме, че алгоритъма за функция $f(X)$ по-горе е "бърз".
- Известните класически алгоритми за намиране на прости делители и периоди са бавни, понеже са посъществу "бързи проверки" на $N = 2^n$ възможности.
- Квантовите алгоритми за намиране на прости делители им на периоди не са детерминистични и също изискват на финала "бързи проверки". Те обаче генерират специални вероятностни разпределения върху множеството за обръщане, така че за $\forall p \in (0,1)$ да $\exists$ определен полиномиален ръст $T(n)$, 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

[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

б) Базисни гейтове

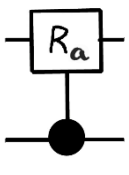
- 

$H e_x = 2^{-1/2}\left(e_0 + (-1)^x e_1\right) = 2^{-1/2} \sum_{y=0,1} \exp\left(2\pi i \frac{xy}{2}\right) e_y$

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

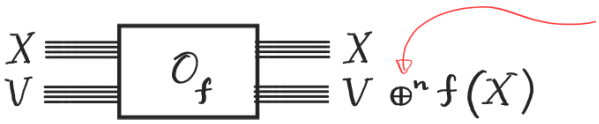
- 

$R_a\, e_x \otimes e_y = \exp\left(2\pi i \frac{xy}{2^a}\right) e_x \otimes e_y$



} алтернативно означение (тип "условна операция")

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

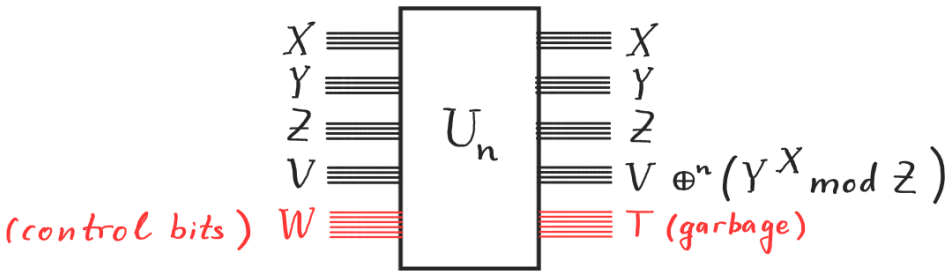


за $X, Y \leq 2^n$.

Ще я пренесем "квантово" със същото означение

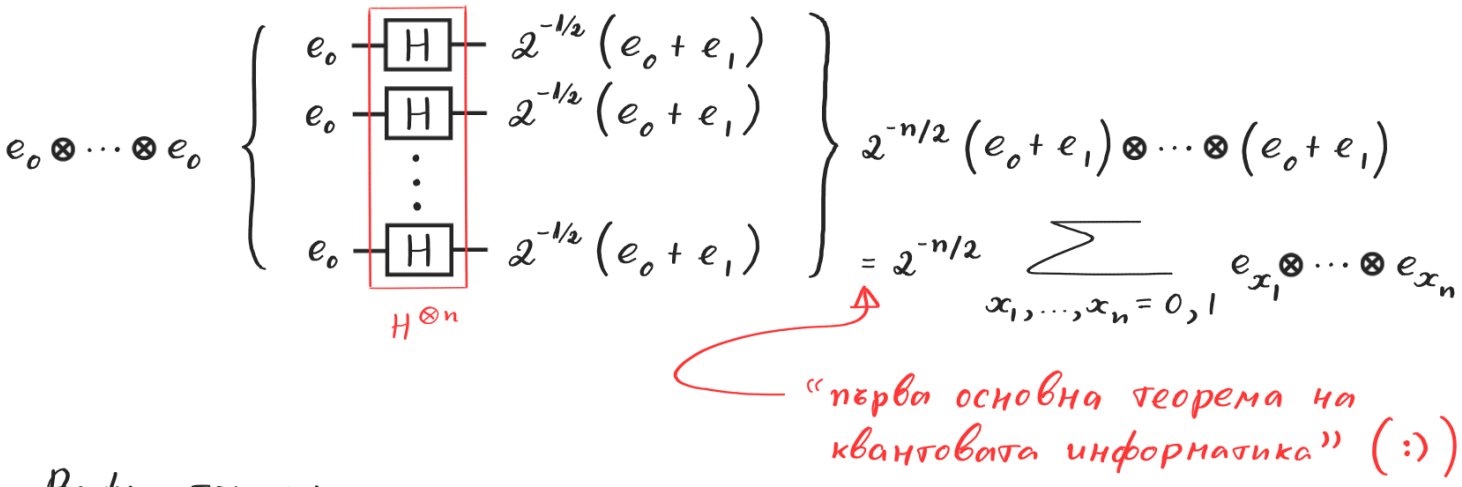
По-такъв начин, най-общо ще построим квантов алгоритъм, чийто "вход" е класически алгоритъм.

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



Трябва да се внимава с остатъчните помощни битове (garbage), които накрая не се чистят: нужно е техните състояния да не се променят при промяна на "оперативните данни" – иначе техните вектори няма да се "сфакторизират".

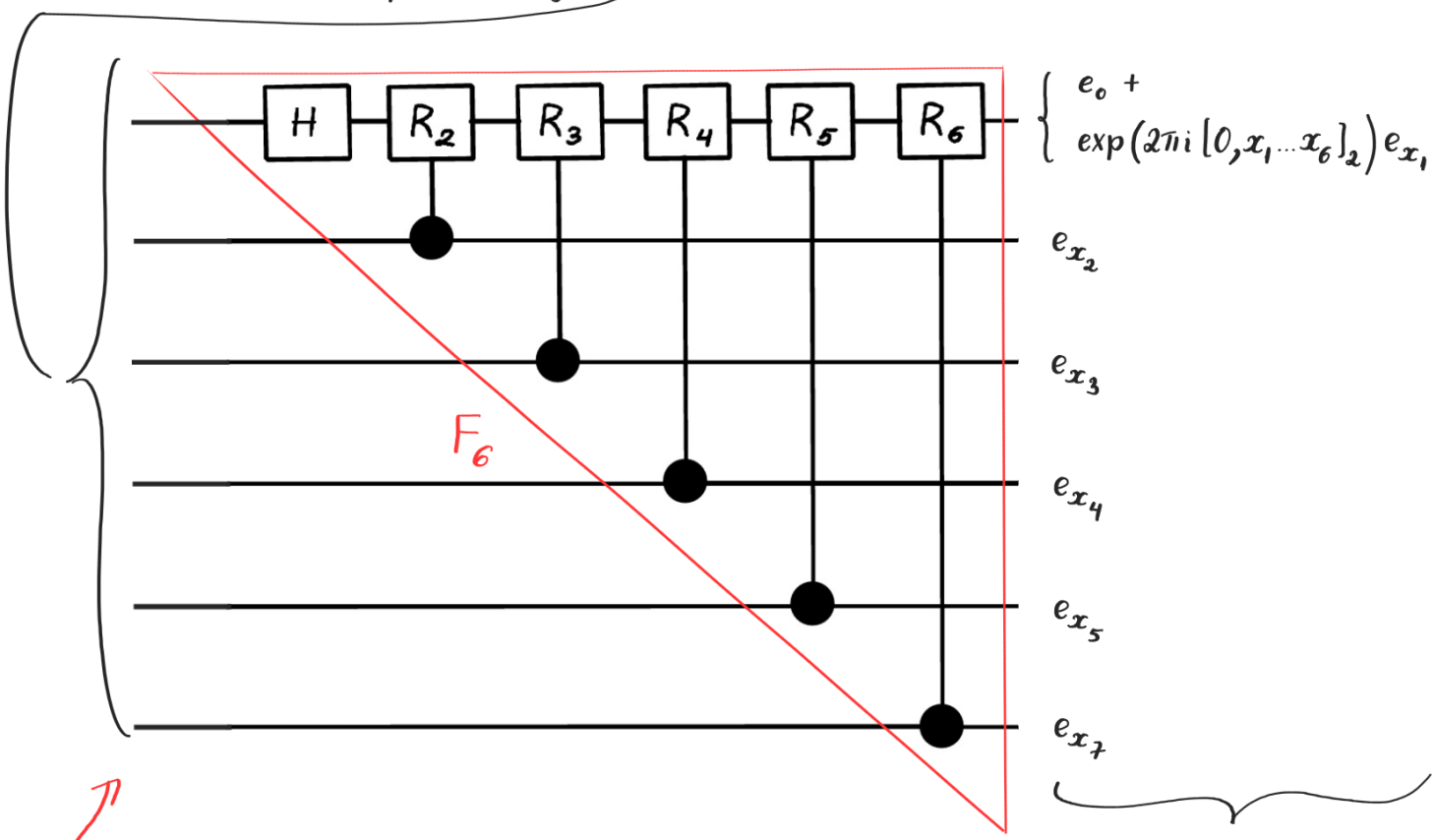
Упражнение 1:



Важна терминология:

"разпространение по линейност" =: "квантов паралелелизъм" (:)

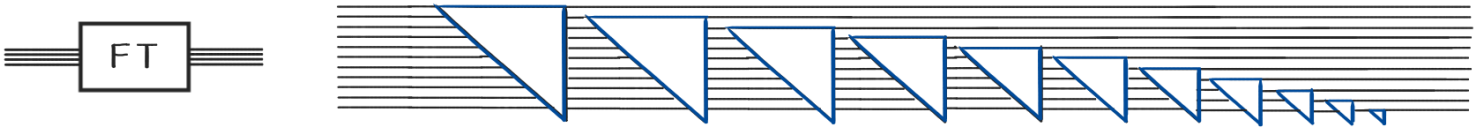
Упражнение 2: $e_{x_1}\otimes\cdots\otimes e_{x_5}$



$$2^{-1/2}\left(e_0 + \exp(2\pi i\,[0,x_1\ldots x_6]_2)\; e_{x_1}\right)\otimes e_{x_2}\otimes\cdots\otimes e_{x_5}$$

"основна лема на квантовата информатика" (:)

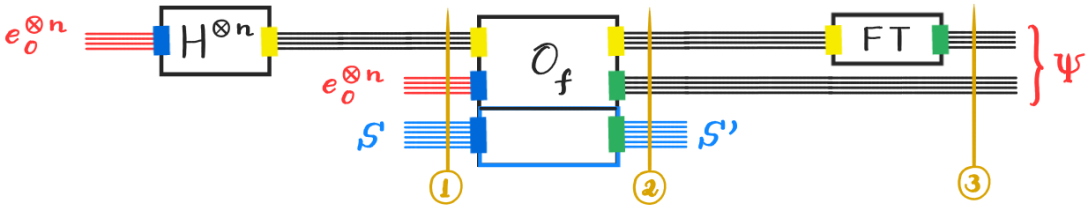
Означаваме:



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

– полиномиална

Тогава



легенда:

- нулеви входни битове
- входни битове за удлата верига
- междинно изчисление
- междинни битове на веригата
- стандартен ред на четене
- изходни битове за удлата верига
- допълнителни помощни битове

Пресмятания

1) $FT(e_{x_1}\otimes\cdots\otimes e_{x_n})$

$$= 2^{-n/2}\bigotimes_{k=0}^{n-1}\left(e_0+\exp\left(2\pi i\,\frac{[x_1\ldots x_n]_2}{2^n}\,2^k\right)e_1\right)$$

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

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

а означим $e_X := e_{x_1}\otimes\cdots\otimes e_{x_n}$, ако $X=[x_1\ldots x_n]_2$

$\overline{X} := [x_n\ldots x_1]_2$

Тогава $FT(e_X)=2^{-n/2}\sum_{Y=0}^{2^n-1}\exp\left(\frac{2\pi i}{2^n}X\overline{Y}\right)e_Y$

2) Пресмятане на Ψ

Предполагаме: $f(X) = f(X')$ за $X < X'$ $\Leftrightarrow$ P дели $X'-X$

① $= H^{\otimes n} e_{\underline{0}} \otimes e_{\underline{0}} \otimes e_S$ — фиксирани, но неотчетени

$\mathcal{O}_f$ $= 2^{-n/2} \sum_{X=0}^{2^n-1} e_X \otimes e_{\underline{0}} \otimes e_S$

② $= 2^{-n/2} \sum_{X=0}^{2^n-1} \mathcal{O}_f(e_X \otimes e_{\underline{0}}) \otimes e_{S'}$

$= 2^{-n/2} \sum_{X=0}^{2^n-1} e_X \otimes e_{f(X)} \otimes e_{S'}$

$= 2^{-n/2} \sum_{C=0}^{F'-1} \sum_{L=0}^{K_C'-1} e_{X_C+LP} \otimes e_C \otimes e_{S'} +$ remainder

самата наредба няма да се използва

кодово за простота (конвенция) $\mathrm{Image}\, f = \{0', 1', \dots, F'-1\}$,

също ще ги предполагаме само имплицитно зададени

$$\begin{cases} X_{k'} := \min f^{-1}(k'), \\ K_C' - 1 := \max\{L = 0,1,\dots \mid X_C + LP < 2^n\} \end{cases}$$

FT

③ $\approx 2^{-n/2} \sum_{C=0}^{F'-1} \sum_{L=0}^{K_C'-1} \mathrm{FT}(e_{X_C+LP}) \otimes e_C \otimes e_{S'}$

$= 2^{-n/2} \sum_{C=0}^{F'-1} \sum_{L=0}^{K_C'-1} 2^{-n/2} \sum_{Y=0}^{2^n-1} \exp\left(\frac{2\pi i}{2^n}(X_C + LP)\overline{Y}\right) e_Y \otimes e_C \otimes e_{S'}$

$= 2^{-n} \sum_{C=0}^{F'-1} \sum_{Y=0}^{2^n-1} \left[\exp\left(\frac{2\pi i}{2^n} X_C \overline{Y}\right) \sum_{L=0}^{K_C'-1} \exp\left(\frac{2\pi i}{2^n} P\overline{Y}\right)^L\right] e_Y \otimes e_C \otimes e_{S'}$

$\Psi_{Y,C,S}$

$= 2^{-n} \sum_{C=0}^{F'-1} \sum_{Y=0}^{2^n-1} \Psi_{Y,C,S}\, e_Y \otimes e_C \otimes e_{S'} = \Psi$ – remainder

Вероятност за изход (Y,C,S): $\|\Psi_{Y,C,S}\|^2 = 2^{-2n} \dfrac{\sin^2 K_C' \delta(Y,C,S)}{\sin^2 \delta(Y,C,S)}$

Вероятност за грешка от remainder: $\|\text{remainder}\|^2 \approx \dfrac{\text{остатък } 2^n/P}{2^n}$

Анализ на най-вероятния резултат

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

$$= 2^{-2n}\left|\frac{\exp\left(2\pi i\,\frac{P}{2^n/\overline{Y}}\right)^{K_C'} - 1}{\exp\left(2\pi i\,\frac{P}{2^n/\overline{Y}}\right) - 1}\right|^2 \quad\Rightarrow\quad \delta(Y,C,S) = \pi\,\frac{P}{\underbrace{2^n/\overline{Y}}_{\approx \text{цяло число}}}$$

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