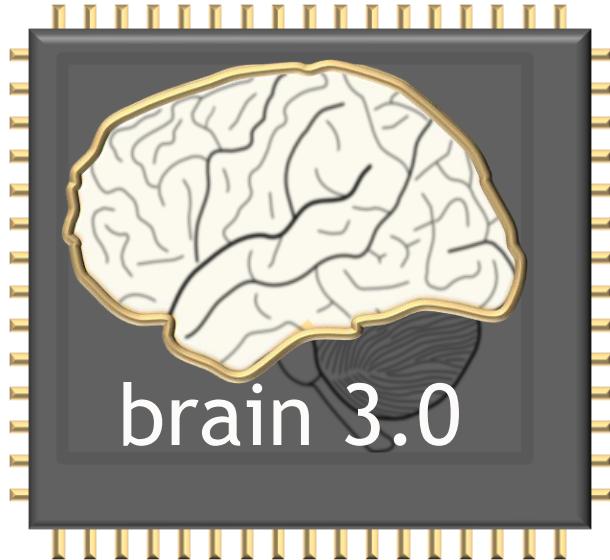


Nevromorfno računalništvo

Neuromorphic computing



Naši možgani so čudoviti stroj

Možgani in računalnik

NEVRONI: prednost v prostoru

Nevroni v človeških možganih tvorijo do 10^5 povezav s svojimi sosedji.

El. moč: 10 W,
 10^{10} nevronov, 10^{13} sinaps,
50000 nevronov/ mm³

EL. VEZJA: prednost v času

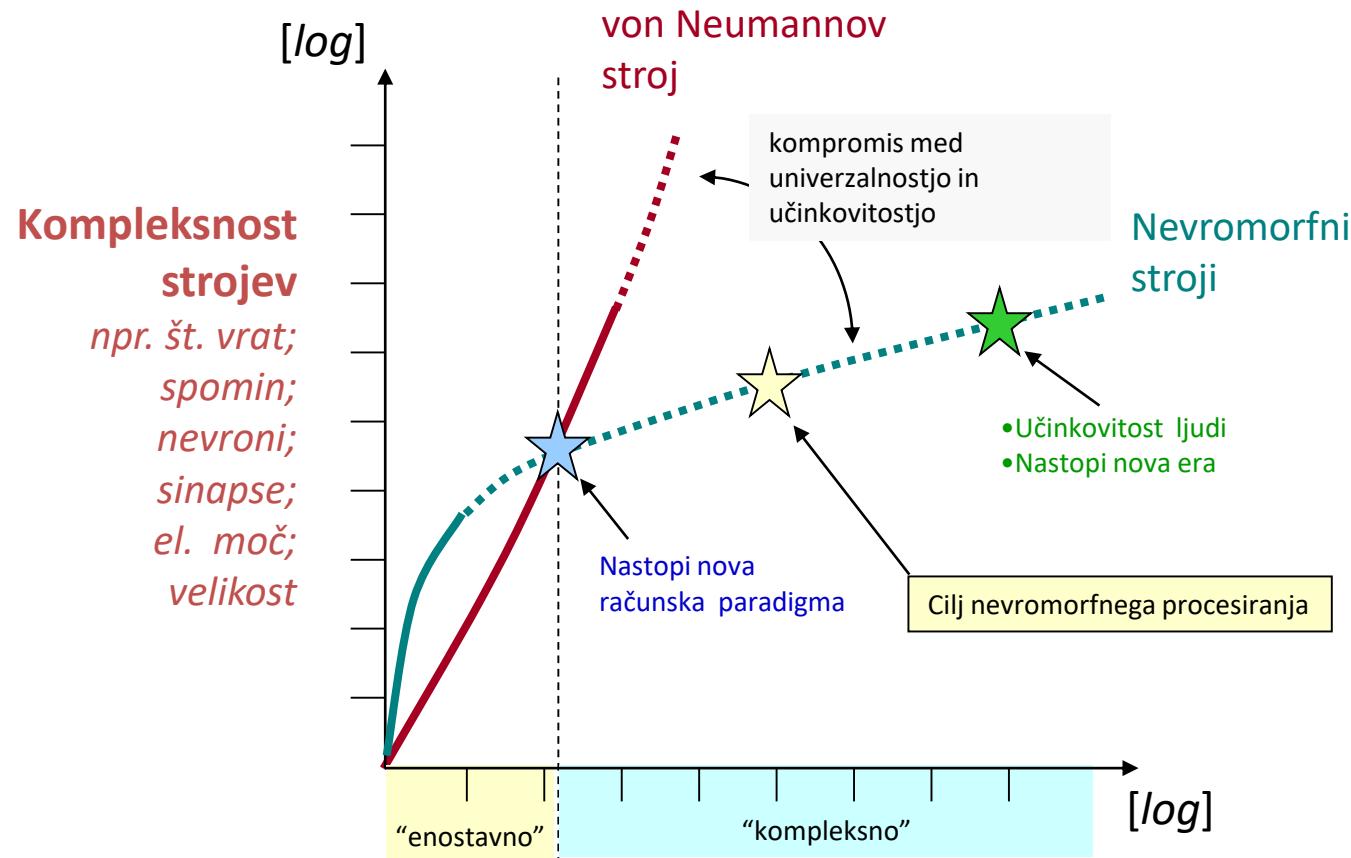
Integrirana vezja opravijo komunikacijski cikel 10^6 x hitreje kot nevroni.

Pentium 4:
El. moč: 40 W,
3 GHz CPE,
 42×10^6 tranzistorjev

Možgani in računalnik

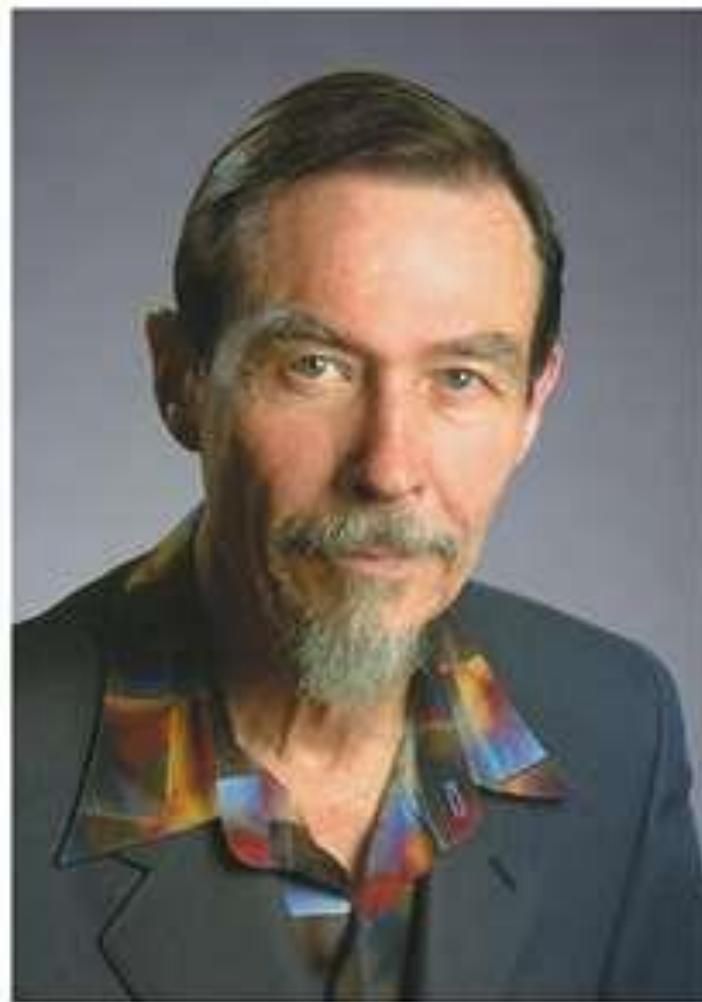
Izzivi:

- V primerjavi z biološkimi sistemi so današnji inteligentni stroji več kot milijonkrat manj učinkoviti v kompleksnih okoljih.
- Inteligentni stroji bodo resnično uporabni šele ko bodo tekmovali s človekom.



Človeški kortex	Simuliran človeški kortex
15 W	10^{10} W
1 liter	4×10^{10} litrov

Kompleksnost okolja
npr. kombinatorika vhodnih podatkov



Carver Andress Mead

Neuromorphic Electronic Systems

CARVER MEAD

Invited Paper

“The fact that we can build devices that implement the same basic operations as those the nervous system uses leads to the inevitable conclusion that we should be able to build entire systems based on the organizing principles used by the nervous system. I will refer to these systems generically as *neuromorphic systems*”.

Carver Mead, 1990

Nevromorfno računanje

Uporaba bioloških principov

=> konstrukcija bolj učinkovitih strojev

Konstrukcija nevromorfnih strojev

=> pridobljeno znanje o bioloških principih

Projekt FACETS



MoNETA: A Mind Made from Memristors

<https://spectrum.ieee.org/robotics/artificial-intelligence/moneta-a-mind-made-from-memristors>

Nevromorfne arhitekture: “The great BRAIN race”

From BrainScales to Human Brain Project: Neuromorphic Computing Coming of Age

<https://www.youtube.com/watch?v=g-ybKtY1quU>

Growing Number of NM Projects in the EU and the US

The Five Complementary Approaches to Neuromorphic Computing

- Commodity microprocessors (SpiNNaker, HBP)
 - Custom fully digital (IBM Almaden)
 - Custom Mixed-Signal (BrainScaleS, HBP)
 - Custom subthreshold analog cells (Stanford, ETHZ)
 - Custom Hybrid (Qualcomm)
- Soft-binary-modelHard-binary-model
- Physical model (accelerated)Physical model (real time)
- Hybrid NM-traditional

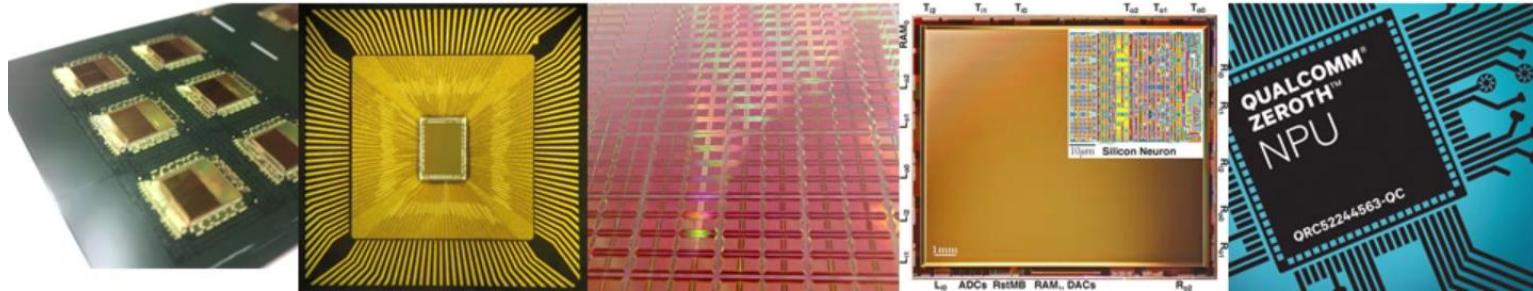
Anything in common ?

Massively parallel

Asynchronous communication

Configurability

COMPLEMENTARITY OF APPROACHES ESSENTIAL !



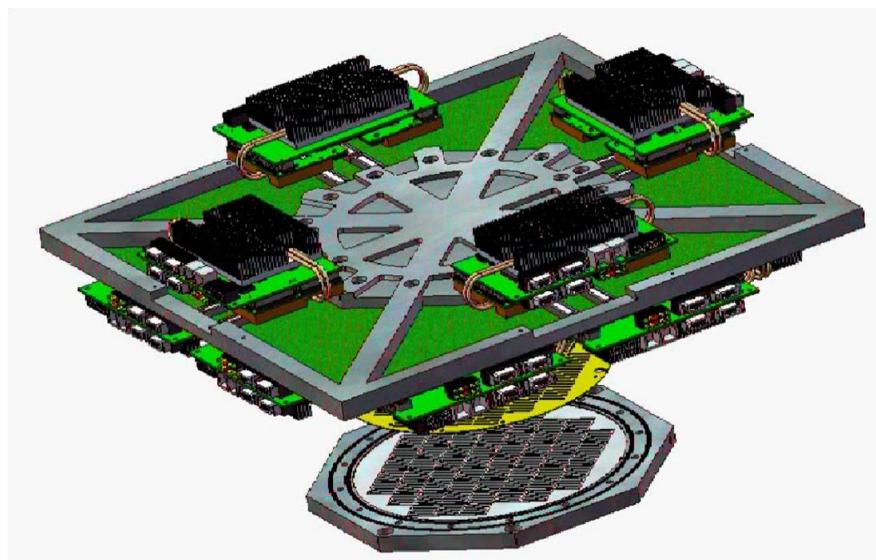
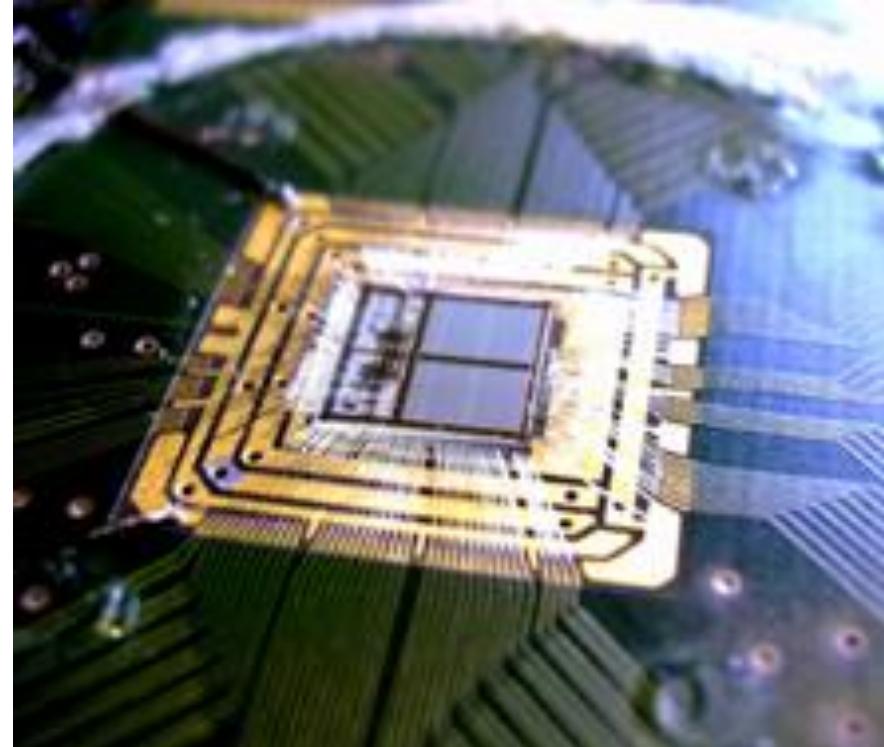
Projekt FACETS

Fast Analog Computing with Emergent Transient States (FACETS)

- mednarodni projekt znanstvenikov in inženirjev, financiran s strani EU.
- Zadnji razvit čip vsebuje 200,000 nevronov (t.j. nevronske vezije), ki so med sabo povezani s 50 milijoni sinaps.
- Projekt zaključen 2010
- Delo se nadaljuje v okviru projekta BrainScale



<http://brainscales.kip.uni-heidelberg.de/>



<http://facets.kip.uni-heidelberg.de/index.html>

Projekt FACETS

<http://facets.kip.uni-heidelberg.de/index.html>

Razvita strojna oprema: zelo veliko **VLSI nevronsko vezje**, ki emulira signifikanten predel človeškega korteksa. Vezje temelji na novi računski paradigm in izkorišča masovne **parallelne, kompleksne in dinamične povezave**, takšne kot jih zasledimo v človeških možganih.

Omenjena računska paradigma temelji na **podrobnih teoretičnih modelih strukture** (vzorci povezovanja, raznolikost nevronov) in **funkcionalnosti** (dinamična stanja aktivnosti, plastičnost) **človeškega korteksa**.

Strojna oprema izkazuje visoko stopnjo **skalabilnosti in nastavljivosti** in dovoljuje eksperimentalne študije nevronskega obnašanja v različnih časovnih in prostorskih merilih (od posameznih proženj nevrona do plastičnosti nevronskih vezij in dolgoročnega učenja). **Namenjena je predvsem raziskovanju v nevroznanosti in razvoju umetnih sistemov, ki bi odpravili potrebo po invazivnih študijah in vivo.**

http://facets.kip.uni-heidelberg.de/images/4/48/Public--FACETS_15879_Summary-flyer.pdf

Human Brain Project

<https://www.humanbrainproject.eu>

BrainScaleS

- Fizična analogno-digitalna emulacija nevronov, sinaps in plastičnosti živčevja z digitalnimi povezavami (ASIC implementacija)
- Vsak nevron ima adaptivno eksponentno dinamiko ([AdEx](#) neuron dynamics)
- Sinapsa: 4-bitna ločljivost in princip delovanja STDP ([spike-timing-dependent plasticity](#))
- Prilagodljiva topologija mreže
- 10.000x hitrejša od bioloških nevronskeih mrež

SpiNNaker

- Večjederna ARM arhitektura
- 30.000 čipov z 18 jedri in 128 MB pomnilnika: skupaj 500.000 jeder

BrainScaleS

<https://electronicvisions.github.io/hbp-sp9-guidebook/pm/pm.html>

Vsak modul ima 384 medsebojno povezanih čipov **HICANNs**. Vsak izmed njih ima 114.688 programabilnih sinaps in 512 nevronov (skupaj 44×10^6 sinaps in 196.608 nevronov na modul).

Programiranje v Python-u, z orodjem za mapiranje nevronov [Marocco](#)

HBP Neuromorphic Computing Platform Guidebook

<https://electronicvisions.github.io/hbp-sp9-guidebook/index.html>

BrainScaleS

https://electronicvisions.github.io/hbp-sp9-guidebook/pm/using_pm_newflow.html#details-of-the-software-stack

```
neuron_count = 1 # size of the Population we will create

# Set the neuron model class
neuron_model = pynn.EIF_cond_exp_isfa_ista # an Adaptive Exponential I&F Neuron

neuron_parameters = {
    'a' : 4.0,      # adaptation variable a in ns
    'b' : 0.0805,   # adaptation variable b in pA
    'cm' : 0.281,   # membrane capacitance nF
    'delta_T' : 1.0, # delta_T from Adex mod in mV, determines the sharpness of spike initiation
    'e_rev_E' : 0.0, # excitatory reversal potential in mV
    'e_rev_I' : -80.0, # inhibitory reversal potential in mV
    'i_offset' : 0.0, # offset current
    'tau_m' : 9.3667, # membrane time constant
    'tau_refrac' : 0.2, # absolute refractory period
    'tau_syn_E' : 20.0, # excitatory synaptic time constant
    'tau_syn_I' : 20.0, # inhibitory synaptic time constant
    'tau_w' : 144.0, # adaptation time constant
    'v_reset' : -70.6, # reset potential in mV
    'v_rest' : -70.6, # resting potential in mV
    'v_spike' : -40.0, # spike detection voltage in mV
    'v_thresh' : -50.4, # spike initiation threshold voltage in mV
}

# We create a Population with 1 neuron of our neuron model
N1 = pynn.Population(size=neuron_count, cellclass=neuron_model, cellparams=neuron_parameters)
```

```
#!/usr/bin/env python

import pyhmf as pynn
import Coordinate as C
from pymarocco import PyMarocco, Defects
from pymarocco.results import Marocco

import pylogging
for domain in ["Calibtic", "marocco"]:
    pylogging.set_loglevel(pylogging.get(domain), pylogging.LogLevel.INFO)

marocco = PyMarocco()
marocco.calib_backend = PyMarocco.CalibBackend.Default
marocco.defects.backend = Defects.Backend.None
marocco.persist = "results.bin"
pynn.setup(marocco = marocco)

pop = pynn.Population(1, pynn.IF_cond_exp)

marocco.manual_placement.on_hicann(pop, C.HICANNOOnWafer(C.X(5), C.Y(5)), 4)

pynn.run(10)
pynn.end()

results = Marocco.from_file(marocco.persist)

for neuron in pop:
    for item in results.placement.find(neuron):
        for denmem in item.logical_neuron():
            print denmem
```

DARPA

SYNAPSE:

<https://www.darpa.mil/news-events/2014-08-07>

μBRAIN:

<https://www.darpa.mil/program/microbrain>

AI Next Campaign:

<https://www.darpa.mil/work-with-us/ai-next-campaign>

Strojna podpora umetni inteligenci:

Loihi – Intel

Loihi 1 - <https://en.wikichip.org/wiki/intel/loihi>

Loihi 2 - <https://www.intel.com/content/www/us/en/newsroom/news/intel-unveils-neuromorphic-loihi-2-lava-software.html>

<https://download.intel.com/newsroom/2021/new-technologies/neuromorphic-computing-loihi-2-brief.pdf>

Lava – <https://lava-nc.org/>
<https://github.com/lava-nc/lava>

Neural Fields: http://www.scholarpedia.org/article/Neural_fields

Strojna podpora umetni inteligenco na mobilnih platformah:

Apple - Neural engine: https://apple.fandom.com/wiki/Neural_Engine

„...a group of specialized cores functioning as a neural processing unit (NPU) dedicated to the acceleration of artificial intelligence operations and machine learning tasks...“

Google - Tensor Processing Unit: <https://cloud.google.com/tpu/docs/tpus>

„...A tensor processing unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning...“

Samsung - Neural Processing Unit (NPU):

<https://semiconductor.samsung.com/emea/insights/topic/ai/>

„...NPU is a processor that is optimized for deep learning algorithm computation, designed to efficiently process thousands of these computations simultaneously...“

Literatura

- [1] Neuromorphic, <<http://en.wikipedia.org/wiki/Neuromorphic>>.
- [2] Hammerstrom, D. "A Survey of Bio-Inspired and Other Alternative Architectures," in Waser, Rainer (ed.) Nanotechnology. Volume 4: Information technology II. Weinheim: Wiley-VCH, pp. 251-282, 2006.
- [3] Carver Mead, <http://en.wikipedia.org/wiki/Carver_Mead>
- [4] Holler, M., et al. "*An Electrically Trainable Artificial Neural Network (ETANN) with 10240 "Floating Gate" Synapses,*" *International Joint Conference on Neural Networks, 1989.*
- [5] Nestor, I., *Ni1000 Recognition Accelerator - Data Sheet, 1-7, 1996.*
- [6] Ramacher, U. et al. "SYNAPSE-1: a high-speed general purpose parallel neurocomputer system," *IPPS (774-781).* 1995.
- [7] R. Serrano-Gotarredona, T. et al. "A Neuromorphic Cortical Layer Microchip for Spike Based Event Processing Vision Systems," *IEEE Trans. on Circuits and Systems, Part-I.* Vol. 53, No. 12, pp. 2548-2566, December 2006.
- [8] Serrano-Gotarredona, R., et al. "AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems," *Advances in Neural Information Processing Systems (NIPS), 18:* 1217-1224, Dec, Y. Weiss and B. Schölkopf and J. Platt (Eds.), MIT Press, 2005
- [9] Brains in Silicon,<<http://www.stanford.edu/group/brainsinsilicon/index.html>>.
- [10] FACETS: Fast Analog Computing with Emergent Transient States, <<http://facets.kip.uni-heidelberg.de/index.html>>.
- [11] Graham-Rowe, D. "Building a Brain on a Silicon Chip," in *Technology Review*, March 25, 2009. [Online]. Available: <<http://www.technologyreview.com/computing/22339/page1/>>. [Accessed March 28, 2009].
- [12] C. Torres-Huitzil, et. al. "On-chip Visual Perception of Motion: A Bio-inspired Connectionist Model on FPGA," *Neural Networks Journal, 18(5-6):557-565,* 2005.

Umetne nevronske mreže

(Artificial Neural Networks)

Knjiga: Martin T. Hagan, Howard B. Demuth, Mark H. Beale: *Neural Network Design*, 2002 <https://hagan.okstate.edu/NNDesign.pdf>

Algoritmi učenje NM (*learning rule*)

postopek spreminjanja uteži (*weights*) in praga (*bias*).

Nadzorovano učenje (supervised learning)

Učno pravilo je določeno z učno množico vhodnih in izhodnih vrednosti :

$$\{\mathbf{p}_1, \mathbf{t}_1\} \{\mathbf{p}_2, \mathbf{t}_2\} \dots \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

kjer je \mathbf{p} vektor vseh vhodov v nevronske mreže in \mathbf{t} vektor ustreznih pravilnih izhodov (**target**).

Vhodni vektorji postopno vstopajo v nevronske mreže, njeni izhodi pa se primerjajo s pričakovanimi izhodi \mathbf{t} . Uteži in pragovi se nastavijo tako da se minimizira napaka med izhodi nevronske mreže in pričakovanimi izhodi \mathbf{t} .

Učenje z ojačitvijo (reinforcement learning)

je podobno nadzorovanemu učenju, le da nimamo podanih pričakovanih izhodnih vrednosti \mathbf{t} , temveč je dana cenitvena funkcija delovanja nevronske mreže. Ta cenitvena funkcija tipično meri delovanje nevronske mreže preko več različnih vhodov nevronske mreže (mehanizem nagrajevanja in kaznovanja skozi daljše časovno obdobje).

Nenadzorovano učenje (unsupervised learning)

Uteži in pragovi se spreminjajo samo glede na dane vhode nevronske mreže. Pričakovane izhodne vrednosti niso podane. Večina nevronske mreže s tem učenjem izvaja neke vrste gručenje (*clustering*). Naučijo se kategorizirati vhodne vzorce v končno število razredov.

Perceptron

- Odločitvena meja je premica (hiper-ravnina v večdimenzionalnem vhodnem prostoru, če je število vhodov p večje od 2):

$$\mathbf{w}^T \mathbf{p} + b = 0 \quad (1)$$

- Vse točke (vhodi \mathbf{p}) na odločitveni meji imajo isto vrednost skalarnega produkta (1), t.j. 0. To pomeni, da imajo vsi ti vhodi isto projekcijo na vektor uteži \mathbf{w} in ležijo na ravnini, ki je **ortogonalna na vektor uteži \mathbf{w}** .
- Katerikoli vhodni vektor nad odločitveno mejo ima skalarni produkt (1) večji od 0, vektorji pod odločitveno mejo pa imajo skalarni produkt (1) manjši od 0.
- Torej bo vektor uteži \mathbf{w} , narisani v prostoru vhodnih vektorjev \mathbf{p} , vedno **kazal proti vhodnemu področju, kjer je izhod nevronске mreže pozitiven**.
- Vrednost praga se običajno nastavi za tem, ko so se nastavile uteži (t.j. za rotacijo oz. poravnavo vektorja uteži \mathbf{w}), izbrana pa je tako, da je izpolnjena enačba meje odločitve (1).

Perceptron: povzetek učnega algoritma

- napaka $e = t - a$
- **splošno učno pravilo** (neglede na dimenzijo vektorja \mathbf{p}):

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} + e \mathbf{p}_i = \mathbf{w}^{\text{old}} + (t - a) \mathbf{p}_i$$

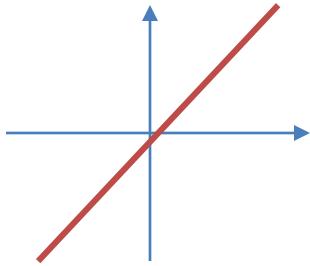
$$b^{\text{new}} = b^{\text{old}} + e$$

- To pravilo skonvergira v končnem številu korakov, če je le zastavljen problem linearно rešljiv (t.j. če so dani vhodni vektorji \mathbf{p} učne množice linearno ločljivi).

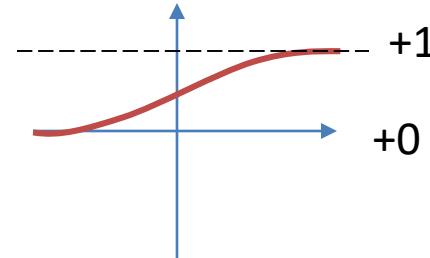
Dokaz: glej Poglavlje 4 knjige M.T. Hagan, H.B. Demuth, M.H. Beale: *Neural Network Design*, 2002.

Splošen nevron

odvodljiva aktivacijska funkcija



linearna aktivacijska funkcija
(žurka za osamelce – „daj muzko ful na glas!“)



sigmoidna aktivacijska funkcija
(osamelci ne morejo bistveno preglasiti ostalih)

Zamisel: Če je aktivacijska funkcija odvedljiva, lahko nevron učimo s pomočjo **gradientne optimizacije njegovih uteži in praga**.

Vse kar potrebujemo je **kriterijska oz. cenitvena funkcija!**

Postopek najmanjših kvadratičnih pogreškov LMS ali Widrow-Hoff-ov učni algoritem

1. Učna množica

$$\{\mathbf{p}_1, \mathbf{t}_1\} \{\mathbf{p}_2, \mathbf{t}_2\} \dots \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

2. Cenitvena funkcija: kvadratični pogreški (Mean-Square-Error - MSE):

$$F(e) = E(e^2) = E((\mathbf{t}_Q - \mathbf{a})^2) = E((\mathbf{t}_Q - \mathbf{w}^T \mathbf{p}_i - b)^2)$$

kjer je E matematično upanje.

3. Gradientna optimizacija vsake uteži w_{ji} :

$$w_{ji}^{new} = w_{ji}^{old} - \alpha \Delta F(e)$$

α – stopnja učenja

$\Delta F(e)$ – gradient cenitvene funkcije

$$\Delta F(e) = \partial F(e) / \partial w_{ji} = \partial F(e) / \partial e \cdot \partial e / \partial w_{ji} = 2E(t_i - a_i) \cdot -p_{ji}$$

4. Gradientna optimizacija celotnega vektorja uteži \mathbf{w} :

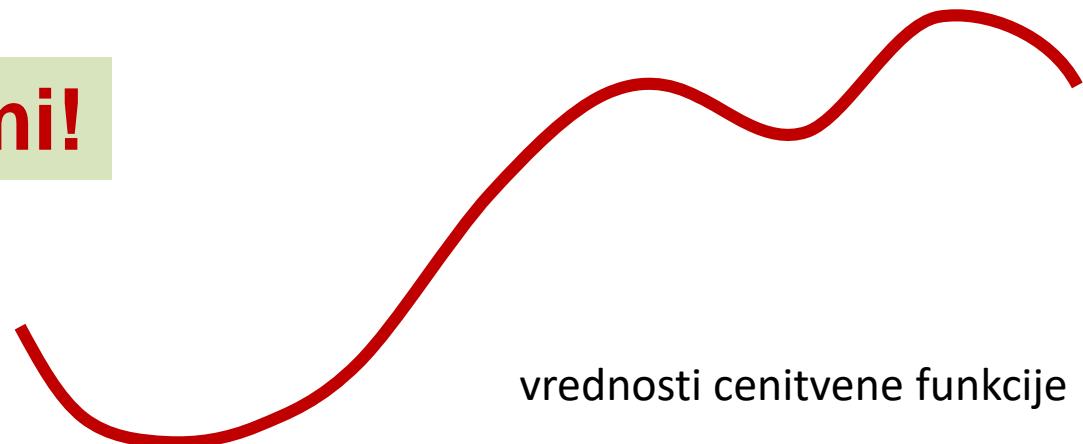
$$\mathbf{w}^{new} = \mathbf{w}^{old} + \alpha E(t_i - a_i) \cdot \mathbf{p}_i$$

Učenje nevronske mreže

- **Vzvratno učenje (Backpropagation)**

- potrebuje učno množico (pare vhodov in izhodov)
- prične z majhnimi naključnimi utežmi
- uporablja cenitveno funkcijo (npr. MSE) za prilagajanje posameznih uteži (nadzorovano učenje)
 - Gradientna optimizacija je iskanje optimuma po površju cenitvene funkcije

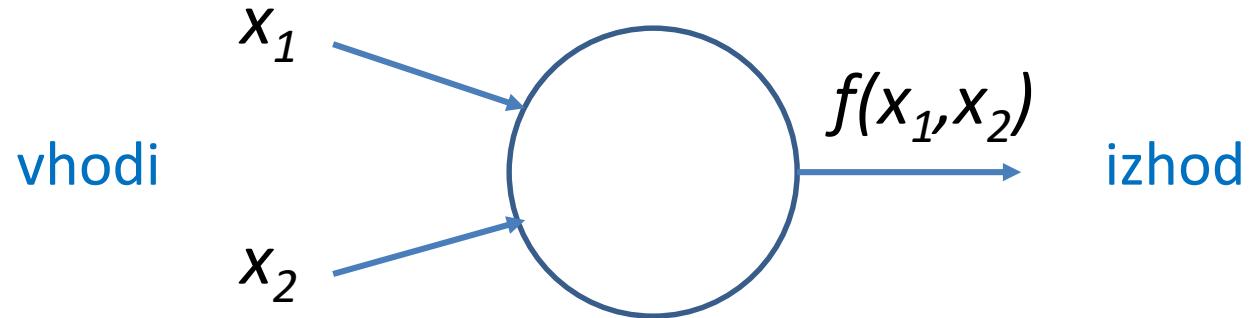
Lokalni optimumi!



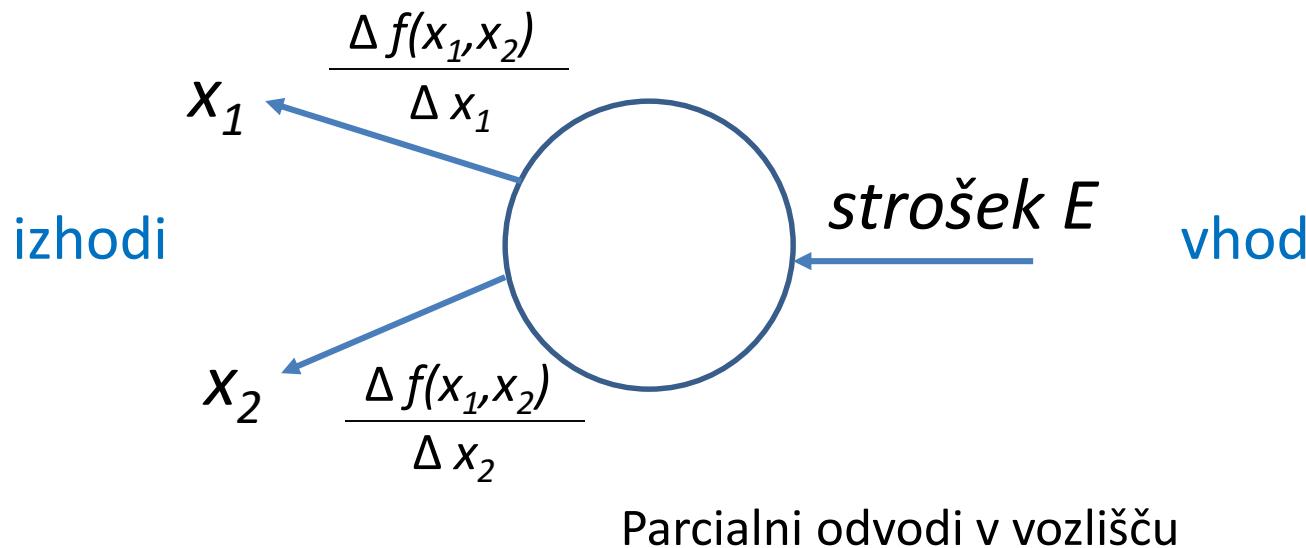
Vzvratno učenje – grafična predstavitev

nevron z aktivacijsko funkcijo f in njenim odvodom Δf

računanje naprej (feed forward)



in vzvratno učenje (backpropagation)



Hebbova teorija

"cells that fire together, wire together"

Hebbovo učenje: asociativno učenje, pri katerem sočasna aktivacija nevronov vodi v povečano sinaptično aktivnost (komunikacijo) med temi nevroni.

S stališča umetnih nevronov in umetnih nevronskeih mrež je Hebbov princip mogoče uporabiti kot metodo za nastavljanje uteži povezav med posameznimi nevroni. **Utež povezave med dvema nevronoma se poveča, če oba nevrona prožita istočasno** (t.j. pri istem vhodnem vektorju oz. vzorcu). Utež povezave se zmanjša, če nevrona prožita ob različnih časih. Tisti nevroni, ki dajejo isti odziv na vhodne vzorce pridobijo močne pozitivne povezave (eksitacija). **Tisti, ki prožijo asinhrono pridobijo negativno obtežene povezave (inhibicija).** Opisani mehanizmi omogočajo **asociacije** (delni vhodni vzorec bo vzbudil odziv vseh močno povezanih nevronov).



Donald O. Hebb

Hebbova teorija

"cells that fire together, wire together"

$$w_{ij} = x_i x_j$$

kjer je w_{ij} utež povezave med nevronom j in nevronom i in x_i vhod nevrona i

V primeru več vhodnih vrednosti:

$$w_{ij} = \frac{1}{p} \sum_{k=1}^p x_i^k x_j^k$$

kjer je p število učnih vzorcev.

Unsupervised
learning!

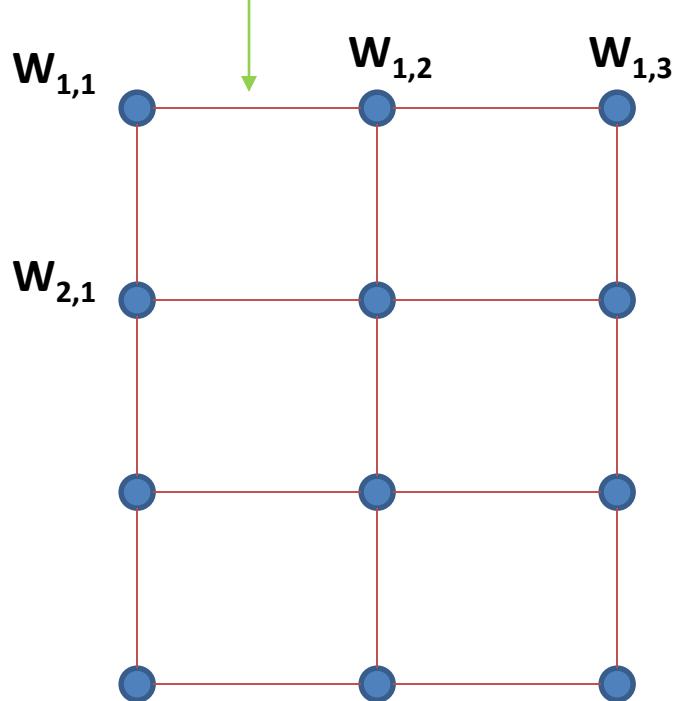
Pogojno učenje ([pogojni refleks](#)):

- **Ivan Petrovič Pavlov** & njegovi psi: pozvoni z zvoncem kadarkoli hraniš pse. Čez čas se bodo psi slinili kadarkoli pozvoniš z zvoncem (z ali brez hrane).
- 2. svetovna vojna: ruski psi z bombami & hrana pod nemškimi tanki.

Samoorganizirajoče nevronske mreže (SOM)

pred učenjem

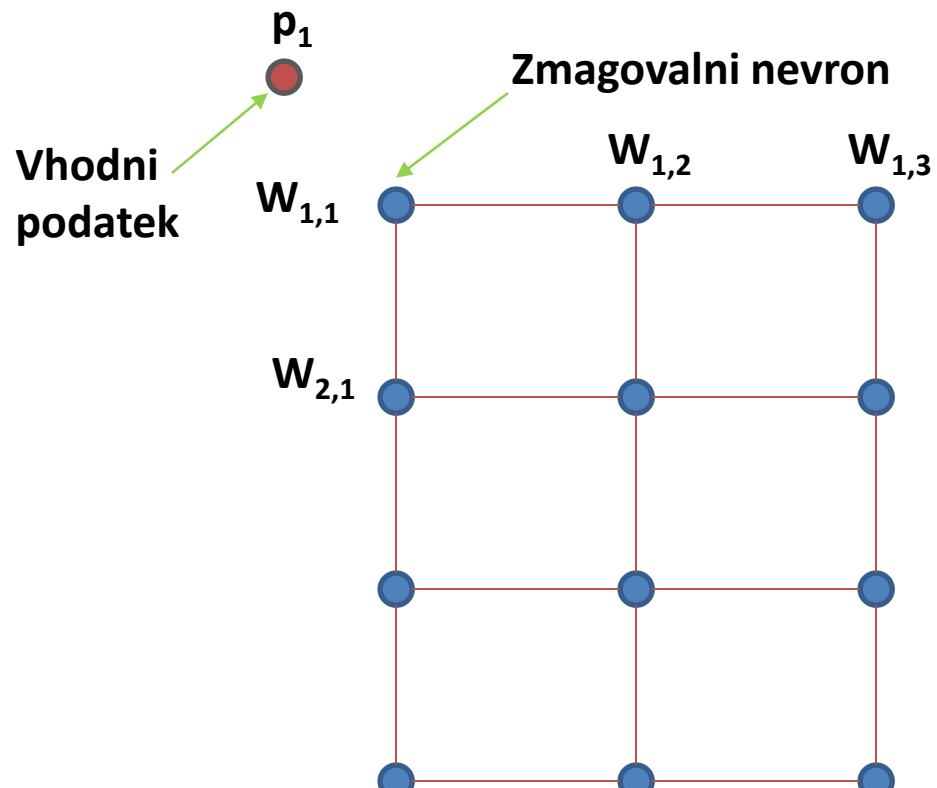
Sosedskost pred učenjem (to niso fizične povezave med nevroni)



Skupni vektorski prostor
uteži nevronov w in tudi
vhodnih vektorjev p

med učenjem

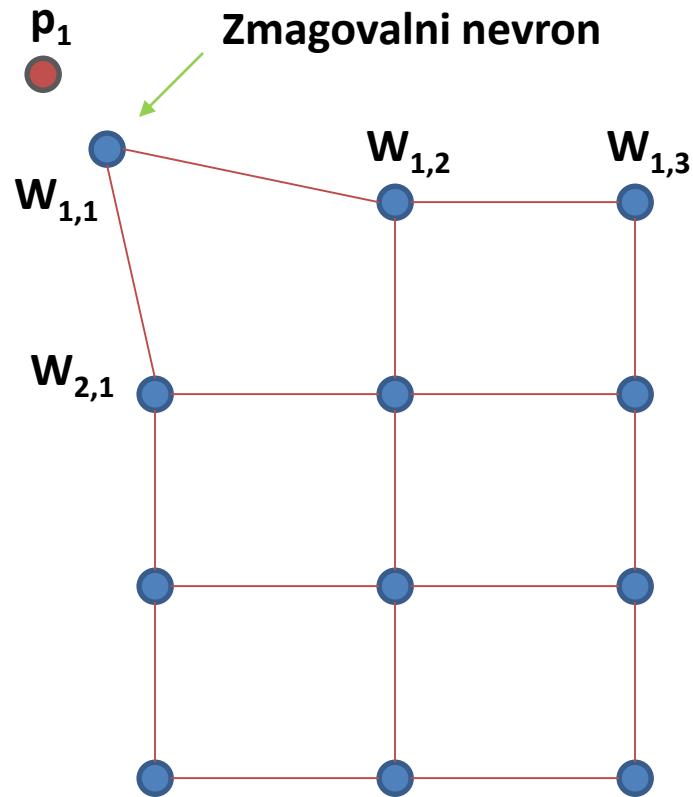
(zmagovalni nevron še ni posodobil svojih uteži)



Samoorganizirajoče nevronske mreže (SOM)

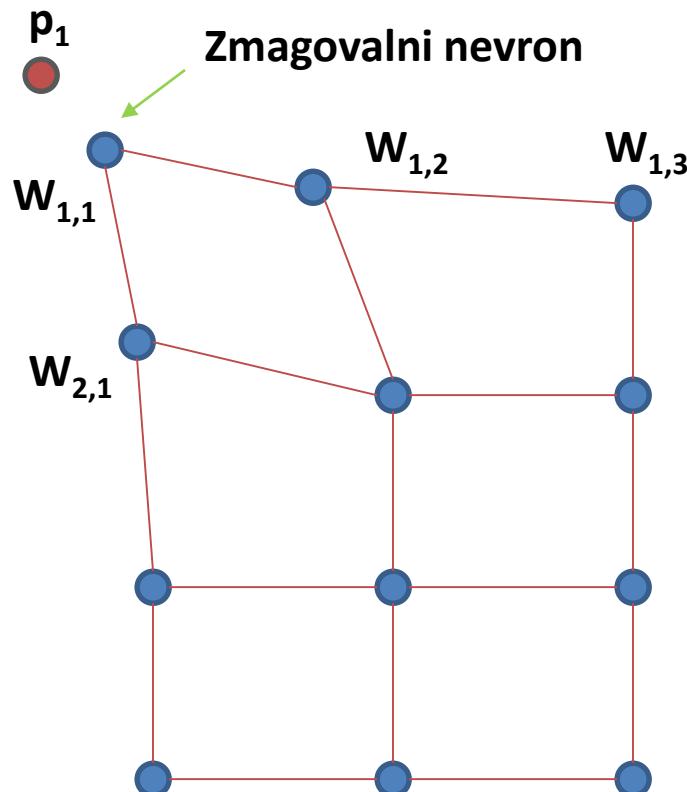
med učenjem

(zmagovalni nevron posodobi svoje uteži
tekmovalne NN)



med učenjem

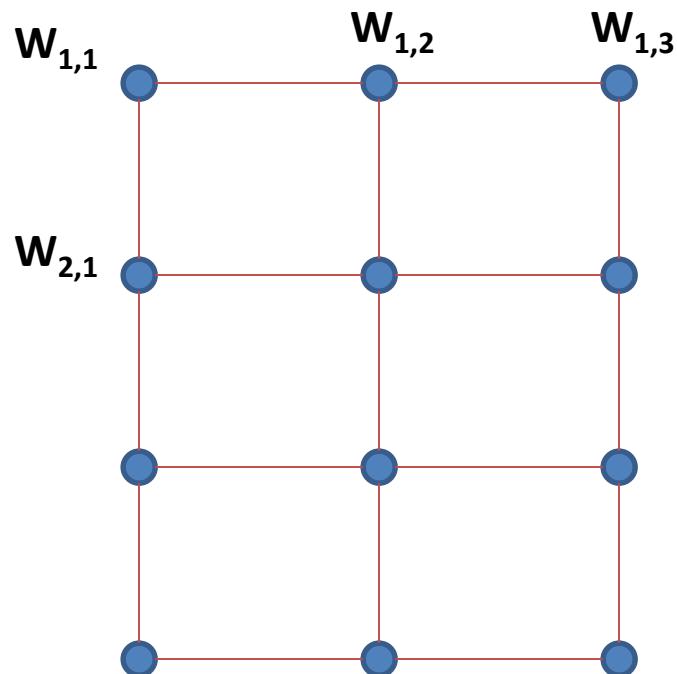
(zmagovalni nevron in njegovi sosedje
posodobijo svoje uteži - SOM)



Samoorganizirajoče nevronske mreže (SOM)

pred učenjem

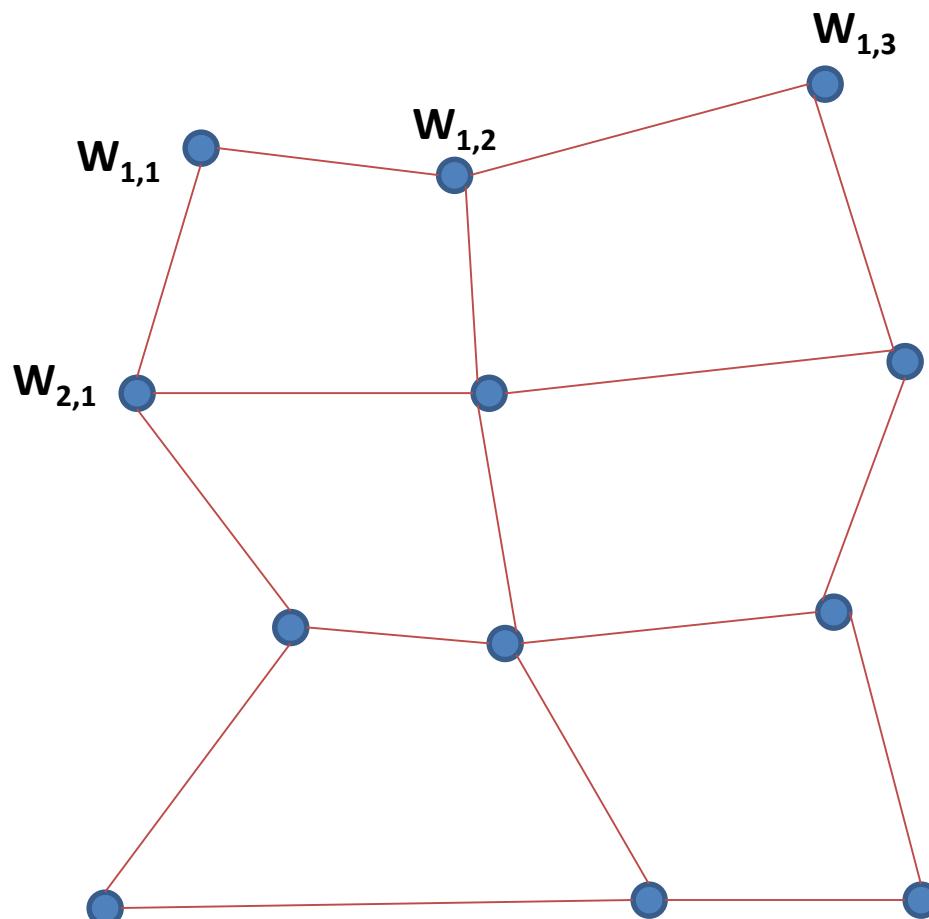
Sosedskost pred učenjem (to niso fizične povezave med nevroni)



Skupni vektorski prostor
uteži nevronov w in tudi
vhodnih vektorjev p

po učenju z veliko vhodnimi podatki

(zmagovalni nevron in njegovi sosedje posodobijo svoje uteži - SOM)

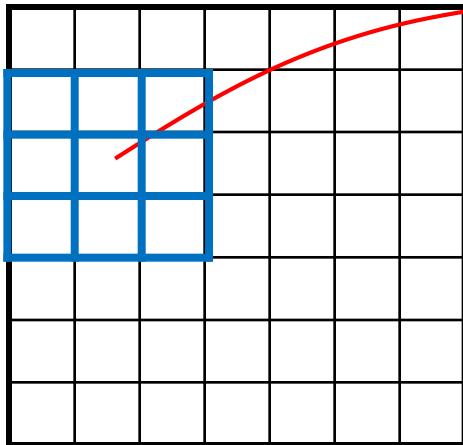


Konvolutivne nevronske mreže

(Convolutional Neural Networks) –

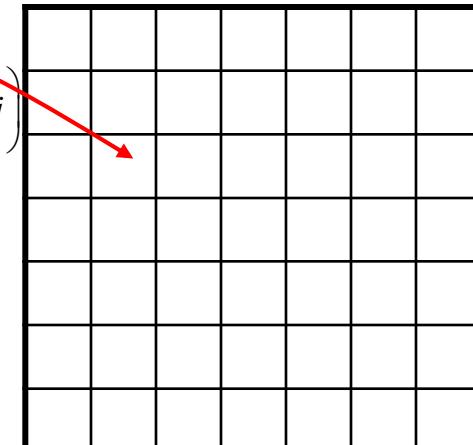
<http://cs231n.github.io/convolutional-networks/>

Konvolucija:



$$I_{iz}(x, y) = \sum_{i=1}^m \sum_{j=1}^n C(i, j) \cdot I_{vh}\left(x - \left\lceil \frac{m}{2} \right\rceil + i, y - \left\lceil \frac{n}{2} \right\rceil + j\right)$$
$$I_{vh}(i, j) = 0, i < 0 \text{ ali } j < 0$$

Slikovni operator C (2D filter)



Vhodna slika I_{vh}

Izhodna slika I_{iz}

- Slikovni operator C (2D filter) se izračuna za vsako lokacijo na sliki.
- Izračuni za posamezne izhodne piksele so neodvisni in lahko tečejo paralelno.
- En nevron implementira en operator (2D filter) na izbrani lokaciji
- Uteži nevrona so uteži/koeficienti 2D filtra
- Vsi nevroni, ki implementirajo isti filter (samo na drugih lokacijah na sliki) se lahko učijo skupaj! To precej pospeši učenje nevronske mreže!

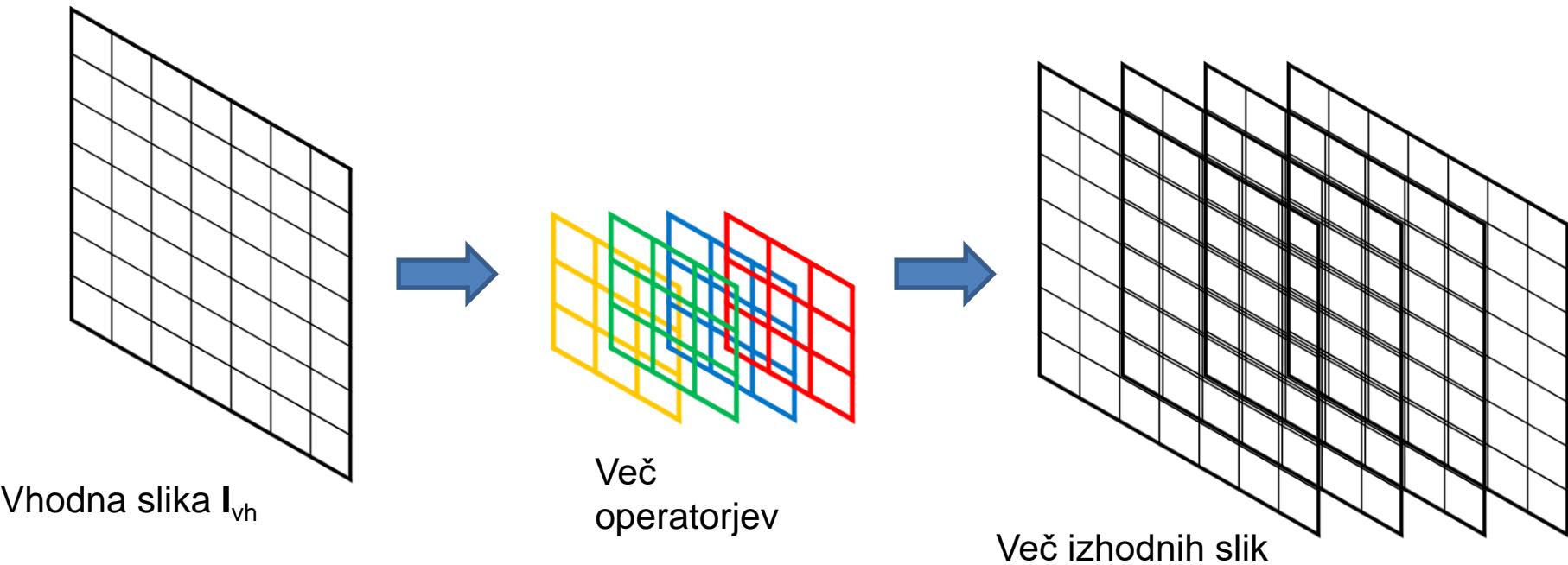
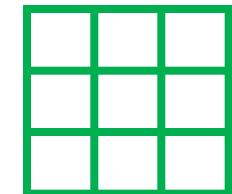
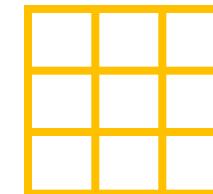
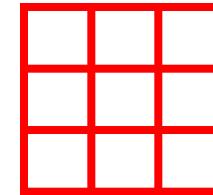
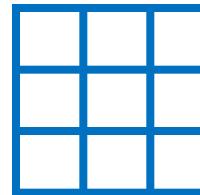
Konvolutivne nevronske mreže

(Convolutional Neural Networks) –

<http://cs231n.github.io/convolutional-networks/>

Možni 2D filtri:

- Detektor robov
- Detektor kotov
- Detektor homogenosti slikovne regije
- Detektor izbrane barve
- ...



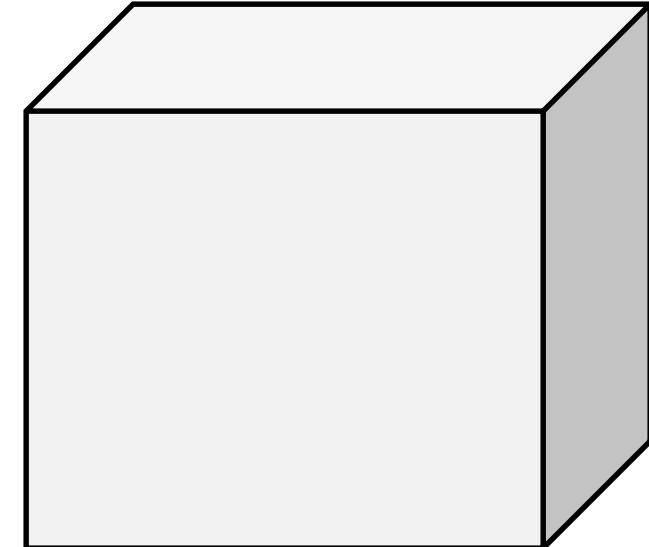
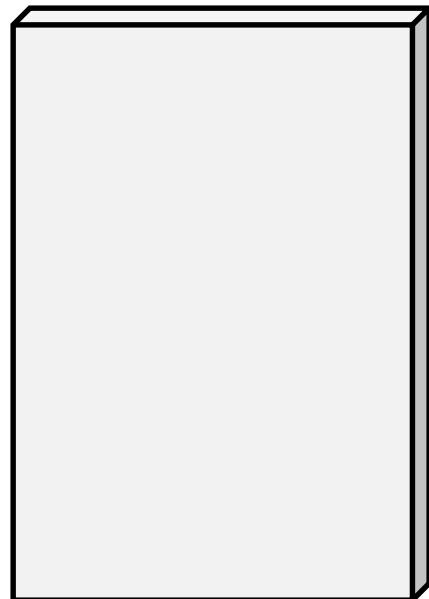
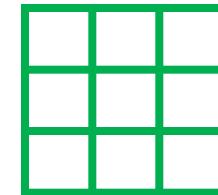
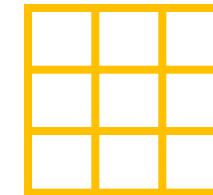
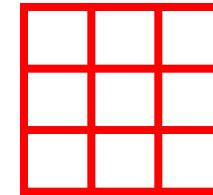
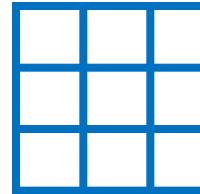
Konvolutivne nevronske mreže

(Convolutional Neural Networks) –

<http://cs231n.github.io/convolutional-networks/>

Možni 2D filtri:

- Detektor robov
- Detektor kotov
- Detektor homogenosti slikovne regije
- Detektor izbrane barve
- ...



Vhodna slika I_{vh}

Več operatorjev

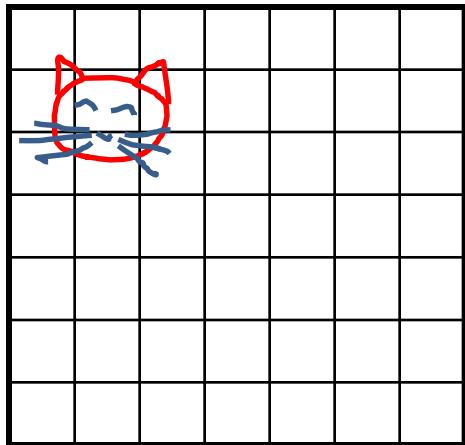
Več izhodnih slik

Konvolutivne nevronske mreže

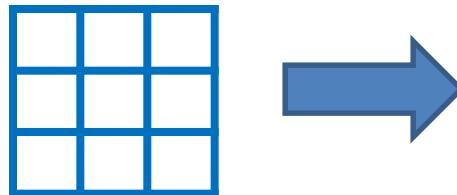
(Convolutional Neural Networks) –

<http://cs231n.github.io/convolutional-networks/>

Kje na sliki je mačka?



Detektor (2D filter) mačke



1	2	1	0	0	0	0	0
3	9	4	0	0	0	0	0
2	3	2	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Aha, mačka je v
zgornjem levem
kotu!

9	0	0
0	0	0
0	0	0

Max pool plast

Osnovni hiperparametri (poleg plasti in št. nevronov)

Kaj je vzorec (angl. sample)?

Vzorec je ena vrstica podatkov. Vsebuje vhode, ki se vnesejo v nevronske mreže in izhod, ki se uporablja za primerjavo z izhodom nevronske mreže (za izračun napake oz. stroškovne funkcije).

Kaj je serija (angl. batch)?

Velikost serije je hiperparameter, ki določa število vzorcev, ki jih je treba obdelati pred posodobitvijo notranjih parametrov nevronske mreže, torej prej posodobitvijo uteži nevronov.

Kaj je število ponovitev (angl. epoch)?

Število ponovitev je hiperparameter, ki določa, kolikokrat bo učni algoritem obdelal celoten nabor podatkov o v učni množici, preden proglasimo nevronske mreže za naučeno.

Kaj je učni algoritem (angl. optimizer)?

Učni algoritem določa strategijo spreminjanja uteži v nevronske mreže, vključno s strategijo izogibanja lokalnih minimumov.

Konvolucijske nevronske mreže trpijo za katastrofalnim pozabljanjem in imajo težava z dolgoročnim sklepanjem

J. Božic, D. Skočaj: Katastrofalno pozabljanje pri inkrementalnem učenju konvolucijske nevronske mreže, ERK2019

[https://erk.fe.uni-lj.si/2019/papers/bozic\(katastrofalno_pozabljanje\).pdf](https://erk.fe.uni-lj.si/2019/papers/bozic(katastrofalno_pozabljanje).pdf)

Michael Nguyen: Illustrated Guide to LSTM's and GRU's: A step by step explanation

<https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>

Christopher Olah: Understanding LSTM Networks

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>



Erwin Schrödinger

Uvod v kvantno mehaniko

Trend miniaturizacije

- **Svet tranzistorjev**
 - Velikost tranzistorjev se zmanjšuje (Moorov zakon)
 - Posamezen tranzistor je tako majhen, da je bližje svetu kvantne mehanike kot pa svetu klasične mehanike
 - Vendar pa se informacija, ki jo oblikujejo ti tranzistorji še vedno interpretira na klasičen način.
 - Zato tvorijo takšni tranzistorji klasična vrata in posledično klasičen računalnik.
- **Kvantni nivo daje računalnikom dostop do novih fizikalnih učinkov**
 - Superpozicija, Interferenca, Kvantno prepletanje (entangulacija), nelokalnost, nedeterminističnost, neizvedljivost kloniranja
 - novi pristopi k algoritmom
- **Nanotehnologija “per se” ne izkorišča vseh kvantnih pojavov**
 - Da bi povečali učinkovitost računalnikov, je potrebno izkoristiti edinstvenost vseh kvantnih učinkov, npr. tudi kvantnega prepletanja (entangulacije)

Atomi in praznina

- *"Po dogovoru obstaja sladkost. Po dogovoru obstaja grenkoba. Po dogovoru obstaja barva. V resnici obstajajo le atomi in praznina."*
Demokrit, 450 pr. n. št., Abdera
- razprava še vedno poteka med atomisti in anti-atomisti: vprašanje v tej razpravi je, ali sta prostor in čas sestavljeni iz nedeljivih delcev, na Planckovi ločljivosti 10^{-33} centimetrov oz. 10^{-43} sekunde.

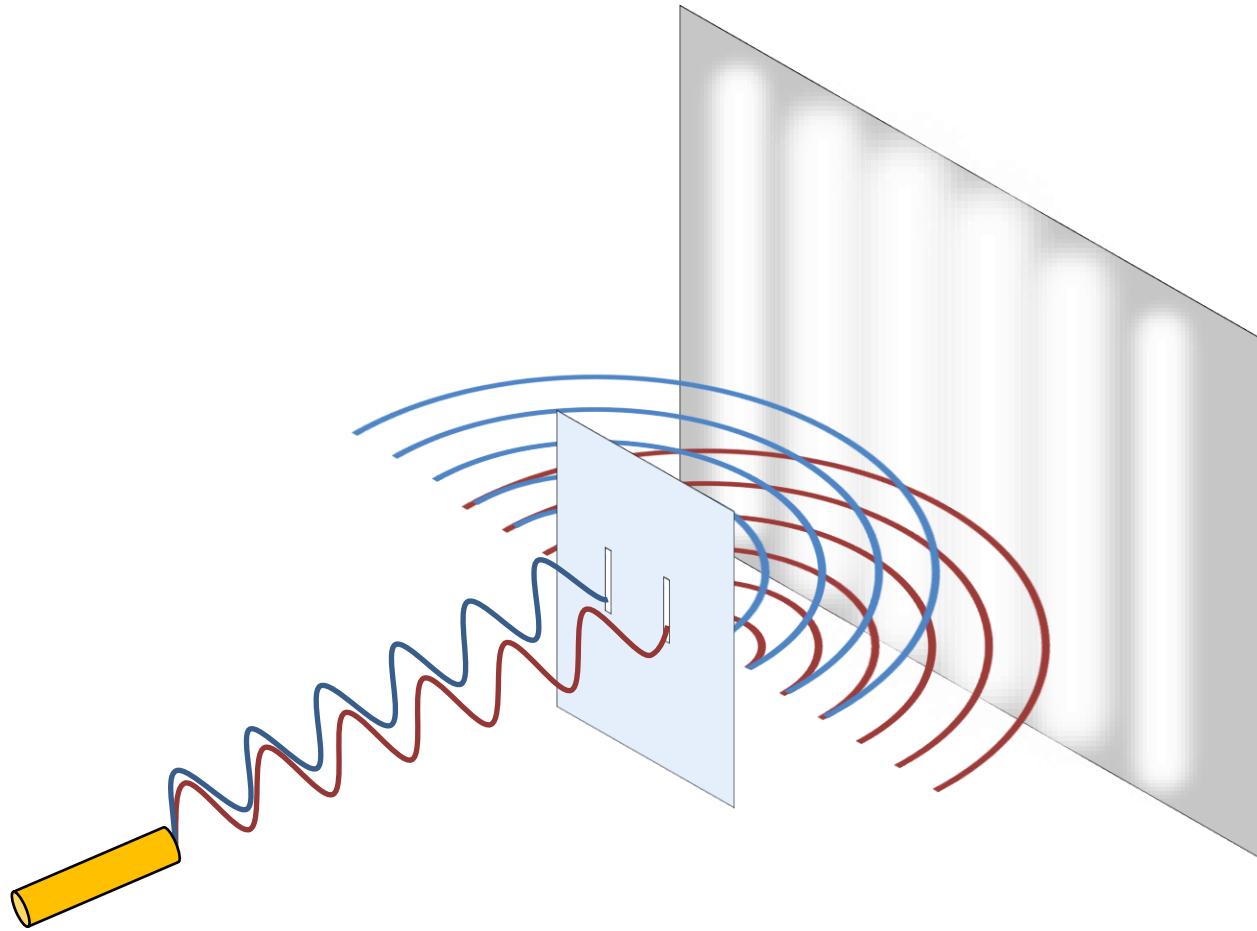
Schrödingerjeva enačba

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r},t) = \hat{H} \Psi(\mathbf{r},t)$$

- kjer je
 - i imaginarna enota ($i^2 = -1$)
 - \hbar Planckova konstanata (6.582×10^{-16} eV·s)
 - $\Psi(\mathbf{r},t)$ valovna funkcija; verjetnostna amplituda različnih konfiguracij opazovanega sistema v času t in poziciji \mathbf{r}
 - \hat{H} Hamiltonov operator
- Za vsako izolirano regijo v vesolju, ki jo želite obravnavati, opisuje ta enačba razvoj stanja te regije v času. Stanje opišemo kot normalizirano linearno kombinacijo - superpozicijo - vseh možnih konfiguracij osnovnih delcev v tej regiji.

Double Slit Experiment

Eksperiment z dvojno režo

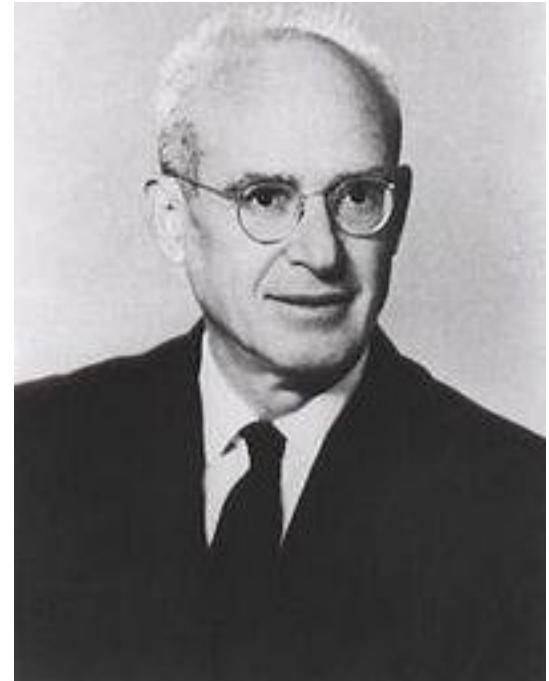
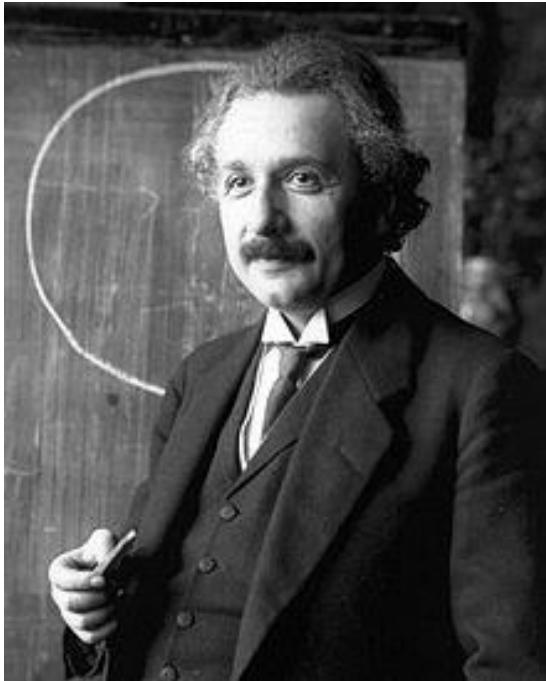


Dr Quantum - Double Slit Experiment
(youtube)

<https://www.youtube.com/watch?v=Q1YqgPAtzho>

Človeški opazovalec in superpozicija

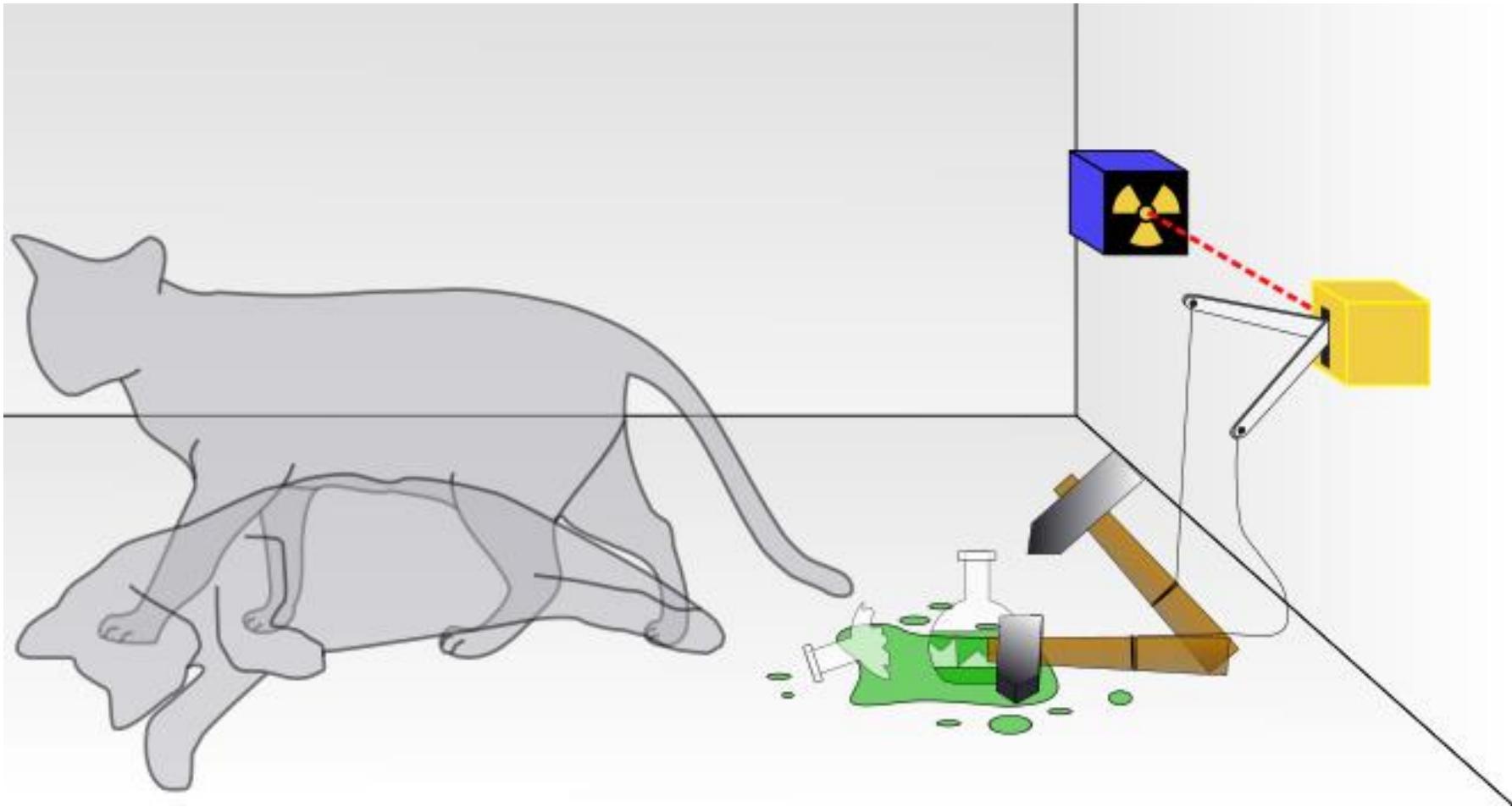
- **Kopenhagenska (epistemološka) interpretacija**
 - Kvantna mehanika opisuje **izide naših meritev in naše dojemanje** in ne podaja nujno objektiven opis oz. resnično stanje kvantnega sistema. Je le teorija o našem znanju.
(Niels Bohr)
- **Interpretacija več svetov**
 - Ob vsakem dogodku z več možnimi izidi se **vesolje razcepi**.
- **Kvantna informacija**
 - **Merjen sistem in merilni sistem se med meritvijo neločljivo prepleteta.** Dobimo novo prepleteno stanje (kvantni preplet ali entangulacija), v katerem merjenega in merilnega sistema ne moremo več informacijsko ločiti.



Einstein–Podolsky–Rosen-ov paradoks

1935

Schrödingerjeva mačka



[Dhatfield](#) - Own work, vir: wikipedia (https://en.wikipedia.org/wiki/Schrödinger's_cat)

Kvantna mehanika

Razmislimo abstraktno o dogodku z N možnimi izidi. Verjetnosti vseh izidov lahko zapišemo z vektorjem N realnih števil :

$$(p_1, \dots, p_N)$$

Kaj lahko povemo o tem vektorju?

- verjetnosti so nenegativne: $p_i \geq 0$
- njihov seštevek je 1.

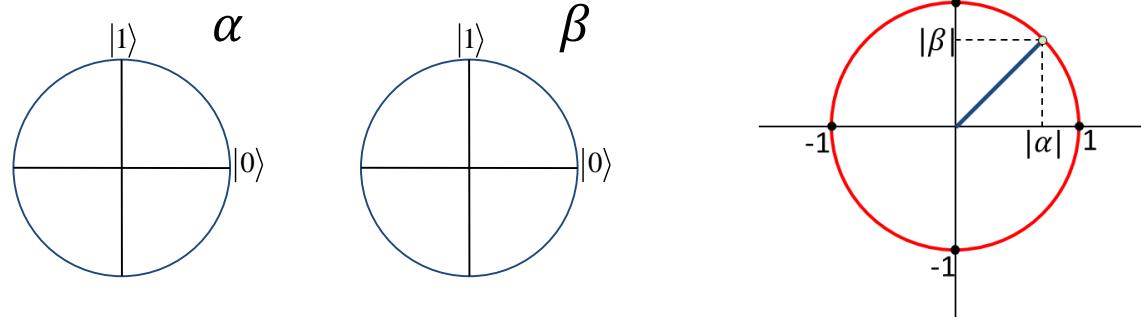
Zadnjo dejstvo lahko izrazimo s pomočjo prve norme: 1-norma vektorja verjetnosti mora biti 1 (1-norma je seštevek absolutnih vrednosti)

Ampak 1-norma ni edina norma na svetu - to ni edini način opredelitve "velikosti" vektorja. Obstajajo tudi drugi načini, in eden od najpogosteje uporabljenih, vsaj od Pitagorovih dni, je druga norma (**2-norma**) ali **Euklidska norma**.

Negativne verjetnosti

Obravnavajmo en sam bit. V verjetnostnem računu lahko bit opišemo z dvema izidoma 0 in 1: bit zavzame vrednost 0 z verjetnostjo p in vrednost 1 z verjetnostjo $1-p$.

Če pa namesto **1-norme** uporabimo **2-normo**, ne želimo več seštevati števil ampak njihove kvadratne vrednosti (Pitagorov izrek), njihov seštevek pa mora biti 1. Z drugimi besedami, želimo vektor (α, β) kjer $|\alpha|^2 + |\beta|^2 = 1$. Množica vseh takšnih vektorjev tvori krog v ravnini.



Toda zakaj v tem primeru ne pozabimo na α in β in samo opišemo stanje bita neposredno v smislu verjetnosti? Razlika nastopi pri transformaciji vektorja oz. v tem kako se vektor spremeni, ko ga vstavimo v linearino operacijo.

Kvantna mehanika & kvantni bit

Če je objekt lahko v dveh stanjih $|0\rangle$ ali $|1\rangle$, potem je lahko tudi **superpoziciji** teh stanj

$$\alpha|0\rangle + \beta|1\rangle$$

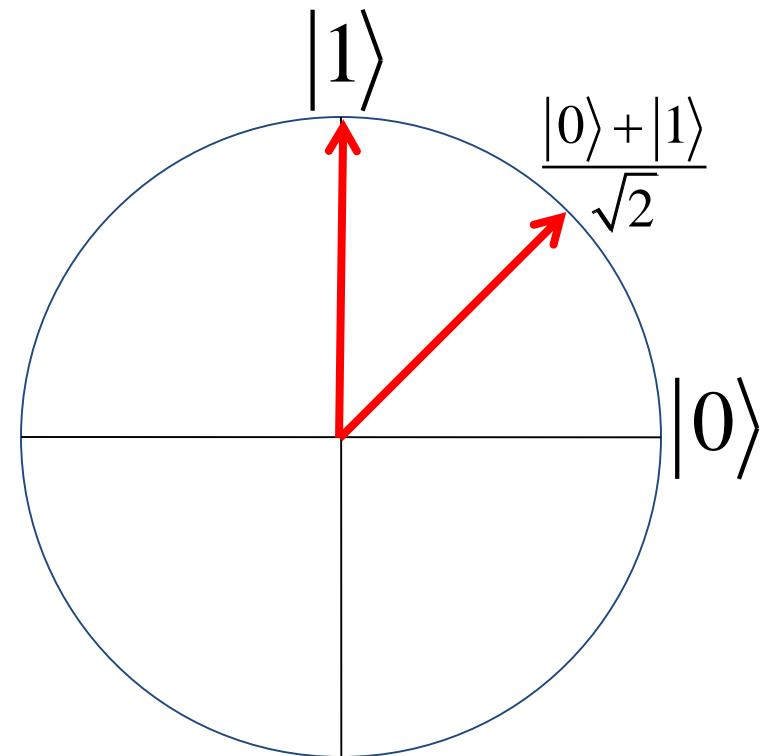
Tu sta α in β kompleksni **amplitudi verjetnosti**

$$|\alpha|^2 + |\beta|^2 = 1$$

Če opazujemo ta objekt, bomo videli

$$|0\rangle \text{ z verjetnostjo } |\alpha|^2$$

$$|1\rangle \text{ z verjetnostjo } |\beta|^2$$



Takoj ko objekt pogledamo, le ta **kolapsira** v katerokoli izmed obih osnovnih stanj, $|0\rangle$ ali $|1\rangle$.

Negativne verjetnosti

"Enotski vektor 2-norme" se imenuje kvantni bit (***qubit***) in fiziki ga navadno predstavijo s "**Diracovo ket notacijo**" v kateri vektor (α, β) postane $\alpha|0\rangle + \beta|1\rangle$.

- α je ***amplituda verjetnosti*** izida $|0\rangle$,
- β je ***amplituda verjetnosti*** izida $|1\rangle$.

Analogno lahko vsak kvantni bit predstavimo v dvodimenzionalnem vektorskem prostoru kot

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

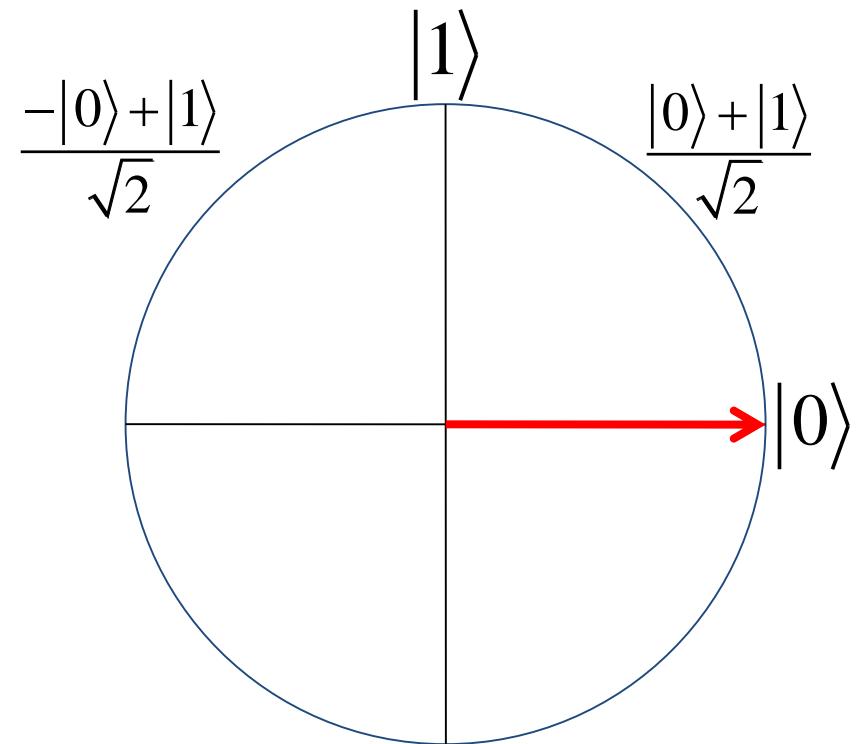
Potem lahko vsak kvanti bit s katerokoli unitarno matriko velikosti 2×2 pretvorimo v nov kvantni bit. Slavno kvantno interferenco lahko na primer zapišemo z Hadamardovo matriko **H**

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Privzemimo, da je kvantni bit v stanju $|0\rangle$. Če ga pomnožimo z zgornjo matriko **H** dobimo $1/\sqrt{2}(|0\rangle + |1\rangle)$. Če ta rezultat še enkrat pomnožimo z **H** dobimo $|0\rangle$

Kvantne transformacije: Hadamardova transformacija

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



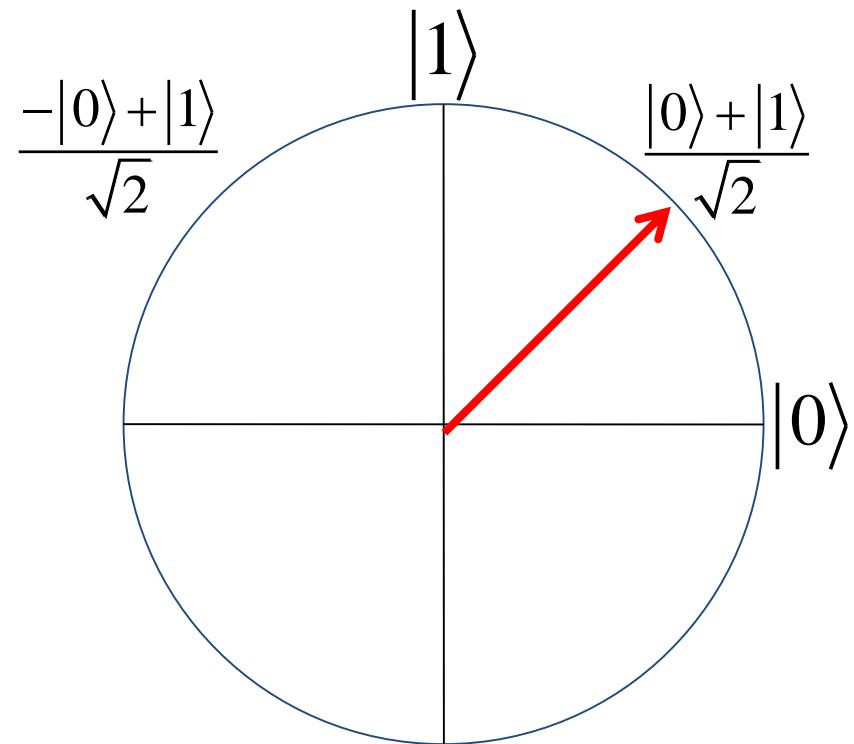
$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Hadamardova transformacija implementira **interferenco** amplitud verjetnosti – vir vse “kvantne čudaškosti”

Kvantne transformacije: Hadamardova transformacija

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

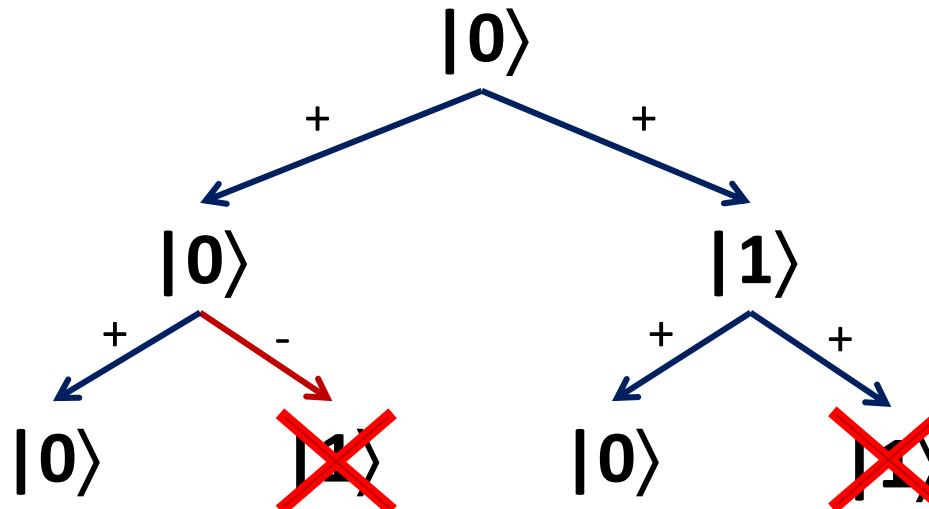
$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$



Hadamardova transformacija implementira **interferenco** amplitud verjetnosti – vir vse “kvantne čudaškosti”

Negativne verjetnosti

Čeprav sta v obravnavani matriki H dve poti, ki vodita do izida $|1\rangle$, ima ena pot pozitivno amplitudo verjetnosti, druga pa negativno amplitudo verjetnosti. Zaradi tega obe poti ***destruktivno interferirata*** in se medsebojno izničita. Poti ki vodita do izida $|0\rangle$ imata obe pozitivni amplitudi verjetnosti in ***interferirata konstruktivno***.



Izničenje pozitivnih in negativnih amplitud verjetnosti poti, ki vodijo k določenem izidu je izvor vseh "kvantnih čudes" in tista lastnost, ki predstavlja glavno razliko med klasično in kvantno verjetnostjo.

Zakaj 2-norma?

- Vzemimo teorijo, ki temelji na p-normi kjer je $p \in \{1, 2\}$.
- Vektor (v_1, \dots, v_N) je *enotski vektor p-norme* če $|v_1|^p + \dots + |v_N|^p = 1$.
- Poščimo linearno transformacijo, ki preslika katerikoli enotski vektor p-norme v drug enotski vektor p-norme.
- Za katerikoli izbrani p lahko najdemo linearne transformacije, ki ohranijo p-norm:
 - lahko na primer permutiramo elemente vektorja
 - lahko vstavimo negativne predznaKE.
- Toda, **če poleg teh trivialnih trasformacij obstaja še katerakoli druga linearNA trasformacija, ki ohranja p-normo, potem je p=1 ali p=2.**
 - če $p=1$, dobimo klasični verjetnostni račun,
 - če $p=2$ dobimo kvantno verjetnost (amplitudo verjetnosti).

Realna vs. kompleksna števila

- Amplitude verjetnosti kvantne mehanike so kompleksna števila. To pomeni, da moramo kvadrirati absolutne vrednosti amplitud, da dobimo verjetnost. Z drugimi besedami, če je amplituda verjetnosti za nek izid $\alpha = \beta + \gamma i$, kjer sta β in γ realni števili, potem je verjetnost tega izida enaka $|\alpha|^2 = \beta^2 + \gamma^2$.
- **Vprašanje:** Zakaj je narava izbrala kompleksna števila in ne realna?
- **Odgovor:** Kompleksna števila so algebrajsko zaprta. Z drugimi besedami, za katerokoli linearno transformacijo U , obstaja linearnejša transformacija V tako da velja $V^2 = U$.

Zgoraj podana relacija v bistvu definira **zveznost**: če je smiselno, da operacijo izvedemo za časovni interval ene sekunde, mora biti smiselno tudi, da jo izvedemo za interval pol sekunde...

Zakaj linearne transformacije?

- Abrams and Lloyd, 1998 [1]: “***if quantum mechanics were nonlinear, then one could build a computer to solve NP-complete problems in polynomial time***”.
- Torej bi lahko, če bi bila kvantna mehanika nelinearna, v polinomskem času izračunali karkoli

KARKOLI!

- Torej, ali je naš svet linearen ali pa lahko vnaprej napovemo/izračunamo katerikoli dogodek na svetu, od gibanja delnic, do življenjske dobe človeka in njegove svobodne volje...

[1] D. S. Abrams, S. Lloyd: Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems, Phys.Rev.Lett. 81 (1998) 3992-3995

Izjave o kvantni mehaniki

- “Basically, quantum mechanics is the operating system that other physical theories run on as application software (with the exception of general relativity, which hasn't yet been successfully ported to this particular OS)”
- “it's not about matter, or energy, or waves, or particles - it's about information and probabilities and observables, and how they relate to each other”.

“Quantum mechanics is what you would inevitably come up with if you started from probability theory, and then said, let's try to generalize it so that the numbers we used to call "probabilities" can be negative numbers. As such, the theory could have been invented by mathematicians in the 19th century without any input from experiment. It wasn't, but it could have been.”

Scott Aaronson

Quantum Computing Since Democritus

(<http://scottaaronson.com/democritus/default.html>)

Tenzorski produkt

- Če poznamo stanji dveh kvantnih bitov, potem lahko njuno **kombinirano stanje** zapišemo s **tensorskim produkтом**:
- **primer**
 - če je prvi kvanti bit: $\alpha|0\rangle+\beta|1\rangle$
 - in drugi kvantni bit: $\gamma|0\rangle+\delta|1\rangle$
 - potem njuno kombinirano stanje zapišemo kot

$$(\alpha|0\rangle+\beta|1\rangle) \otimes (\gamma|0\rangle+\delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

Ločljiva vs. prepletena stanja

- **Ločljiva stanja:** stanja dveh ali več kvantnih bitov, ki jih lahko zapišemo kot tenzorski produkt posameznih kvantnih bitov.
- **Prepletena stanja:** stanja dveh ali več kvantnih bitov, ki jih ne moremo zapisati kot tenzorski produkt posameznih kvantnih bitov. Najbolj slavno prepleteno stanje para kvantnih bitov je stanje EPR (Einstein-Podolsky-Rosen) :

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

- **Formalna definicija:** Če lahko kombinirano stanje ρ dveh podsistemov A in B, ki sta v stanjih $|\psi_A\rangle$ in $|\psi_B\rangle$, zapišemo z verjetnostno porazdelitvijo tenzorskega produkta stanj $\alpha|\psi_A\rangle\otimes\beta|\psi_B\rangle$, potem je ρ **ločljivo** stanje. V nasprotnem primeru je ρ **prepleteno** stanje. V zgornjem primeru sta α in β amplitudi verjetnosti.

<https://www.youtube.com/watch?v=O8Ia3kcQydc>

**Dr Quantum - entanglement
(youtube)**

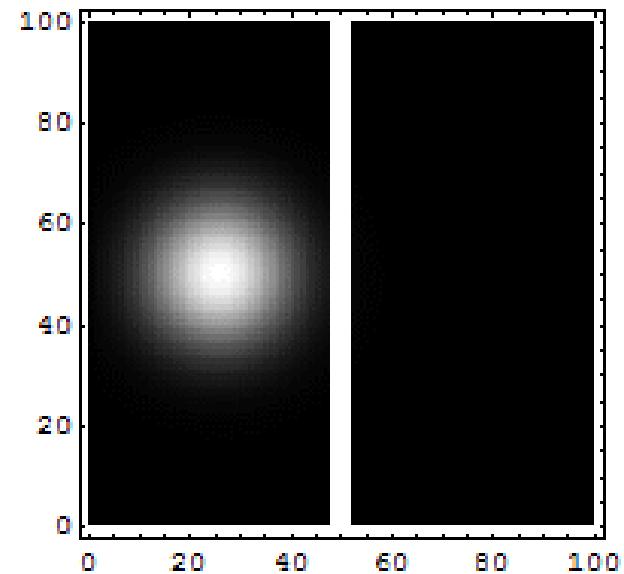
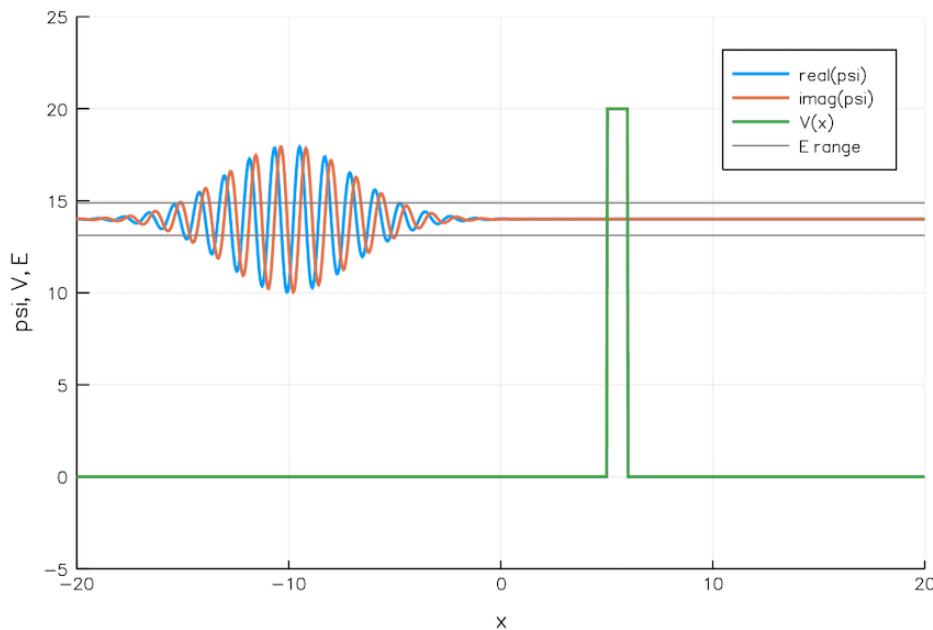
Teorem neizvedljivosti kloniranja (No-cloning theorem)

- prepoveduje kloniranje identičnih kopij poljubnega kvantnega stanja
- zagotovljen je s strani linearnosti kvantne mehanike.
- Posledice:
 - ne moremo izdelati varnostnih (backup) kopij kvantnega stanja in jih med kvantnim izračunom uporabiti za odpravo računskih napak oz. za odkrivanje rešitev sistema enačb (Shorov algoritem).
 - omogoča varno izmenjavo ključev: prisluškovalec ne more izdelati kopije poslanega kvantnega kriptografskega ključa.

Kvantno tuneljenje

(Quantum tunneling)

- **Klasična mehanika:** delci, ki nimajo dovolj energije, da bi prešli oviro, je ne bodo nikoli prešli.
- **Kvantna mehanika:** Ti delci lahko z zelo majhno verjetnostjo preidejo na drugo stran ovire. Pojavu pravimo kvantno tuneljenje.





Richard Feynman

Kvantno računalništvo (quantum computing)

[Richard Feynman](#) (1982). "Simulating physics with computers".
International Journal of Theoretical Physics **21**: 467.

Klasična predstavitev podatkov

- Osnovna enota klasičnih podatkov je bit, ki lahko zavzame vrednosti 0 ali 1.
- Klasični računalnik predstavi podatke kot niz bitov.
- Črko 'A' zapišemo kot 0100 0001
- Število 165 zapišemo kot 1010 0101

binarne kode:

$$10100101_2 =$$

$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$

$$1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 =$$

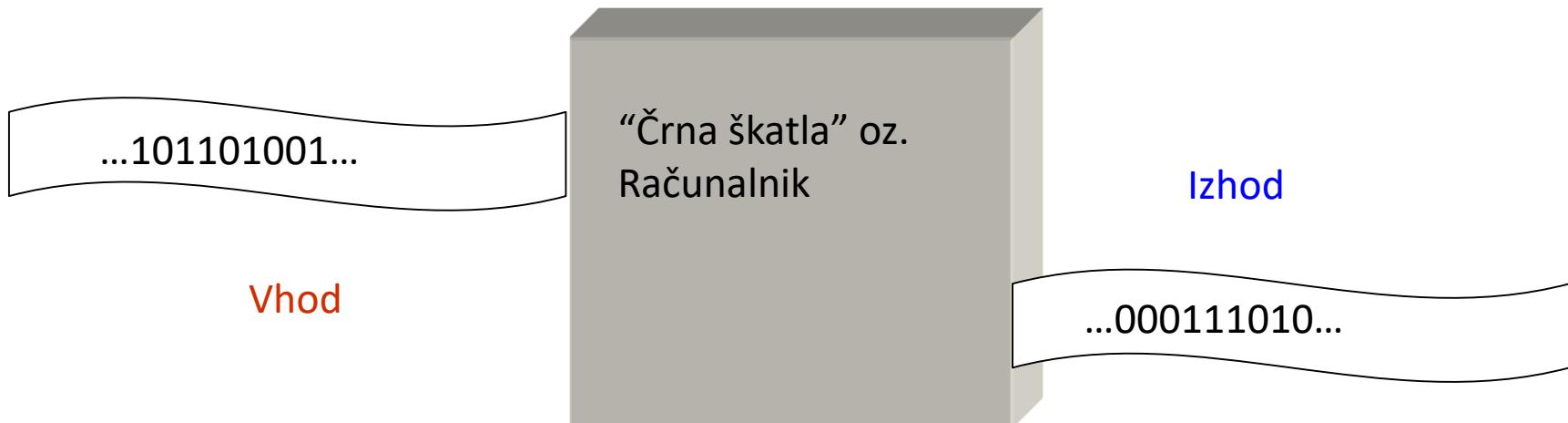
$$165_{10}$$

Klasične operacije

- Vse operacije temeljijo na logičnih vratih.
- Na primer, logična vrata IN (AND) sprejmejo dva vhodna bita in vrnejo 1, če in samo če sta oba vhoda na 1.

Klasični algoritem

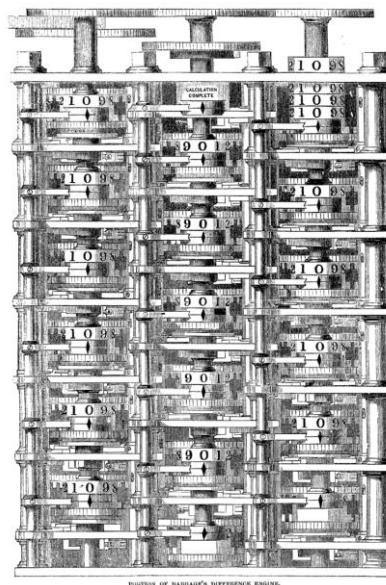
- Klasični algoritem je vsako zaporedje klasičnih operacij.
- Klasičen računalnik je vsaka naprava, ki lahko implementira klasični algoritem.



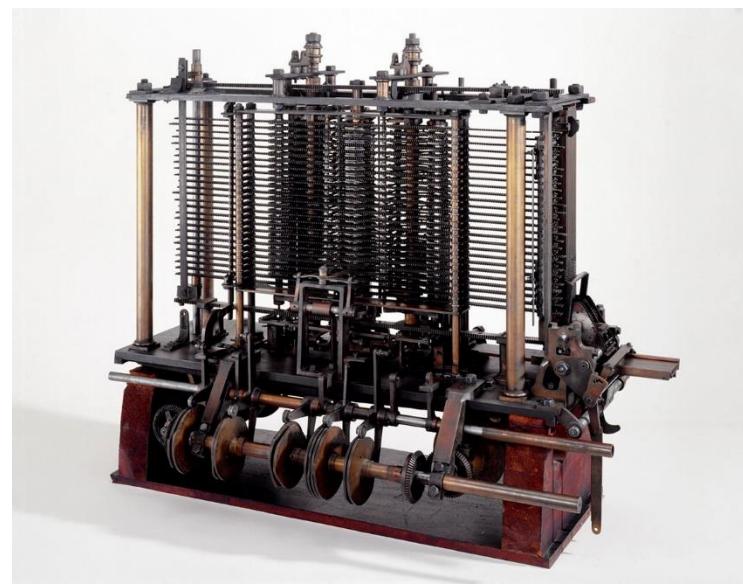
Klasični računalnik

- Čeprav moderni računalniki temeljijo na kvanti mehaniki (tranzistorjih), še vedno poganjajo klasične algoritme.
- V principu bi lahko izdelali klasični računalnik, ki ne bi temeljil na kvantni mehaniki (npr. mehanski stroj iz zobnikov).

Charles Babbage
(1791-1871)



[https://en.wikipedia.org/wiki/Charles_Babbage#/media/
File:Babbage_difference_engine_drawing.gif](https://en.wikipedia.org/wiki/Charles_Babbage#/media/File:Babbage_difference_engine_drawing.gif)



[https://en.wikipedia.org/wiki/Analytical_engine#/media/
File:Babbages_Analytical_Engine,_1834-1871_\(9660574685\).jpg](https://en.wikipedia.org/wiki/Analytical_engine#/media/File:Babbages_Analytical_Engine,_1834-1871_(9660574685).jpg)

Kaj torej je kvantni računalnik?

Kvantni računalnik

- Kvantni računalnik je računalnik, ki v izračunih neposredno uporablja lastnosti kvantnega sveta:
 - Superpozicija stanj
 - Prepletost stanj
 - Kvantno tuneljenje
- S tem izrazno krepko razširi klasično računanje (veliko dodatnih prostostnih stopenj) in eksponentno pohitri iskanje rešitev.
- Številni problemi, ki imajo na klasičnem računalniku eksponentno računsko zahtevnost, so na kvantnem računalniku rešljivi v polinomskem času.

Kvantna mehanika & kvantni bit

Če je objekt lahko v dveh stanjih $|0\rangle$ ali $|1\rangle$, potem je lahko tudi **superpoziciji** teh stanj

$$\alpha|0\rangle + \beta|1\rangle$$

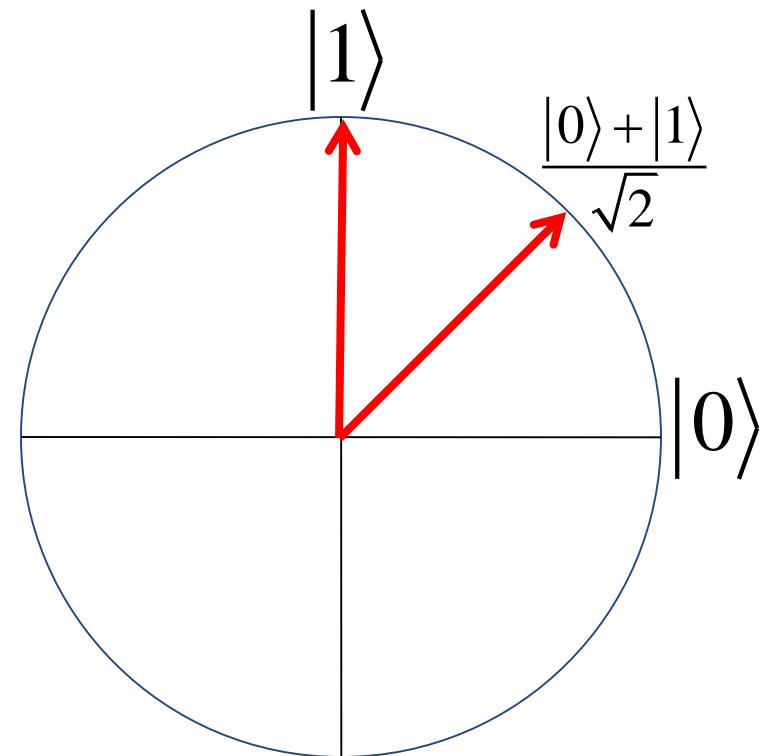
Tu sta α in β kompleksni **amplitudi verjetnosti**

$$|\alpha|^2 + |\beta|^2 = 1$$

Če opazujemo t objekt, bomo videli

$$|0\rangle \text{ z verjetnostjo } |\alpha|^2$$

$$|1\rangle \text{ z verjetnostjo } |\beta|^2$$



Takoj ko objekt pogledamo, le ta **kolapsira** v katerokoli izmed obeh osnovnih stanj, $|0\rangle$ ali $|1\rangle$.

Kvantni biti

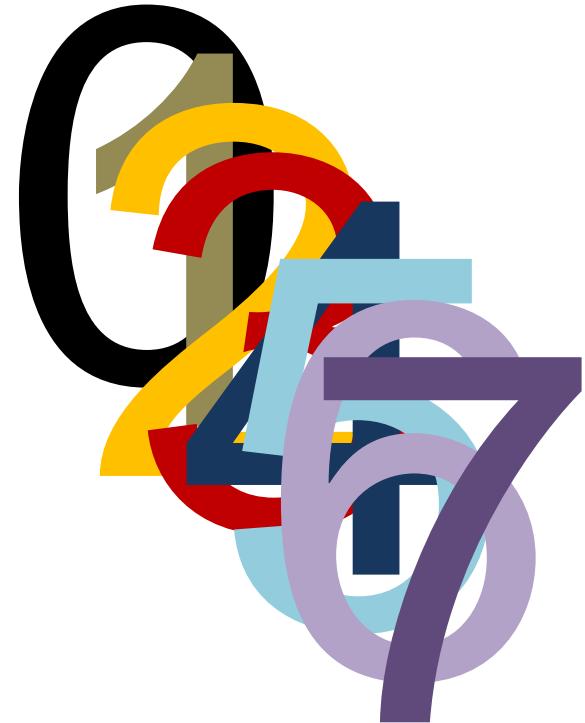
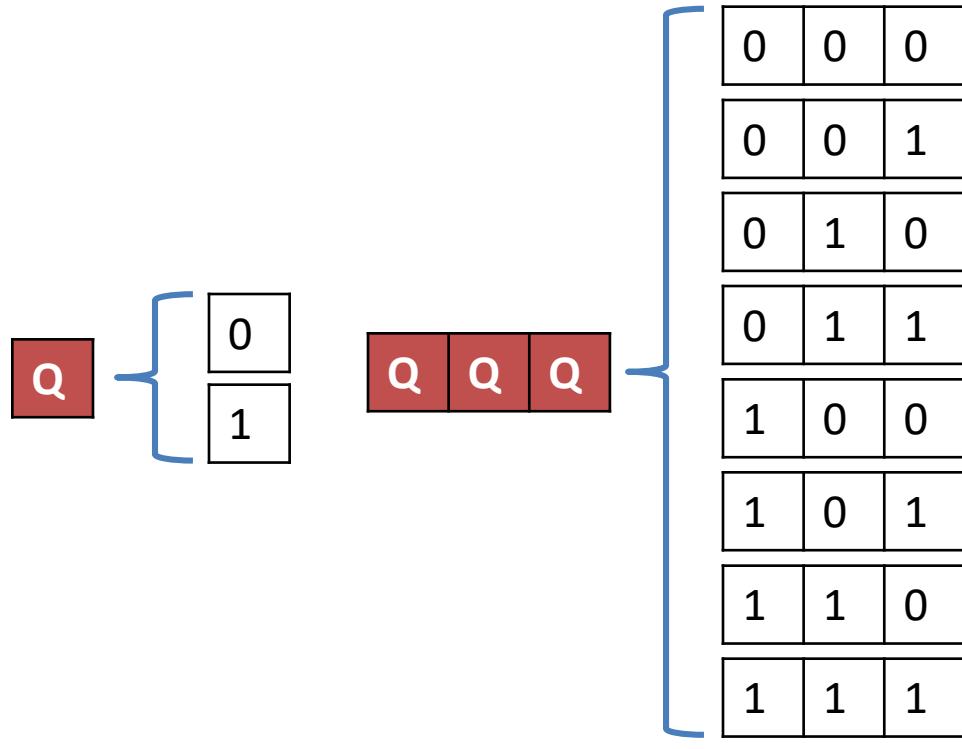
- Za razliko od klasičnega bita, ki je zagotovo v enem od dveh stanj, je stanje kvantnega bita v splošnem mešanica obeh stanj $|0\rangle$ in $|1\rangle$.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- Amplitudi verjetnosti sta normirani (2-norma):

$$|\alpha|^2 + |\beta|^2 = 1$$

Kvantni register



100 kvantnih bitov lahko hrani več klasičnih bitov informacij kot je atomov v znanem vesolju!

Kvantni register: superpozicija

- zbir n kvantnih bitov:

$$|a\rangle = |a_{n-1}\rangle \otimes |a_{n-2}\rangle \dots |a_1\rangle \otimes |a_0\rangle$$

- **primjeri:**

- $|3\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle = |011\rangle$

- $|7\rangle = |1\rangle \otimes |1\rangle \otimes |1\rangle = |111\rangle$

- $|21\rangle = |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle = |10101\rangle$

- $1/\sqrt{2} (|3\rangle + |7\rangle) = 1/\sqrt{2} (|0\rangle + |1\rangle) \otimes |1\rangle \otimes |1\rangle$

- $2^{-3/2} (|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle) =$
 $1/\sqrt{2} (|0\rangle + |1\rangle) \otimes$
 $1/\sqrt{2} (|0\rangle + |1\rangle) \otimes$
 $1/\sqrt{2} (|0\rangle + |1\rangle)$

Kvantna vrata

- Kvantna logična vrata implementirajo unitarno transformacijo stanja enega ali več kvantnih bitov v novo stanje kvantnih bitov.
- predstavimo jih lahko kot linearne operatorje v Hilbertovem prostoru.
- **Nelinearne transformacije so PREPOVEDANE!**
- Kvanta vrata je najprimernejše predstaviti z matrikami, kjer stanje posameznega kvantnega bita zapišemo v bazi stanj $|0\rangle$ in $|1\rangle$:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Kvantna vrata & unitarna matrika

Stanje

$$\sum_{i=1}^n \alpha_i |i\rangle$$

spremenimo tako, da ga predmnožimo z
unitarno matriko — takšno matriko, ki
ohranja relacijo.

$$\sum_{i=1}^n |\alpha_i|^2 = 1$$

Kvantna vrata NOT

- Kot pri klasičnem računanju, kvantna vrata NOT vrnejo $|0\rangle$, če je vhod $|1\rangle$ in $|1\rangle$, če je vhod $|0\rangle$.
- Matrična predstavitev vrat NOT:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

zgornja matrika je podana v bazi:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Kvantna vrata NE

- Kot pri klasičnem računanju, kvantna vrata NOT vrnejo $|0\rangle$, če je vhod $|1\rangle$ in $|1\rangle$, če je vhod $|0\rangle$:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Rezultat postane zanimiv, ko vrata uporabimo nad superpozicijo stanj

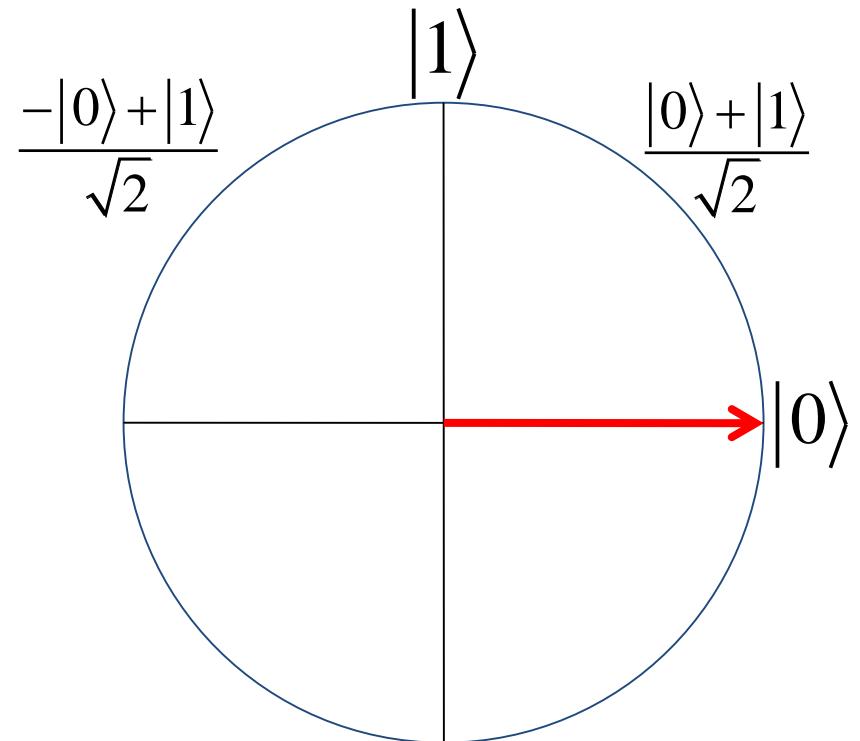
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

- V tem primeru vrata NE zamenjajo amplitudi verjetnosti kvantnega bita!

Kvantna vrata: Hadamard-ova vrata

- prejmejo en sam vhodni kvantni bit

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

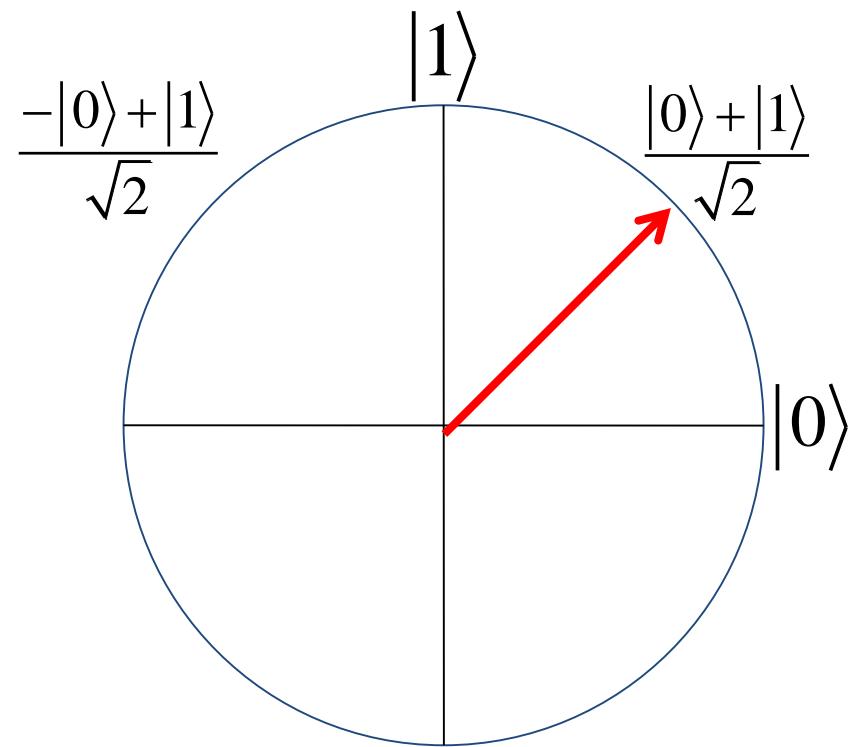
Kvantna vrata: Hadamard-ova vrata

- prejmejo en sam vhodni kvanti bit

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

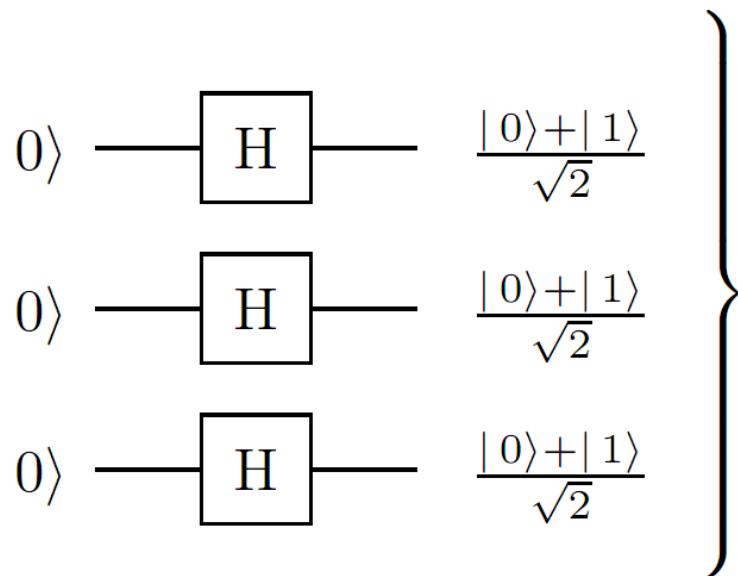


Kvantna vrata: Hadamard-ova vrata

- H je sama sebi inverz: $H = H^T = H^{-1}$

$$H \cdot H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- aplikacija na več kvantnih bitov:



IN BINARY

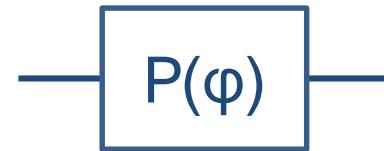
$$= \frac{1}{2^{3/2}} \left\{ |000\rangle + |001\rangle + |010\rangle + |011\rangle + \right. \\ \left. + |100\rangle + |101\rangle + |110\rangle + |111\rangle \right\}$$
$$= \frac{1}{2^{3/2}} \left\{ |0\rangle + |1\rangle + |2\rangle + |3\rangle + \right. \\ \left. + |4\rangle + |5\rangle + |6\rangle + |7\rangle \right\}$$

IN DECIMAL

Kvantna vrata: Fazna vrata

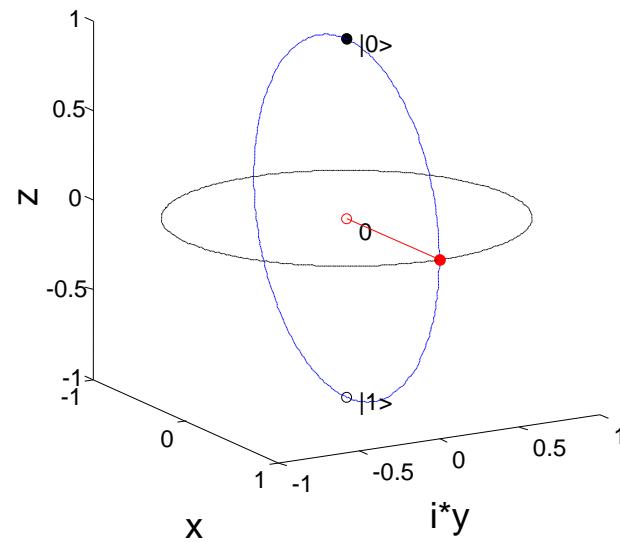
- prejmejo en vhodni kvantni bit

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$$



$$\varphi = 0 \Rightarrow P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

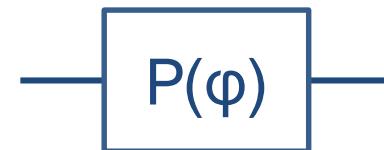
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$



Kvantna vrata: Fazna vrata

- prejmejo en vhodni kvantni bit

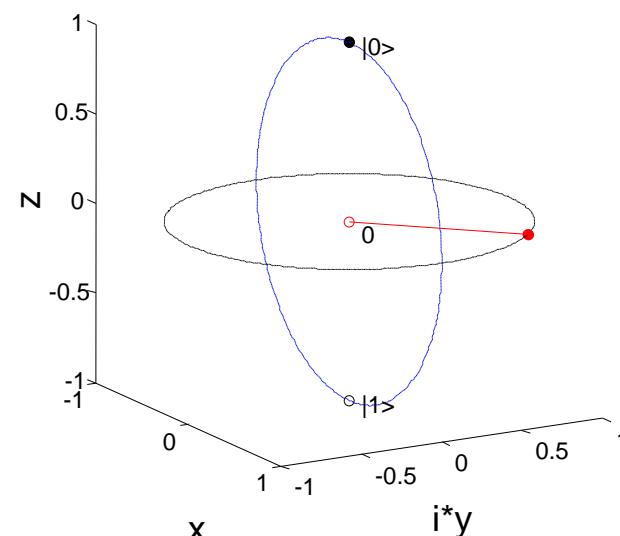
$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$$



$$\varphi = \frac{\pi}{4} \Rightarrow P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} + i \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} + i \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{2} + i \frac{1}{2} \end{bmatrix}$$



Kvantna vrata: Fazna vrata

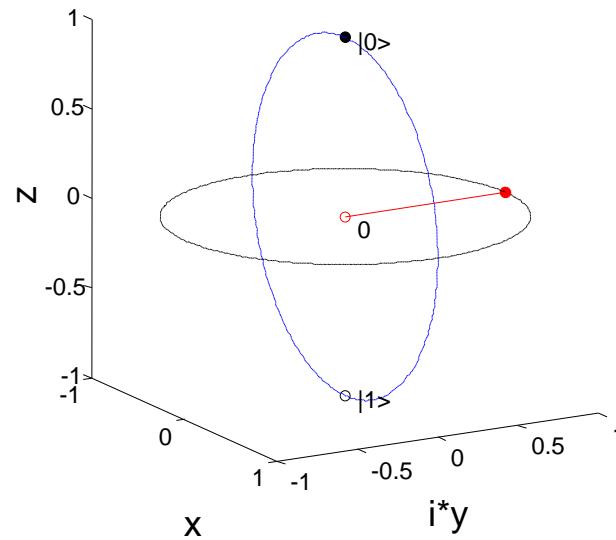
- prejmejo en vhodni kvantni bit

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$$



$$\varphi = \frac{\pi}{2} \Rightarrow P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

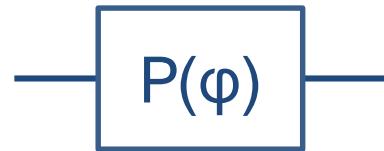
$$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ i \frac{1}{\sqrt{2}} \end{bmatrix}$$



Kvantna vrata: Fazna vrata

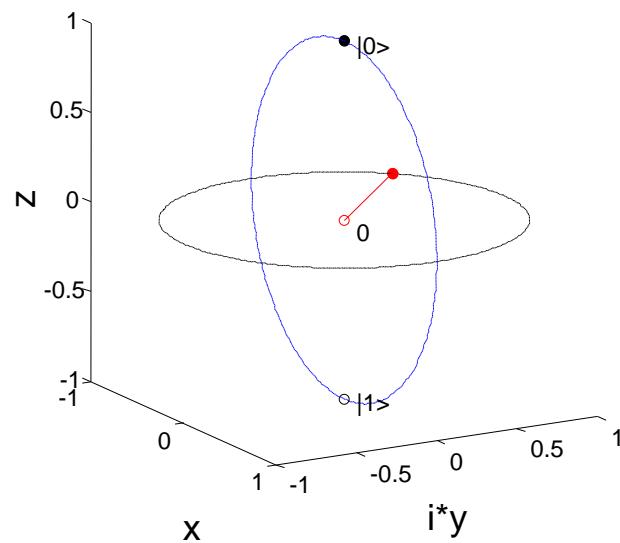
- prejmejo en vhodni kvantni bit

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$$



$$\varphi = \frac{3\pi}{4} \Rightarrow P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{\sqrt{2}} + i\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{\sqrt{2}} + i\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{2} + i\frac{1}{2} \end{bmatrix}$$



Kvantna vrata: Fazna vrata

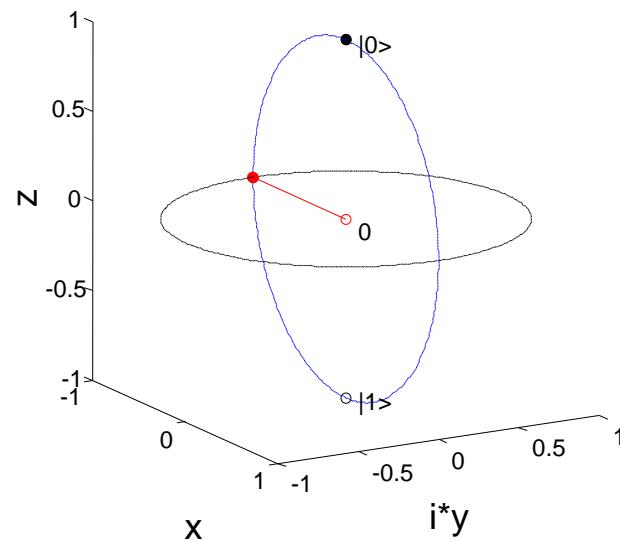
- prejmejo en vhodni kvantni bit

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$$



$$\varphi = \pi \Rightarrow P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

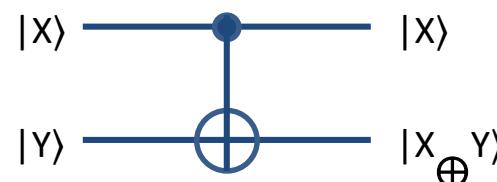
$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$



Kontrolirana NE vrata (Controlled-NOT)

- prejmejo dva kvantna bita, implementirajo XOR

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot |00\rangle \\ 0 \cdot |01\rangle \\ 0 \cdot |10\rangle \\ 0 \cdot |11\rangle \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot |00\rangle \\ 1 \cdot |01\rangle \\ 0 \cdot |10\rangle \\ 0 \cdot |11\rangle \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot |00\rangle \\ 0 \cdot |01\rangle \\ 0 \cdot |10\rangle \\ 1 \cdot |11\rangle \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \cdot |00\rangle \\ 0 \cdot |01\rangle \\ 1 \cdot |10\rangle \\ 0 \cdot |11\rangle \end{bmatrix}$$

Kvantna prepletjenost in Bellovo stanje

- Dana sta dva kvantna bita. Oba sta v stanju $|0\rangle$:

$$|0\rangle, |0\rangle$$

- Nad prvim bitom uporabimo Hadamardova vrata:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, |0\rangle$$

- Tvorimo tenzorski produkt obeh kvantnih bitov:

$$\frac{1}{\sqrt{2}}|00\rangle + 0|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + 0|11\rangle$$

- Nad registrom uporabimo vrata CNOT

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|00\rangle + 0|01\rangle + 0|10\rangle + \frac{1}{\sqrt{2}}|11\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

- To stanje imenujemo Bellovo stanje (po fiziku John Stewart Bell-u iz Severne Irske) ali stanje ERP (Einstein, Podolsky & Rosen).

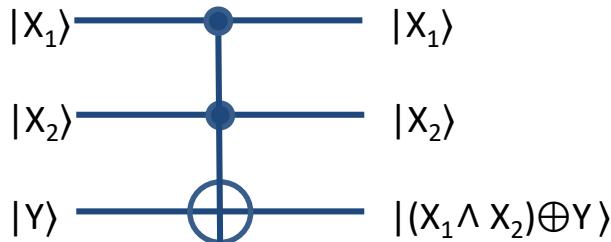
Kvantna prepletjenost in Bellovo stanje

$$\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

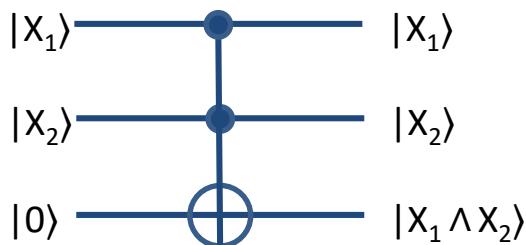
- Bellovega stanja ali stanja ERP ne moremo faktorizirati v tenzorski produkt posameznih kvantnih bitov (poskusite sami!).
- Pravimo, da je stanje kvantno prepleteno. Ko register izmerimo, dobimo s 50% verjetnostjo klasično vrednost 00, s 50% pa klasično vrednost 11. Kvantna bita sta torej prepletena in meritev enega pove vrednost drugega. Če izmerimo prvega in dobimo vrednost 0, potem je tudi drugi v vrednosti 0. Če pa je po meritvi prvi bit v vrednosti 1, je v vrednosti 1 tudi drugi bit.
- Zanimivost kvantnega prepleta je, da lahko bita ločimo v prostoru, pa bosta še vedno oba odreagirala na meritev enega.
- To pa ni edino možno Bellovo stanje. Vsa možna Bellova stanja dobimo, če na začetku predstavljenega vezja nastavimo vrednosti kvantnih bitov na $|00\rangle$, $|01\rangle$, $|10\rangle$ in $|11\rangle$.

Dvojna kontrolirana NE vrata: c^2 -NOT ali Toffoli

- c^2 -NOT ali vrata Toffoli:

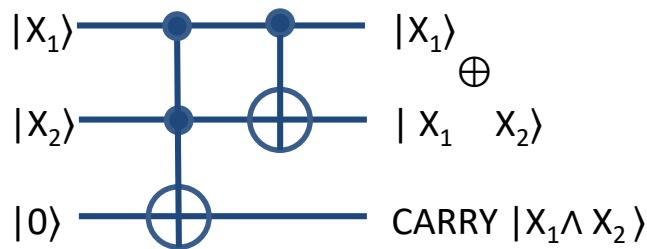


- reverzibilna vrata AND ($|y\rangle$ postavimo na $|0\rangle$):



Kvantna vrata: Univerzalen nabor

- S kvantnimi vrati NOT, AND in C-NOT lahko ovrednotimo katerokoli Booleovo funkcijo $\{0, 1\}^n \rightarrow \{0, 1\}^m$, ki preslika n vhodnih kvantnih bitov v m izhodnih kvantnih bitov.
- Ni nujno, da je takšno vezje učinkovito (merjeno v številu vrat, ki ga sestavljajo).



KVANTNO SEŠTEVANJE

Kvantna vrata: Literatura

Lep pregled kvantnih vrat, njihovih simbolov in njihovih unitarnih matrik najdete na Wikipediji:

- https://en.wikipedia.org/wiki/List_of_quantum_logic_gates



Fakulteta za elektrotehniko,
računalništvo in informatiko



Inštitut za računalništvo

Kvantno računalništvo in kriptografija

Avtor: izr. prof. dr. Aleš Holobar

Maribor, 2016

Univerza v Mariboru,

Fakulteta za elektrotehniko, računalništvo in informatiko

Avtor: dr. Aleš Holobar, univ. dipl. inž. rač. in inf., izredni profesor na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru.

Naslov: Kvantno računalništvo in kriptografija

Vrsta publikacije/gradiva: e-učbenik

Dostopno na: <https://dk.um.si>

Strokovna recenzenta:

red. prof. dr. Damjan Zazula, univ. dipl. inž. el., Fakulteta za elektrotehniko, računalništvo in informatiko Univerze v Mariboru

izr. prof. dr. Boštjan Šimunič, univ. dipl. inž. rač. in inf., Znanstveno-raziskovalno središče Univerze na Primorskem

Izdajatelj: Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za računalništvo

Kraj in leto izdaje: Maribor, 2016

ISBN 978-961-248-516-0



9 789612 485160

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

004.421.2:530.145.81(075.8)(0.034.2)

HOLOBAR, Aleš

Kvantno računalništvo in kriptografija [Elektronski vir] / avtor Aleš Holobar. - El. učbenik. - Maribor : Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za računalništvo, 2016

ISBN 978-961-248-516-0

COBISS.SI-ID [88168193](https://cobs.si/88168193)

Kazalo

Predgovor	iv
1 Turingov stroj in izračunljivost.....	1
1.1 Računska zahtevnost.....	3
2 Kompleksna števila	11
3 Uvod v kvantno mehaniko	16
3.1 Eksperiment z dvojno režo in opis kvantnih fizikalnih pojavov	16
3.2 Schrödingerjeva enačba.....	19
3.3 Interpretacije kvantne mehanike.....	26
3.3.1 Kopenhagenska interpretacija	27
3.3.2 Interpretacija več svetov.....	27
3.3.3 Interpretacija kvante informacije	28
3.4 Kvantna mehanika, linearost in amplitude verjetnosti	29
3.5 Kvantni bit	31
3.5.1 Ločljiva in prepletena stanja kvantnih bitov	32
3.6 Teorem neizvedljivosti kloniranja	33
4 Kvantno računalništvo.....	35
4.1 Klasični računalnik	35
4.2 Kvanti register	36
4.3 Kvantna vrata.....	38
4.3.1 Kvantna vrata NE	40
4.3.2 Hadamardova kvantna vrata.....	41
4.3.3 Kvantna fazna vrata.....	43
4.3.4 Nadzorovana NE-vrata	43
4.3.5 Nadzorovana U-vrata	45
4.3.6 Toffolijeva vrata	46
4.3.7 Univerzalen nabor kvantnih vrat	47
4.4 Kvantna Fourierova transformacija amplitud verjetnosti	48
4.5 Meritev.....	53
5 Kvantni algoritmi	57

5.1	Deutschov algoritem.....	58
5.1.1	Razлага algoritma.....	62
5.2	Groverjev algoritem.....	65
5.3	Shorov algoritem	73
5.3.1	Klasični del Shorovega algoritma	73
5.3.2	Shorov algoritem – kvanti del	76
6	Kriptografija in kvantno dešifriranje	83
6.1	Asimetrično šifriranje s faktorizacijo celih števil.....	84
6.1.1	RSA	85
6.1.2	Podpisovanje sporočil	87
6.1.3	Računska učinkovitost.....	87
6.2	Diskretni logaritem	89
6.2.1	Diffie–Hellmanov algoritem za izmenjavo ključev	90
6.3	Eliptične krivulje	91
6.3.1	Algoritem ECMQV	94
6.4	Bločni simetrični šifrirni algoritmi	95
6.4.1	Mreže za zamenjavo in permutacijo.....	96
6.4.2	Mreža Feistel	98
6.5	Pretočni simetrični šifrirni algoritmi	99
6.6	Kvantni algoritmi in napadi na asimetrične in simetrične šifrirne algoritme	101
6.7	Kvantna izmenjava skritega ključa.....	104
6.7.1	Protokol BB84.....	104
6.7.2	Protokol E91.....	105
7	Kvantni računalniki in kvantna omrežja	108
7.1	Ionske pasti	109
7.2	Druge izvedbe kvantnih računalnikov	115
7.2.1	Procesorji podjetja Dwave in adiabatno kvantno računanje	116
7.3	Kvantna omrežja.....	118
8	Viri in Literatura.....	119

Predgovor

Študijsko gradivo, zbrano v tem učbeniku, je povzeto iz mnogo virov. Veliko jih je navedenih v seznamu virov na koncu knjige, vseh, ki so vplivali na moje razumevanje kvantnega računalništva pa mi gotovo ni uspelo navesti, saj je moje znanje raslo skozi več let. Prav tako je gradivo, ki je zbrano v tej knjigi raslo skozi pet generacij študentov, ki so na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru poslušali predmet Nevro, nano in kvantno računalništvo. Vsaka generacija študentov je prispevala svoje kritike razlag in me silila v njihovo izpopolnjevanje, za kar sem jim iz srca hvaležen. Z njihovo pomočjo sem razlage računsko poenostavil in podkrepil s številnimi zgledi. Kvantno računalništvo sem skušal predstaviti v jeziku računalniškega inženirja, ki so mu čudesa kvantne mehanike neznana. Gradivo je torej namenjeno predvsem študentom računalništva in vsem, ki jih zanimajo kritike Turingove arhitekture, na kateri temelji večina sodobne komunikacijsko-informacijske infrastrukture. Gradivo ni namenjeno za poglobljen študij fizike in matematike, saj je uporabljen matematičen aparat namenoma močno poenostavljen in le redko preseže osnovne izreke kompleksne analize in linearne algebri.

V prvem poglavju podamo kratek pregled Turingove arhitekture in klasifikacije zahtevnosti računskih problemov. V drugem poglavju podamo osnovna pravila računanja s kompleksnimi števili, s katerimi kasneje v učbeniku izrazimo amplitude verjetnosti kvantnih stanj. Bralci z zadostnim predhodnim znanjem kompleksne analize lahko to poglavje preskočijo. V tretjem poglavju skušamo na preprost način pojasniti osnove kvantne mehanike. S pomočjo Schrödingerjeve enačbe vpeljemo valovno funkcijo in razložimo superpozicijo kvantnih stanj oziroma sočasnost vseh možnih dogodkov ter prepletost kvantnih stanj. V tem poglavju tudi povzamemo tri najpogosteje interpretacije kvantne mehanike, s katerimi lahko tolmačimo superpozicijo kvantnih stanj in vpliv meritve na kvantni sistem. V četrtem poglavju podamo opise kvantnih vrat, ki so osnovni gradnik kvantnih algoritmov. Pri opisih izhajamo iz dobro znanih definicij klasičnih logičnih vrat, ki sestavljajo klasične računalnike, pojasnimo pa tudi Hadamardova kvantna vrata, s katerimi opišemo prehod iz klasičnega stanja kvantnega bita v superpozicijo dveh klasičnih stanj. V petem poglavju podrobnejše predstavimo Deutschov, Groverjev in Shorov algoritem. Pri tem izhajamo iz definicij, ki smo jih podali v četrtem poglavju, zato je priporočljivo ti dve poglavji brati v predlaganem zaporedju. V šestem poglavju opišemo zasnove asimetričnih in simetričnih kodirnikov in njihovo ranljivost na kvantne algoritme, ki smo jih predstavili v petem poglavju. V sedmem poglavju podamo

poljuden opis trenutnih fizičnih izvedb kvantnih računalnikov, zlasti računalnikov, ki so implementirani s pomočjo ionskih pasti.

V času nastajanja tega gradiva se je Slovenija priključila veliki Evropski iniciativi, ki bo izdatno finančno podprla raziskave kvante mehanike in z njo povezanih prihodnjih in nastajajočih tehnologij [Tou2016], s tem pa razvitim državam, ki prepoznavajo pomen druge kvantne revolucije. Trenutno smo še vedno na njenem pragu in dovoljen nam je le bežen vpogled v tehnologije prihodnosti. Vendar vidimo dovolj, da vemo, da se bodo vzorci programiranja kvantnih računalnikov bistveno razlikovali od programskev, ki so jih generacije računalnikarjev v zadnjih desetletjih izpilile na klasični Turingovi arhitekturi. Kvantno računalništvo odpira nove miselne paradigmе in nove načine reševanja računskih problemov. Revidiralo bo tudi definicije izračunljivosti funkcij in rešljivosti računskih problemov, saj je bila izračunljivost definirana s pomočjo Turingove arhitekture, ki je dokazano manj učinkovita od kvantnih arhitektur. Številni problemi, ki so se zdeli s Turingovim strojem težko rešljivi, so na kvantni arhitekturi našli presenetljivo preproste rešitve. Takšno je tudi področje kriptografije, ki je bilo v zadnjem desetletju precej medijsko izpostavljeno in je ravno zaradi razvoja kvantnih algoritmov doživelо številne spremembe.

Trenutno ni verjeti, da bo mogoče s kvantnimi računalniki in algoritmi v doglednem času rešiti vsak problem. Tudi ni verjetno, da bodo kvantni računalniki nadomestili prenosne računalnike ali celo pametne telefone. Upamo, da bodo kmalu nadomestili velike superračunalnike in s tem bistveno izboljšali računsko moč človeške rase. Navadnemu uporabniku bodo torej verjetno dostopni preko storitev v oblaku. Do takrat je kljub izredno hitremu napredku še vsaj nekaj let, saj so današnje fizične izvedbe kvantnih računalnikov omejene na le okoli 15 kvantnih bitov. Večje sisteme je s trenutno tehnologijo težko vzdrževati v stanju superpozicije vseh možnih stanj, vsaj dovolj dolgo, da bi z njimi izvedli netrivialen izračun. Stik kvantnega sistema z zunanjim svetom pomeni izgubo superpozicije kvantnih stanj in je v fizikalnih krogih znan pod izrazom dekoherenca. Kot bomo videli v nadaljevanju, narava superpozicije vseh stanj in vloga meritve takšne superpozicije še vedno begata kvantne fizike in trenutno ni univerzalno sprejete interpretacije kvantnih pojavov.

V nasprotju s kvantnimi računalniki so kvantna omrežja dobro preizkušena in že nekaj let del vsakdanje stvarnosti. V sosednji Avstriji že vrsto let preizkušajo kvantno omrežje v dolžini nekaj sto kilometrov, postopoma pa so se jim pri testiranju pridružile tudi ostale razvite države. Številne vlade so prepoznale strateško pomembnost vlaganja v razvoj kvantnega računalništva in kvantne komunikacije in druga kvantna revolucija je pred vradi. Ali smo nanjo pripravljeni tudi v Sloveniji, pa bo pokazal čas.

Maribor, avgust 2016

1 Turingov stroj in izračunljivost

Leta 1936 je v angleščini beseda »computer« pomenila osebo, katere delo je bilo računanje s pisalom in papirjem [Aar2013]. Alan Turing je prvi postavil hipotezo, da lahko takšen poklic opravlja tudi stroj [Tur1950]. Dokazal je obstoj univerzalnega računskega stroja, ki je postal osnova sodobni informacijsko-komunikacijski tehnologiji in se po izumitelju imenuje Turingov stroj (slika 1.1).



Slika 1.1: Shematski prikaz Turingovega stroja z neskončno dolgim pomnilniškim trakom.

Turingov stroj sestavlja [Bar2016]:

- stroj, ki mora biti fizično izvedljiv, torej mora imeti končno število notranjih stanj;
- dolg, enodimensionalen trak, ki se lahko premika naprej in nazaj, pri čemer stroj z njega bere simbole in jih po potrebi tudi modificira;
- modifikacija simbola se izvrši izključno na podlagi prebranega simbola in notranjega stanja stroja.

Turingov stroj se je kasneje razvil v procesni element, neskončni pomnilniški trak pa v zaporedje pomnilniških elementov, ki jih še danes označujemo z linearo naraščajočimi pomnilniškimi naslovi.

Razvoj Turingovega stroja je vplival tudi na definicijo izračunljivosti funkcije oziroma odločljivosti problema:

- Funkcija je **izračunljiva** (angl. *computable*), če obstaja Turingov stroj, ki jo izračuna. V praksi to pomeni, da lahko sestavimo program ali podamo algoritem, ki računa vrednosti funkcije. Meje definicijskega območja določajo argumenti, pri katerih se Turingov stroj ustavi. Če želimo poudariti, da je izračunljiva funkcija povsod definirana, ji rečemo **popolnoma izračunljiva**.
- Problem je **odločljiv** ali **rešljiv**, če zanj obstaja algoritem ali Turingov stroj, ki ga reši. Če tak algoritem ne obstaja, je problem **nerešljiv** ali **neodločljiv**.

Turing je svoja dognanja o izračunljivosti funkcij strnil v dve tezi [New1955]:

Church-Turingova teza: Katerakoli funkcija, ki je naravno izračunljiva, je izračunljiva s Turingovim strojem.

S pojmom naravno izračunljiva funkcija navadno označujemo tiste funkcije, za katere ljudje menimo, da so izračunljive. Definicija izračunljivosti je torej antropocentrično obarvana. Trenutno Church-Turingova teza še vedno drži.

Kasneje je Turing svojo tezo razširil in z njo skušal oceniti učinkovitost svojega stroja.

Razširjena Church-Turingova teza: katerakoli funkcija, ki je naravno učinkovito izračunljiva, je učinkovito izračunljiva s Turingovim strojem.

V svojem članku z naslovom »Računski stroji in inteligenco«, ki je leta 1950 izšel v reviji Mind, je Turing napovedal [Tur1950]:

»Verjamem, da bo čez petdeset let možno programirati računalnike, ki bodo imeli 10^9 bitov spomina in bodo sposobni v odgovorih na vprašanja posnemati človeško bitje do takšne potankosti, da povprečni izpraševalec ne bo imel več kot 70 % možnosti, da po petih minutah izpraševanja razloči med strojem in človekom.«

Kasneje je ta napoved postala znana kot Turingov test inteligenčnosti stroja.

Leta 2000 smo res imeli računalnike z 10^9 bitov (GB) spomina, računalniško imitiranje človeškega bitja pa ni doseglo Turingove napovedi. Številni poskusi so privedli do bolj ali manj ponesrečenih poskusov, kjer so računalniki večinoma ponavljali vprašanja človeških izpraševalcev. Poskusi so porodili tudi nekaj ironije in kmalu se je pojavila hudomušna nadgradnja Turingovega testa, tako imenovani dopolnjen Turingov test, ki pravi, da moramo, če želimo preveriti inteligenco stroja, najprej zagotoviti minimalen nivo intelligence človeškega izpraševalca [Aar2013].

Z razvojem računalnikov je hitro postalo jasno, da je komunikacija med procesnim elementom in pomnilnikom ozko grlo Turingove arhitekture, in uvedene so bile številne tehnoške rešitve (predpomnilnik, cevovod, hitrejša sistemská vodila), ki bi to ozko grlo odpravile. Hkrati je z razvojem nevrofiziologije in slikovnih tehnik možganov dozorelo spoznanje, da narava ne loči med procesnim in pomnilnim elementom. Nevron, osnovna celica možganov, je pomnilni in procesni element hkrati. Čeprav ima odrasel človek nekaj milijard nevronov, možgani v povprečju za svoje delovanje porabijo le 20 W moči. Ta energetska učinkovitost presega sodobna elektronska vezja za nekaj velikostnih razredov. Človeške možgane odlikuje tudi velika robustnost in univerzalna sposobnost reševanja problemov. Človeška bitja in ostala živa bitja se precej učinkovito spopadamo z zelo različnimi izzivi zunanjega sveta in smo spodbjeni delovati v zelo kompleksnih okoljih. Če bi takšno univerzalnost hoteli doseči s Turingovim strojem, bi za to potrebovali superračunalnik,

ki bi porabil nekaj MW električne moči. Omenjena spoznanja so vodila v razvoj tako imenovane nevromorfne arhitekture, ki s pomočjo nanotehnologije posnema delovanje večjega števila nevronov. Njen razvoj so v zadnjem desetletju izdatno podprle številne vlade sveta in trenutno poteka velika bitka za uspešno simulacijo človeških možganov, v kateri sodelujejo predvsem znanstveniki iz Evrope in ZDA.

Drugi izziv učinkovitosti Turingove arhitekture je prišel s strani kvantne mehanike, relativno mlade, a zelo uspešne veje fizike, ki jo je leta 1905 s razlago kvantnega ustroja svetlobe zasnoval Albert Einstein. Leta 1982 je znani ameriški fizik Richard P. Feynman v članku z naslovom »Simulacije fizike z računalniki« [Fey1982], nakazal, da na Turingovi arhitekturi ni mogoče učinkovito simulirati kvantne mehanike. Spoznanje je obrnil v vprašanje: Če s Turingovo arhitekturo ni mogoče učinkovito simulirati kvantnih pojavov, kakšno učinkovitost bi lahko dosegel kvantni računalnik, ki bi izkorisčal princip superpozicije vseh možnih stanj in bi lahko, na primer, v kvantni register z osmimi biti shranil vseh 256 števil hkrati? Ta razprava je sprožila številne študije kvantnih računalnikov, ki pa niso bile izdatno finančno podprtne. Slednje se je spremenilo leta 1994, ko je ameriški matematik Peter Shor objavil kvantni algoritem, ki v polinomskem času poišče faktorje velikega sestavljenega celega števila. S tem je teoretično ogrozil praktično vse takrat znane šifrirne sisteme, ki temeljijo na asimetričnem šifriranju. Takšno šifriranje uporabljajo današnji računalniki za varovanje vseh bančnih transakcij in vseh podatkov, ki jih hranijo vlade, zavarovalnice, borze in podjetja. Shorov kvantni algoritem je povzročil skokovit porast financiranja razvoja kvantnih računalnikov in novih kriptografskih sistemov, ki bi bili varni pred kvantnimi algoritmi.

V nekaj letih so se finančna vlaganja v razvoj kvantnih računalnikov in algoritmov podeseterila in danes so številni kvantni znanstveniki prepričani, da smo že vstopili v drugo kvantno revolucijo. Prva kvantna revolucija je omogočila razvoj tranzistorjev in polprevodniških elementov, s katerimi gradimo današnje Turingove stroje. Čeprav delujejo zaradi zakonov kvantne mehanike, tranzistorji obdelujejo informacije na klasičen način, ki se od kvantnega precej razlikuje. Na primer, v klasičnem registru in klasičnem procesnem elementu lahko v izbranem trenutku hranimo oziroma obdelamo eno samo število. Druga kvantna revolucija se je pričela z izumom kvantnega računalnika in ostalih kvantnih vezij ter kvantnih algoritmov, ki tečejo na teh vezjih in izkorisčajo najvišjo mogočo paralelnost računalniških izračunov.

Razširjeno Church-Turingovo tezo sta torej izzvala in ovrgla kvantno in nevromorfno računalništvo in danes ne velja več.

1.1 Računska zahtevnost

S pomočjo Turingovega stroja je definirana tudi računska zahtevnost problemov. V grobem delimo probleme na lahko in težko izračunljive oziroma rešljive, težavnost računanja pa lahko merimo tako v času kot v pomnilniškem prostoru, ki je potreben za rešitev izbranega

problema. Časovno zahtevnost navadno opredelimo s številom računskih operacij, prostorsko pa s količino zasedenih pomnilniških zlogov.

Označimo z imenom $\text{ČAS}(f(n))$ (angl. $TIME(f(n))$) razred računskih problemov velikosti n , ki so rešljivi v računskem času oziroma številu računskih operacij $\alpha \cdot f(n)$, kjer je α konstanta. Podobno naj bo $PROSTOR(f(n))$ (angl. $SPACE(f(n))$) razred vseh problemov, rešljivih z uporabo pomnilnika, katerega velikost raste premo sorazmerno s funkcijo $f(n)$. Vemo, da je za vsako funkcijo $f(n)$ razred $\text{ČAS}(f(n))$ vsebovan v razredu $PROSTOR(f(n))$, saj lahko Turingov stroj dostopa do največ enega pomnilniškega elementa v enem časovnem koraku.

Časovno in prostorsko zahtevnost urejata dva teorema. **Teorem časovne hierarhije** pravi, da je več računskih problemov rešljivih v času $f(n) \cdot \log(f(n))$ kot v času $f(n)$. Podobno teorem **prostorske hierarhije določa**, da je več računskih problemov rešljivih s $f(n) \cdot \log(f(n))$ zlogi pomnilnika kot s $f(n)$ zlogi pomnilnika [Wal2016]. Teorema časovne in prostorske hierarhije torej opravičuje nadaljnjo klasifikacijo računskih problemov v več razredov računske zahtevnosti. Tako velja opozoriti, da ta delitev ni trivialna in da meje med posameznimi razredi niso dokončno raziskane. Ne vemo torej, ali se ti razredi med sabo dopolnjujejo ali pa je kateri izmed razredov samo podrazred izbranega nadrazreda. Podrobnejše bomo problem mej med razredi računske zahtevnosti osvetlili v nadaljevanju.

Podajmo hiter pregled osnovnih razredov računske zahtevnosti:

- **P** (polinomski angl. *polynomial*) je razred računskih problemov, ki so s Turingovim strojem rešljivi v polinomskem času. Z drugimi besedami, **P** je unija vseh razredov $\text{ČAS}(n^k)$, kjer je k poljubno naravno (celo in pozitivno) število.
- **PPROSTOR** (prostorsko polinomski angl. *polynomial space - PSPACE*) je razred problemov, ki so rešljivi s pomnilnikom polinomske velikosti (čas obdelave je neomejen!). Z drugimi besedami, **PPROSTOR** je unija vseh razredov $PROSTOR(n^k)$, kjer je k poljubno naravno število.
- **EKSP** (eksponenten angl. *exponential*) je razred vseh problemov, ki so rešljivi v eksponentnem času. Z drugimi besedami, EKSP je unija vseh razredov, $\text{ČAS}(2^{n^k})$, kjer je k poljubno naravno število. Dokazano je bilo, da je PPROSTOR vsebovan v razredu EKSP.
- **NP** (nedeterministično polinomski angl. *nondeterministic polynomial time*) je razred problemov, za katere lahko pravilnost odgovora preverimo v polinomskem času. Primer takšnega problema je faktorizacija celega števila n na produkt dveh praštevil. Ko imamo praštevila dana, lahko zelo hitro (v polinomskem času) preverimo, ali sta dve praštevili res faktorja števila n .

NP je vsebovan v **PPROSTOR**, saj lahko v pomnilniku polinomske velikosti preiščemo vse možne n^k dokazov, enega za drugim. Če je pravilen odgovor na zastavljenou vprašanje »da«,

potem bo to dokazal eden izmed dokazov. Če je odgovor »ne«, potem ne bo deloval noben dokaz.

Zagotovo je razred **P** vsebovan v razredu **NP**. Toda ali velja **P**=**NP**? Odgovor na to vprašanje še ni znan. Osrednji problem, ki ga bo moral rešiti kakršenkoli dokaz trditve $P \neq NP$, je ločitev problemov **NP**, ki so **resnično težki**, od tistih, ki so samo videti težki (morda samo ne poznamo preproste rešitve) [Aar2013].

- **NP-težek problem** (angl. *non-deterministic polynomial-time hard problem*) je nadrazred razreda **NP**. Problem B je **NP-težek**, če lahko nanj enakovredno in učinkovito preslikamo katerikoli **NP** problem. Z drugimi besedami, če imamo črno škatlo, ki v trenutku reši **NP-težek** problem B, potem lahko v polinomskem času rešimo katerikoli problem **NP**. **NP-težek** problem ni nujno v razredu **NP** in je lahko težji od najtežjih primerov v razredu **NP**.
- **NP-poln problem** (angl. *NP complete*): je razred problemov, ki so **NP-težki** in so v razredu **NP**. Če velja $P = NP$, potem velja tudi $P = NP\text{-poln} = NP$. Če velja $P \neq NP$, je **P** vsebovan v **NP**, hkrati pa je presek razredov **P** in **NP-poln** prazna množica. V vsakem primeru je **NP-poln** podmnožica razreda **NP**.

Primera **NP-polnih** problemov sta problem nahrbtnika (angl. *knapsack problem*) in problem trgovskega potnika (angl. *travelling salesman problem*).

Faktorizacija celega števila n ni **NP-poln** problem!

Do sedaj smo predpostavili, da je naš Turingov stroj determinističen, torej da vedno najde rešitev problema, če le ta obstaja. Z drugimi besedami, deterministični Turingov stroj se ustavi šele, ko najde rešitev. Kako pa je z razredi računske zahtevnosti, če je Turingov stroj stohastičen? V tem primeru govorimo o verjetnostnih algoritmih, torej o algoritmih, ki z določeno verjetnostjo rešijo zastavljeni problem. Verjetnostni algoritmi so navadno precej hitrejši od determinističnih, cena za njihovo hitrost pa je možnost, da so na zastavljeni vprašanju odgovorili napačno oziroma da so ponudili napačno rešitev zastavljenega problema. Rezultat verjetnostnih algoritmov moramo torej naknadno preveriti in po potrebi ponoviti tek verjetnostnega algoritma. Pri tem nam je v pomoč meja Chernoffa, poimenovana po ameriškem statistiku Hermanu Chernoffu [Che1981, Wal2016].

Meja Chernoffa (angl. *Chernoff bound*): Mečimo nepristranski kovanec n -krat in naj bo h število poskusov, pri katerih dobimo cifro. Potem velja

$$P \left[\left| h - \frac{n}{2} \right| \geq \alpha \right] \leq 2e^{-2\alpha^2 n}, \quad (1.1)$$

kjer je P verjetnost in je α poljubno pozitivno realno število.

Meja Chernoffa torej določa, da lahko s ponavljanjem poizkusa poljubno zmanjšamo verjetnost razhajanja med skupnim (povprečnim) rezultatom poskusov in dejansko rešitvijo

problema. Torej moramo ob nepravilnem rezultatu verjetnostnega algoritma samo dovoljkrat ponoviti tek algoritma, saj verjetnost, da algoritom ne bo vrnil pravega odgovora, pada eksponentno s številom tekov algoritma. Meja Chernoffa je ključnega pomena tudi v kvantnem računalništvu, saj so kvantni algoritmi večinoma verjetnostni (stohastični).

Verjetnostnih algoritmov ne smemo zamenjevati z aproksimacijskimi algoritmi, ki poiščejo bolj ali manj dober približek optimalne rešitve problema. Verjetnostni algoritmi dajejo eksaktne odgovore, vendar le z določeno verjetnostjo.

Razredi zahtevnosti verjetnostnih algoritmov so navadno zelo veliki. Na primer, razred **VP** (verjetnostni polinomski čas angl. *probabilistic polynomial-time* - *PP*) vsebuje vse odločljive probleme, za katere obstaja verjetnostni algoritem s polinomskim izvajalnim časom, ki odgovori pozitivno z verjetnostjo $> 1/2$, če je pravilni odgovor »da«, in odgovori pozitivno z verjetnostjo $\leq 1/2$, če je pravilni odgovor »ne«.

VP je zelo velik razred in vsebuje NP-polne probleme.

Zgled 1.1: Problem izpolnjevanja Boolove formule (angl. *boolean satisfiability problem*): dana je Boolova formula $F(x_1, x_2, \dots, x_n)$ z n Boolovimi spremenljivkami x_1, x_2, \dots, x_n . Ali lahko nastavimo vrednosti spremenljivk x_1, x_2, \dots, x_n tako, da bo $F(x_1, x_2, \dots, x_n)$ vrnila vrednost PRAVILNO (angl. TRUE)?

Problem izpolnjevanja Boolove formule je zelo znan NP-poln problem in je hkrati v razredu VP. Zanj lahko namreč zasnujemo naslednji algoritmom [Aar2013].

VP algoritem: vzemimo verjetnostni algoritem, ki za formulo $F(x_1, x_2 \dots x_n)$ naključno izbere vrednosti spremenljivk x_1, x_2, \dots, x_n . Potem algoritom preveri vrednost formule F . Če vrne F vrednost TRUE, vrne algoritrom odgovor »da«. Drugače vrne odgovor »da« z verjetnostjo $1/2$ in odgovor »ne« z verjetnostjo $1/2$. Na ta način bo algoritom vračal odgovor »da« z verjetnostjo $> 1/2$, če obstaja vsaj en nabor spremenljivk, za katere vrne F vrednost PRAVILNO, in z verjetnostjo $\leq 1/2$, če takšen nabor spremenljivk ne obstaja. Torej sodi opisani algoritmom v razred VP.

Razred VP je zaradi svoje splošnosti in velikosti pri klasifikaciji težavnosti problemov manj uporaben, zato so v preteklosti zasnovali manjši razred VPO.

VPO (verjetnostno polinomski z omejeno napako angl. *bounded-error probabilistic polynomial-time* - **BPP**) je razred odločljivih problemov, za katere obstaja naključni algoritem s polinomskim izvajalnim časom, ki vrne odgovor »da« z verjetnostjo $> 2/3$, če je pravilen

odgovor pozitiven, in z verjetnostjo $\leq 1/3$, če je pravilen odgovor negativen. Algoritem se torej zmoti z verjetnostjo $\leq 1/3$.

Če verjetnost napake $1/3$ ni dovolj majhna, lahko algoritem preprosto modifciramo tako, da se bo motil npr. z verjetnostjo $< 10^{-100}$. To dosežemo tako, da algoritem večkrat zaženemo. Če upoštevamo večinski odgovor T neodvisnih poskusov, potem nam meja Chernoffa pove, da se bomo motili z verjetnostjo, ki pada eksponentno hitro s številom poskusov T.

Predstavnik razreda VPO je Miller-Rabinov test praštevilskosti [Rab1980], ki za dano celo liho število n določi, ali je n praštevilo (soda števila so izpuščena, ker niso praštevila). Algoritem temelji na Fermatovem malem izreku (angl. *Fermat's little theorem*).

Fermatov mali izrek [Gra1975]: Če je n praštevilo, potem za vsako celo število a velja:

$$a^n \bmod n = a, \quad (1.2)$$

kjer smo z $\bmod n$ označili operacijo deljenja po modulu n . Enačbo (1.2) lahko zapišemo v malo spremenjeni obliki:

$$a^{n-1} \bmod n = 1 \quad (1.3)$$

ozziroma

$$(a^{n-1} - 1) \bmod n = 0. \quad (1.4)$$

Ponovimo, da je $n - 1$ sodo število, saj mora biti n liho število (v nasprotnem primeru n ni praštevilo, torej je test praštevilskosti nepotreben). Torej lahko zapišemo $n - 1 = 2^v \cdot d$, kjer je d liho število in je $v > 0$. Enačbo (1.4) lahko sedaj zapišemo kot

$$\begin{aligned} (a^{n-1} - 1) \bmod n &= (a^{2^v \cdot d} - 1) \bmod n = \\ &= ((a^{2^{v-1} \cdot d})^2 - 1) \bmod n = ((a^{2^{v-1} \cdot d} - 1)(a^{2^{v-1} \cdot d} + 1)) \bmod n = \\ &= ((a^{2^{v-2} \cdot d} - 1)(a^{2^{v-2} \cdot d} + 1)(a^{2^{v-1} \cdot d} + 1)) \bmod n = \\ &= ((a^{2^0 \cdot d} - 1)(a^{2^0 \cdot d} + 1) \cdots (a^{2^{v-2} \cdot d} + 1)(a^{2^{v-1} \cdot d} + 1)) \bmod n = 0. \end{aligned} \quad (1.5)$$

Torej mora za vsaj eden s , kjer je $0 < s < v$, veljati $a^{2^{s-1} \cdot d} \bmod n = -1$ ali pa mora veljati $a^d \bmod n = 1$.

Sedaj lahko Miller-Rabinov test praštevilskosti zapišemo z naslednjim psevdokodom:

Vhoda: liho število n in število tekov algoritma k .

Izhod: odgovor, da je n sestavljen število, ali pa odgovor, da je n verjetno praštevilo.

Postopek:

1. Število $n - 1$ zapišemo v obliki $n - 1 = 2^v \cdot d$, kjer je d liho število.
 2. Ponavljaj, dokler je $k > 0$
 - {
 3. Naključno izberemo pozitivno celo število $a \in [2, n - 2]$ in inicializiramo $s: s = 1;$
 4. Če $a^d \bmod n = 1$ ali $a^d \bmod n = -1$ potem dekrementiramo števec tekov $k = k - 1$ in se vrnemo na korak 2.
 5. Ponavljaj, dokler je $s < v$
 - {
 6. Če velja $a^{2^s \cdot d} \bmod n = 1$, potem je a je dokaz, da je n sestavljen število. Zaključimo izvajanje algoritma in vrnemo odgovor » n je sestavljen število».
 7. Če velja $a^{2^s \cdot d} \bmod n = -1$, potem dekrementiramo števec tekov $k = k - 1$ in se vrnemo na korak 2.
 8. Povečamo vrednost $s: s = s + 1$
 9. Vrnemo odgovor » n je sestavljen število».
10. Zaključimo izvajanje algoritma in vrnemo odgovor » n je verjetno praštevilo».

Zgled 1.2: Preverimo, ali je $n = 289$ praštevilo. Zapišimo $n - 1 = 288$ kot $288 = 2^5 \cdot 9$, tako da je $s = 5$ in $d = 9$. Izberemo naključno število $a = 251 < n - 2$ in izračunamo:

1. $a^{2^0 \cdot d} \bmod n = 251^9 \bmod 289 = 251 \neq 1$ ali -1
2. $a^{2^1 \cdot d} \bmod n = 251^{18} \bmod 289 = 288 \bmod 289 = -1$

Ker je $288 \bmod n = -1$, se moramo po navodilih koraka 7 vrniti na korak 2. Če smo število tekov nastavili na $k = 1$, zaključimo izvajanje algoritma (korak 10) in vrnemo ugotovitev, da je 289 verjetno praštevilo (v nasprotnem primeru pa je 251 močan lažnivec za število 289).

Poskusimo še eno naključno število $a = 171$:

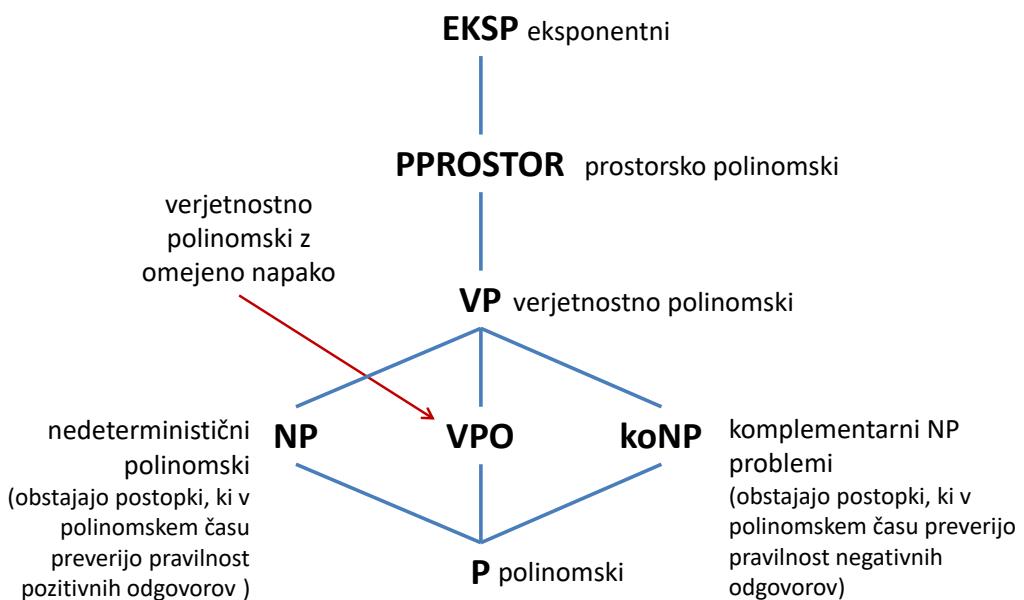
1. $a^{2^0 \cdot d} \bmod n = 171^9 \bmod 289 = 86 \neq 1$ ali -1
2. $a^{2^1 \cdot d} \bmod n = 171^{18} \bmod 289 = 171 \neq -1$
3. $a^{2^2 \cdot d} \bmod n = 171^{36} \bmod 289 = 52 \neq -1$
4. $a^{2^3 \cdot d} \bmod n = 171^{72} \bmod 289 = 103 \neq -1$
5. $a^{2^4 \cdot d} \bmod n = 171^{144} \bmod 289 = 205 \neq -1$

Prišli smo do koraka 9, torej je $a = 171$ priča, da je 289 sestavljeno število, $a = 251$ pa je bil močan lažnivec.

Miller-Rabinov test nam ne pove ničesar o faktorjih števila 289 (faktorja sta 17 in 17).

Podrobnejša analiza pokaže, da je v primeru, ko je n sestavljeno število, na intervalu $[2, n - 2]$ vsaj 75 % prič, da je n sestavljeno število [Con2016]. Torej se Miller-Rabinov test zmoti z verjetnostjo $\frac{1}{4^k}$, kjer je k število tekom algoritma. Pri petih tekih algoritma je verjetnost napake manjša od 0,001, pri desetih tekih pa je manjša od 0,000001.

Slika 1.2 prikazuje hierarhijo razredov računske zahtevnosti.



Slika 1.2: Hierarhija razredov računske zahtevnosti.

Relacije med razredi niso povsem znane.

Naloge:

1. Katera komponenta sodobnih računalnikih sistemov predstavlja Turingov stroj in katera predstavlja neskončen pomnilniški trak?
2. Kaj pravi prva Church-Turingova teza?
3. Zakaj mora imeti Turingov stroj končno število notranjih stanj?
4. Kaj pravi razširjena Church-Turingova teza?
5. Kaj določata teorema časovne in prostorske hierarhije?
6. Kaj je nedeterminističnega v razredu problemov NP?
7. Kateri problemi so težji, NP-težki ali NP-polni?
8. Kakšna je razlika med determinističnim in stohastičnim Turingovim strojem?
9. Kolikokrat moramo vreči kovanec, da bi z verjetnostjo 99,99 % ugotovili, ali je kovanec pristranski ali ne?
10. Če je n sestavljeno število, potem Miller–Rabinov test razglasiti, da je n verjetno praštevilo z verjetnostjo $\leq 4^{-k}$, kjer je k število ponovitev testa. Kolikokrat moramo test ponoviti, da bomo verjetnost napačnega odgovora zmanjšali pod prag 10^{-15} ?

2 Kompleksna števila

Kvantna mehanika in kvantno računalništvo temeljita na kompleksni analizi oziroma na algebri kompleksnih števil. Kompleksna števila razširjajo os realnih števil v dvodimenzionalno kompleksno ravnino. Za razliko od realnih števil so algebrajsko zaprta, kar pomeni, da je rezultat katerekoli računske operacije s kompleksnimi števili kompleksno število. Slednje ne velja za realna števila, saj $\sqrt{-1}$ ni realno število.

Kompleksno ravnino tvorita realna os (označimo jo z Re) in imaginarna os (označimo jo z Im). Enota realne osi je realno število 1, enota imaginarne osi pa število i , ki je definirano kot kvadratni koren števila -1 : $i = \sqrt{-1}$ (v literaturi je enota imaginarne osi večkrat označena s črko $j = \sqrt{-1}$, predvsem v elektrotehniki, kjer z »i« označujejo tok). Obe osi se sekata v številu 0 (koordinatnem izhodišču) in sta medsebojno pravokotni oziroma ortogonalni. Zaradi ortogonalnosti osi lahko poljubno kompleksno število z zapišemo kot kombinacijo odmika od števila 0 (koordinatnega izhodišča) za α enot po realni osi in za β enot po imaginarni osi, torej $z = \alpha + i\beta$.

Kompleksni števili $z_1 = \alpha + i\beta$ in $z_2 = \gamma + i\delta$ seštevamo in odštevamo tako, da ločeno seštejemo oziroma odštejemo realni in imaginarni komponenti števil:

$$\begin{aligned} z_1 + z_2 &= (\alpha + \gamma) + i(\beta + \delta), \\ z_1 - z_2 &= (\alpha - \gamma) + i(\beta - \delta). \end{aligned} \tag{2.1}$$

Množimo ju tako, da vsako komponento prvega števila pomnožimo z vsako komponento drugega števila:

$$z_1 \cdot z_2 = (\alpha + i\beta) \cdot (\gamma + i\delta) = \alpha\gamma + i\alpha\delta + i\beta\gamma - \beta\delta, \tag{2.2}$$

kjer smo upoštevali pravilo $i \cdot i = i^2 = -1$.

Podobno je deljenje kompleksnih števil definirano kot:

$$\frac{z_1}{z_2} = \frac{\alpha\gamma + \beta\delta + i(\beta\gamma - \alpha\delta)}{\gamma^2 + \delta^2}. \tag{2.3}$$

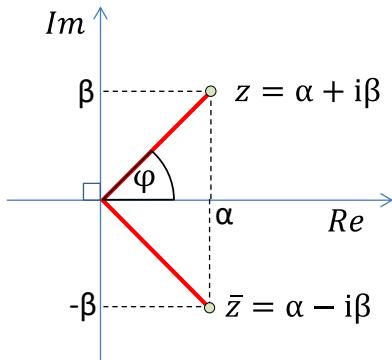
Kompleksna števila lahko tudi konjugiramo. Operacija kompleksnega konjugiranja spremeni predznak imaginarni komponenti kompleksnega števila:

$$z_1^* = \bar{z}_1 = \alpha - i\beta, \tag{2.4}$$

kjer smo operacijo kompleksnega konjugiranja označili z nadpisano $*$, konjugirano število z_1 pa z \bar{z}_1 . Kopleksno konjugiranje pogosto uporabimo pri izračunu absolutne vrednosti kompleksnega števila, ki je definirana kot

$$|z_1| = z_1 \cdot \bar{z}_1 = \sqrt{\alpha^2 + \beta^2}. \tag{2.5}$$

Absolutno vrednost kompleksnega števila izračunamo torej tako, da po Pitagorovem izreku seštejemo kvadrata obeh komponent kompleksnega števila. Izračunana absolutna vrednost predstavlja razdaljo kompleksnega števila z_1 od števila 0 oziroma od koordinatnega izhodišča kompleksne ravnine. Po definiciji je absolutna vrednost realno nenegativno število in je enaka 0 samo, ko je $z_1 = 0$. Velja tudi $|z_1| = |\bar{z}_1|$.



Slika 2.1: Predstavitev kompleksnega števila $z = \alpha + i\beta$ in njegove konjugirane vrednosti $\bar{z}_1 = \alpha - i\beta$ v kompleksni ravnini. Z rdečo črto je predstavljena razdalja obeh števil od koordinatnega izhodišča. Ta razdalja je po definiciji enaka absolutni vrednosti števila z .

Do sedaj smo spoznali zapis kompleksnega števila v karteziskem koordinatnem sistemu. Vsako kompleksno število lahko zapišemo tudi v polarnem koordinatnem sistemu, torej z radijem in kotom. Pri pretvorbi iz karteziskskega v polarni koordinatni sistem radij r kompleksnega števila izračunamo kot absolutno vrednost kompleksnega števila

$$r = |z_1| = z_1 \cdot \bar{z}_1 = \sqrt{\alpha^2 + \beta^2}, \quad (2.6)$$

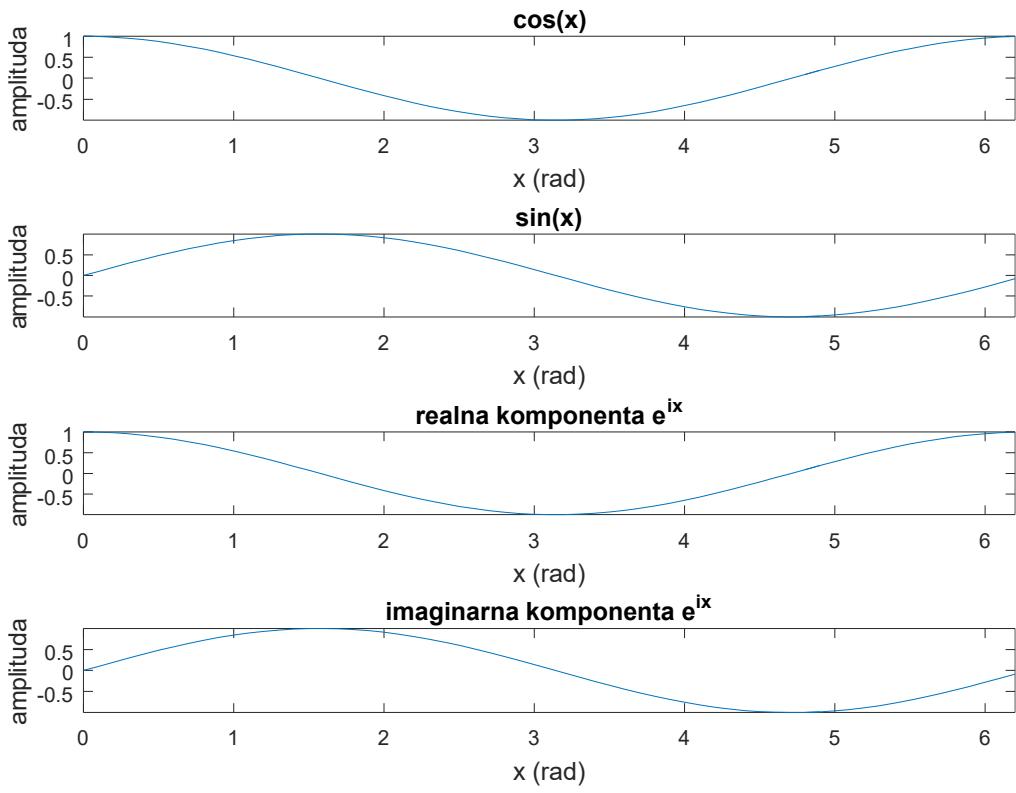
kot φ pa kot

$$\varphi = \arctg\left(\frac{\beta}{\alpha}\right). \quad (2.7)$$

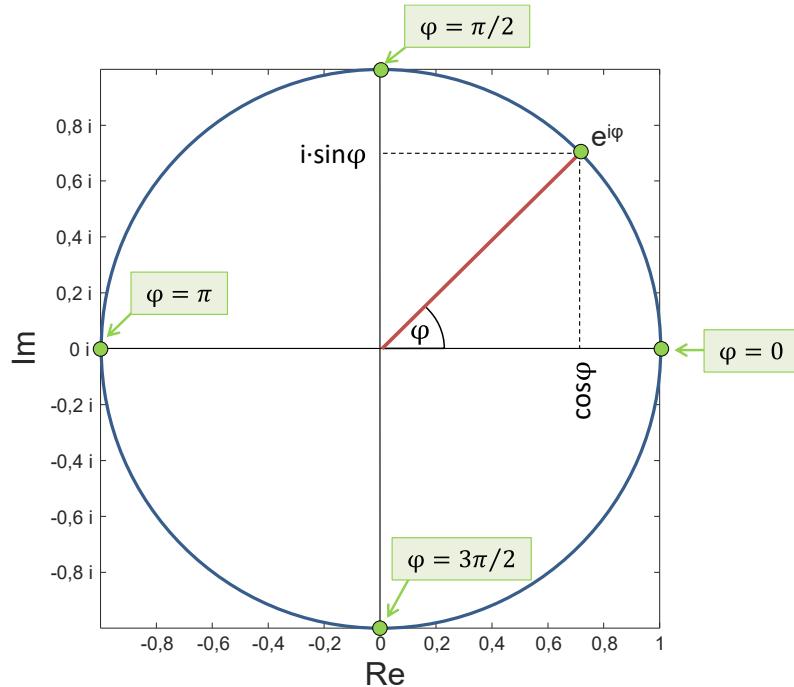
Pretvorbo iz polarnega v karteziski koordinatni sistem zapišemo kot

$$z = r \cdot (\cos\varphi + i \cdot \sin\varphi). \quad (2.8)$$

Posebno vlogo v kompleksni analizi, v Fourierovi analizi in v kvantnem računalništvu ima eksponentna funkcija. Poznamo jo iz teorije realnih števil, v kompleksni ravnini pa so njene lastnosti razširjene. Z namenom, da bi bolje spoznali te lastnosti, moramo pogledati obnašanje eksponentne funkcije v kompleksni ravnini. Slika 2.2 prikazuje vrednosti funkcije e^{ix} na intervalu $[0, 2\pi]$. Vidimo, da se lahko realna in imaginarna komponenta funkcije e^{ix} izrazita s funkcijama sinus in kosinus.



Slika 2.2: Vrednosti funkcije kosinus v kompleksni ravnini.



Slika 2.3: Vrednosti funkcije $e^{i\varphi}$ v kompleksni ravnini.

Ko realno število φ zavzame vrednosti od 0 do 2π , opiše funkcija $e^{i\varphi}$ v kompleksni ravnini enotin krog. Slike 2.3 sledi Eulerjeva enakost:

$$e^{i\varphi} = \cos\varphi + i \cdot \sin\varphi \quad (2.9)$$

in kompleksno število z lahko iz polarnih v kartezijiske koordinate pretvorimo tudi na naslednji način:

$$z = r \cdot e^{i\varphi}. \quad (2.10)$$

Naloge:

1. Izračunajte naslednje izraze in jih narišite v kompleksni ravnini:

- | | | |
|------------------------|-------------------------|--------------------------|
| a) $3 + (1 - i)$ | b) $3i + (2 - 2i)$ | c) $(5 + 3i) + (-2 + i)$ |
| d) $(1 + i) + (1 - i)$ | e) $(3 + i) + (2 - 2i)$ | f) $(3 + i) + (3 + i)$ |

2. Izračunajte naslednje izraze:

- | | | |
|----------------------------|-----------------------------|------------------------------|
| a) $3 \cdot (1 - i)$ | b) $3i \cdot (2 - 2i)$ | c) $(5 + 3i) \cdot (-2 + i)$ |
| d) $(1 + i) \cdot (1 - i)$ | e) $(3 + i) \cdot (2 - 2i)$ | f) $(3 + i) \cdot (3 + i)$ |

3. Izračunajte naslednje izraze:

- | | | |
|----------|----------|----------|
| a) i^2 | b) i^3 | c) i^4 |
| d) i^5 | e) i^6 | f) i^7 |

4. Izračunajte naslednje izraze:

- | | | |
|----------------|----------------|----------------|
| a) $(1 - i)^2$ | b) $(1 - i)^3$ | c) $(1 - i)^4$ |
| d) $(1 - i)^5$ | e) $(1 - i)^6$ | f) $(1 - i)^7$ |

5. Izračunajte naslednje izraze:

- | | | |
|----------------------|-----------------------|------------------------|
| a) $3/(1 - i)$ | b) $3i/(2 - 2i)$ | c) $(5 + 3i)/(-2 + i)$ |
| d) $(1 + i)/(1 - i)$ | e) $(3 + i)/(2 - 2i)$ | f) $(3 + i)/(3 + i)$ |

6. Izračunajte naslednje izraze:

- | | | |
|-----------------|-----------------|-----------------|
| a) $\sqrt{-1}$ | b) $\sqrt{-4}$ | c) $\sqrt{-9}$ |
| d) $\sqrt{-16}$ | e) $\sqrt{-32}$ | f) $\sqrt{-2i}$ |

7. Izračunajte naslednje izraze in jih narišite v kompleksni ravnini:

a) $|1 - i|$

b) $|3i|$

c) $|5 - 3i|$

d) $|1 + i|$

e) $|2 - 2i|$

f) $|3 + i|$

8. Konjugirajte naslednja kompleksna števila in jih narišite v kompleksni ravnini:

a) $1 + i$

b) $3 + i$

c) $5 + 3i$

d) $1 - 2i$

e) 2

f) i

9. Pretvorite v polarne koordinate:

a) $1 - i$

b) $3i$

c) $5 + 3i$

d) $1 + i$

e) $2 - 2i$

f) $3 + i$

10. Pretvorite v kartezijiske koordinate:

a) $r = 1, \varphi = 0$

b) $r = 1, \varphi = \pi$

c) $r = 0, \varphi = \pi/2$

d) $r = 1, \varphi = \pi/2$

e) $r = 1, \varphi = -\pi/2$

f) $r = 5, \varphi = \pi/2$

11. Izračunajte naslednje izraze in jih narišite v kompleksni ravnini:

a) $e^{i\pi}$

b) $e^{-i\pi}$

c) $e^{-i2\pi}$

d) $e^{-i\pi/2}$

e) $e^{-i5\pi}$

f) $e^{-i3\pi/2}$

12. Izračunajte naslednje izraze in jih narišite v kompleksni ravnini:

a) $e^{i\pi} \cdot e^{i\pi}$

b) $e^{i\pi} \cdot e^{-i\pi}$

c) $e^{i\pi} \cdot e^{i0}$

d) $e^{i\pi/3} \cdot e^{i2\pi/3}$

e) $e^{-i\pi/4} \cdot e^{-i\pi/4}$

f) $e^{i\pi/2} \cdot e^{i\pi/2}$

3 Uvod v kvantno mehaniko

Klasični računalniki, ki temeljijo na Turingovi arhitekturi, izdatno uporabljajo tranzistorje. Njihova konstantna miniaturizacija omogoča povečevanje njihovega števila na prostorninsko enoto in je eno izmed osnovnih gonil napredka sodobnih procesorjev. V zadnjih letih so tranzistorji dosegli nivo velikosti nekaj nm. Na tem nivoju prevladujejo učinki kvantne mehanike, ki se od naše, na izkustvih makro sveta temelječe intuicije precej razlikujejo. Zgodbe, ki jih piše kvantna mehanika, presegajo domišljijo marsikaterega inženirja in ni malo strokovnjakov, ki vztrajno zavračajo osnovne postulate kvantnega sveta. A čeprav je v svojem bistvu samo matematičen formalizem, ki se dotika samih mej našega dojemanja ustroja narave, je kvantna mehanika prestala stoletje znanstvenih preverjanj, njene teoretične napovedi pa se vedo znova potrjujejo s številnimi novimi eksperimentalnimi odkritji.

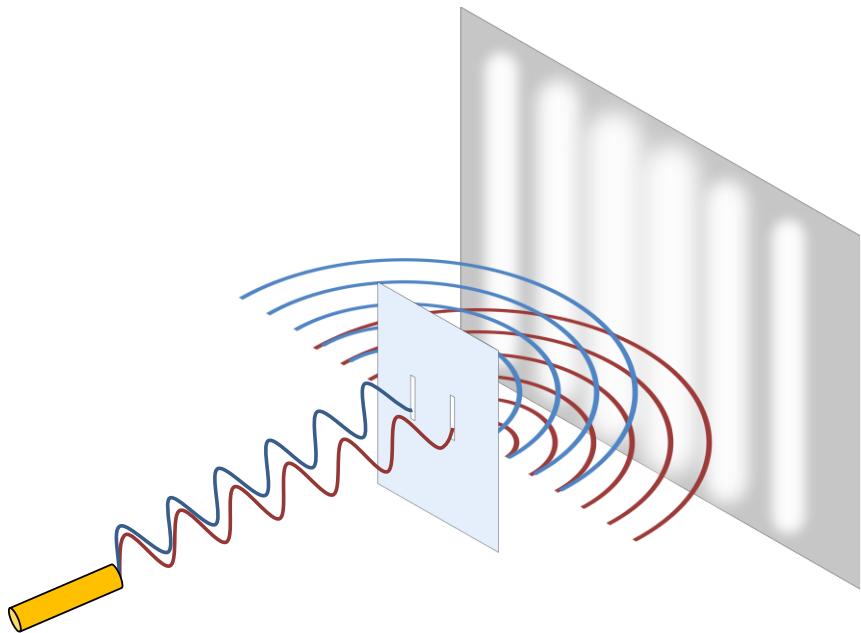
V tem poglavju bomo spoznali nekaj osnovnih lastnosti kvantnega sveta. Dotaknili se jih bomo s perspektive računalništva, zato ne bomo zahajali v striktne fizikalne opise, še manj v matematične izpeljave. Namesto tega bomo skušali opisati splošne principe, ki omogočajo zasnovno in razvoj kvantnih algoritmov.

3.1 Eksperiment z dvojno režo in opis kvantnih fizikalnih pojavov

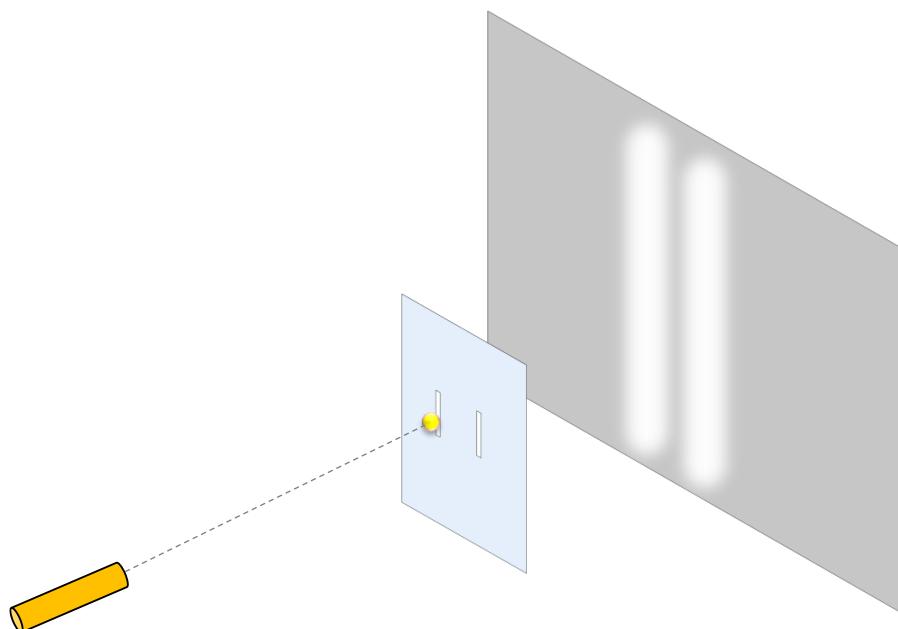
Eksperiment z dvojno režo (angl. *double-slit experiment*) je eden izmed najslavnnejših eksperimentov kvantne mehanike. Demonstrira namreč dualno naravo svetlobe, ki se včasih obnaša kot valovanje, včasih pa kot kvanti delci (fotoni). Še vedno ne razumemo te dualnosti in eksperiment z dvojno režo lepo ponazorji omejitve našega dojemanja sveta [Bac2013].

Dan naj bo izvor koherentnega svetlobnega valovanja, torej svetlobnega valovanja, ki ima konstanto frekvenco (npr. laser). S koherentnim svetlobnim valovanjem osvetlimo ploščo z dvema režama (slika 3.1). Za ploščo z režama namestimo zaslon in na njem opazujemo vzorec, ki ga tvori svetloba, ki prehaja skozi reži. Če je svetloba valovanje, bo na zaslonu nastal interferenčni vzorec, ki bo sestavljen iz več svetlih in temnih prog (slika 3.1). Po prehodu skozi rež se namreč svetlobno valovanje siplje - širi se v krogih, kot valovanje v vodi, ko vanjo vržemo kamen. Valovanji obeh rež se v prostoru prepletata in v točkah zaslona, kjer se valovanji fazno ujameta, nastanejo svetle (osvetljene) proge, v točkah zaslona, kjer se valovanji srečata z nasprotnima fazama, pa nastanejo temne (neosvetljene) proge.

Svetlobo lahko obravnavamo tudi kot skupek kvantnih delcev, fotonov. Omenjeno obravnavo je leta 1905 prvi uvedel Einstein in za razlogo kvantnega fotoelektričnega efekta prejel Nobelovo nagrado za fiziko. Kasneje so kvantni ustroj narave potrdile številne eksperimentalne študije. Toda če je svetloba sestavljena iz delcev (fotonov), se bo na našem zaslonu pojavit odsev obeh rež (slika 3.2) in ne interferenčni vzorec.



Slika 3.1: Eksperiment z dvojno režo – svetloba kot valovanje.

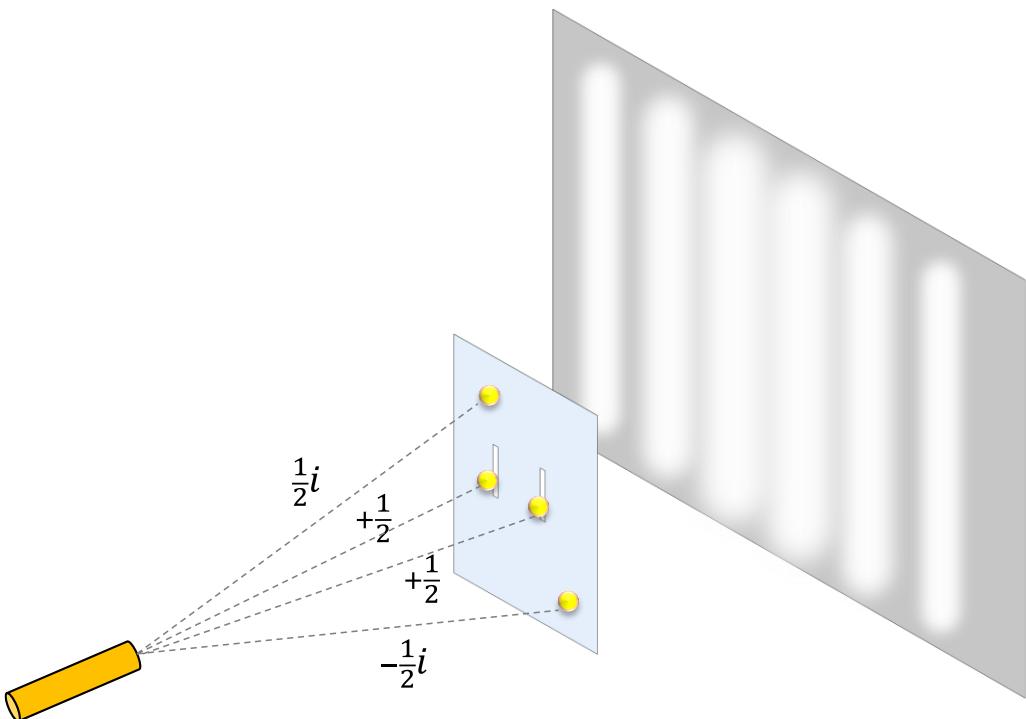


Slika 3.2: Eksperiment z dvojno režo – svetloba kot klasični delec.

Izkaže se, da lahko v opisanem eksperimentu izmerimo ali interferenčni vzorec (v tem primeru je svetloba valovanje) ali pa odsev dveh rež (v tem primeru je svetloba skupek

delcev), rezultat pa je odvisen od meritve prehoda svetlobe skozi reži. Kakor hitro v našem eksperimentu z dvojno režo svetlobo obravnavamo kot skupek fotonov in s posebnim detektorjem izmerimo, skozi katero režo je potoval foton, interferenčni vzorec izgine in na zaslonu se pojavit samo dve osvetljeni proggi (slika 3.2). V tem primeru se torej svetloba prične obnašati kot skupek klasičnih delcev. Ko detektor fotonov odstranimo in ne obstaja informacija o tem, skozi katero režo je potoval posamezen foton, se interferenčni vzorec pojavi in svetloba se obnaša kot valovanje (slika 3.1).

Eksperiment z dvojno režo razkriva odločilen vpliv meritve. Ko ne merimo, skozi katero režo je šel posamezen foton, se na opazovanem zaslonu pojavi več svetlobnih prog, ki tvorijo vzorec interference dveh valovanj. Interferenčni vzorec se pojavi tudi, ko s posebnim izvorom streljamo en sam delec naenkrat [Bac2013]. Iz opisanega sledi, da je delec šel skozi obe reži hkrati in interferiral sam s sabo. Še več, kvantna mehanika pravi, da je šel delec hkrati skozi eno in skozi drugo, skozi obe in skozi nobeno izmed rež (slika 3.3). Vsi ti dogodki so se zgodili sočasno. Pravimo, da je bil delec v superpoziciji vseh možnih stanj. A samo če ne izmerimo, skozi katero izmed rež je šel. Tako ko izmerimo prehod delca skozi izbrano režo, dobimo odgovor, da se je zgodil le eden izmed zgoraj naštetih sočasnih dogodkov, in interferenčni vzorec na zaslonu izgine (slika 3.2).



Slika 3.3: Eksperiment z dvojno režo – svetloba kot kvantni delec. Delec je sočasno na več mestih in interferira sam s sabo. A takoj ko izmerimo, kje je, se obnaša kot klasičen delec (slika 3.2). Na sliki so prikazane tudi amplitude verjetnosti za položaj delca.

Sorodni eksperiment, znan pod imenom kvantni izbris (angl. quantum eraser), razkriva še večjo bizarnost kvantnega sveta [Wal2002, Yoo2000, Xia2016]. Informacijo o preletu delca skozi režo lahko z instrumenti izmerimo tudi po prehodu delca skozi režo. Prav tako z instrumenti izmerimo informacijo o vzorcu na zaslonu ter obe informaciji shranimo, ne da bi se z njima seznanili. Ko informacijo o preletu delca skozi režo trajno izbrišemo in nato preverimo informacijo o vzorcu na merilnem zaslonu, se na zaslonu pojavi interferenčni vzorec. Če pa meritve preleta delca skozi režo ne izbrišemo (ni pomembno, ali se z njo seznanimo ali ne), se ob preverbi podatkov merilnega zaslona izkaže, da sta se na zaslonu pojavili dve osvetljeni progi in ne interferenčni vzorec. Ali torej meritev učinkuje retrospektivno (sedanjost lahko spreminja preteklost) ali pa je vesolje že vnaprej izračunalo, katere meritve oziroma informacije bomo izbrisali, katerih pa ne. Prvi primer nasprotuje našemu razumevanju prostor-časa, v slednjem primeru pa se moramo sprijazniti, da ni svobodne volje. Je le njena iluzija.

3.2 Schrödingerjeva enačba

Schrödingerjeva enačba je linearne parcialne diferencialna enačba, ki opisuje spremembe kvantnega sistema skozi čas. Pojasnjuje kvantno dualnost delec-valovanje in opisuje stanje sistema kot valovno funkcijo verjetnostne porazdelitve posamezne lastnosti sistema. Leta 1926 jo je objavil avstrijski fizik Erwin Schrödinger [Sch1926]. V svoji najbolj splošni obliki je definirana kot

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = \hat{H}\Psi(\mathbf{r}, t), \quad (3.1)$$

kjer je i imaginarna enota ($i = \sqrt{-1}$), \hbar je Planckova konstanta, deljena z 2π ($\hbar = \frac{\hbar}{2\pi}$), $\frac{\partial}{\partial t}$ označuje parcialni odvod po času, $\Psi(\mathbf{r}, t)$ je valovna funkcija v prostoru \mathbf{r} in času t , \hat{H} pa je Hamiltonov operator, ki opredeljuje celotno energijo sistema (navadno kinetično in potencialno energijo) in je definiran za vsak sistem posebej (glejte zgleda 3.1 in 3.2).

Valovna funkcija $\Psi(\mathbf{r}, t)$ podaja kompleksne amplitude verjetnosti v prostor-času, iz katerih lahko izračunamo, s kakšno verjetnostjo bomo izmerili posamezna stanja kvantnega sistema. Razliko med amplitudo verjetnosti in verjetnostjo bomo podrobnejše razložili v podpoglavlju 3.4. Sedaj samo opozorimo, da na amplitudo verjetnosti ne smemo gledati kot na verjetnost dogodka v prostor-času. Amplituda verjetnosti opisuje **sočasnost dogodkov v prostor-času**. Res pa je, da lahko iz amplitud verjetnosti izračunamo verjetnost, da bomo nekje v prostor-času izmerili izbrani dogodek.

Vlogo valovnih funkcij pri opisu kvantnega sistema navadno razložimo s preprostim zgledom. Imejmo sistem z enim samim delcem, ki se giblje v enodimensionalnem prostoru. V tem primeru opišemo z valovno funkcijo verjetnosti vseh pozicij delca v času t :

$$p(r, t) = |\Psi(r, t)|^2 = \Psi(r, t) \cdot \Psi^*(r, t), \quad (3.2)$$

kjer je $p(r, t)$ gostota verjetnosti, da smo delec v času t izmerili v poziciji r , z zvezdico * pa smo označili operacijo kompleksnega konjugiranja (konjugacija spremeni predznak imaginarnega dela kompleksnega števila, torej število $\alpha + i\beta$ spremeni v število $\alpha - i\beta$).

Verjetnost, da se bo delec nahajal na intervalu med točko A in B, zapišemo kot:

$$P(A \leq r \leq b) = \int_A^B p(r, t) dr = \int_A^B |\Psi(r, t)|^2 dr. \quad (3.3)$$

Veljati mora tudi

$$\int_{-\infty}^{\infty} p(r, t) dr = 1, \quad (3.4)$$

saj se delec, ki ga lahko izmerimo, nahaja nekje v našem 1D-prostoru. Valovna funkcija $\Psi(r, t)$ mora torej biti normirana

$$\Psi(r, t) = \frac{\Psi(r, t)}{\int_{-\infty}^{\infty} |\Psi(r, t)|^2 dr}, \quad (3.5)$$

kar pomeni, da mora integral konvergirati na intervalu od $-\infty$ do ∞ .

Opazovani delec ima tudi gibalno količino g (angl. *momentum*), ki jo tako kot pozicijo v prostoru izrazimo z valovno funkcijo $\Psi_g(r, t)$. Izkaže se, da sta valovni funkciji pozicije in gibalne količine povezani s Fourierovo transformacijo (valovna funkcija pozicije je Fourierova transformiranka valovne funkcije gibalne količine in obratno) in da v popolnosti določata druga drugo (Fourierovo transformacijo razložimo v poglavju 4).

Zgled 3.1: Delec, ki se prosto giblje v enodimenzionalnem prostoru

Predpostavimo, da se delec giblje po osi x v 1D-prostoru, ima maso m , gibalno količino p in potencialno energijo V . V tem primeru je Hamiltonov operator definiran kot [Gri2004, Cre2005]:

$$\hat{H} = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x, t). \quad (3.6)$$

Pri tem smo predpostavili, da je potencialna energija funkcija prostora in časa.

Pozicijo delca opisuje Schrödingerjeva enačba (3.1):

$$i\hbar \frac{\partial}{\partial t} \Psi(x, t) = \left(-\frac{\hbar^2}{2m} + V(x, t) \right) \Psi(x, t) = -\frac{\hbar^2}{2m} \frac{d^2 \Psi(x, t)}{dx^2} + V(x, t) \Psi(x, t). \quad (3.7)$$

Za delec, ki se giblje prosto po prostoru (ne čuti zunanjega polja), je $V(x, t) = 0$ in enačba (3.7) se poenostavi v

$$i\hbar \frac{\partial}{\partial t} \Psi(x, t) = -\frac{\hbar^2}{2m} \frac{d^2 \Psi(x, t)}{dx^2}. \quad (3.8)$$

Celotna energija prosto gibajočega se delca je torej enaka njegovi kinetični energiji

$$E = \frac{1}{2}mv^2 = \frac{p^2}{2m}, \quad (3.9)$$

kjer je $p = mv$ gibalna količina delca. Če delec obravnavamo kot valovanje in upoštevamo Planckovo hipotezo, lahko celotno energijo delca izrazimo tudi kot

$$E = \hbar\omega, \quad (3.10)$$

kjer je ω krožna frekvenca valovanja, ($\omega = 2\pi f$, pri čemer je f frekvenca valovanja). Upoštevamo še DeBroglievo hipotezo in gibalno količino delca p zapišemo kot

$$p = h/\lambda = hk/2\pi = \hbar k, \quad (3.11)$$

kjer je $k = 2\pi/\lambda$. Potem ima rešitev diferencialne enačbe (3.8) splošno obliko

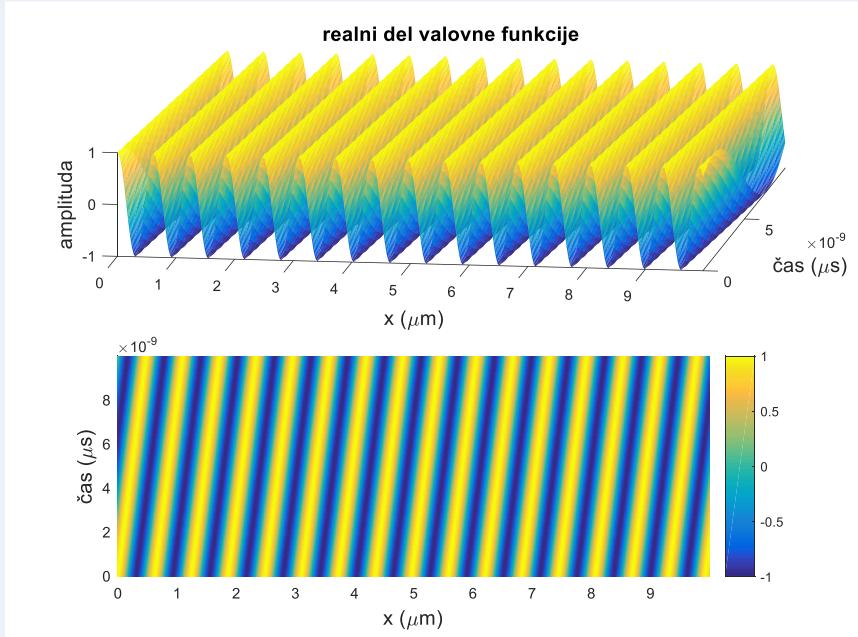
$$\Psi(x, t) = Ae^{i(kx - \omega t)}, \quad (3.12)$$

kjer je A poljubna konstanta.

Preverimo: Velja $\frac{\partial}{\partial t}\Psi(x, t) = -i\omega\Psi(x, t)$ in $\frac{d^2}{dx^2}\Psi(x, t) = -k^2\Psi(x, t)$ in Schrödingerjeva enačba (3.8) se poenostavi v enačbo za celotno energijo delca:

$$\hbar\omega = \frac{\hbar^2 k^2}{2m} = \frac{p^2}{2m} = E. \quad (3.13)$$

Spremembe valovne funkcije $\Psi(x, t)$ skozi prostor x in čas t prikazuje slika 3.4. Ponovimo, da opisuje valovna funkcija sočasnost dogodkov v prostor-času. To lastnost smo opisali že pri eksperimentu z dvojno režo in prav ta lastnost je ključnega pomena za obstoj kvantnih registrov in kvantnih algoritmov.



Slika 3.4: Valovna funkcija $\Psi(x, t)$ kvantnega delca, ki se prosto giblje v enodimensionalnem prostoru.

Zgled 3.2: Delec, ki se prosto giblje na omejenem prostoru

Zgled 3.1 podaja preprost primer uporabe Schrödingerjeve enačbe, vendar je praktično neizvedljiv. Predvideva namreč, da se lahko delec prosto giblje po celotnem vesolju, ne da bi kadarkoli prišel v interakcijo s katerimkoli delcem ali poljem. Zato je pozicija delca nedoločljiva, na kar nakazuje tudi dejstvo, da valovna funkcija v enačbi (3.12) ne izpolni enačbe (3.4), saj integral $\int_{-\infty}^{\infty} |\Psi(x, t)|^2 dx$ ne konvergira.

Veliko bolj realističen je zgled, kjer je prosto gibanje delca v enodimensionalnem prostoru omejeno na interval dolžine D . V tem primeru je njegova valovna funkcija pozicije podana z enačbo (3.7), kjer je

$$V(x, t) = \begin{cases} 0, & x_c - D/2 < x < x_c + D/2 \\ \infty, & \text{drugače} \end{cases} \quad (3.14)$$

in je x_c sredina intervala. V tem primeru enačba (3.9) ne drži več, saj je sedaj celotna energija delca seštevek njegove kinetične in potencialne energije $V(x, t)$. Upoštevati moramo tudi, da se delec ne more nahajati izven opazovanega intervala, torej mora biti verjetnost $p(x, t) = |\Psi(x, t)|^2$, da najdemo delec izven podanega intervala, enaka 0. Valovna funkcija tudi ne sme imeti nenadnih skokov vrednosti [Dav2006], zato mora imeti relativno nizke absolutne vrednosti tudi na mejah opazovanega intervala. Izpeljava valovne funkcije pozicije delca je v tem primeru torej malo težja kot v zgledu 1, zato jo bomo na tem mestu izpustili in podali samo njen končen rezultat [Gri2004].

$$\Psi(x, t) = \begin{cases} A \sin(k_n(x - x_c + D/2)) e^{-i\omega_n t}, & x_c - D/2 < x < x_c + D/2 \\ 0, & \text{drugače} \end{cases}, \quad (3.15)$$

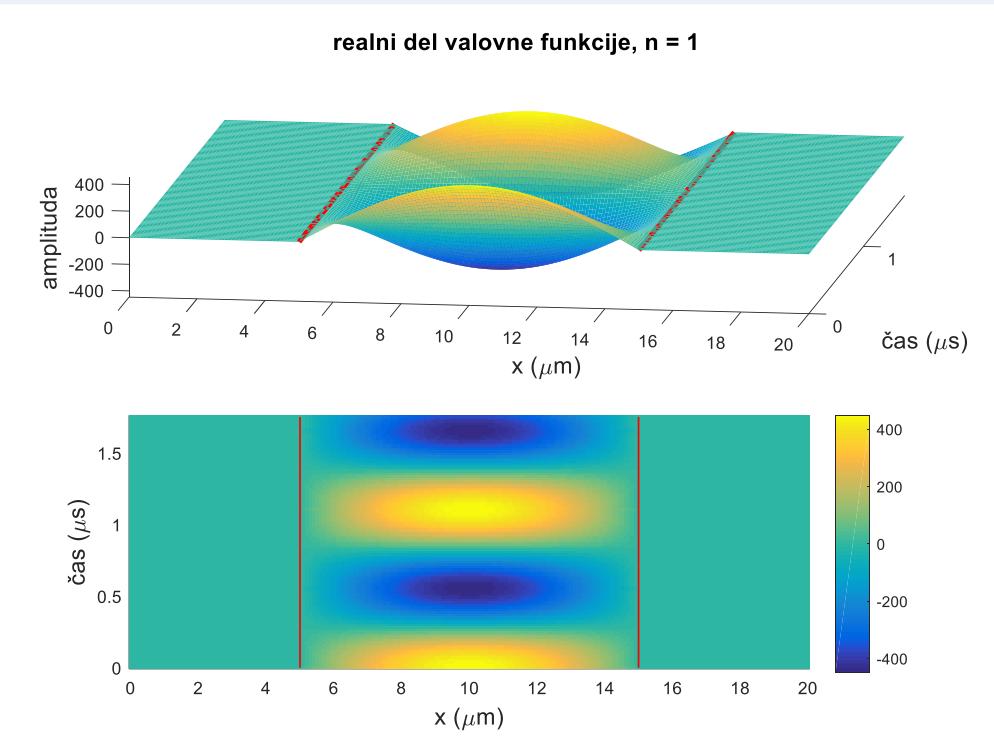
kjer je A poljubna konstanta (kompleksno število), ki izpolnjuje pogoj $|A| = \sqrt{\frac{2}{D}}$, in $k_n = \frac{n\pi}{D}$ ter $\omega_n = \frac{n^2\pi^2\hbar}{2mD^2}$, pri čemer je n pozitivno celo število ($n=1,2,3\dots$). Izkaže se, da n določa diskretne energijske nivoje delca [Gri2004, Cre2005]:

$$E_n = \frac{n^2\hbar^2}{8mD}. \quad (3.16)$$

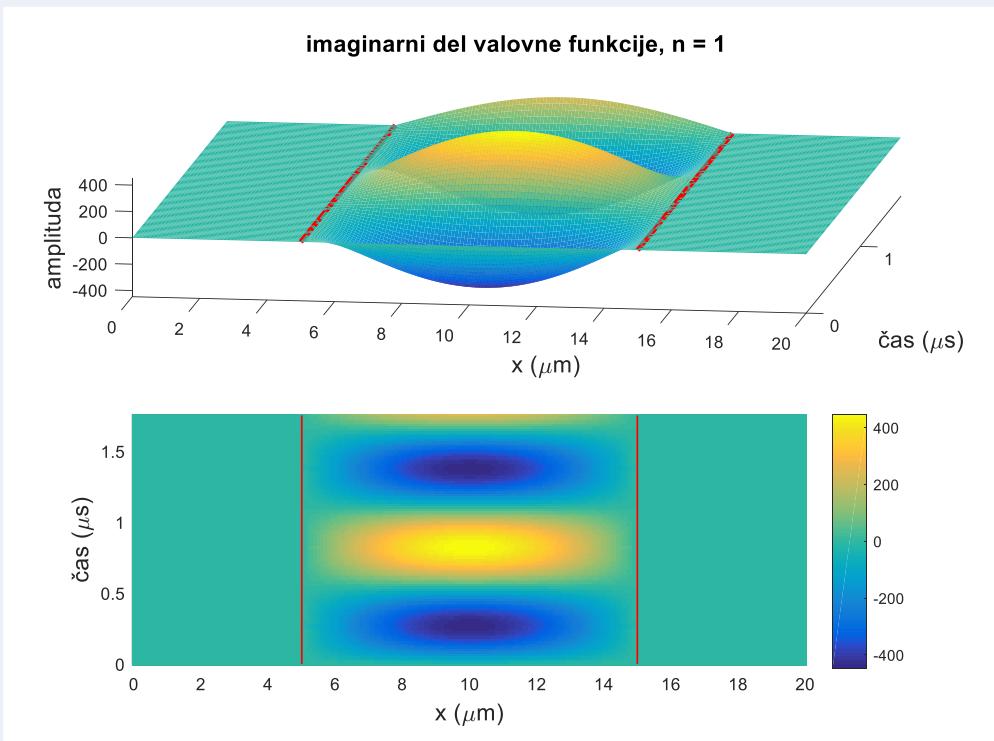
Verjetnost $p(x, t) = |\Psi(x, t)|^2$, da najdemo delec na poziciji x , lahko izrazimo kot

$$\Psi(x, t) == \begin{cases} \frac{D}{2} \sin^2(k_n(x - x_c + D/2)), & x_c - D/2 < x < x_c + D/2 \\ 0, & \text{drugače} \end{cases}. \quad (3.17)$$

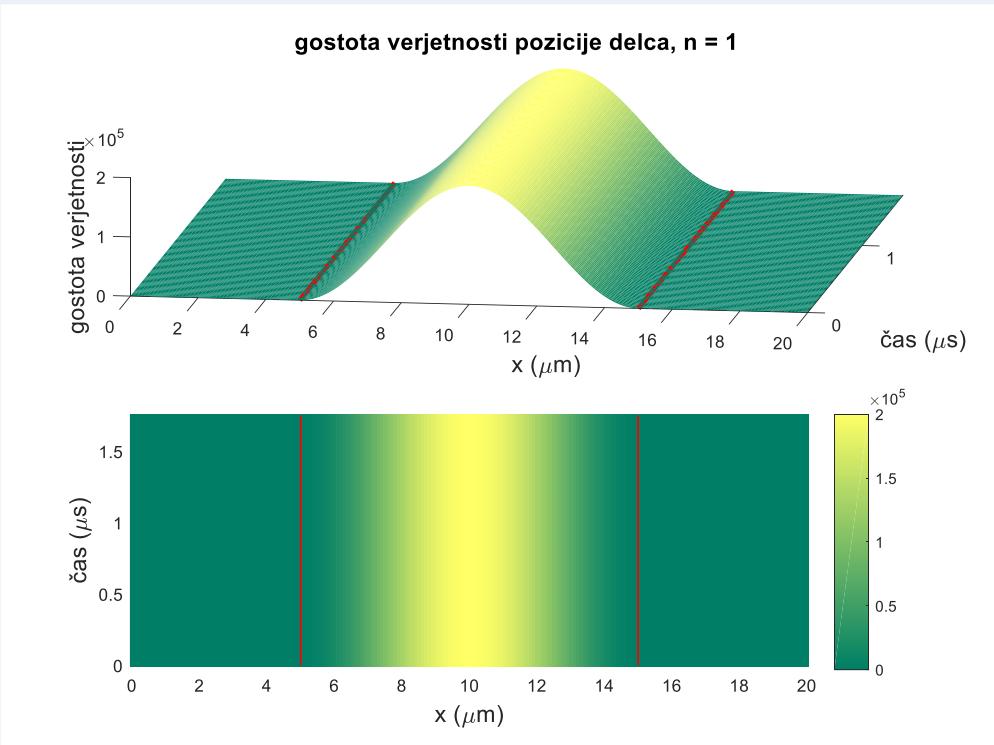
Spremembe valovne funkcije $\Psi(x, t)$ skozi prostor x in čas t za različne energijske nivoje prikazujejo slike 3.5, 3.6, 3.8 in 3.10. Slike 3.7, 3.9 in 3.11 prikazujejo porazdelitev gostote verjetnosti za pozicijo delca. Vidimo, da niso vse pozicije na intervalu enako verjetne, saj smo v Schrödingerjevi enačbi zahtevali, da je valovna funkcija na robih intervala enaka 0.



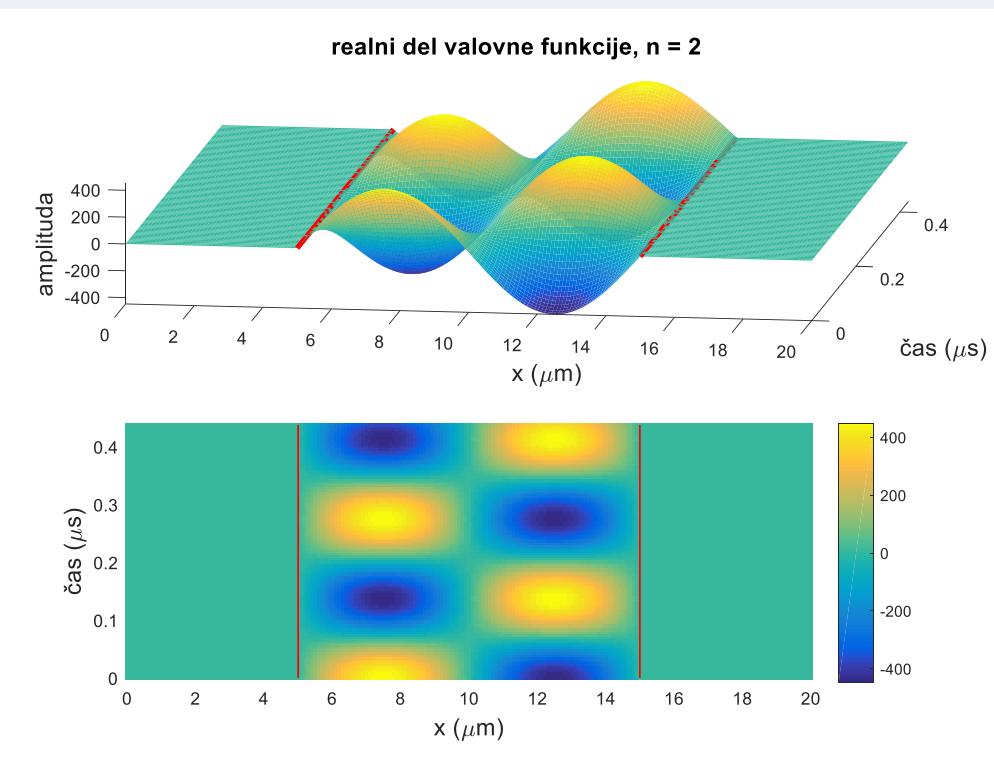
Slika 3.5: Realni del valovne funkcije $\Psi(x, t)$ kvantnega delca ($n=1$), ki se prosto giblje na omejenem intervalu v 1D-prostoru (meje intervala so narisane z rdečimi črtami).



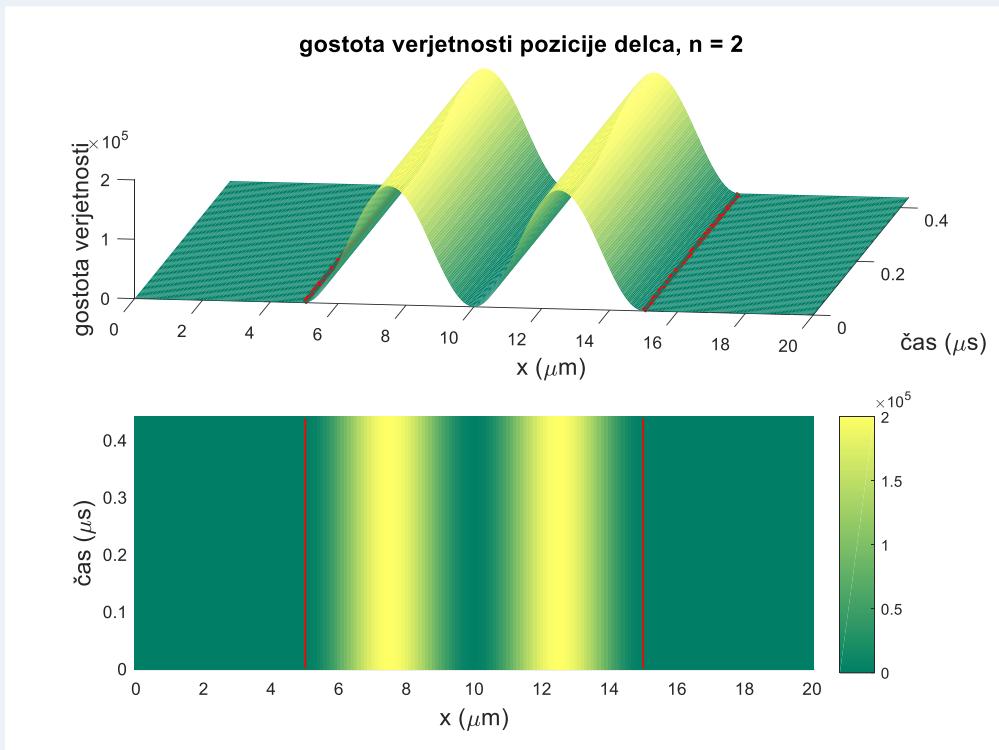
Slika 3.6: Imaginarni del valovne funkcije $\Psi(x, t)$ kvantnega delca ($n=1$), ki se prosto giblje na omejenem intervalu v 1D-prostoru (meje intervala so narisane z rdečimi črtami).



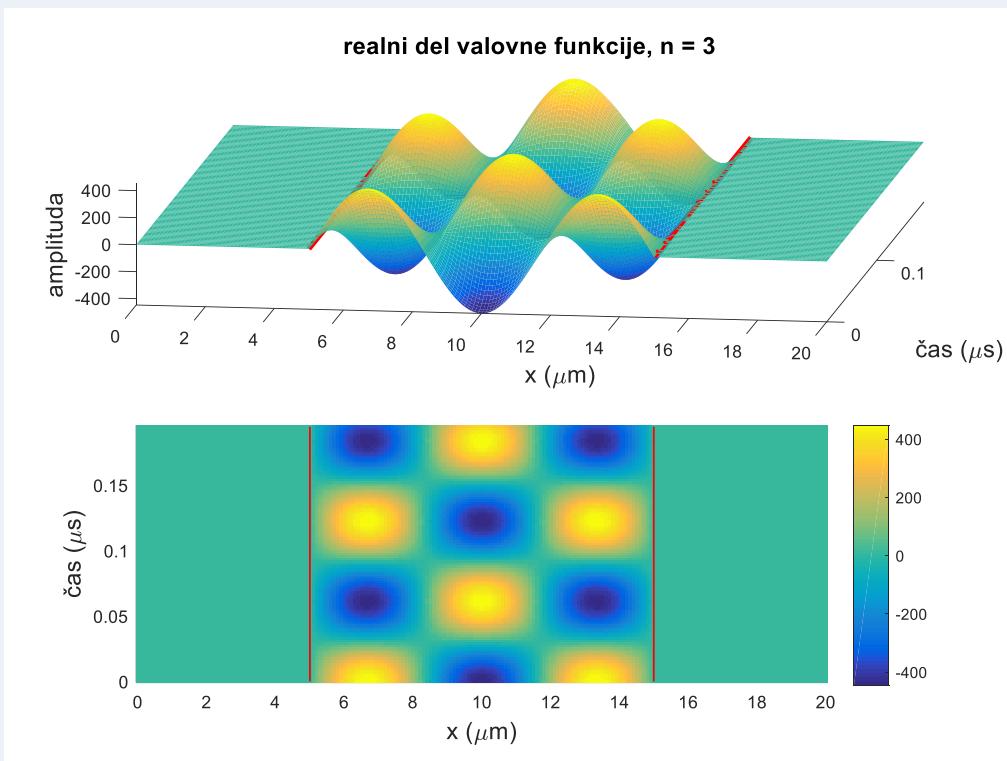
Slika 3.7: Gostota verjetnosti pozicije delca v različnih časovnih trenutkih ($n=1$). Valovna funkcija delca je prikazana na slikah 3.5 in 3.6.



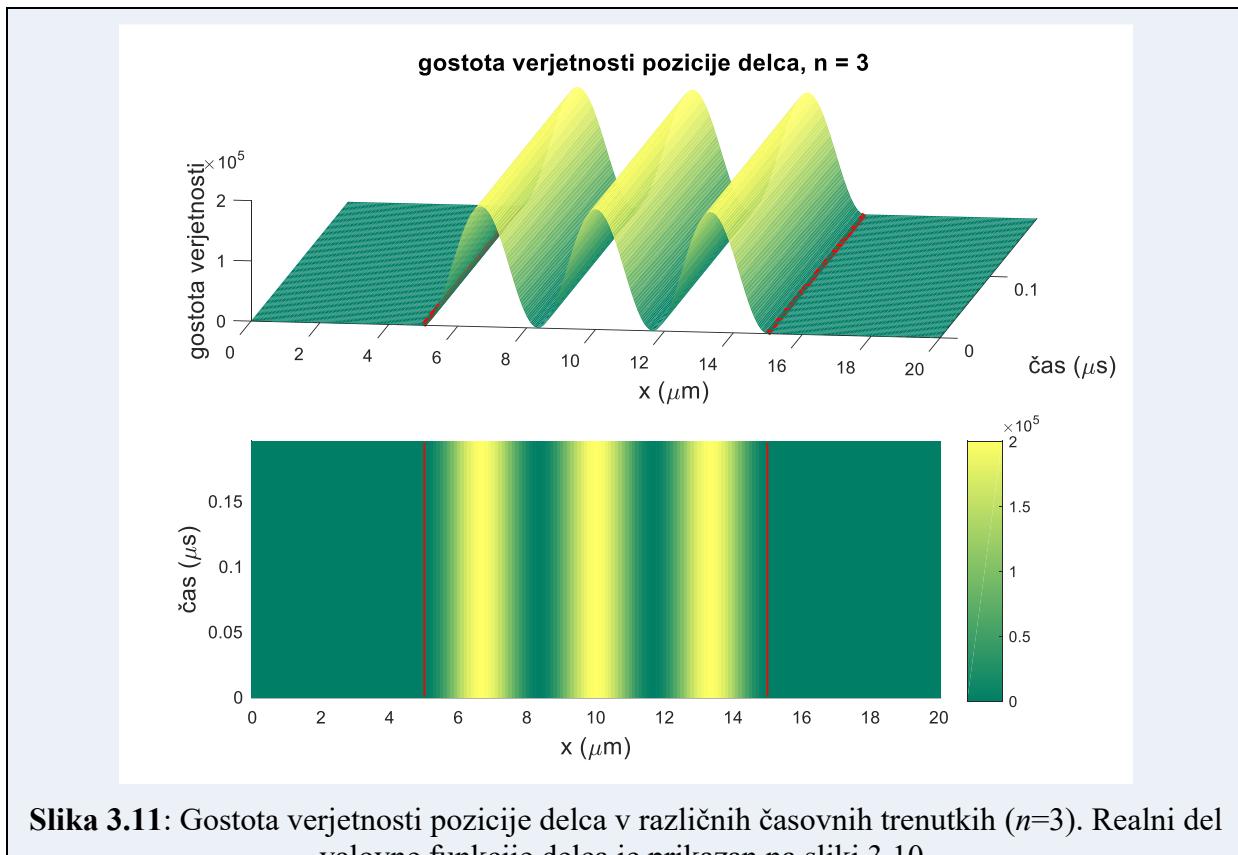
Slika 3.8: Realni del valovne funkcije $\Psi(x, t)$ kvantnega delca ($n=2$), ki se prosti giblje na omejenem intervalu v 1D-prostoru (meje intervala so narisane z rdečimi črtami).



Slika 3.9: Gostota verjetnosti pozicije delca v različnih časovnih trenutkih ($n=2$). Realni del valovne funkcije delca je prikazan na sliki 3.8.



Slika 3.10: Realni del valovne funkcije $\Psi(x, t)$ kvantnega delca ($n=3$), ki se prosto giblje na omejenem intervalu v 1D-prostoru (meje intervala so narisane z rdečimi črtami).



Slika 3.11: Gostota verjetnosti pozicije delca v različnih časovnih trenutkih ($n=3$). Realni del valovne funkcije delca je prikazan na sliki 3.10.

3.3 Interpretacije kvantne mehanike

Vrnimo se k eksperimentu z dvojno režo. Meritev fotona, ki leti skozi režo povzroči kolaps valovne funkcije v eno samo vrednost in foton se prične obnašati kot klasičen delec. Ni pomembno, ali se z meritvijo dejansko seznanimo ali ne. Dovolj je, da informacijo o meritvi shranimo. Če bo nekje obstajal dokaz, skozi katero režo je šel posamezen foton, se bosta na merilnem zaslonu pojavili samo dve osvetljeni progi (slika 3.2), če pa ta dokaz ne bo obstajal in ne bo na noben način močno izvedeti, skozi katero režo je letel foton, se bo na merilnem zaslonu pojavil interferenčni vzorec (slika 3.3).

Kaj torej je meritev in kako vpliva na našo interpretacijo sveta? Kako lahko meritev retrospektivno vpliva na izid eksperimenta? In zakaj nikoli ne vidimo osnovnih delcev v superpoziciji vseh možnih stanj? Odgovor na ta vprašanja še vedno bega znanstvenike in predlaganih je bilo več možnih interpretacij kvantne mehanike. V nadaljevanju se bomo omejili le na tri bolj razširjene interpretacije. Njihova podrobnejša obravnava presega namen tega učbenika. Omenimo, da se vse omenjene interpretacije strinjajo glede Schrödingerjeve enačbe in valovnih funkcij, torej glede sočasnosti vseh možnih dogodkov in kvante prepletosti (kvantno prepletost bomo podrobneje razložili v poglavju 3.5).

3.3.1 Kopenhagenska interpretacija

Kopenhagensko interpretacijo sta med leti 1925 in 1927 vpeljala Niels Bohr in Werner Heisenberg [Fay2014] in ta razлага ostaja do današnjih dni ena izmed najbolj razširjenih interpretacij. Interpretacija pravi, da imamo kvantni sistem in imamo meritno napravo in obstaja meja med njima. Kvantni sistem pred meritvijo nima dorečenih lastnosti, le verjetnosti, da te lastnosti nastopijo. Kvantna mehanika torej le predvidi rezultat meritve, ob meritvah pa izmerimo enega izmed možnih rezultatov, skladno z verjetnostmi, ki jih napove kvantna mehanika, natančneje Schrödingerjeva enačba. Opazovalci smo vedno na klasični strani, torej vedno vidimo kvantni sistem po meritvi.

Čeprav je široko sprejeta, ima ta interpretacija številne kritike. Kopenhagenska interpretacija daje meritvam poseben status. Meni, da so meritve klasične (niso kvantne), in jih s tem ločuje od kvantnega sistema. Ob tem se pojavi problem zaprtosti vesolja: kdo meri celotno vesolje? Kaj v vesolju je kvantni sistem in kaj meritve? Kje je in kdo je merilec? Zagovorniki te interpretacije menijo, da merilec nima posebnega statusa - lahko je naprava, lahko je človek, to ni pomembno. Pomembno je le zabeleženje informacije o realizaciji naključnega dogodka. Kakor hitro zabeležimo dogodek, se njegovo stanje spremeni iz mogočega v gotovo. Einstein je tej interpretaciji (in tudi drugim interpretacijam kvantne mehanike) odločno nasprotoval in znana je njegova trditev, da Bog ne kocka. Do konca svojih dni ni sprejel trditev kvante mehanike, čeprav je sam izdatno prispeval k njenemu razvoju, ko je leta 1905 pojasnil kvantno naravo svetlobe.

Težava kopenhagenske interpretacije je tudi, da predvideva, da se kolaps verjetnostne valovne funkcije zgodi v trenutku, ne glede na to, kako velik kvantni sistem opisuje. Predstavljam si dva fotona, ki sta prepletena (angl. *entangled* - pojem kvantne prepletosti bomo podrobnejše razložili v poglavju 3.5) in si torej delita eno samo valovno funkcijo. Pošljimo oba fotona v nasprotni strani in počakajmo eno leto. Takrat sta fotona dve svetlobni leti narazen, a ko izmerimo enega, povzročimo takojšen kolaps valovne funkcije, ne glede na to, da sta zelo daleč narazen. Kolaps valovne funkcije torej ne upošteva omejitev splošne teorije relativnosti, ki pravi, da ne more nič potovati hitreje od svetlobe. Omenjeni zaplet kvantne mehanike s prostor-časom ali svobodno voljo smo pojasnili že pri opisu eksperimenta z dvojno režo.

3.3.2 Interpretacija več svetov

Interpretacijo več svetov (angl. *many-worlds interpretation*) je leta 1957 predlagal Hugh Everett [Eve1956, Eve1957]. Zasnovana je na ideji, da meritve ne izbere samo ene možne realizacije dogodka v kvantnem sistemu, temveč vse. Vendar se vsaka možna realizacija zgodi v enem izmed paralelnih vesolj, ki med sabo ne komunicirajo. Z drugimi besedami, ob vsaki meritvi se vesolje razcepi v več vesolj in v vsakem izmed njih da meritve kvantnega sistema drugačen rezultat. Vesolje torej nima enoumne preteklosti in ne enoumne prihodnosti.

Vedno so prisotne vse preteklosti in bodo vse prihodnosti. S tem se interpretacija izogne potrebi po verjetnostnem opisu kvantnega sveta. Nadomesti ga s skupkom determinističnih enačb in se s tem približa klasični fiziki (vključno s teorijo splošne relativnosti). Kot smo opisali v prejšnjem podpoglavlju, je kolaps verjetnostne valovne funkcije nedeterminističen in nelokalen, obe lastnosti pa močno nasprotujeta našemu klasičnemu dojemanju sveta. Interpretacija več svetov torej odpravlja poseben status meritve oziroma meritca in je v tem pogledu veliko preprostejša od kopenhagenske interpretacije. Glede na to interpretacijo postaneta tako merjen sistem kot merilec ob meritvi prepletena in ju od trenutka meritve ni moč več ločiti.

Kritiki interpretaciji več svetov očitajo nejasno definiran akt in trenutek razcepa vesolja. Zagovorniki odgovarjajo, da je razcep definiran z meritvijo. Druge kritike vključujejo asimetričnost teorije na potek časa (vesolje se razcepi, nikoli pa se dve vesolji ne združita) in kršenje teorema o ohranitvi energije (v vsakem trenutku se ustvari nešteto mnogo kopij materije v vesolju). Seveda so zagovorniki te interpretacije vse kritike bolj ali manj uspešno teoretično ovrgli, tako da te interpretacije ne moremo preprosto zavrniti.

Zasnovali so tudi eksperimente, ki bi nedvoumno pokazali, ali velja kopenhagenska interpretacija ali interpretacija več svetov. Žal pa ti eksperimenti vključujejo ali makroskopska telesa v superpoziciji vseh stanj ali pa fizično realizacijo velikega kvantnega računalnika. Oboje je trenutno še izven našega tehnološkega dosega.

3.3.3 Interpretacija kvante informacije

Interpretacija kvantne informacije (angl. *quantum information interpretation*) predvideva, da kvantna mehanika ne opisuje objektivnega stanja kvantnega sistema, temveč le meritčeve znanje o kvantnem sistemu [Bay2001, Sch2013]. Kolaps verjetnostne valovne funkcije torej odraža pridobljeno informacijo o sistemu in ne predstavlja spremembe kvantnega sistema samega. Kolaps je konstrukt meritčeve zavesti in njegovega predhodnega znanja o kvantnem sistemu. Merilna naprava in človeška zavest sta kvantna in nista klasična. Sta stohastična in povzročita ireverzibilno pridobitev informacije o stanju vesolja [Inf2016]. Svobodna volja je torej izražena z našim delovanjem, s katerim ireverzibilno pridobivamo informacijo o stanju sveta.

Ta interpretacija domneva, da obstaja en sam kvantni svet in da se prehodi med kvantnim in klasičnim obnašanjem zgodijo implicitno za vsak dovolj velik objekt, ki vsebuje veliko število atomov [Inf2016]. V velikih sistemih se namreč nedoločenosti kvantnega sveta s povprečjem izničijo in obnašanje objekta postane deterministično. S tem se odpravi poseben status meritve ali meritca, hkrati pa se ohranijo vsi postulati kvantnega sveta, od superpozicije vseh možnih stanj, Schrödingerjeve enačbe in valovnih funkcij do kvante prepletosti. Slednja je ključna za interpretacijo meritve. Z meritvijo postaneta merjeni sistem in merilec kvantno prepletena in ju ni več mogoče ločiti na dva neodvisna sistema. Merilec bo torej, takoj ko bo pridobil informacijo o izbrani lastnosti kvantnega sistema, videl eno samo klasično vrednost te

lastnosti in ne superpozicije več možnih vrednosti. Meritev namreč ireverzibilno poveže meritca z merjenim sistemom in iz možnih dogodkov ustvari dejanskost. Niti ni potrebno, da je merilec poseben objekt. Vesolje lahko meri samo sebe.

Interpretacija kvantne informacije je najmlajša in najmanj razširjena interpretacija izmed vseh, ki smo jih omenili v tem poglavju. Vendar v zadnjem času pridobiva priljubljenost med kvantnimi fiziki.

3.4 Kvantna mehanika, linearost in amplitude verjetnosti

Razmislimo abstraktno o dogodku z N možnimi izidi. Verjetnosti vseh izidov lahko zapišemo z vektorjem N realnih števil [Aar2013]:

$$\mathbf{p} = [p_1, \dots, p_N]$$

Kaj lahko povemo o tem vektorju?

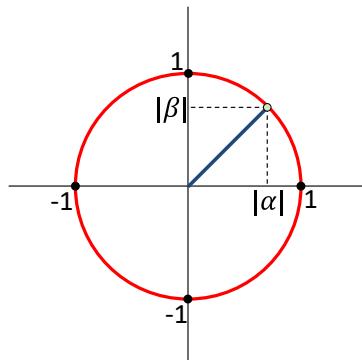
1. Vse verjetnosti so nenegativne: $\forall i \ p_i \geq 0$.
2. Njihov seštevek je enak 1.

Zadnje dejstvo lahko izrazimo s pomočjo prve norme (prva norma ali krajše 1-norma je seštevek absolutnih vrednosti): Prva norma vektorja verjetnosti \mathbf{p} mora biti enaka 1.

Ampak 1-norma ni edina znana norma - to ni edini način opredelitve dolžine vektorja [Aar2013]. Obstajajo tudi drugi načini in eden od najpogosteje uporabljenih, vsaj od Pitagorovih dni, je druga norma (**2-norma**) ali **evklidska norma**.

Obravnavajmo en sam bit. V verjetnostnem računu lahko bit opišemo z izidoma 0 in 1: bit zavzame vrednost 0 z verjetnostjo p in vrednost 1 z verjetnostjo $1-p$.

Ko pa namesto **1-norme** uporabimo **2-normo**, ne seštevamo števil, ampak kvadratne njihovih absolutnih vrednosti (Pitagorov izrek), njihov seštevek pa mora biti enak 1. Z drugimi besedami, imamo vektor kompleksnih števil $[\alpha, \beta]$, kjer je $|\alpha|^2 + |\beta|^2 = 1$. Množica vseh takšnih vektorjev tvori krog v ravnini (slika 3.12).



Slika 3.12: Shematski prikaz odvisnosti vrednosti $|\alpha|$ od vrednosti $|\beta|$ v vektorju kompleksnih števil $[\alpha, \beta]$, ki ima drugo normo enako 1 ($|\alpha|^2 + |\beta|^2 = 1$).

Toda zakaj v tem primeru ne pozabimo na kompleksni števili α in β in samo opišemo stanje bita neposredno z verjetnostmi njegovih stanj $|\alpha|^2$ in $|\beta|^2$? Razlika nastopi pri transformaciji vektorja $[\alpha, \beta]$ oziroma se izrazi s tem, kako se vektor spremeni, ko ga vstavimo v računsko operacijo [Aar2013].

Zakaj 2-norma in zakaj ne višja norma? Vzemimo teorijo, ki temelji na p -normi, kjer je $p \in \{1, 2, \dots\}$. Vektor $\mathbf{v} = [v_1, \dots, v_N]$ je *enotski vektor p-norme*, če velja $|v_1|^p + \dots + |v_N|^p = 1$. Poiščimo linearno transformacijo, ki preslika katerikoli enotski vektor p -norme v drug enotski vektor p -norme. Za katerikoli izbrani p lahko najdemo linearne transformacije, ki ohranijo p -normo [Aar2013]:

- lahko na primer permutiramo elemente vektorja,
- pred elemente vektorja lahko vstavimo negativne predznaake.

Toda če poleg teh trivialnih transformacij obstaja še katerakoli druga linearna transformacija, ki ohranja p -normo, potem je $p=1$ ali $p=2$. Ko je $p=1$, dobimo klasični verjetnostni račun, ko je $p=2$, dobimo kvantno verjetnost (amplitudo verjetnosti) [Aar2013].

Amplitude verjetnosti kvantne mehanike so kompleksna števila. To pomeni, da moramo kvadrirati absolutne vrednosti amplitud, da dobimo verjetnost. Z drugimi besedami, če je amplituda verjetnosti za neki izid $\alpha = \beta + \gamma i$, kjer sta β in γ realni števili, i pa je enota imaginarni osi v kompleksni ravnini ($i = \sqrt{-1}$), potem je verjetnost tega izida enaka $|\alpha|^2 = \beta^2 + \gamma^2$.

Zakaj je narava izbrala kompleksna števila in ne realna? Kompleksna števila so algebrajsko zaprta. Z drugimi besedami, za katerokoli linearno transformacijo U obstaja linearna transformacija V , tako da velja $V^2 = U$. Ta relacija v bistvu definira *zveznost prostor-časa*: če je smiselno, da operacijo izvedemo za časovni interval ene sekunde, mora biti smiselno tudi, da jo izvedemo za interval pol sekunde [Aar2013]. Torej, ali je prostor-čas zvezen in narava uporablja kompleksna števila ali pa je prostor-čas diskreten. Omenjena dilema je ena izmed najbolj perečih vprašanj teoretične fizike in trenutno poteka intenzivna razprava med atomisti in anti-atomisti: vprašanje v tej razpravi je, ali sta prostor in čas sestavljeni iz nedeljivih prostorskih in časovnih korakov, s Planckovo ločljivostjo 10^{-35} metra oziroma 10^{-43} sekunde [Aar2013]. Omenjeni ločljivosti sta daleč pod našo trenutno zmožnostjo merjenja razdalj in časa (najmanjša eksperimentalno izmerjena razdalja je trenutno $10^{-18} m$), zato bo omenjena razprava še nekaj časa omejena na področje fizikalnih teorij.

Zakaj linearne transformacije? Abrams in Lloyd [Abr1998] sta leta 1998 dokazala, da bi lahko, če bi bila kvantna mehanika nelinearna, v polinomskem času izračunali NP-polne probleme. Torej, ali je naš svet linearen ali pa lahko vnaprej napovemo/izračunamo katerikoli dogodek na svetu, od gibanja delnic do življenjske dobe človeka in njegove svobodne volje [Aar2013]. Ker je svobodna volja neločljivo prepletena z našim dojemanjem sveta in

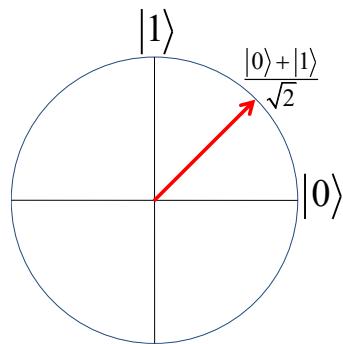
predstavlja temelj našega delovanja, je večina znanstvenikov sprejela razlago o linearnosti kvantne mehanike. Slednja je bila posredno potrjena tudi z močjo napovedovanja razvitih linearnih kvantnih teorij in trenutno ni nobenega dokaza, da bi bila kvantna mehanika nelinearna. Hkrati pa je treba opozoriti, da kvantna mehanika še vedno ni popolnoma pojasnila ustroja sveta in je slabo združljiva z Einsteinovo teorijo splošne relativnosti, ki zelo natančno razlaga vesolje na velikih razdaljah. Kvantna mehanika torej ni končna razlaga sveta, čeprav veliko znanstvenikov meni, da bo zelo verjetno del končnega odgovora, ki ga večkrat omenjajo pod imenom teorija velikega poenotenja (angl. *great unified theory – GUT*).

3.5 Kvantni bit

V nadaljevanju bomo z enotskim vektorjem 2-norme opisali **kvantni bit**. Fiziki ga navadno predstavijo z **Diracovo notacijo bra-ket** [Zwi2013, Aar2013], v kateri vektor $[\alpha, \beta]$ postane $\alpha|0\rangle + \beta|1\rangle$:

- α je amplituda verjetnosti stanja $|0\rangle$,
- β je amplituda verjetnosti stanja $|1\rangle$.

Če je kvantni bit lahko v dveh stanjih $|0\rangle$ ali $|1\rangle$, potem je lahko tudi v superpoziciji teh stanj $\alpha|0\rangle + \beta|1\rangle$ (slika 3.13). Ko izmerimo tak kvanti bit, bomo videli stanje $|0\rangle$ z verjetnostjo $|\alpha|^2$ in stanje $|1\rangle$ z verjetnostjo $|\beta|^2$. Tako ko kvanti bit izmerimo oziroma pogledamo, povzročimo takojšnji kola superpozicije v katerokoli izmed obeh osnovnih stanj: $|0\rangle$ ali $|1\rangle$, kar je v skladu z vsemi v poglavju 3.3 omenjenimi interpretacijami kvante mehanike.



Slika 3.13: Grafična ponazoritev kvantnega bita, ki je v superpoziciji stanj $1/\sqrt{2}(|0\rangle + |1\rangle)$. Opozorimo, da sta amplitudi verjetnosti stanj $|0\rangle$ in $|1\rangle$ kompleksni števili in da je prikazana ponazoritev kvantnega bita poenostavljena. Za eksaktno grafično ponazoritev kvantnega bita bi potrebovali štiridimenzionalen prostor oziroma dve kompleksni ravnini.

3.5.1 Ločljiva in prepletena stanja kvantnih bitov

Če poznamo stanji dveh kvantih bitov, potem lahko njuno **kombinirano stanje** zapišemo s **tenzorskim produkтом**. Če je prvi kvanti bit v stanju $\alpha|0\rangle + \beta|1\rangle$ in drugi kvantni bit v stanju $\gamma|0\rangle + \delta|1\rangle$, potem njuno kombinirano stanje zapišemo kot

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle. \quad (3.18)$$

Ločljiva stanja so stanja dveh ali več kvantih bitov, ki jih lahko zapišemo kot tenzorski produkt posameznih kvantih bitov:

$$\alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle). \quad (3.19)$$

Prepletena stanja so stanja dveh ali več kvantih bitov, ki jih ne moremo zapisati kot tenzorski produkt posameznih kvantnih bitov. Najbolj slavno prepletene stanje para kvantih bitov je stanje EPR (Einstein-Podolsky-Rosen):

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}. \quad (3.20)$$

Zgled 3.3: Poskusimo faktorizirati stanje bitov EPR. Zapišimo prvi bit kot $(\alpha|0\rangle + \beta|1\rangle)$, drugega pa kot $(\gamma|0\rangle + \delta|1\rangle)$. Potem mora glede na enačbo (3.19) veljati:

$$\alpha\gamma = \frac{1}{\sqrt{2}}, \quad (3.21)$$

$$\alpha\delta = 0, \quad (3.22)$$

$$\beta\gamma = 0, \quad (3.23)$$

$$\beta\delta = \frac{1}{\sqrt{2}}. \quad (3.24)$$

Glede na enačbi (3.21) in (3.24) mora veljati $\alpha \neq 0$, $\beta \neq 0$, $\gamma \neq 0$, $\delta \neq 0$, kar pa nasprotuje enačbam (3.22) in (3.23).

Formalna definicija kvantne prepletosti: Če lahko kombinirano stanje ρ dveh podsistemov A in B, ki sta v stanjih $|\psi_A\rangle$ in $|\psi_B\rangle$, zapišemo z verjetnostno porazdelitvijo tenzorskega produkta stanj $\rho = \alpha|\psi_A\rangle \otimes \beta|\psi_B\rangle$, potem je ρ **ločljivo** stanje. V nasprotnem

primeru je ρ **prepleteno** stanje [Aar2013]. V zgornjem primeru sta α in β amplitudi verjetnosti pri stanjih $|\psi_A\rangle$ in $|\psi_B\rangle$.

3.6 Teorem neizvedljivosti kloniranja

Teorem neizvedljivosti kloniranja (angl. *no cloning theorem*) prepoveduje kloniranje kvantnega stanja oziroma izdelavo identičnih kopij poljubnega kvantnega stanja. Zagotavlja ga linearost kvantne mehanike [Aar2013].

Teorem je pomemben tako v programiraju kvantnih računalnikov kot v kvantni kriptografiji:

1. Ne moremo izdelati varnostnih (angl. *backup*) kopij kvantnega stanja in jih med kvantnim izračunom uporabiti za odpravo računskih napak oziroma za odkrivanje rešitev sistema enačb (za primer uporabe glejte razlago Shorovega algoritma v poglavju 5).
2. Teorem omogoča varno izmenjavo ključev: prisluškovalec ne more izdelati kopije poslanega kvantnega kriptografskega ključa, ne da bi ga pri tem spremenil.

Omenimo, da je kvantno teleportiranje kljub temu mogoče in je bilo tudi eksperimentalno potrjeno [Ben1993]. Teleportiranje namreč ne vključuje kloniranja oziroma kopiranja kvantnega stanja. Teleportirano kvantno stanje se na izvoru uniči (bolje rečeno, povzroči kolaps superpozicije v klasično stanje sistema), na cilju pa se s pomočjo novih kvantnih delcev vzpostavi novo kvantno stanje, ki je identično tistemu na izvoru. Kvantno teleportiranje navadno vključuje vzporedno komunikacijo po kvantnem in klasičnem komunikacijskem kanalu [Bou1997].

Naloge:

1. Opišite vlogo Schrödingerjeve enačbe v kvantni mehaniki.
2. Zakaj valovna funkcija $\Psi(x, t) = e^{i(kx - \omega t)}$ ni fizikalno sprejemljiva rešitev Schrödingerjeve enačbe?
3. Kako tolmači učinek meritve kvantnega sistema kopenhagenska interpretacija?
4. Kako tolmači učinek meritve kvantnega sistema interpretacija več svetov?
5. Kakšna je razlika med amplitudo verjetnosti in verjetnostjo?

6. Izračunajte verjetnosti, da ob meritvi kvantnega bita izmerimo vrednosti 0 in 1
- a) $\frac{1}{2}(1-i)|0\rangle + \frac{1}{2}(1+i)|1\rangle$ b) $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
c) $\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{2}(1+i)|1\rangle$ d) $\frac{1}{\sqrt{7}}(2-i)|0\rangle + \frac{1}{\sqrt{7}}(1-i)|1\rangle$
e) $\frac{2}{3}|0\rangle + \frac{1}{3}(2+i)|1\rangle$ f) $\frac{1}{3\sqrt{2}}(2-2i)|0\rangle + \frac{1}{3\sqrt{2}}(1+3i)|1\rangle$
7. Ustrezno normirajte amplitude verjetnosti kvantnega bita:
- a) $(10-i)|0\rangle + (5+i)|1\rangle$ b) $i|0\rangle + 5|1\rangle$
c) $(3-3i)|0\rangle + (4-4i)|1\rangle$ d) $(3-\sqrt{2}i)|0\rangle + (2+i)|1\rangle$
e) $(2+i)|0\rangle + (2+3i)|1\rangle$ f) $(3-5i)|0\rangle + (4+3i)|1\rangle$
8. S tenzorskim produktom izračunajte kombinirano stanje dveh kvantnih bitov (stanje kvantnega registra z dvema kvantnima bitoma):
- a) $\left(\frac{1}{2}(1-i)|0\rangle + \frac{1}{2}(1+i)|1\rangle\right) \otimes (i|0\rangle + 5|1\rangle)$
b) $\left(\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{2}(1+i)|1\rangle\right) \otimes \left(\frac{1}{\sqrt{7}}(2-i)|0\rangle + \frac{1}{\sqrt{7}}(1-i)|1\rangle\right)$
c) $\left(\frac{2}{3}|0\rangle + \frac{1}{3}(2+i)|1\rangle\right) \otimes \left(\frac{1}{3\sqrt{2}}(2-2i)|0\rangle + \frac{1}{3\sqrt{2}}(1+3i)|1\rangle\right)$
d) $\left(\frac{1}{2}(1-i)|0\rangle + \frac{1}{2}(1+i)|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{2}(1+i)|1\rangle\right)$
9. Faktorizirajte naslednja stanje kvantnega registra in jih zapiši kot ločeni stanji obeh kvantnih bitov:
- a) $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$
b) $\frac{1}{2}i|00\rangle - \frac{1}{2}i|01\rangle - \frac{1}{2}i|10\rangle + \frac{1}{2}i|11\rangle$
c) $\frac{1}{2}(1-i)|00\rangle + \frac{1}{2}(1+i)|10\rangle$
d) $\frac{1}{2}i|00\rangle + \frac{1}{2}i|01\rangle + \frac{1}{2}i|10\rangle + \frac{1}{2}i|11\rangle$
10. Katera stanja kvantnega registra so ločljiva in katera prepletena?
- a) $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$
b) $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$
c) $\frac{1}{\sqrt{2}}|10\rangle + \frac{1}{\sqrt{2}}|01\rangle$
d) $-\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$

4 Kvantno računalništvo

4.1 Klasični računalnik

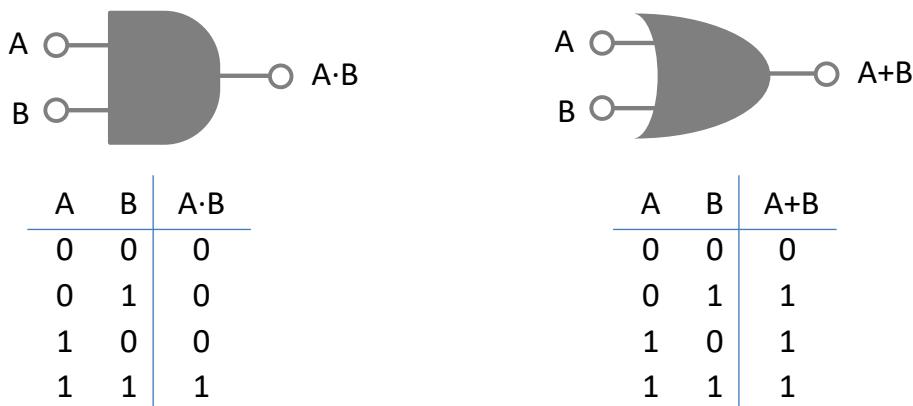
Osnovna enota klasičnih podatkov je bit. Zavzame lahko vrednosti 0 ali 1. Klasični računalnik predstavi podatke kot niz klasičnih bitov.

Zgled 4.1: Črko 'A' lahko na primer zapišemo z zaporedjem klasičnih bitov 01000001.

Število 165 zapišemo z bitnim zaporedjem 10100101:

$$\begin{aligned}10100101_2 &= \\1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 &= \\1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 &= \\165_{10}\end{aligned}$$

Vse računske operacije klasičnega računalnika temeljijo na logičnih vratih. Vzemimo na primer, logična vrata IN (angl. AND) sprejmejo dva vhodna klasična bita in vrnejo stanje 1, če in samo če sta oba vhoda v stanju 1 (slika 4.1). Podobno klasična vrata ALI (angl. OR) sprejmejo dva bita in vrnejo 0, če sta oba vhodna bita v stanju 0, drugače pa vrnejo 1.

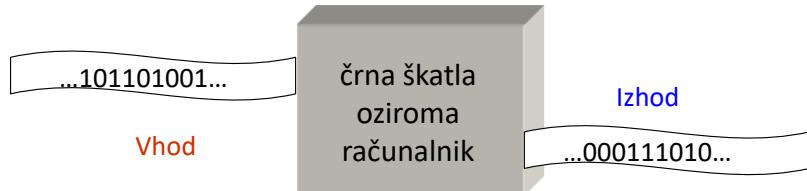


Slika 4.1: Klasična vrata IN (levo) in ALI (desno).

Klasični algoritem je vsako zaporedje klasičnih operacij (logičnih vrat). Klasičen računalnik je naprava, ki lahko izvede klasični algoritem (slika 4.2). Čeprav moderni računalniki temeljijo na kvanti mehaniki, ki razlaga delovanje tranzistorjev, še vedno poganjajo klasične

algoritme. V principu bi lahko izdelali klasični računalnik, ki ne bi temeljil na kvantni mehaniki. Lahko bi na primer izdelali mehanskih računalnik iz zobnikov in zatičev, ki ne bi uporabljaj tranzistorjev.

Poudarimo, da lahko tudi determinističen klasični računalnik poganja verjetnostne (stohastične) algoritme. Miller-Rabinov test praštevilskosti, ki smo ga predstavili v prvem poglavju, teče na sodobnih osebnih računalnikih, ki so deterministični Turingov stroj [Aar2013]. Ne moremo pa na klasičnih računalnikih poganjati kvantnih algoritmov. Slednji izkoriščajo superpozicijo vseh možnih stanj kvantnih bitov, povezanih v kvantne registre.



Slika 4.2: Shematski prikaz klasičnega računalnika.

4.2 Kvanti register

Za razliko od klasičnega bita, ki je zagotovo v enem od dveh klasičnih stanj, je stanje kvantnega bita v splošnem poljubna superpozicija obeh stanj $|0\rangle$ in $|1\rangle$ [Aar2013]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (4.1)$$

kjer sta kompleksni števili α in β amplitude verjetnosti, za kateri velja

$$|\alpha|^2 + |\beta|^2 = 1. \quad (4.2)$$

Amplitudi verjetnosti stanj kvantnega bita sta torej normirani glede na 2-normo.

Kvantni register je zbir n kvantnih bitov. Lahko ga zapišemo s tenzorskim produktom stanj posameznih kvantnih bitov:

$$|\mathbf{a}\rangle = |a_{n-1}\rangle \otimes |a_{n-2}\rangle \dots |a_1\rangle \otimes |a_0\rangle. \quad (4.3)$$

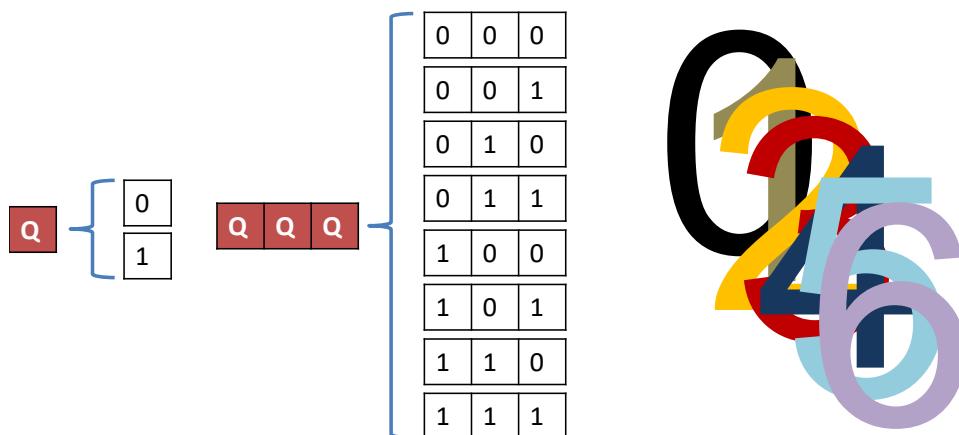
Kvantni register se lahko nahaja v kateremkoli klasičnem stanju, lahko pa je tudi v superpoziciji katerihkoli klasičnih stanj. Torej lahko v kvantnem registru z n kvantnimi biti sočasno hranimo 2^n klasičnih števil. Poglejmo si možna stanja kvantnega registra s tremi kvantnimi biti.

Zgled 4.2: Kvantni register v klasičnem stanju hrani eno samo število. Lahko pa je v superpoziciji več klasičnih stanj. Takrat je v njem sočasno shranjenih več števil:

- Klasično stanje $|1\rangle = |0\rangle \otimes |0\rangle \otimes |1\rangle = |011\rangle$
- Klasično stanje $|3\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle = |111\rangle$
- Klasično stanje $|21\rangle = |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle = |10101\rangle$
- Kvantno stanje $1/\sqrt{2} (|1\rangle + |3\rangle) = 1/\sqrt{2} |0\rangle \otimes (|0\rangle + |1\rangle) \otimes |1\rangle$
- Kvantno stanje $2^{-3/2} (|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle) = 1/\sqrt{2} (|0\rangle + |1\rangle) \otimes 1/\sqrt{2} (|0\rangle + |1\rangle) \otimes 1/\sqrt{2} (|0\rangle + |1\rangle)$

kjer smo števila, ki so zapisana v desetiškem številskem sistemu pisali krepko, števila, ki so zapisana v dvojiškem številskem sistemu pa ne.

Zadnji dve stanji kvantnega registra, ki sta podani v zgledu 4.2, predstavljata superpoziciji klasičnih stanj. Prvo med njima hkrati hrani števili 3 in 7, saj je najpomembnejši kvantni bit registra v superpoziciji stanj $|0\rangle$ in $|1\rangle$. V drugem kvantnem stanju so v superpoziciji vsi trije biti. Torej bo v tem primeru kvantni register sočasno hranil vsa števila od 0 do 7 (slika 4.3).



Slika 4.3: Shematski prikaz kvantnega registra treh kvantnih bitov, ki so v superpoziciji stanj $|0\rangle$ in $|1\rangle$.

Kvantni registri z n kvantnimi biti omogočajo sočasno izvedbo računskih operacij nad 2^n števili. Omogočajo torej najvišjo znano stopnjo paralelnega računanja, ki krepko presega sposobnosti sodobnih procesorjev, vključno z grafičnimi procesorji ali super računalniki. Z osmimi kvantnimi biti lahko sočasno izvedemo $2^8 = 256$ računskih operacij, s šestnajstimi kvantnimi biti 65.536 računskih operacij, z dvaintridesetimi kvantnimi biti pa že

4.294.967.296 računskih operacij. Kot zanimivost povzemimo dejstvo, da lahko 100 kvantnih bitov hrani več klasičnih bitov informacije kot je atomov v trenutno znanem vesolju [Aar2013].

4.3 Kvantna vrata

Kvantna logična vrata opravljajo unitarno transformacijo stanja enega ali več kvantnih bitov v novo stanje kvantnih bitov. Predstavimo jih lahko kot linearne operatorje v Hilbertovem prostoru. **Nelinearne transformacije so PREPOVEDANE** [Aar2013, Bro2014, Eke2008].

Kvantna vrata stanje

$$\sum_{i=0}^{n-1} \alpha_i |i\rangle, \quad (4.4)$$

kjer je

$$\sum_{i=0}^{n-1} |\alpha_i|^2 = 1, \quad (4.5)$$

spremenijo v novo kvantno stanje

$$\sum_{i=0}^{n-1} \beta_i |i\rangle, \quad (4.6)$$

kjer je

$$\sum_{i=0}^{n-1} |\beta_i|^2 = 1. \quad (4.7)$$

Kvanta vrata je najprimernejše predstaviti z matrikami, kjer stanje posameznega kvantnega bita zapišemo v bazi stanj $|0\rangle$ in $|1\rangle$ [Aar2013, Bro2014, Eke2008].

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (4.8)$$

Analogno lahko superpozicijo stanj $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ v kvatnem bitu predstavimo z vektorjem

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (4.9)$$

Kvantni bit torej predstavimo kot dvodimensionalni vektor. Kvantna vrata, ki sprejmejo en sam kvantni bit predstavimo z **unitarno** matriko \mathbf{U} velikosti 2×2 [Aar2013, Bro2014, Eke2008]:

$$U = \begin{bmatrix} \gamma_{11} + i\delta_{11} & \gamma_{12} + i\delta_{12} \\ \gamma_{21} + i\delta_{21} & \gamma_{22} + i\delta_{22} \end{bmatrix}, \quad (4.10)$$

kjer so γ_{ij} in δ_{ij} realna števila. Elementi matrike U so torej kompleksna števila. Matrika U ima drugo normo enako 1 in ohranja drugo normo amplitud verjetnosti kvantnega bita (norma amplitud verjetnosti kvantnega bita se po množenju z matriko U ne spremeni). Inverz matrike U izračunamo s pomočjo hermitskega konjugiranja:

$$U \cdot U^H = I, \quad (4.11)$$

kjer je

$$U^H = \begin{bmatrix} \gamma_{11} - i\delta_{11} & \gamma_{21} - i\delta_{21} \\ \gamma_{12} - i\delta_{12} & \gamma_{22} - i\delta_{22} \end{bmatrix}. \quad (4.12)$$

Analogno lahko kvantna vrata, ki povezujejo dva kvantna bita, zapišemo z unitarno matriko velikosti 4×4 , vrata, ki povezujejo tri kvantne bite, pa z unitarno matriko velikosti 8×8 . Zgled za vse omenjene matrike podajamo v naslednjih podpoglavljih.

Zgled 4.3: Poseben primer unitarne matrike je identiteta:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Identiteta ohranja stanje kvantnega bita $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \cdot \alpha + 0 \cdot \beta \\ 0 \cdot \alpha + 1 \cdot \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Zgled 4.4: Matrika

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$$

je unitarna, saj je njena 2-norma enaka 1 in velja $U \cdot U^H = I$. Stanje kvantnega bita $|\psi\rangle = \frac{1}{2}(1+i)|0\rangle + \frac{1}{2}(1-i)|1\rangle$ spremeni v

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2}(1+i) \\ \frac{1}{2}(1-i) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}}(1+i) \end{bmatrix}.$$

Kvantno stanje bita je torej $|\psi\rangle = |0\rangle + \frac{1}{\sqrt{2}}(1+i)|1\rangle$ in ob njegovi meritvi bomo vedno izmerili vrednost 1. Zgled lepo prikaže izrazno moč amplitud verjetnosti. Čeprav imata kvantna bita $|\psi\rangle = |0\rangle + \frac{1}{\sqrt{2}}(1+i)|1\rangle$ in $|\Omega\rangle = |0\rangle + |1\rangle$ enako verjetnost, da ob njuni meritvi zmerimo vrednost 1, se močno razlikujeta v tem, kako unitarna matrika \mathbf{U} spremeni njuno stanje.

Poglejmo še, kako se je pri množenju z matriko \mathbf{U} spremenila norma kvantnega bita $|\psi\rangle$. Pred množenjem z matriko \mathbf{U} je imel kvanti bit 2-normo enako $\left|\frac{1}{2}(1+i)\right|^2 + \left|\frac{1}{2}(1-i)\right|^2 = \frac{1}{2} + \frac{1}{2} = 1$. Po množenju je bila 2-norma enaka $|0|^2 + \left|\frac{1}{\sqrt{2}}(1-i)\right|^2 = 0 + 1 = 1$. Torej je matrika \mathbf{U} ohranila 2-normo kvantnega bita, kot zahtevata enačbi (4.5) in (4.7).

4.3.1 Kvantna vrata NE

Kvantna vrata NE (angl. NOT) operirajo nad enim samim kvantnim bitom in obrnejo stanje kvantnega bita [Aar2013, Bro2014, Eke2008]. Če je kvantni bit v izvornem stanju $|0\rangle$, je po transformaciji z vrati NE v stanju $|1\rangle$ in obratno. Kvantna vrata NE se torej zgledujejo po klasičnem računanju in vrnejo $|0\rangle$, ko je vhod v stanju $|1\rangle$, in $|1\rangle$, ko je vhod v stanju $|0\rangle$.

Matrična predstavitev kvantnih vrat NE:

$$\mathbf{U} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (4.13)$$

Zgornja matrika je zapisana z bazo, ki je podana z enačbo (4.8).

Negacijo stanja $|0\rangle$ torej zapišemo kot

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (4.14)$$

Vrata NE so sama sebi inverz:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4.15)$$

Zgled 4.4: Transformirajmo z vrati NE kvanti bit, ki je v stanju $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Pred množenjem z vrati NE sta bili amplitudi verjetnosti stanj $|0\rangle$ in $|1\rangle$ enaki, zato se kvantnemu bitu stanje po množenju ni spremenilo.

Zgled 4.5: Transformirajmo z vrati NE kvanti bit, ki je v stanju $|\psi\rangle = \frac{1}{2}(1+i)|0\rangle + \frac{1}{2}(1-i)|1\rangle$:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2}(1+i) \\ \frac{1}{2}(1-i) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(1-i) \\ \frac{1}{2}(1+i) \end{bmatrix}.$$

Vrata NE so zamenjala amplitudi verjetnosti pri stanjih $|0\rangle$ in $|1\rangle$.

4.3.2 Hadamardova kvantna vrata

Hadamardova vrata predstavimo z unitarno matriko \mathbf{H} velikosti 2×2 [Aar2013, Bro2014, Eke2008]:

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (4.16)$$

označujemo pa jih s simbolom [Aar2013, Bro2014, Eke2008]:



Privzemimo, da je kvantni bit v stanju $|\psi\rangle = 1|0\rangle + 0|1\rangle$. Ko ga pomnožimo z zgornjo matriko \mathbf{H} , dobimo stanje $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Hadamardova transformacija torej vpelje **interferenco amplitud verjetnosti**:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}. \quad (4.17)$$

Z diagramom poteka to operacijo predstavimo kot



Če rezultat enačbe (4.17) še enkrat pomnožimo s \mathbf{H} , dobimo stanje $|0\rangle$:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \quad (4.18)$$

Izpišimo račun za prvo vrstico:

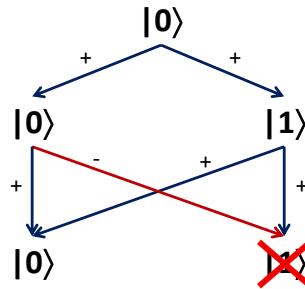
$$\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} = \frac{1}{2} + \frac{1}{2} = 1$$

V tem primeru se amplitudi verjetnosti seštejeta in ojačita.

Drugo vrstico izračunamo kot

$$\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} = \frac{1}{2} - \frac{1}{2} = 0 .$$

V tem primeru se amplitudi verjetnosti medsebojno izničita in verjetnost, da bi po končani operaciji izmerili stanje $|1\rangle$, je enaka 0. Čeprav sta v obravnavanem zgledu dvakratnega množenja stanja $|0\rangle$ z matriko \mathbf{H} dve poti, ki vodita k izidu $|1\rangle$, ima ena pot pozitivno, druga pa negativno amplitudo verjetnosti. Zaradi tega obe poti **destruktivno interferirata** in se medsebojno izničita [Aar2013, Bro2014, Eke2008]. Poti, ki vodita k izidu $|0\rangle$, imata obe pozitivni amplitudi verjetnosti in **interferirata konstruktivno** (slika 4.4). Izničenje pozitivnih in negativnih amplitud verjetnosti na poteh, ki vodijo k določenem izidu, je tista lastnost, ki predstavlja glavno razliko med klasično verjetnostjo in kvantno amplitudo verjetnosti ter, posledično, med klasičnim in kvantnim računanjem [Aar2013, Bro2014, Eke2008].



Slika 4.4: Rezultat dvakratnega množenja kvantnega bita, ki je v klasičnem stanju $|0\rangle$, s Hadamardovimi vrati. Amplitudi verjetnosti stanja $|1\rangle$ sta po absolutni vrednosti enaki, vendar imata nasproten predznak. Zato se v končnem stanju kvantnega bita medsebojno izničita.

Pravimo, da destruktivno interferirata. Nasprotno sta obe amplitudi verjetnosti stanja $|0\rangle$ pozitivni in medsebojno konstruktivno interferirata. Ob meritvi takšnega kvantnega bita bomo klasično stanje $|0\rangle$ opazili z verjetnostjo 1, klasično stanje $|1\rangle$ pa z verjetnostjo 0.

Podobno kot vrata NE so tudi Hadamardova vrata \mathbf{H} sama sebi inverz: $\mathbf{H} = \mathbf{H}^H = \mathbf{H}^{-1}$:

$$\mathbf{H} \cdot \mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4.19)$$

4.3.3 Kvantna fazna vrata

Kvantna fazna vrata spreminja stanje $|1\rangle$ v stanje $e^{i\varphi}|1\rangle$, kjer je e konstanta, ki označuje osnovo naravnega logaritma ($e \approx 2,71828$), in i imaginarna enota ($i = \sqrt{-1}$) [Aar2013, Bro2014, Eke2008]. Stanja $|0\rangle$ ta vrata ne spreminja. Matrično jih zapišemo kot

$$\Phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}. \quad (4.20)$$

Zgled 4.6: Izberimo kot $\varphi = \frac{\pi}{4}$. Potem velja:

$$\varphi = \frac{\pi}{4} \Rightarrow \Phi = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} + i \frac{1}{\sqrt{2}} \end{bmatrix}$$

Če sedaj s to matriko pomnožimo kvantni bit, ki je v stanju $1/\sqrt{2}(|0\rangle + |1\rangle)$, dobimo

$$\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\sqrt{2}} + i \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{2} + i \frac{1}{2} \end{bmatrix}$$

Vidimo, da se je spremenila samo amplituda verjetnosti stanja $|1\rangle$, medtem ko je amplituda verjetnosti stanja $|0\rangle$ ostala nespremenjena.

4.3.4 Nadzorovana NE-vrata

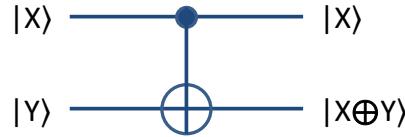
Nadzorovana NE-vrata (angl. *controlled-NOT gate* ali *CNOT gate*) operirajo nad dvema kvantnima bitoma in invertirajo drugi kvantni bit, če in samo če je prvi kvantni bit v stanju $|1\rangle$ [Aar2013, Bro2014, Eke2008]. Predstavimo jih z unitarno matriko velikosti 4×4 :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (4.21)$$

ki je zapisana v bazi vektorjev oziroma stanj $|00\rangle$, $|01\rangle$, $|10\rangle$ in $|11\rangle$:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (4.22)$$

V diagramu nadzorovana vrata NE označimo s simbolom [Aar2013, Bro2014, Eke2008]:



V zgornji sliki smo s simbolom \oplus označili izključujoči ALI, s simbolom zapoljenega kroga pa smo označili konjunkcijo.

Veljajo naslednje osnovne enačbe za spremembe kvantnih stanj pri obeh vhodnih bitih po operaciji s kontroliranimi vrtati NE:

$$\begin{array}{l} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot |00\rangle \\ 0 \cdot |01\rangle \\ 0 \cdot |10\rangle \\ 0 \cdot |11\rangle \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot |00\rangle \\ 1 \cdot |01\rangle \\ 0 \cdot |10\rangle \\ 0 \cdot |11\rangle \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot |00\rangle \\ 0 \cdot |01\rangle \\ 1 \cdot |10\rangle \\ 0 \cdot |11\rangle \end{bmatrix} \end{array} \quad (4.23)$$

Vidimo, da se stanje drugega kvantnega bita negira, če in samo če je prvi kvantni bit v stanju $|1\rangle$.

Zgled 4.7: S kontroliranimi vrtati NE spremenimo stanje kvantnega registra, ki ima prvi bit v stanju $|\psi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, drugi bit pa v stanju $|\psi_2\rangle = |1\rangle|0\rangle + |0\rangle|1\rangle$. S tenzorskim produktom, ki smo ga definirali v enačbi (3.18), zapišimo stanje kvantnega registra:

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right) \otimes (|1\rangle|0\rangle + |0\rangle|1\rangle) = \frac{1}{\sqrt{2}}|00\rangle + 0|01\rangle + \frac{1}{\sqrt{2}}|10\rangle + 0|11\rangle.$$

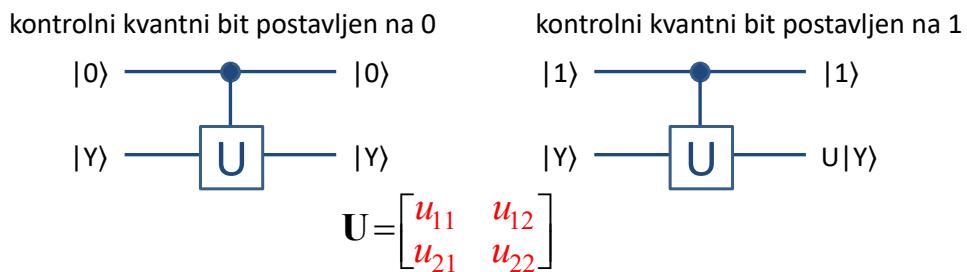
Ko transformiramo kvantni register s kontroliranimi NE-vrati, dobimo

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Vidimo, da sta se spremenili samo amplitudi verjetnosti drugega kvantnega bita.

4.3.5 Nadzorovana U-vrata

Nadzorovana U-vrata (angl. *controlled U gate*) so vrata nad dvema kvantnima bitoma, ki uporabijo unitarno operacijo (matriko) U nad drugim kvantnim bitom, a samo če je prvi, kontrolni kvantni bit v stanju $|1\rangle$ (slika 4) [Aar2013, Bro2014, Eke2008].



Slika 4.5: Primer uporabe kontroliranih U-vrat.

Predstavimo jih z naslednjo matriko velikosti 4×4 :

$$U_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix}$$

baza prostora

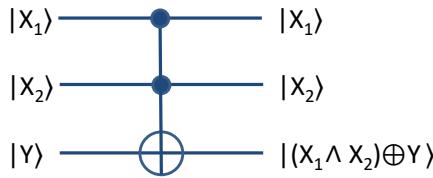
kontrolni bit drugi bit

Primer kontroliranih U-vrat so nadzorovana NE-vrata. Drug primer kontroliranih U-vrat so nadzorovana fazna vrata, ki so definirana kot:

$$R(\varphi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix}. \quad (4.24)$$

4.3.6 Toffolijeva vrata

Dvojna nadzorovana vrata NE (angl. *CCNOT gate*) operirajo nad tremi kvantnimi biti in invertirajo stanje tretjega bita, če in samo če sta prva dva bita v stanju $|1\rangle$ [Aar2013, Bro2014, Eke2008]:



V zgornji sliki smo s simbolom \wedge označili logično operacijo IN HKRATI.

Predstavimo jih z matriko velikosti 8×8 , ki je zapisana v bazi vektorjev oziroma stanj $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$ in $|111\rangle$ [Aar2013, Bro2014, Eke2008]:

$$CC = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (4.25)$$

kjer so posamezna klasična stanja kvantnega registra predstavljena z naslednjimi vektorji:

$$\begin{aligned} |000\rangle &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & |001\rangle &= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & |010\rangle &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & |011\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \\ |100\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & |101\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, & |110\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, & |111\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \end{aligned} \quad (4.26)$$

Zgled 4.7: S Toffolijevi vrati spremenimo stanje kvantnega registra, ki ima prvi bit v stanju $|\psi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, drugi bit v stanju $|\psi_2\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, tretji bit pa v stanju $|\psi_3\rangle = 1|0\rangle + 0|1\rangle$. S tenzorskim produktom, ki smo ga definirali v enačbi (3.18), zapišimo stanje kvantnega registra:

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle) = \\ \frac{1}{2}|000\rangle + 0|001\rangle + \frac{1}{2}|010\rangle + 0|011\rangle + \frac{1}{2}|100\rangle + 0|101\rangle + \frac{1}{2}|110\rangle + 0|111\rangle.$$

Ko transformiramo kvantni register s Toffolijevi vrati, dobimo

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix}.$$

Vidimo, da sta se invertirali samo amplituda verjetnosti tretjega kvantnega bita in kvantni register je po Toffolijevih vratih v stanju

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (0|0\rangle + 1|1\rangle).$$

4.3.7 Univerzalen nabor kvantnih vrat

S kvantnimi Hadamardovimi vrat, nadzorovanimi faznimi vrat $\mathbf{R}(\pi/4)$ in nadzorovanimi NE-vrati lahko ovrednotimo katerokoli Booleovo funkcijo $\{0, 1\}^n \rightarrow \{0, 1\}^m$, ki preslika n vhodnih kvantnih bitov v m izhodnih kvantnih bitov [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000]. Poudarimo, da ni nujno, da je takšno vezje učinkovito (merjeno v številu vrat, ki ga sestavljajo).

4.4 Kvantna Fourierova transformacija amplitud verjetnosti

Imejmo kvantni register z N kvantnimi biti. Vanj lahko hkrati zapišemo $Q = 2^N$ števil. Označimo njihove amplitude verjetnosti z α_x , kjer je x shranjeno število. Stanje kvantnega registra lahko opišemo s splošno superpozicijo vseh števil $\sum_{x=0}^{Q-1} \alpha_x |x\rangle$. V kolikor določeno število ni shranjeno v registru, njegovo amplitudo verjetnosti α_x nastavimo na 0.

Kvantna Fourierova transformacija (KFT) splošno superpozicijo

$$\sum_{x=0}^{Q-1} \alpha_x |x\rangle \quad (4.27)$$

vhodnega registra pretvori v novo superpozicijo stanj

$$\frac{1}{\sqrt{Q}} \sum_{z=0}^{Q-1} \sum_{x=0}^{Q-1} \alpha_x e^{\frac{i2\pi zx}{Q}} |z\rangle, \quad (4.28)$$

kjer je z število, ki je shranjeno v registru po transformaciji [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000]. Torej

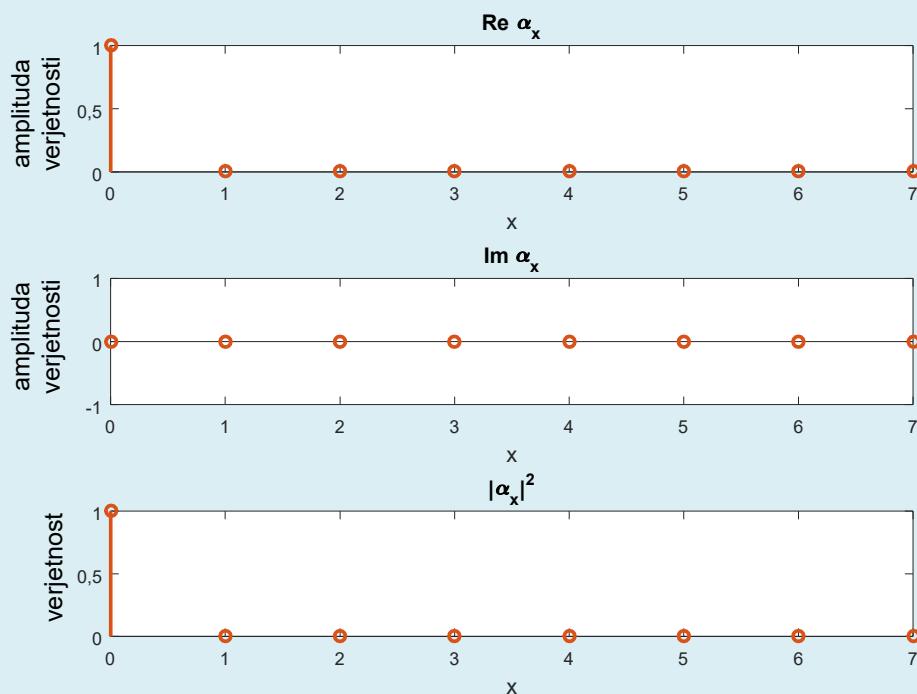
$$\sum_{x=0}^{Q-1} \alpha_x |x\rangle \xrightarrow{KFT} \frac{1}{\sqrt{Q}} \sum_{z=0}^{Q-1} \sum_{x=0}^{Q-1} \alpha_x e^{\frac{i2\pi zx}{Q}} |z\rangle. \quad (4.29)$$

Vsako število z v registru ima sedaj novo amplitudo verjetnosti $\beta_z = \sum_{x=0}^Q \alpha_x e^{\frac{i2\pi zx}{Q}}$.

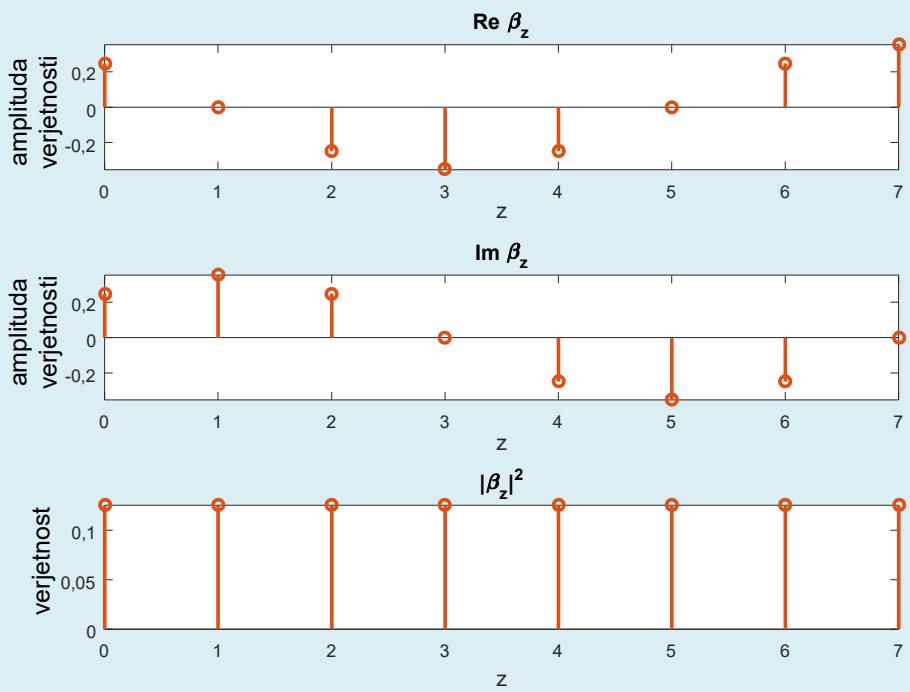
Zgled 4.8: Imejmo kvantni register s tremi kvantnimi biti. Vanj lahko sočasno shranimo vsa števila od 0 do 7. Naj bo register najprej v klasičnem stanju $|000\rangle$, torej je v registru shranjeno le število 0. To stanje kvantnega registra zapišemo z naslednjimi amplitudami verjetnosti (slika 4.6):

$$\begin{aligned} \alpha_0 &= 1, \alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 0, \\ \alpha_4 &= 0, \alpha_5 = 0, \alpha_6 = 0, \alpha_7 = 0. \end{aligned}$$

Verjetnost, da bomo ob meritvi v registru zaznali število 0, je $|\alpha_0|^2 = 1$, verjetnost, da bomo izmerili katerokoli drugo število pa 0 (slika 4.6). Stanje registra po kvantni Fourierovi transformaciji je prikazano na sliki 4.7.



Slika 4.6: Amplitude verjetnosti (zgornja dva grafa) in verjetnost meritve posameznega števila v tribitnem kvantnem registru, ki je v klasičnem stanju $|000\rangle$ (register hrani eno samo število, in sicer število 0).



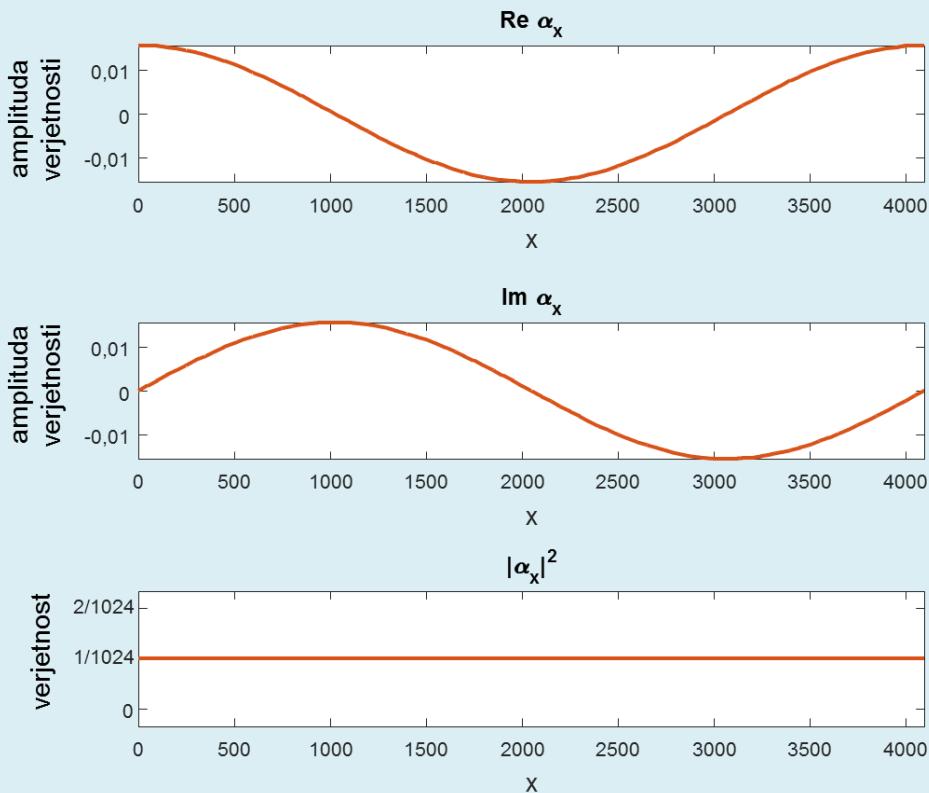
Slika 4.7: Amplitude verjetnosti (zgornja dva grafa) in verjetnost meritve posameznega števila v tribitnem kvantnem registru po kvantni Fourierovi transformaciji. Pred transformacijo je bil kvantni register v klasičnem stanju $|000\rangle$ (slika 4.6).

Po kvantni Fourierovi transformaciji so amplitude verjetnosti posameznih števil:

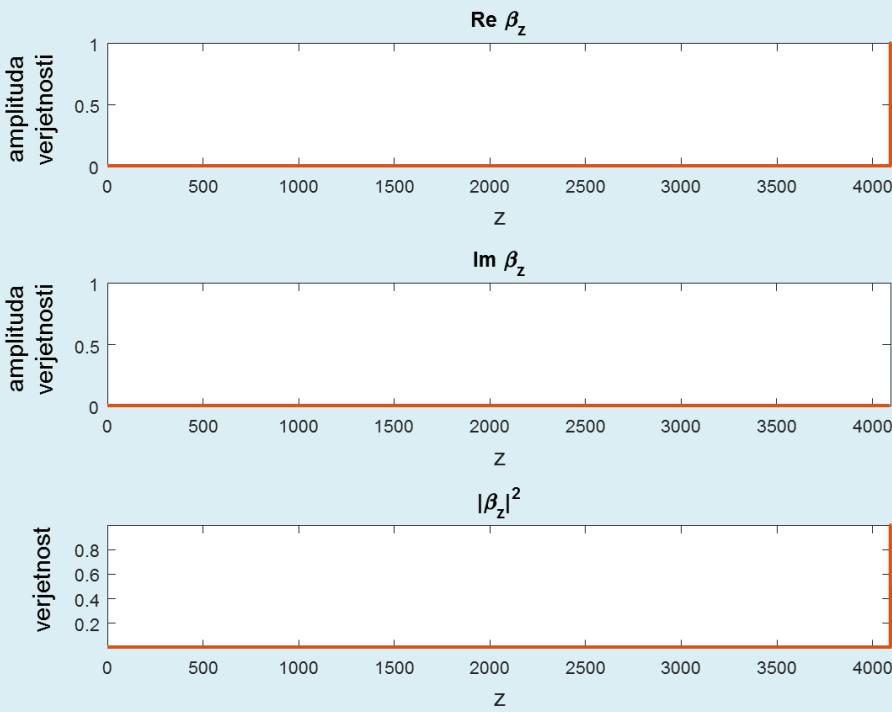
$$\begin{aligned}\beta_0 &= 0,25 + i0,25, \beta_1 = 0,35i, \beta_2 = -0,25 + i0,25, \beta_3 = -0,35, \\ \beta_4 &= -0,25 - i0,25, \beta_5 = 0,35i, \beta_6 = 0,25 - i0,25, \beta_7 = 0,35.\end{aligned}$$

Torej je register v popolni superpoziciji vseh stanj in verjetnost, da bomo izmerili katerokoli število od 0 do 7, je enaka $1/8$ (slika 4.7).

Zgled 4.9: Dan je kvantni register z dvanajstimi kvantnimi biti. Vanj lahko sočasno shranimo 1024 števil, od 0 do 1023. Postavimo register v klasično stanje $|00000000\rangle$, tako da vanj shranimo samo število 0. Sedaj naredimo kvantno Fourierovo transformacijo vsebine registra in ustvarimo superpozicijo vseh možnih stanj. Rezultat transformacije prikazuje slika 4.8. Ko nad rezultatom še enkrat uporabimo kvantno Fourierovo transformacijo, dobimo rezultat, ki je prikazan na sliki 4.9. Po drugi kvantni Fourierovi transformaciji se torej klasično stanje $|00000000\rangle$ (število 0) preslika v klasično stanje $|11111111\rangle$ (število 4095).



Slika 4.8: Amplitude verjetnosti (zgornja dva grafa) in verjetnost meritve posameznega števila v osembitnem kvantnem registru po kvantni Fourierovi transformaciji. Pred transformacijo je bil kvantni register v klasičnem stanju $|00000000\rangle$.



Slika 4.9: Amplituda verjetnosti (zgornja dva grafa) in verjetnost meritve posameznega števila v osebitnem kvantnem registru po dvakratni kvantni Fourierovi transformaciji (po kvantni Fourierovi transformaciji kvantnega registra, ki je prikazan na sliki 4.8).

Zgled 4.9 je demonstriral, da kvantna Fourierova transformacija ni sama sebi inverz. Inverzna kvantna transformacija je v resnici definirana kot [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000]:

$$\sum_{z=0}^{Q-1} \beta_z |z\rangle \xrightarrow{IKFT} \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} \sum_{z=0}^{Q-1} \beta_z e^{\frac{-i2\pi zx}{Q}} |x\rangle. \quad (4.30)$$

Preverimo:

$$\begin{aligned} \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} \sum_{z=0}^{Q-1} \beta_z e^{\frac{-i2\pi zx}{Q}} |x\rangle &= \frac{1}{Q} \sum_{x=0}^{Q-1} \sum_{z=0}^{Q-1} \sum_{y=0}^{Q-1} \alpha_y e^{\frac{i2\pi zy}{Q}} e^{\frac{-i2\pi zx}{Q}} |x\rangle \\ &= \frac{1}{Q} \sum_{y=0}^{Q-1} \alpha_y \sum_{x=0}^{Q-1} \sum_{z=0}^{Q-1} e^{\frac{i2\pi z(y-x)}{Q}} |x\rangle \end{aligned} \quad (4.31)$$

Sedaj upoštevamo, da velja, ko je $y \neq x$

$$\sum_{z=0}^{Q-1} e^{\frac{i2\pi z(y-x)}{Q}} = 0. \quad (4.32)$$

Ko je $y = x$, velja

$$e^{\frac{i2\pi z(y-x)}{Q}} = 1 \quad (4.33)$$

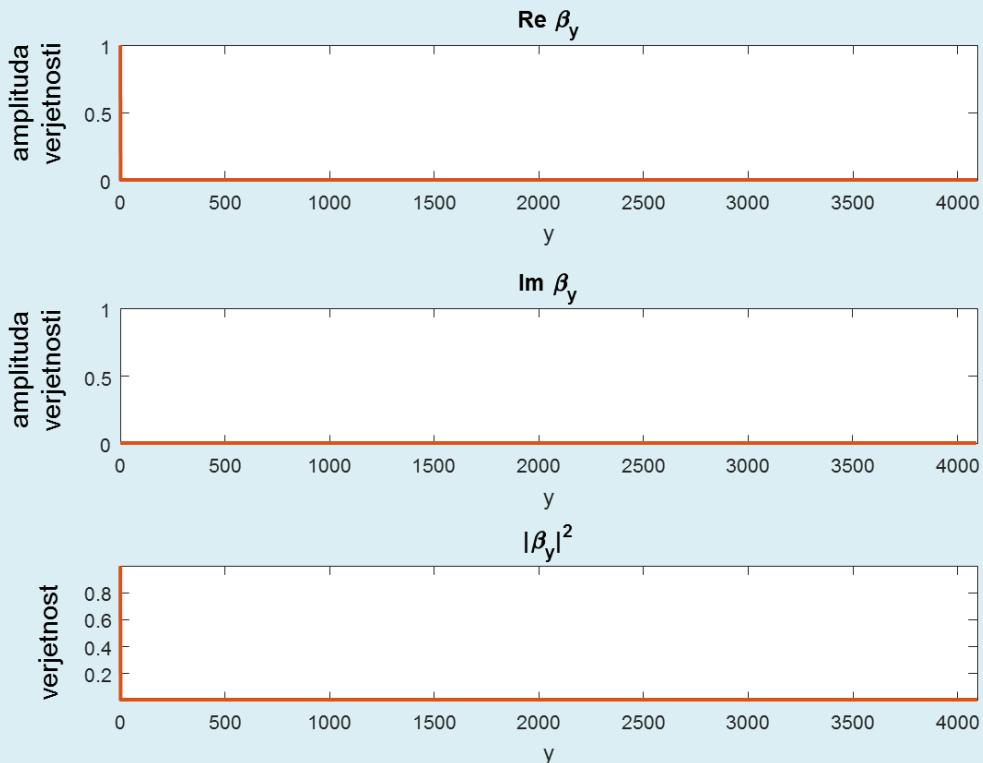
in

$$\sum_{z=0}^{Q-1} e^{\frac{i2\pi z(y-x)}{Q}} = Q. \quad (4.34)$$

Torej velja

$$\frac{1}{Q} \sum_{x=0}^{Q-1} \sum_{z=0}^{Q-1} \sum_{y=0}^{Q-1} \alpha_y e^{\frac{i2\pi zy}{Q}} e^{\frac{-i2\pi zx}{Q}} |x\rangle = \frac{1}{Q} \sum_{y=0}^{Q-1} Q \alpha_y |y\rangle = \sum_{y=0}^{Q-1} \alpha_y |y\rangle. \quad (4.35)$$

Zgled 4.10: Spremenimo zgled 2 tako, da kvantni register z 12 kvantnimi biti, ki je v klasičnem stanju $|00000000\rangle$, transformiramo s kvantno Fourierovo transformacijo, nato pa rezultat transformacije takoj obdelamo z inverzno kvantno Fourierovo transformacijo. Rezultat prikazuje slika 4.10. V skladu s pričakovanji smo po obeh transformacijah zopet prešli v začetno klasično stanje $|00000000\rangle$ (število 0).



Slika 4.10: Amplitude verjetnosti (zgornja dva grafa) in verjetnost meritve posameznega števila v osembitnem kvantnem registru po kvantni Fourierovi in inverzni kvantni Fourierovi transformaciji. Končno stanje je enako začetnemu stanju $|00000000\rangle$.

Kvantno Fourierovo transformacijo lahko zapišemo tudi v obliki unitarne matrike. Slednje zahteva tudi kvantna mehanika, saj prepoveduje nelinearne transformacije, torej lahko vse transformacije zapišemo z matričnim računom. Kvantno Fourierovo transformacijo v splošnem zapišemo z naslednjo matriko:

$$\mathbf{F} = \frac{1}{\sqrt{Q}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & W^3 & \cdots & W^{Q-1} \\ 1 & W^2 & W^4 & W^6 & \cdots & W^{2(Q-1)} \\ 1 & W^3 & W^6 & W^9 & \cdots & W^{3(Q-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{Q-1} & W^{2(Q-1)} & W^{3(Q-1)} & \cdots & W^{(Q-1)(Q-1)} \end{bmatrix}, \quad (4.36)$$

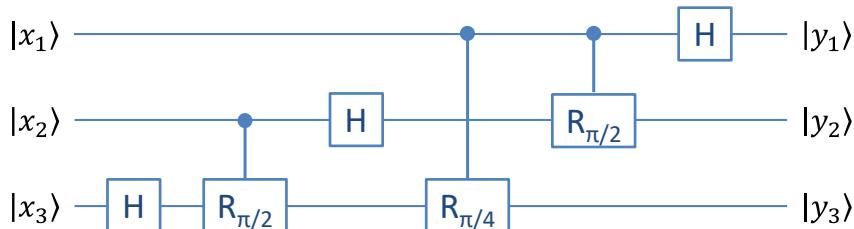
kjer je $W = e^{\frac{i2\pi}{Q}}$ Q-ti koren števila 1. Kot zahteva kvantna mehanika, je druga norma matrike \mathbf{F} enaka 1.

Inverzno kvantno Fourierovo transformacijo izračunamo tako, da matriko \mathbf{F} hermitsko transponiramo, to je transponiramo in nato kompleksno konjugiramo vse njene elemente:

$$\mathbf{F}^{-1} = \mathbf{F}^H. \quad (4.37)$$

Ker je matrika \mathbf{F} simetrična, je operacija transponiranja v bistvu odveč.

Kvantno Fourierovo transformacijo lahko učinkovito izvedemo s Hamaardovimi vrati (podpoglavlje 4.3.2) in kontroliranimi faznimi vrati (podpoglavlje 4.3.5) [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000]. Za njeno implementacijo potrebujemo $\mathcal{O}(n^2)$ vrat, kjer je n število kvantnih bitov v kvantnem registru. Primer implementacije kvantnega vezja, ki izvaja kvantno Fourierovo transformacijo nad registrom s tremi kvantnimi biti, prikazuje slika 4.11.



Slika 4.11: Kvantno vezje, ki izvaja kvantno Fourierovo transformacijo vsebine registra s tremi kvantnimi biti.

4.5 Meritev

Meritev kvantnega sistema smo razložili že v poglavju 3. Meritev je edina ireverzibilna kvantna operacija in je ne moremo predstaviti z unitarnimi matrikami. Meritev povzroči kolaps superpozicije vseh kvantnih stanj v eno samo klasično stanje.

Omenimo še, da trenutno številni simulatorji kvantnih sistemov na sodobnih računalnikih ob simuliranju meritve kolapsa superpozicije ne simulirajo (npr. Google Quantum Playground [Wro2014]). To je nedoslednost simulatorjev in ne predstavlja realnega stanja kvantne mehanike. Gre samo za poenostavitev oziroma trik, ki omogoča prihranek računske moči na klasičnem računalniku.

Naloge:

1. Katera števila so shranjena v kvantnem registru, ko so posamezni kvantni biti v naslednjih stanjih:
 - a) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$
 - b) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (0|0\rangle + 1|1\rangle)$
 - c) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$
 - d) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$
 - e) $(1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$
 - f) $(1|0\rangle + 0|1\rangle) \otimes (1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$
2. S tenzorskim produktom zapišite stanja kvantnih bitov, ki v štiribitnem kvantnem registru sočasno hranijo naslednja števila:

a) $\frac{1}{2} 0\rangle + \frac{1}{2} 1\rangle$	b) $\frac{1}{4} 0\rangle + \frac{1}{4} 1\rangle + \frac{1}{4} 2\rangle + \frac{1}{4} 3\rangle$
c) $\frac{1}{4} 12\rangle + \frac{1}{4} 13\rangle + \frac{1}{4} 14\rangle + \frac{1}{4} 15\rangle$	d) $\frac{1}{4} 8\rangle + \frac{1}{4} 9\rangle + \frac{1}{4} 10\rangle + \frac{1}{4} 11\rangle$
e) $\frac{1}{2} 10\rangle + \frac{1}{2} 14\rangle$	f) $\frac{1}{8} 0\rangle + \frac{1}{8} 1\rangle + \frac{1}{8} 2\rangle + \frac{1}{8} 3\rangle + \frac{1}{8} 8\rangle + \frac{1}{8} 9\rangle + \frac{1}{8} 10\rangle + \frac{1}{8} 11\rangle$

3. Katere izmed spodaj naštetih matrik so unitarne?

a) $\frac{1}{2}(1-i)|0\rangle + \frac{1}{2}(1-i)|1\rangle$

b) $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

c) $\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{2}(1+i)|1\rangle$

d) $\frac{1}{\sqrt{7}}(2-i)|0\rangle + \frac{1}{\sqrt{7}}(1-i)|1\rangle$

e) $\frac{2}{3}|0\rangle + \frac{1}{3}(2+i)|1\rangle$

f) $\frac{1}{3\sqrt{2}}(2-2i)|0\rangle + \frac{1}{3\sqrt{2}}(1+3i)|1\rangle$

4. Z vrati NE spremenite stanja naslednjih kvantnih bitov:

a) $\frac{1}{2}(1-i)|0\rangle + \frac{1}{2}(1-i)|1\rangle$

b) $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

c) $\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{2}(1+i)|1\rangle$

d) $\frac{1}{\sqrt{7}}(2-i)|0\rangle + \frac{1}{\sqrt{7}}(1-i)|1\rangle$

e) $\frac{2}{3}|0\rangle + \frac{1}{3}(2+i)|1\rangle$

f) $\frac{1}{3\sqrt{2}}(2-2i)|0\rangle + \frac{1}{3\sqrt{2}}(1+3i)|1\rangle$

5. S Hadamardovimi vrati spremenite stanja naslednjih kvantnih bitov:

a) $\frac{1}{2}(1-i)|0\rangle + \frac{1}{2}(1-i)|1\rangle$

b) $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

c) $\frac{1}{\sqrt{2}}i|0\rangle + \frac{1}{2}(1+i)|1\rangle$

d) $\frac{1}{\sqrt{7}}(2-i)|0\rangle + \frac{1}{\sqrt{7}}(1-i)|1\rangle$

e) $\frac{2}{3}|0\rangle + \frac{1}{3}(2+i)|1\rangle$

f) $\frac{1}{3\sqrt{2}}(2-2i)|0\rangle + \frac{1}{3\sqrt{2}}(1+3i)|1\rangle$

6. S kontroliranimi vrati NE spremenite stanja naslednjih dvobitnih kvantnih registrov:

a) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$

b) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$

c) $(1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$

d) $(1|0\rangle + 0|1\rangle) \otimes (1|0\rangle + 0|1\rangle)$

e) $(0|0\rangle + 1|1\rangle) \otimes (1|0\rangle + 0|1\rangle)$

7. S kontroliranimi faznimi vrati spremenite stanja naslednjih dvobitnih kvantnih registrov:

a) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$

b) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$

c) $(1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$

d) $(1|0\rangle + 0|1\rangle) \otimes (1|0\rangle + 0|1\rangle)$

e) $(0|0\rangle + 1|1\rangle) \otimes (1|0\rangle + 0|1\rangle)$

8. Zapišite matriko, ki izvede nadzorovana Hadamardova vrata.

9. S Toffolijevimi vrati spremenite stanja naslednjih tribitnih kvantnih registrov:

a) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$

b) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (0|0\rangle + 1|1\rangle)$

c) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$

d) $\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$

e) $(1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes (1|0\rangle + 0|1\rangle)$

f) $(1|0\rangle + 0|1\rangle) \otimes (1|0\rangle + 0|1\rangle) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)$

10. S kvantno Fourierovo transformacijo spremenite superpozicijo stanj štiribitnih kvantnih registrov, ki hranijo naslednja števila:

a) $\frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle$

b) $\frac{1}{4}|0\rangle + \frac{1}{4}|1\rangle + \frac{1}{4}|2\rangle + \frac{1}{4}|3\rangle$

c) $\frac{1}{4}|12\rangle + \frac{1}{4}|13\rangle + \frac{1}{4}|14\rangle + \frac{1}{4}|15\rangle$

d) $\frac{1}{4}|8\rangle + \frac{1}{4}|9\rangle + \frac{1}{4}|10\rangle + \frac{1}{4}|11\rangle$

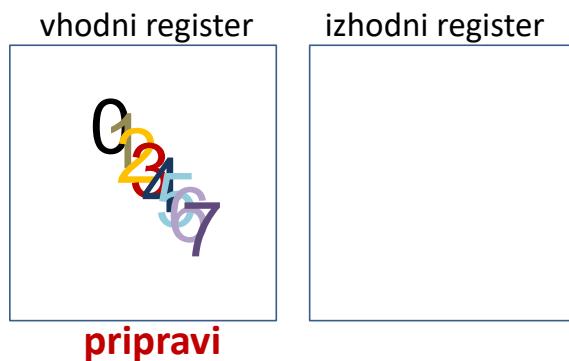
e) $\frac{1}{2}|10\rangle + \frac{1}{2}|14\rangle$

f) $\frac{1}{8}|0\rangle + \frac{1}{8}|1\rangle + \frac{1}{8}|2\rangle + \frac{1}{8}|3\rangle + \frac{1}{8}|8\rangle + \frac{1}{8}|9\rangle + \frac{1}{8}|10\rangle + \frac{1}{8}|11\rangle$

5 Kvantni algoritmi

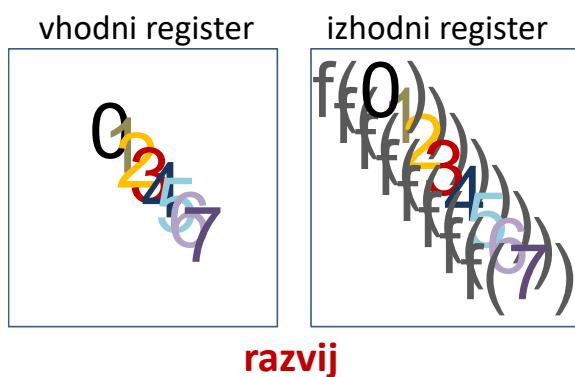
Kvanti algoritmi združujejo kvantna vrata in kvantne registre in izkoriščajo superpozicijo kvantnih stanj oziroma sočasni zapis več števil v kvantnih registrih. Večinoma sledijo programskemu vzorcu pripravi-razvij-izmeri [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000], ki ga podrobnejše opisujemo v nadaljevanju.

Korak pripravi navadno postavi vhodni kvantni register v stanje superpozicije vseh kvantnih stanj (slika 5.1). To lahko dosežemo ali s Hadamardovimi vrati, ki jih ločeno uporabimo nad vsakim kvantnim bitom vhodnega registra, ali pa s kvantno Fourierovo transformacijo.



Slika 5.1: Korak pripravi.

Korak razvij izračuna vrednost vnaprej določene funkcije f za vsa vhodna kvantna stanja. Izhodni register preide za vsa stanja vhodnega registra v superpozicijo vseh možnih rezultatov funkcije f . S tem kvantno preplete vhodni in izhodni register (slika 5.2).



Slika 5.2: Korak razvij.

Korak izmeri opravi meritev izhodnega registra. Meritev povzroči kolaps superpozicije vseh možnih izhodov funkcije f v eno izmed možnih klasičnih stanj. Hkrati zaradi kvantne prepleteneosti vhodnega in izhodnega registra delno ali v celoti kolabirajo tudi stanja vhodnega registra (slika 5.3). Povedano natančneje, meritev izhodnega registra povzroči kolaps vhodnega registra v superpozicijo vseh tistih vhodov x , ki prispevajo k izmerjeni vrednosti $f(x)$ v izhodnem registru. Če torej funkcija $f(x)$ vrne enak rezultat za različne vhodne vrednosti x , bo vhodni register še vedno v superpoziciji tistih vhodnih stanj, ki vodijo do izmerjene vrednosti funkcije $f(x)$. To lastnost prepleteneosti vhodnega in izhodnega kvantnega registra izkorišča tudi Shorov algoritmom (podpoglavlje 5.2).



Slika 5.3: Korak izmeri.

Funkcijo f izvedemo s pomočjo kvantnih vrat, ki smo jih opisali v poglavju 3. Pri tem nam je v pomoč spoznanje, da za katerokoli Boolovo funkcijo $f: \{0,1\}^n \rightarrow \{0,1\}$ obstaja unitarna transformacija kvantnega stanja [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000]:

$$|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle. \quad (5.1)$$

Vendar vseh funkcij f ne moremo izvesti učinkovito, torej z relativno majhnim številom kvantnih vrat [Aar2013]. Zato nas zanimajo predvsem tiste funkcije f , ki jih lahko sestavimo iz relativno majhnega števila kvantnih vrat.

5.1 Deutschov algoritem

Deutschov algoritem [Deu1992] sodi med najpreprostejše kvantne algoritme. Kljub svoji preprostosti nazorno prikaže prednosti kvantnih pred klasičnimi algoritmi. Zato je pogosto uporabljen kot uvodni zgled kvantnih algoritmov [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000].

Dana naj bo črna škatla (slika 5.4), ki prejme en bit in izračuna neznano enobitno funkcijo $f(a)$.



Slika 5.4: Črna škatla, ki izračunava funkcijo f .

Funkcija f je konstantna, če vedno vrača ali 0 ali 1, ne glede na stanje vhodnega bita. Funkcija f je uravnotežena, če je njen izhod odvisen od vhoda [Deu1992]. Imamo torej štiri možne enobitne funkcije f (slika 5.5).

Konstantni funkciji:

$$\begin{array}{l} f(0)=0 \text{ ali } f(0)=1 \\ f(1)=0 \text{ ali } f(1)=1 \end{array}$$

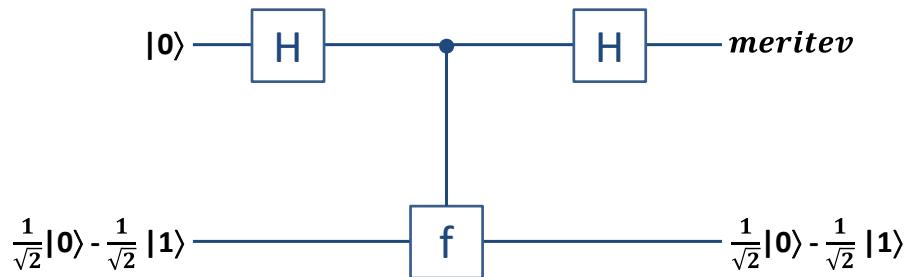
Uravnoteženi funkciji:

$$\begin{array}{l} f(0)=0 \text{ ali } f(0)=1 \\ f(1)=1 \text{ ali } f(1)=0 \end{array}$$

Slika 5.5: Definicija konstantne in uravnotežene funkcije f .

Radi bi vedeli, ali je neznana funkcija f konstantna ali uravnotežena. To lahko vedno ugotovimo z dvema izračunoma, tako da izračunamo $f(0)$ in $f(1)$. Ali lahko to ugotovimo z enim samim izračunom?

Odgovor z eno samo evalvacijo funkcije f dobimo s pomočjo naslednjega kvantnega algoritma [Deu1992]:



Slika 5.6: Diagram Deutschovega algoritma.

Deutschov algoritem bomo podrobneje pojasnili v podpoglavlju 5.1.1. Najprej pojasnimo izvedbo funkcije f . Funkcija f je dana, izvedena pa je s pomočjo kontroliranih U-vrat, in sicer tako, da velja [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000]:

$$|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle, \quad (5.2)$$

kjer je $|x\rangle$ zgornji in $|y\rangle$ spodnji kvantni bit v vezju na sliki 5.6 in smo z \oplus označili izključujoči ALI (angl. XOR). Spomnimo, da je funkcija f dana, le poznamo je ne. Ker imamo štiri možne implementacije funkcije f (slika 5.5), imamo tudi štiri možne izvedbe kontroliranih U-vrat.

Možnost 1: $f(0) = f(1) = 0$

V tem primeru nam preslikava $|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle$ vrne naslednje vrednosti:

- $|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|0\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|1\rangle$

Nadzorovana U-vrata, ki izvedejo to preslikavo, zapišemo kot:

$$\mathbf{U}_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.3)$$

Pri tem smo zgornjo matriko zapisali v bazi prostora, ki jo določa enačba (4.22).

Možnost 2: $f(0) = f(1) = 1$

V tem primeru nam preslikava $|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle$ vrne naslednje vrednosti:

- $|0\rangle|0\rangle \rightarrow |0\rangle|1\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

Nadzorovana U-vrata, ki izvedejo to preslikavo, zapišemo kot

$$\mathbf{U}_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (5.4)$$

Možnost 3: $f(0) = 0, f(1) = 1$

V tem primeru nam preslikava $|x\rangle|y\rangle \rightarrow |x\rangle|f(x)\oplus y\rangle$ vrne naslednje vrednosti:

- $|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

Nadzorovana U-vrata, ki izvedejo to preslikavo, zapišemo kot

$$\mathbf{U}_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (5.5)$$

Možnost 4: $f(0) = 1, f(1) = 0$

V tem primeru nam preslikava $|x\rangle|y\rangle \rightarrow |x\rangle|f(x)\oplus y\rangle$ vrne naslednje vrednosti:

- $|0\rangle|0\rangle \rightarrow |0\rangle|1\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|0\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|1\rangle$

Nadzorovana U vrata, ki izvedejo to preslikavo, zapišemo kot

$$\mathbf{U}_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.6)$$

Sedaj lahko podamo razlago Deutschovega algoritma.

5.1.1 Razlaga algoritma

Po prvih Hadamardovih vratih na sliki 5.6 je stanje obeh kvantnih bitov [Eke2008]:

$$\frac{1}{\sqrt{2}}(1|0\rangle + 1|1\rangle) \otimes \frac{1}{\sqrt{2}}(1|0\rangle - 1|1\rangle) \quad (5.7)$$

V primeru **možnosti 1** ($f(0) = f(1) = 0$) imamo po kontroliranih U-vratih naslednje stanje kvantnih bitov (če izpustimo normalizacijo amplitud verjetnosti s $\sqrt{2}$):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}. \quad (5.8)$$

V tem primeru lahko stanje dveh kvantnih bitov

$$\frac{1}{2}(1|00\rangle - 1|01\rangle + 1|10\rangle - 1|11\rangle) \quad (5.9)$$

zapišemo kot tenzorski produkt

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad (5.10)$$

torej je zgornji kvantni bit na sliki 5.6 v stanju $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, spodnji pa v stanju $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Po drugih Hadamardovih vratah na sliki 5.6 je zgornji kvantni bit v stanju $|0\rangle$:

$$H\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle. \quad (5.11)$$

V primeru **možnosti 2** ($f(0) = f(1) = 1$) imamo po kontroliranih U-vratih naslednje stanje kvantnih bitov:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \end{bmatrix}. \quad (5.12)$$

V tem primeru lahko stanje dveh kvantnih bitov

$$\frac{1}{2}(-1|00\rangle + 1|01\rangle - 1|10\rangle + 1|11\rangle) \quad (5.13)$$

zapišemo kot tenzorski produkt

$$\frac{1}{\sqrt{2}}(-|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad (5.14)$$

torej je zgornji kvantni bit na sliki 5.6 v stanju $\frac{1}{\sqrt{2}}(-|0\rangle - |1\rangle)$, spodnji pa v stanju $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Po drugih Hadamardovih vratah na sliki 5.6 je zgornji kvantni bit v stanju $|0\rangle$:

$$H\left(-\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} = |0\rangle. \quad (5.15)$$

V primeru **možnosti 3** ($f(0) = 0, f(1) = 1$) imamo po kontroliranih U-vratih naslednje stanje kvantnih bitov:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \end{bmatrix}. \quad (5.16)$$

V tem primeru lahko stanje dveh kvantnih bitov

$$\frac{1}{2}(+1|00\rangle - 1|01\rangle - 1|10\rangle + 1|11\rangle) \quad (5.17)$$

zapišemo kot tenzorski produkt

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad (5.18)$$

torej je zgornji kvantni bit na sliki 5.6 v stanju $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, spodnji pa ostaja v stanju $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Po drugih Hadamardovih vratah na sliki 5.6 je zgornji kvantni bit v stanju $|0\rangle$:

$$H\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle. \quad (5.19)$$

V primeru **možnosti 4** ($f(0) = 1, f(1) = 0$) imamo po kontroliranih U-vratih naslednje stanje kvantnih bitov:

$$\begin{bmatrix} 0 & \textcolor{red}{1} & 0 & 0 \\ \textcolor{red}{1} & 0 & 0 & 0 \\ 0 & 0 & \textcolor{red}{1} & 0 \\ 0 & 0 & 0 & \textcolor{red}{1} \end{bmatrix} \cdot \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ +1 \\ +1 \\ -1 \end{bmatrix}. \quad (5.20)$$

V tem primeru lahko stanje dveh kvantnih bitov

$$\frac{1}{2}(-1|00\rangle + 1|01\rangle + 1|10\rangle - 1|11\rangle) \quad (5.21)$$

zapišemo kot tenzorski produkt

$$\frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad (5.22)$$

torej je zgornji kvantni bit na sliki 5.6 v stanju $\frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$, spodnji pa ostaja v stanju $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Po drugih Hadamardovih vratih na sliki 5.6 je zgornji kvantni bit v stanju $|0\rangle$:

$$H\left(-\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = |1\rangle. \quad (5.23)$$

Torej je po drugih Hadamardovih vratih stanje prvega kvantnega bita [Eke2008]:

- $|0\rangle$ če je f konstantna
- $|1\rangle$ če je f uravnotežena.

Deutschov algoritem je torej z eno samo evalvacijo funkcije f ugotovil, ali je funkcija konstantna ali uravnotežena. Poleg U-vrat smo potrebovali še dvoje Hadamardovih vrat.

Računska učinkovitost Deutschovega algoritma pride do izraza šele, če predpostavimo, da je evalvacija funkcije f stroškovno ali časovno izrazito neugodna. V nasprotnem primeru je težje upravičiti pohitritev Deutschovega algoritma v primerjavi s klasičnimi vrednotenji funkcije f .

5.2 Groverjev algoritem

Groverjev kvantni algoritem [Gro1996] omogoča zelo učinkovito iskanje elementa v neurejeni podatkovni bazi. Klasičen računalnik mora v neurejeni bazi s Q elementi v povprečju opraviti $Q/2$ poizvedb (v najboljšem primeru je potrebna 1, v najslabšem pa Q poizvedb). Časovna zahtevnost neurejenega iskanja je torej $\mathcal{O}(Q)$. Groverjev kvantni algoritem najde iskani element s časovno zahtevnostjo $\mathcal{O}(\sqrt{Q})$. Za velike baze (za velik Q) je lahko to zelo velika pohitritev. Dokazano je tudi bilo, da je takšna pohitritev optimalna in da noben kvantni algoritem ne preseže te učinkovitosti (vsako kvantno iskanje potrebuje vsaj toliko iskanj kot Groverjev algoritem) [Zal1999].

Algoritem predpostavi, da lahko posamezen element baze označimo z indeksom x . Če ima baza Q elementov, mora biti indeks velikosti vsaj $N = \lceil \log_2 Q \rceil$ bitov, kjer smo z $\lceil \cdot \rceil$ označili operacijo zaokroževanja navzgor. Indekse vseh elementov baze shranimo v kvantni register z N biti.

Algoritem tudi predpostavi, da imamo na voljo označevalno funkcijo $f(x)$, ki prejme indeks x in vrne vrednost 1, če je x indeks iskanega elementa, in 0, če x ni indeks iskanega elementa. Poudarimo, da je lahko v bazi več primerkov iskanega elementa. V tem primeru bo funkcija $f(x)$ vrnila vrednost 1 za več različnih indeksov.

Funkcija $f(x)$ ne izvaja iskanja po bazi, le prepozna indeks iskanega elementa in ga označi. Strošek te označevalne funkcije običajno ni vključen v vrednotenje časovne zahtevnosti Groverjevega algoritma, saj navadno predpostavimo, da je funkcija zelo hitra. Ali je to res v primeru vseh podatkovnih baz ali ne, ostaja odprt vprašanje in tudi ena izmed kritik Groverjevega algoritma [Aar2013, Bro2014, Eke2008, Mon2016, Nie2000].

Algoritem izvedemo v štirih korakih [Gro1996]:

1. Najprej postavi kvantni register v stanje superpozicije vseh indeksov:

$$|\omega\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} 1|x\rangle \quad (5.24)$$

Nato iterativno ponavlja naslednja dva koraka.

2. S pomočjo označevalne funkcije $f(x)$ spremenimo predznak amplitude verjetnosti indeksa iskanega elementa. Predzname amplitude verjetnosti indeksov ostalih elementov pustimo nespremenjene.

$$\begin{aligned} \alpha_x|x\rangle &\rightarrow -\alpha_x|x\rangle, & \text{če } f(x) = 1, \\ \alpha_x|x\rangle &\rightarrow \alpha_x|x\rangle, & \text{če } f(x) = 0. \end{aligned} \quad (5.25)$$

3. Izračunamo inverz amplitud verjetnosti vseh indeksov okoli njihove povprečne vrednosti $\bar{\alpha}$:

$$\bar{\alpha} = \frac{1}{Q} \sum_{x=0}^{Q-1} \alpha_x \quad (5.26)$$

$$\forall x: \alpha_x = 2 \cdot \bar{\alpha} - \alpha_x$$

Koraka 2 in 3 ponovimo $\frac{\pi}{4} \sqrt{\binom{Q}{k}}$ -krat, kjer je k število elementov v bazi, ki so enaki iskanemu elementu. Omenjeno število iteracij je dokazano optimalno in ga ni priporočljivo preseči (glejte zgled 5.3).

4. Opravimo meritev kvantnega registra.

Zgled 5.1: Dana je baza štirih črk: A, B, C in D, shranjenih v naključnem vrstnem redu, na primer C, A, B, D. V njej želimo poiskati črko A. Elementom baze dodelimo indekse, in sicer črki C indeks 0, črki A indeks 1, črki B indeks 2 in črki D indeks 3. Ker iščemo črko A, ki ima indeks 1, imamo naslednjo označevalno funkcijo:

$$f(0) = 0, f(1) = 1, f(2) = 0 \text{ in } f(3) = 0.$$

Indekse elementov shranimo v kvantni register z $N=2$ biti. V prvem koraku Groverjevega algoritma postavimo kvantni register v naslednjo superpozicijo stanj:

$$|\omega\rangle = \frac{1}{\sqrt{4}} \sum_{x=0}^{4-1} 1 |x\rangle = \frac{1}{2} |0\rangle + \frac{1}{2} |1\rangle + \frac{1}{2} |2\rangle + \frac{1}{2} |3\rangle.$$

V drugem koraku s pomočjo označevalne funkcije $f(x)$ spremenimo predznak amplitude verjetnosti drugega elementa:

$$\frac{1}{2} |0\rangle - \frac{1}{2} |1\rangle + \frac{1}{2} |2\rangle + \frac{1}{2} |3\rangle.$$

V tretjem koraku izračunamo inverz amplitud verjetnosti vseh indeksov okoli njihove povprečne vrednosti $\bar{\alpha}$:

$$\bar{\alpha} = \frac{1}{4} \left(\frac{1}{2} - \frac{1}{2} + \frac{1}{2} + \frac{1}{2} \right) = \frac{1}{4},$$

$$\begin{aligned} \forall x: \alpha_x &= 2 \cdot \bar{\alpha} - \alpha_x \Rightarrow 2 \cdot \frac{1}{4} - \left(\frac{1}{2} |0\rangle - \frac{1}{2} |1\rangle + \frac{1}{2} |2\rangle + \frac{1}{2} |3\rangle \right) = \\ &0|0\rangle + 1|1\rangle + 0|2\rangle + 0|3\rangle. \end{aligned}$$

V zadnjem, četrtem koraku opravimo meritvev. Ker imajo indeksi $|0\rangle$, $|2\rangle$ in $|3\rangle$ amplitude verjetnosti 0, je tudi verjetnost, da jih izmerimo, enaka 0. Nasprotno pa je verjetnost, da bomo izmerili indeks $|1\rangle$, enaka $|1|^2 = 1$. Z meritvijo bomo torej vedno prišli do indeksa $|1\rangle$.

Potrebovali smo eno samo iteracijo Groverjevega algoritma. Podajmo še teoretični izračun števila iteracij: $\frac{\pi}{4} \sqrt{\left(\frac{Q}{k}\right)} = \frac{\pi}{4} \sqrt{\left(\frac{4}{1}\right)} = \frac{\pi}{2} = 1,57$.

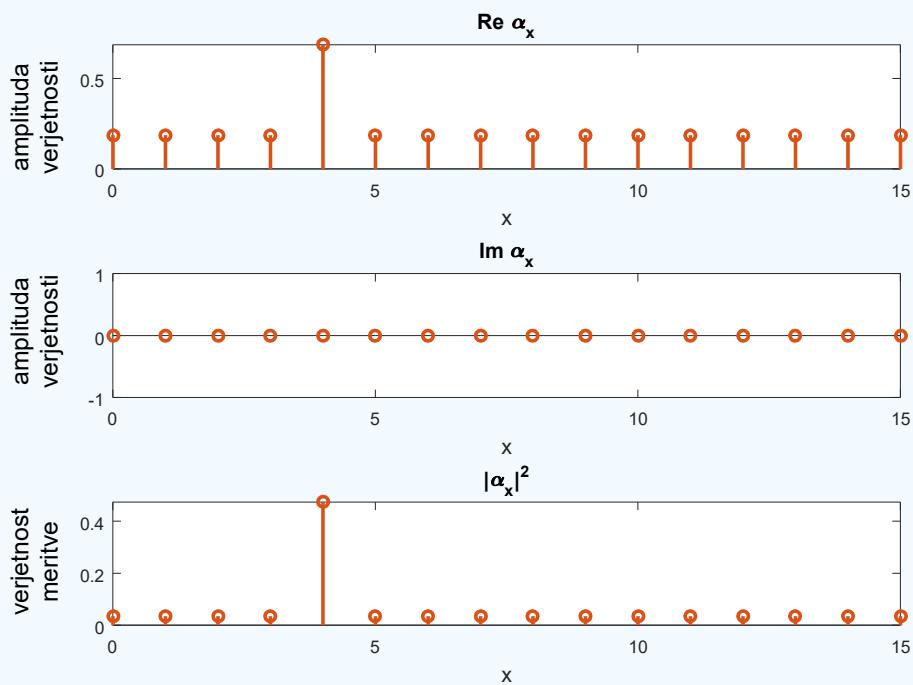
Zgled 5.2: Dana je baza šestnajstih skritih gesel. V njej želimo poiskati geslo, ki dešifrira niz zakodiranih znakov. Elementom baze dodelimo indekse od 0 do 15. Predpostavimo, da naš zakodirani niz znakov dešifrira samo geslo, ki je v bazi shranjeno v elementu z indeksom 4. Imamo torej naslednjo označevalno funkcijo:

$$f(x) = \begin{cases} 1, & \text{ko } x = 4 \\ 0, & \text{drugače} \end{cases}.$$

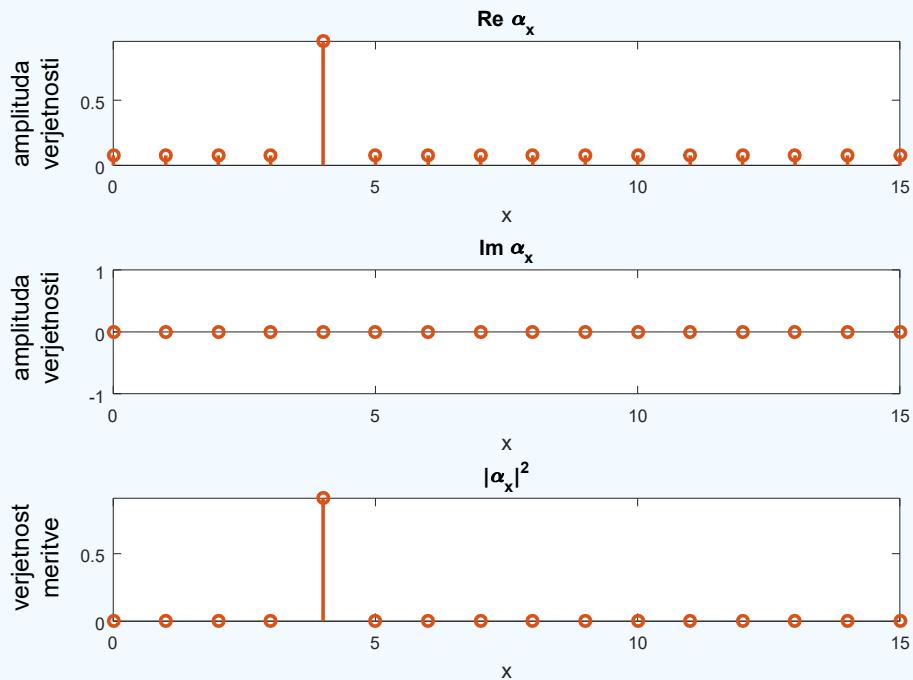
Indekse elementov shranimo v kvantni register z $N=4$ biti. V prvem koraku Groverjevega algoritma postavimo kvantni register v naslednjo superpozicijo stanj:

$$|\omega\rangle = \frac{1}{\sqrt{16}} \sum_{x=0}^{15} 1|x\rangle$$

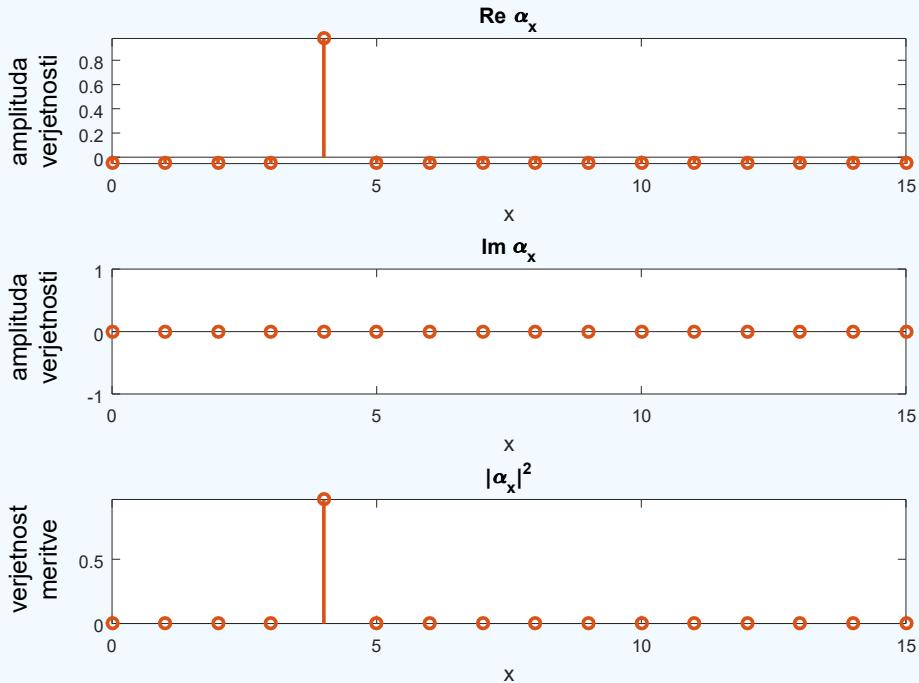
Nato iterativno izvajamo drugi in tretji korak Groverjevega algoritma. Po prvi iteraciji dobimo amplitude verjetnosti in verjetnosti, ki jih prikazuje slika 5.7. Stanje amplitud verjetnosti po drugi in tretji iteraciji prikazujeta slike 5.8 in 5.9.



Slika 5.7: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po prvi iteraciji drugega in tretjega koraka Groverjevega algoritma.



Slika 5.8: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po drugi iteraciji drugega in tretjega koraka Groverjevega algoritma.



Slika 5.9: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po tretji iteraciji drugega in tretjega koraka Groverjevega algoritma.

Ker je $\frac{\pi}{4} \sqrt{\binom{Q}{k}} = \frac{\pi}{4} \sqrt{\binom{16}{1}} = \pi = 3,14$, po tretji iteraciji opravimo meritve. Indeks $|4\rangle$ bomo izmerili z verjetnostjo 0,9613, katerikoli drugi indeks pa z verjetnostjo 0,0026.

Zgled 5.3: Kaj bi se zgodilo, če bi pred meritvijo v zgledu 5.2 opravili še četrto in peto iteracijo drugega in tretjega koraka Groverjevega algoritma?

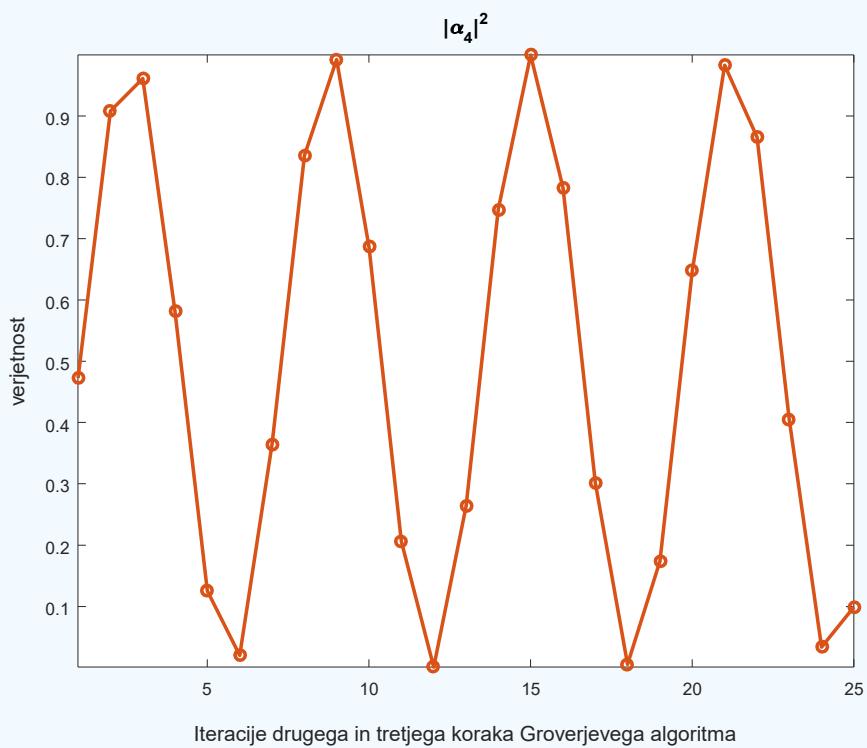
Po četrti iteraciji bi verjetnost, da bomo izmerili indeks $|4\rangle$ padla na 0,5817, verjetnost meritve kateregakoli drugega indeksa pa bi narasla na 0,0278.

Po peti iteraciji bi verjetnost meritve indeksa $|4\rangle$ padla na 0,1255, verjetnost meritve kateregakoli drugega indeksa pa bi narasla na 0,0583.

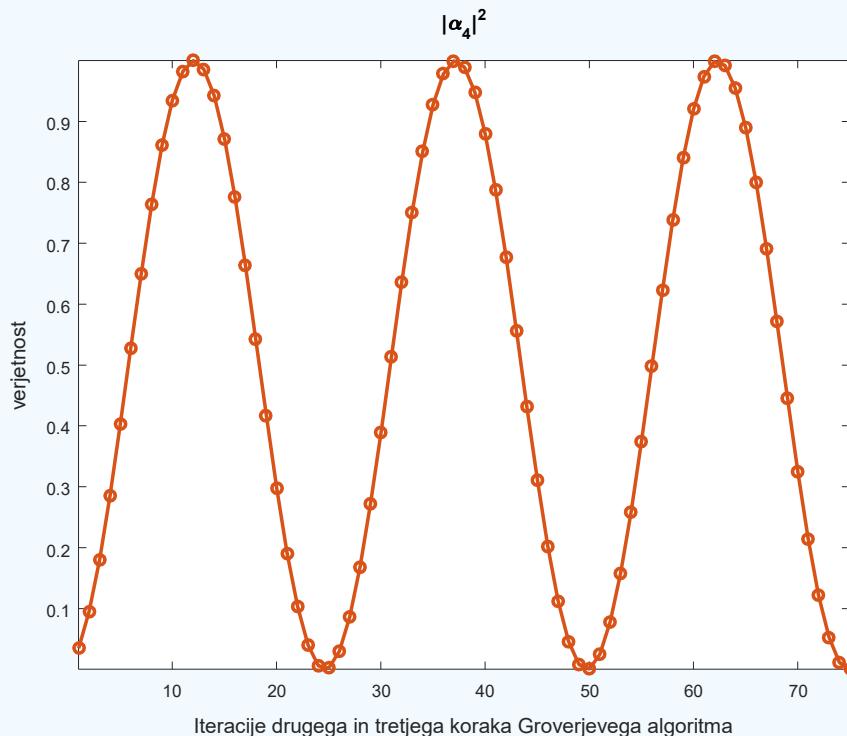
Nihanje verjetnosti meritve indeksa $|4\rangle$ v odvisnosti od števila iteracij drugega in tretjega koraka Gorverjevega algoritma prikazuje slika 5.10.

Slika 5.11 kaže nihanje verjetnosti meritve iskanega indeksa v odvisnosti od števila iteracij drugega in tretjega koraka Groverjevega algoritma v bazi z 256 elementi ($Q=256$, $N=8$) .

Vidimo, da je iteracije Groverjevega algoritma res treba končati po $\frac{\pi}{4} \sqrt{\binom{Q}{k}}$ korakih.



Slika 5.10: Nihanje verjetnosti meritve indeksa $|4\rangle$ v odvisnosti od števila iteracij drugega in tretjega koraka Groverjevega algoritma v bazi s 16 elementi.



Slika 5.11: Nihanje verjetnosti meritve iskanega indeksa v odvisnosti od števila iteracij drugega in tretjega koraka Groverjevega algoritma v bazi z 256 elementi.

Prvi korak Groverjevega algoritma lahko izvedemo ali s pomočjo Hadamardovih vrat ali pa s kvantno Fourierovo transformacijo [Aar2013, Bro2014, Eke2008, Koš2009, Mon2016, Nie2000]. Drugi korak, definiran v enačbi (5.25), lahko izvedemo z naslednjo unitarno matriko:

$$\mathbf{U}_f = \mathbf{I} - 2 \cdot |f\rangle\langle f|, \quad (5.27)$$

kjer je \mathbf{I} matrična identiteta velikosti $Q \times Q$, $|f\rangle$ je označevalni vektor dimenzije $Q \times 1$, ki ima na poziciji iskanega elementa v bazi zapisano vrednost 1, vse ostale vrednosti pa so enake 0 (glejte zgled 5.4).

Notacija $|x\rangle\langle x|$ označuje zunanji produkt vektorja $|x\rangle$ s samim sabo [Koš2009]. Zunanji produkt vektorjev je znan tudi kot tenzorski ali diadni produkt in je za vektorja $x = [x_1, x_2 \dots x_Q]^T$ in $y = [y_1, y_2 \dots y_Q]^T$ definiran kot naslednja matrika velikosti $Q \times Q$:

$$|x\rangle\langle y| = x \cdot y^T = \begin{bmatrix} x_1 y_1 & \cdots & x_1 y_Q \\ \vdots & \ddots & \vdots \\ x_Q y_1 & \cdots & x_Q y_Q \end{bmatrix} \quad (5.28)$$

Podobno lahko tretji korak Groverjevega algoritma, ki je definiran v enačbi (5.26), izvedemo z naslednjo unitarno matriko [Aar2013, Bro2014, Eke2008, Koš2009, Mon2016, Nie2000]:

$$\mathbf{U}_\omega = 2 \cdot |\omega\rangle\langle\omega| - \mathbf{I}, \quad (5.29)$$

kjer je \mathbf{I} matrična identiteta velikosti $Q \times Q$, $|\omega\rangle$ pa je definiran v enačbi (5.24) in je izražen v bazi vseh Q klasičnih stanj kvantnega registra (glejte zgled 5.4).

Zgled 5.4: Izračunajmo unitarna operatorja \mathbf{U}_f in \mathbf{U}_ω za primer iskanja v bazi črk, ki je podan v zgledu 5.1. V bazi $Q = 4$ črk smo iskali drugi element, zato je označevalni vektor $|f\rangle$ definiran kot

$$|f\rangle = [0, 1, 0, 0]^T.$$

Torej je

$$|f\rangle\langle f| = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

in je matrika $\mathbf{U}_f = \mathbf{I} - 2 \cdot |f\rangle\langle f|$ enaka

$$\mathbf{U}_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - 2 \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matrika \mathbf{U}_f torej spremeni predznak amplitude verjetnosti drugega (iskanega) indeksa elementa v bazi, kakor je to zahtevano v enačbi (5.25).

Vektor $|\omega\rangle$ je za zgled 5.1 definiran kot

$$|\omega\rangle = \frac{1}{\sqrt{4}} \sum_{x=0}^{4-1} 1 |x\rangle = \frac{1}{2} |0\rangle + \frac{1}{2} |1\rangle + \frac{1}{2} |2\rangle + \frac{1}{2} |3\rangle.$$

Ko upoštevamo matrični zapis klasičnih stanj kvantnega registra z dvema kvantnima bitoma

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad |2\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad |3\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

imamo

$$|\omega\rangle = \frac{1}{\sqrt{4}} \sum_{x=0}^{4-1} 1 |x\rangle = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Zunanji produkt $|\omega\rangle\langle\omega|$ je torej

$$|\omega\rangle\langle\omega| = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

in matrika $\mathbf{U}_\omega = 2 \cdot |\omega\rangle\langle\omega| - \mathbf{I}$ je

$$\mathbf{U}_\omega = \frac{2}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}.$$

Preverite, da matrika \mathbf{U}_ω res izračuna inverz amplitud verjetnosti, ki je definiran v enačbi (5.25), in da sta obe matriki \mathbf{U}_f in \mathbf{U}_ω unitarni:

$$\mathbf{U}_f \cdot \mathbf{U}_f^T = \mathbf{I}$$

in

$$\mathbf{U}_\omega \cdot \mathbf{U}_\omega^T = \mathbf{I}.$$

5.3 Shorov algoritem

Shorov kvantni algoritem za faktorizacijo celih števil [Sho1997] je daleč najbolj vpliven in slaven med vsemi kvantnimi algoritmi. Zaradi njega je močno poskočilo zanimanje za razvoj kvantnih računalnikov, saj algoritem omogoča učinkovito dešifriranje asimetričnih kodirnikov (na primer v algoritmu RSA in Diffie-Hellmanovem algoritmu za izmenjavo skritih ključev), s katerimi so zaščitene milijarde evrov premoženja (vsi podatki bančnih sistemov, nepremičninskih trgov, borz, vlad, velikih podjetij itd.). Peter Shor je pokazal, kako lahko na kvantnem računalniku velika števila faktoriziramo v polinomskem času [Sho1997], na klasičnem računalniku pa je za to potreben eksponenten čas. Za izum algoritma in njegov doprinos k teoretičnemu računalništву je Peter Shor leta 1999 prejel Gödelovo nagrado.

Shorov algoritem poišče faktorizacijo števila N , če je dana perioda r funkcije $f(x) = a^x \bmod N$, kjer je a poljubno celo število, ki je N tuje (največji skupni delitelj števil a in N je 1), N pa je sestavljeni število: $N = pq$, kjer sta p in q praštevili.

Algoritem je sestavljen iz klasičnega in kvantnega dela.

5.3.1 Klasični del Shorovega algoritma

Podajmo psevdokod klasičnega dela Shorovega algoritma [Sho1997], ki ga zaradi povzemamo iz opisa na Wikipediji [WiS2015]:

1. Izberi naključno celo število $a < N$.
2. S pomočjo Evklidovega algoritma izračunaj največji skupni delitelj (angl. *greatest common divisor*) števil a in N : $gcd(a, N)$.
3. Če $gcd(a, N) \neq 1$, potem smo našli netrivialni faktor števila N (trivialna faktorja števila N sta 1 in N), torej smo na cilju.
4. Če $gcd(a, N) = 1$, uporabi kvantni del Shorovega algoritma in poišči periodo r funkcije

$$f(x) = a^x \bmod N. \quad (5.30)$$

5. Če je r liho število, se vrni na korak 1.
6. Če $a^{r/2} \bmod N = -1$, se vrni na korak 1.
7. Drugače je $\gcd(a^{r/2} - 1, N)$ netrivialen faktor števila N .

Povzemimo še dokaz klasičnega dela Shorovega algoritma [Wis2015, Aar2013, Bro2014, Eke2008, Mon2016, Nie2000, Sho1997]. Po definiciji periode r imamo $f(r) = a^r \bmod N = 1$. Torej N deli $a^r - 1$. Po koraku 5 imamo takšen a , da je $\gcd(a, N) = 1$ in r sodo število.

Definirajmo $b = a^{r/2} \bmod N$. Torej je b kvadratni koren števila 1 po modulu N . Velja $b \neq 1$, saj je po definiciji perioda funkcije $f(x)$ enaka r in ne $r/2$. Korak 6 zagotavlja tudi $b \neq -1$.

Trdimo, da je $d = \gcd(b - 1, N)$ netrivialen faktor števila N (torej $d \neq 1$ in $d \neq N$).

1. Ker velja $d < b - 1 < N$, velja tudi $d \neq N$.
2. Če bi veljalo $d = \gcd(b - 1, N) = 1$, potem bi po Bezoutovi enakosti (angl. *Bézout's identity*), poimenovani po francoskem matematiku Étiennu Bézoutu, obstajali takšni celi števili u in v , da bi veljalo [Béz1779]

$$(b - 1)u + Nv = 1. \quad (5.31)$$

Ko pomnožimo obe strani enačbe (5.31) z $(b + 1)$, dobimo:

$$(b^2 - 1)u + N(b + 1)v = b + 1.$$

Ker N deli $b^2 - 1 = a^r - 1$, bi moral glede na zgornjo enačbo N deliti tudi $(b + 1)$, torej bi veljalo $b \bmod N = -1$, kar je v nasprotju s korakom 6.

Torej je $d = \gcd(b - 1, N)$ res netrivialen faktor števila N .

Opomba: Zgornji dokaz temelji na predpostavki, da obstaja takšno število $b = a^{r/2} \bmod N$, da $b \neq -1$ in $b \neq 1$. Obstoj takšnega števila b zagotavlja teorem kitajskih ostankov (angl. *Chinese remainder theorem*) [Kat2007], saj je $N = pq$ sestavljen iz praštevil.

Zgled 5.5: Faktorizirajmo število $N = 15$. Izberimo $a = 7$ in izračunajmo vrednosti funkcije $f(x) = a^x \bmod N$:

$$\begin{aligned}f(0) &= 7^0 \bmod 15 = 1, \\f(1) &= 7^1 \bmod 15 = 7, \\f(2) &= 7^2 \bmod 15 = 4, \\f(3) &= 7^3 \bmod 15 = 13, \\f(4) &= 7^4 \bmod 15 = 1, \\f(5) &= 7^5 \bmod 15 = 7, \\f(6) &= 7^6 \bmod 15 = 4, \\f(7) &= 7^7 \bmod 15 = 13, \\f(8) &= 7^8 \bmod 15 = 1,\end{aligned}$$

Perioda funkcije $f(x) = 7^x \bmod 15$ je $r = 4$. Perioda je torej soda in izpolnjuje pogoj v 5. koraku klasičnega dela Shorovega algoritma. Preverimo še pogoj v 6. koraku: $7^{4/2} \bmod 15 = 4 \neq -1$. Sedaj lahko izračunamo $p = \gcd(7^2 - 1, 15) = \gcd(48, 15) = 3$. Drugi faktor je $q = 15/3 = 5$.

Zgled 5.6: Faktorizirajmo število $N = 35$. Izberimo $a = 3$ in izračunajmo vrednosti funkcije $f(x) = a^x \bmod N$:

$$\begin{aligned}f(0) &= 3^0 \bmod 35 = 1, \\f(1) &= 3^1 \bmod 35 = 3, \\f(2) &= 3^2 \bmod 35 = 9, \\f(3) &= 3^3 \bmod 35 = 27, \\f(4) &= 3^4 \bmod 35 = 11, \\f(5) &= 3^5 \bmod 35 = 33, \\f(6) &= 3^6 \bmod 35 = 29, \\f(7) &= 3^7 \bmod 35 = 17, \\f(8) &= 3^8 \bmod 35 = 16, \\f(9) &= 3^9 \bmod 35 = 13, \\f(10) &= 3^{10} \bmod 35 = 4, \\f(11) &= 3^{11} \bmod 35 = 12, \\f(12) &= 3^{12} \bmod 35 = 1,\end{aligned}$$

Perioda funkcije $f(x) = 3^x \bmod 35$ je $r = 12$ in izpolnjuje pogoj v 5. koraku klasičnega dela Shorovega algoritma. Tudi pogoj v 6. koraku je izpolnjen, saj je $3^6 \bmod 35 = 29 \neq -1$. Sedaj lahko izračunamo $p = \gcd(3^6 - 1, 35) = \gcd(728, 35) = 7$. Drugi faktor je $q = 35/7 = 5$.

Zgled 5.7: Faktorizirajmo število $N = 319$. Izberimo $a = 7$ in izračunajmo vrednosti funkcije $f(x) = 7^x \bmod 319$ in izmerimo periodo $r = 70$. Perioda izpoljuje pogoja v 5. in 6. koraku klasičnega dela Shorovega algoritma in $p = \gcd(7^{35} - 1, 319) = 29$. Drugi faktor je $q = 319/29 = 11$.

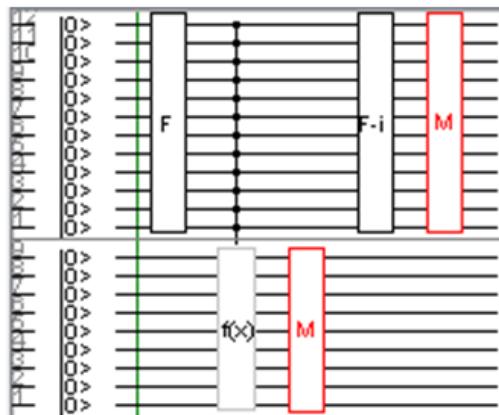
5.3.2 Shorov algoritem – kvanti del

Algoritem poišče periodo funkcije $f(x) = a^x \bmod N$, kjer je a poljubno število, ki je N tuje ($\gcd(a, N) = 1$), N pa je sestavljeno število: $N = pq$, kjer sta p in q praštevili.

Potrebna strojna oprema [Sho1997]:

- vhodni kvantni register takšne velikosti Q , da je vanj možno shraniti število N^2 ,
- izhodni kvantni register takšne velikosti P , da je vanj možno shraniti število N ,
- Fourierova kvantna transformacija in njen inverz,
- kvantno vezje, ki izvede funkcijo $f(x) = a^x \bmod N$ (za vsak a in za vsak N potrebujemo posebno vezje).

Potek kvantnega dela Shorovega algoritma prikazuje slika 5.12. Na začetku sta oba registra v stanju 0, nato pa si sledijo (od leve proti desni) kvantna Fourierova transformacija, funkcija $f(x)$, meritev izhodnega registra, inverzna kvantna Fourierova transformacija in meritev vhodnega registra.



Slika 5.12: Vezje kvantnega dela Shorovega algoritma z vhodnim registrom velikosti 12 kvantnih bitov in izhodnim registrom velikosti 9 kvantnih bitov (vir slike: simulator JQuantum [Vri2010]).

Posamezne korake algoritma opisujemo v nadaljevanju [Wis2015, Aar2013, Bro2014, Eke2008, Mon2016, Nie2000, Sho1997]:

1. Inicializacija:

- Vhodni kvantni register je v klasičnem stanju 0.
- Izhodni kvantni register v klasičnem stanju 0.

2. Superpozicija vhodnega registra:

- S kvantno Fourierovo transformacijo (ali pa Hadamardovo transformacijo) postavimo vhodni kvantni register v popolno superpozicijo vseh možnih stanj:

$$\sum_x \frac{1}{Q} |x\rangle. \quad (5.32)$$

- Izhodni kvantni register je še vedno v klasičnem stanju 0.

3. Apliciranje kvantne funkcije $f(x)$:

- Vhodni kvantni register je še vedno v stanju $\sum_x \frac{1}{Q} |x\rangle$.
- Izhodni kvantni register je v stanju $f\left(\sum_x \frac{1}{Q} |x\rangle\right) = \frac{1}{Q} \sum_x f(|x\rangle)$. Ker ima funkcija periodo r , zavzame samo r različnih vrednosti. Vse so enakovredno zastopane v izhodnem registru (vse imajo eno verjetnost, da jih izmerimo).
- Vhodni in izhodni kvantni register sta kvantno prepletena (prepletla ju je funkcija $f(x)$).

4. Meritev izhodnega registra:

- Izhodni kvantni register kolabira v eno samo opazovano vrednost $y_0 = f(x_0)$ (eno izmed tistih, ki so bile prej v superpoziciji izhodnega registra).
- Vhodni register posledično kolabira v superpozicijo vseh tistih vhodov x_r , za katere velja $y_0 = f(x_r)$. Ker je $f(x)$ periodična funkcija s periodo r , lahko to superpozicijo vhodnega registra zapišemo kot:

$$\frac{1}{B} \sum_{b=0}^B |x_0 + b \cdot r\rangle,$$

kjer je b celo število, ki teče od 0 do B , pri tem pa je B maksimalno celo število, pri katerem $x_0 + B \cdot r$ še ne preseže velikosti vhodnega registra Q .

5. Inverzna kvantna Fourierova transformacija vhodnega registra:

- Vhodni kvantni register transformiramo z inverzno kvantno Fourierovo transformacijo, ki splošno superpozicijo registra $\sum_{x=0}^{Q-1} \alpha_x |x\rangle$ spremeni v novo

superpozicijo stanj $\frac{1}{\sqrt{Q}} \sum_{z=0}^{Q-1} \sum_{x=0}^{Q-1} \alpha_x e^{\frac{-i2\pi zx}{Q}} |z\rangle$ (poglavlje 4.4). Po tej operaciji je v našem primeru vhodni kvantni register torej v stanju

$$\frac{1}{\sqrt{Q}} \sum_{z=0}^{Q-1} \sum_{b=0}^B e^{\frac{-i2\pi z(x_0+b \cdot r)}{Q}} |z\rangle,$$

saj so bila prej v vhodnem registru samo števila $x = x_0 + b \cdot r$ (vsa ostala so imela amplitudo verjetnosti $\alpha_x = 0$).

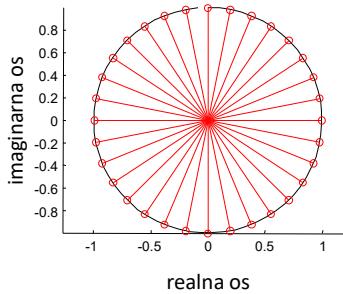
- Izhodni register še vedno vsebuje eno samo vrednost $y_0 = f(x_0)$.

6. Meritev vhodnega registra:

- Izmerimo vhodni register. Velja

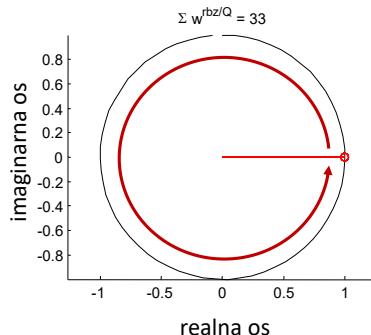
$$\frac{1}{\sqrt{Q}} \sum_{z=0}^{Q-1} \sum_{b=0}^B e^{\frac{-i2\pi z(x_0+b \cdot r)}{Q}} |z\rangle = \frac{1}{\sqrt{Q}} \sum_{z=0}^{Q-1} e^{\frac{-i2\pi zx_0}{Q}} \sum_{b=0}^B e^{-i2\pi b \frac{zr}{Q}} |z\rangle$$

Amplitude verjetnosti vseh tistih števil z , za katera velja, da $\frac{zr}{Q}$ ni blizu pozitivnemu celemu številu, bodo v vsoti po b tvorile 2D enotske vektorje vseh možnih orientacij (slika 5.13), zato se bodo v vsoti po b izničile in bo njihova vsota enaka ali vsaj blizu 0 (zaradi končnosti vsote po b , ki izvira iz končnosti vhodnega kvantnega registra, ni rečeno, da bo čisto enaka 0).



Slika 5.13: Destruktivna superpozicija amplitud verjetnosti: amplitude verjetnosti paroma kažejo v diametalno nasprotne strani in se med sabo odštejejo, zato je njihova vsota enaka 0. Posledično je tudi verjetnost meritve takšnega števila z enaka 0.

Amplitude verjetnosti vseh tistih števil z , za katera velja, da je $\frac{zr}{Q}$ zelo blizu pozitivnemu celiemu številu (idealno $\frac{zr}{Q} = c$), pa bodo v vsoti po b tvorila konstruktivno superpozicijo, zato se bo njihova verjetnost precej ojačila (slika 5.14)



Slika 5.14: Konstruktivna superpozicija amplitud verjetnosti: vse amplitude verjetnosti kažejo v isto smer in se seštevajo. Posledično je tudi verjetnost meritve takšnega števila z relativno velika in je vedno večja od 0.

Torej je veliko verjetneje, da bomo ob meritvi v vhodnem registru izmerili takšno število z_0 , da bo veljalo $\frac{z_0 r}{Q} = c$, kjer je c celo število.

7. Ocenitev periode r :

- Z veliko verjetnostjo torej velja $\frac{z_0}{Q} = \frac{c}{r}$, in ker mora biti perioda r manjša od N , velja tudi $r < N$. Pri tem sta c in r celi števili.
- S pomočjo verižnih ulomkov najdemo takšen približek $\frac{c}{r} \approx \frac{z_0}{Q}$, da velja $r < N$. Običajno dobimo več kandidatov za r in preveriti moramo, kateri med njimi izpoljuje pogoj $f(x) = f(x + r)$.
- Če nismo uspešni, izberemo nov a in ponovimo celoten kvantni del Shorovega algoritma.

Zgled 5.8: Faktorizirajmo število $N = 437$. Izberimo $a = 3$ in s pomočjo kvantnega dela Shorovega algoritma izračunajmo vrednosti funkcije $f(x) = 3^x \bmod 437$. Ker je $N = 437$, potrebujemo 18-bitni vhodni kvantni register ($Q = 262144$) in 9-bitni izhodni kvantni register. Skonstruiramo vezje s slike 5.12 in po meritvi izhodnega registra (korak 4 kvantnega dela algoritma) dobimo vrednost 101. Po meritvi vhodnega registra (korak 6 kvantnega dela algoritma) dobimo vrednost $z_0 = 1322$.

S pomočjo verižnih ulomkov izračunamo naslednje približke $\frac{c}{r}$ ulomka $\frac{z_0}{Q} = \frac{1322}{262144}$, pri čemer mora biti $r < 437$ (korak 7 kvantnega dela algoritma):

$$\frac{c}{r} = \frac{1}{198}, \quad \frac{c}{r} = \frac{3}{595}, \quad \frac{c}{r} = \frac{7}{1388} \text{ in } \frac{c}{r} = \frac{17}{3371}.$$

Pogoj $r < 437$ izpolnjuje samo prvi približek

$$\frac{c}{r} = \frac{1}{198},$$

zato preizkusimo periodo $r = 198$:

$$\begin{aligned} f(1) &= 3^1 \bmod 437 = 3, \\ f(1 + 198) &= 3^{199} \bmod 437 = 3, \\ f(1 + 2 \cdot 198) &= 3^{397} \bmod 437 = 3. \end{aligned}$$

Perioda je soda in $a^{r/2} \bmod N = 3^{99} \bmod 437 = 208 \neq -1$, torej perioda $r = 198$ izpolnjuje pogoja v 5. in 6. koraku klasičnega dela Shorovega algoritma. Izračunamo $p = \gcd(3^{99} - 1, 437) = 19$. Drugi faktor je $q = 437/19 = 23$.

Shorov algoritem ima časovno zahtevnost $\mathcal{O}((\log N)^3)$. Kvantni del je sestavljen iz kvante Fourierove transformacije in implementacije funkcije $f(x) = a^x \bmod N$. V poglavju 4.4 smo spoznali, da lahko kvantno Fourierovo transformacijo učinkovito izvedemo s pomočjo Hadamardovih in kontroliranih faznih vrat. Implementacija kvantnega vezja za funkcijo $f(x) = a^x \bmod N$ je precej bolj kompleksna in zahteva tudi več kvantnih vrat kot kvantna Fourierova transformacija. Funkcija $f(x)$ je tudi kritičen del Shorovega algoritma, saj potrebujemo specifično vezje za vsako izbrano osnovo a . Pri njeni implementaciji si lahko pomagamo s spoznanjem, da je v kvantnem registru velikosti N vsota po modulu 2^N ena izmed najbolj splošnih unitarnih operacij (operacija XOR je vsota po modulu 2). Shor je za implementacijo funkcije $f(x) = a^x \bmod N$ uporabil algoritem zaporednega kvadriranja, ki je podrobnejše opisan v [Sho1997].

Na koncu poudarimo, da je Shorov algoritem nedeterminističen (verjetnostni) [Wis2015, Aar2013, Bro2014, Eke2008, Mon2016, Nie2000, Sho1997]. Ne najde namreč vedno netrivialnega faktorja števila N (trivialna faktorja števila 21 sta 1 in 21, 7 in 3 pa sta netrivialna faktorja).

Zgled 5.9: Faktorizirajmo število $N = 15$. Izberimo $a = 14$ in izračunajmo vrednosti funkcije $f(x) = 14^x \bmod 15$. Potem se bodo v izhodnem registru vrstila naslednja zaporedja funkcije $f(x)$:

$$1, 14, 1, 14, 1, 14 \dots$$

Perioda je enaka $r = 2$, torej sta edina faktorja števila 15, ki ju vrne Shorov algoritmom, $\gcd(14 - 1, 15) = 1$, in $\gcd(14 + 1, 15) = 15$, torej trivialna faktorja števila 15. V tem primeru nas korak 6 kvantnega dela algoritma usmeri k ponovni izberi osnove a .

Naloge:

1. V poljubnem programskem jeziku, ki podpira matrični račun, implementirajte Deutschov algoritmom.
2. Zakaj je pomembno, da je spodnji bit na diagramu Deutshovega algoritma, ki ga podaja slika 5.6, v superpoziciji stanj $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$? Ali bi algoritmom deloval pravilno, če bi bil spodnji bit v superpoziciji stanj $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$?
3. Izračunajte matriki \mathbf{U}_f in \mathbf{U}_ω za Groverjev algoritmom, ki poišče peti element v podatkovni bazi z osmimi elementi.
4. Izračunajte matriki \mathbf{U}_f in \mathbf{U}_ω za Groverjev algoritmom, ki poišče osmi element v podatkovni bazi s šestnajstimi elementi.
5. V poljubnem programskem jeziku, ki podpira matrični račun, implementirajte Groverjev algoritmom, ki poišče osmi element v podatkovni bazi z 32 elementi. Koliko iteracij 2. in 3. koraka potrebuje algoritmom?
6. Razmislite o definiciji operatorjev \mathbf{U}_f in \mathbf{U}_ω v Groverjevem algoritmu in o učinkovitosti implementacije Groverjevega algoritma na klasičnem računalniku (Turingovem stroju). Kako hitro rasteta dimenziji matrik \mathbf{U}_f in \mathbf{U}_ω z velikostjo podatkovne baze in kje črpa Groverjev algoritmom svojo moč na kvantnem računalniku?
7. Določite periodo funkcije $f(x) = 3^x \bmod 15$.
8. Določite periodo funkcije $f(x) = 7^x \bmod 21$.
9. S Shorovim algoritmom v poljubnem simulatorju kvantnega računalnika (npr. JQuantum [Vri2010], Quantum playground [Wro2014] ali LIQUi|> [Mic2016]) faktorizirajte število $N=187$.

10. S Shorovim algoritmom v poljubnem simulatorju kvantnega računalnika faktorizirajte število $N=357$.
11. S Shorovim algoritmom v poljubnem simulatorju kvantnega računalnika faktorizirajte število $N=289$.
12. Ocenite, kolikokrat moramo v povprečju pri Shorovem algoritmu izbrati novo osnovo a in ponoviti kvantni del algoritma. Kako je povprečno število ponovitev odvisno od velikosti števila N , ki ga faktoriziramo?
13. Razmislite o teoremu neizvedljivosti kloniranja in njegovem vplivu na Shorov algoritmom. Kako bi se Shorov algoritem poenostavil, če bi lahko vhodni register po meritvi izhodnega registra klonirali? Bi v tem primeru še vedno potrebovali inverzno kvantno Fourierovo transformacijo?
14. Kako velik vhodni kvanti register bi potrebovali za implementacijo Shorovega algoritma, ki bi faktoriziral 1024-bitno število N ?

6 Kriptografija in kvantno dešifriranje

Kvantni algoritmi so širšo pozornost pritegnili predvsem po spoznanju, da predstavljajo resno grožnjo sodobnim kriptografskim sistemom, s katerimi so zaščitene vse občutljive informacije današnje informacijske družbe, od bančnih transakcij do državnih in poslovnih skrivnosti [Wis2015, Aar2013, Bro2014, Eke2008, Mon2016, Nie2000, Sho1997]. Do nedavnega je vsa kriptografija temeljila na simetričnih in asimetričnih šifrirnih algoritmih. Simetrični šifrirni algoritmi uporabljajo enak skriti ključ za šifriranje in dešifriranje vsebine, asimetrični pa uporabljajo dva ključa, in sicer javni ključ za šifriranje ter privatni (skriti) ključ za dešifriranje vsebine. Shorov algoritem je pokazal, da lahko na dovolj velikem kvantnem računalniku v polinomskem času faktoriziramo cela števila in s tem pridobimo privatni ključ asimetričnih šifrirnih algoritmov.

Na drugi strani je Grover zasnoval kvantni algoritem, ki omogoča zelo hitro nestrukturirano iskanje. Groverjev algoritem sicer ne omogoča tako velike pohitritev iskanja skritega ključa kot Shorov algoritem, vendar je veliko splošnejši. Groverjevo iskanje bi lahko tako uporabili tudi za iskanje skritega ključa simetričnih šifrirnih algoritmov. Ker pa je Groverjev algoritem optimalen (hitrejši iskalni kvantni algoritem ne obstaja [Zal1999]) in je njegova časovna zahtevnost velikostnega reda $\mathcal{O}(\sqrt{Q})$, kjer je Q bitna velikost skritega ključa, zadostuje, da bitno velikost skritih ključev sodobnih simetričnih šifrirnih algoritmov podvojimo. Simetrično šifriranje je torej relativno težko razkriti z zanimi kvantnimi algoritmi. Njegova največja slabost je, da mora biti skriti ključ poznan obema komunicirajočima strankama, torej pošiljalj in prejemniku. Ker je treba skriti ključ redno posodabljati, je simetrično šifriranje precej odvisno od postopkov varne izmenjave skritega ključa. Slednji so do nedavnega temeljili na asimetričnem šifriranju. Z objavo Shorovega algoritma pa je postala ogrožena tudi varna izmenjava skritega ključa simetričnih šifrirnih algoritmov.

Zato so vlade razvitih držav pričele izdatno financirati raziskovanja kvantnih računalnikov in kvantnih algoritmov. Močno so se razvila tudi kvantna omrežja, ki omogočajo varno izmenjavo informacij in skritih ključev, pospešeno pa se razvijajo pa so tudi novi kodirniki, ki jih tudi kvantni računalniki ne morejo hitro dešifrirati in so trenutno širši javnosti znani kot postkvantna kriptografija (angl. *post-quantum cryptography*) [Ber2009]. Ocenjuje se, da bodo asimetrični šifrirni algoritmi opuščeni med 2020 in 2030, novi standardi kriptografije, ki naj bi jih nadomestili, pa se še intenzivno razvijajo.

V tem poglavju najprej podajamo kratek pregled asimetričnih in simetričnih šifrirnih algoritmov in opišemo njihovo izpostavljenost kvantnemu dešifriranju. V nadaljevanju opišemo algoritme za kvantno izmenjavo ključev, ki omogočajo, da s poljubno verjetnostjo zaznamo prisluškovanje.

Trenutno so v uporabi trije principi asimetričnega šifriranja [Sti2002, Ber2009]:

- faktorizacija celih števil (angl. *integer factorization*),
- diskretni logaritmi (angl. *discrete logarithm*) in
- eliptične krivulje (angl. *elliptic curve*).

6.1 Asimetrično šifriranje s faktorizacijo celih števil

Asimetrično šifriranje s faktorizacijo celih števil uporablja privatni in javni ključ [Sti2002]. Javni ključ je poznan vsem in služi za šifriranje podatkov, ki jih lahko dešifriramo samo s skritim zasebnim ključem. Običajno lahko javni ključ uporabnika A uporablja katerakoli stranka (seveda samo za šifriranje podatkov, ki so namenjeni uporabniku A). Uporabnik A lahko podatke dešifrira samo s pomočjo svojega zasebnega ključa. Javni in zasebni ključ vedno nastopata v parih, tako kot ključavnica in njen ključ.

Faktorizacija celih števil temelji na predpostavki, da je za dano veliko celo število $n > 0$ težko najti praštevili $p > 1$ in $q > 1$ tako, da velja $n = pq$, četudi vemo, da je mogoče število n zapisati kot produkt dveh praštevil. Število praštevil, ki so manjša od x , je namreč sorazmerno številu $x/\log(x)$. Torej, če je število n zgrajeno kot produkt dveh, recimo, b -mestnih praštevil (zapisano v desetiškem številskem sistemu), je računska zahtevnost za njegovo faktorizacijo z metodo zaporednih poskusov velikostnega reda 10^b . Že za zmerno velikost $b = 40$ je računski napor te velikosti izven dosega sodobnih klasičnih računalnikov [Sti2002].

Obstajajo algoritmi faktorizacije, ki so veliko hitrejši od metode zaporednih poskusov, in zahtevana bitna dolžina modulov $p > 1$ in $q > 1$ raste veliko hitreje kot linearna funkcija želene stopnje varnosti. Metoda z eliptičnimi krivuljami (angl. *elliptic curve method – ECM*) [Sti2002], tretja najhitrejša znana metoda za faktorizacijo celih števil na klasičnih računalnikih, najde manjši faktor p števila n z računskim naporom $\mathcal{O}(e^{(1 + o(1))\sqrt{(\ln(p) \ln(\ln(p)))}})$, torej pol poti med eksponentnim in polinomskim časom. Primerna je zlasti za manjše faktorje p , približno do velikosti 80 števk.

Algoritem številskih sit (angl. *number field sieve – NFS*) [Len1993], trenutno najhitrejša znana metoda za faktorizacijo celih števil na klasičnih računalnikih, potrebuje v povprečju $\mathcal{O}(e^{(b \cdot 64/9)^{1/3} \cdot \log(b)^{2/3}})$ operacij za faktorizacijo b -bitnega celega števila n , torej $2/3$ poti med eksponentnim in polinomskim časom. Leta 2004 je ocenjeni denarno-časovni vložek, ki je potreben za napad na 1024-bitni skriti ključ algoritma RSA z uporabo namenske strojne opreme in algoritma NFS znašal največ 400 M\$-dni. Januarja 2010 je bila oznanjena

faktorizacija ključa RSA-768. V napadu v dolžini več kot dveh let je sodelovalo več sto računalnikov.

Pri vrednotenju kriptografske varnosti je treba upoštevati tudi **dvojni Moorov zakon kriptoanalyze** [Sti2002], ki pravi, da se strošek napada na katerikoli šifrirni sistem zmanjša za faktor 2 vsakih 9 mesecev, in sicer za faktor 2 vsakih 18 mesecev zaradi napredka kriptoanalyze in za faktor 2 vsakih 18 mesecev zaradi napredka strojne opreme.

Pri algoritmu RSA, najpogostejšem asimetričnem šifrirnem algoritmu, vsebuje uporabnikov javni ključ kot celo število n , njemu ustrezan zasebni ključ pa vsebuje faktorja števila n , torej praštevili p in q . Število n je za vsakega uporabnika unikatno. Glede na zgoraj navedeni dvojni Moorov zakon je treba dolžine ključev pri algoritmih RSA konstanto podaljševati [Gir2015]:

- leta 2010 je minimalno zahtevano varnost zagotavljal ključ z dolžino 1112 bitov,
- leta 2020 bo dolžina ključa, ki bo zagotavlala minimalno zahtevano kriptografsko varnost, narasla na 1387 bitov.

6.1.1 RSA

RSA je prvi algoritem, ki je praktično primeren tako za podpisovanje kot za šifriranje dokumentov in ena prvih demonstracij prednosti šifriranja z javnim ključem. RSA se na široko uporablja v protokolih, ki so namenjeni komercialnim komunikacijam, in velja za varen način šifriranja, če je le dolžina ključev dovolj velika. Zaradi računske kompleksnosti se algoritom uporablja večinoma le za izmenjavo simetričnih ključev, nato pa se podatki šifrirajo z veliko hitrejšimi simetričnimi šifrirnimi algoritmi.

Algoritom so leta 1977 na tehnološkem inštitutu *Massachusetts Institute of Technology – MIT* razvili Ron Rivest, Adi Shamir in Len Adleman [Riv1978]. Črke RSA so začetnice njihovih priimkov. Clifford Cocks, britanski matematik, zaposlen pri britanski vladni službi *Government Communications Headquarters*, je leta 1973 v notranjem dokumentu predstavil ekvivalenten sistem [Sti2002]. Predstavljeni sistem bi potreboval relativno drage računalniške komponente in algoritom se ni razširil. Leta 1983 je MIT algoritom patentiral pod patentno številko U.S. Patent 4,4050,829. Patentne pravice so prenehale veljati 21. 9. 2000.

Tvorba javnega in zasebnega ključa

1. Izbereta se dve praštevili, p in q , tako da velja $p \neq q$. Izbereta se naključno, morata pa biti dovolj veliki in neodvisni drugo od drugega. Učinkovit postopek iskanja praštevil lahko zasnujemo s pomočjo testa praštevilskosti, na primer s pomočjo Miller-Rabinovega testa praštevilskosti, ki smo ga predstavili v poglavju 1.

2. Izračuna se sestavljeni število $n = p \cdot q$.
3. Izračuna se vrednost Eulerjeve funkcije $\varphi(n)$ pri argumentu n : $\varphi(n) = (p - 1)(q - 1)$.
4. Izbere se celo število e , tako da velja $1 < e < \varphi(n)$ in sta e in $\varphi(n)$ tuji števili (nimata skupnega delitelja razen števila 1).
5. Izračuna se d , tako da je ostanek produkta $d \cdot e$ po modulu $\varphi(n)$ enak $1: (d \cdot e) \text{ mod } \varphi(n) = 1$ oziroma je $d \cdot e = 1 + k \cdot \varphi(n)$ za poljubno pozitivno celo število k . Pogosto se d izračuna s pomočjo razširjenega Evklidovega algoritma (angl. *extended Euclidean algorithm*) [Par1998].

Javni ključ je sestavljen iz generatorja n in šifrirnega eksponenta e .

Zasebni ključ je sestavljen iz generatorja n , ki je javen, in zasebnega (dešifrirnega) eksponenta d , ki mora ostati skriven. Vrednosti p in q sta varnostno občutljivi, ker sta faktorja števila n in omogočata izračun skrivnega števila d če poznamo e . Treba ju je torej varno izbrisati in paziti, da p in q nista dosegljiva zunaj računalniškega sistema, za katerega sta prispevala zasebni ključ. Prav tako, kot je treba izbrisati tudi ostale vrednosti, ki so bile ustvarjene pri računanju para javnega in skrivnega ključa (na primer vrednost Eulerjeve funkcije $\varphi(n)$).

Šifriranje in dešifriranje besedila poteka po postopku potenciranja po modulu n [Riv1978]:

$$c = m^e \text{ mod } n,$$

$$m = c^d \text{ mod } n,$$

kjer je m nešifrirano, c pa šifrirano besedilo. Obe besedili sta predstavljeni kot celi števili, torej je potrebno pred šifriranjem bitno zaporedje besedila ustrezno pretvoriti v celoštevilski zapis, po dešifriranju pa je potrebno celoštevilski zapis pretvoriti nazaj v besedilo.

Zgled 6.1: Privzemimo, da smo generirali naslednji 40-bitni privatni ključ:

praštevilo p :	902333
praštevilo q :	811919
modul n :	732621307027
javni eksponent e :	65537
privatni eksponent d :	12117729505

Šifriranje in dešifriranje poljubnega odseka besedila se izvedeta z računske operacijo

$a \text{ mod } b$, ki vrne ostanek po celoštevilskem deljenju a/b . V našem primeru imamo:

- Šifriranje besedila $m = 65$:

$$c = m^e \text{ mod } n = 65^{65537} \text{ mod } 732621307027 = 301927525300.$$

- Dešifriranje besedila:

$$m = c^d \text{ mod } n = 301927525300^{12117729505} \text{ mod } 732621307027 = 65$$

Zgoraj predstavljena šifriranje in dešifriranje potrebujeta programsko podporo aritmetiki z velikimi števili, saj so v praksi uporabljena števila, ki z več sto biti močno presegajo velikosti strojno podprtih podatkovnih tipov na 32-bitnih in 64-bitnih procesorjih. Na spletu obstaja veliko prosto dostopnih knjižnic (npr. *OpenSSL* [Ope2016], *Crypto++* [Cry2016] in druge), ki na učinkovit način izvajajo računajo z velikimi števili. Kljub temu se RSA po računski učinkovitosti ne more kosati s simetričnimi šifrirnimi algoritmi. Razlog tiči ravno v računski zahtevnosti aritmetike z velikimi števili.

6.1.2 Podpisovanje sporočil

Recimo, da Špela uporabi javni Tinetov ključ in mu pošlje šifrirano sporočilo. V sporočilu lahko trdi, da je Špela, vendar Tine tega ne more preveriti, saj lahko vsakdo uporablja Tinetov javni ključ, da mu pošlje šifrirana sporočila. Da bi preverili poreklo sporočila, lahko za kriptografsko varno podpisovanje sporočila uporabimo algoritmom RSA.

Recimo, da želi Špela poslati podpisano sporočilo Tinetu. Pri tem uporabi svojo zasebni ključ, tako da s sekjalno funkcijo (angl. *hash function*) [Par1998, Sti2002] izračuna sekjalno vrednost (angl. *hash code*) h celotnega sporočila, jo zakodira s svojim privavnim ključem $h' = h^d \text{ mod } n$ (kot ob dešifriranju sporočila) in jo kot podpis pripne k sporočilu. Ko Tine prejme podpisano sporočilo, uporabi Špelin javni ključ in dešifrira pripeto sekjalno vrednost $h = (h')^e \text{ mod } n$. Dešifrirano kodo primerja s sekjalno vrednostjo prejetega sporočila (brez podpisa) in primerja obe vrednosti. Če se ujemata, Tine ve, da ima avtor sporočila v lasti Špelin zasebni ključ in da sporočilo od podpisa dalje ni bilo spremenjeno.

6.1.3 Računska učinkovitost

Generiranje ključev se izvede samo občasno, zato je njegova računska učinkovitost manj pomembna. Kodiranje uporablja modularno potenciranje $y = x^e \text{ mod } n$, za katero obstaja več algoritmov za učinkovito izvajanje. Eden izmed njih je binarna metoda od-leve-proti-desni (angl. *right-to-left binary method*) [Sch1996b]. Algoritom temelji na enakosti

$$(a \cdot b) \text{ mod } n = ((a \text{ mod } n) \cdot (b \text{ mod } n)) \text{ mod } n.$$

Za rešitev $y = x^e \bmod n$ je treba e zapisati v binarni notaciji:

$$e = e_{k-1}e_{k-2}\dots e_1e_0,$$

kjer je e_{k-1} najpomembnejši od nič različni bit in e_0 najmanj pomemben bit.

Zapišimo psevdokod algoritma od-leve-proti-desni [Sch1996b, Par1998, Sti2002]:

Vhodi: število x , modul p in eksponent e , ki je zapisan s k biti
 $(e = e_{(k-1)}e_{(k-2)}\dots e_{(1)}e_{(0)})$
Izhod: $y=x^e \bmod p$

```

y := 1
for bit j = from k downto 0
begin
    y := y * y mod n /* kvadriraj */
    if e(j) == 1 then y = y * x mod n /* zmnoži */
end
return y

```

Čas za izvedbo modularnega potenciranja se povečuje s številom bitov, ki so v eksponentu e postavljeni na 1. Pri šifriranju lahko s primerno izbiro e zmanjšamo računsko zahtevnost za $c = m^e \bmod n$ [Sch1996b, Par1998, Sti2002]. Priljubljene izbire za e so 3, 17 in 65537, saj so vsa ta praštevila določena le z dvema neničelnima bitoma: 3 = 0011 (binarno), 17 = 10001, 65537 = 100000000000000001.

Bitov v privatnem eksponentu d ne moremo tako ugodno nadzorovati, zato navadno dešifriranje traja dlje kot šifriranje [Sch1996b, Par1998, Sti2002]. Zaradi varnostnih razlogov d tudi ne sme biti majhno število. Računska zahtevnost šifriranja in dešifriranja z algoritmom RSA je torej odvisna od izvedbe modularnega potenciranja in od velikosti eksponenta (e za šifriranje in d za dešifriranje). Navadno so stroški šifriranja z asimetričnimi algoritmi neprimerljivo višji od stroškov kodiranja s simetričnimi algoritmi. Ugodnost, da javnega ključa ni treba skrivati in ščititi, je torej plačana z nižjo pretočnostjo asimetričnih šifrirnih algoritmov. To je tudi razlog, zakaj je algoritmom RSA uporabljen predvsem za varno izmenjavo skritega ključa simetričnih šifrirnih algoritmov [Sch1996b, Par1998, Sti2002].

6.2 Diskretni logaritem

Osnovni koncept diskretnih logaritmov je podan z naslednjim opisom matematičnega problema: za dani element h končne ciklične multiplikativne grupe $G_n = \{0, 1, 2, \dots, n-1\}$ z n elementi, ki jo po modulu n napenja multiplikativni generator g , najdi celo število k , tako da velja $h = g^k \text{ mod } n$. Najmanjši nenegativni k se imenuje diskretni logaritem (angl. *discrete logarithm*) števila h glede na osnovo g in je označen s $k = \log_g(h)$. Katerikoli celi števili k_1 in k_2 , za kateri velja $g^{k_1} \text{ mod } n = g^{k_2} \text{ mod } n$, sta, po definiciji, kongruenčni po modulu n (imata enak ostanek po deljenju z n) in pripadata istemu kongruenčnemu razredu (angl. *congruence class*) po modulu n [Par1998, Sti2002].

Diskretni logaritem je verjetno najlažje razumeti v povezavi z grupo $(\mathbf{Z}_p)^\times$, ki združuje vsa števila, kongruenčna glede na množenje po modulu p [Par1998, Sti2002]. Navadno je p praštevilo, k -to potenco poljubnega števila g , ki pripada grupi $(\mathbf{Z}_p)^\times$, pa izračunamo kot $g^k \text{ mod } p$. Ta računski proces se imenuje diskretno potenciranje. Za zgled vzemimo $g = 7$, ki pripada $(\mathbf{Z}_{11})^\times$, in izračunamo $7^4 \text{ mod } 11 = 3$. Torej je $k = 3$ v grupi $(\mathbf{Z}_{11})^\times$.

Diskretni logaritem je inverzna operacija diskretnemu potencirанию [Par1998, Sti2002]. Na primer, poiščimo rešitev enačbe $7^k \text{ mod } 11 = 3$. Kot prikazuje zgornji primer, je ena izmed rešitev $k = 4$. Toda to ni edina rešitev. Ker je $7^{10} \text{ mod } 11 = 1$, velja $5^{4+10n} \text{ mod } 11 = 3$. Torej ima enačba neskončno mnogo rešitev, njihova splošna oblika pa je $k = 4 + 10n$. Ker je $m = 10$ najmanjše pozitivno celo število, ki zadosti enačbi $3^m \text{ mod } 11 = 1$, so rešitve oblike $k = 4 + 10n$ edine rešitve diskretnega logaritma $7^k \text{ mod } 11 = 3$ v $(\mathbf{Z}_{11})^\times$. Pravimo, da je 10 red števila 7 v grupi $(\mathbf{Z}_{11})^\times$.

Algoritmi, ki temeljijo na diskretnem logaritmu, se najpogosteje uporabljajo v hibridnih sistemih, kot je asimetrični kodirniki za varno izmenjavo kriptografskih ključev (glejte opis Diffie-Hellmanovega algoritma v podoglavlju 6.2.1). Po izmenjavi kriptografskih ključev se, kot pri algoritmu RSA, zaradi večje učinkovitosti za nadaljnjo komunikacijo uporabljajo simetrični šifrirni algoritmi.

Trenutno ni znan noben klasični algoritem za izračun splošnega diskretnega logaritma $k = \log_g(h)$ [Par1998, Sti2002]. Naivna metoda zaporednih poskusov temelji na postopnem dvigovanju potenc k pri osnovi g , dokler ne najdemo želenega h . Računski čas te metode je sorazmeren velikosti grupe G in raste eksponentno s številom števk s katerimi zapišemo velikost grupe G . Obstajajo bolj sofisticirani klasični algoritmi, ki se navadno zgledujejo po podobnih algoritmih za faktorizacijo celih števil [Par1998, Sti2002]. Ti algoritmi tečejo hitreje kot metoda zaporednih poskusov, vendar nobeden od njih ne deluje v polinomskem času.

6.2.1 Diffie–Hellmanov algoritem za izmenjavo ključev

Diffie–Hellmanov algoritem za izmenjavo ključev je eden od prvih praktičnih primerov izmenjave ključev na področju kriptografije. Metoda omogoča dvema stranema, ki nimata predhodnih stikov, da vzpostavita skupni privatni ključ skozi javno opri (kriptografsko nevaren) komunikacijski kanal. Ta ključ se lahko nato uporabi za šifriranje sporočili z uporabo simetričnih šifrirnih algoritmov. Postopek izmenjave ključa temelji na diskretnem logaritmu in je bil patentiran s patentom U.S. Patent 4200770, ki je že potekel [Par1998, Sti2002].

Algoritem [Dif1976, Wid2015]:

1. Špela in Tine skupaj izbereta praštevilo p in osnovo g (ti dve vrednosti sta lahko znani vsem).
2. Špela izbere naključno naravno število a in pošlje Tinetu število $g^a \text{ mod } p$.
3. Tine izbere naključno naravno število b in pošlje Špeli število $g^b \text{ mod } p$.
4. Špela izračuna $k = (g^b)^a \text{ mod } p$.
5. Tine izračuna $k = (g^a)^b \text{ mod } p$.

Samo a, b in $g^{ab} \text{ mod } p = g^{ba} \text{ mod } p$ morajo ostati skriti. Vse ostale računske entitete p , g , $g^a \text{ mod } p$ in $g^b \text{ mod } p$ so javno dostopne. Ko Špela in Tine izračunata skupni skrivni ključ, ga lahko uporabita kot ključ za simetrični šifrirni algoritem. Seveda so za zagotovitev varnosti potrebne velike vrednosti a, b in p , saj je, na primer, preprosto preizkusiti vse možne vrednosti $g^{ab} \text{ mod } 23$ (obstaja največ 22 takih vrednosti, četudi sta a in b velika). Če pa je p vsaj 300-mestno število ter a in b najmanj 100-mestni števili, celo najboljši klasični algoritmi, ki jih poznamo danes, ne morejo najti skrivnega ključa zgolj iz $g, p, g^a \text{ mod } p$ in $g^b \text{ mod } p$. Ni treba, da je osnova g velika, zato se v praksi izbereta vrednosti 2 ali 5 [Par1998, Sti2002, Wid2015].

Zgled 6.2: Špela in Tine si izbereta praštevilo $p = 2221$ in osnovo $g = 2$. Obe števili sta lahko javno znani.

Špela naključno izbere število $a = 6$ in pošlje Tinetu $2^6 \text{ mod } 2221 = 64$.

Tine naključno izbere število $b = 15$ in pošlje Špeli $2^{15} \text{ mod } 2221 = 1674$.

Špela izračuna $k = 1674^6 \text{ mod } 2221 = (2^{15})^6 \text{ mod } 2221 = 1737$.

Tine izračuna $k = 64^{15} \text{ mod } 2221 = (2^6)^{15} \text{ mod } 2221 = 1737$.

Skupna skrivnost (skriti ključ) je torej 1737.

Diffie–Hellmanov algoritem velja za varnega, če sta le p in g pravilno izbrana. Modul p mora biti praštevilo ali imeti veliko praštevilo kot svoj faktor, da se prepreči uporaba Pohlig–Hellmanovega algoritma za razkritje a ali b [Poh1978, Wid2015]. Navadno za tvorjenje varnega praštevila p uporabimo posebno praštevilo q , poimenovano po francoski matematičarki Sophie Germain, za katerega velja, da je tudi $p = 2q + 1$ praštevilo [Wid2015]. Učinkovit algoritem za rešitev problema diskretnega logaritma, kot na primer kvantni algoritem Petra Shora, omogoča enostaven izračun števil a ali b in resno ogroža varnost izmenjave ključa.

Originalni Diffie–Hellmanov algoritem ne zagotavlja pristnosti strank in je tako izpostavljen napadu osebe na sredini (angl. *man-in-the-middle attack*). Oseba na sredini lahko določi dva različna ključa v Diffie–Hellmanovi izmenjavi, enega s Špelo in drugega s Tinetom in se tako Tinetu učinkovito predstavi kot Špela in obratno, kar omogoča napadalcu dešifriranje (in branje ali shranjevanje) in nato ponovno šifriranje sporočil med Špelo in Tinetom. Za preprečevanje tovrstnega napada je treba v komunikacijski protokol vključiti še metodo za zagotavljanje pristnosti komunicirajočih strank [Par1998, Sti2002, Wid2015].

Šifriranje z Diffie–Hellmanovim algoritmov zahteva dve potenciranji z velikimi potencami, vendar pa sta ti dve potenciranji neodvisni od sporočila in ju je mogoče v številnih primerih izračunati vnaprej. Dešifriranje zahteva eno samo potenciranje. Vseeno se zaradi časovne zahtevnosti potenciranj algoritmi z diskretnim logaritmom v praksi običajno uporabljajo le za izmenjavo skritih ključev simetričnih kodirnikov [Wid2015].

6.3 Eliptične krivulje

Poseben primer diskretnih logaritmov so eliptične krivulje. Njihov razvoj je bil spodbujen z velikostjo ključev v asimetričnih šifrirnih algoritmih. Zaradi nenehnega napredka strojne opreme in kriptografskega znanja se velikost ključev šifrirnih algoritmov, ki temeljijo na faktorizaciji celih števil, in diskretnih logaritmов nenehno povečuje, kar zmanjšuje njihovo uporabnost na platformah z omejeno računsko in pomnilniško zmogljivostjo. Prostorsko učinkovitejšo alternativo predstavlja kriptografija, ki temelji na diskretnem logaritmu eliptičnih krivulj (angl. *elliptic curve discrete logarithm*) [Par1998, Sti2002, WiC2015].

Eliptična krivulja je v splošnem definirana z enačbo [Par1998, Sti2002, WiC2015]:

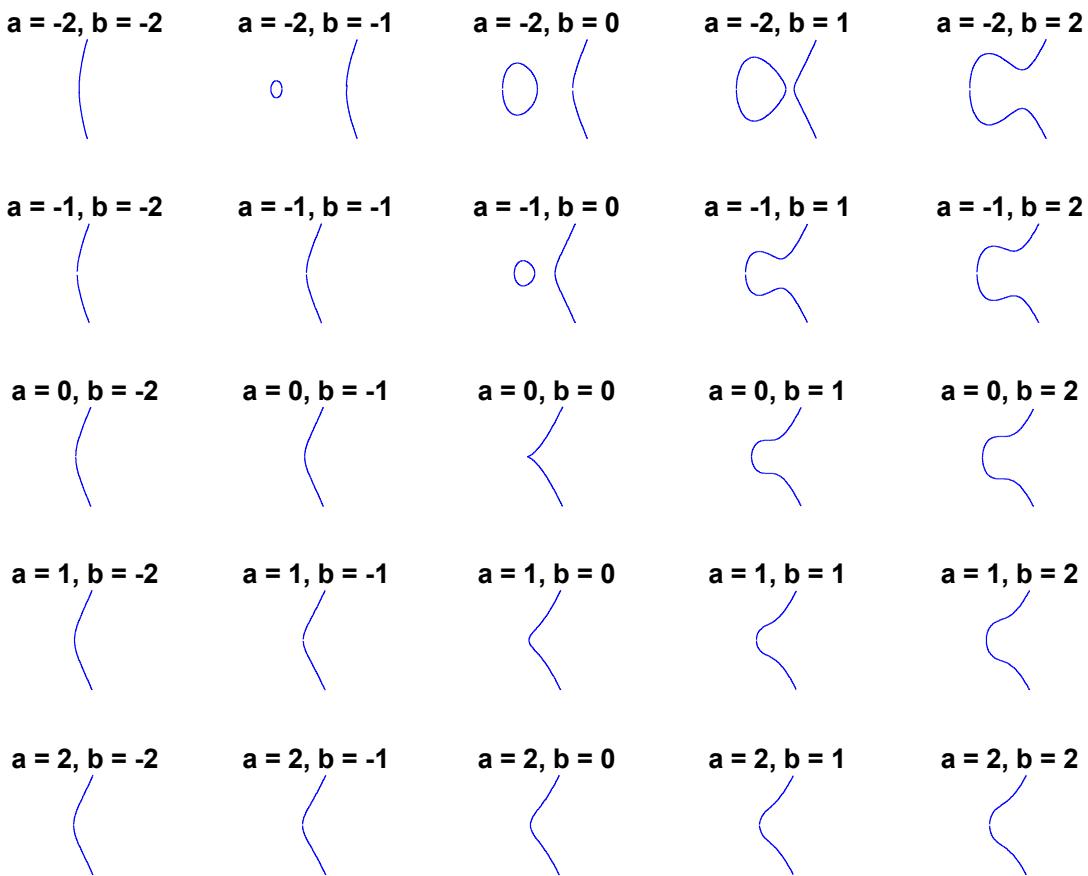
$$y^2 = x^3 + ax + b, \quad (6.1)$$

kjer mora zaradi varnostnih razlogov veljati $4a^3 + 27b^2 \neq 0$. V nasprotnem primeru obstajajo na krivulji singularnosti, npr. presečišča krivulje same s sabo ali pa izolirane točke. Slika 6.1 prikazuje primere različnih eliptičnih krivulj.

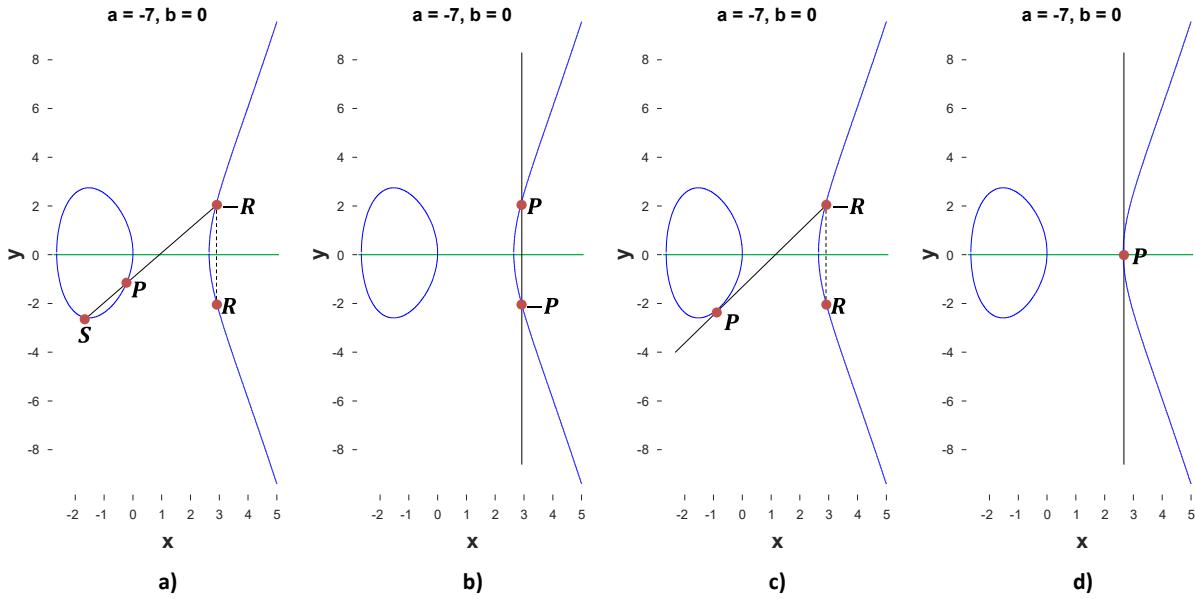
Navadno krivuljo omejimo na končni obseg (angl. *finite field*) $G(p^m) = \mathbb{Z} \text{ mod } p$, torej obseg vseh celih števil po modulu p , kjer je p veliko praštevilo. V prostoru $G(p^m)$ nato vpeljemo seštevanje, ki je definirano kot seštevanje dveh točk S in P na izbrani eliptični

krivulji in ga pojasnjujemo v nadaljevanju in prikazujemo na sliki 6.2. Rezultat seštevanja je točka, ki je zopet na krivulji (slika 6.2). Definirati moramo še element $\mathbf{0}$, ki ne izpolnjuje enačbe krivulje (6.1), a po definiciji še vedno leži na eliptični krivulji. Element $\mathbf{0}$ predstavlja identiteto seštevanja, postavimo pa ga v neskončnost (slika 6.2). Pod omenjenimi pogoji ima pravilo seštevanja vse lastnosti navadnega seštevanja števil in tvori Abelovo grupo [Koš2009].

Slika 6.2 prikazuje seštevanje točk \mathbf{S} in \mathbf{P} na eliptični krivulji. Seštevek obeh točk $\mathbf{S} + \mathbf{P} = \mathbf{R}$ lahko enoznačno opišemo s tretjo točko \mathbf{R} , ki je zrcaljenje presečišča krivulje s premico skozi \mathbf{S} in \mathbf{P} preko osi x (primer a) na sliki 6.2). Točko $-\mathbf{P}$ definiramo kot preslikavo točke \mathbf{P} preko osi x (primer b) na sliki 6.2). Če se premica samo dotika krivulje v eni točki, se ta točka šteje dvakrat (primer c) na sliki 6.2: $\mathbf{P} + \mathbf{P} = \mathbf{R}$). Če je premica vzporedna z osjo y, definiramo tretjo točko $\mathbf{0}$ kot točko v neskončnosti (primera b) in d) na sliki 6.2: $\mathbf{P} + (-\mathbf{P}) = \mathbf{0}$). Natanko eden od teh pogojev velja za vsak par točk na eliptični krivulji. Rezultat seštevanja torej vedno leži na izvorni eliptični krivulji [WiC2015].



Slika 6.1: Eliptične krivulje za različne vrednosti parametrov a in b.



Slika 6.2: Grafični prikaz seštevanja točk S in P na eliptični krivulji $y^2 = x^3 - 7x$.

Preko operacije seštevanja točk lahko definiramo tudi množenje točke s skalarjem kot

$$2P = P + P. \quad (6.2)$$

Za vsako točko P na eliptični krivulji v obsegu $G(p^m)$ velja [Par1998, Sti2002, WiC2015]

$$\lim_{k \rightarrow \infty} kP = \mathbf{0}, \quad (6.3)$$

torej za vsako točko obstaja par skalarjev a in b , $b > a$, za katera velja $aP = bP$. Iz tega sledi $cP = \mathbf{0}$, kjer je $c = b - a$. Najmanjši c , ki izpolnjuje ta pogoj, se imenuje red točke P (angl. *order of the point*) [Par1998, Sti2002].

Za zagotovitev varnosti je treba izbrati takšno krivuljo in fiksno točko F na njej, da je red točke F veliko praštevilo. Če je namreč red točke F n -bitno praštevilo, potem je za izračun faktorja k iz dane točke kF potrebnih vsaj $2^{n/2}$ računskih operacij. Ta lastnost naredi eliptične krivulje privlačne za kriptografijo, saj lahko z njihovo uporabo zagotovimo enako stopnjo kriptografske varnosti ključev in podpisov pri precej manjših ključih kot pri algoritmih, ki temeljijo na faktorizaciji celih števil ali na diskretnih algoritmih, ki smo jih predstavili v podoglavlju 6.2.

Izbira parametrov a , b , $G(p^m)$ in c se običajno ne opravi ločeno za vsakega udeleženca v komunikaciji, saj gre za štetje števila točk na krivulji, ki je zamudno in težavno s stališča izračunave. Zaradi tega je več institucij za standarde objavilo varna območja za parametre eliptičnih krivulj [Nis1999].

Zaradi kompleksnejših pravil seštevanja točk potrebujejo šifrirni algoritmi z eliptičnimi krivuljami precej krajše skrite ključe kot algoritmi, ki temeljijo na faktorizaciji števil. Tabela 6.1 povzema primerjavo varnostno ekvivalentnih ključev, ki je objavljena v [Gur2004].

Tabela 6.1: Primerjava velikosti po varnosti primerljivih ključev algoritma RSA in šifriranja z eliptičnimi krivuljami ECC (angl. *elliptic curve cryptography*). Vir: [Gur2004].

Pričakovani čas, potreben za uspešen napad na procesorju z milijon računskih operacij na sekundo (v letih)	Velikost ključa pri algoritmu RSA (v bitih)	Velikost ključa pri algoritmu ECC (v bitih)
10^4	512	106
10^8	768	132
10^{11}	1024	160
10^{20}	2048	210
10^{78}	21000	600

Čeprav so eliptične krivulje predane v javno uporabo, so številni postopki za hitro aritmetiko točk na eliptičnih krivuljah še vedno patentno zaščiteni. Zato je potrebna pri izvedbi šifrirnih algoritmov, ki temeljijo na eliptičnih krivuljah, dobršna mera previdnosti.

6.3.1 Algoritem ECMQV

Algoritem ECMQV [Men1995] je razširitev Diffie–Hellmanovega algoritma na problem eliptičnih krivulj. Njegovo ime je sestavljeno iz angleškega akronima za eliptične krivulje (EC) in začetnic njegovih avtorjev Menezes, Qu in Vanstone. Algoritem je bil objavljen leta 1995.

Algoritem: Špela in Tine se javno dogovorita o eliptični krivulji in o stalni točki \mathbf{F} na tej krivulji. Špela izbere naključno celo število A_s , ki je njen skrivni ključ, in objavi točko na krivulji $\mathbf{A}_j = A_s \mathbf{F}$ kot njen javni ključ. Tine sledi enakemu postopku: izbere skrivno število B_s in objavi svoj javni ključ $\mathbf{B}_j = B_s \mathbf{F}$.

Recimo, da želi Špela poslati sporočilo Tinetu na kriptografsko varen način. V ta namen lahko preprosto izračuna $\mathbf{K} = A_s \mathbf{B}_j$ in rezultat uporabi kot skrivni ključ za konvencionalni simetrični šifrirni algoritem. Tine lahko izračuna isto točko $\mathbf{K} = B_s \mathbf{A}_j$, saj velja:

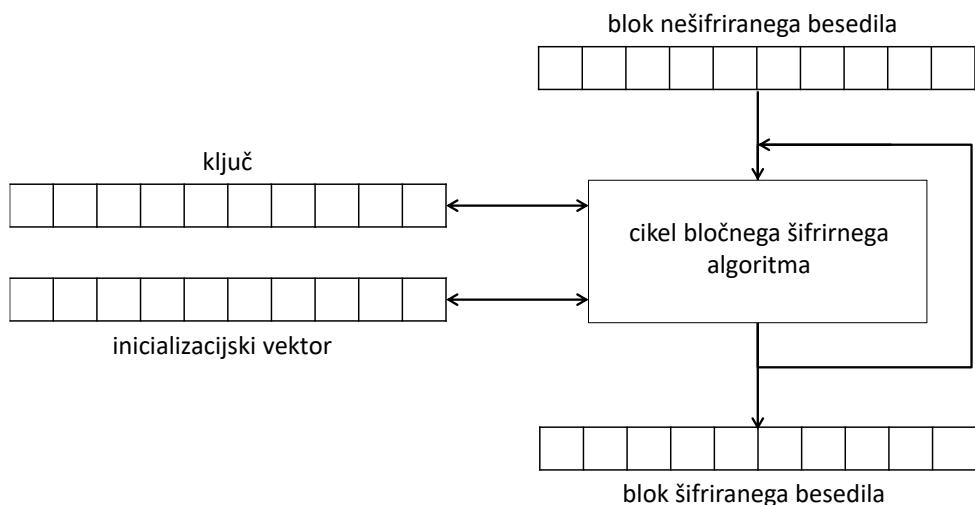
$$B_s \mathbf{A}_j = B_s A_s \mathbf{F} = A_s B_s \mathbf{F} = A_s \mathbf{B}_j. \quad (6.4)$$

Varnost izmenjave temelji na dejstvu, da je težko izračunati faktor k , če sta dani samo točki na krivulji F in kF . Problem izračuna diskretnih logaritmov za eliptične krivulje je namreč NP-težek [Par1998, Sti2002, WiC2015].

Čeprav ima algoritem ECMQV nekaj varnostnih pomenljivosti in je njegova izvedba delno zaščitena s strani patentov kanadskega podjetja Certicom (<http://www.certicom.com>), so njegove izvedenke vključene v praktično vse kriptografske standarde (npr. standarda IEEE P1363 in NSA Suite B).

6.4 Bločni simetrični šifrirni algoritmi

Bločni šifrirni algoritem (angl. *block ciphers*) je algoritem za simetrično šifriranje blokov podatkov (odsekov besedila s konstantno dolžino) v naključno zaporedje znakov iste dolžine (šifrirano besedilo) z uporabo drugega bloka podatkov, ki se imenuje ključ [WiB2015]. V simetričnem načinu kodiranja se uporablja isti ključ za šifriranje in dešifriranje. Ker je dešifriranje brez poznavanja ključa računsko zelo zahtevna operacija, so šifrirani podatki zavarovani pred nezaželenim prisluškovanjem, vsaj dokler je ključ poznan samo pošiljatelju in prejemniku. Bločni šifrirnik je navadno sestavljen iz več ciklov preprostih kriptografskih operacij. Šifrirni ključ se najprej razširi (angl. *key schedule*) na več podključev, podključi pa se nato v več različnih ciklih premešajo in šifrirajo vhodni podatkovni blok (slika 6.3). Navadno je šifriranje realizirano s pomočjo bitne operacije izključujuči ali (angl. *exclusive or* –*XOR*) med blokom nešifriranega besedila in podključem. Zaradi svoje zmogljivosti se bločni šifrirniki pogosto uporabljajo v različnih aplikacijah varne komunikacije, kot so osnovno šifriranje podatkov v internetnih protokolih (IPsec in SSL/TLS) [Par1998, Sti2002, WiB2015], brezžična komunikacija in upravljanje z digitalnimi pravicami.



Slika 6.3: Večplastna zasnova bločnih simetričnih šifrirnih algoritmov. Šifrirni ključ se najprej razširi na več podključev, podključi pa se nato v več različnih ciklih premešajo in se preko operacije XOR združijo z vhodnim podatkovnim blokom.

Ključni korak v posameznem ciklu simetričnih bločnih šifrirnih algoritmov temelji na konceptu zmede in difuzije vhodnih podatkov. Tipični predstavniki omenjenega koraka so mreže za zamenjavo in permutacijo (angl. *substitution-permutation network – SPN*) [Par1998, Sti2002] in mreže Feistel (angl. *Feistel network*) [Par1998, Sti2002], ki jih opisujemo v naslednjih podpoglavljih.

6.4.1 Mreže za zamenjavo in permutacijo

Čeprav se permutacija običajno nadomesti z obrnljivimi linearimi transformacijami, s čemer se izboljša odpornost proti diferencialni in linearni kriptoanalizi, so iz zgodovinskih razlogov mreže za zamenjavo in permutacijo ohranile svoje originalno ime [Par1998, Sti2002]. Med šifriranjem se običajno vhodni podatki v vsakem ciklu premešajo s podključem, nato pa se preslikajo s pomočjo tabele za zamenjavo (angl. *substitution box*) ali krajše S-tabele (slika 6.4). Izhodi S-tabel se dodatno spremenijo z linearno transformacijo, katere namen je širitev (difuzija) statističnih učinkov šifriranja v izhodnih podatkih. Dobri šifrirni algoritmi zagotavljajo izhodne podatke, ki statistično ne odstopajo od naključnega tvorjenja zaporedij znakov.

Dešifriranje je sestavljeno iz inverzne linearne transformacije, inverzne S-tabele in mešanja podključev v obratnem vrstnem redu kot pri šifriranju. Da se ohrani enaka pretočnost podatkov tako pri šifriranju kot pri dešifriranju, se običajno izpustijo linearne transformacije zadnjega cikla [Par1998, Sti2002].

6.4.1.1 S-tabela

Vsaka S-tabela velikosti $m \times n$ opravlja nelinearno bijektivno preslikavo m vhodnih bitov v n izhodnih bitov, kar ustvarja navidezno zmedo v podatkih. Izvedena je lahko v obliki poizvedbene tabele (angl. *lookup table*) z 2^m besedami, dolgimi n bitov (slika 6.4, zgoraj). Običajno so tabele fiksne, kot npr. v šifrinskem algoritmu DES (angl. *data encryption standard*) [Par1998, Sti2002]. Nekateri šifrirni algoritmi tabele ustvarijo dinamično, na podlagi ključa, npr. algoritom EAS [Dae2002] (slika 6.4, spodaj). Šifrirni algoritom lahko ima več S-tabel, ki lahko opravijo različne preslikave.

Učni primer S-tabele predstavlja tabela šifrirnega algoritma DES (slika 6.4 zgoraj). Glede na 6-bitno vhodno zaporedje se 4-bitni izhod v S-tabeli s slike 6.4 določi tako, da se s pomočjo zunanjih dveh bitov (prvega in zadnjega bita) določi vrstica tabele, sredinski štirje biti pa določajo stolpec tabele. Na primer, vnos »100100« ima zunanjega bita »10« in notranje bite »0010«, kar ustrezna izhodu »0001« (slika 6.4 zgoraj).

DES		Srednji štirje vhodni biti															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Zunanja vhodna bita	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

N2 \ N1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Slika 6.4: Primer substitucijske tabele (S-tabele) šifrirnega algoritma DES (zgoraj) in šifrirnega algoritma EAS (spodaj). V prikazanih primerih smo s S-tabelo algoritma DES vrednost 100100 pretvorili v vrednost 0001, s S-tabelo algoritma AES pa vrednost 75 v 9D.

S-tabele so edina nelinearna komponenta v obeh predstavljenih kodirnih arhitekturah, zato so intenzivno preiskovali njihovo varnost. Pokazano je bilo, da so S-tabele skrbno zasnovane in da zagotavljajo maksimalno odpornost na diferencialno kriptoanalizo [Par1998, Sti2002].

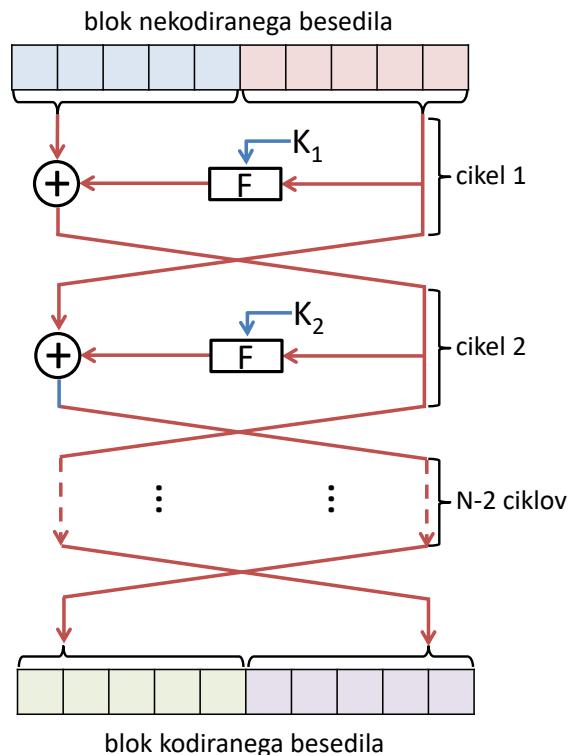
Nekateri sodobni kodirniki ne uporabljajo S-tabel, saj naj bi le-te predstavljalje preveliko breme za predpomnilnik sodobnih procesorjev in šifrirne algoritme preveč izpostavile napadom z merjenji procesorskega časa (angl. *timing attacks*) [Par1998, Sti2002].

Leta 2001 se je zaključila pobuda za razvoj novih standardov bločnih šifrirnih algoritmov. Kot naslednika standarda DES je ameriški nacionalni inštitut za standarde in tehnologijo (angl. *National Institute of Standards and Technology – NIST*) izbral na mreži za zamenjavo in permutacijo temelječi algoritem Rijndael [Dae2002, Rij2013] in ga objavil pod imenom Advanced Encryption Standard (AES).

6.4.2 Mreža Feistel

Drugo najpogosteje uporabljeno arhitekturo bločnih šifrirnih algoritmov predstavlja mreža Feistel (slika 6.5). V i -tem ciklu mreže se desna polovica vhodnih podatkov posreduje na vhod nelinearne funkcije F [Par1998, Sti2002]. Slednja prejme kot vhodni parameter tudi podključ K_i , pogosto pa je sestavljena iz korakov za mešanje ključa, S-tabel in linearne transformacije. Izhod iz funkcije F je preko bitne operacije izključujuči ALI (angl. XOR) združen z levo polovico vhodnih podatkov. Izhod cikla je nato oblikovan z zamenjavo nešifriranih in šifriranih vhodnih podatkov (slika 6.5).

Predstavnik mreže Feistel je Camellia, ki je bila skupaj z algoritmom AES februarja 2003 vključena v portfelj NESSIE2 priporočenih 128-bitnih bločnih šifrirnih algoritmov [Nes2004].

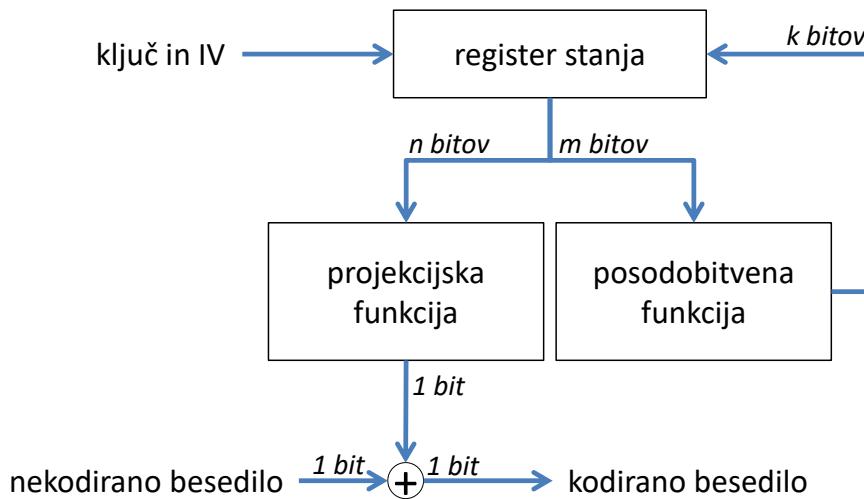


Slika 6.5: Mreža Feistel z N cikli. S simbolom \oplus smo označili bitno operacijo izključujuči ALI.

6.5 Pretočni simetrični šifrirni algoritmi

Pretočni šifrirni algoritem (angl. *stream cipher*) je simetrični šifrirni algoritem, ki se uporablja v primerih, ko je količina podatkov vnaprej neznana, podatkovni tok pa je časovno nezvezen oziroma heterogen. Algoritem ustvarja zaporedje kriptografsko varnih bitov, imenovanih bitno zaporedje ključa (angl. *key stream*) [Par1998, Sti2002]. To zaporedje se nato na nivoju bitov, z uporabo bitne operacije XOR, kombinira z nekodiranim ali kodiranim besedilom (slika 6.7). Postopka šifriranja in dešifriranja sta zaradi bijektivnosti bitne operacije XOR popolnoma enaka.

Osnovna topologija pretočnega kodirnika je sestavljena iz registra za shranjevanje internega stanja kodirnika (slika 6.6), ki se na začetku določi s pomočjo ključa in inicializacijskega vektorja (IV) [Par1998, Sti2002]. Vsebina registra se redno posodablja s funkcijo za posodobitev stanja (angl. *feedback function*). Algoritem vsebuje še nelinearno projekcijsko funkcijo (angl. *reduction function*), ki vzame del vsebine ali celotno vsebino registra s stanjem algoritma in jo združi v en sam bit psevdonaključnega bitnega zaporedja ključa. Ta bit se nato preko operacije XOR združi s trenutno obdelovanim bitom nešifriranega ali pa šifriranega besedila. Po operaciji XOR se, s funkcijo za posodobitev, posodobi internalno stanje registra in postopek šifriranja se ponovi za naslednji bit vhodnega besedila (slika 6.6).



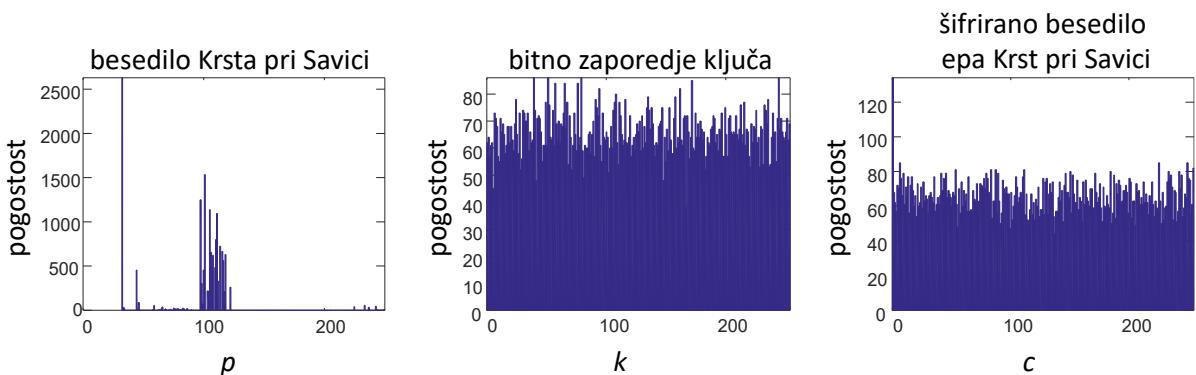
Slika 6.6: Splošna struktura pretočnega šifrirnega algoritma.

S stališča varnosti je zelo pomembna inicializacija stanja šifrirnega algoritma s pomočjo vhodnega ključa in inicializacijskega vektorja IV. Nepravilna inicializacija notranjega stanja lahko namreč vodi v odkritje ključa [Par1998, Sti2002]. Zaradi tega je zelo pomembno, da je

funkcija za posodobitev registra stanja (v bistvu sekljalna funkcija) kriptografsko zelo močna in da se ob inicializaciji šifrirnega algoritma njegovo stanje posodobi v dovolj velikem številu iteracij. Večina algoritmov ima to funkcionalnost implicitno vgrajeno. Redukcijska funkcija, ki se uporablja za preslikavo stanja v bitno zaporedje ključa, je lahko kriptografsko precej šibkejša [Par1998, Sti2002].

Posodobitev internega stanja je lahko od prejetega besedila odvisna ali neodvisna. Glede na to delimo pretočne kodirnike v dve večji skupini [Par1998, Sti2002]:

1. **Sinhroni pretočni šifrirni algoritmi** (angl. *synchronous stream cipher*) posodabljajo interno stanje kodirnika neodvisno od prejetega šifriranega oziroma nešifriranega besedila. Pošiljatelj in prejemnik morata biti popolnoma sinhronizirana, drugače dešifriranje ni mogoče. Če se med pošiljanjem kodiranih podatkov podatkovnemu toku dodajo ali odvzamejo dodatni znaki, se sinhronizacija izgubi.
2. **Asinhroni šifrirni algoritmi oziroma šifrirni algoritmi s samodejno sinhronizacijo** (angl. *asynchronous or self-synchronising stream cipher*) posodabljajo interno stanje glede na poslane ali prejete znake šifriranega besedila in omogočajo samodejno sinhronizacijo pošiljatelja in prejemnika. Postopek samodejne sinhronizacije je bil patentiran že leta 1946.



Slika 6.7: Primerjava histogramov vrednosti besedila Prešernovega Krsta pri Savici, zapisanega v ASCII-kodu (p), naključnega bitnega zaporedja ključa k in kodiranega besedila Krsta pri Savici v ASCII-kodu (c). Besedilo je bilo preko operacije XOR šifrirano z bitnim zaporedjem ključa, ki je predstavljen na srednjem grafu.

Pretočni simetrični šifrirni algoritmi imajo večinoma boljšo pretočnost podatkov in nižje stroške šifriranja kot bločni algoritmi, a so vse do leta 2004 veljali za manj učinkovite pri zagotavljanju varnosti [Cry2016]. Večina pretočnih kodirnikov namreč ne zagotavlja pristnosti prejetega sporočila, temveč samo zaupnost – šifrirano sporočilo se lahko med prenosom spremeni.

Glede na Shannonov teorem [Sha1949] se mora v pretočnih šifrirnih algoritmih uporabiti popolnoma naključno bitno zaporedje ključa, da je zagotovljena varnost. To zaporedje mora biti vsaj dolžine sporočila, ki ga šifriramo, saj se z njim kombinira s pomočjo bitne operacije XOR. Vsi pretočni šifrirni algoritmi uporabljam veliko krajše ključe, tipične dolžine 128 ali 256 bitov. Na podlagi tega ključa tvorijo omenjeno psevdonaključno bitno zaporedje ključa. S tem je teoretično ogrožena varnost kodirnikov, saj je generirani niz zlogov sedaj psevdonaključen in ne povsem naključen, kot zahteva Shannonov teorem.

Pretočnih kodirnikov dolgo niso podrobno preučili s kriptoanalizo. Za primer navedimo algoritem A5/1, ki se v Evropi uporablja pri mobilnih telefonih, a je zelo nevaren [Bir2000]. Enako velja za algoritem RC4, ki se uporablja v standardih WEP in WPA [AlF2013]. Še odmevnje je dejstvo, da je bilo vseh šest algoritmov, ki so bili v okviru iniciative NESSIE leta 2003 oddani v preizkus, spoznanih za prešibke [Nes2004]. Nobeden od njih ni bil izbran kot ustrezan, kar namiguje, da je težko zasnovati varen pretočni kodirnik. Varnost pretočnih šifrirnih algoritmov je bila temeljito preučena šele v Evropskem projektu eSTREAM [eST2012], ki je izbral in kot varnostno ustrezne predlagal štiri izmed 34 oddanih kandidatov.

6.6 Kvantni algoritmi in napadi na asimetrične in simetrične šifrirne algoritme

V poglavju 5.3 smo opisali učinkovit kvantni algoritem za faktorizacijo celih števil, ki ga je izumil Peter Shor. Algoritom celo število n faktorizira v polinomskem času $O((\log n)^3)$, pri tem pa potrebuje dovolj velik vhodni kvantni register, da lahko vanj shrani število n^2 . Za faktorizacijo števil, ki so uporabljeni v trenutnih ključih šifrirnega algoritma RSA, bi torej potrebovali kvantne registre velikosti vsaj 3000 kvantnih bitov. Ta velikost precej presega trenutne laboratorijske rezultate, kjer so eksperimentalno dokazali superpozicijo kvantnih stanj 14-bitnega registra [Mon2011]. Kanadsko podjetje D-wave systems [Dwa2016] sicer trdi, da ima 1152-bitni kvantni procesor, a ta trditev je bila v preteklosti ovržena s strani številnih akademskih raziskovalnih institucij [Cho2014]. Poleg tega je kvantni računalnik podjetja D-wave, tudi po priznanju proizvajalca, omejen na adiabatično optimizacijo, torej na optimizacijo problema, ki mora upoštevati poljubno, a vnaprej določeno število omejitve. Omenjena optimizacija ne omogoča učinkovitega iskanja skritega ključa v asimetričnih šifrirnih algoritmih, podrobnejše pa jo opišemo v podpoglavlju 7.2.1.

Prav tako obstaja učinkovit kvantni algoritem za izračun diskretnega logaritma, ki ga je po zgledu algoritma za faktorizacijo celih števil predlagal Peter Shor [Sho1997b]. Shorov algoritem za diskrete logaritme zahteva tri kvantne registre, dva vhodna in enega izhodnega, drugače pa je precej podoben algoritmu za faktorizacijo celih števil, ki smo ga predstavili v poglavju 5.3. Podrobnejši opis algoritma najdemo v [Par2008].

Posebna oblika diskretnih logaritmov so eliptične krivulje in Shorov algoritmom za eliptične krivulje je opisan v [Pro2003]. Oba omenjena algoritmi lahko resno ogrozita asimetrične šifrirne algoritme na kvantnih računalnikih z registri v velikosti nekaj tisoč kvantnih bitov. Taki sistemi trenutno še ne obstajajo, saj znanstveniki še ne znajo učinkovito nadzirati tako velikega števila kvantnih bitov (glejte poglavje 7). Kljub tem omejitvam prevladuje prepričanje, da je vesolje velik kvantni sistem in da bo izdelava kvantnih računalnikov z nekaj tisoč kvantnimi biti mogoča v bližnji prihodnosti.

Simetrični šifrirni algoritmi, v katerih se sporočilo preko bitne operacije XOR kombinira s (psevdo)naključnim bitnim zaporedjem kriptografskega ključa, so lahko zelo varni, če se uporabljam pravilno. Vendar pa so ranljivi, če se ne upoštevajo naslednji ukrepi:

1. Isti kriptografski ključ se ne sme nikoli uporabljati dvakrat (glejte zgleda 6.3 in 6.4).
2. Uspešno dešifrirani prejeti podatki se ne smejo nikoli obravnavati kot zagotovilo za pristnost komunicirajočih strank.

Zgled 6.3: Napad s ponovno uporabo ključa

Recimo, da pošljemo sporočili A in B , ki sta enake dolžine, in da obe sporočili šifriramo z istim ključem K . Pretočni kodirnik pripravi bitno zaporedje ključa $C(K)$ v dolžini sporočila. Šifrirani različici sporočil sta potem:

$$E(A) = A \text{ xor } C(K) \text{ in}$$

$$E(B) = B \text{ xor } C(K),$$

kjer se operacija *xor* izvaja bitno.

Recimo, da tretja oseba prestreže $E(A)$ in $E(B)$ in izračuna

$$E(A) \text{ xor } E(B).$$

Ker je *xor* komutativna bitna operacija in ima lastnost, da je $X \text{ xor } X = 0$, velja

$$E(A) \text{ xor } E(B) = (A \text{ xor } C) \text{ xor } (B \text{ xor } C) = A \text{ xor } B \text{ xor } C \text{ xor } C = A \text{ xor } B.$$

Če je eno izmed sporočil znano, je rešitev trivialna. Tudi ko sta obe sporočili neznani, je možno s premetavanjem bitov hitro razvozlati obe sporočili, posebno če sta v naravnih jezikih. Navadno zadostuje nekaj minut procesorskega časa na osebnem računalniku.

Zgoraj opisani scenarij je pogost, zlasti ko se stalno pošiljajo komunikacijski simboli in se, tudi ko dejanskega sporočila ni, pošiljajo ustrezno šifrirani znaki NULL. Slednje se pogosto uporablja v vojaških komunikacijah.

Zgled 6.4: Napad z zamenjavo besedila

Kadar tretja oseba pozna vsebino vseh ali dela enega od naših sporočil, lahko prestreže podatkovni tok in spremeni vsebino sporočila, ne da bi poznala ključa K . Na primer, če vemo, da specifičen del sporočila vsebuje v ASCII-kodu zapisan niz »ni dobro«, lahko s pomočjo bitne operacije xor ta niz nadomestimo s katerimkoli sporočilom iste dolžine:

$$(C(K) \ xor \ »je\ dobro«) \ xor \ (»je\ dobro« \ xor \ »ni\ dobro«) = C(K) \ xor \ »ni\ dobro«,$$

kjer je $C(K)$ bitno zaporedje ključa.

Napadi z zamenjavo besedila se preprečijo z vključitvijo sekljane kode za preverjanje avtentičnosti sporočila (angl. *message authentication code – MAC*). Ta poveča verjetnost, da bomo takšen napad odkrili.

Trenutno ni znan kvantni algoritem, ki bi omogočal učinkovito iskanje skritega ključa simetričnih šifrirnih algoritmov. Najbolj se temu približa Groverjev algoritem za nestrukturirano iskanje, ki smo ga podrobnejše opisali v poglavju 5.2. Bitni ključ z dolžino n bitov uspe najti v $\mathcal{O}(\sqrt{2^n}) = \mathcal{O}(2^{n/2})$ iskanjih. Čeprav je za velike n ta pohitritev zelo velika, so simetrični šifrirni algoritmi relativno odporni na to grožnjo. Vse, kar je treba narediti, je, da podvojimo dolžino ključev. Ta korak so številna strokovna združenja in svetovalna telesa že naredila in namesto 128-bitnih skritih ključev, ki so dovolj varni na klasičnih računalnikih, je sedaj priporočena uporaba 256-bitnih ključev.

Slabost in ranljivost simetričnih šifrirnih algoritmov je njihova odvisnost od varne izmenjave skritega ključa. Že leta 1940 je Claude Shannon dokazal, da je šifrirni algoritem teoretično varen, če in samo če tako pošiljatelj kot prejemnik uporablja bitno zaporedje ključa, ki je dolžine komunikacijskega sporočila [Sha1949]. Bitno zaporedje ključa se ne sme nikoli ponoviti, saj s ponavljanjem bitnega zaporedja tvegamo napad s ponovno uporabo ključa (glejte zgled 6.3). Ker je izmenjava celotnih bitnih zaporedij ključa nesmiselna in neučinkovita, navadno komunicirajoči stranki izmenjata veliko krajsko skrivno zaporedje bitov, ki se nato uporabi kot seme v psevdonaključnem generatorju bitnega zaporedja ključa. Zaradi zagotavljanja varnosti je treba periodično posodabljati skupno skrivnost komunicirajočih strani (seme psevdonaključnega generatorja bitnega zaporedja ključa). Omenjena posodobitev je potrebna tudi, ko se oddajnih in prejemnik zaradi izgube podatkov ali kakšnega drugega razloga desinhronizirata (poglavlje 6.5).

Izmenjava skupne skrivnosti navadno temelji na asimetričnih šifrirnih algoritmih (glejte opis Diffie-Hellmanovega algoritma v poglavju 6.2.1). Zato je izmenjava skrivnosti (semena psevdonaključnega generatorja) ranljiva in ogrožena s strani Shorovih kvantnih algoritmov. Pojavila se je torej potreba po varnejši izmenjavi ključev simetričnih šifrirnih algoritmov. Ena izmed možnih rešitev je varna izmenjava bitov preko kvantnih omrežij. Opisujemo jo v naslednjem poglavju.

6.7 Kvantna izmenjava skritega ključa

Kvantna izmenjava skritega ključa omogoča detekcijo prisluškovana s poljubno visoko verjetnostjo in trenutno sodi med teoretično najbolj varne načine za izmenjavo skritega ključa. Navadno temelji na izmenjavi fotonov, ki si jih komunicirajoči stranki pošiljata po optičnem vodniku, možna pa je tudi brezžična komunikacija, ki je navadno izvedemo s pomočjo laserja [Urs2007]. Za razliko od kvantnih računalnikov, ki so šele v začetni fazi razvoja, so kvantna omrežja, ki omogočajo varno komunikacijo, relativno dobro preizkušena in že danes omogočajo relativno velike komunikacijske hitrosti na razdaljah, daljših od 100 km (glejte poglavje 7). V praksi se uporabljata predvsem protokola BB84 in E91, ki sta predstavljena v nadaljevanju.

6.7.1 Protokol BB84

Protokol BB84 je bil predlagan leta 1984, poimenovan pa je po avtorjih Charlesu H. Bennetu in Gillesu Brassardu [Ben1984]. Protokol predvideva, da se komunicirajoči stranki, recimo Špela in Tine, javno dogovorita o dveh bazah za merjenje polarizacije fotonov (tabela 6.2). Pomembno je, da sta ti dve bazi prostorov medsebojno zasukani za 45 kotnih stopinj, tako imamo ob meritvi fotona, ki smo ga polarizirali v bazi $+$ in izmerili v bazi \times , 50 % možnost, da izmerimo polarizacijo \nearrow , in 50 % možnost, da izmerimo polarizacijo \nwarrow . Ravno tako imamo ob meritvi fotona, ki smo ga polarizirali v bazi \times in izmerili v bazi $+$, 50 % možnost, da izmerimo polarizacijo \rightarrow , in 50 % možnost, da izmerimo polarizacijo \uparrow .

Tabela 6.2: Baza za merjenje polarizacije fotona.

Baza meritve	Vrednost bita 0	Vrednost bita 1
$+$	\rightarrow	\uparrow
\times	\nearrow	\nwarrow

Špela tvori naključno zaporedje bitov in za vsak bit naključno izbere eno izmed dveh baz meritev, ki sta predstavljeni v tabeli 6.2. Zaporedje bitov in izbira baz je Špelina skrivnost. Glede na vrednost bita in izbrano bazo meritev Špela polarizira posamezen foton in ga pošlje Tinetu (tabela 6.3). Tine naključno izbere bazo meritev in izmeri polarizacijo fotona. Če se baza meritev ujema s Špelino bazo polarizacije, bo Tine izmeril pravo polarizacijo fotona in posledično pravo vrednost bita. V nasprotnem primeru pa ima 50 % možnosti, da izmeri pravo, in 50 % možnosti, da izmeri napačno vrednost bita. Po končani izmenjavi celotnega niza bitov Špela in Tine javno in po čim več komunikacijskih potek izmenjata informacije o bazah polarizacije fotonov. Bite, pri katerih se Špelina in Tineta baza ujemata, obdržita kot skupno skrivnost (tabela 6.3), ostale bite pa zavrzeta.

V protokolu BB84 je možno prisluškovanje zaznati s poljubno verjetnostjo. Zadostuje, da žrtvujemo N od M bitov skritega ključa (skriti ključ zmanjšamo na $M - N$ bitov) in jih medsebojno primerjamo (Špela in Tine jih izmenjata preko javnega omrežja, zopet po čim več komunikacijskih poteh). Predpostavimo, da je komunikaciji med Špelo in Tinom prisluškovala Eva. Verjetnost, da bo prisluškovalka Eva izbrala napačno bazo pri posameznem bitu, je 50 %. Eva ponovno pošlje foton, ki ima naključno polarizacijo, torej bo v bazi, ki jo je izbrala Špela (in po naključju tudi Tine), zavzel napačno vrednost z verjetnostjo 50 %. Skupna verjetnost, da bo pri prisluškovanju prišlo do napake v bitu skritega ključa, je potem enaka $0,5 \cdot 0,5 = 0,25$, torej 25% [Ben1984]. Verjetnost, da bomo ob prisluškovanju odkrili napako v skritem ključu, je torej enaka

$$P(\text{detekcija prisluškovanja}) = 1 - (3/4)^N, \quad (6.5)$$

pri čemer je N število bitov skritega ključa, ki smo jih žrtvovali za preverjanje. Če želimo prisluškovanje zaznati z **verjetnostjo 99,9999 %**, je treba žrtvovati $N=50$ bitov skritega ključa. Napad s prisluškovalcem na sredini je možen le, če se prisluškovalec med Tineto in Špelo vrne v prav vseh komunikacijskih poteh. Zato je pomembno, da Špela in Tine po izmenjavi fotonov vse informacije izmenjata preko čim več javnih kanalov komunikacije in s tem izdatno otežita prisluškovanje.

Tabela 6.3: Prikaz delovanja protokola BB84.

Špelini naključni biti	1	0	1	0	1	1	1	0	0
Špelina naključna izbira baze	+	+	×	×	+	×	+	×	+
Polarizacija fotonov, ki jih pošlje Špela	↑	→	↖	↗	↑	↖	↑	↗	→
Tinetova naključna baza meritev	×	+	×	+	+	+	×	×	×
Tinetove izmerjene polarizacije fotonov	↗	→	↖	→	↑	↑	↖	↗	↖
Javna diskusija o bazi meritev za vsak izmerjeni foton	!	OK	OK	!	OK	!	!	OK	!
Skupen skriti ključ	0	1			1			0	

6.7.2 Protokol E91

Protokol E91, poimenovan po avtorju Arturju Ekertu in letnici objave 1991 [Eke1991], temelji na parih kvantno prepletenih fotonov. V predlagani komunikacijski shemi si dve komunicirajoči stranki, recimo Špela in Tine, izmenjujeta fotone s kvantno prepletenu polarizacijo. Recimo, da Špela generira par fotonov s kvantno prepleteno polarizacijo. Enega izmed fotonov pošlje Tinetu, drugega pa obdrži. Ko Tine prejme foton, izmeri njegovo polarizacijo. Posledično povzroči kolaps valovne funkcije obeh prepletentih fotonov (poglavlje 3.5.1). Kvanti preplet fotonov zagotavlja, da bo Tine vedno izmeril polarizacijo, ki je ortogonalna polarizaciji Špelinega fotona. Torej, ko Tine v bazi meritev + izmeri polarizacijo fotona \uparrow (glejte tabelo 6.2), ve, da bo Špela v isti bazi meritev + zanseljivo izmerila polarizacijo \rightarrow . Če pa Tine v bazi meritev \times izmeri polarizacijo fotona \nearrow , ve, da bo Špela v isti bazi meritev 100-odstotno izmerila polarizacijo \nwarrow . To velja za katerokoli merilno bazo in je zagotovljeno zaradi kvantnega preleta fotonov [Eke1991]. Pomembno je le, da oba, Špela in Tine, polarizacijo merita v isti bazi meritev.

Poudarimo, da za noben foton niti Tine niti Špela ne vesta vnaprej, katero polarizacijo mu bosta izmerila. Če na primer Špela par fotonov generira v polarizacijski bazi \times , bo rezultat meritve para fotonov v bazi + povsem naključen. Tine ima 50 % možnosti, da izmeri polarizacijo \uparrow , in 50 % možnosti, da izmeri polarizacijo \rightarrow . Ve le, da bo Špela izmerila polarizacijo v smeri, ki je ortogonalna njegovi izmerjeni polarizaciji. Sedaj Tine polarizaciji \uparrow dodeli vrednost bita 1, polarizaciji \rightarrow pa vrednost 0. Špela mora narediti obratno, polarizaciji \uparrow dodeli bitno vrednost 0, polarizaciji \rightarrow pa vrednost 1. Po meritvi N prepletentih parov fotonov si Špela in Tine delita N bitov skupne skrivnosti.

Detekcija prisluškovanja poteka na podoben način kot pri protokolu BB84. Tine in Špela naključno izbirata baze meritev posameznih fotonov (tabela 6.2), izbira posamezne baze pa ostane zasebna skrivnost vsakega izmed njiju vse do konca izmenjave fotonov. Nato si Špela in Tine javno in po čim več komunikacijskih poteh izmenjata informacijo o bazah meritev in upoštevata le tiste bite, katere sta oba izmerila v isti bazi meritev. Od tu dalje je postopek za odkrivanje prisluškovanja identičen postopku po protokolu BB84, ki smo ga opisali v prejšnjem podpoglavlju.

Naloge:

1. Opišite razliko med simetričnimi in asimetričnimi kodirniki.
2. Koliko praštevil je manjših od števila 10000? Koliko pa od števila 1000000?
3. Generirajte javni in privatni ključ algoritma RSA za celo število n , ki je sestavljeno iz praštevil $p = 2719$ in $q = 383$.
4. Generirajte javni in privatni ključ algoritma RSA za celo število n , ki je sestavljeno iz praštevil $p = 2749$ in $q = 1901$.

5. Izmerite razliko med hitrostjo kodiranja in dekodiranja podatkov z algoritmom RSA, ki ima javni ključ $n = 257573$ in $e = 65537$ ter privatni ključ $d = 249569$.
6. Diffie-Hellmanov algoritem v zgledu 6.2 spremenite tako, da uporabite osnovo $g = 5$.
7. Koliko bitov skrivnosti si v povprečju izmenjata Špela in Tine, če pri Diffie-Hellmanovem algoritmu uporabita praštevilo $p = 2221$ in osnovo $g = 2$ (glejte zgled 6.2)?
8. Kako izgleda eliptična krivulja, definirana z enačbo (6.1), če velja $4a^3 + 27b^2 = 0$?
9. Grafično preverite, da velja $\lim_{k \rightarrow \infty} k\mathbf{P} = \mathbf{0}$ za eliptično krivuljo in točko \mathbf{P} , ki sta prikazani na sliki 6.2.c.
10. Kako je definiran red točke \mathbf{P} na eliptični krivulji?
11. Opišite izmenjavo ključev po protokolu ECMQV.
12. Razmislite o njihovi odpornosti na znane kvantne algoritme in opiši pomen S-tabel kot edinih nelinearnih operacij simetričnih kodirnikov.
13. Opišite izmenjavo skritega ključa po protokolu BB84.
14. Koliko fotonov si morata v povprečju izmenjati Špela in Tine, da bosta generirala 128-bitni skupni skriti ključ, prisluškovanje pa zaznala z verjetnostjo 99,9999 %?
15. Koliko skupnih bitov je treba žrtvovati v protokolu BB84, da zaznamo prisluškovanje z verjetnostjo 99,99 %, in koliko, da prisluškovanje zaznamo z verjetnostjo 99,99999999 %?
16. Opišite izmenjavo skritega ključa po protokolu E91.

7 Kvantni računalniki in kvantna omrežja

Po definiciji sta koherentna sistema tista sistema, ki sta fizično ali informacijsko povezana. V kvantni fiziki s samostalnikom koherenca označujemo sposobnost interference delca s samim sabo. Slednje opišemo z valovno funkcijo in Schrödingerjevo enačbo (3.1). Pravimo, da je lahko kvantni delec v superpoziciji vseh možnih stanj in da so kvantna stanja delca medsebojno koherentna. Koherenca je torej lastnost, po kateri se kvantni delec ali, gledano širše, sistem loči od klasičnega.

Z besedo dekoherenca označujemo izgubo koherence kvantnega sistema, torej izgubo kvantnih superpozicij vseh možnih stanj in prehod v klasično stanje. Čeprav lahko meritev razložimo z dekoherenco kvantnega sistema, navadno razlikujemo med obema pojmom. Dekoherenca pomeni postopno in navadno neželeno izgubo superpozicije vseh stanj, torej postopno zmanjševanje prostostnih stopenj valovne funkcije oziroma postopno prehajanje kvantnega v klasični sistem. Meritev pa, vsaj po kopenhagenski interpretaciji (poglavlje 3.3.1), pomeni trenutni kolaps valovne funkcije, torej trenutni želeni prehod iz kvantnega v klasično stanje.

Kvantni sistemi so v neprestani interakciji z okoljem, zato se njihova kvantna stanja vedno prepletajo s kvantnimi stanji okolja. To povzroči prepletanje valovnih funkcij sistema z valovnimi funkcijami okolja. V jeziku kvantnega računalništva lahko sistem opišemo z naborom kvantnih bitov (s kvantnim registrom), njegovo okolje pa z naborom drugih kvantnih bitov, torej z drugim kvantnim registrom. Pred interakcijo sistema z okoljem sta oba kvantna registra neodvisna, kvantni register sistema pa je lahko v poljubni notranji superpoziciji stanj. Pri interakciji z okoljem se kvantna regisra prepletata, posledično pa se izgubi superpozicija vseh možnih stanj prvega kvantnega registra. Slednja postanejo prepletena s kvantnimi biti okolja in preko teh prepletov zunanje okolje (drugi kvantni register oziroma sistem) pridobiva informacije o prvem kvantnem registru oziroma sistemu in vpliva na superpozicijo njegovih stanj (glejte podpoglavlje 3.3.3). V skrajnem primeru prvi kvantni register kolabira v eno izmed klasičnih stanj, torej v stanje, ki ga pričakujemo po meritvi kvantnega sistema.

Interakcijo z okoljem je zelo težko preprečiti, saj so kvantni delci, ki fizično tvorijo naš prvi kvantni register, vedno del širšega sveta, ki jih obdaja. Lahko jih bolj ali manj uspešno ločimo od okolja, velja pa, da je ta ločitev veliko težja za velike kot za majhne kvantne sisteme. Preprečevanje dekoherence je torej težaven problem, njegova težavnost pa raste s številom kvantnih bitov v kvantnem registru. Pri makroskopskih telesih, ki vključujejo veliko število osnovnih delcev, je stopnja dekoherence tako velika, da se telesa obnašajo kot klasična telesa (glejte podpoglavlje 3.3.3).

Dekoherence ne moremo nikoli popolnoma odpraviti. Poskrbimo lahko le, da poteka dovolj počasi, da lahko nad superpozicijo vseh kvantnih stanj izvedemo zahtevane računske operacije. Pravimo, da mora biti čas dekoherence kvantnega registra nekaj velikostnih

razredov večji od časa posameznih računskih operacij nad tem registrom. Omenjeno pravilo je del **DiVincenzovih meril za fizično izvedbo kvantnega računalnika** [DiV1997]:

1. Imeti moramo dobro definiran in razširljiv register kvantnih bitov, torej stabilen pomnilnik.
2. Register mora biti nastavljen v klasično stanje »000...«.
3. Zagotoviti moramo dolge čase dekoherence ($> 10^4 \cdot \text{čas procesiranja}$).
4. Imeti moramo univerzalni nabor vrat.
5. Meritev posameznih kvantnih bitov mora biti preprosta.
6. Omogočena mora biti pretvorba med stacionarnimi in premikajočimi se kvantnimi biti (fotoni).
7. Omogočen mora biti transport premikajočih se kvantnih bitov (fotonov) med dvema fizičnima lokacijama.

Merili 6 in 7 sta bili dodani za potrebe komunikacije med kvantnimi sistemi in jih za implementacijo enega samega kvantnega računalnika ne potrebujemo.

Na podlagi DiVincenzovih meril so bili zasnovani številni fizični sistemi, ki so kandidati za izvedbo kvantnega računalnika. Trenutno še ni znano, kateri izmed njih (če sploh kateri) bo na koncu vodil k izvedbi velikih kvantnih računalnikov. Ne vemo še namreč, katera izmed tehnologij se bo izkazala za najoptimalnejšo, zato moramo preveriti vse. Trenutno sodijo med najobetavnejše naslednje tehnologije:

- superprevodniki (angl. *superconductors*): kvantni biti so predstavljeni s stanji majhnih superprevodnih vezij;
- ionske pasti (angl. *ion traps*): kvantni biti so predstavljeni z ioni, ki so ujeti v elektromagnetni pasti;
- optične mreže (angl. *optical lattices*): kvantni biti so predstavljeni z notranjimi stanji atomov, ki so ujeti v optične mreže;
- kvantne pike (angl. *quantum dots*): kvantni biti so predstavljeni s spinii elektronov, ki so ujeti v kvantne pike;
- magnetna resonanca (angl. *nuclear magnetic resonance*): kvantni biti so predstavljeni s spinii atomskih jeder.

7.1 Ionske pasti

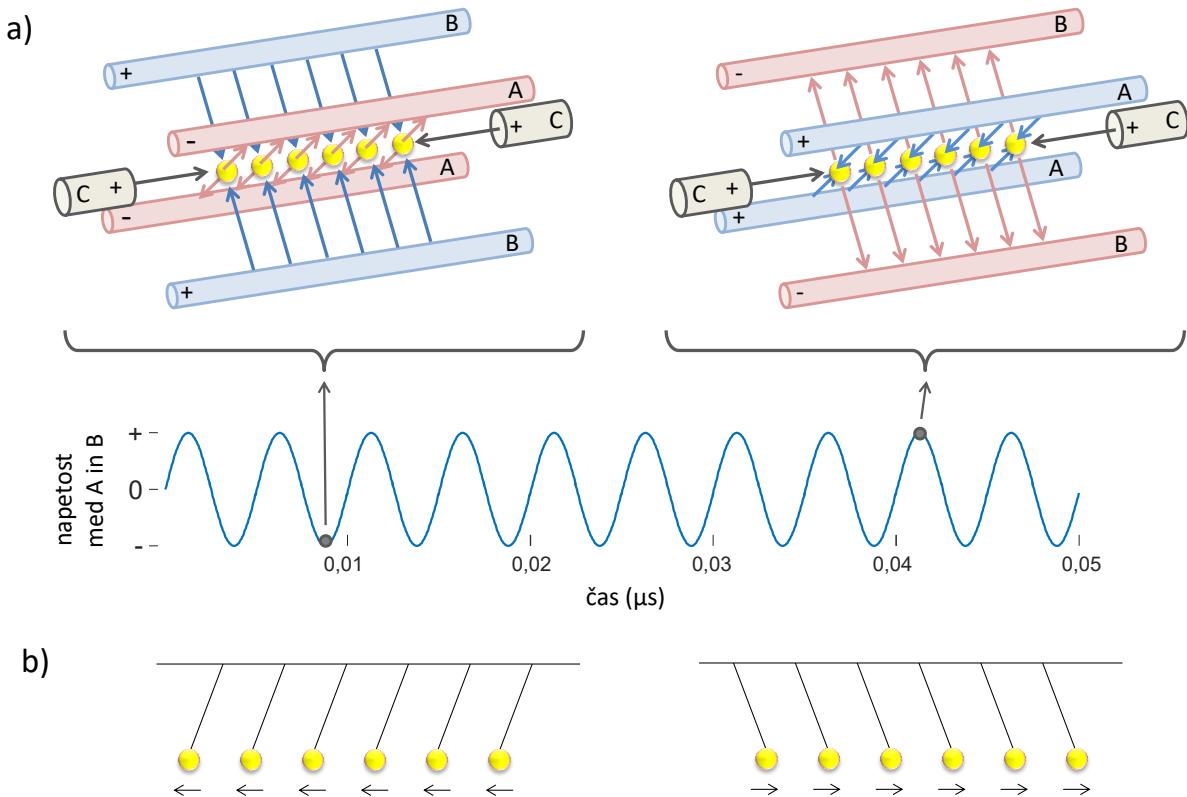
Takoj po odkritju Shorovega algoritma za faktorizacijo celih števil sta leta 1995 Ignacio Cirac in Peter Zoller [Cir1995] za izgradnjo malih kvantnih računalnikov predlagala ionske pasti, v katerih prostolebdeče ione vzbujamo z lasersko svetlobo. Razlog je bil večleten razvoj in

obvladovanje tehnologij za nadzor enega (ali več) ionov na področju ultravisoko natančne spektroskopije in atomskih ur.

Pozitivno nabite ione vstavimo v Paulovo ionsko past, ki je poimenovana po nemškem fiziku Wolfgangu Paulu. Sestavlja jo trije pari elektrod, ki so na sliki 7.1.a označeni s črkami A, B in C. Elektrodi C sta pod konstantno pozitivno napetostjo in odbijata pozitivne ione. Na elektrodah A in B je izmenična napetost, ki niha s frekvenco f_{RF} . V časovnem trenutku, ki je prikazan v zgornjem levem kotu slike 7.1 a), je par elektrod A pozitivno nabit in ione odbija, par elektrod B pa je negativno nabit, zato ione privlači. V časovnem trenutku, ki je prikazan na sliki 7.1.b, pa je pozitivno nabit par elektrod B, medtem ko sta elektrodi A negativno nabit. Elektrode A in B torej izmenično privlačijo in odbijajo ione. Ker imajo ioni maso, na spremembo električnega polja ne odreagirajo takoj. Če izmenično napetost na elektrodah spremenjamo z dovolj visoko frekvenco, ki je v razponu radijskih frekvenc (RF), se ioni ne uspejo pravočasno odzvati na spremembe električnega polja in se ustalijo v energijskem minimumu električnega polja elektrod A in B. Ker se zaradi Colombove sile ioni odbijajo, se razvrstijo v ravni črti na sredino med elektrode A in B. Hkrati elektrode C s svojo odbojno električno silo preprečijo, da bi ioni iz pasti pobegnili na skrajnih robovih elektrod A in B. Celotna past je vakuumsko zaprta, vendar ostalih nabitih delcev pa dodatno preprečimo z zunanjimi električnimi polji, ki na sliki 7.1 niso prikazana [Cir1995].

Ujete ione lahko ohladimo tako, da ostanejo omejeni na določeno regijo prostora in se gibljejo z najmanjšo možno energijo (slika 7.1 b). Njihova notranja stanja lahko natančno spremenjamo z laserjem, meritve pa opravimo s 100-odstotno natančnostjo. Ioni so medsebojno močno sklopljeni z odbojno Coulombovo silo (pomembno za kvantno prepletanje), hkrati pa so učinkovito ločeni od okolja (pomembno za preprečevanje dekoherence).

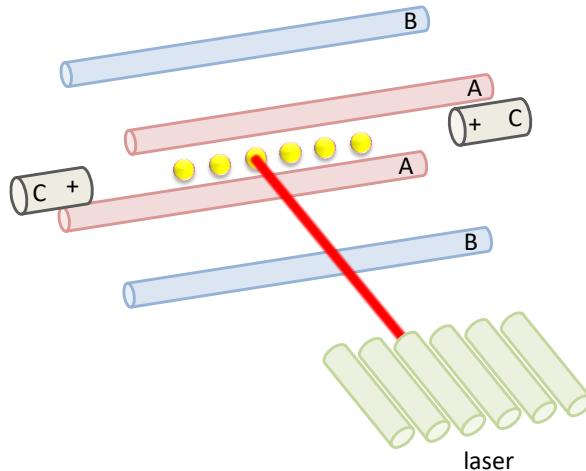
Kvantna informacija je shranjena v energijskih stanjih posameznih ionov (notranje stanje iona) ter v skupnem nihanju ionov (zunanje stanje ionov). Notranje energijsko stanje iona spremenjamo tako, da s svetlobo zelo natančno določene frekvence vzbudimo njegove elektrone na višje energijsko raven. Frekvenca svetlobe, ki vzbuja elektrone, je v mikrovalovnem območju, zaradi česar je mogoče za upravljanje stanj uporabiti mikrovalovno sevanje. Vendar trenutno ne poznamo vira mikrovalovnega sevanja, ki bi tvoril dovolj ozek curek fotonov, s katerim bi lahko nadzorovali posamezen ion v ionski pasti. Namesto tega je mogoče uporabiti par laserskih pulzov z različnima frekvencama (razlika frekvenc mora biti enaka zahtevani frekvenci za prehod energijskih stanj elektronov). Z laserjem lahko proizvedemo curek svetlobe, ki je širok okoli $2 \mu\text{m}$, ioni pa so v pasti na razdalji med 10 in $30 \mu\text{m}$. Z lasersko svetlobo lahko torej zelo natančno osvetlimo izbrani ion (slika 7.2).



Slika 7.1: Paulova ionska past (a) in skupno nihanje ionov v pasti (b).

Napetost na elektrodah A in B izmenično niha s frekvenco v območju radijskih frekvenc, kar je ponazorjeno s sinusido na spodnjem delu slike a). Zato elektrode A in B izmenično privlačijo in odbijajo ione. Ioni (rumene kroglice) zaradi svoje mase ne uspejo slediti hitrim spremembam električnega polja in se razvrstijo v lego povprečnega energijskega minimuma izmeničnega električnega polja, ki se nahaja na sredini vzdolž elektrod A in B. Elektrodi C sta konstantno pozitivno nabiti in odbijata pozitivne ione. Slednji se medsebojno odbijajo s Colombovo silo in se ustalijo na medsebojni razdalji okoli $10\text{-}30 \mu\text{m}$. Ioni nikoli popolnoma ne mirujejo in lahko nihajo na več različnih načinov. Način nihanja, ki ima najnižjo energijo (in temperaturo), je skupno nihanje ionov vzdolž osi, ki povezuje elektrodi C. Tak način nihanja je prikazan na grafu b). Intenzivnost nihanja ionov je kvantizirana in jo merimo v fononih.

Zunanje energijsko stanje ionov je odvisno od nihanja ionov. Earnshawov teorem pravi, da skupna točkasto nabitih delcev ni mogoče ohraniti v stabilni stacionarni legi in da se ioni vedno gibljejo [Ern1842]. Za N ionov v pasti obstaja N načinov nihanja v vzdolžni smeri pasti (med elektrodama C na sliki 7.1). Energijo najnižji način je skupno nihanje vseh ionov v levo in desno (slika 7.1 b)). Intenzivnost nihanja je kvantizirana in jo merimo v fononih [Cha2003]. Ioni lahko skupno nihajo z enim, dvema, tremi ali več fononi.



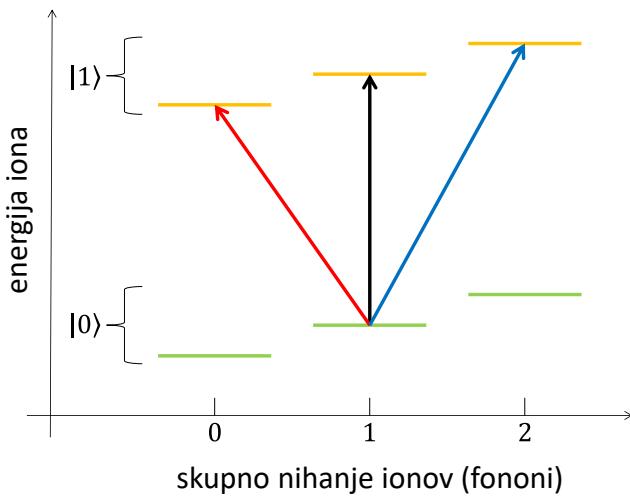
Slika 7.2: Osvetlitev izbranega iona z lasersko svetlobo. Curek laserske svetlobe je širok okoli $2 \mu\text{m}$, ioni pa so na razdalji med 10 in $30 \mu\text{m}$.

Ob obstreljevanju s fotoni posamezen ion skače navzgor in navzdol med parom elektronskih stanj in pri tem seva svetlobo. Pri tem mora imeti svetloba zelo natančno določeno frekvenco $f_{|0\rangle \rightarrow |1\rangle}$, da lahko vzbudi elektron v višje energijsko stanje. Nižje energijsko stanje kodira stanje kvantnega bita $|0\rangle$, višje pa stanje $|1\rangle$ (slika 7.3).

Če ima svetloba manjšo frekvenco, se bo lahko elektron vzbudil v višje energijsko stanje samo, če del energije odvzame skupnemu nihanju ionov. Spomnimo, da je energija skupnega nihanja ionov kvantizirana. Torej elektron ne more vzeti poljubne količine energije, temveč le večkratnik fononov. To pomeni, da bo elektron za svoj prehod in nižjega v višje energijsko stanje odvzel skupnemu nihanju ionov k fononov energije, ko in samo ko bo frekvence svetlobe enaka $f_{|0\rangle \rightarrow |1\rangle} - k\Delta f$, kjer smo z Δf označili frekvenco, ki ustreza energijskemu primanjkljaju enega fonona (slika 7.3).

Podobno bo elektron, ki ga osvetljujemo s svetobo frekvence $f_{|0\rangle \rightarrow |1\rangle} + k\Delta f$, ob prehodu v višje energijsko stanje oddal k fononov energije skupnemu nihanju ionov (slika 7.3). Z osvetljevanjem kateregakoli posameznega iona torej spremojamo intenzivnost skupnega nihanja ionov. V bistvu gre za spodbujen Ramanov prehod (angl. *Raman transition*) [Cha2003]. Frekvanca Δf je precej manjša od frekvence $f_{|0\rangle \rightarrow |1\rangle}$ in frekvenčni pasovi $f_{|0\rangle \rightarrow |1\rangle} \pm k\Delta f$ so dovolj ozki in dovolj razmaknjeni, da omogočajo relativno enostaven nadzor notranjih in zunanjih energijskih stanj ionov.

Z laserjem izbrane frekvence lahko torej vrednost kvantnega bita spremojamo med stanjem $|0\rangle$ in stanjem $|1\rangle$. Pri tem je pomemben tudi čas osvetlitve in za klasično negacijo bita (prehod med klasičnima stanjema $|0\rangle$ in $|1\rangle$) moramo ion osvetljevati $d \mu\text{s}$. Osvetlitev s trajanjem $d/2 \mu\text{s}$ pa ion iz stanja $|0\rangle$ vzbudi v stanje superpozicije $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ [Hol2002]



Slika 7.3: Prehod elektronov posameznega iona iz nižjih v višje energetsko stanje in izmenjava kvantov energije med elektronji in skupnim nihanjem ionov. Če ion osvetlimo s frekvenco $f_{|0\rangle \rightarrow |1\rangle}$ (črna puščica na sliki), preide elektron in nižjega v višje energijsko stanje.

Če ga osvetlimo s svetlobo frekvence $f_{|0\rangle \rightarrow |1\rangle} - \Delta f$ (rdeča puščica na sliki), bo v višje energetsko stanje prišel le tako, da bo skupnemu nihanju ionov odvzel en fonon energije. Če pa ga osvetlimo s svetlobo $f_{|0\rangle \rightarrow |1\rangle} + \Delta f$, bo ob prehodu v višje energijsko stanje en fonon energije oddal skupnemu nihanju ionov.

Povzemimo zgled, ki je opisan v [Hol2002].

Zgled 7.1: Prepletanje kvantnih bitov v ionskih pasteh

Predpostavimo, da imamo v ionski pasti dva iona, ki sta oba v stanju $|0\rangle$, in da skupaj nihata z intenzivnostjo enega fonona:

$$|00\rangle_1, \quad (7.1)$$

kjer smo z indeksom 1 označili skupno nihanje z intenzivnostjo 1 fonona.

S curkom svetlobe s frekvenco $f_{|0\rangle \rightarrow |1\rangle} + \Delta f$ in trajanjem $d/2 \mu\text{s}$ osvetlimo prvi ion in ga vzbudimo v stanje $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Hkrati se presežek energije (zaradi povečanja zahtevane frekvence za Δf) prenese v višjo energijo skupnega nihanja ionov. Iona torej sedaj nihata z intenzivnostjo dveh fononov:

$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)_2. \quad (7.2)$$

Sedaj s svetlobo frekvenco $f_{|0\rangle \rightarrow |1\rangle} - \Delta f$ in dolžine d μs osvetlimo drugi ion. Zaradi manjka energije (drugi ion potrebuje za prehod v stanje $|1\rangle$ frekvenco $f_{|0\rangle \rightarrow |1\rangle}$), drugi ion energijo odvzame skupnemu nihanju ionov, ki se sedaj zopet gibljeti z intenzivnostjo enega fonona. S tem smo prešli v stanje

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)_1. \quad (7.3)$$

Oba iona sta torej kvantno prepletla svoji energijski stanji in kodirata zelo znano stanje EPR (Einstein-Podolsky-Rosen), ki smo ga z enačbo (3.20) opisali v poglavju 3.5.1.

Ioni nimajo samo dveh energijskih stanj elektronov, temveč je teh energijskih stanj več. **Meritev kvantnih bitov** v ionski pasti opravimo z laserjem takšne frekvence, da v višje (tretje) energijsko stanje vzbuja eno izmed stanj $|0\rangle$ ali $|1\rangle$, drugega pa ne. Recimo, da osvetljujemo ion s frekvenco $f_{|0\rangle \rightarrow |\nu\rangle}$, ki vzbudi ion v stanju $|0\rangle$ v višje energijsko stanje $|\nu\rangle$. Če je ion v stanju $|0\rangle$, se bo vzbudil, ob spontani vrnitvi iz stanja $|\nu\rangle$ v stanje $|0\rangle$ pa bo oddal foton, ki ga je moč zaznati s kamero CCD. Če pa je ion v stanju $|1\rangle$, se ne bo vzbudil in tudi ne bo oddal fotona. Z detekcijo fotonov lahko torej s 100-odstotno natančnostjo izmerimo stanje kvantnega bita po meritvi. Omenjena meritev povzroči kolaps valovne funkcije in kolaps superpozicije vseh koherenih kvantnih stanj. V primeru kvantnega stanja $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)_1$, ki smo ga opisali v zgledu 7.1, bomo torej s 50-odstotno verjetnostjo izmerili stanje $|00\rangle$, s verjetnostjo 50 % pa stanje $|11\rangle$. Omenimo, da bosta oba kvantna bita vedno vrnila isto vrednost, ali $|0\rangle$ ali $|1\rangle$, kar lahko razumemo kot dokaz superpozicije, ki smo jo opisali z enačbo (7.3).

Poudarimo še, da imata opisani energijski stanji $|0\rangle$ in $|1\rangle$ ujetih ionov življenjsko dobo vsaj nekaj sekund, višje energijsko stanje $|\nu\rangle$ pa je precej manj stabilno. Ob tem je procesorski čas kvantnih vrat v primeru ionskih pasti velikostnega reda 1 μs [Hol2002].

Opisano nestabilnost višjega energijskega stanja $|\nu\rangle$ lahko uporabimo tudi za preprosto implementacijo kontroliranih vrat NE.

Zgled 7.2: Izvedba kontroliranih vrat NE

Predpostavimo, da imamo v ionski pasti dva iona, ki skupaj nihata z intenzivnostjo nič fononov. S curkov svetlobe s frekvenco $f_{|1\rangle \rightarrow |\nu\rangle} + \Delta f$ in dolžino d μs osvetlimo prvi ion. Če je prvi ion v stanju $|1\rangle$, se vzbudi v stanje $|\nu\rangle$. Hkrati se presežek energije prenese v višjo energijo skupnega nihanja ionov. Iona torej sedaj nihata z intenzivnostjo enega fonona. Če pa je prvi ion v stanju $|0\rangle$, ga curek svetlobe s frekvenco $f_{|1\rangle \rightarrow |\nu\rangle} + \Delta f$ ne bo vzbudil in tudi ne bo povečal intenzivnosti skupnega nihanja ionov.

Sedaj s svetlobo frekvence $f_{|0\rangle \rightarrow |1\rangle} - \Delta f$ in dolžine d μs osvetlimo drugi ion. Zaradi manjka energije (drugi ion potrebuje za prehod v stanje $|1\rangle$ frekvenco $f_{|0\rangle \rightarrow |1\rangle}$) mora drugi ion za prehod v stanje $|1\rangle$ energijo odvzeti skupnemu nihanju ionov. To je mogoče le, če je bil na začetku prvi ion v stanju $|1\rangle$ in smo začetno skupno nihanje iz nič fononov dvignili na en fonon. Torej zavzame drugi ion vrednost $|1\rangle$ le, če je bil prej v stanju $|0\rangle$ in je bil prvi ion v stanju $|1\rangle$. V nasprotnem primeru ostane v stanju $|0\rangle$.

Sedaj drugi ion za d μs osvetlimo s svetlobo frekvence $f_{|1\rangle \rightarrow |\nu\rangle} - \Delta f$. Zopet mora zaradi manjka energije drugi ion za prehod v stanje $|\nu\rangle$ energijo odvzeti skupnemu nihanju ionov. To je mogoče le, če je bil na začetku prvi ion v stanju $|1\rangle$ in smo začetno skupno nihanje iz nič fononov dvignili na en fonon. Torej zavzame drugi ion vrednost $|\nu\rangle$ le, če je bil prej v stanju $|1\rangle$ in je bil prvi ion v stanju $|1\rangle$.

Za izvedbo kontroliranih NE-vrat je še pomembno, da je višje energijsko stanje $|\nu\rangle$ precej manj stabilno od stanj $|0\rangle$ in $|1\rangle$ in da relativno hitro spontano preide v stanje $|0\rangle$.

Po celotnem opisanem postopku imamo naslednje bitne preslikave:

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle, \\ |10\rangle &\rightarrow |\nu 1\rangle \rightarrow |01\rangle, \\ |01\rangle &\rightarrow |01\rangle, \\ |11\rangle &\rightarrow |\nu 0\rangle \rightarrow |00\rangle. \end{aligned} \tag{7.4}$$

Rezultat kontroliranih NE-vrat je zapisan v drugem kvantnem bitu, stanja vhodnih kvantnih bitov pa so se med izračunom uničila. Poudarimo, da je prikazan zgled močno poenostavljen in da ne odraža v celoti uspešnosti in zapletenosti dejanskih fizikalnih poskusov.

7.2 Druge izvedbe kvantnih računalnikov

Ionske pasti omogočajo relativno preprosto nadzorovanje in meritve kvantnih bitov, žal pa so omejene na relativno majhno število ionov. Leta 2011 so v Avstriji v strogo nadzorovanih laboratorijskih razmerah uspeli vzpostaviti superpozicijo stanj 14 kalcijevih ionov [Mon2011]. Pri večjem številu ionov se le-ti pričnejo izrivali iz ravne vrste in se razvrstijo v cikcakasto zaporedje. Coulombove sile med ioni namreč narastejo do velikosti, ko zunanje električno polje elektrod A in B na sliki 7.1 ne uspe prisiliti ionov v ravno vrsto. Struktura postane nestabilna, upravljanje posameznih ionov pa težavno. Omenjeno težavo skušajo rešiti z zaporednimi vezavami več ionskih pasti, vendar je omenjena tehnologija šele v povojuh. Intenzivno raziskujejo tudi optične mreže (angl. *optical lattice*), v katerih je možno ione razvrstiti v dvodimenzionalne strukture [Lew2012].

Dokaj uspešne so izvedbe kvantnih računalnikov, ki temeljijo na magnetni resonanci. Leta 2001 so strokovnjaki podjetja IBM s pomočjo magnetne resonance zgradili kvantni računalnik s sedmimi kvantnimi biti in na njem uspešno izvedli Shorov algoritem, ki je uspešno faktoriziral število 15 [Van2001, Mon2016]. Prav tako so na kvantnem računalniku, ki temelji na magnetni resonanci, implementirali Gorverjev iskalni algoritem v bazi z osmimi elementi [Jon1998]. Harrow, Hassidim in Lloyd so leta 2008 objavili kvantni algoritem za reševanje sistema linearnih enačb z računsko zahtevnostjo $\mathcal{O}(\log N)$ [Har2009, Mon2016]. Algoritem je bil s primerom dveh enačb in dveh neznank uspešno preizkušen na kvantnem računalniku z magnetno resonanco [Mon2016].

Ker je tehnologija za izdelavo silicijevih vezij izpopolnjena, so zanimive tudi kvantne pike, kjer je kvantna informacija shranjena v spinu elektronov, ki so ujeti v kvantne pike. Teoretično lahko tehnologijo kvantnih pik razširimo in jo uspešno združimo s sodobnimi klasičnimi procesorji. Intenzivno jo razvijamo zaradi številnih tehnoloških ciljev izven kvantnega računalništva. Žal večjega preboja na področju kvantnega računalništva s kvantnimi pikami še ni bilo, so pa z njimi že zgradili kvantna kontrolirana NE-vrata [Los1998].

7.2.1 Procesorji podjetja Dwave in adiabatno kvantno računanje

Kanadsko podjetje Dwave je leta 2007 predstavilo kvantni procesor, ki naj bi omogočal adiabatno kvantno računanje (angl. *adiabatic quantum computation*) [Das2005]. Adiabatno kvantno računanje je posebna oblika kvantnega ohlajanja (angl. *quantum annealing*), računske optimizacije, ki je podobna tehniki simuliranega ohlajanja (angl. *simulated annealing*). Njen cilj je poiskati globalni optimum (minimum ali maksimum) podane stroškovne oziroma energijske funkcije. Iskalni prostor je navadno zelo velik, stroškovna funkcija pa ima navadno več lokalnih optimumov, zato je iskanje globalnega optimuma težavno. Simulirano ohlajanje izkorišča naključnost skokov po iskalnem prostoru z namenom globalne optimizacije dane kriterijske funkcije in dovoljuje, da se postopek iskanja z naključnim skokom odmakne od trenutno optimalne pozicije, četudi se ob tem oddalji od trenutno znanega optimuma. Na začetku so naključni skoki veliki in omogočajo relativno velike premike po iskalnem prostoru, s časom pa se skoki postopoma krajšajo (postopek simulira ohlajanje in z njim povezano izgubo energije). Simulirano ohlajanje se navadno kombinira z gradientnimi optimizacijami, ki v dani lokalni okolici iskalnega prostora vedno poiščejo lokalni optimum stroškovne funkcije.

Kvantno ohlajanje izkorišča kvantno tuneljenje [Aar2013, Das2005], eno izmed čudes kvantne mehanike. V klasični fiziki delec s premajhno energijo ne more preko energijske ovire, zato je njegova pozicija vedno omejena na prostor pred oviro. V kvantni mehaniki pa lahko tudi delec s premajhno energijo preide preko ovire, le verjetnost za to je zelo majhna. Kvantno tuneljenje med drugim razloži, zakaj sonce sveti. Atomi vodika v soncu nimajo

dovolj energije, da bi se zlili v atom helija. Po pravilih klasične fizike sonce torej ne bi smelo svetiti. Kvantna mehanika pa pravi, da je sicer zlivanje vodika v helij v razmerah na soncu malo verjetno, ni pa nemogoče. In Sonce sveti. Kvantno ohlajanje se od klasičnega (simuliranega) ohlajanja razlikuje v tem, da lahko s pomočjo kvantnega tuneljenja v iskalnem prostoru preskočimo ovire, ki jih drugače ne moremo.

Kot rečeno, je adiabatno kvantno računanje posebna oblika kvantnega ohlajanja. Adiabatni sistemi so sistemi, ki skozi časovne spremembe ohranijo (ne pridobijo in ne izgubijo) energijo. Adiabatno kvantno računanje izkorišča adiabatni teorem, ki pravi, da bo kvantni sistem pri dovolj počasnih spremembah ohranil svoje ravnovesno stanje energijskega minimuma.

Dana naj bo energijska funkcija, ki jo želimo optimizirati. Predpostavimo, da je energijska funkcija relativno kompleksna, torej je kompleksen tudi Hamiltonov operator, ki opredeljuje celotno energijo sistema (glejte poglavje 3). Adiabatno kvantno računanje izhaja iz zamisli, da za dani optimizacijski problem najprej poiščemo preprost Hamiltonov operator, ki dovolj dobro aproksimira kompleksnejšega in ima relativno preprosto rešitev. Poiščemo njegovo rešitev in sistem, ki ga opisuje preprostejši Hamiltonov operator, postavimo v izračunani energijski minimum. Nato skušamo s pomočjo adiabatnih sprememb preprostejši sistem postopoma preoblikovati v kompleksnejšega. Če nam to uspe, nam adiabatni teorem zagotavlja, da je sistem še vedno v energijskem minimumu, vendar je sedaj ta minimum rešitev kompleksnejšega Hamiltonovega operatorja in torej optimizira originalno kompleksno energijsko funkcijo [Das2005].

Omenjenega računskega pristopa ne moremo uporabiti za rešitev vseh računskih problemov. Z njim lahko na primer rešimo problem izpolnjevanja Boolove enačbe, ki smo ga opisali v zgledu 1.1. Omenjeni problem spada med NP-polne probleme, kar namiguje, da lahko z adiabatnim kvantnim računanjem rešimo vse NP-polne probleme. Dodajmo, da v skladu s kvantnih ohlajanjem procesorji podjetja Dwave skušajo rešiti probleme optimalno in ne jamčijo njegove eksaktne rešitve.

Od leta 2007 dalje podjetje Dwave vztrajno povečuje svoje kvantne procesorje. Razvoj je potekal od 16-bitnega prototipa po imenu Orion, preko 128-bitnega procesorja D-Wave One in 512-bitnega procesorja D-Wave Two do 1152-bitnega procesorja D-Wave 2X, ki ga je podjetje izdelalo leta 2015 (D-Wave 2X naj bi imel 2048 bitov, vendar je polovica bitov onemogočena) [Bro2014]. Vendar so omenjene kvantne procesorje zelo kritizirali, predvsem v akademskih krogih in pri ostalih razvijalcih kvantnih računalnikov, saj z njimi ni mogoče preprosto demonstrirati osnovnih pojavov kvantne mehanike, kot je na primer superpozicija več klasičnih stanj.

Kvantne procesorje D-Wave je kupilo tudi nekaj odmevnih podjetij in ustanov, med njimi Google, Lockheed Martin in NASA. Leta 2012 je skupina strokovnjakov s Harvarda s pomočjo procesorja D-Wave One rešila problem oviganja aminokislin, ki so ga opisali z 81 kvantnimi biti [Ort2012]. Po besedah strokovnjakov je procesor D-Wave One deloval, a je

pravilno rešitev vrnil le v trinajstih od 10000 simulacijskih tekov. Leta 2011 je bil v reviji Nature objavljen neodvisen članek, ki je dokazal, da vsebujejo procesorji D-Wave vsaj nekaj kvantnih lastnosti [Jon2011], leta 2013 pa so strokovnjaki londonskega Inštituta za fiziko (angl. *Institute of Physics, London*) objavili posreden dokaz za kvantno prepletanje v procesorju D-wave [Aro2013]. Istega leta so raziskovalci z ETH v Zürichu objavili študijo, v kateri so pokazali, da je lahko klasičen računalnik, na katerem teče klasičen algoritem simuliranega ohlajanja, petnajstkrat hitrejši od 128-bitnega procesorja D-Wave One [Aar2013b]. Leta 2014 je bila v reviji Science objavljena neodvisna študija, ki ni našla dokazov, da bi procesorji podjetja Dwave omogočali kvantne pohitritve izračunov [Cho2014].

7.3 Kvantna omrežja

Z uporabo optičnega voda so v ZDA po protokolu BB84 uspešno izmenjali ključa na razdalji 140 km [Dix2008]. Pri tem so dosegli hitrosti 1,02 Mbit/s na razdalji 20 km in 10,1 kbit/s na razdalji 100 km. Evropski raziskovalci so s protokolom BB84 in brezzično lasersko komunikacijo izmenjali skriti ključ med dvema Kanarskima otokoma La Palma in Tenerife, ki sta oddaljena 144 km [Urs2007]. Septembra 2014 so švicarski in ameriški raziskovalci opravili eksperiment, v katerem so s pomočjo kvantne teleportacije preko standardnega telekomunikacijskega optičnega voda v popolnosti prenesli foton na razdalji 25 km [Bus2014]. Ker stanje kvantnega bita opišemo z dvema kompleksnima številoma (amplitudama verjetnosti), lahko potrebujemo za natančen zapis enega samega kvantnega bita neskončno mnogo klasičnih bitov (kadar, na primer, amplitudi verjetnosti sestavljajo iracionalna števila). Teoretično je izmenjava enega samega kvantnega bita torej enakovredna izmenjavi poljubno velikega števila klasičnih bitov.

Trenutno so najbolj znana raziskovalna kvantna omrežja v Avstriji, ZDA, na Japonskem, na Kitajskem in v Švici [Chi2002, Pee2009]. Kvantna izmenjava ključev je prodrla tudi v komercialne vode in raziskujejo in tržijo jo številna podjetja, med njimi tudi IBM, Toshiba, Hewlett Packard, NEC, Mitsubishi, id Quantique in SeQureNet.

Omenimo še, da so tudi kvantna omrežja podvržena izgubam in napakam zaradi šuma in izgub. Vendar v primeru kvantnih omrežij za nazor izgub ne moremo uporabiti redundancy (ponavljanja) informacij, saj to preprečuje teorem o neizvedljivosti kloniranja (podpoglavlje 3.6). Zato raziskovalci intenzivno razvijajo posebne kode, ki omogočajo odkrivanje in popravljanje izgub oziroma napak v prejetih kvantnih bitih. Več o teh kodah si lahko preberete v [Nie2000].

8 Viri in Literatura

- Aar2013** S. Aaronson: Quantum Computing since Democritus, Cambridge University Press, 2013.
- Aar2013b** S. Aaronson: D-Wave: Truth finally starts to emerge, dostopno na <http://www.scottaaronson.com/blog/?p=1400>, 2013.
- Aro2013** J. Aron: Controversial quantum computer aces entanglement tests, New Scientist, 2013.
- Abr1998** D. S. Abrams, S. Lloyd: Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems, Phys.Rev.Lett. 81, 3992-3995, 1998.
- AlF2013** N. AlFardan, D. Bernstein, K. Paterson, B. Poettering, J. Schuldt: On the Security of RC4 in TLS, Royal Holloway University of London, dostopno na <http://www.isg.rhul.ac.uk/tls/>, 2013.
- Bac2013** R. Bach, D. Pope, S.H. Liou, H. Batelaan: Controlled double-slit electron diffraction, New Journal of Physics, 15, 2013.
- Bar2016** D. Barker-Plummer: Turing Machines, The Stanford Encyclopedia of Philosophy, Spring, Edward N. Zalta (ur.), 2016.
- Bay2001** H. C. von Baeyer: In the beginning was the bit, New Scientist magazine, 2001.
- Ben1984** C. H. Bennett, G. Brassard: Quantum cryptography: Public key distribution and coin tossing, Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, 175: 8-12, 1984.
- Ben1993** C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, W. K. Wootters, Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels, Phys. Rev. Lett. 70, 1895–1899, 1993.
- Ber2009** D.J. Bernstein, J. Buchmann, E. Dahmen: Post-Quantum Cryptography, Springer, 2009.
- Béz1779** É. Bézout: Théorie générale des équations algébriques. Pariz, Francija, 1779.
- Bir2000** A. Biryukov, A. Shamir, D. Wagner: Real Time Cryptanalysis of A5/1 on a PC. Fast Software Encryption - FSE 2000: 1–18, 2000.

- Bou1997** D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, A. Zeilinger, Experimental Quantum Teleportation, *Nature* 390, 6660, 575-579, 1997.
- Bro2014** V. Brownell: The Future Of Quantum Computing. D-Wave, 2014.
- Bus2014** F. Bussieres, C. Clausen, A. Tiranov, B. Korzh, V. B. Verma, S. W. Nam, F. Marsili, A. Ferrier, P. Goldner, H. Herrmann, C. Silberhorn, W. Sohler, M. Afzelius, N. Gisin: Quantum teleportation from a telecom-wavelength photon to a solid-state quantum memory, *Nature Photonics* 8, 775-778, 2014.
- Cha2003** L. J. Challis: Electron-phonon Interactions in Low-dimensional Structures, Oxford University Press, 2003.
- Che1981:** H. Chernoff: A Note on an Inequality Involving the Normal Distribution, *The Annals of Probability*, 9: 533–535, 1981.
- Chi2002** E.Chip: Building the quantum network, *New Journal of Physics*, 4(1): 46, 2002.
- Cho2014** Adrian Cho: Quantum or not, controversial computer yields no speedup, *Science* 344 (6190): 1330–1331, 2014.
- Cir1995** I. Cirac, P. Zoller: Quantum Computations with Cold Trapped Ions, *Physical Review Letters*. APS. 74 (20):4091–4094, 1995.
- Con2016** K. Conrad: The Miller-Rabin test, dostopno na <http://www.math.uconn.edu/~kconrad/blurbs/ugradnumthy/millerrabin.pdf>, 2016.
- Cre2005** J. D. Cresser, Wave Mechanics Notes for PHYS201, Macquarie University, 2005.
- Cry2016** Crypto++, dostopno na <https://www.cryptopp.com/>, 2016.
- Dae2002** J. Daemen, V. Rijmen: The Design of Rijndael, AES - The Advanced Encryption Standard, Springer-Verlag, 2002.
- Das2005** A. Das, B. K. Chakrabarti: Quantum Annealing and Related Optimization Methods, Springer, 2005.
- Dav2006** J. H. Davies: The Physics of Low-Dimensional Semiconductors: An Introduction Cambridge University Press. 2006.
- Deu1992** D. Deutsch, R. Jozsa: Rapid solutions of problems by quantum computation, *Proceedings of the Royal Society of London A.* 439: 553-558, 1992.

- Dif1976** W. Diffie, M. Hellman: New directions in cryptography, IEEE Transactions on Information Theory, 22(6): 644–654, 1976.
- DiV1997** D. P. DiVincenzo: The Physical Implementation of Quantum Computation, Fortschritte der Physik 48: 771-780, 2000.
- Dix2008** A. R. Dixon, Z. L. Yuan, J. F. Dynes, A. W. Sharpe, A. J. Shields: Gigahertz decoy quantum key distribution with 1 Mbit/s secure key rate, Optics Express 16 (23): 18790-18797, 2008.
- Dwa2016** D-Wave Systems, dostopno na <http://www.dwavesys.com/>, 2016.
- Eke1991** Artur K. Ekert: Quantum cryptography based on Bell's theorem, Phys. Rev. Lett. 67, 661, 1991.
- Eke2008** A. Ekert, P. Hayden H. Inamori: Basic concepts in quantum computation, 2008.
- Ern1842** S. Earnshaw: On the Nature of the Molecular Forces which Regulate the Constitution of the Luminiferous Ether. Trans. Camb. Phil. Soc. 7: 97–112, 1842.
- eST2012** eSTREAM: the ECRYPT Stream Cipher Project, dostopno na <http://www.ecrypt.eu.org/stream/>, 2012.
- Eve1956** H. Everett: Theory of the Universal Wavefunction, Thesis, Princeton University. 1956.
- Eve1956** H. Everett: Relative State Formulation of Quantum Mechanics. Reviews of Modern Physics 29: 454–462, 1957.
- Fay2014** J. Faye: Copenhagen Interpretation of Quantum Mechanics, The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ur.), 2014.
- Fey1982** R. P. Feynman: Simulating Physics with Computers, International Journal of Theoretical Physics, 21: 6/7, 1982.
- Gir2015** D. Giry: BlueKrypt – Cryptographic key length Recommendation, dostopno na <https://www.keylength.com/en/compare/>, 2015.
- Gra1975** J. Grasselli: Osnove teorije števil. Ljubljana: DMFA, 1975.
- Gri2004** D. J. Griffiths: Introduction to Quantum Mechanics, Prentice Hall, 2004.

- Gro1996** L.K. Grover: A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, 212-219, 1996.
- Gur2004** N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz: Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, 3156: 119-132, 2004.
- Har2009** A. W. Harrow, A. Hassidim, S. Lloyd: Quantum algorithm for solving linear systems of equations Phys. Rev. Lett. 15 (103): 150502, 2009.
- Hol2002** M. H. Holzscheiter: Ion-Trap Quantum Computation, Los Alamos Science Number 27:264-283, 2002.
- Inf2016** Information Philosopher: The Information Interpretation of Quantum Mechanics dostopno na <http://www.informationphilosopher.com/introduction/physics/interpretation/>, 2016.
- Jon1998** J. A. Jones, M. Mosca, R. H. Hansen: Implementation of a quantum search algorithm on a quantum computer, Nature 393, 344-346, 1998.
- Jon2011** M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E.M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, G. Rose: Quantum annealing with manufactured spins, Nature 473: 194–198 2011.
- Kat2007** V. J. Katz: The Mathematics of Egypt, Mesopotamia, China, India and Islam : A Sourcebook, Princeton University Press, 2007.
- Koš2009** Tomaž Košir: Linearna algebra za študente praktične matematike, Univerza v Ljubljani, Fakulteta za matematiko in fiziko, 2009.
- Len1993** A. K. Lenstra, H.W. Lenstra: The Development of the Number Field Sieve, Springer, 1993.
- Lew2012** M. Lewenstein, A. Sanpera, V. Ahufinger: Ultracold Atoms in Optical Lattices: Simulating quantum many-body systems, Oxford university press, 2012.
- Los1998** D. Loss and D. P. DiVincenzo: Quantum computation with quantum dots, Phys. Rev. A 57:120, 1998.

- Men1995** A. Menezes, M. Qu, S. Vanstone: Some new key agreement protocols providing mutual implicit authentication, Workshop on Selected Areas in Cryptography (SAC '95), 22-32, 1995.
- Mic2016** Mircosoft: Language-Integrated Quantum Operations: LIQUi|>, dostopno na <https://www.microsoft.com/en-us/research/project/language-integrated-quantum-operations-liqui/>, 2016.
- Mon2011** T. Monz, P. Schindler, J. T. Barreiro, M. Chwalla, D. Nigg, W. A. Coish, M. Harlander, W. Hänsel, M. Hennrich, R. Blatt: 14-Qubit Entanglement: Creation and Coherence. Phys. Rev. Lett. 106, 130506, 2011.
- Mon2016** A. Montanaro: Quantum algorithms: an overview, NPJ Quantum Information, 2:15023: 2016.
- Nes2004** Bart Preneel (koordinator projekta): NESSIE - New European Schemes for Signatures, Integrity, and Encryption, dostopno na <https://www.cosic.esat.kuleuven.be/nessie/>, 2004.
- New1955** M. H. A. Newman, Alan Mathison Turing, Biographical Memoirs of Fellows of the Royal Society of London 1, 253-263, 1955.
- Nie2000** M. A. Nielsen, I. L. Chuang: Quantum Computation and Quantum Information, Cambridge University Press, 2000.
- Nis1999** NIST: Priporočila za uporabo eliptičnih krivulj za potrebe vlade ZDA, dostopno na <http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf>, 1999.
- Ope2016** OpenSSL Software Foundation: OpenSSL, dostopno na <https://www.openssl.org/>, 2016.
- Ort2012** A. P. Ortiz, N. Dickson, M. D. Brook, G. Rose, A. A. Guzik: Finding low-energy conformations of lattice protein models by quantum annealing, Scientific Reports 2: 571, 2012.
- Par1998** C. Paar, J. Pelzl: Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 1998.
- Par2008** S. Pranab: Quantum Algorithm for the Discrete Logarithm Problem - Shor, 1994, Encyclopedia of Algorithms, 1-99, Springer, 2008.

- Pee2009** M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda; W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J.F. Dynes: The SECOQC quantum key distribution network in Vienna, New Journal of Physics, IOP Publishing, 11 (7): 075001, 2009.
- Poh1978** S. Pohlig, M. Hellman: An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance, IEEE Transactions on Information Theory 24: 106–110, 1978.
- Pro2003** J. Proos, C. Zalka: Shor's discrete logarithm quantum algorithm for elliptic curves, Journal Quantum Information & Computation archive, 3 (4): 317-344, 2003.
- Rab1980** M. O. Rabin: Probabilistic algorithm for testing primality, J. Number Theory 12, 128-138, 1980.
- Rij2013** J. Daemen, V. Rijmen: The design of Rijndael: AES-the advanced encryption standard, Springer Science & Business Media, 2013.
- Riv1978** R.L. Rivest, A. Shamir, and L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM* 21.2: 120-126, 1978.
- Sha1949** C. Shannon: Communication Theory of Secrecy Systems, Bell System Technical Journal, 28 (4): 656–715, 1949.
- Sch1926** E. Schrödinger: An Undulatory Theory of the Mechanics of Atoms and Molecules, Physical Review 28 (6): 1049–1070, 1926.
- Sch1994** B. Schneier: Fast Software Encryption, Cambridge Security Workshop Proceedings, Springer-Verlag, 191-204, 1994.
- Sch1996** B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson: The Twofish Encryption Algorithm - A 128-Bit Block Cipher, John Wiley & Sons, 1996.
- Sch1996b** B. Schneier: Applied Cryptography: Protocols, Algorithms, and Source Code in C, Wiley 1996.
- Sch2013** M. Schlosshauer, J. Kofler, A. Zeilinger: A Snapshot of Foundational Attitudes Toward Quantum Mechanics, Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics 44 (3): 222–230. 2013.

- Sho1997** Peter W. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J.Sci.Statist.Comput. 26, 1484, 1997.
- Sho1997b** P. Shor: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing, , 26(5):1484–1509, 1997.
- Sti2002** D. R. Stinson, Cryptography: Theory and Practice, CRC Press, 2002.
- Tou2016** A. de Touzalin, C. Marcus, F. Heijman, I. Cirac, R. Murray, T. Calarco: Quantum Manifesto, A New Era of Technology, Evropska komisija, 2016.
- Tur1950** A. Turing: Computing machinery and intelligence. Mind, 59, 433-460, 1950.
- Urs2007** R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. Ömer, M. Fürst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter, A. Zeilinger: Entanglement-based quantum communication over 144 km, Nature Physics 3: 481 – 486, 2007.
- Van2001** L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang: Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, Nature 414: 883-887, 2001.
- Vri2010** Andreas de Vries: jQuantum - Quantum Computer Simulator, dostopno na <http://jquantum.sourceforge.net/>, 2010.
- Wal2002** S. P. Walborn, M. O. Terra Cunha, S. Pádua, C. H. Monken: Double-Slit Quantum Eraser, Phys. Rev. A 65, 033818, 2002.
- Wal2016** D. Walter: Computational Complexity Theory, The Stanford Encyclopedia of Philosophy, Spring, Edward N. Zalta (ur.), 2016.
- WiB2015** Wikipedia: Block cipher, dostopno na https://en.wikipedia.org/wiki/Block_cipher, 2015.
- WiC2015** Wikipedia: Elliptic-curve cryptography, dostopno na https://en.wikipedia.org/wiki/Elliptic-curve_cryptography, 2015.
- WiD2015** Wikipedia: Diffie–Hellman key exchange , dostopno na https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange, 2015.

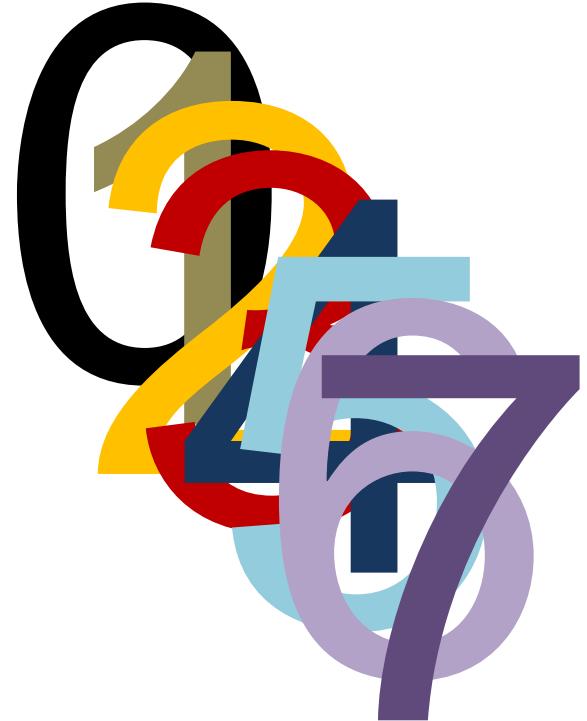
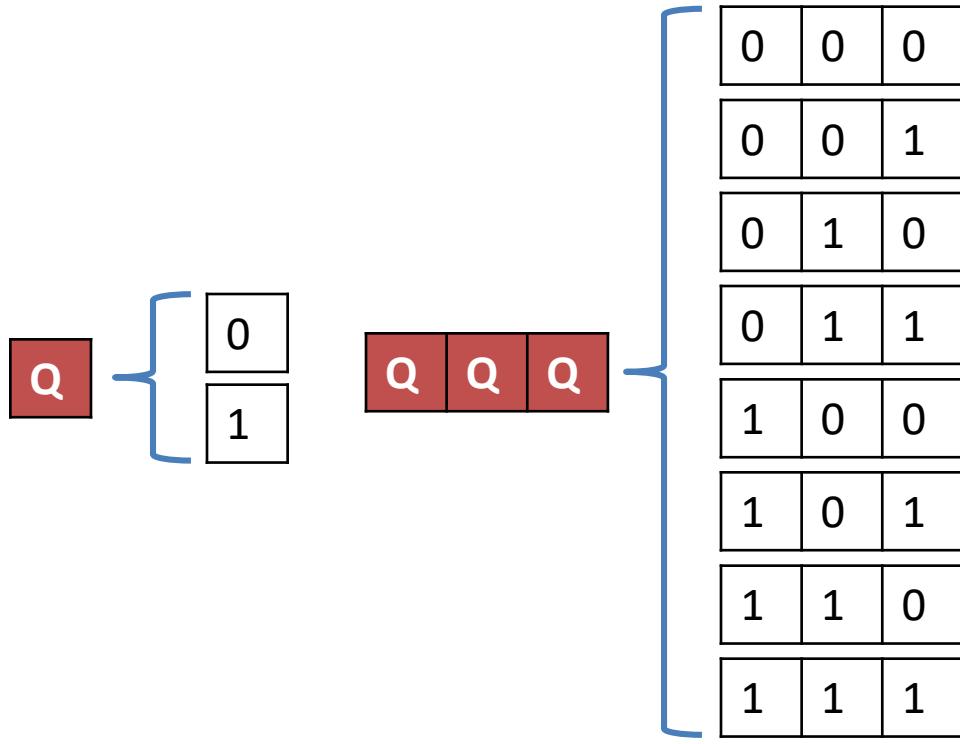
- WiS2015** Wikipedia: Shor's algorithm, dostopno na https://en.wikipedia.org/wiki/Shor%27s_algorithm, 2015.
- Wro2014** G. Wroblewski, L. Culp: Quantum Computing Playground, dostopno na <http://www.quantumplayground.net/#/home>, 2014.
- Yoo2000** K. Yoon-Ho, R. Yu, S.P. Kulik, Y.H. Shih, M. Scully: A Delayed Choice Quantum Eraser. Physical Review Letters 84: 1–5. 2000.
- Xia2016** M. Xiao-song; K. Johannes, A. Zeilinger: Delayed-choice gedanken experiments and their realizations. Rev. Mod. Phys. 88 (1): 015005. 2016.
- Zal1999** C. Zalka: Grover's quantum searching algorithm is optimal, Phys.Rev. A60: 2746-2751, 1999.
- Zwi2013** B. Zwiebach: Dirac's bra and ket notation, 8.05 Quantum Physics II, MIT OpenCourseWare, 2013.



Peter Shor

Kvantni algoritmi

Kvantni register



100 kvantnih bitov lahko hrani več klasičnih bitov informacij kot je atomov v vidnem vesolju!

Kvantni register in funkcije

$$f\left(\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \right) = f^0(f^1(f^2(f^3(f^4(f^5(f^6(f^7(0))))))))$$

Pripravi Razvij Izmeri

Kvantni register in funkcije

$$f\left(\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}\right) = f^0(f^1(f^2(f^3(f^4(f^5(f^6(f^7(0))))))))$$

Pripravi Razvij Izmeri

Kvantno računalništvo

Kvantno stanje z n kvantnimi biti potrebuje 2^n kompleksnih števil za opis stanja:

$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

Cilj kvantnega računalništva je izkoristiti to superpozicijo eksponentno mnogo stanj v izračunih in s tem algoritme, ki imajo eksponentno časovno zahtevnost izračunati v polinomskem času.

Ideja: Amplitude verjetnosti moramo nastaviti tako, da bodo poti, ki vodijo do nepravilnih odgovorov interferirale destruktivno in se s tem izničile, poti, ki vodijo do pravilnih odgovorov pa bodo interferirale konstruktivno.

Kvantna vrata: Controlled-U

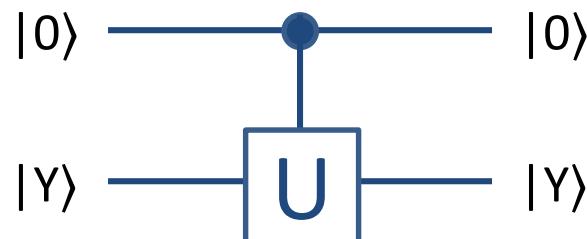
- vrata nad dvema kvantnima bitoma, ki uporabijo unitarno operacijo (matriko) \mathbf{U} nad drugim kvantnim bitom, a samo če je prvi, kontrolni (prvi) kvantni bit postavljen na 1.

$$\text{Controlled}-U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix}$$

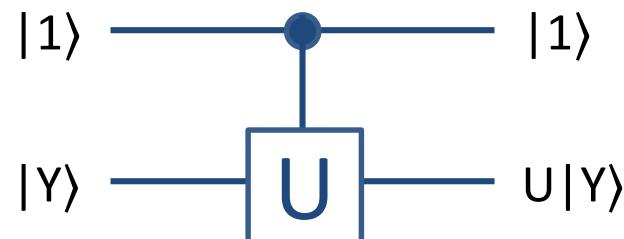
baza prostora

kontrolni bit drugi bit

kontrolni kv. bit postavljen na 0



kontrolni kv. bit postavljen na 1

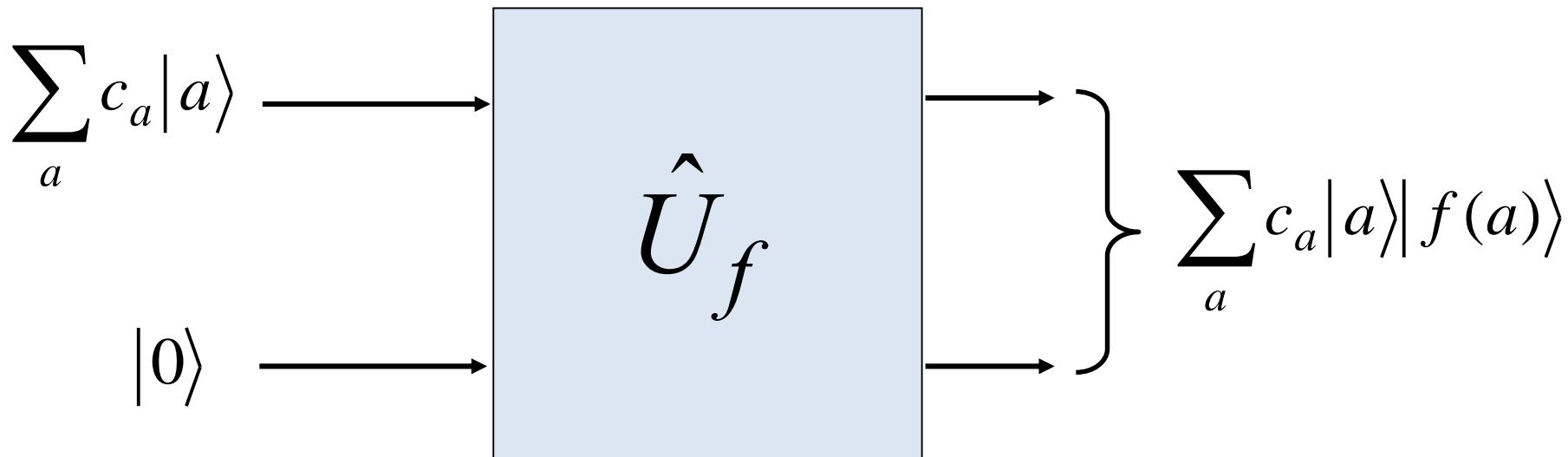


$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$$

Unitarne transformacije

- Za katerokoli Boolovo funkcijo $f: \{0,1\}^n \rightarrow \{0,1\}$ obstaja unitarna transformacija kvantnega stanja
$$|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$$
- Toda večino funkcij f ne moremo implementirati **učinkovito**. Zato nas trenutno zanimajo le tiste funkcije f , ki jih lahko sestavimo iz relativno majhnega števila kvantnih vrat (glede na velikost vhodih podatkov n).

Unitarne transformacije in kvantni registri



- Namesto vhodnega (kontrolnega) in izhodnega (drugega) kv. bita lahko imamo celotne kvantne registre.
- Če je vhodni register v **superpoziciji** več bitnih zaporedij (bitnih nizov) a , je izhodni register v **superpoziciji (kvantni entangulaciji)** vrednosti $f(a)$ (po ena vrednost $f(a)$ za vsako vhodno vrednost a).

Deutsch-ov algoritem



Črna škatla izračuna eno izmed štirih možnih enobitnih funkcij:

Konstantna funkciji:

$$\begin{array}{l} f(0)=0 \\ f(1)=0 \end{array}$$

ali

$$\begin{array}{l} f(0)=1 \\ f(1)=1 \end{array}$$

Uravnoteženi funkciji:

$$\begin{array}{l} f(0)=0 \\ f(1)=1 \end{array}$$

ali

$$\begin{array}{l} f(0)=1 \\ f(1)=0 \end{array}$$

Radi bi vedeli, ali je naša črna škatla konstantna ali uravnotežena. To lahko vedno ugotovimo z dvema izračunoma: $f(0)$ in $f(1)$.

Ali lahko to ugotovimo z enim samim izračunom?

- skonstruirajmo funkcijo: $|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle$
- če $f(0)=f(1)=0$, potem

- $|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|0\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|1\rangle$

$$U_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

baza prostora

$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
--------------	--------------	--------------	--------------

- če $f(0)=f(1)=1$, potem

- $|0\rangle|0\rangle \rightarrow |0\rangle|1\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

$$U_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$ 00\rangle$	$ 01\rangle$	$ 10\rangle$	$ 11\rangle$
--------------	--------------	--------------	--------------

- skonstruirajmo funkcijo: $|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle$

- če $f(0)=0, f(1)=1$, potem

- $|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

$$U_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

baza prostora

$ 00\rangle$	$ 0\rangle 0\rangle$
$ 01\rangle$	$ 0\rangle 1\rangle$
$ 10\rangle$	$ 1\rangle 0\rangle$
$ 11\rangle$	$ 1\rangle 1\rangle$

- če $f(0)=1, f(1)=0$, potem

- $|0\rangle|0\rangle \rightarrow |0\rangle|1\rangle$
- $|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$
- $|1\rangle|0\rangle \rightarrow |1\rangle|0\rangle$
- $|1\rangle|1\rangle \rightarrow |1\rangle|1\rangle$

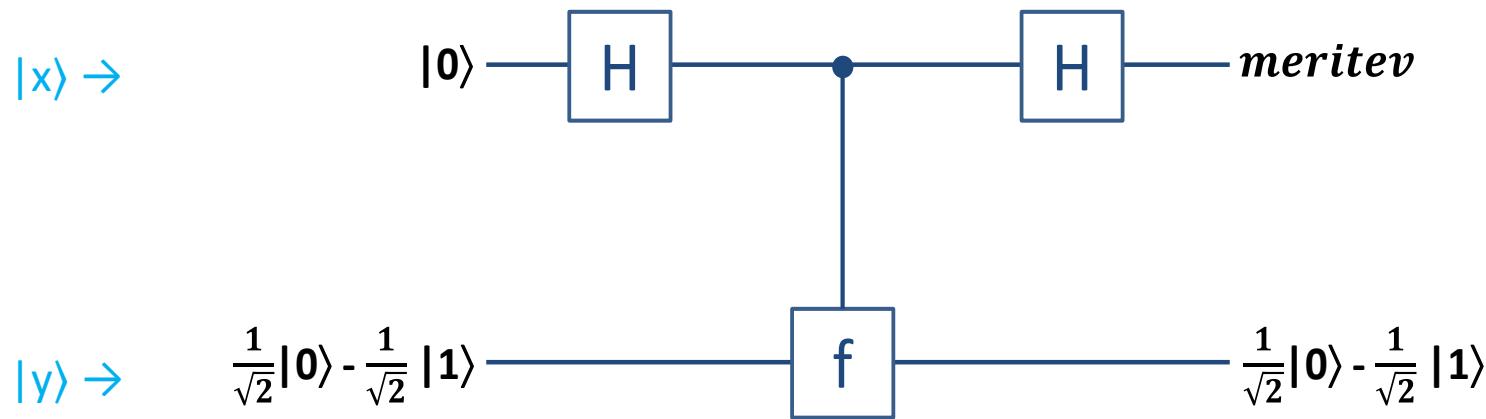
$$U_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$ 00\rangle$	$ 0\rangle 0\rangle$
$ 01\rangle$	$ 0\rangle 1\rangle$
$ 10\rangle$	$ 1\rangle 0\rangle$
$ 11\rangle$	$ 1\rangle 1\rangle$

Deutsch-ov algoritem

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Odgovor z eno samo evalvacijo funkcije f dobimo s pomočjo naslednjega kvantnega algoritma:



- Po prvih Hadamardovih vratih je stanje obeh kv. bitov (če izpustimo normalizacijo s $\sqrt{2}$) [1]:

$$(|0\rangle + |1\rangle) (|0\rangle - |1\rangle)$$

$|x\rangle$ $|y\rangle$

- Po prvih Hadamardovih vratih \mathbf{H} je stanje: $(\underbrace{|0\rangle + |1\rangle}_{|x\rangle})(\underbrace{|0\rangle - |1\rangle}_{|y\rangle})$
- če $f(0)=f(1)=0$, potem

$$U_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix} = \begin{bmatrix} |+1\rangle \\ |-1\rangle \\ |+1\rangle \\ |-1\rangle \end{bmatrix} \rightarrow (\underbrace{|0\rangle + |1\rangle}_{|x\rangle})(\underbrace{|0\rangle - |1\rangle}_{|y\rangle})$$

in po drugih vratih \mathbf{H} imamo $\mathbf{H}x = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = |0\rangle$

- če $f(0)=f(1)=1$, potem

$$U_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix} = \begin{bmatrix} |-1\rangle \\ |+1\rangle \\ |-1\rangle \\ |+1\rangle \end{bmatrix} \rightarrow (\underbrace{|-0\rangle - |1\rangle}_{|x\rangle})(\underbrace{|0\rangle - |1\rangle}_{|y\rangle})$$

in po drugih vratih \mathbf{H} imamo $\mathbf{H}x = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \end{bmatrix} = |0\rangle$

Tenzorski produkt

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

faktorizacija

ZGLEDI:

$$(1|0\rangle + 1|1\rangle)(1|0\rangle - 1|1\rangle) = 1|00\rangle + -1|01\rangle + 1|10\rangle + -1|11\rangle$$

$$(-1|0\rangle - 1|1\rangle)(1|0\rangle - 1|1\rangle) = \underline{\hspace{1cm}}|00\rangle + \underline{\hspace{1cm}}|01\rangle + \underline{\hspace{1cm}}|10\rangle + \underline{\hspace{1cm}}|11\rangle$$

- Po prvih Hadamardovih vratih \mathbf{H} je stanje: $(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$
- če $f(0)=0, f(1)=1$, potem

$$U_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix} = \begin{bmatrix} |+1\rangle \\ |-1\rangle \\ |+1\rangle \\ |+1\rangle \end{bmatrix} \rightarrow (|0\rangle - |1\rangle)(|0\rangle - |1\rangle)$$

$|x\rangle$ $|y\rangle$

in po drugih vratih \mathbf{H} imamo $\mathbf{H}x = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} = |1\rangle$

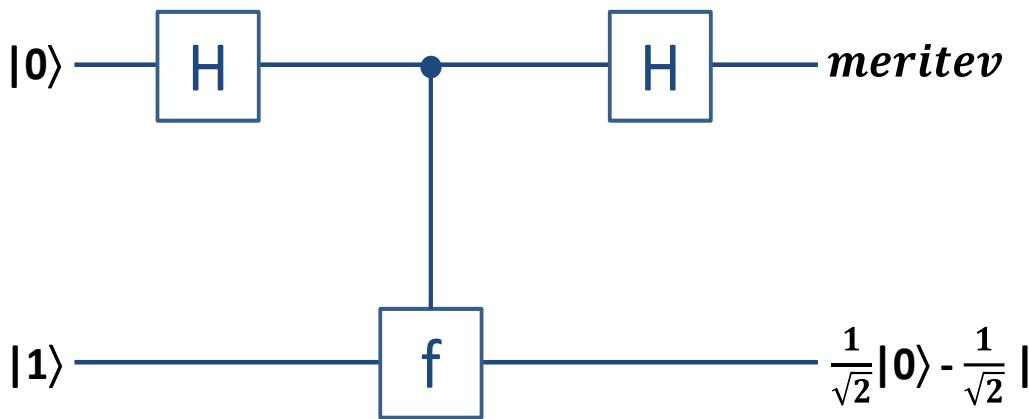
- če $f(0)=1, f(1)=0$, potem

$$U_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix} = \begin{bmatrix} |-1\rangle \\ |+1\rangle \\ |+1\rangle \\ |-1\rangle \end{bmatrix} \rightarrow (-|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$

$|x\rangle$ $|y\rangle$

in po drugih vratih \mathbf{H} imamo $\mathbf{H}x = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ +1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix} = |1\rangle$

Deutsch-ov algoritem (krajše) $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$



- po evalvaciji funkcije f , sta stanji $|1\rangle$:

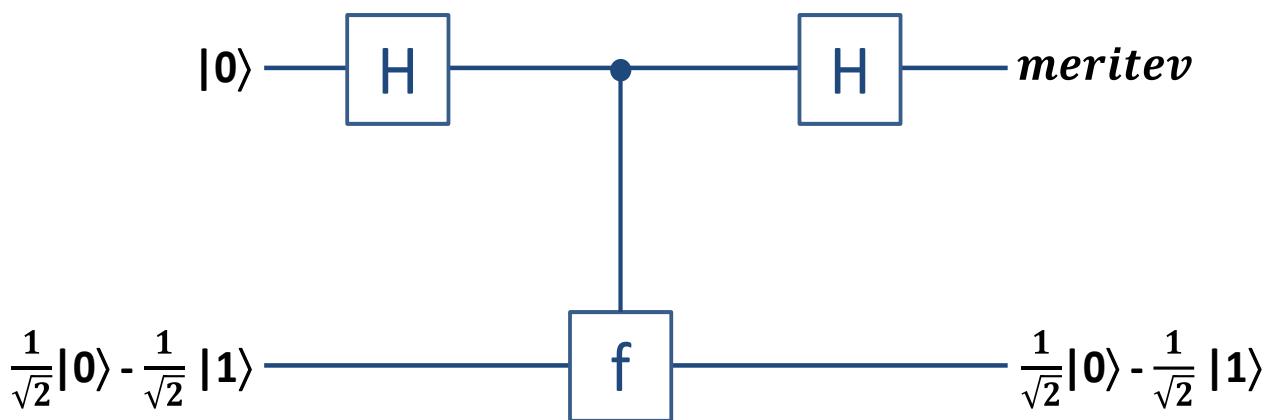
$$|x\rangle (|0\rangle - |1\rangle) \xrightarrow{f} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle) = \\ [(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle] (|0\rangle - |1\rangle)$$

- torej je prvi kv. bit($|x\rangle$) v stanju

$$\pm (|0\rangle + |1\rangle), \text{ če } f(0) = f(1)$$

$$\pm (|0\rangle - |1\rangle), \text{ če } f(0) \neq f(1)$$

Deutsch-ov algoritem (krajše) $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$



- Po drugih Hadamardovih vratih je stanje prvega kv. bita [1]:
 - $|0\rangle$ če je f **konstantna**
 - $|1\rangle$ če je f **uravnotežena**.



Shor-ov algoritem

Shor-ov algoritem faktorizacije

Shor-ov kvantni algoritem za faktorizacijo velikih celih števil je daleč najbolj vpliven in slaven med vsemi kvantnimi algoritmi (1994).

Zaradi njega je močno poskočilo zanimanje in finančno vlaganje v razvoj kvantnih računalnikov saj omogoča učinkovito dekripcijo asimteričnih kodirnikov (RSA in diskretnih logaritmov).

Miljarde evrov so zaščitene s kriptografijo (vsi bančni sistemi, nepremičninski trgi, borze itd.)

Peter Shor je pokazal, kako lahko faktoriziramo velika števila v polinomskem času, za kar je na klasičnem računalniku potreben eksponenten čas. Shorov algoritem lahko faktorizira 2^N krat hitreje, kjer je N bitna velikost ključa.

Aritmetika po modulu: modularni inverz

- Celo število $a \geq 2$ je praštevilo, če je deljivo samo z 1 in z a
- Največji skupni delitelj $d = \gcd(a, b)$ je največje celo število d , ki deli celi števili a in b .
- Celi števili a in b sta **tuji števili** če $\gcd(a, b) = 1$;
- Za tuji celi števili a in n vedno obstaja unikatno število $d \in \{0, \dots, n - 1\}$ tako da velja

$$ad = 1 \text{ mod } n$$

Število d imenujemo **inverz števila a po modulu n** in ga označimo z a^{-1}

Karl Friedrich Gauss (1777-1855)

Shor-ov algoritem: klasičen del

1. Izberi naključno število $a < N$
2. izračunaj $\gcd(a, N)$.
3. Če $d = \gcd(a, N) \neq 1$, potem je d iskani netrivialni faktor N , torej smo končali.
4. V nasprotnem primeru uporabimo kvantno rutino za iskanje periode r funkcije:

$$f(x) = a^x \bmod N$$

r je red števila a v $(\mathbb{Z}_N)^\times$, torej najmanjše celo število, za katerega velja $f(x + r) = f(x)$

5. Če je r liho število, se vrni na korak 1.
6. Če $a^{r/2} \equiv -1 \pmod{N}$, se vrni na korak 1.
7. $\gcd(a^{r/2} \pm 1, N)$ je netrivialni faktor N , torej smo končali.

Shor-ov algoritem: primer

Poskusimo faktorizirati število $N = 15$. Izberimo $a=8$ (8 in 15 sta tuji števili).

Torej imamo $f_{15}(x) = 8^x \text{ mod } 15$

Za $x = 0, 1, 2, \dots$ imamo ciklični vzorec

$$\begin{aligned}f_{15}(0) &= 1, & f_{15}(1) &= 8, & f_{15}(2) &= 4, & f_{15}(3) &= 2, \\f_{15}(4) &= 1, & f_{15}(5) &= 8, & f_{15}(6) &= 4, & \dots\end{aligned}$$

Vidimo, da je vzorec res cikličen $1,8,4,2,1,8,4,2,1,8,4,2\dots$ s periodo $r = 4$.

Izračunamo $d=\gcd(a^{r/2}-1, N) = \gcd(63, 15) = 3$. Drugi faktor (5) lahko najdemo z deljenjem (N/d).

Poskusimo faktorizirati še število $N = 85$. Izberimo $a=31$ (31 in 85 sta tuji števili). Torej imamo $f_{85}(x) = 31^x \text{ mod } 85$, ki za izbrane $x=0,1,2,\dots$ tvori ciklični vzorec $1, 31, 26, 41, 81, 46, 66, 6, 16, 71, 76, 61, 21, 56, 36, 11, 1, 31, \dots$

Perioda $r=16$ in $d=\gcd(a^{r/2}-1, N) = 5$.

Preizkusite še sami razne vrednosti N in a in se prepričajte, da postopek res deluje.

Srce Shor-ovega algoritma je iskanje periode r s pomočjo kvantne funkcije. Ko najdemo r , je faktorizacija N preprosta.

Shor-ov algoritem: dokaz klasičnega dela

Po definiciji periode r imamo $f(r) = a^r \text{ mod } N = 1$. Torej N deli $a^r - 1$. Po koraku 5 imamo takšen a , da je $\gcd(a, N) = 1$ in r sodo število.

Definirajmo $b = a^{r/2} \text{ mod } N$. Torej je b kvadratni koren števila 1 po mod N . Velja $b \neq 1$, saj je po definiciji perioda funkcije $f(x)$ enaka r in ne $r/2$. Korak 6 zagotavlja tudi $b \neq -1$.

Trdimo, da je $d = \gcd(b-1, N)$ netrivialen faktor števila N (torej $d \neq 1$ in $d \neq N$).

1. Ker velja $d < b-1 < N$, velja tudi $d \neq N$
2. Če bi veljalo $d = \gcd(b-1, N) = 1$, potem bi po Bezoutovi enakosti (poimenovani po francoskem matematiku Étiennu Bézoutu) obstajala takšni celi števili u in v , da bi veljalo

$$(b-1)u + Nv = 1$$

Ko pomnožimo obe strani zgornje enačbe z $(b+1)$, dobimo:

$$(b^2 - 1)u + N(b+1)v = b+1$$

Ker N deli $b^2 - 1 = a^r - 1$, bi moral glede na zgornjo enačbo N deliti tudi $(b+1)$, torej bi veljalo $b \text{ mod } N = -1$, kar je v nasprotju s korakom 6.

Torej je $d = \gcd(b-1, N)$ res netrivialen faktor števila N .

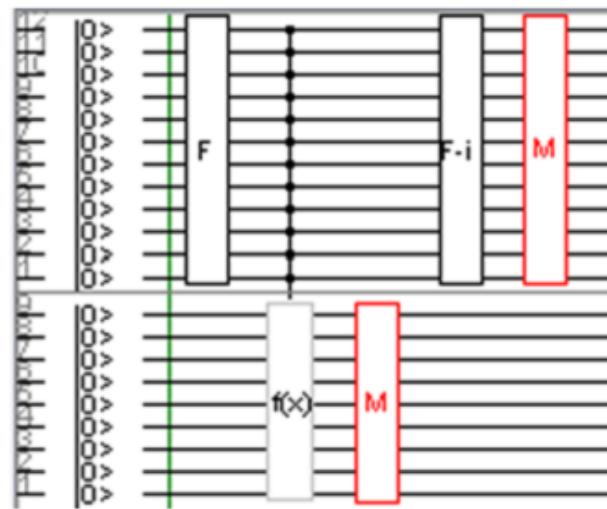
Opomba: Zgornji dokaz temelji na predpostavki, da obstaja takšno število $b = a^{r/2} \text{ mod } N$, da $b \neq -1$ in $b \neq 1$. Obstoj takšnega števila b zagotavlja Teorem kitajskih ostankov, saj je $N = pq$ sestavljen iz praštevil.

Shor-ov algoritem: Kvantni del

Opis: Algoritem poišče periodo funkcije $f(x) = a^x \bmod N$, kjer je a poljubno število, ki je N tuje ($\text{gdc}(a, N)=1$), N pa je sestavljeni število: $N = pq$, kjer sta p in q praštevili.

Potrebna strojna oprema:

- vhodni kvantni register takšne velikosti Q , da je vanj možno hraniti število N^2 .
- izhodni kvantni register takšne velikosti P , da je vanj možno hraniti število N .
- Fourierova kvantna vrata
- Hadamardova kvantna vrata
- kvantno vezje, ki implementira funkcijo $f(x) = a^x \bmod N$ (za vsak a in za vsak N potrebujemo posebno vezje).



vir slike:
<http://jquantum.sourceforge.net/>

Slika: vezje kvantnega dela Shorovega algoritma z vhodnim registrom velikosti 12 qubitov in izhodnim registrom velikosti 9 qubitov.

Shor-ov algoritem: Kvantni del

Koraki algoritma:

1. INICIALIZACIJA:

- vhodni kvantni register je v stanju 0
- izhodni kvantni register v stanju 0

2. SUPERPOZICIJA VHODNEGA REGISTRA:

- preko Hadamardove transformacije ali pa kvantne Fourierove transformacije postavimo vhodni kvantni register v popolno superpozicijo vseh možnih stanj:

$$\sum_x \frac{1}{\sqrt{Q}} |x\rangle$$

- izhodni kvantni register je še vedno v stanju 0

Shor-ov algoritem: Kvantni del

3. APLICIRANJE KVANTNE FUNKCIJE $f(x)$:

- vhodni kvantni register je še vedno v stanju $\sum_x \frac{1}{Q} |x\rangle$
- izhodni kvantni register je v stanju $f\left(\sum_x \frac{1}{Q} |x\rangle\right) = \frac{1}{Q} \sum_x f(|x\rangle)$. Ker ima funkcija periodo r , zavzame samo r različnih vrednosti. Vse so enakovredno zastopane v izhodnem registru.

4. MERITEV IZHODNEGA REGISTRA:

- izhodni kvantni register kolapsira v eno samo opazovano vrednost $y_0 = f(x_0)$ (eno izmed tistih, ki so bile prej v superpoziciji izhodnega registra).
- vhodni register posledično kolapsira v superpozicijo vseh tistih vhodov x_r , za katere velja $y_0 = f(x_r)$. Ker je $f(x)$ periodična funkcija s periodo r , lahko to superpozicijo vhodnega registra zapišemo kot:

$$\frac{1}{Q} \sum_b |x_0 + b \cdot r\rangle$$

kjer je b celo število, ki teče od 0 dokler $x_0 + rb$ ne preseže velikosti vhodnega registra Q .

Shor-ov algoritem: Kvantni del

5. INVERZNA KVANTNA FOURIEROVO TRANSFORMACIJA VHODNEGA REGISTRA:

- vhodni kvantni register transformiramo z inverzno kvantno Fourierovo transformacijo, ki tvori superpozicijo vseh možnih števil v vhodnem registru.
- DEFINICIJA: Kvantna Fourierova transformacija splošno superpozicijo $\sum_{x=0}^Q \alpha_x |x\rangle$ vhodnega registra pretvori v novo superpozicijo $\frac{1}{\sqrt{Q}} \sum_{z=0}^Q \sum_{x=0}^Q \alpha_x e^{\frac{i2\pi zx}{Q}} |z\rangle$, torej
$$\sum_{x=0}^Q \alpha_x |x\rangle \xrightarrow{QFT} \frac{1}{\sqrt{Q}} \sum_{z=0}^Q \sum_{x=0}^Q \alpha_x e^{\frac{i2\pi zx}{Q}} |z\rangle$$
- po tej operaciji je v našem primeru vhodni kvantni register torej v stanju
$$\frac{1}{Q} \sum_z \sum_b e^{\frac{i2\pi z(x_0+rb)}{Q}} |z\rangle$$
saj so bila prej v vhodnem registru samo števila $x = x_0 + rb$ (vsa ostala so imela amplitudo verjetnosti $\alpha_x = 0$).
- izhodni register še vedno vsebuje eno samo vrednost $y_0 = f(x_0)$

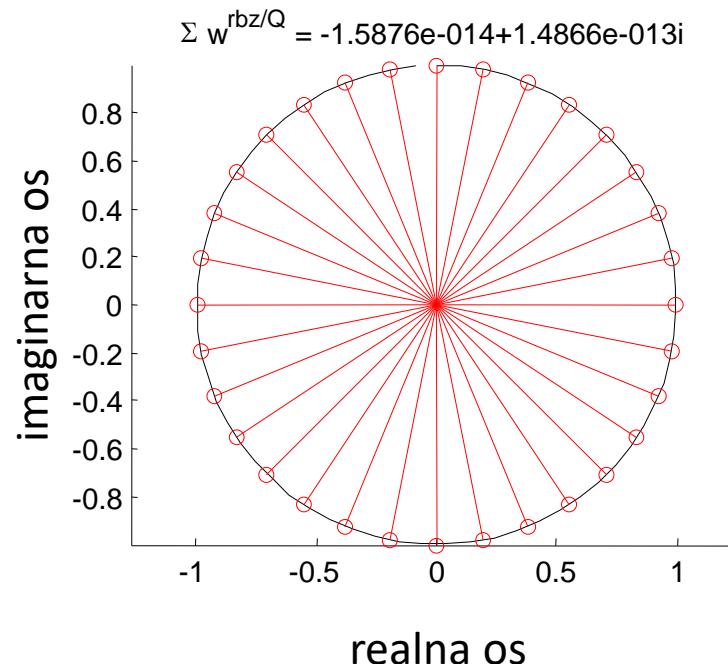
Shor-ov algoritem: Kvantni del

6. MERITEV VHODNEGA REGISTRA:

- izmerimo vhodni register. Velja

$$\frac{1}{Q} \sum_z \sum_b e^{\frac{i2\pi z(x_0+br)}{Q}} |z\rangle = \frac{1}{Q} \sum_z e^{\frac{i2\pi zx_0}{Q}} \sum_b e^{\frac{i2\pi zbr}{Q}} |z\rangle$$

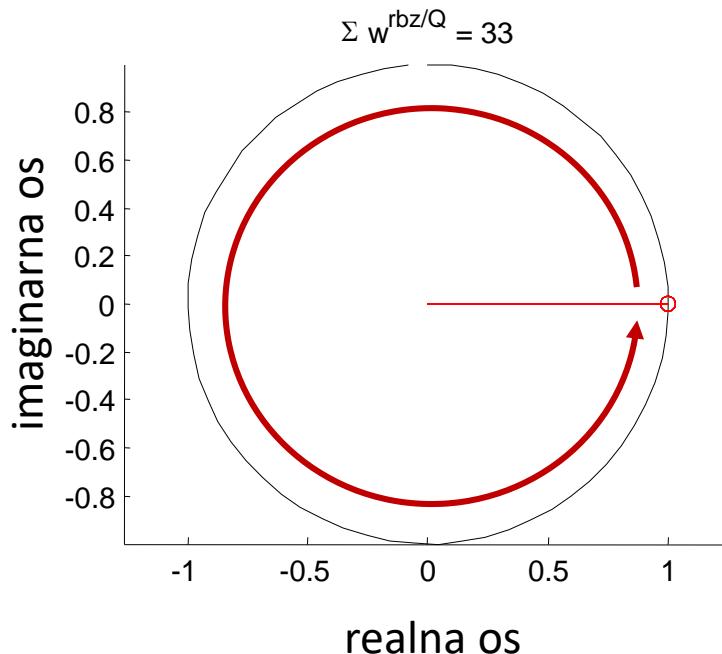
amplitude verjetnosti vseh tistih števil z , za katere velja, da $\frac{zr}{Q}$ ni blizu pozitivnemu celemu številu, bodo v vsoti preko b -ja tvorile 2D enotske vektorje vseh možnih orientacij:



zato se bodo v vsoti preko b -ja izničile in bo njihova vsota enaka ali vsaj blizu 0 (zaradi končnosti vsote, ki izvira iz končnosti vhodnega kvantnega registra ni rečeno, da bo čisto enaka 0).

Shor-ov algoritem: Kvantni del

Amplitude verjetnosti vseh tistih števil z , za katere velja, da je $\frac{zr}{Q}$ zelo blizu pozitivnemu celiemu številu (idealno $\frac{zr}{Q} = \text{celo število } c$), pa bodo v vsoti preko b-ja tvorila konstruktivno superpozicijo, zato se bo njihova verjetnost precej ojačala:



Torej je veliko verjetneje, da bomo ob meritvi v vhodnem registru izmerili takšno število z_0 , da bo veljalo $\frac{z_0 r}{Q} = c$, kjer je c celo število.

Shor-ov algoritem: Kvantni del

7. OCENITEV PERIODE r:

- Z veliko verjetnostjo torej velja $\frac{z_0}{Q} = \frac{c}{r}$ in ker mora biti perioda r manjša od N , velja tudi $r < N$. Pri tem sta c in r celi števili.
- s pomočjo verižnih ulomkov najdemo takšen približek $\frac{c}{r} \approx \frac{z_0}{Q}$, da velja $r < N$. Običajno dobimo več kandidatov za r in preveriti moramo, kateri med njimi izpolnjuje pogoj $f(x) = f(x+r)$.
- če nismo uspešni ponovimo celoten kvantni del Shorovega algoritma

Aritmetika po modulu N & Kvantna vezja

- V kvantnem registru velikosti N je vsota po modulu 2^N ena izmed najbolj splošnih unitarnih operacij (xor je vsota po modulu 2):

$$x \in \{0, 1\}^n \quad \text{and} \quad a \in \{0, 1\}^n$$
$$|x\rangle \rightarrow |(x + a) \bmod 2^n\rangle$$

- Shor je uporabil algoritem zaporednega kvadriranja za implementacijo funkcije $f(x)=a^x \bmod N$
- Implementacija kvantnega vezja za funkcijo $f(x)$ je precej bolj kompleksna od DFT in zahteva tudi več kvantnih vrat (specifično vezje za vsako izbrano osnovo a)

Shor-ov algoritem: nekaj lastnosti

1. Shor-ov algoritem je nedeterminističen (*probabilistic*). Ne najde vedno netrivialnega faktorja števila N (trivialna faktorja števila 21, na primer, sta 1 in 21, 7 in 3 pa sta netrivialna faktorja).

Na primer, faktorizirajmo število 15 z $x = 14$. Potem se bodo v izhodnem registru vrstila naslednja zaporedja funkcije $f(x)$:

$$1, 14, 1, 14, 1, 14, \dots$$

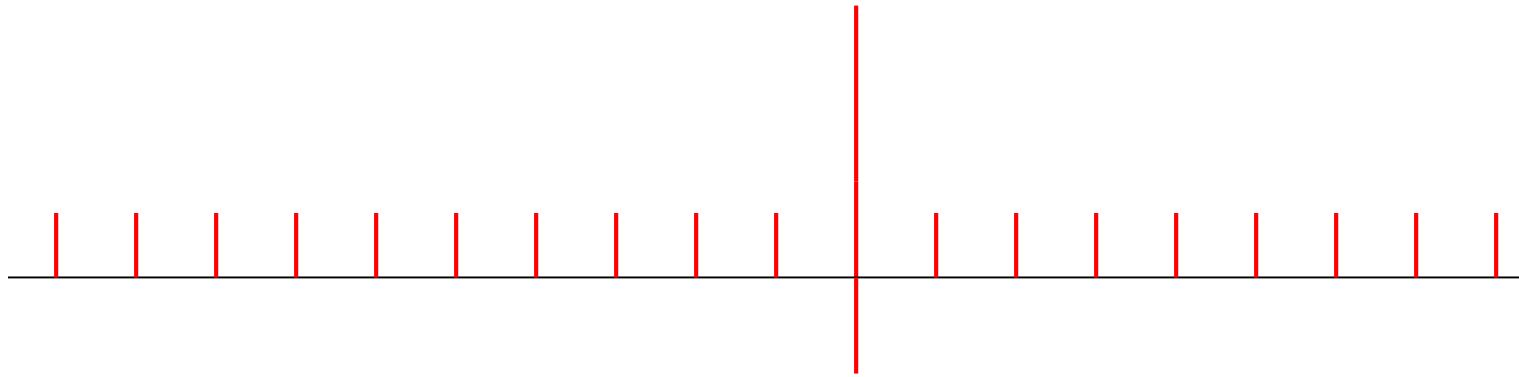
Perioda $r = 2$, torej sta edina faktorja števila 15, ki jih vrne Shorov algoritem $\gcd(14-1, 15) = 1$, in $\gcd(14+1, 15) = 15$, torej trivialna faktorja števila 15.

2. Kvantno vezje Shorovega algoritma je specifično za vsak N in naključno vrednost a v funkciji $f(x) = a^x \bmod N$
3. Časovna zahtevnost Shorovega algoritma je $O((\log N)^3)$
4. Peter Shor je leta 1999 za svoj algoritem in njegov pridonos k teoretičnemu računalništvu prejel [Gödelovo nagrado](#).

Grover-jev kvantni algoritem

Kvantno iskanje po podatkovni bazi.

Najde element v podatkovni bazi v $O(\sqrt{n})$ poizvedbah.



Kakršenkoli klasičen algoritem, determinističen ali ne, potrebuje v povprečju $O(n)$ poizvedb!

Grover-jev kvantni algoritem

1. Postavi kvantni register v stanje superpozicije vseh indeksov:

$$|\omega\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle$$

2. S pomočjo označevalne funkcije $f(x)$ spremenimo predznak amplitude verjetnosti indeksa iskanega elementa. Predzname amplitud verjetnosti indeksov ostalih elementov pustimo nespremenjene.

$$\alpha_x |x\rangle \rightarrow -\alpha_x |x\rangle, \quad \text{če } f(x) = 1,$$

$$\alpha_x |x\rangle \rightarrow \alpha_x |x\rangle, \quad \text{če } f(x) = 0.$$

3. Izračunamo inverz amplitud verjetnosti vseh indeksov okoli njihove povprečne vrednosti $\bar{\alpha}$:

$$\forall x: \alpha_x = 2 \cdot \bar{\alpha} - \alpha_x \quad \bar{\alpha} = \frac{1}{Q} \sum_{x=0}^{Q-1} \alpha_x$$

Koraka 2 in 3 ponovimo $\frac{\pi}{4} \sqrt{\frac{Q}{k}}$ -krat, kjer je k število elementov v bazi, ki so enaki iskanemu elementu. Omenjeno število iteracij je dokazano optimalno in ga ni priporočljivo preseči.

Grover-jev kvantni algoritem - Zgled 1

Zgled: Dana je baza šestnajstih skritih gesel. V njej želimo poiskati geslo, ki dešifrira niz zakodiranih znakov. Elementom baze dodelimo indekse od 0 do 15.

Predpostavimo, da naš zakodirani niz znakov dešifrira samo geslo, ki je v bazi shranjeno v elementu z indeksom 4. Imamo torej naslednjo označevalno funkcijo:

$$f(x) = \begin{cases} 1, & \text{ko } x = 4 \\ 0, & \text{drugače} \end{cases}$$

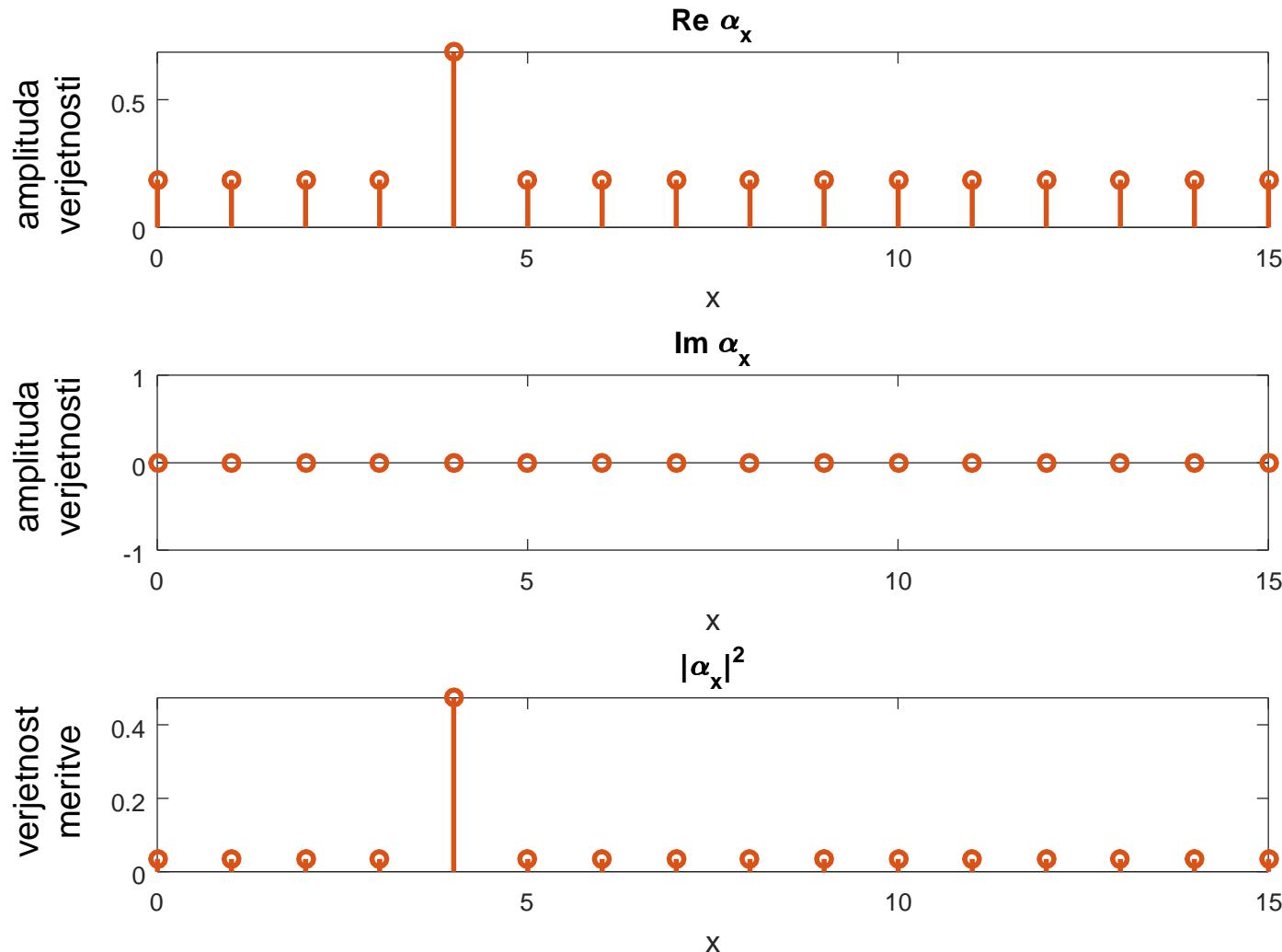
Indekse elementov shranimo v kvantni register z $N=4$ biti. V prvem koraku Groverjevega algoritma postavimo kvantni register v naslednjo superpozicijo stanj:

$$|\omega\rangle = \frac{1}{\sqrt{16}} \sum_{x=0}^{15} 1 |x\rangle$$

Nato iterativno izvajamo drugi in tretji korak Groverjevega algoritma.

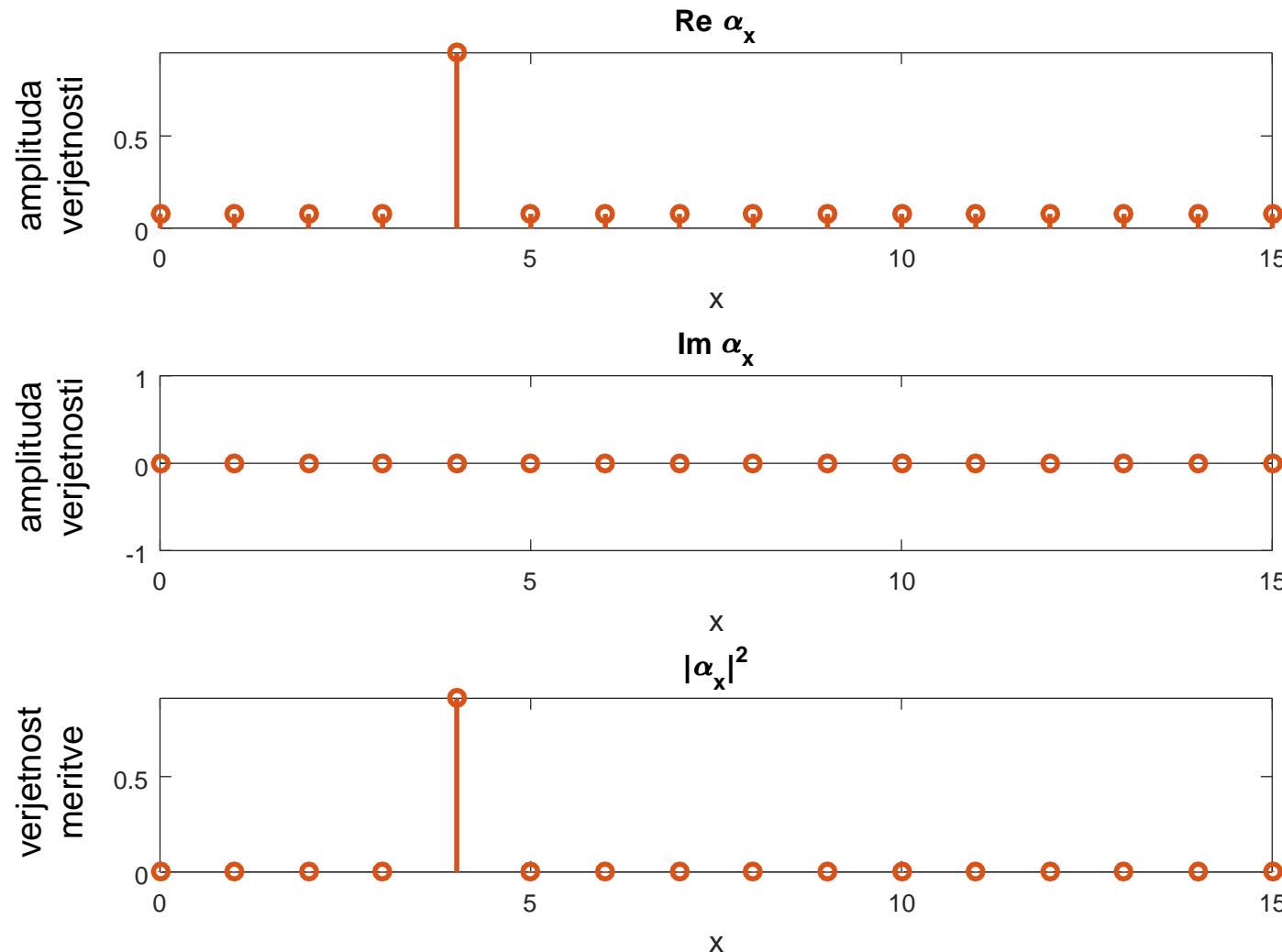
Ker je $\frac{\pi}{4} \sqrt{\left(\frac{Q}{k}\right)} = \frac{\pi}{4} \sqrt{\left(\frac{16}{1}\right)} = \pi = 3,14$, po tretji iteraciji opravimo meritev.

Grover-jev kvantni algoritem - Zgled 1



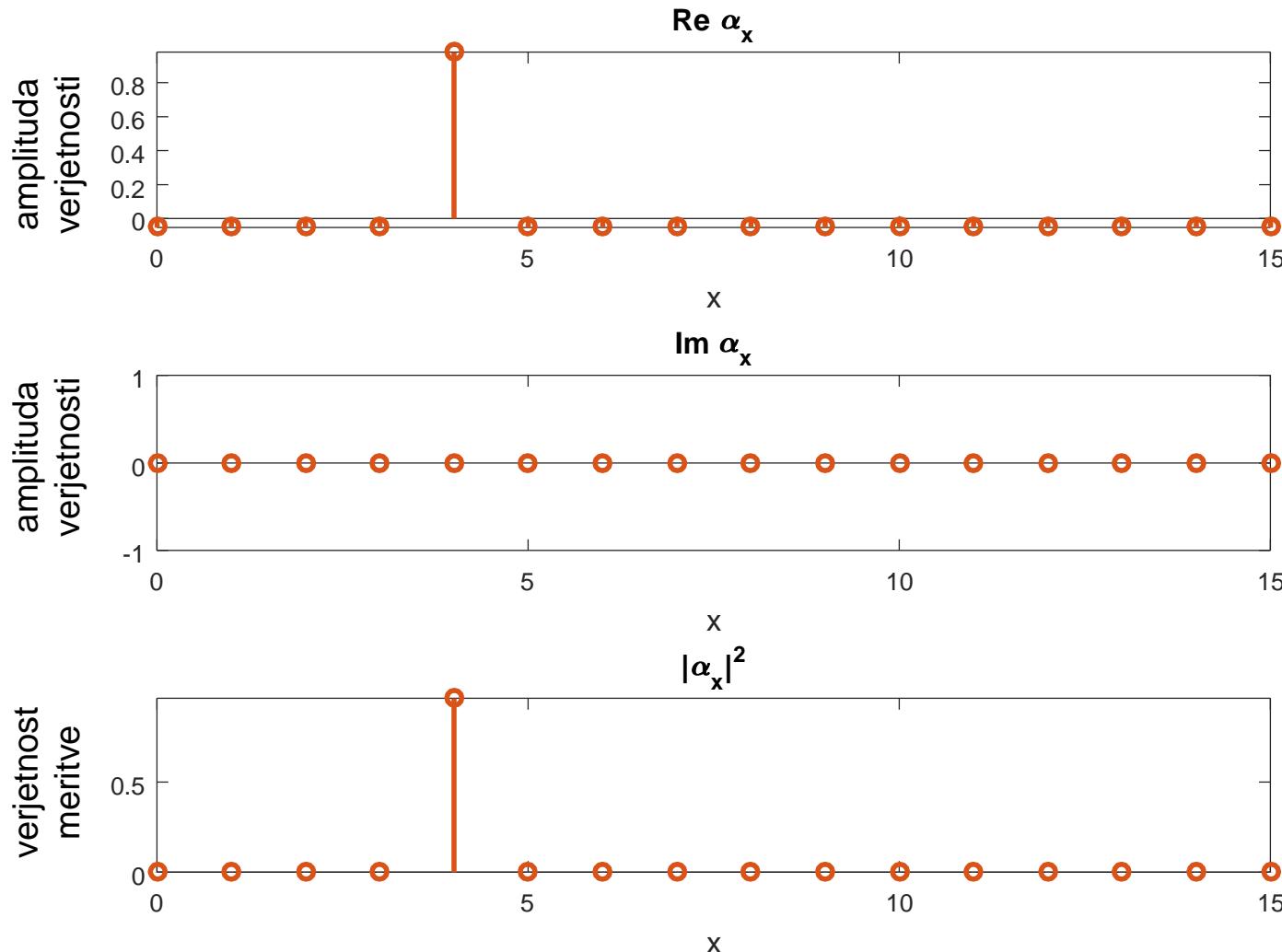
Slika 1: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po prvi iteraciji drugega in tretjega koraka Groverjevega algoritma.

Grover-jev kvantni algoritem - Zgled 1



Slika 2: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po drugi iteraciji drugega in tretjega koraka Groverjevega algoritma.

Grover-jev kvantni algoritem - Zgled 1

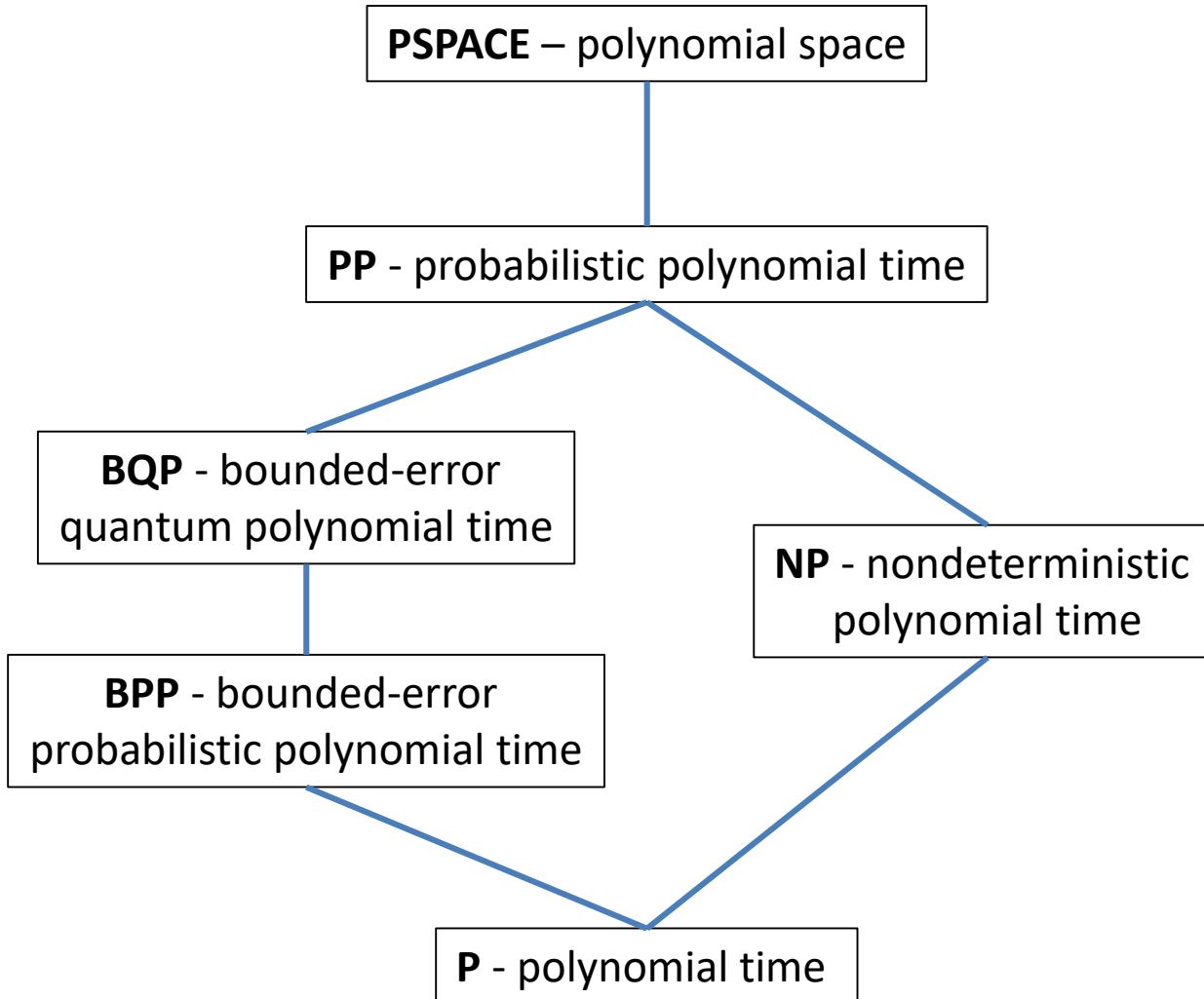


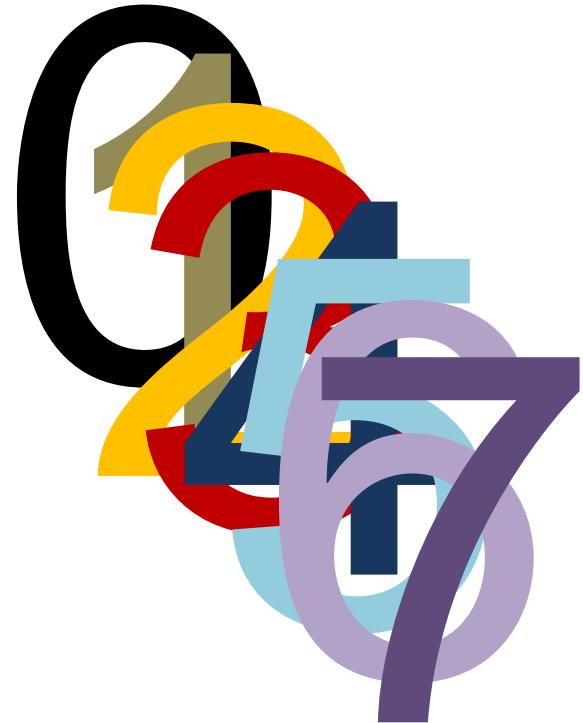
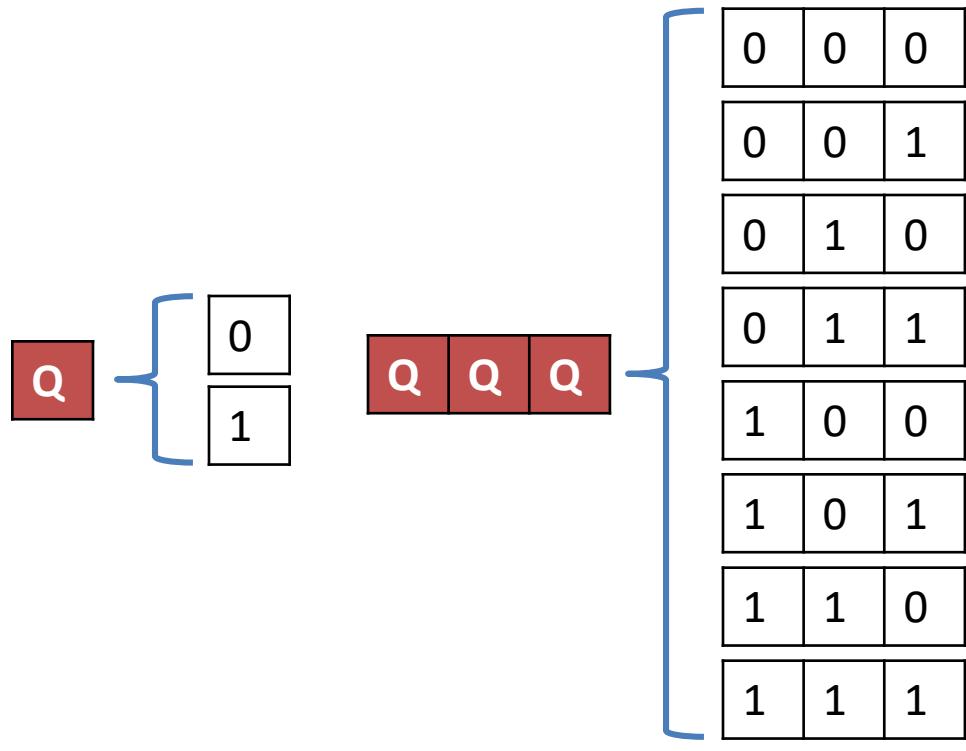
Slika 3: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po tretji iteraciji drugega in tretjega koraka Groverjevega algoritma.

Razredi kvantne računske kompleksnosti

- **BQP** (Bounded-Error Quantum Polynomial-Time) je razred odločitvenih problemov rešljivih v polinomskem času na kvantnem računalniku, pri čemer je verjetnost napake manjša ali enaka $1/3$.
- Analogno z razredom BPP (“bounded error probabilistic polynomial time”) je izbira mejne verjetnosti $1/3$ samo stvar dogovora. Algoritmom lahko izvedemo poljubno mnogokrat in izberemo najpogostejši odgovor. Na ta način se lahko verjetnost pravilnega odgovora dvignemo poljubno blizu 1 (**Chernoff-ova zgornja meja - Chernoff bound**).

Razredi računske kompleksnosti

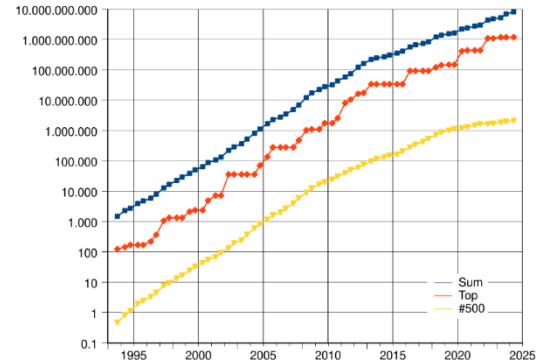




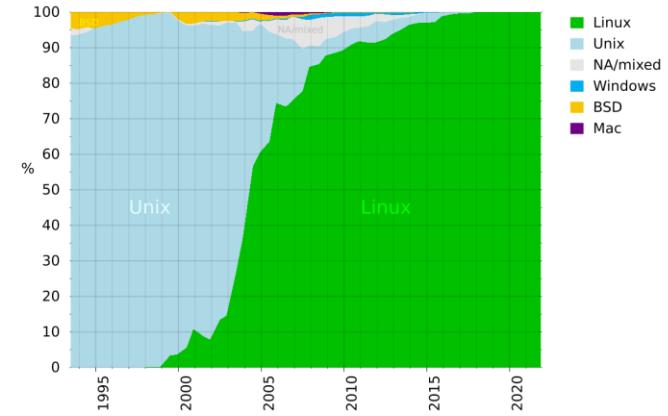
Kvantni računalniki

Najhitrejši superračunalniki na svetu

Rank (previous)	Rmax (PetaFLOPS)	Name	Model	CPU cores	Accelerator (e.g. GPU) cores	Total Cores (CPUs + Accelerators)	Interconnect	Manufacturer	Site country	Year	Operating system
1 NEW	1,742.00 2,746.38	Eli Capitan	HPE Cray EX255a	1,051,392 (43,808 × 24-core Optimized 4th Generation EPYC 24C @1.8 GHz)	9,988,224 (43,808 × 228 AMD Instinct MI300A)	11,039,616	Slingshot-11	HPE	Lawrence Livermore National Laboratory United States	2024	Linux (TOSS)
2 ▼	1,353.00 2,055.72	Frontier	HPE Cray EX235a	614,656 (9,604 × 64-core Optimized 3rd Generation EPYC 64C @2.0 GHz)	8,451,520 (38,416 × 220 AMD Instinct MI250X)	9,066,176	Slingshot-11	HPE	Oak Ridge National Laboratory United States	2022	Linux (HPE Cray OS)
3 ▼	1,012.00 1,980.01	Aurora	HPE Cray EX	1,104,896 (21,248 × 52-core Intel Xeon Max 9470 @2.4 GHz)	8,159,232 (63,744 × 128 Intel Max 1550)	9,264,128	Slingshot-11	HPE	Argonne National Laboratory United States	2023	Linux (SUSE Linux Enterprise Server 15 SP4)
4 ▼	561.20 846.84	Eagle	Microsoft Nv5	172,800 (3,600 × 48-core Intel Xeon Platinum 8480C @2.0 GHz)	1,900,800 (14,400 × 132 Nvidia Hopper H100)	2,073,600	NVIDIA Infiniband NDR	Microsoft	Microsoft United States	2023	Linux (Ubuntu 22.04)
5 NEW	477.90 606.97	HPC6	HPE Cray EX235a	213,120 (3,330 × 64-core Optimized 3rd Generation EPYC 64C @2.0 GHz)	2,930,400 (13,320 × 220 AMD Instinct MI250X)	3,143,520	Slingshot-11	HPE	Eni S.p.A European Union, Ferrara Erbognone, Italy	2024	Linux (RHEL 8.9)



<https://en.wikipedia.org/wiki/TOP500#/media/File:Supercomputers-history.svg>



<https://en.wikipedia.org/wiki/TOP500>

Najhitrejši kvantni računalniki

- **IBM** (ZDA): QPE Condor, 1121 **suprevodnih** kvantnih bitov. Modularni kvantni računalnik (Quantum System Two), zasnovan za kvantno usmerjeno superračunalništvo. **Qiskit** & quantum serverless computing. Načrti za sistem s 100.000 kvantimi biti in razvoj logičnih kvantnih bitov s korekcijo napak.
- **Intel** (ZDA): 12-bitni Tunnel Falls chip /49- bitni Tangle Lake, kvantni biti na osnovi **spina v siliciju**,
- **Google's Quantum AI** : Fokusira se na razvoj kvantnih algoritmov za strojno učenje in umetno inteligenco. Procesor Sycamore temelji na **suprevodnosti** (trdijo, da dosega kvantno nadvlogo). Prizadevajo si za izdelavo kvantnega računalnika z milijonom kvantnih bitov. Podpirajo neodvisnost SW od HW.
- **Quantinuum** (VB): „full-stack **trapped-ion** quantum computer“, kvantna programska oprema TKET (<https://docs.quantinuum.com/tket/>), podpirajo neodvisnost SW od HW.
- **Rigetti** (ZDA): **Superprevodna vezja**, ohlajena skoraj do absolutne ničle, celovite rešitve (full-stack) in kvantno-klasična integracija.
- **IonQ** (ZDA): kvantni biti so **atomi**, ujeti in manipulirani z elektromagnetnimi polji. Stabilni sistemi, natančnost manipulacij, težave s skalabilnostjo.
- **Xanadu** (Kanada): računanje s **fotoni**, težave s nadzorom napak
- **Atom computing** (ZDA): arrays of **optically-trapped neutral atoms**, >1,000 kvantnih bitov

Najhitrejši kvantni računalniki

- **D-Wave** (Kanada): kvantno žarjenje/ohlajanje (quantum annealing), 5000 kvantnih bitov, posebna oblika kvantnega računalništva, ki se razlikuje od univerzalnih kvantnih računalnikov in se osredotoča na specifične izzive optimizacije (iskanje najnižjega energijskega potenciala kvatnih bitov...)
- **Amazon** (ZDA): dostop do različnih kvantnih računalnikov (D-Wave, IonQ and Rigetti). Ponuja hibridne kvantno-klasične računalniške vire in simulator kvantnih vezij.
- **Microsoft's Azure Quantum platform** (ZDA): dostop do kvantnih računalnikov (IonQ & Quantinuum). Bogat nabor orodij (Q#, QDK). Raziskujejo topološke kvantne bite (topological qubits), ki so potencialno bolj stabilna oblika kvantnega bita.

Najhitrejši kvantni računalniki

Manufacturer	Name/ codename designation	Architecture	Layout	Fidelity (%)	Qubits (physical)	Release date	Quantum volume
Atom Computing	N/A	Neutral atoms in optical lattices			1180 ^{[6][7]}	October 2023	
IBM	IBM Condor ^{[16][6]}	Superconducting	N/A	N/A	1121 ^[15]	December 2023	
CAS	Xiaohong ^[67]	Superconducting	N/A	N/A	504 ^[67]	2024	
IBM	IBM Osprey ^{[6][7]}	Superconducting	N/A	N/A	433 ^[15]	November 2022	
Xanadu	Borealis ^[65]	Photonics (Continuous-variable)	N/A	N/A	216 ^[65]	2022 ^[65]	
M Squared Lasers	Maxwell	Neutral atoms in optical lattices		99.5 (3-qubit gate), 99.1 (4-qubit gate) ^[32]	200 ^[33]	November 2022	
IBM	IBM Heron R2 ^[17]	Superconducting	Heavy hex	96.5 (2 qubits)	156	November 2024	
IBM	IBM Heron ^{[16][6]}	Superconducting	N/A	N/A	133	December 2023	
IBM	IBM Eagle	Superconducting	N/A	N/A	127 ^[15]	November 2021	
Atom Computing	Phoenix	Neutral atoms in optical lattices			100 ^[5]	August 10, 2021	
Rigetti	Ankaa-2	Superconducting transmon	N/A	98 (Two-qubit gates)	84 ^[54]	December 20, 2023	

https://en.wikipedia.org/wiki/List_of_quantum_processors

Najhitrejši kvantni računalniki

Annealing quantum processors [edit]

These QPUs are based on quantum annealing, not to be confused with digital annealing.^[68]

Manufacturer	Name/Codename /Designation	Architecture	Layout	Fidelity (%)	Qubits	Release date
D-Wave	D-Wave One (Rainier)	Superconducting	$C_4 = \text{Chimera}(4,4,4)^{[69]} = 4 \times 4 K_{4,4}$	N/A	128	May 11, 2011
D-Wave	D-Wave Two	Superconducting	$C_8 = \text{Chimera}(8,8,4)^{[69]} = 8 \times 8 K_{4,4}$	N/A	512	2013
D-Wave	D-Wave 2X	Superconducting	$C_{12} = \text{Chimera}(12,12,4)^{[69]} = 12 \times 12 K_{4,4}$	N/A	1152	2015
D-Wave	D-Wave 2000Q	Superconducting	$C_{16} = \text{Chimera}(16,16,4)^{[69]} = 16 \times 16 K_{4,4}$	N/A	2048	2017
D-Wave	D-Wave Advantage	Superconducting	Pegasus $P_{16}^{[70]}$	N/A	5760	2020
D-Wave	D-Wave Advantage 2 ^{[71][72][73][74]}	Superconducting ^{[71][72]}	Zephyr $Z_{15}^{[74][75]}$	N/A	7000+ ^{[71][72]} ^{[73][74][75]}	Late 2024 either 2025 ^{[71][72][73][74][75]}

Analog quantum processors [edit]

These QPUs are based on analog Hamiltonian simulation.

Manufacturer	Name/Codename/Designation	Architecture	Layout	Fidelity (%)	Qubits	Release date
QuEra	Aquila	Neutral atoms	N/A	N/A	256 ^[76]	November 2022

DiVincenzo-va merila za fizično implementacijo kvantnega računalnika

1. Dobro definiran in razširljiv register kvantnih bitov
– stabilen pomnilnik
 2. Register mora biti nastavljiv v stanje “000...”
 3. Dolgi časi dekoherence ($>10^4 \cdot$ čas procesiranja)
 4. Univerzalni nabor vrat
 5. Enostavna meritev posameznih kvantnih bitov
 6. Pretvorba med stacionarnimi in premikajočimi kvantnimi biti (fotoni)
 7. Transport premikajočih kvantnih bitov (fotonov) med dvema lokacijama
- } za komunikacijo,
enkripcijo

D. P. DiVincenzo, in *Mesoscopic Electron Transport*, eds. Sohn, Kowenhoven, Schoen (Kluwer 1997), p. 657, cond-mat/9612126; “The Physical Implementation of Quantum Computation,” quant-ph/0002077.

Delovanje kvantnega računalnika

Koncept delovanja pripravi – razvij - izmeri:

- **Pripravi:** postavitev kvantnega registra v začetno stanje (npr. vsi kv. biti se postavijo v stanje $|0\rangle$)
- **Razvij:** izvaja se zaporedje operacij, ki spremeni začetno stanje registra v potencialna superpozicionirana stanja (sprejemljive rešitve)
- **Izmeri:** vrne eno samo stanje, ki je prisotno v superpoziciji (kolaps superpozicije)

Dekoherenca

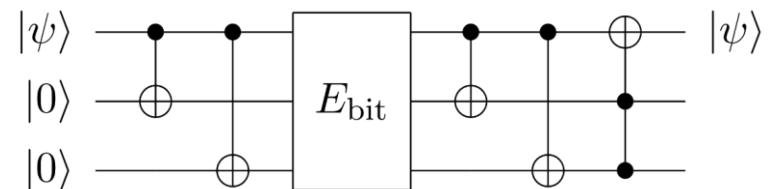
- **Kvantna dekoherenca** je proces, pri katerem kvantni sistem izgubi svoje kvantne lastnosti, kot sta superpozicija in prepletost, zaradi interakcij z okoljem.
- Dekoherenco lahko preprečimo z:
 - **Izolacija kvantnega sistema** (npr. zelo nizke temperature, zaščita s pomočjo elektromagnetnih polj ali posebnih materialov...)
 - **Popravljanjem napak**, ki ga imajo tudi klasični računalniki. A v klasičnem računalniku lahko popravljanje napak izvedemo s pomočjo dodatnih kopij klasičnih bitov, kvantnih bitov pa ne moremo kopirati (klonirati). Zato so potrebni drugačni pristopi.

Quantum Decoherence: The Death of Quantum Computing – MIT Technology Review (2023).
How Quantum Decoherence Affects Quantum Computers – IBM Quantum Blog (2021).

Kvantno popravljanje napak (Quantum error correction - QEC)

- Namesto kopiranja se v QEC pogosto uporablja kvantno prepletanje:
 - Namesto v 1 kvantnem bitu hrani informacijo v treh prepletenih kvantnih bitih (preplet s pomočjo CNOT vrat).
 - Dekoherenca bo spremenila ali enega ali dva ali tri kvantne bite (tudi njihov kvantni preplet)
 - Če se torej kvantni biti razlikujejo, je prišlo do napake.
 - Pomembno je, da lahko izmerimo, ali se kvantni biti razlikujejo, ne da bi dejansko izmerili posamezen kvantni bit in s tem uničili njegovo kvantno stanje.

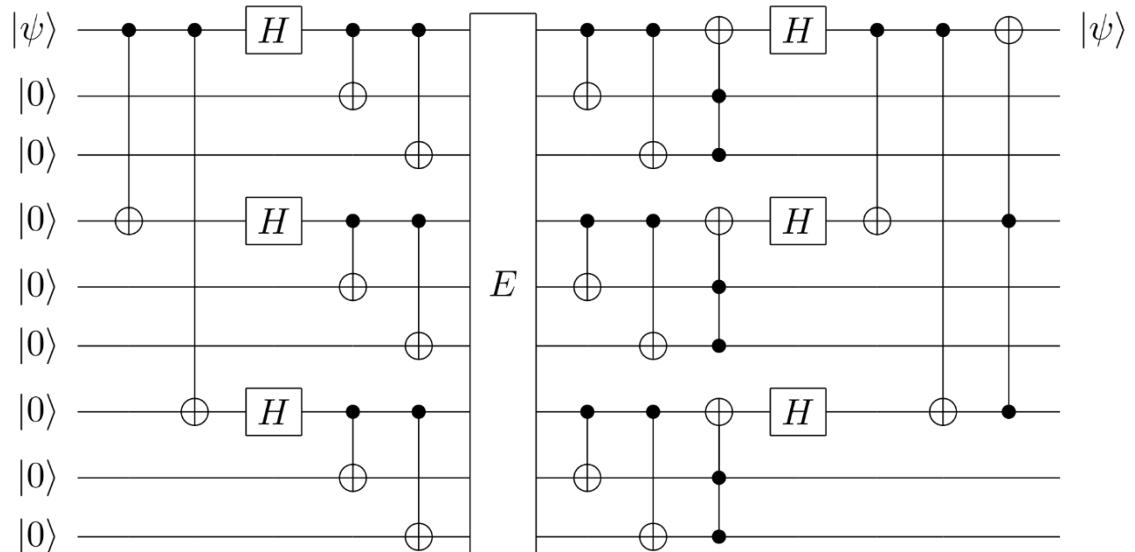
Algoritem za detekcijo napak v enim kvantnem bitu, ki uporablja dva dodatna kvantna bita (ancillary qubits)



https://commons.wikimedia.org/wiki/File:Quantum_error_correction_of_bit_flip_using_three_qubits.svg

Shorov algoritem za kvantno popravljanje napak

Leta 1995 je Peter Shor objavil algoritem, ki s pomočjo dodatnih 8 kvantnih bitov popravi napake (učinek dekoherence) enega kvantnega bita.



https://commons.wikimedia.org/wiki/File:Shore_code.svg

Shor, Peter W. (1995). "Scheme for reducing decoherence in quantum computer memory". *Physical Review A*. **52** (4): R2493–R2496. [Bibcode:1995PhRvA..52.2493S](#). doi:[10.1103/PhysRevA.52.R2493](https://doi.org/10.1103/PhysRevA.52.R2493). PMID [9912632](#).

https://en.wikipedia.org/wiki/Quantum_error_correction

Ionske pasti: How do you build a quantum computer? | Jonathan Home

Enjoy this unedited talk by Jonathan Home
Recorded at TED2015 Fellows

TEDArchive

Previously unpublished talks from TED conferences

<https://youtu.be/ArW8x1NpqGs?si=mXUPFMsffqgmF9Sa>

Ioni v pasti

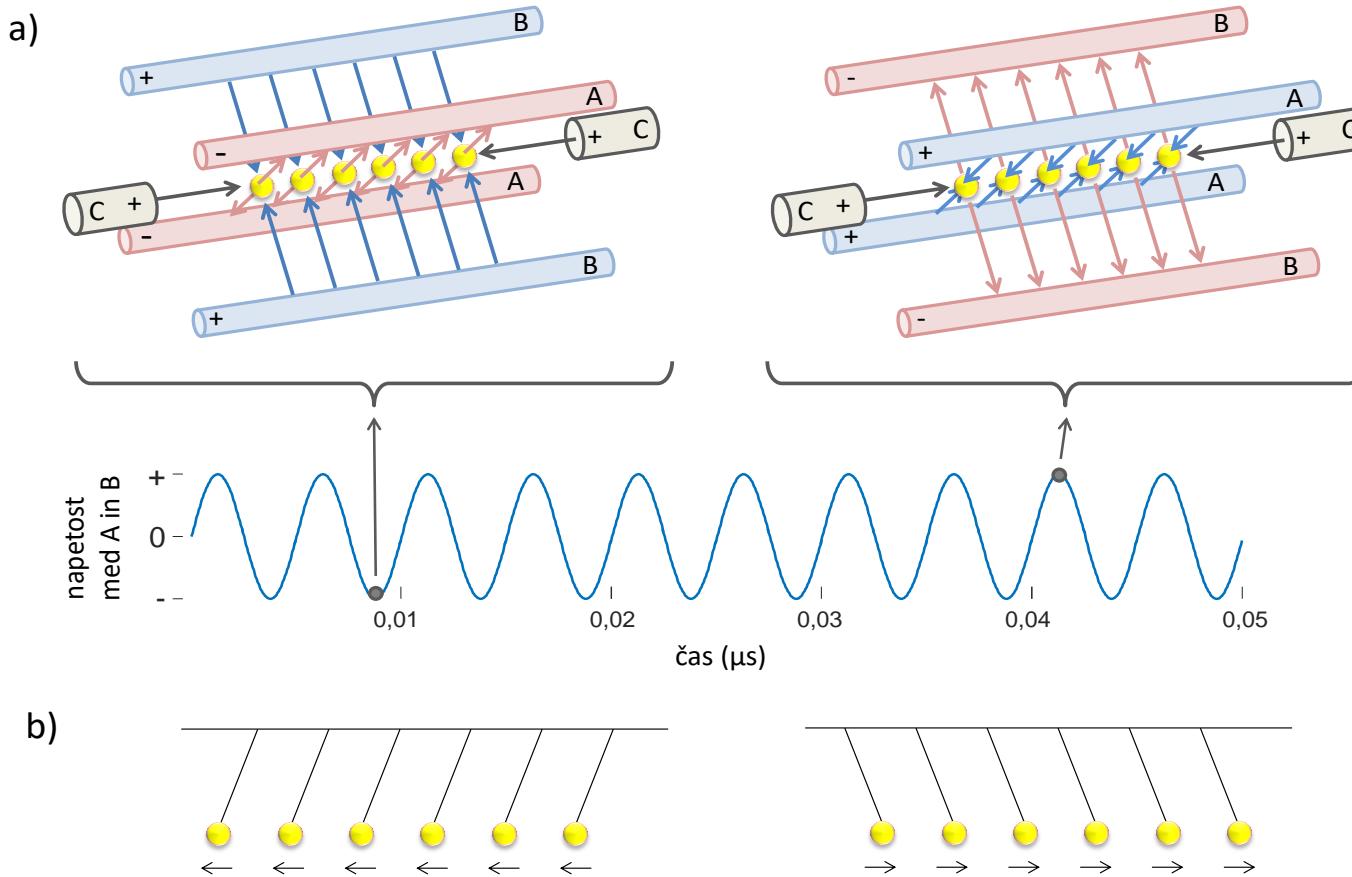
Tako po odkritju Shorovega algoritma za faktorizacijo celih števil (leta 1994) so bile kot eden najbolj obetavnih kandidatov za izgradnjo malih kvantnih računalnikov predlagane ionske pasti, v katerih prostolebdeče ione vzbujamo z lasersko svetlobo. Razlog je bil večleten razvoj in obvladovanje tehnologij za nadzor in manipulacijo enega (ali več) ionov na področju ultravisoko natančne spektroskopije in atomskih ur. Ione lahko ujamemo in ohladimo tako, da ostanejo praktično zamrznjeni v določeni regiji prostora. Njihova notranja stanja lahko natančno manipuliramo z laserjem, meritve pa opravimo s praktično 100% natančnostjo. Ioni so medsebojno močno sklopljeni z odbojno Coulombovo silo (pomembno za entangulacijo oz. kvantno prepletanje), hkrati pa so učinkovito ločeni od okolja (pomembno zaradi preprečevanja dekoherence oz. nehotenih meritev kvantnega stanja).

Earnshaw-ov teorem: skupek točkasto nabitih delcev ni mogoče ohraniti v stabilni stacionarni konfiguraciji -> ioni se vedno gibljejo.

J.I. Cirac, P. Zoller, Quantum Computations with Cold Trapped Ions, Phys. Rev. Lett. 74 20, 4091–4094 (1995)

<http://www.uibk.ac.at/th-physik/qo/research/trappedions.html>

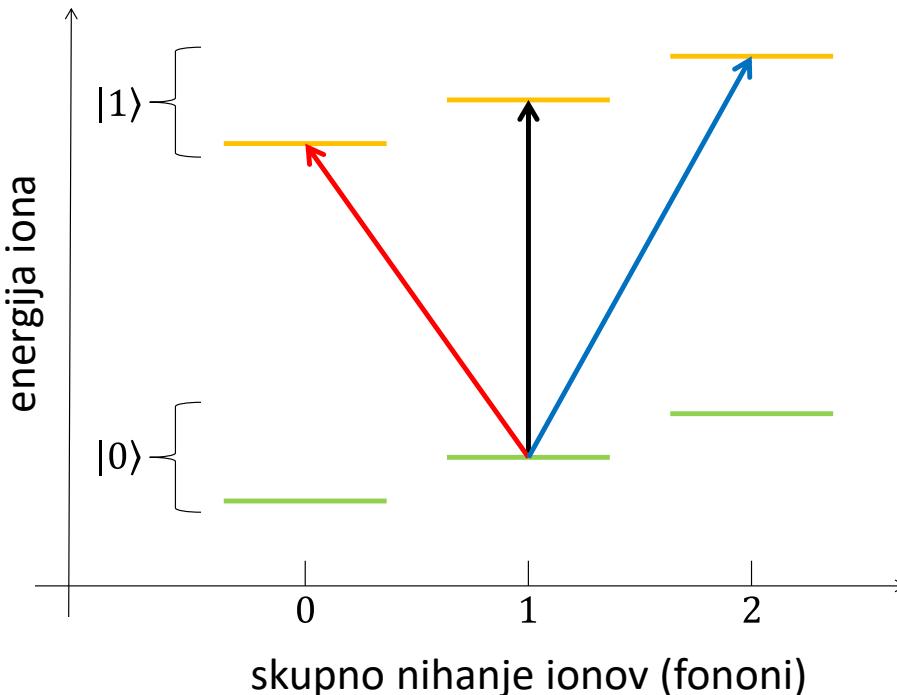
Ionske pasti in kvantni register



Osnovna struktura pasti je razporeditev štirih palic (z alternirajočo RF-napetostjo za stabilizacijo ionov v ravno vrsto) in dveh obročev (z DC-napetostjo za omejitev razmika med ioni). Obstaja veliko variacij te strukture, vključno s ploščatimi elektrodami vgrajenimi na površino mikrovezja.

Energijski nivoji elektronov v ionih

(samo ena izmed številnih možnih implementacij)

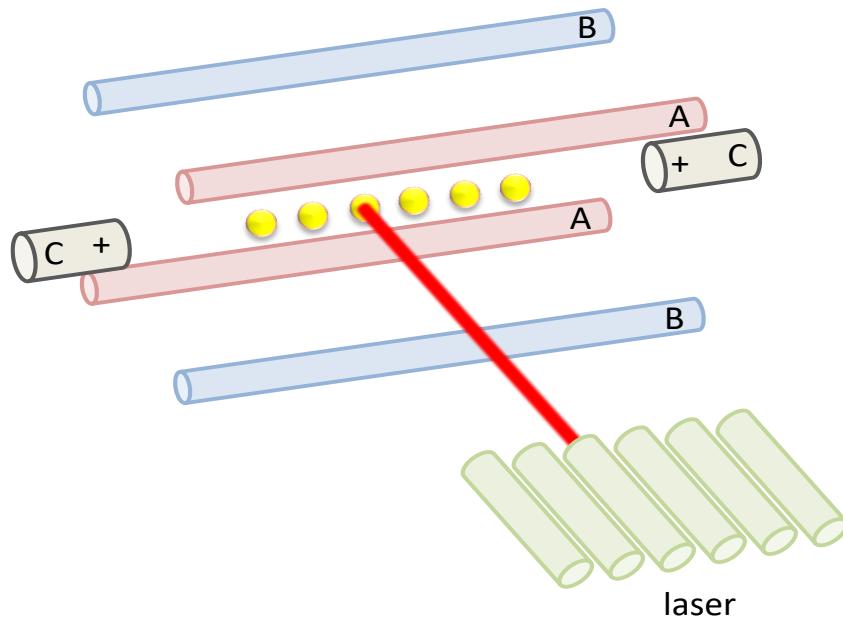


Prehod elektronov posameznega iona iz nižjih v višje energetsko stanje in izmenjava kvantov energije med elektroni in skupnim nihanjem ionov. Če ion osvetlimo s frekvenco $f_{|0\rangle \rightarrow |1\rangle}$ (črna puščica na sliki), preide elektron in nižjega v višje energijsko stanje. Če ga osvetlimo s svetlobo frekvence $f_{|0\rangle \rightarrow |1\rangle} - \Delta f$ (rdeča puščica na sliki), bo v višje energetsko stanje prišel le tako, da bo skupnemu nihanju ionov odvzel en fonon energije. Če pa ga osvetlimo s svetlobo $f_{|0\rangle \rightarrow |1\rangle} + \Delta f$, bo ob prehodu v višje energijsko stanje en fonon energije oddal skupnemu nihanju ionov.

Ujeti ioni in kvantni biti

- Hiperfina energijska stanja ujetih ionov se običajno uporablja za shranjevanje stanj kvantnih bitov v ionski pasti. Imajo zelo dolgo življenjsko dobo, več kot ~ 10 min (tipičen procesorski čas kvantnih vrat $\sim 1 \mu\text{s}$).
- frekvenca, povezana z energijo stanja je v mikrovalovnem območju, zaradi česar je mogoče za manipulacijo stanj uporabiti mikrovalovno sevanje. Vendar trenutno ne poznamo vira mikrovalovnega sevanja, ki bi tvoril dovolj ozek curek, da bi z njim lahko nadzorovali posamezen ion v zaporedju ionov v ionski pasti. Namesto tega je mogoče uporabiti par laserskih pulzov z različnima frekvencama (razlika frekvenc mora biti enaka zahtevani frekvenci za prehod stanj). Fiziki ta pojav poznajo kot spodbujen Ramanov prehod (angl. *Raman transition*).

Ionske pasti in kvantni register

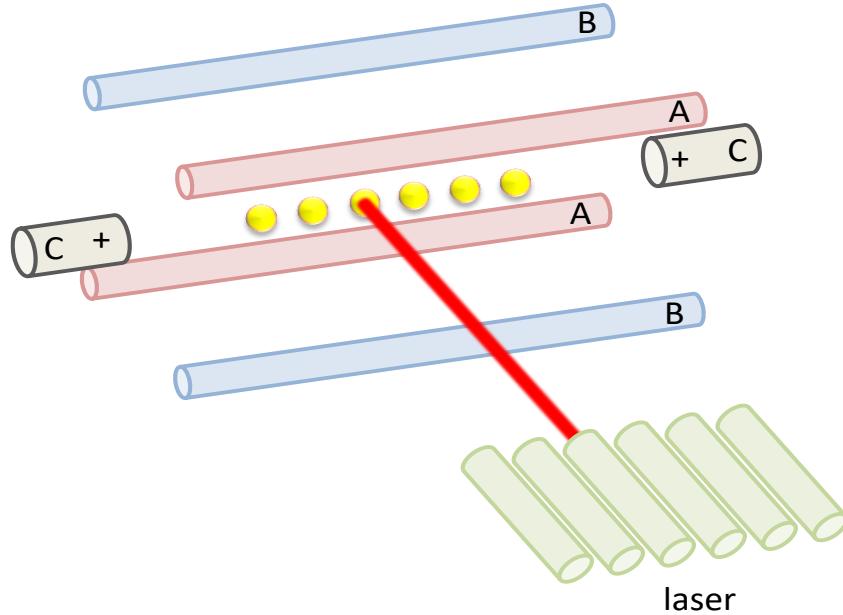


S pomočjo laserja ione ohladijo, tako da se umirijo in tvorijo niz kvantih bitov – kvantni register.

Kvantna informacija je shranjena v energijskih stanjih posameznih ionov.

Ob obstreljevanju s fotoni posamezen ion skače navzgor in navzdol med parom svojih energijskih stanj in pri tem seva svetlobo. Če se med tem procesom ion vzbudi v tretje energijsko stanje, ni več na voljo za vzbujanje z izbrano frekvenco in tako preneha oddajati svetlobo. Ko torej ion prehaja med osnovnima in tretjim stanjem, njegovo razmeroma močno oddajanje svetlobe izgine in se ponovno pojavi. Ti značilni skoki njegove svetilnosti (flourescence) kažejo temeljni kvantni proces kvantnega bita – ion se pod vplivom opazovanja odloči v katero izmed stanj se bo vzbudil.

Ionske pasti in kvantni register

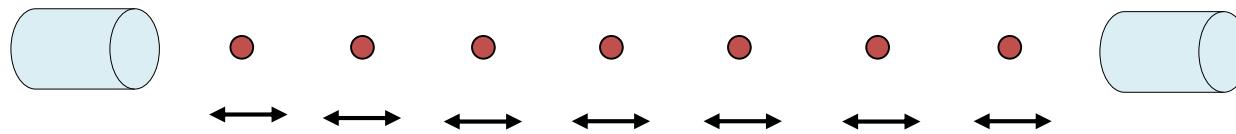


Incializacija: ione osvetlimo z laserjem takšne frekvence, da vzbuja eno izmed osnovnih stanj npr. stanje $|1\rangle$, drugega stanja (torej stanja $|0\rangle$) pa laser ne vzbuja. To počnemo tako dolgo, dokler se ne zgodi spontan prehod iz vzbujenega stanja iona v stanje $|0\rangle$. Od tega trenutka dalje je ion v stabilnem stanju $|0\rangle$.

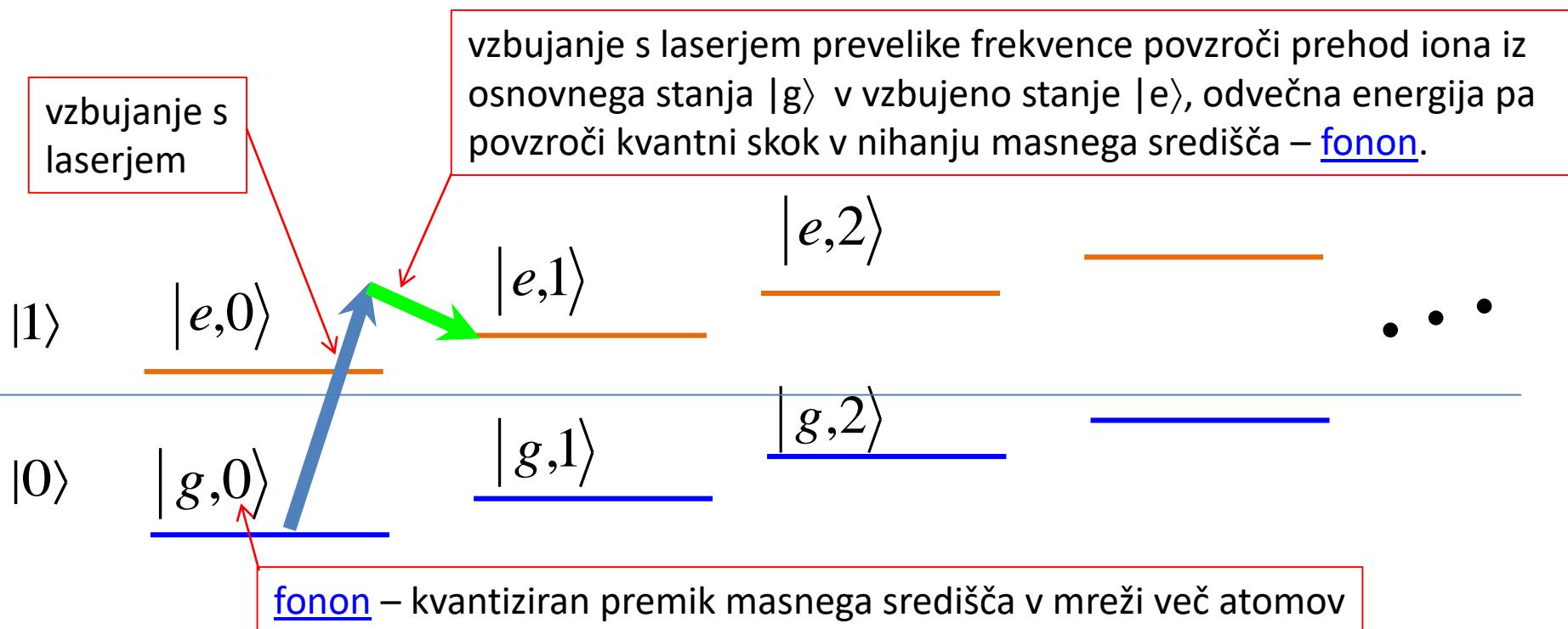
Superpozicija: superpozicijo osnovnih stanj $|0\rangle$ in $|1\rangle$ lahko dosežemo z manipulacijo dolžine osvetlitve. Če je potrebno za prehod iz stanja $|0\rangle$ v stanje $|1\rangle$ ion osvetljevati d mikrosekund, potem bo osvetlitev z dolžino $d/2$ mikrosekund ion iz stanja $|0\rangle$ pripeljala v stanje superpozicije $|0\rangle + |1\rangle$.

Meritev: meritev opravimo z laserjem takšne frekvence, da vzbuja eno izmed osnovnih stanj $|0\rangle$ ali $|1\rangle$, drugega pa ne. Recimo, da osvetljujemo ion s frekvenco, ki vzbudi ion iz stanja $|0\rangle$ v vzbujeno stanje. Če je ion v stanju $|0\rangle$ se bo vzbudil, ob povratku v stanje $|0\rangle$ pa bo oddal foton, ki ga je moč zaznati z CCD kamero. Če bo ion v stanju $|1\rangle$ se ne bo vzbudil in tudi ne bo oddal fotona.

Ionske pasti - Cirac & Zoller, Wineland *et al*, Blatt *et al*.

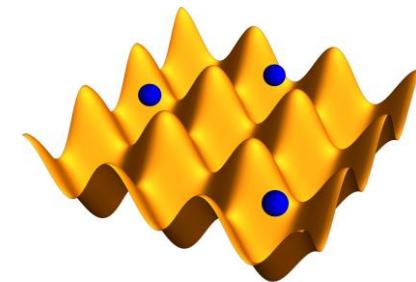


Posamezni ioni (kvantni biti) so vezani preko gibanja masnega središča (centre of mass motion), ki se prenaša preko Coulombove sile – na ta način lahko kodiramo kvantno prepletanje (entangulacijo)

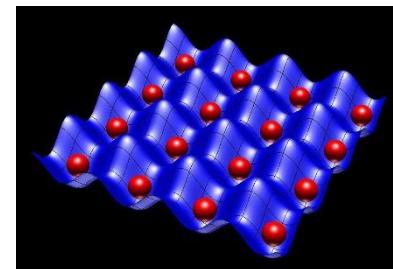


Optične mreže (Optical lattice)

- uporablja lasersko svetlobo za ustvarjanje optične mreže, v kateri so ujeti nevtralni atomi ali ioni.
- omogoča natančno kontrolo in manipulacijo kvantnih stanj
- Optična mreža se ustvari z uporabo prekrivajočih laserskih žarkov, ki ustvarijo periodičen potencial. Ta potencial deluje kot past za nevtralne atome, ki se uporabljajo kot kvantni biti
- Atomi v mreži med seboj komunicirajo z laserskimi ali elektromagnетnimi polji, ki omogočajo manipulacijo (implementacijo kvantnih vrat)
- Natančna pozicija atomov v mreži zagotavlja daljše koherenčne čase kvantnih bitov
- Teoretično lahko v mreže shranimo veliko število atomov, a ostajajo praktični izzivi glede skaliranja.



[https://commons.wikimedia.org
/wiki/File:AtomsInLattice.png](https://commons.wikimedia.org/wiki/File:AtomsInLattice.png)



[https://upload.wikimedia.org/
wikipedia/commons/9/9f/Qu
bits_%285940500587%29.jpg](https://upload.wikimedia.org/wikipedia/commons/9/9f/Qu_bits_%285940500587%29.jpg)

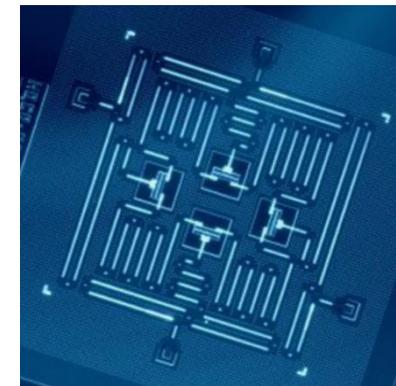
Kvantne pike (quantum dots)

- Nanometrski strukturirani polprevodniki, ki imajo lastnosti kvantnih sistemov.
- So zelo majhne 3D pasti za elektrone, zaradi njihove majhnosti pa lahko imajo elektroni v njih le določena, kvantizirana energijska stanja.
- Kvantne operacije temeljijo na manipulaciji energijskih stanj elektronov v pikah.
- Kvantne pike so izjemno majhne in jih je mogoče enostavno integrirati v različne tehnologije.
- Kvantna vrata za kvantne pike
 - **laserskih pulzi**, ki natančno spreminjajo energijske nivoje elektronov v kvantnih pikah
 - **elektromagnetnih polja** spremenimo način interakcije z elektroni, in omogočajo implementacijo Hadamardovih, CNOT in faznih vrat
 - Spremembra **spina** omogoča ustvarjanje prepletenih stanj (entanglement)

Superprevodnost

- Superprevodnost omogoča prenos električne energije brez upora (navadno pri zelo nizkih temperaturah).
- V kvantnih računalnikih omogoča dolgo koherenco, visoko natančnost manipulacije kvantnih bitov, enostavno integracijo.
- Kompleksnost superprevodnih krogov predstavlja izviv za skalabilnost
- Možne implementacije kvantnih bitov:
 - **Superprevodni krogi (superconducting circuits)**: Kvantni biti implementirani s **Josephsonovi spoji**, ki so sestavljeni iz dveh superprevodnih materialov, med katerima je zelo majhen izolator.
 - **Kvantni tokovni krog: Superprevodni kvantni bit (ali transmon qubit)** je oblikovan tako, da tvori **tokovni krog**, katerega energijska stanja predstavljajo kvantne informacije.
 - **Kvantno tuneljenje**: kvantni biti izkoriščajo **tuneljenje** parov električnih nabojev (Cooperjevih parov), da prečkajo energijske vrzeli, kar omogoča preklop med različnimi stanji kvantnega bita.

4 Qubit, 4 Bus, 4 Resonator IBM Device



https://upload.wikimedia.org/wikipedia/commons/a/a4/4_Qubit%2C_4_Bus%2C_4_Resonator_IBM_Device_%28Jerry_M._Gambetta%2C_Jerry_M._Chow%2C_and_Matthias_Steffen%2C_2017%29.png



<https://www.intel.com/content/www/us/en/research/quantum-computing.html>

A TIMELINE OF QUANTUM COMPUTING



PHENOMENOLOGICAL PHASE 1950s - 1990s

Primarily theoretical research, with limited physical experimentation



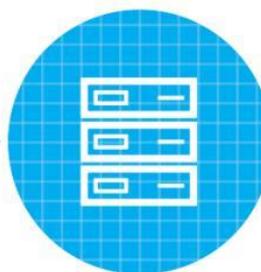
EXPERIMENTAL PHASE 1990s - 2000s

Establishment of fundamental mechanisms with physical apparatus



REALIZATION PHASE 2010s

Development of quantum processors and rudimentary quantum computers



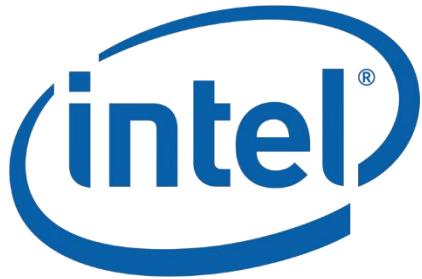
SYSTEM PHASE 2015 - 2025

System-level engineering for practical quantum computers



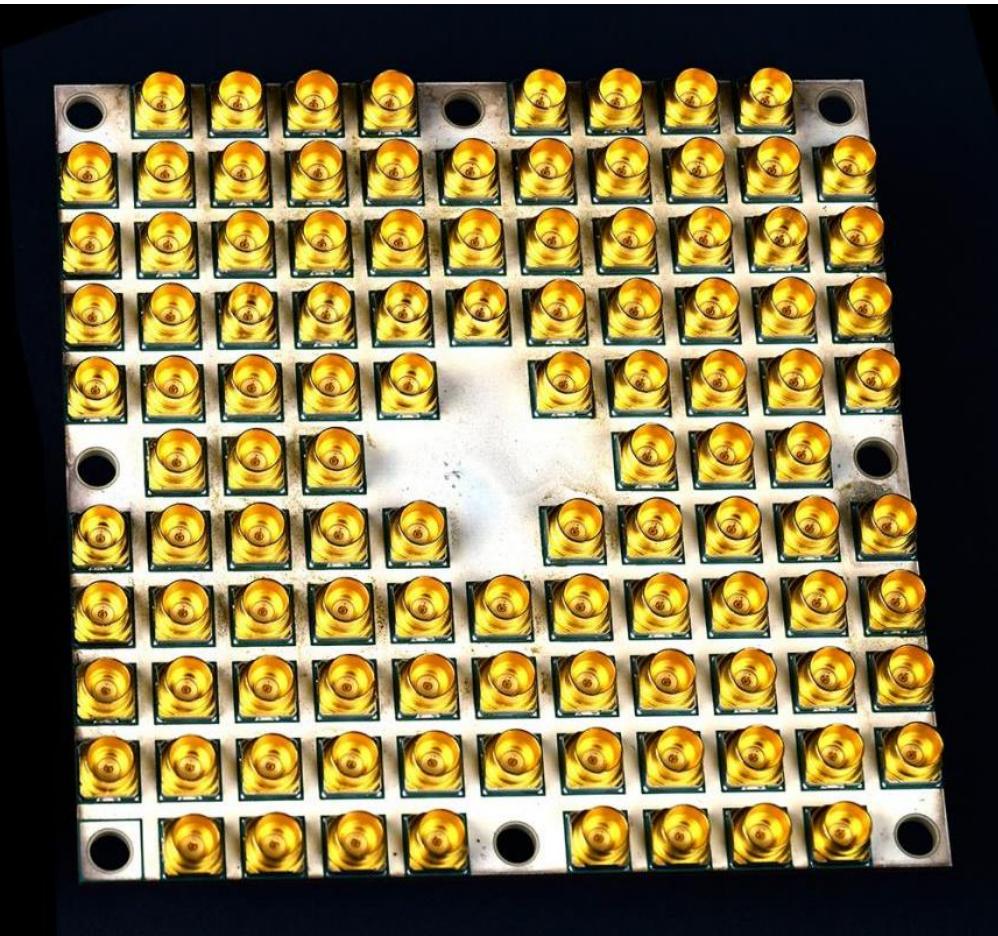
COMMERCIAL PHASE 2025 and beyond

Production use of quantum computing systems to solve real-world problems



<https://www.intel.com/content/www/us/en/research/quantum-computing.html>

Tangle Lake – 49-bitni kvantni procesor (superprevodnost)



D-Wave – KR komercialno

<https://www.dwavesys.com/>

- posebnonamensko “kvantno” procesiranje za reševanje optimizacijskih problemov (ni univerzalnih vrat)
- Leto 2007: 28 qubitni delovni register
- 20. Maj, 2011: D-Wave one: 128 kvantnih bitov
- 2013: D-Wave Two – 512 kvantnih bitov
- 2015: D-Wave 2X – 1152 kvantnih bitov
- 2017: D-Wave 2000Q – 2048 kvantnih bitov
- 2020: Advantage – 5640 kvantnih bitov

http://en.wikipedia.org/wiki/D-Wave_Systems

Adiabatno kvantno računanje:

https://en.wikipedia.org/wiki/Adiabatic_quantum_computation

Kvantna izmenjava ključa (Quantum Key Distribution)

- <http://www.youtube.com/watch?v=SsZ0oXLT1n4&feature=related>

Protokol E91 : Artur Ekert (1991):

Uporablja pare preletenih (entanguliranih) fotonov. Slednje lahko ustvariti ali Alice ali Bob ali tretja oseba, vključno s prisluškovalko Evo. Enega izmed fotonov v prepletenu paru pošljemo Alici, drugega Bobu.

Protokol izkorišča dve lastnosti kvantnega prepletanja (entangulacije):

1. Prepleteno stanje popolnoma določa stanji obeh fotonov – stanji obeh fotonov sta popolnoma naključni, če pa izmerimo enega izmed fotonov, lahko z verjetnostjo 100% določimo stanje drugega, neizmerjenega fotona.
2. Vsak poskus prisluškovanja spremeni stanje opazovanega fotona, kar se z luhkoto zazna.

Kvantna Izmenjava Ključev: Protokol BB84

- Charles H. Bennett in Gilles Brassard (1984)
- <https://www.youtube.com/watch?v=IE5952ExMK8>

Baza meritve	0	1
+	→	↑
×	↗	↖

Alica in Bob se javno dogovorita o dveh bazah prostorov

Alicini naključni biti	1	0	1	0	1	1	1	0	0
Alicina naključna izbira baze	+	+	×	×	+	×	+	×	+
Polarizacija fotonov, ki jih pošlje Alice	↑	→	↖	↗	↑	↗	↑	↗	→
Bobova naključna baza meritvev	×	+	×	+	+	+	×	×	×
Bobove izmerjene polarizacije fotonov	↗	→	↖	→	↑	↑	↖	↗	↖
Javna diskusija baze meritvev za vsak izmerjeni foton	!	OK	OK	!	OK	!	!	OK	!
Skupen skriti ključ		0	1		1			0	

Kvantna Izmenjava Ključev: Protokol BB84

- Vsako prislушкиvanje je možno zaznati s poljubno verjetnostjo.
- Zadostuje, da žrtvujemo N od M bitov skritega ključa (skriti ključ zmanjšamo na M-N bitov) in jih medsebojno primerjamo (Alica in Bob jih izmenjata preko javnega omrežja).
- Verjetnost, da bo prisluskovalka Eva izbrala napačno bazo pri posameznem bitu je 50%. Eva ponovno pošlje foton, ki ima naključno polarizacijo, torej bo v bazi, ki jo je izbrala Alica (in po naključju tudi Bob) zavzel napačno vrednost z verjetnostjo 50%.
- Torej je verjetnost, da bo pri prisluskyvanju prišlo do napake v bitu skritega ključa enaka $50\% \cdot 50\% = 25\%$.
- Verjetnost, da bomo ob prisluskyvanju odkrili napako v skritem ključu je torej enaka

$$P(\text{spy detected}) = 1 - (3/4)^N$$

pri čemer je N število bitov skritega ključa, ki smo jih žrtvovali za preverjanje.

- Če želimo prisluskyvanje zaznati z **verjetnostjo 99, 9999%**, potrebujemo žrtvovati **N=50** bitov skritega ključa.

Zanimive povezave

- **Introduction to quantum cryptography - Vadim Makarov:**
<https://www.youtube.com/watch?v=ToOLbdrWst4>
- **Quantum Cryptography in 6 Minutes**
<https://www.youtube.com/watch?v=uiaAJ3c6dM>
- **Why Quantum Computing Requires Quantum Cryptography**
<https://www.youtube.com/watch?v=pi7YwxxZQ5A>
- **Quantum Computers Explained – Limits of Human Technology**
<https://www.youtube.com/watch?v=JhHMJCUmq28>

http://en.wikipedia.org/wiki/Timeline_of_quantum_computing

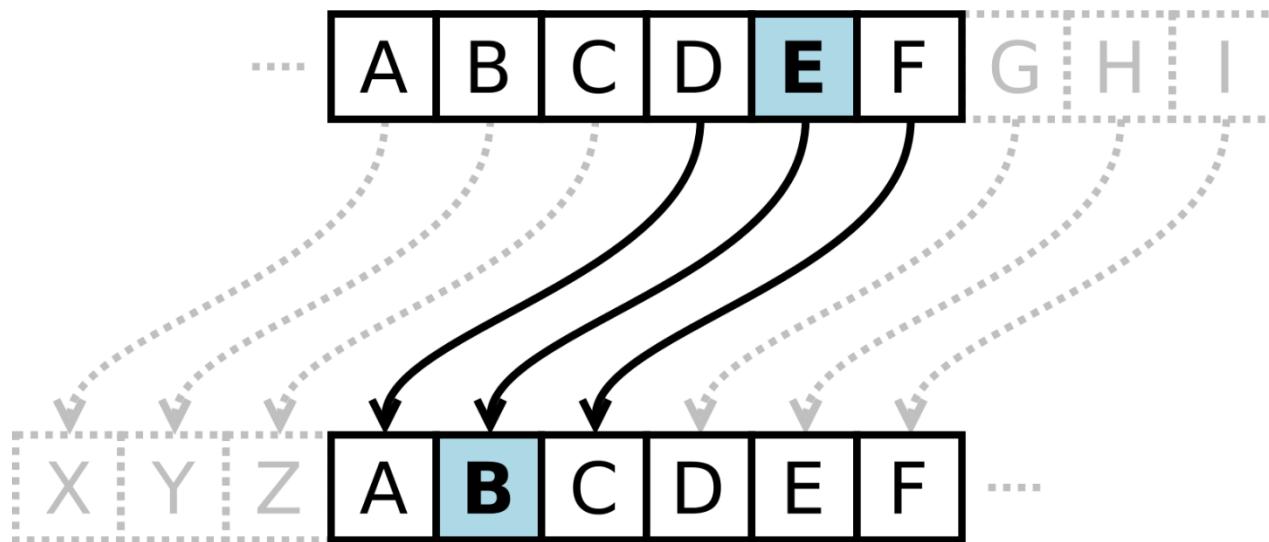
Kriptografija

Cryptography has been a major force in human history for more than 3,000 years

The Codebreakers by David Kahn

Zgodovina kriptografije

- **"Cesarjev kodirnik"** v času Rimskega imperija:
 - Za vsako črko se premakni za n mest naprej po abecedi in za Z ponovno priči pri A.
 - enostavno za dekodiranje: analiza frekvence črk v kriptiranem besedilu (samoglasniki so pogostejši).



Zgodovina kriptografije

- Leta 1920 je bil zasnovan prvi teoretično dokazano varen informacijski kodirnik: **one-time pad**:
 - nekodirano besedilo P zapišemo z bitnim zaporedjem in ga preko operacije XOR prištejemo k bitnemu zaporedju kodirnega ključa K , ki je enake dolžine kot osnovno besedilo. Dobimo kodirano besedilo C .
 - Prejemnik (ki pozna K) lahko kodirano besedilo dekodira z novo operacijo XOR:
$$C \oplus K = P \oplus K \oplus K = P \quad \text{kjer je } \oplus \text{ bitna operacija XOR}$$
 - Težava kodirnika “one-time pad” je v tem, da **morata bitno zaporedje ključa poznati tako oddajnik kot prejemnik in da mora biti to zaporedje dolžine sporočila, ki ga kodiramo.**
 - Če je isto zaporedje ključa uporabljen pri dveh ali več sporočilih, potem kodirnik ni več varen, saj velja

$$C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$

Stopnje varovanja podatkov

- Varnost komunikacij in varovanje podatkov sta v širšem pomenu opredeljena z načeli lastništva (*possession*), zaupnosti (*confidentiality*), celovitosti (*integrity*), dostopnosti (*availability*), pristnosti (*authenticity*), nezatajljivosti (*non-repudiation*) in uporabnosti (*utility*).
- **Zaupnost**

Zaupnost izraža preprečevanje razkrivanja informacij nepooblaščeni osebi ali sistemu. Navadno se zagotavlja s šifriranjem podatkov, z omejevanjem mesta dostopnosti podatkov (v podatkovnih bazah, dnevnikih sistemov, varnostnih kopijah, obvestilih uporabnikom itd.) ter z omejevanjem dostopa do mesta, kjer so podatki shranjeni.

Zaupnost je potreben (a ne zadosten) pogoj za zagotavljanje zasebnosti oseb in za varovanje njihovih podatkov.
- **Celovitost**

Celovitost podatkov pomeni, da podatkov ni mogoče spremnjati brez dovoljenja. Definicija celovitosti na področju informacijske varnosti se torej močno razlikuje od definicije podatkovne celovitosti v podatkovnih bazah. Celovitost je na primer kršena, če oseba ali program po naključju, napaki ali zlonamerno izbriše ali spremeni pomembne podatke ali ko na kakršnikoli način zlonamerno vpliva na njihovo zbiranje (npr. ko ista oseba odda veliko število glasov na spletni anketi).

Stopnje varovanja podatkov

- **Dostopnost**

Za kateri koli informacijski sistem morajo biti podatki na voljo, ko je to potrebno. To pomeni, da morajo pravilno delovati vse komponente sistema od računalniška sistema, ki ga uporabljam za shranjevanje in obdelavo informacij, do komponent za nadzor varnosti in komunikacijskih kanalov, ki se uporabljajo za dostop do podatkov. Sistemi z visoko dostopnostjo omogočajo neokrnjen dostop do podatkov, preprečujejo motnje v delovanju zaradi izpadov električne energije, okvar strojne opreme in nadgradnje sistema. Zagotavljanje dostopnosti vključuje tudi preprečevanje napadov za preprečevanje dostopa (denial-of-service attacks).

- **Pristnost**

V računalništvu, e-poslovanju in informacijskem varovanju je potrebno zagotoviti, da so podatki, transakcije, sporočila ali dokumenti (elektronski ali fizični) pristni. Prav tako je za verodostojnost pristnosti pomembno, da so v komunikacijo vpletene stranke to, kar trdijo, da so.

- **Nezatajljivost**

V pravu, nezatajljivost pomeni namero pogodbene stranke, da izpolni svoje s pogodbo določene obveznosti. Pomeni tudi, da stranka v transakciji ne more zanikati prejetja informacije, ki jo je poslala druga stranka. Elektronsko poslovanje zagotavlja verodostojnost in nezatajljivost s pomočjo tehnologij digitalnih podpisov in šifriranja.

Kriptoanaliza

- **Kriptografska definicija varnosti podatkov:**
 - V splošnem, nudi kriptografski sistem **zaščito stopnje λ** , če potrebuje uspešen napad na sistem približno **vložek v velikosti $2^{\lambda-1}$** .
 - Velikost zahtevanega vložka se v praksi ocenjuje s **produktom zahtevanega časa in cene zahtevane opreme**.
 - Po splošno sprejetem konsenu nudi kriptografski sistem primerno varnost do leta X, če znaša strošek uspešnega napada na sistem v danem letu vsaj **40 M\$-dni**. **V zadnjem času se ta strošek meri v številu operacij in znaša nekje med 2^{80} in 2^{100} operacij.**
 - Zaradi nenehnega napredka programske in strojne opreme, ki vpliva tako na učinkovitost kot na ceno uporabljeni opreme, prihaja do **naravne degradacije stopnje varnosti** hranjenih podatkov.
 - Po Moorovem zakonu, se zaradi tehnološkega razvoja zahtevan vložek prepolovi vsakih 9 mesecev (vsakih 18 mesecev zaradi napredka v opremi in vsakih 18 mesecev zaradi napredka v algoritmih kriptoanalyze). To imenujemo **dvojni Moorov zakon faktorizacije**.

Varnost in uporabnost kriptografije

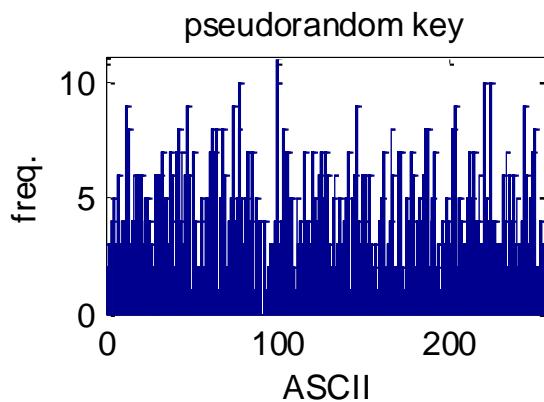
- Leta 1940 je Claude Shannon dokazal:

Kodirnik je teoretično varen, če in samo če tako oddajnik kot prejemnik uporabljata bitno zaporedje ključa, ki je dolžine komunikacijskega sporočila.

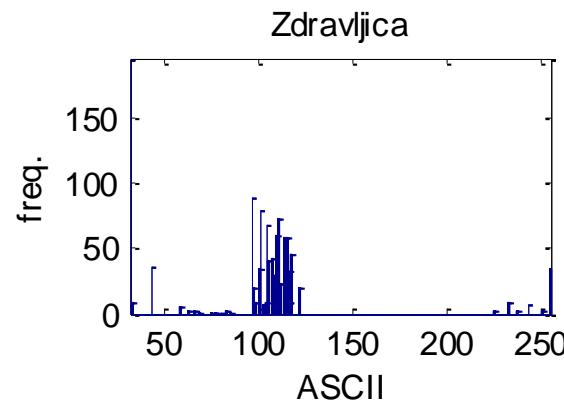
- Toda izmenjava tako velikih ključev je povsem nepraktična. Ali lahko uporabimo krajske ključe?
- Kaj če predpostavimo, da je prisluškovalec omejen s polinomskim izvajalnim časom (torej, da je časovno omejen)?

Psevdonaključni generatorji

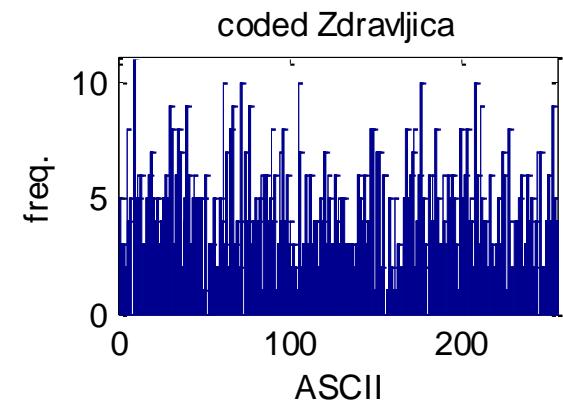
- psevdonaključni generator je funkcija f , ki ima naslednje lastnosti:
 1. f preslika n -bitno vhodno vrednost (imenovano *seme*) v $p(n)$ -bitno izhodno zaporedje, kjer je $p(n)$ polinomsko večji od n .
 2. f je izračunljiva v polinomskem času po n .
 3. zlonamerna oseba ne more v polinomskem času razlikovati izhoda funkcije f od resnično naključnega zaporedja.
- Zgled: matlab & funkcija rand



```
k = uint8(rand(1,10000)*255);
```



```
p = 'Prijatlj! obrodile so...'
```



```
c = bitxor(uint8(p), ...  
k(1:length(p)));
```

Kriptografske sekljalne funkcije

(Cryptographic hash functions)

- Večina kriptografski sekljalnih funkcij prejme na vhodu niz poljubne dolžine in vrne sekljano kodo fiksne dolžine.
- Splošne lastnosti dobrih sekljalnih funkcij lahko strnemo v naslednje postavke:
 1. enostaven izračun sekljane vrednosti kateregakoli sporočila;
 2. praktično nemogoče je najti sporočilo, ki daje izbrano sekljano vrednost;
 3. nemogoče je spremeniti sporočilo, ne da bi spremenili njegovo sekljano vrednost;
 4. praktično nemogoče je najti dve različni sporočili z isto sekljano vrednostjo;
 5. efekt plaza (*avalanche effect*): sprememba enega samega bita na vhodu povzroči veliko spremembo izračunane sekljane vrednosti.
- Znana kriptografske sekljalne funkcije:
 - MD4, MD5 ($H=128$, $\lambda=64$),
 - RIPEMD-160 ($H=160$, $\lambda=80$),
 - SHA-1 ($H=160$, $\lambda=80$),
 - SHA-256 ($H=256$, $\lambda=128$)

H označuje **bitno dolžino** ključa sekljanja: Da bi dosegli **nivo varnosti λ** in da bi zadostili prvima dvema zgornjima zahtevama (1. & 2.), mora biti $H \geq \lambda$.

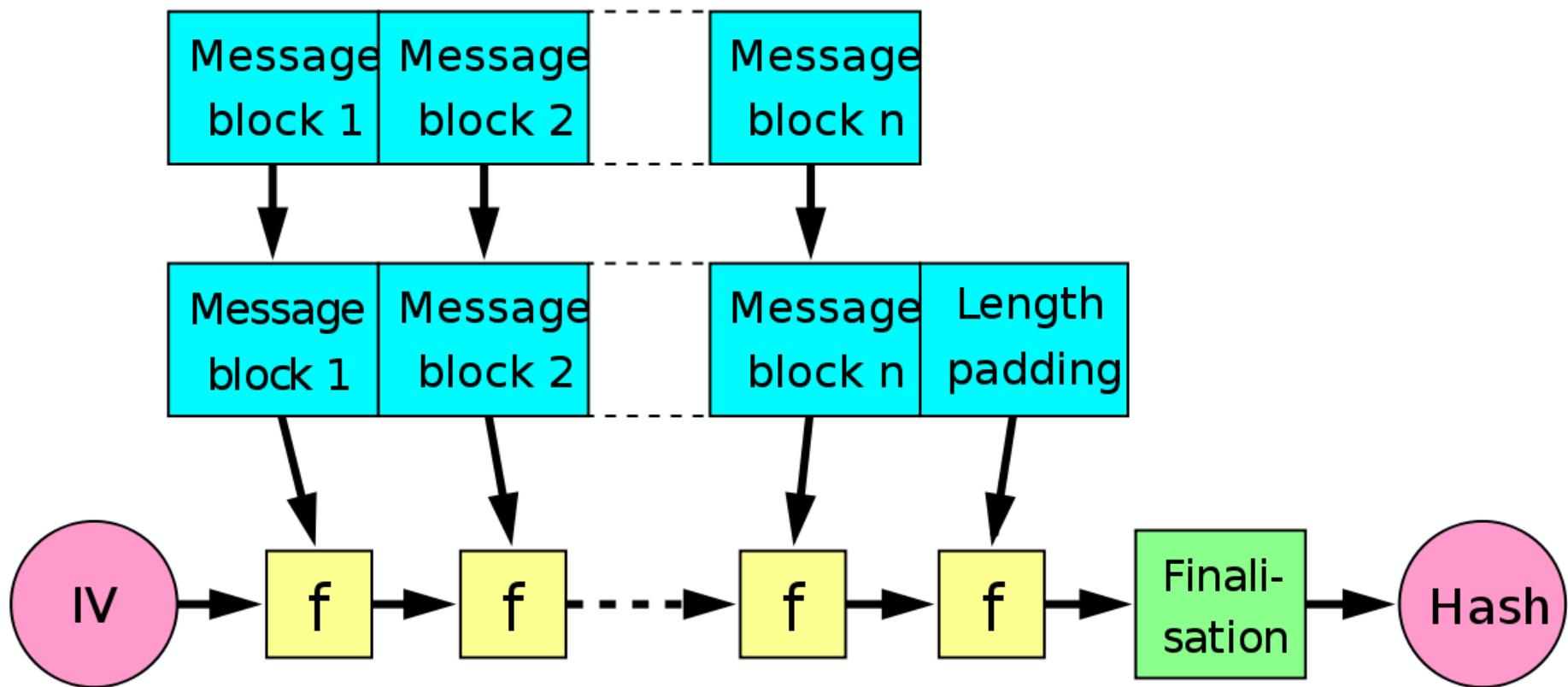
Kriptografske sekljalne funkcije

(Cryptographic hash functions)

- H označuje **bitno dolžino** ključa sekljanja.
- **Paradoks rojstnega dne** (*birthday paradox*): Če izbiramo naključne elemente množice s kardinalnostjo C je povprečno število elementov, ki bodo izvlečeni preden bo isti element izvlečen dvakrat (t.j. preden nastopi kolizija) enako $1.25\sqrt{C}$.
 - Če računamo seklijano vrednost različnih naključno izbranih vhodov in ima sekljalna funkcija H možnih izhodov, lahko podvojitev sekljalne vrednosti pričakujemo po $1.25 \cdot 2^{H/2}$ poskusih.
 - Za dosego stopnje varnosti λ in za zagotovitev odpornosti na zgoraj opisane kolizije mora veljati $H \geq 2\lambda$, kjer je H bitna dolžina sekljalne vrednosti in λ stopnja varnosti.
- **Konkatinacija kriptografskih sekljalnih funkcij:** SSL uporablja konkatinacijo sekljalnih funkcij MD5 in SHA-1, da bi zagotovila varnost komunikacijskega protokola v primeru razvozlanja ene izmed sekljalnih funkcij. V splošnem sta MD5 in SHA-1 od leta 2009 dalje dve najbolj pogosto uporabljeni kriptografski sekljalni funkciji. Vendar pa je bila MD5 razvozlana in napad nanjo je bil leta 2008 uporabljen za zlom SSL [Sotirov 2009].

Merkle-Damgård-ova konstrukcija sekljalne vrednosti

- za sekljanje podatkov poljubne dolžine:



Kodirniki (ciphers)

- Moderni kodirniki se glede načina enkripcije delijo v dva velika razreda
 - **Simetrično kodiranje** uporablja isti privatni ključ tako za kodiranje kot za dekodiranje sporočil. Primera simetričnega kodirnika sta standarda DES in AES.
 - **Asimetrično kodiranje** temelji na privatnem in javnem ključu. Javni ključ je poznan vsem in služi za enkripcijo podatkov, ki jih lahko dekodiramo samo s skritim privatnim ključem. Analogija z realnim svetom daje javnemu kriptografskemu ključu podobo fizične ključavnice, s katero lahko vsakdo zaklene poljuben zabolj, privatnemu ključu pa podobo fizičnega ključa, ki to ključavnico odklepa. Primer asimetričnega kodirnika je standard RSA.

Simetrični kodirniki se dodatno delijo v **pretočne (stream)**, ki operirajo nad posameznimi zlogi podatkov, in **bločne (block)**, ki hkrati obdelajo celoten blok podatkov (trenutno je tipična velikost bloka 128 zlogov). Meja med bločnimi in pretočnimi kodirniki ni izrazita in številni bločni kodirniki se lahko uporabijo kot pretočni kodirniki.

Dolžina kriptografskega ključa

- Obstaja široko soglasje o dolžinah simetričnih ključev in kriptografskih sekljalnih funkcij, ki so "konzervativne", torej imajo dobro možnost, da ponujajo zelo dolgoročno varnost.
- Veliko manj konsenza je o konzervativnih dolžinah asimetričnih kriptosistemov, kot je RSA.
- Po trenutnem konsenzu nudijo vsi naslednji kodirniki podobne stopnjo zaščite:
 - simetrični kodirnik s 128 bitnim ključem
 - asimetrični kodirnik s 3072 bitnim ključem,
 - asimetrični kodirnik z uporabo eliptičnih krivulj s 512 bitov dolgim ključem.
- Dolžino ključa, ki izbrani metodi enkripcije zagotavlja ustrezeno stopnjo varnosti do poljubnega leta si lahko izračunamo na spletni strani: <http://www.keylength.com/>

Asimetrični kodirniki

(Asymmetric ciphers)

- Asimetrično kodiranje temelji na privatnem in javnem ključu.
- Javni ključ je poznan vsem in služi za enkripcijo podatkov, ki jih lahko dekodiramo samo s skritim zasebnim ključem. Običajno lahko javni ključ uporabnika A uporablja katera koli stranka za šifriranje podatkov (seveda samo za podatke, ki so namenjeni uporabniku A).
- Uporabnik A lahko podatke dešifrira samo s pomočjo zasebnega ključa. Javni in zasebni ključ vedno nastopata v parih, tako kot ključavnica in njen ključ.
- Trenutno so v uporabi trije implementacijski koncepti kodiranja z javnim/zasebnim ključem:
 - faktorizacija celih števil (*integer factorization*)
 - diskretni logaritmi (*discrete logarithm*)
 - eliptične krivulje (*elliptic curve*)

Faktorizacija celih števil

- **Osnovni problem faktorizacije celih števil:** Za dano sestavljenou celo število $n > 0$, najdi celi števili $p > 1$ in $q > 1$ tako, da velja $n = pq$.
- Pri RSA, najpogostejšem faktorizacijskem algoritmu za asimetrične kriptosisteme, vsebuje uporabnikov javni ključ celo število n , njemu ustrezen zasebni ključ pa vsebuje praštevili p in q . Število n je za vsakega uporabnika unikatno.
- **Nekaj ključnih lastnosti faktorizacije praštevil:**
 - **Število praštevil do x je sorazmerno $x/\log(x)$:** torej, če je število n zgrajeno kot produkt dveh, recimo, b -mestnih praštevil, je računska kompleksnost za faktorizacijo števila n z metodo zaporednih poskusov velikostnega reda 10^b . Že za zmero velikost $b=50$ je računski napor te velikosti izven dosega sodobnih klasičnih računalnikov.
 - Obstajajo algoritmi faktorizacije, ki so veliko hitrejši od metode zaporednih poskusov in **zahtevana bitna dolžina modulov $p > 1$ in $q > 1$ raste veliko hitreje kot linearne funkcije želene stopnje varnosti.**

Faktorizacija celih števil

- Algoritem *Number Field Sieve* (NFS), trenutno najhitrejša znana metoda za faktorizacijo celih števil na klasičnih računalnikih, potrebuje v povprečju $O(\exp((b \cdot 64/9)^{1/3} \cdot \log(b)^{2/3}))$ operacij za faktorizacijo b -bitnega celega števila n , torej $2/3$ poti med eksponentnim in polinomskim časom.
 - Maja 2005 je bila oznanjena uspešna faktorizacija RSA-200 (663-bitnega številka z 200 decimalnimi številkami). Zahtevala je več mesecov računskega časa na platformi z 80 procesorji AMD Opteron.
 - Januarja 2010 je bila oznanjena faktorizacija RSA-768. V napadu v skupni dolžini več kot dveh let je sodelovalo več sto računalnikov.
- **Dvojni Moorov zakon faktorizacije:** strošek faktorizacije poljubnega celega števila pade za faktor 2 vsakih 9 mesecov (za 2 vsakih 18 mesecov zaradi napredka kriptoanalyze, in za 2 vsakih 18 mesecov zaradi napredka strojne opreme).
- **Dolžine ključev RSA, ki nudijo ustrezeno varnostno zaščito**
 - Leto 2010 ($\lambda = 75$): 1112 bitov
 - Leto 2020 ($\lambda = 82$): 1387 bitov
- Obstaja **učinkovit kvantni algoritem** za faktorizacijo celih števil, ki ga je izumil Peter Shor

Diskretni logaritem

- **Osnovni matematični koncept:** Za dani element h končne ciklične multiplikativne grupe $(\mathbb{Z}_n)^\times = \{0, 1, 2, \dots, n-1\}$ z n elementi, ki jo po modulu n napenja **multiplikativni generator g** , najdi celo število k , tako da velja

$$h = g^k \pmod{n}.$$

Najmanjši nenegativni k se imenuje **diskretni logaritem** (*discrete logarithm*) od h glede na osnovo g in je označen z $k = \log_g(h)$.

- Katerikoli celi števili k_1 in k_2 za kateri velja $g^{k_1} \pmod{n} = g^{k_2} \pmod{n}$ sta, po definiciji, **kongruenčni po modulu n** (t.j. imata enak ostanek po deljenju z n) in pripadata istemu kongruenčnemu razredu (*congruence classes*) po modulu n .
- **Zgled:** poiščimo rešitev enačbe $3^k \equiv 13 \pmod{17}$. Ena izmed rešitev je $k=4$. Toda to ni edina rešitev. Ker je $3^{16} \pmod{17} = 1$, velja $3^{4+16n} \pmod{17} = 13 \times 1^n \equiv 13$. Torej ima enačba neskončno mnogo rešitev, njihova splošna oblika pa je $k = 4 + 16n$. Ker je $m=16$ najmanjše pozitivno celo število, ki zadosti enačbi $3^m \pmod{17} = 1$ (t.j. 16 je red števila 3 v grupi $(\mathbb{Z}_{17})^\times$), so rešitve oblike $k = 4 + 16n$ edine rešitve diskretnega logaritma v $(\mathbb{Z}_{17})^\times$. Rešitev navadno zapišemo v obliki $k \equiv 4 \pmod{16}$.

Izmenjava ključa

(Diffie–Hellman key exchange)

Diffie-Hellmanov (D-H) algoritem za izmenjavo ključev je eden od prvih praktičnih primerov izmenjave ključev na področju kriptografije. Metoda omogoča dvema stranema, ki nimata predhodnih stikov, da skupaj vzpostavita skupni privatni ključ skozi javno odprt (kriptografsko nevaren) komunikacijski kanal. Ta ključ se lahko nato uporabi za šifriranje sporočili z uporabo simetričnega kodirnika.

Algoritem:

- Alica in Bob skupaj izbereta praštevilo p in osnovo g (ti dve vrednosti sta lahko znani vsem).
- Alica izbere naključno naravno število a in pošlje Bobu $g^a \text{ mod } p$.
- Bob izbere naključno naravno število b in pošlje Alici $g^b \text{ mod } p$.
- Alica izračuna $k = (g^b)^a$.
- Bob izračuna $k = (g^a)^b$.

Samo a , b in $g^{ab} \text{ mod } p = g^{ba} \text{ mod } p$ morajo ostati skrivni. Vse ostale računske entitete p , g , $g^a \text{ mod } p$, in $g^b \text{ mod } p$ so javno dostopni. Ko Alica in Bob izračunata skupni skrivni ključ, ga lahko uporabita kot ključ za simetrični kodirnik.

Seveda so za zagotovitev varnosti potrebne velike vrednosti a , b in p , saj je, na primer, enostavno preizkusiti vse možne vrednosti $g^{ab} \text{ mod } 23$ (obstaja največ 22 takih vrednosti, četudi sta a in b velika).

Diskretni logaritem vs. faktorizacija celih števil

- Čeprav sta računanje diskretnih logaritmov in faktorizacije celih števil matematično različna problema, si delita številne skupne lastnosti:
 - Oba problema sta računsko težka (v smislu klasičnega računanja, torej mimo kvantnih računalnikov).
 - Za oba obstaja učinkovit kvantni algoritem.
 - Algoritmi za enega izmed problemov so pogosto prilagojeni še za drug problem.
 - Oba problema imata računsko zelo učinkovita inverzna problema (t.j. množenje celih števil oz. diskretno potenciranje) kar je ugodno za namene kriptografije.

Eliptične krivulje

- Zaradi nenehnega napredka strojne opreme in kriptografskega znanja se velikost kriptirnih ključev algoritma RSA nenehno povečuje, kar zmanjšuje njegovo uporabnost na platformah z omejeno zmogljivostjo.
- Alternativo predstavlja kriptografija, ki temelji na diskretnem logaritmu eliptičnih krivulj (*Elliptic Curve Discrete Logarithm*). Eliptična krivulja je v splošnem definirana z enačbo

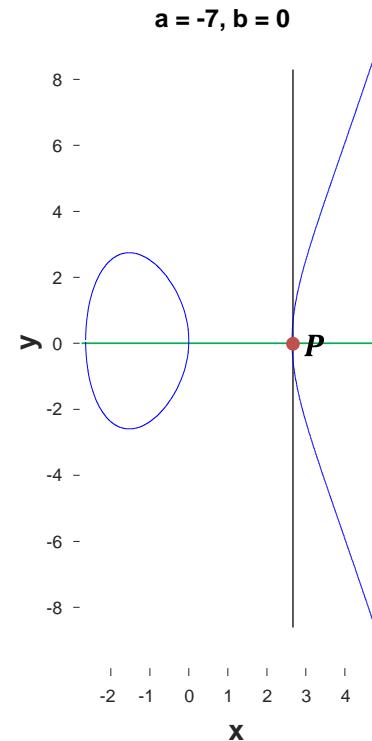
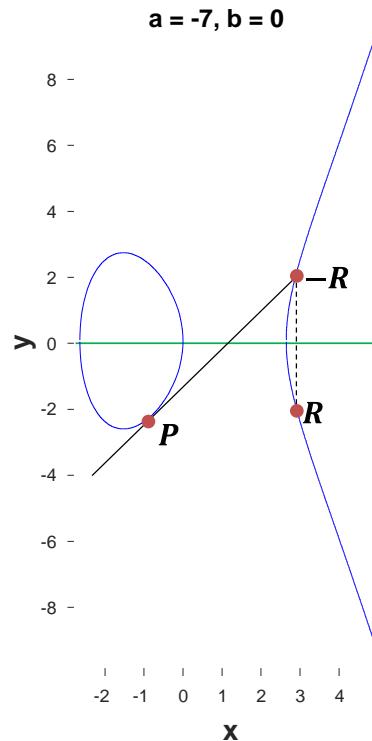
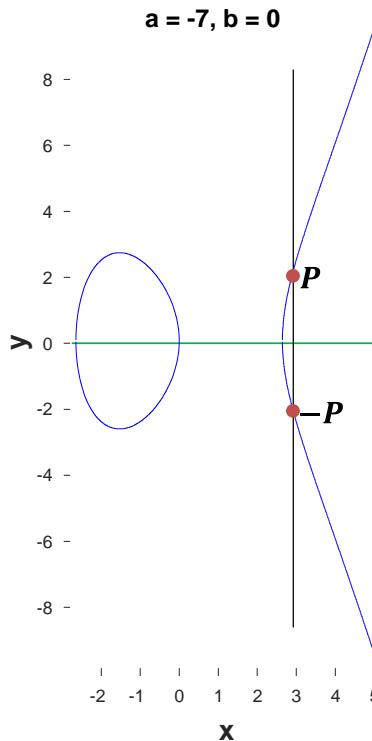
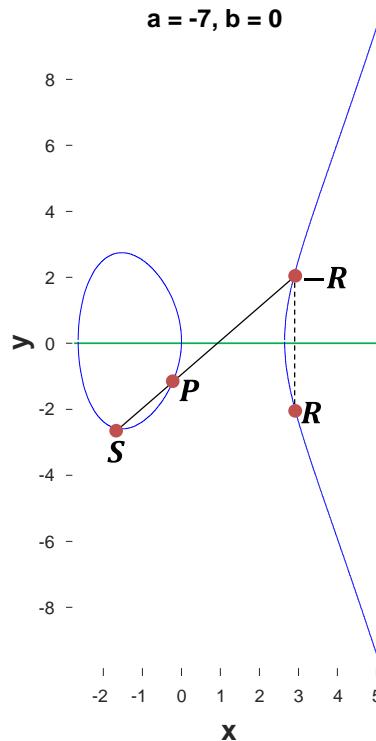
$$y^2 = x^3 + ax + b$$

kjer, zaradi varnostnih razlogov, $4a^3 + 27b^2 \neq 0$ (v nasprotnem primeru obstajajo na krivulji singularnosti, npr. presečišča krivulje same s sabo ali pa izolirane točke).

- Navadno krivuljo omejimo na končni obseg (*finite field*) $GF(p^m) = \mathbb{Z} \text{ mod } p$, torej obseg vseh celih števil po modulu p , kjer je p veliko praštevilo.

Eliptične krivulje

$$y^2 = x^3 + ax + b,$$



Grafični prikaz seštevanja točk S in P na eliptični krivulji. Seštevek obeh točk lahko enoznačno opišemo s tretjo točko R , ki je preslikava presečišča krivulje s premico skozi S in P preko osi x : $S+P=R$ (primer a) na sliki). Točko $-P$ definiramo kot preslikavo točke P preko osi x (primer b)). Če se premica samo dotika krivulje na točki, se ta točka šteje dvakrat (primer c, na sliki: $P+P=R$). Če je premica vzporedna osi y , definiramo tretjo točko 0 kot točko "v neskončnosti" (primera b) in d) na sliki: $P+(-P)=0$). Natanko en od teh pogojev velja za vsak par točk na eliptični krivulji. Rezultat seštevanja torej vedno leži na izvorni eliptični krivulji

(vir: <http://www.certicom.com/index.php/ecc-tutorial>).

Eliptične krivulje

Preko operacije seštevanja točk lahko definiramo tudi množenje točke s skalarjem kot $2P = P + P$.

Za vsako točko P na eliptični krivulji v obsegu $GF(p^m)$ velja

$$\lim_{k \rightarrow \infty} kP = \mathbf{0}$$

torej za vsako točko obstaja par skalarjev α in β , $\beta > \alpha$, za katere velja $\alpha P = \beta P$. Iz tega sledi $cP = \mathbf{0}$ kjer je $c = \beta - \alpha$. Najmanjši c , ki izpolnjuje ta pogoj se imenuje red točke P (*order of the point*).

Za zagotovitev varnosti je potrebno izbrati takšno krivuljo in fiksno točko F na njej, da je **red točke F veliko praštevilo**. Namreč, če je red točke F n -bitno praštevilo, potem je za izračun faktorja k , iz kF potrebnih vsaj $2^{n/2}$ računskih operacij. Ta lastnost naredi eliptične krivulje privlačne, saj lahko z njihovo uporabo zagotovimo enako stopnjo kriptografske varnosti ključev in podpisov pri precej manjših ključih kot z RSA.

Izbira parametrov $a, b, GF(p^m)$ in c se običajno ne opravi ločeno za vsakega udeleženca v komunikacijski, saj gre za štetje števila točk na krivulji, ki je zamudno in težavno s stališča implementacije. Zaradi tega je več institucij za standarde objavilo varne domene parametrov eliptičnih krivulj:

- NIST, Priporočila za uporabo eliptičnih krivulj za potrebe vlade ZDA (<http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf>)
- SEC, SEC 2: Priporočila za izbiro parametrov eliptičnih krivulj (http://www.sec.org/download/aid-386/sec2_final.pdf)

Algoritem ECMQV

- Algoritem ECMQV [Menezes 1995] je razširitev Diffie–Hellmanovega algoritma na problem eliptičnih krivulj. Njegovo ime je sestavljeno iz akronima za eliptične krivulje (EC) in začetnic njegovih avtorjev Menezes, Qu in Vanstone.
- **Algoritem:** Alice in Bob se javno dogovorita o eliptični krivulji in o stalni točki F na tej krivulji. Alice izbere skrivno naključno celo število A_k , ki je njen skrivni ključ, in objavi točko na krivulji $A_P = A_k F$ kot njen javni ključ. Bob sledi enakemu postopku: izbere skrivno število B_k in objavi svoj javni ključ $B_P = B_k F$.
- Recimo, da želi Alice poslati sporočilo Bobu na kriptografsko varen način. V ta namen lahko preprosto izračuna $k = A_k B_P$ in rezultat uporabi kot skrivni ključ za konvencionalni simetrični kodirnik. Bob lahko izračuna isto število $B_k A_P$ saj velja:

$$B_k A_P = B_k (A_k F) = A_k (B_k F) = A_k B_P$$

- Varnost izmenjave temelji na dejstvu, da je težko izračunati faktor k , če sta dani samo točki na krivulji F in kF . Problem diskretnih logaritmov eliptičnih krivulj velja za NP-težek problem.

Primerjava asimetričnih kodirnikov

- Intelektualne pravice in zaščita:** Patent algoritma RSA je potekel. Določeni koraki algoritmov z eliptičnimi krivuljami so še vedno patentno zaščiteni, čeprav nekateri avtorji zagotavljajo, da lahko izmenjavo ključev implementiramo brez kršitve patentnih pravic. Vsekakor je potrebna pri implementaciji kodirnikov, ki temeljijo na eliptičnih krivuljah dobršna mera previdnosti.
- Primerjava kriptografske varnosti ključev RSA in ključev eliptičnih krivulj (Elliptic curve cryptography – ECC) (*vir: Gura et al: Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs Cryptographic Hardware and Embedded Systems - CHES 2004*)

Pričakovani čas, potreben za uspešen napad (v MIPS-letih)	Velikost ključa RSA (v bitih)	Velikost ključa ECC (v bitih)
10^4	512	106
10^8	768	132
10^{11}	1024	160
10^{20}	2048	210
10^{78}	21000	600

Primerjava asimetričnih kodirnikov

- Kanadsko podjetje Certicom (<http://www.certicom.com>) je opravilo številne študije in promocije kodirnikov z eliptičnimi krivuljami (ECC). Tabela podaja nekaj njihovih primerjav časovnih zahtevnosti algoritmov ECC z algoritmom RSA.

Korak algoritma	RSA – 1024 bitni ključ (v ms)	ECC – 163 bitni ključ (v ms)
Generiranje ključa	4708,3	3,8
Podpisovanje dokumenta	228,4	2,1 (ECNRA) 3,0 (ECDSA)
Verifikacija podpisa	12,7	9,9 (ECNRA) 10,7 (ECDSA)

Simetrični kodirniki

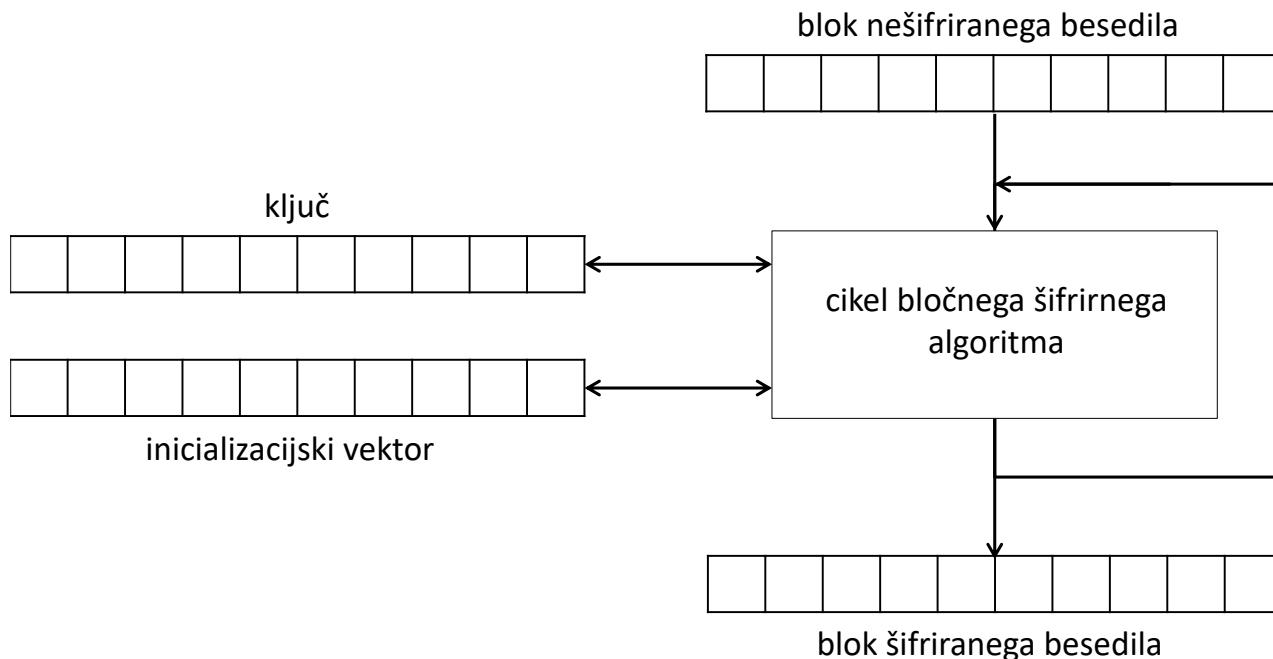
Bločni simetrični kodirniki

(Block Ciphers)

- Bločni kodirnik je algoritem za šifriranje blokov podatkov (nekodiranega besedila) s **konstantno dolžino** v naključno zaporedje znakov iste dolžine (šifrirano besedilo) z uporabo drugega bloka podatkov, ki se imenuje **ključ**.
- V simetričnem načinu kodiranja se uporablja **isti ključ za enkripcijo in dekripcijo**. Ker je dešifriranje brez poznavanja ključa računsko zelo zahtevna operacija so šifrirani podatki zavarovani pred nezaželeno prisluškovanjem, vsaj dokler je ključ poznan samo oddajniku in sprejemniku.
- Bločni kodirnik je navadno sestavljen iz **več krogov preprostih kriptografskih operacij**. Šifrirni ključ se najprej razširi na več podključev (*key schedule*), podključi pa nato v več različnih ciklih premešajo in kodirajo vhodni podatkovni blok.
- Tipično so omenjena kodiranja realizirana s pomočjo bitnih operacij XOR. Zaradi svoje visoke zmogljivosti se bločni kodirniki pogosto uporabljajo v različnih aplikacijah varne komunikacije, kot so osnovno šifriranje podatkov v internetnih protokolih (IPsec in SSL / TLS), brezžične komunikacije in upravljanje z digitalnimi pravicami.

Bločni simetrični kodirniki (Block Ciphers)

- Simetrični bločni kodirniki temeljijo na konceptu **zmede** in **difuzije vhodnih podatkov**.
- Tipični predstavniki implementacije teh konceptov so **mreže za zamenjavo in permutacijo** (*Substitution-Permutation Network - SPN*) in **mreže Feistel** (*Feistel network*).

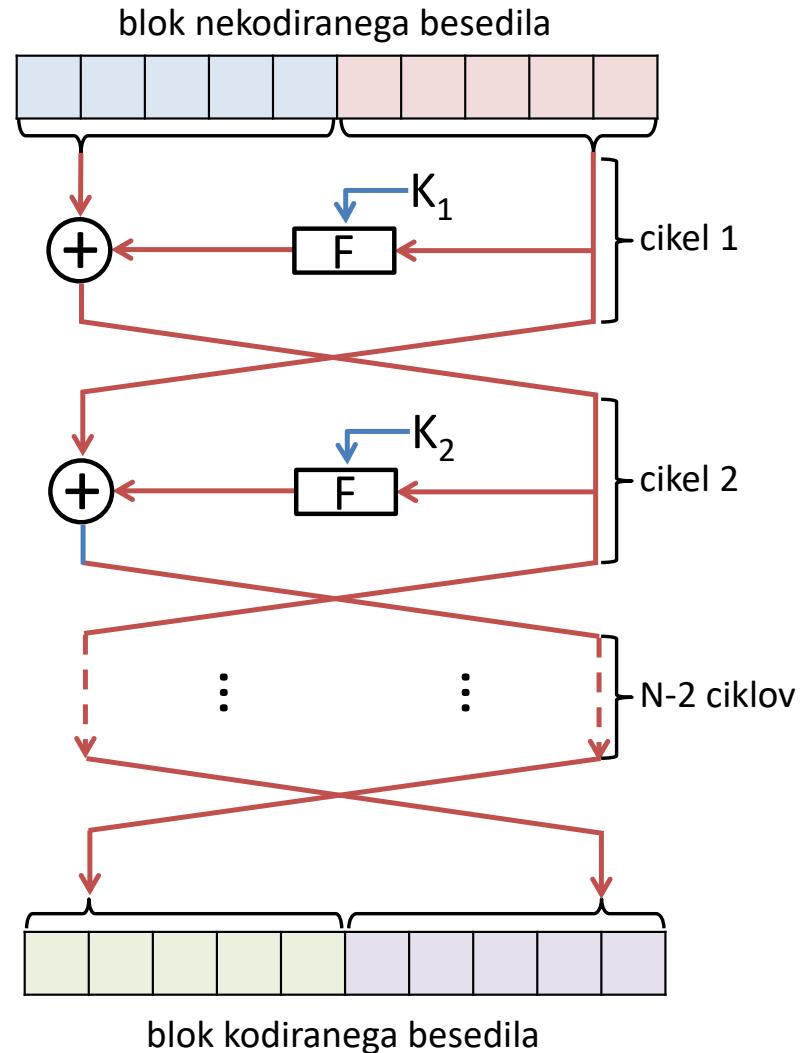


Mreže Feistel (Feistel network)



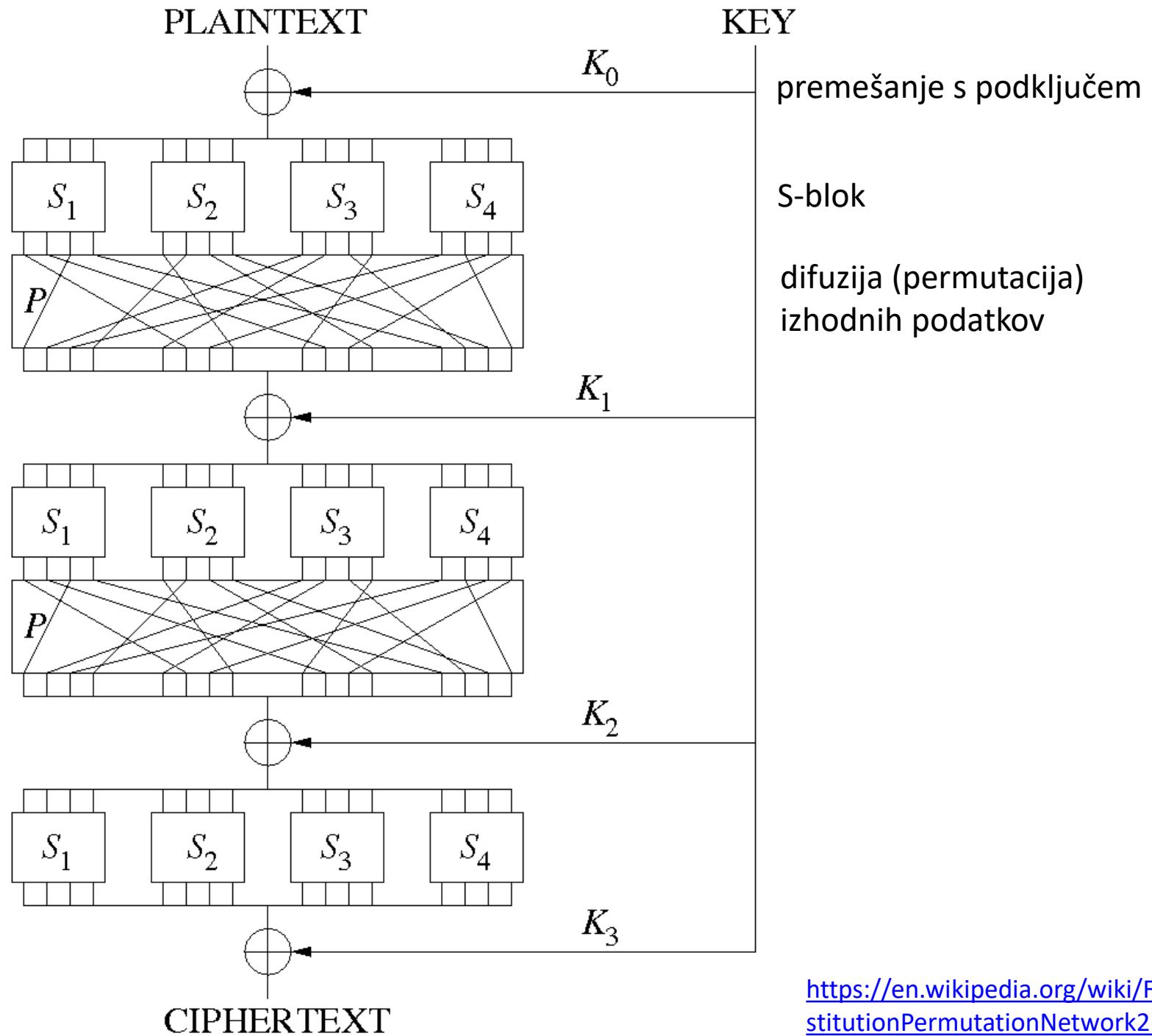
Horst Feistel

- V i-tem ciklu mreže Feistel se desna polovica vodnih podatkov (označena z X_i) posreduje na vhod funkcije F . Slednja prejme kot vhodni parameter tudi podključ K_i , pogosto pa je sestavljena iz korakov za mešanje ključa (**S-blokov**) in linearne transformacije.
- Izhod iz funkcije F , označen z Y_i , je preko bitne operacije XOR združen z levo polovico vodnih podatkov X_{i-1} .
- Izhod cikla je nato oblikovan z zamenjavo vodnih podatkov X_i in kodiranih podatkov $X_{i-1} \oplus Y_i$, kjer je " \oplus " bitni XOR.



Mreže za zamenjavo in permutacijo

(Substitution-Permutation Network - SPN)



<https://en.wikipedia.org/wiki/File:SubstitutionPermutationNetwork2.png>

Mreže za zamenjavo in permutacijo (Substitution-Permutation Network - SPN)

- Med šifriranjem s kodirnikom SPN, se običajno vhodni podatki v vsakem ciklu premešajo s podključem, nato pa vstopijo v blok za zamenjavo (*Substitution boxes*) ali krajše **S-blok** (*S-box*).
- Izhodi S-blokov se dodatno spremenijo z linearo transformacijo, katere namen je širitev (difuzija) statističnih učinkov kodiranja v izhodnih podatkih (dobri kodirniki zagotavljajo izhodne podatke, ki navidezno in statistično ne odstopajo od naključno tvorjenih zaporedij znakov).
- Dešifriranje je sestavljenoto iz inverzne linearne transformacije, inverznega S-bloka in mešanja podključev v obratnem vrstnem redu.
- Z namenom, da se ohrani enaka pretočnost podatkov tako pri kodiranju kot pri dekodiranju se v SPN tipično izpustijo linearne transformacije zadnjega cikla.

S-blok (S-box)

- Vsak S-blok velikosti $m \times n$ opravlja nelinearno preslikavo m hodnih bitov v n izhodnih bitov kar ustvarja navidezno zmedo v podatkih.
- Implementiran je lahko v obliki poizvedbene tabele (*lookup table*) z 2^m besedami dolgimi n bitov. Običajno so tabele fikse, kot npr. v Data Encryption Standard (DES), vendar nekateri kodirniki tabele ustvarijo dinamično, na podlagi ključa (npr. algoritma Blowfish in Twofish).
- Učni primer S-bloka predstavlja tabela z 6×4 -biti kodirnika DES (S_5):

DES		Srednji štirje vhodni biti															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Zunanja vhodna bita	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Glede 6-bitno vhodno zaporedje se 4-bitni izhod določi tako, da s pomočjo zunanjih dveh bitov (prvega in zadnjega bita) določi vrstica tabele, sredinski štirje biti pa določajo stolpec tabele. Na primer, vnos "100100" ima zunanja bita "10" in notranje bite "0010", kar ustreza izhodu "0001".

S-blok (S-box)

- Različni S-bloki v bločnem kodirniku lahko imajo različne preslikave ali pa se enaka preslikava uporablja v vseh S-blokih kodirnika.
- Kot edina nelinearna komponenta v obeh predstavljenih kodirnih arhitekturah, so bili **S-bloki predmet intenzivnih varnostnih študij** saj se je sumilo, da imajo **vgrajene ranljivosti**, ki so znane oblikovalcem kodirnih algoritmov. Pokazano je bilo, da temu ni tako in da so **S-bloki skrbno zasnovani v smeri maksimalne odpornosti na diferencialno kriptoanalizo**. Druge raziskave so še pokazale, da lahko tudi majhne spremembe S-bloka bistveno oslabijo varnost bločnih kodirnikov.
- Nedavne pobude v kriptografiji so osredotočene na razvoj novih standardov bločnih kodirnikov. Kot naslednik Data Encryption Standard (DES) se je pojavil algoritem Rijndael, ki ga je ameriški Nacionalni inštitut za standarde in tehnologijo novembra 2001 izbral za **Advanced Encryption Standard (AES)**

AES – Advanced Encryption Standard

1. **Razširitev ključa (KeyExpansion)**: Iz kriptografskega ključa se po posebnem postopku izračunajo ključi posameznih ciklov.

2. Začetni cikel

- Prištevanje ključa (AddRoundKey): vsak zlog v stanju algoritma AES se preko bitne operacije xor kombinira s istoležnim zlogom ključa.

3. Osrednji cikel

- Substitucija zlogov (SubBytes): nelinearna sustitucija zlogov v kateri se vsak zlog stanja algoritma AES nadomesti z ustrezno vrednostjo v poizvedbeni tabeli (S blok).
- Premik vrstic (ShiftRows): vsaka vrstica stanja se premakne ciklično za določeno število mest v levo. Ta korak, skupaj z mešanjem stolpcev, zagotavlja učinek difuzije podatkov.
- Mešanje stolpcev (MixColumns): vsak stolpec stanja AES se s pomočjo obrnljive linearne transformacije preslika v izhoden stolpec, kjer vrednosti vseh štirih vhodnih zlogov vplivajo na vrednost vsakega izmed štirih izhodnih zlogov.
- Prištevanje ključa (AddRoundKey): enako kot v začetnem ciklu.

4. Zaključni cikel (brez mešanja stolpcev)

- Substitucija zlogov (SubBytes): enako kot v osrednjem ciklu.
- Premik vrstic (ShiftRows): enako kot v osrednjem ciklu.
- Prištevanje ključa (AddRoundKey): enako kot v začetnem in osrednjem ciklu.

AES – Advanced Encryption Standard

R I J N D A E L
C I P H E R

128-bit version (data block and key)
Encryption

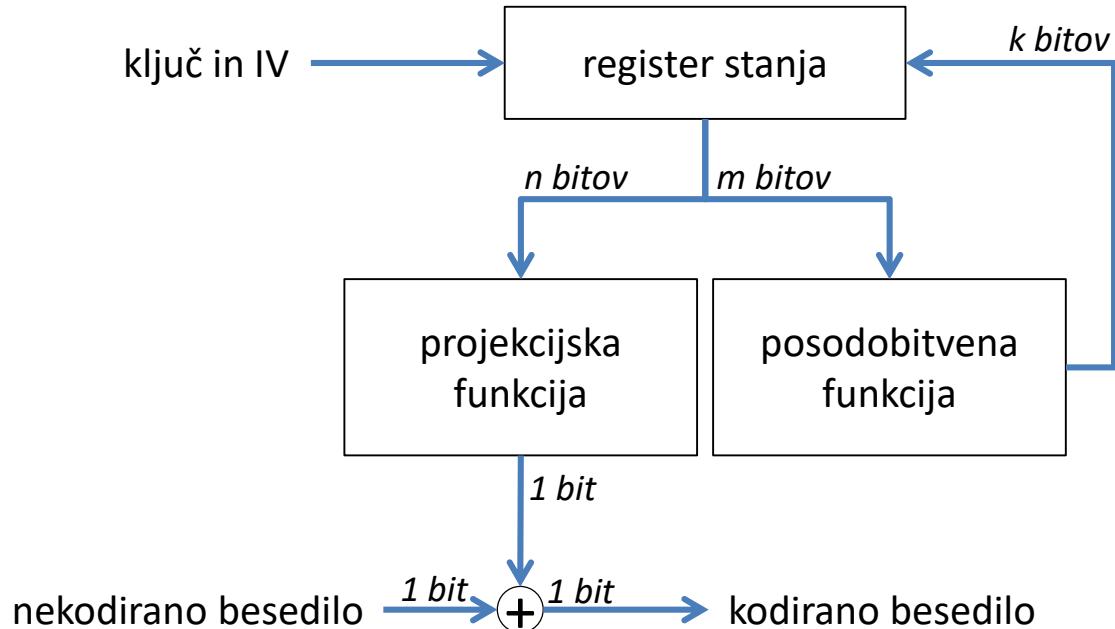
Animation by **Enrique Zabala** / Universidad ORT / Montevideo / Uruguay
e-mail: ezabala@adinet.com.uy

Pretočni simetrični kodirniki (Stream ciphers)

- Pretočni kodirnik je simetrični kodirnik, ki se uporablja v primerih, ko je količina podatkov vnaprej neznana, podatkovni tok pa je časovno nezvezen oz. heterogen.
- Kodirnik ustvarja zaporedje kriptografsko varnih bitov, imenovanih bitno zaporedje ključa (*key stream*). To zaporedje se nato na bitni ravni, z uporabo operacije xor, kombinira z nekodiranim ali kodiranim besedilom.
- Postopka enkripcije in dekripcije sta zaradi bijektivnosti operacije xor popolnoma enaka.
- Posodobitev internega stanja je lahko od prejetega besedila odvisna ali neodvisna. Glede na to delimo pretočne kodirnike v dve večji skupini:
 1. **sinhroni pretočni kodirniki** (*synchronous stream cipher*):
 - Interno stanje kodirnika se spreminja neodvisno od prejetega kodiranega oz. nekodiranega besedila
 - Pošiljatelj in prejemnik morata biti popolnoma sinhronizirana, drugače dekripcija ni mogoča.
 - Če se med pošiljanjem kodiranih podatkov podatkovnemu toku dodajo ali odvzamejo dodatni znaki se sinhronizacija izgubi.
 2. **Asinhroni oz. kodirniki s samodejno sinhronizacijo** (*asynchronous or self-synchronising stream cipher*):
 - Posodabljujo interno stanje glede na poslane/prejete znake kodiranega besedila.

Pretočni simetrični kodirniki (Stream ciphers)

- 1 Osnovna topologija pretočnega kodirnika je sestavljena iz registra za shranjevanje internega stanja kodirnika (shift register), ki se na začetku določi s pomočjo ključa in inicializacijskega vektorja (IV). Vsebina registra se redno posodablja preko **funkcije za posodobitev stanja**.



- 2 Naslednja komponenta je **nelinearna projekcijska funkcija**, ki vzame del vsebine ali celotno vsebino registra s stanjem kodirnika in jo združi v en sam bit psevdonaključnega zaporedja kodirnika. Ta bit se nato preko operacije xor združi s trenutno obdelovanim bitom nekodiranega oz. kodiranega besedila.

- 3 Po operaciji xor se, preko funkcije za posodobitev, posodobi interno stanje kodirnika in postopek kodiranja se ponovi za naslednji bit nekodiranega oz. kodiranega besedila.