Programování na kvantovém počítači

Adéla Mládková[1], Aleš Horák[2], Jan Provazník[3], and Jiří Šáda[4]

[1] *AG Štěpánská, Praha, a.mladek@email.cz*
[2] *Střední průmyslová škola Třebíč, horak.alda@seznam.cz*
[3] *Gymnázium Voděradská, j.a.n.provaznik@email.cz*
[4] *Gymnázium Voděradská, jiri.sada@gmail.com*

Abstrakt

V našem miniprojektu jsme se podívali na rozdíly mezi klasickým a kvantovým počítáním. Demonstrovali jsme je pomocí Deutschova algoritmu v prostředí IBM Q, zpřístupňujícím reálný kvantový počítač a dokázali tak, že kvantový počítač může být efektivnější než ten klasický.

1 Úvod

Ještě donedávna se všechny informace zpracovávaly na počítačích reprezentujících data jako jedničky a nuly. V některých případech však operace dělané na klasických počítačích trvají příliš dlouho a s příslibem kratšího času potřebného k provedení výpočtů začal výzkum počítačů využívajících kvantovou mechaniku. V tomto miniprojektu se pokusíme demonstrovat na příkladu Deutschova algoritmu, že kvantové počítače mohou být v některých úlohách efektivnější.

2 Klasické počítání

2.1 Binární soustava

Běžné počítače pracují s informacemi, reprezentovanými pomocí bitů. Bit je jednotka informace, která může nabývat hodnoty 1 nebo 0, z čehož plyne název binární soustava – existují v ní jen dvě hodnoty. Obvyklý způsob reprezentace přirozeného čísla n v binární soustavě funguje takto

$$n = 2^0 c_0 + 2^1 c_1 + 2^2 c_2 + 2^3 c_3 + 2^4 c_4 + \ldots \quad (n)_2 = \ldots c_4 c_3 c_2 c_1 c_0\,, \tag{1}$$

$$25 = 1 + 2 \cdot 0 + 4 \cdot 0 + 8 \cdot 1 + 16 \cdot 1 \qquad (25)_2 = 11001\,, \tag{2}$$

tedy číslo n rozepíšeme do mocnin čísla 2 a jednotlivé bity $c_i \in \{0, 1\}$ určují, zda se daná mocnina v rozkladu vyskytuje. Naopak, pokud chceme vědět hodnotu v binární soustavě zapsaného čísla v desítkové soustavě, sečteme všechny mocniny čísla 2, jímž odpovídá bit s hodnotou 1.

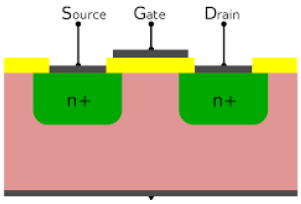
2.2 Tranzistor

Tranzistor má tři kontakty, z nichž dva jsou pracovní (Source, Drain) a jeden je řídící (Gate). Na Drain přivedeme proud a na kontakt Source připojíme zařízení. Aby do zařízení tekl proud, musíme přivést napětí do řídícího kontaktu Gate. Uvnitř tranzistoru se tak udělá tenká vrstva vodivého pásu, kudy mohou procházet elektrické náboje. Gate tedy kontroluje vodivost obvodu. Skládáním tranzistorů lze vytvářet fyzické realizace logických operací (viz níže).

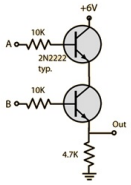
2.3 Základní operace v binární soustavě

V binární soustavě pracujeme s bity - číslicemi které mohou nabývat pouze hodnot 1 nebo 0, a tím reprezentovat logickou pravdu či nepravdu. S nimi pak lze provádět logické operce pomocí logických hradel (operací). Hodnota na výstupu je funkcí všech hodnot na vstupu.

Pracujeme například s dvoubitový (binárními) logickými hradly AND, OR, XOR, NAND a NOR, jejichž pravdivostní hodnoty zobrazuje následujuící tabulka



Obrázek 1: Ze Source teče proud do Drain pouze tehdy, teče-li proud skrze Gate, Zdroj: www.wikipedia.cz



Obrázek 2: Napojení dvou tranzistorů v do série vytváří logickou operaci AND mezi jejich Gaty

A	B	AND	OR	XOR	NAND	NOR
1	1	1	1	0	0	0
0	1	0	1	1	1	0
1	0	0	1	1	1	0
0	0	0	0	0	1	1

Obrázek 3: Binární operace na dvou bitech

3 Kvantové počítání

3.1 Kvantová mechanika

Kvantová mechanika vznikla počátkem 20. století a zabývá se popisem částic, pro něž neplatí klasické fyzikální zákony. Charakteristické kvantové jevy jsou třeba to, že některé veličiny kvantové fyziky (například energie atomu anebo spin) mohou nabývat hodnot pouze z diskrétní množiny - říkáme, že jsou *kvantované*.

Dalším jevem je *superpozice*, která vyjadřuje, že kvantový objekt může existovat ve více stavech (např. energetických) najednou. Podstatný je také vliv měření (pozorovatele), které silně zaahuje do systému. Měření způsobuje kolaps superponovaného stavu měřeného objektu do jedné konkrétní možné realizace.

Dalším z důležitých jevů je *kvantová provázanost* (entanglement), kdy při měření jedné částice (v našem případě qubitu) okamžitě získáme i hodnotu druhé částice, bez toho abychom jí měřili. Provázání je podstatné pro kvantové programování, šifrování a kvantovou teleportaci.

3.2 Pravděpodobnosti a qubit

Qubit je komplexní vektor ve tvaru $|q\rangle = \alpha|0\rangle + \beta|1\rangle$, kde α , $\beta \in C$ a $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ a $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$, jsou tzv. *bazické stavy*, přičemž požadujeme, aby platilo $|\alpha|^2 + |\beta|^2 = 1$. Pravděpodobnost naměření $|q\rangle$ ve stavu $|0\rangle$ je $P(|q\rangle \to |0\rangle) = |\langle 0|q\rangle| = |\alpha|^2$ a ve stavu $|1\rangle$ je $|\beta|^2$.

3.3 Základní operace a hradla

Protože jsou qubity vektory, lze na na ně působit vektorovými transformacemi, například rotacemi, zrcadlením anebo škálováním (které v případě kvantových operací zakážeme). Všechny operace jsou reverzibilní jako důsledek toho, že nemění velikost vektoru.

- Operace na jednom qubitu

  1. Operátor X převádí $|0\rangle$ na $|1\rangle$ a $|1\rangle$ na $|0\rangle$

$$\text{QNOT} \equiv X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \tag{3}$$

  2. Hadamardův operátor H ze stavů $|0\rangle$, $|1\rangle$ udělá superpozice

$$H = \begin{pmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}\end{pmatrix}, \qquad H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{4}$$

- Operace na dvou qubitech

1. CNOT pracuje na kvantovém registru skládajícím se ze dvou qubitů. CNOT funguje tak, že když první qubit je $|1\rangle$ tak se druhý qubit se neguje.

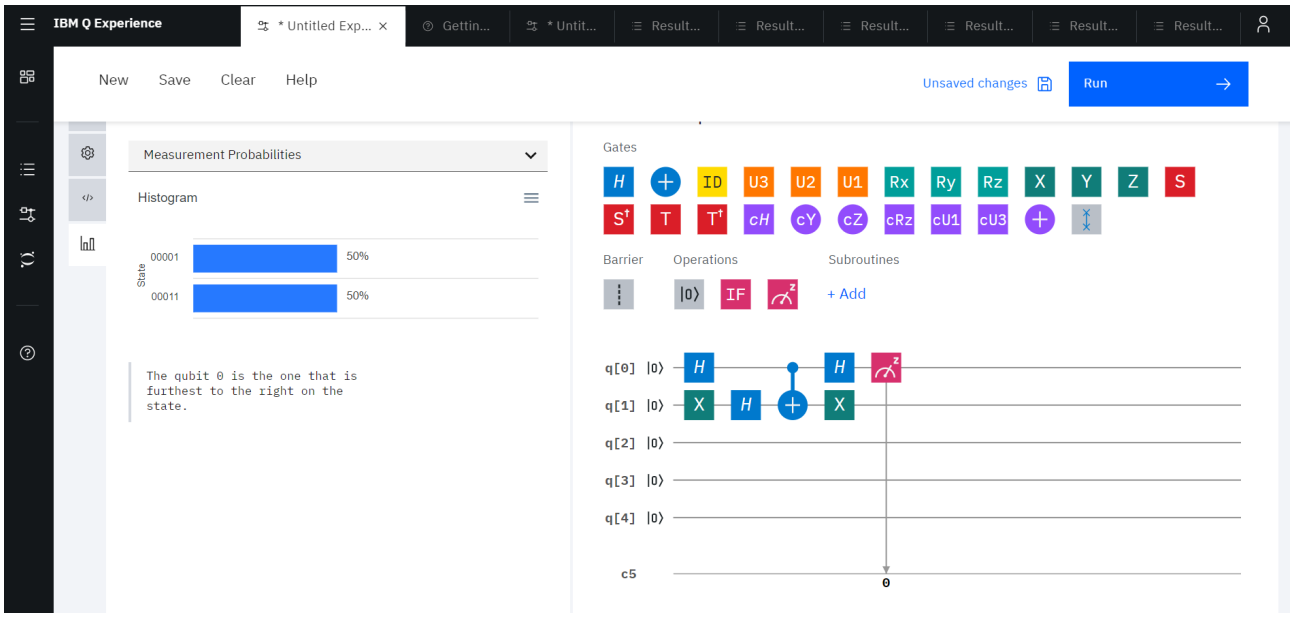
$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\|q_1\rangle$	$\|q_2\rangle$	CNOT
$\|0\rangle$	$\|0\rangle$	$\|00\rangle$
$\|0\rangle$	$\|1\rangle$	$\|01\rangle$
$\|1\rangle$	$\|0\rangle$	$\|11\rangle$
$\|1\rangle$	$\|1\rangle$	$\|10\rangle$

Obrázek: Hradlo CNOT v maticové podobě (vlevo), logická tabulka pro CNOT (vpravo).

4 Programování na kvantovém počítači IBM Q

Společnost IBM zpřístupnila webové rozhraní, kde lze vytvářet a testovat kvantové algoritmy a odeslat je k vyhodnocení na reálném kvantovém počítači.



Obrázek 4: Webové rozhraní IBM Q

4.1 Deutschův algoritmus

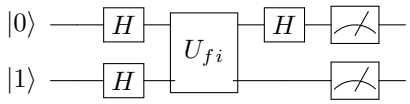
Deutschův algoritmus je nejjednodušší příklad toho, že kvantové počítače jsou schopny výpočtů, které efektivitou přesahují schopnosti klasických počítačů. Nejjednodušeji to lze demonstrovat na problému, kdy chceme určit, zda je nějaká jednobitová funkce konstantní nebo balancovaná, aniž bychom věděli co se uvnitř funkce děje. V běžném počítači bychom potřebovali provést dvě měření, pro hodnotu 1 a 0. Pomocí Deutschova algoritmu je však možné tuto informaci zjistit pouze jedním měřením.

Algoritmus si předvedeme na čtyřech jednoduchých funkcích. Existují čtyři základní funkce, které lze provést na jednom bitu:

1. Konstantní
   - f_1 nezávisle na vstupu vrátí 0
   - f_2 nezávisle na vstupu vrátí 1
2. Balancovaná
   - f_3 vrátí hodnotu vstupního bitu
   - f_4 vrátí opačnou hodnotu vstupního bitu

Testovací obvod je vyobrazen na Obrázku 5, kam za U_{f_i} dosadíme kvantovou implementaci testované funkce (postupně všechny čtyři) a budeme měřit výsledné hodnoty obou qubitů. Testované funkce můžeme implementovat pomocí následujících základních kvantových hradel

- $U_{f_1} |q_1\rangle |q_2\rangle = |q_1\rangle |q_2\rangle$
- $U_{f_2} |q_1\rangle |q_2\rangle = |q_1\rangle X(|q_2\rangle)$
- $U_{f_3} |q_1\rangle |q_2\rangle = \text{CNOT}\ (|q_1\rangle |q_2\rangle)$
- $U_{f_4} |q_1\rangle |q_2\rangle = \text{CNOT}\ (|q_1\rangle X(|q_2\rangle))$

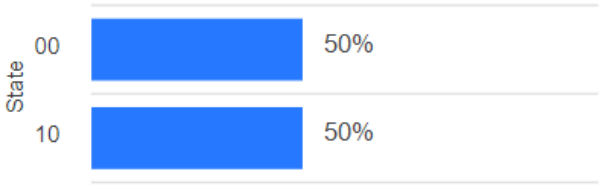


Obrázek 5: Schéma Deutschova algoritmu

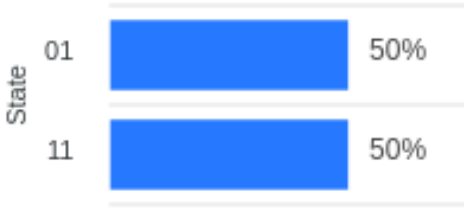
4.2 Experimentální výsledky

Na kvantovém počítači jsme pomocí Deutschova algoritmu spustili zmíněné čtyři základní funkce. Jejich výsledky vypadaly takto:

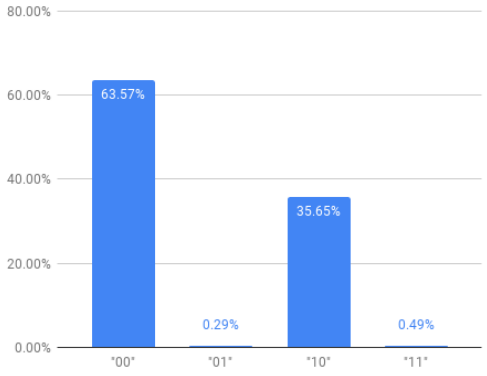
- U_{f_1}, U_{f_2} možné stavy $|00\rangle$, $|01\rangle$
- U_{f_3}, U_{f_4} možné stavy $|10\rangle$, $|11\rangle$



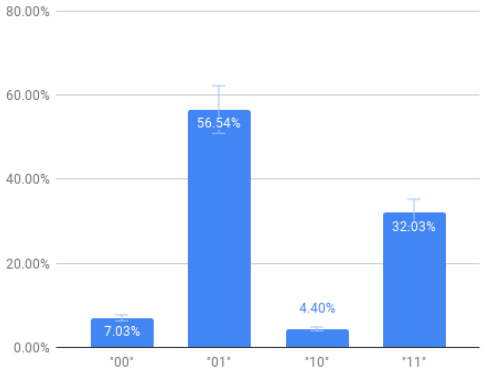
Obrázek 6: Výsledek simulací na IBM Q pro funkce U_{f1}, U_{f2}



Obrázek 7: Výsledek simulací na IBM Q pro funkce U_{f3}, U_{f4}



Obrázek 8: Reálné výsledky Deutschova algorismu z kvantového počítače IMB Q pro funkce U_{f1}, U_{f2}



Obrázek 9: Reálné výsledky Deutschova algorismu z kvantového počítače IMB Q pro funkce U_{f3}, U_{f4}

Z experimentálních i statistických výsledků lze pozorovat, že pro zjištění, zda je funkce balancovaná nebo konstantní stačí změřit první qubit. Pokud má hodnotu 1, funkce je balancovaná, v opačném případě je konstantní. Na reálném kvantovém počítači si můžeme všimnout, že nastaly stavy, které jsme nepředpověděli, jelikož v realitě se potýkáme se šumem a chybami kvantového počítače.

5 Závěr

Seznámili jsme se základy kvantového počítání a osvojili si Deutschův algoritmus, pomocí kterého můžeme u libovolné funkce na jenom bitu zjistit, zda-li je konstatní nebo balancovaná efektivněji než na běžném počítači.

6 Reference

http://www.physics.muni.cz/ tomtyc/kvantovka.pdf
http://fyzika.jreichl.com/main.article/view/731-kvantova-mechanika