Quark 2005/6

Svet kvantovej fyziky V

Kvantové počítače

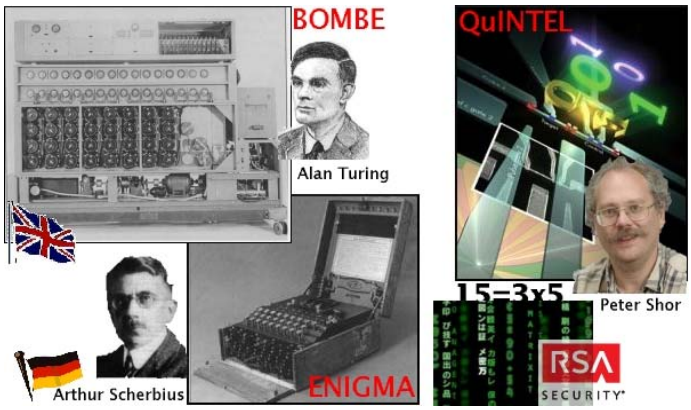
Nie každý si uvedomuje, že pomenovanie počítača nie je odvodené od slova počítať náhodou. Či už ide o hru, alebo o bežné písanie, počítač na našom stole vykonáva výpočty. V zlomku sekundy zvládne milióny jednoduchých matematických operácií, ktoré by nám trvali niekoľko rokov. Napriek tomu však na niektoré úlohy nestačí, resp. čas, ktorý by potreboval presahuje nielen dobu jeho funkčnosti, ale aj vek vesmíru. V tejto časti si povieme niečo o tom, ako využitie kvantových vlastností môže pomôcť pri rýchlejšom riešení problémov.

Za pohodlnosťou, ktorú nám ponúka obyčajná kalkulačka, je skrytý pomerne dlhý vývoj. Ľuďom bolo už dávnejšie jasné, že základné operácie ako sčítanie, násobenie, atď. sa dajú rozložiť do jednoduchších procedúr. V škole sa naučíme konkrétne postupy ako sčitovať, násobiť a deliť ľubovolne veľké čísla. Ide viac-menej o zautomatizované činnosti, ktoré nepotrebujú príliš veľké premýšľanie. A naviac dajú sa zrealizovať pomocou fyzikálnych procesov. Inými slovami, aj obyčajné fyzikálne objekty (nielen človek) sú správnym používaním schopné vykonávať výpočty. Všetci poznáme vzťah *dráha = čas* x *rýchlosť*. Ak chceme vypočítať 45x200, tak nám stačí udeliť napr. autíčku počiatočnú rýchlosť 200 mm/s a počkať 45 sekúnd. Zmeraním prejdenej dráhy v milimetroch získame výsledok násobenia. Autíčko vykonalo výpočet za nás. Samozrejme nejde o ideálny spôsob, pretože zanedbávame trenie a nepresnosti merania času, dĺžky a rýchlosti. Avšak podobným, aj keď oveľa komplikovanejším spôsobom, za nás „fyzika“ vykonáva výpočty aj v kalkulačkách, alebo počítačoch.

Ak chceme postaviť prístroj, ktorý by vyriešil danú úlohu za nás, tak musíme poznať postup, ako sa k výsledku dostať. Takýto postup výpočtu, resp. návod na riešenie nejakého problému sa odborne nazýva *algoritmom*. Algoritmus napísaný v reči, ktorej rozumie napr. počítač, sa nazýva *programom*. Ani si to neuvedomujeme, ale algoritmy vykonávame celkom bežne. Všetci to poznáme: *pri prechode cez cestu pozri vľavo, pozri vpravo, pozri vľavo a ak nič nejde, tak prejdi cez cestu*. Ide o vžitú procedúru, ktorú si vôbec neuvedomujeme. Každý algoritmus je vlastne receptom, alebo návodom, ako vyriešiť nejaký problém, prípadne splniť nejakú úlohu. Tento návod je zložený z postupnosti inštrukcií, pokynov, ktoré je potrebné uskutočniť. Tieto inštrukcie sú akýmisi základnými operáciami prístroja, ktorý ich má splniť, resp. ide o operácie, ktoré je tento prístroj schopný vykonávať.

Turingov stroj

Prvé mechanické prístroje, ktoré nás odbremenili od základnej aritmetiky, zostrojili ešte v 19. storočí. Tieto „primitívne“ kalkulačky sa postupne zdokonaľovali a nakoniec predbehli svojich stvoriteľov nielen v rýchlosti výpočtu. Mechanické zariadenia sa začali používať aj v takých oblastiach ako je napríklad kryptografia. Nemecká armáda (ešte pred druhou svetovou vojnou) používala na zakódovanie a dekódovanie tajných rozkazov prístroj zvaný *ENIGMA*. Žiadny človek nie je schopný dostatočne rýchlo rozlúštiť takto zakódované posolstvo. Počas vojny anglická tajná služba zamestnávala okrem iných aj matematika *Alana Turinga*, ktorý prišiel s myšlienkou postaviť stroj proti stroju. Alan Turing zovšeobecnil pojem výpočtu a vymyslel všeobecný model stroja, ktorý je schopný vykonať ľubovoľný algoritmus. Inými slovami, riešenie akejkoľvek úlohy (aj dekódovania) sa dá uskutočniť pomocou opakovania zopár základných operácií tohto prístroja. Napríklad vieme, že pri sčítaní, násobení, delení, nám stačí používať jednoduché sčitovanie čísel menších ako desať, t.j. „počítanie na prstoch“.



Obr. 1. Turingov stroj a Shorov algoritmus. *Anglický prístroj Bombe počas II. svetovej vojny pomáhal odhaľovať nemecké rozkazy zašifrované zariadením Enigma. Mechanický šifrovací prístroj Enigma bol vyrobený Arthurom Scherbiusom a mal plnú dôveru nemeckého hlavného štábu. Na anglickej strane však tajná služba zamestnávala matematika Alana Turinga, ktorý položil základy modernej informatiky a vymyslel koncepciu univerzálneho počítacieho zariadenia, tzv. Turingov stroj. Turing sa nemalou mierou podielal aj na zotrojení zariadenia Bombe. V dnešnej dobe je situácia podobná. Metóda RSA šifrovania sa považuje za bezpečnú, avšak zotrojenie kvantového počítača (QuIntelu) by túto situáciu zmenilo podobným spôsobom ako kedysi Bombe.*

Idea *Turingovho stroja* bola významným krokom pre rozvoj *informatiky*, ktorá skúma vlastnosti algoritmov, t.j. postupy ako nejaké základné a jednoduché operácie využiť na vykonanie tých najkomplikovanejších úloh. Jedným z hlavných poznatkov informatiky je, že Turingov stroj je ten najvšeobecnejší možný typ počítacieho stroja, pretože dokáže vykonať akýkoľvek výpočet. Moderný počítač nedokáže viac ako Turingov stroj. Vývoj v posledných rokoch však toto tvrdenie informatiky trochu poopravil, resp. upresnil. Záver je dnes taký, že Turingov stroj je najvšeobecnejší možný stroj pracujúci na princípoch *klasickej fyziky*.

Cesta ku kvantovým počítačom

V súvislosti s univerzalitou Turingovho stroja je prirodzenou otázka, ktoré operácie sa skutočne realizujú vo fyzikálnych systémoch, resp. procesoch? Na túto otázku teória okolo Turingových strojov v princípe zodpovedať nemôže. Turingov stroj pracuje s logickými operáciami na abstraktných bitoch informácie. Tieto operácie sú odvodené z našej každodennej skúsenosti, ktorá žiaľ nezahŕňa skúsenosť s kvantovým svetom. Základné funkcie Turingovho stroja sú odpozorované z tzv. *klasickej fyziky*. Kvantové zákony sú však úplne iné

a základnou otázkou je, či v oblasti počítania umožňujú niečo nové a lepšie, alebo nie. Druhou (praktickejšou) otázkou je potom cena, ktorú nás využitie týchto nových operácií bude stáť, resp. či skutočne tieto nové možnosti potrebujeme, alebo nie. Tieto „nové“ prístroje pracujúce na princípoch kvantovej fyziky nazývame *kvantovými počítačmi*.

Jedna z dôležitých úloh, ktorú riešia počítače, je *simulácia* fyzikálnych systémov. Hlavnou motiváciou na postavenie prvého počítača *ENIAC* bola práve simulácia atómovej bomby. Asi ani nenájdeme „lepší“ príklad na užitočnosť simulácií. Ich úlohou je nepriamo zistiť vlastnosti a správanie sa fyzikálnych systémov. Nepriamo v tomto prípade znamená, že na to, aby sme niektoré dôsledky otestovali, nemusíme nechať atómovú bombu skutočne vybuchnúť. Podobne pri konštrukciách lietadiel, áut, alebo aj pri výrobe liekov sú simulácie veľmi užitočným nástrojom, ktorý v neposlednom rade šetrí aj peniaze. Simulácie však nie sú vôbec jednoduchým problémom. Ani pre počítač. V osemdesiatich rokoch známy americký fyzik *Richard Feynman* postrehol fakt, že kvantové systémy nebudeme nikdy schopní efektívne simulovať na našich počítačoch. Príčinou je princíp superpozície, vďaka ktorému počet parametrov narastá s veľkosťou systému príliš rýchlo. Ak chceme modelovať kvantové systémy, tak potrebujeme kvantový počítač, t.j. zariadenie, ktoré má vlastnosti kvantového sveta zabudované priamo vo svojich funkciách.

Čo je zložité?

Vráťme sa teraz späť k algoritmom. Začnime úlohou, v ktorej treba spočítať čísla od 1 do 100. Ako prvý spôsob každému zíde na um jednoducho postupne sčítať všetkých 100 čísiel, t.j. 1+2+3+4+5+... . Existuje však aj jednoduchší postup. Stačí si uvedomiť, že platí 100+1=99+2=98+3=...=51+50=101. Vďaka tomuto faktu celkový súčet dostaneme, ak vynásobíme 50x101. Podobným trikom spočítame čísla od 1 po hocijaké číslo N. Štandardným postupom násobenia, ktorý sme sa naučili v škole, prídeme k výsledku oveľa skôr ako postupným sčítavaním N čísiel. Tento príklad nám ukazuje, že na riešenie aj jedného problému existuje viacero algoritmov, z ktorých jeden môže byť jednoduchší ako ten druhý.

Zložitosť samotného problému potom určujeme podľa času, ktorý je potrebný na vykonanie toho najrýchlejšieho známeho algoritmu. Inými slovami čím je problém zložitejší, tým na jeho vyriešenie potrebujeme uskutočniť viac krokov. Táto definícia zložitosti plne zodpovedá našej intuitívnej predstave. Typickým príkladom zložitého problému je rozklad na prvočísla. Momentálne síce nevieme povedať nakoľko je tento problém zložitý, pretože stále ešte nepoznáme ten najrýchlejší algoritmus na jeho riešenie. Veríme však tomu, že ide o úlohu veľmi ťažkú. Dokonca sme na tomto fakte založili aj bezpečnosť našich informácií na internete (RSA šifra). Skúste si samy zistiť násobkom ktorých dvoch prvočísiel je 403. Opačná úloha, t.j. zistiť výsledok 19x23, je oproti tomu hračkou.

Rozklad na prvočísla

Prvočísla majú vcelku výsadné postavenie medzi číslami. Ide o čísla, ktoré nie sú deliteľné ničím iným ako sebou samým a jednotkou, napr. 1,2,3,5,7,11,13,17,19,...

Každé iné číslo sa dá jednoznačne napísať ako súčin prvočísiel, napr. 182=2x7x13. Úlohou rozkladu na prvočísla je nájsť pre ľubovoľné číslo príslušné prvočísla. Ako na to?

Najjednoduchšou metódou je skúšať dané číslo N postupne deliť všetkými prvočíslami menšími ako odmocnina z N. Takýto postup je však pre veľké čísla veľmi náročný a čas potrebný na jeho realizáciu by trval veky. Poznáme síce niektoré rýchlejšie algoritmy, ale zrýchlenie nie je príliš citeľné, resp. je ďaleko od toho, aby sme problém považovali za jednoduchý. Ak za sekundu počítač vykoná miliardu operácií, tak čas potrebný na rozloženie 50-ciferného čísla je zhruba 23 miliónov rokov. Bola vypísaná odmena 200 000 dolárov pre toho, kto nájde rozklad istého 617-ciferného čísla. Zatiaľ sú však tieto peniaze v bezpečí, pretože kvantový počítač je ešte v nedohľadne a dnešným počítačom by to trvalo dobu, ktorá ďaleko presahuje odhadovanú existenciu vesmíru.

Jeden z trikov na riešenie rozkladu na prvočísla využíva výsledky z teórie čísiel týkajúce sa tzv. *modulárnej aritmetiky*, ktorej prototypom sú napríklad hodiny, ktoré ukazujú čísla iba od 1 po 12. Ak chceme na hodinách vyrátať 7+8, tak výsledok dostaneme tým že ručičku zo 7 posunieme o 8 hodín doprava. Po tomto sčítaní bude ručička ukazovať 3 hodiny. Učene takýto výpočet zapíšeme nasledovne 7+8=3 (mod 12) a čítame 7 plus 8 rovná sa 3 modulo 12. Nájdenie prvočísiel pre N je ekvivalentné problému nájsť periódu postupnosti čísiel $f(k)=y^k \pmod N$ pre nejaké $y<N$ a pre k=0,1,2,3,... . Napríklad postupnosť 1,4,3,6,1,4,3,6,1,4,3,6,.... má periódu rovnú 4. Ak nájdeme periódu r, tak hľadané prvočísla získame nájdením najväčších spoločných deliteľov (NSD) pre dvojice $\left[y^{r/2}+1, N\right]$ a $\left[y^{r/2}-1, N\right]$. Táto úloha je jednoduchá na vypočítanie a získaní delitelia sú hľadanými prvočíslami v rozklade N. Dôvod, prečo tento postup funguje, žiaľ nie je úplne jednoduchý a využíva výsledky *teórie čísiel*. Uveďme si iba ako príklad rozklad čísla N=15 a zvoľme y=7. Funkcia $f(k)=y^k \pmod N$ v tomto prípade predstavuje číselnú postupnosť 1,7,4,13,1,7,4,13,1, ..., ktorá má periódu 4, t.j. $7^4 \pmod{15}=1$. Teraz určíme $y^{r/2}=7^2=49$ a NSD(48,15)= 3, NSD(50,15)=5. Tým dostávame rozklad 15=5x3.

Namiesto hľadania prvočísiel je našou úlohou nájsť periódu. Na analýzu periodických funkcií sa v matematike využívajú metódy tzv. *Fouriérovej analýzy*. Napríklad pri analýzach zvuku pomocou tejto metódy vieme určiť tóny (frekvencie) z akých sa daný zvuk skladá. V našom prípade ide o diskrétnu periodickú funkciu a úlohou je uskutočniť tzv. *diskrétnu Fouriérovu transformáciu*, ktorá robí niečo veľmi podobné. Vďaka tejto transformácií získame hľadanú informáciu o perióde. Realizácia diskrétnej Fouriérovej transformácie je práve tým zložitým krokom, ktorý na klasických počítačoch nevieme uskutočniť efektívne.

Kvantové bity

Základnou jednotkou, s ktorou pracujú dnešné počítače je jeden *bit*, ktorý predstavuje najmenšie možné množstvo informácie. Jeden bit môže mať jednu z dvoch hodnôt: *nula*, alebo *jedna*. Akúkoľvek inú informáciu vieme vyjadriť pomocou týchto základných jednotiek informácie. Iba 8 bitov nám stačí na vyjadrenie ľubovoľného čísla od 0 po 255. Zo školy tento prepis čísla do bitov poznáme ako prepis z desiatkovej sústavy

do dvojkovej. Napríklad číslo 25 zapisujeme ako **11001**, čo znamená **1**.16+**1**.8+**0**.4+**0**.2+**1**.1=25. Akýkoľvek výpočet vieme zapísať ako postupnosť operácii, ktoré menia hodnoty jednotlivých bitov, čím vstupnú informáciu menia na výstupnú, t.j. na výsledok. Napríklad pri výpočte **101**x**101=11001** (5x5) postupujeme presne tak isto ako pri obyčajnom násobení, iba používame vzťahy pre súčin 0x0=1x0=0x1=0, 1x1=1 a pre súčet 0+0=0, 0+1=1+0=1, 1+1=10. Ľubovoľný výpočet vieme napísať ako súčet, prípadne súčin núl a jednotiek. Ak k tomu prirátame operáciu negácie, ktorá zmení hodnotu bitu na opačnú a kopírovanie, tak máme v podstate všetky operácie, ktoré počítač vykonáva.

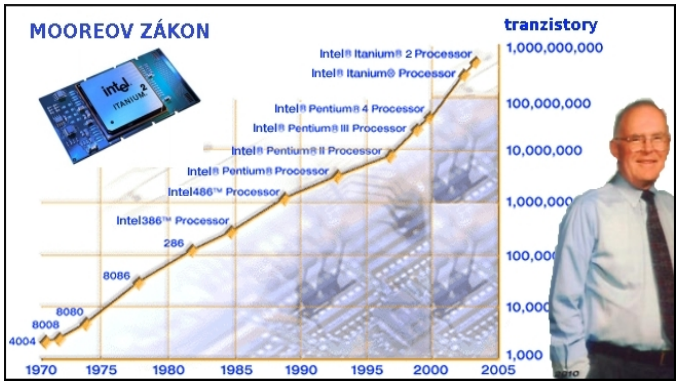
Fyzikálne jeden bit zodpovedá dvom (perfektne rozlíšiteľným) stavom fyzikálneho systému, napr. svieti/nesvieti, prúd tečie/netečie, atď. Každý z týchto stavov zodpovedá inej informácii (alternatíve). Môžeme povedať, že informácia je zakódovaná do stavu fyzikálneho systému. Ako sme si ale povedali v tretej časti tohoto seriálu, v kvantovej teórii okrem týchto dvoch stavov existuje aj stav, ktorý je ich superpozíciou, t.j. niečo medzi nulou a jednotkou. Takéto niečo v klasickom prípade neexistuje. Hovoríme o *kvantovom bite*, ktorý je základným stavebným kameňom kvantového počítača. Stavy, ktoré sú niečo medzi nulou a jednotkou, sú charakteristické tým, že pri zisťovaní, o ktorú hodnotu vlastne ide, s istou pravdepodobnosťou dostávame obidva výsledky, t.j. aj nula, aj jedna. Pomocou viacerých kvantových bitov vieme zapísať ľubovoľné číslo presne tým istým spôsobom ako pomocou „obyčajných“ klasických bitov. Opäť však platí, že okrem toho, že máme ľubovoľné čísla, tak máme aj ich superpozície. Inými slovami existujú stavy, ktoré čiastočne obsahujú všetky čísla. Napríklad pre osem kvantových bitov všetky čísla od 0 po 255.

Pri riešení úlohy nás niekedy zaujíma hodnota nejakej funkcie pre dané číslo. Napríklad pri rozklade N na prvočísla nás zaujíma, či nejaké číslo delí N, alebo nie. Inými slovami zaujíma nás podiel. Problémom je, že výpočet tej istej funkcie (podielu) potrebujeme urobiť veľmi veľa krát. Vlastne pre každé číslo zvlášť až pokým nenarazíme na ten správny výsledok. S kvantovými bitmi však akoby túto funkciu počítame naraz pre všetky hodnoty, pretože máme stavy, ktoré sú superpozíciou všetkých hodnôt. Táto vlastnosť kvantových výpočtov sa nazýva *kvantovým paralelizmom*, pretože kvantový počítač súčasne (paralelne) vykonáva všetky úlohy.

Proces výpočtu si môžeme predstaviť tak, že v algoritmoch na výpočet funkcie používame akúsi čiernu skrinku. Napríklad pri hľadaní rozkladu táto čierna skrinka počíta podiel čísla, ktoré rozkladáme a nášho tipu na výsledok. Ak dostaneme celočíselný výsledok, tak vieme, že sme hľadaný rozklad našli. Ak nie, tak pokračujeme ďalším tipom. Kvantová mechanika nám umožňuje iba pri jednom použití „kvantovej“ čiernej skrinky zistiť hodnoty funkcie pre všetky potenciálne tipy. Jediným problémom zostáva, že tento výsledok má tiež formu superpozície a je stále skrytý v kvantovom svete. Musíme vykonať meranie, ktoré náhodne vyberie iba jedinú hodnotu, čím sme zdanlivo tam, kde sme boli aj predtým. Nie je to však úplne tak. Nás totižto zaujíma tento výpočet iba pre hodnotu, pre ktorú je výsledkom delenia celé číslo bezo zvyšku. O tejto hodnote vieme, že je jediná. Vhodnou manipuláciou (ďalším kvantovým výpočtom) sa pokúsime nastaviť pravdepodobnosti tak, aby sme zvýhodnili práve tento prípad. Vďaka tomu s vysokou pravdepodobnosťou nameriame hodnotu, ktorú hľadáme. Kvantová mechanika za nás skúsi paralelne všetky možnosti a vyberie tie, ktoré hľadáme.

Deutch-Jozsov algoritmus

Teraz si ukážeme jednoduchý príklad ako kvantový paralelizmus funguje v praxi. Ide o nasledovnú úlohu. Majme čiernu skrinku, ktorá počíta funkciu f a nás zaujíma, či je táto funkcia konštantná, alebo nie. Pre jeden bit existujú 4 rôzne čierne skrinky, z ktorých dve počítajú konštantné funkcie, a dve nie: (1) *identita* $f(0)=0$, $f(1)=1$, (2) *negácia* $f(0)=1$, $f(1)=0$, (3) *nula* $f(0)=f(1)=0$ a (4) *jednotka* $f(0)=f(1)=1$. Klasicky nemáme inú možnosť ako vyskúšať čiernu skrinku pre každý možný vstup, t.j. pre obidve hodnoty bitu. Použitím kvantového bitu nám však stačí čiernu skrinku použiť jediný raz a zmeraním jednoznačne zistiť, či je funkcia konštantná, alebo nie. Nevieme, síce povedať, o ktorú čiernu skrinku konkrétne ide, ale to ani nie je našou úlohou. Kľúčovým momentom je použitie kvantového bitu v superpozičnom stave nuly a jednotky. Tento stav obsahuje v rovnakej miere informáciu o obidvoch hodnotách, t.j. pri zisťovaní hodnoty získavame obidva výsledky s rovnakou pravdepodobnosťou. Naviac ešte použijeme jeden ďalší kvantový bit, ktorý je v stave s hodnotou nula aj na začiatku a aj na konci výpočtu. Dá sa povedať, že má funkciu akéhosi „katalyzátora“ výpočtu. Nebudeme popisovať detaily tohoto algoritmu. V skutočnosti nejde o veľmi užitočný algoritmus, ale slúži ako veľmi jednoduchá ukážka toho, že kvantový počítač vďaka superpozícii, resp. paralelizmu, umožňuje efektívnejšie riešenie problému.



Obr. 2. Mooreov zákon pre procesory Intel.
Krivka znázorňuje exponenciálny trend vývoja moderných počítačov.

Shorov algoritmus

V roku 1994 *Peter Shor* našiel spôsob ako efektívne uskutočniť diskrétnu Fouriérovu transformáciu za pomoci kvantového počítača. Ako sme si povedali, táto transformácia je tým komplikovaným krokom v probléme rozkladu na prvočísla. Vďaka tomuto výsledku sa kvantový počítač stal jedným z hlavných potenciálnych nepriateľov niektorých kryptografických protokolov a 200 000 dolárov si začalo hľadať svojho majiteľa. Shorov algoritmus je doposiaľ jedným z najväčších výsledkov dosiahnutých v oblasti kvantových počítačov. V princípe vieme, ako Shorov algoritmus vykonať. Chýba nám už len samotný kvantový počítač. Hlavným problémom pri jeho zostrojení je prílišná citlivosť kvantových

systémov na vplyvy okolia. Tieto interakcie v podstate ničia kvantovú superpozíciu medzi kvantovými bitmi, ktorá, ako sme si povedali, je podstatná. V laboratóriách po celom svete sa testujú tie najrôznejšie fyzikálne systémy, ktoré by nám mali slúžiť ako kvantové bity. Neustále sme svedkami drobných pokrokov, ale na skutočný prielom v technológii ešte len čakáme. V roku 2001 bol Shorov algoritmus zrealizovaný v laboratóriách IBM, kde sa podarilo experimentálne nájsť prvočíselný rozklad čísla 15.

Vývoj súčasných počítačov je extrémne rýchly. V roku 1965 americký elektronik *George Moore* odpozoroval, že počet tranzistorov (základných operačných jednotiek) na jednom elektronickom čipe sa zdvojnásobuje zhruba každého polroka. V tom čase bol tento počet rovný zhruba 50 tranzistorom. Vyslovil domnienku, že tento trend bude pokračovať aj naďalej. Dnes sme svedkami toho, že jeho predpoveď stále platí a hovoríme o tzv. *Mooreovom zákone*. Ak sa však pozrieme obdobia okolo roku 2015, tak zistíme, že miniaturizácia dosiahne úroveň atómov. Na týchto rozmeroch však kraľuje kvantová mechanika a platia iné zákony než na aké sme zvyknutí. Vďaka tejto perspektíve je skúmanie potenciálu kvantových počítačov vlastne nevyhnutnosťou. Skôr či neskôr v praxi narazíme na podivuhodný svet kvantovej teórie. V tejto časti sme sa snažili načrtnúť základné idey kvantového počítania. Pre detailnejšie vysvetlenie odporúčame populárnu knižku od *Georga Johnsona* s názvom „Zkratka napříč časem“. V ďalšej časti sa vrátime späť k fyzike a povieme si niečo o systémoch, z ktorých by sa mohol budúci kvantový počítač skladať.

MÁRIO ZIMAN