

Współczesna fizyka wydaje się przeciętnemu człowiekowi dziedziną bardzo abstrakcyjną, która ma niewiele wspólnego z problemami dzisiejszego świata. Cóż bowiem mogą wnieść do naszego życia wyniki eksperymentów nad wyziębionymi niemal do zera bezwzględnej atomami<sup>1)</sup> lub pojedynczymi fotonami – elementarnymi cząstkami przenoszącymi światło. Każdy zdaje sobie sprawę, że oczywiście mogą one mieć znaczenie fundamentalne dla zrozumienia, jak działa świat. Ale jednocześnie prawie każdy wątpi w to, że mogą one zmienić, albo chociaż ułatwić, nasze codzienne życie. Okazuje się jednak, że wszystkim tym fundamentalnym badaniom nad strukturą mikroświata przyświeca, obok idealistycznych postaw badacza odkrywcy, jedna bardzo praktyczna i bardzo efektowna wizja. Jest to wizja zbudowania komputera kwantowego – supermaszyny, która wykorzystując najbardziej wyszukane prawa fizyki kwantowej, będzie mogła wykonywać obliczenia dużo szybciej i sprawniej niż wszystkie dzisiejsze komputery razem wzięte. Skonstruowanie komputera kwantowego byłoby przełomem technologicznym porównywalnym z tym, jaki nastąpił w XX wieku po ujarzmieniu energii jądrowej. Bez wątpienia doprowadziłby on również do przełomu społecznego i kulturowego, bo otwarłyby się nowe, nieznane dzisiejszej cywilizacji, możliwości przetwarzania i przesyłania informacji. A fizycy mogliby się choć przez chwilę poczuć niemal tak jak Trurl i Klapaucjusz<sup>2)</sup>.



**Tomasz Sowiński** jest fizykiem na Wydziale Biologii i Nauk o Środowisku UKSW i w Centrum Fizyki Teoretycznej PAN. W 2005 roku skończył studia na Wydziale Fizyki Uniwersytetu Warszawskiego

w zakresie fizyki teoretycznej, a trzy lata później uzyskał tam stopień naukowy doktora. Od lat zajmuje się popularyzacją nauk przyrodniczych. W roku 2008 otrzymał tytuł Mistrza Popularyzacji Nauki „Złoty Umysł” w konkursie Prezesa Polskiej Akademii Nauk.

większej pojemności i żywotności, to koncepcyjna zasada ich działania nie zmieniła się od roku 1945, gdy skonstruowano pierwszy komputer ENIAC.

Wszystko opiera się na przetwarzaniu, przesyłaniu i przechowywaniu informacji w postaci ciągów dwóch znaków umownie nazywanych „zerem” i „jedyneką”. Choć może nie każdy zdaje sobie z tego sprawę, ale okazuje się, że te dwa znaki w zupełności wystarczą, aby zapisać każdą, nawet najbardziej wymyślną informację. Niezależnie od tego, czy mamy do czynienia z tekstem, rysunkiem czy muzyką, nasz

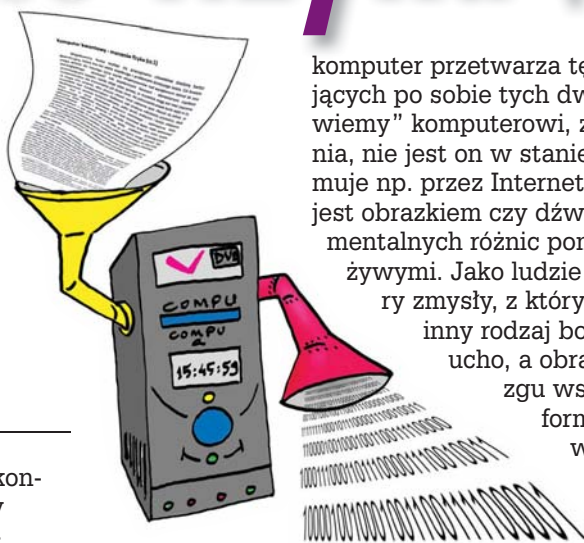
# Komputer kwantowy – marzenie fizyka (cz. 1)

Tomasz Sowiński

Czym miałyby być ten niemalże mityczny komputer kwantowy? Na czym polegać ma ta jego doskonałość i innowacyjność? Aby to dokładnie zrozumieć, proponuję temat rozbić na dwie części i w pierwszej z nich wrócić do korzeni informatyki i komputerów KLASYCZNYCH.

## ŚWIAT ZER I JEDYNEK

Wszystkie dotychczas skonstruowane komputery możemy nazwać maszynami klasycznymi. Choć inżynierowie wciąż projektują coraz szybsze procesory oraz pamięci o coraz



Dwa znaki w zupełności wystarczą, aby zapisać każdą, nawet najbardziej wymyślną informację

komputer przetwarza tę informację jako ciąg następujących po sobie tych dwóch znaków. Jeśli nie „podpowiemy” komputerowi, z jakimi danymi ma do czynienia, nie jest on w stanie stwierdzić, czy to, co otrzymuje np. przez Internet z innego miejsca na ziemi, jest obrazkiem czy dźwiękiem. Jest to jedna z fundamentalnych różnic pomiędzy maszynami a istotami żywymi. Jako ludzie jesteśmy wyposażeni w cztery zmysły, z których każdy jest wyczulony na inny rodzaj bodźca. Dźwięki słyszy nasze ucho, a obrazy widzą oczy. Dopiero w mózgu wszystkie docierające do nas informacje, zamienione wcześniej w sygnały elektryczne, są przetwarzane w jakiś tajemniczy sposób tak, że dokładnie wiemy, co czujemy i widzimy. Do dziś nie wiadomo, jak to dokładnie się dzieje i dlatego nie będziemy się w to

zagłębiali. Tym bardziej że nie jest to przedmiotem naszej dzisiejszej opowieści. W każdym razie komputer od samego początku dysponuje tylko „zerami” i „jedynekami”.

Jak to się dzieje, że każdą informację można przedstawić w tak trywialny sposób?

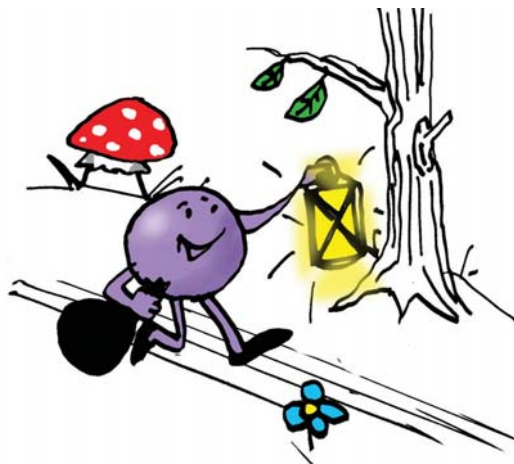
Pierwszym krokiem, aby tego dokonać, jest jej zamiana na ciąg liczb. Jest to operacja czasami bardzo skomplikowana i zależy od tego, z jaką informacją mamy do czynienia. Jeśli jest to tekst napisany w jakimś konkretnym alfabecie, to wystarczy każdej literze tego alfabetu przypisać jakąś konkretną liczbę, tworząc tabelę kodową. Następnie ciąg znaków tworzących tekst zamieniamy na odpowiadający mu ciąg liczb. Ponieważ tabelę kodową ustala się raz na zawsze, oczywiście można bez kłopotu ten ciąg liczb zamienić z powrotem na tekst. Można powiedzieć, że informacja zawarta w ciągu znaków i w ciągu liczb jest dokładnie taka sama. Jest jedynie inaczej zapisana, ale nie została w żaden sposób „zniszczona”.

W tym miejscu warto może podkreślić, że pierwszą taką tabelą kodową dla znaków alfanumerycznych był tzw. kod ASCII stworzony w latach 60. ubiegłego wieku w Bell Laboratories. Kodował on 128 różnych znaków: duże i małe litery alfabetu łacińskiego, cyfry, znaki przystankowe oraz kilka tzw. kodów sterujących pozwalających wydawać podstawowe komendy, np. drukarce. W tym kodzie napis „Młody Technik” to ciąg liczb: 77 108 111 100 121 84 101 99 104 110 105 107. W tym przykładzie od razu widać, jak bardzo kod ASCII był ułomny – nie miał np. polskich liter. Dlatego w dzisiejszych czasach używa się bardziej rozbudowanego systemu kodowania znaków, z których najpowszechniejszym staje się tzw. UNICODE. Pozwala on zakodować w liczby tekst zapisany w dowolnym znanym alfabecie świata (nawet chińskim czy suahili).

Zamiana innego typu informacji na ciąg liczb, np. rysunku, jest nieco trudniejsza, ale również możliwa. Zainteresowany Czytelnik na pewno znajdzie odpowiednie informacje na ten temat w Internecie. Dla nas najważniejsze jest to, że taka zamiana jest faktycznie możliwa i jest jednoznaczna; to znaczy jeśli raz ustalimy reguły, jak to się robi (w przypadku tekstu ustalimy po prostu tabelę kodową), to dwie identyczne informacje będą zakodowane identycznie, a dwie różne różnie.

## POZYCYJNE SYSTEMY LICZBOWE

Pozostaje nam już tylko zamienienie tych ciągów liczb na ciągi „zer” i „jedynek”. Ale tutaj sprawa jest już bardzo prosta, bo z pomocą przychodzi nam matematyka i abstrakcyjne pojęcie liczby. Otóż warto sobie uświadomić, że tak naprawdę liczba nie jest żadnym napisem ani słowem. Jest to pojęcie abstrakcyjne, którego definicja matematyczna wcale nie jest taka prosta. Możemy się jednak odwołać do naszej intuicji i powiedzieć, że liczby (rzeczywiste) to coś, co pozwala nam porównywać dwie wielkości tego samego rodzaju, np. wysokość budynku z wysokością czło-



Pojedynczy foton przenosi światło

wieka, masę ziemniaków z masą cukru. Sztuki jabłek ze sztukami gruszek, itp. Jeśli dwie wielkości są sobie równe, to przypisujemy im tę samą liczbę.

Dzięki temu, że pojęcie liczby jest abstrakcyjne, to jest uniwersalne. Doszła do niego wcześniej czy później praktycznie każda rozwinięta cywilizacja i wszystkie one rozumiały liczbę dokładnie w ten sam spo-

sób – jako narzędzie służące do porównywania ilości rzeczy. To co różniło wszystkie cywilizacje to sposób zapisu liczb. Można powiedzieć, że „co kraj, to obyczaj” i każdy zapisywał liczby po swojemu. Ten mój krótki wywód służy uświadomieniu, że choć sam zapis liczby jest bardzo często utożsamiany z samą liczbą, nie są to jednak te same rzeczy. Każdą liczbę możemy zapisać na bardzo wiele sposobów, a nadal to będzie ta sama liczba; np. liczbę dwanaście, czyli tuzin, możemy zapisać „po naszymu” 12. Możemy też zapisać po rzymsku XII albo po chińsku 拾貳.

Liczbę dwanaście możemy też zapisać za pomocą samych „zer” i „jedynek”! Jak to zrobić? Zastanówmy się najpierw może dlaczego dwanaście to w naszym systemie 12. Ciąg cyfr 12 oznacza: dwie jedności + jedna dziesiątka, tzn.  $2 \cdot 1 + 1 \cdot 10$ . Podobnie 145 oznacza: 5 jedności + 4 dziesiątki + 1 setka, czyli  $5 \cdot 1 + 4 \cdot 10 + 1 \cdot 100$ . Krótko mówiąc, napis „12” albo „145” to nic innego jak skrócony zapis rozkładu liczby na kolejne potęgi liczby dziesięć. To właśnie dlatego nasz system nazywa się systemem dziesiętkowym, bo kolejne cyfry w zapisie liczby oznaczają mnożnik kolejnych potęg dziesiątki (podstawy systemu liczbowego). Aby wszystko było ze sobą spójne i pozwalało zapisać dowolną liczbę tym sposobem, musimy mieć do dyspozycji dziesięć cyfr – znaków graficznych mówiących, przez ile mamy pomnożyć daną wielokrotność dziesiątki. Zwyczajowo oznaczamy je od 0 do 9. Ale równie dobrze moglibyśmy je oznaczać cyframi starożytnych Majów:

0	1	2	3	4
	•	••	•••	••••
5	6	7	8	9
	•	••	•••	••••

Starożytnych Majów przywołuję tu nie bez powodu, gdyż oni tak naprawdę używali systemu dwudziestkowego, tzn. rozkładali liczby na kolejne potęgi liczby dwadzieścia! Używali w związku z tym dwudziestu cyfr. Oprócz tych przedstawionych powyżej były jeszcze takie:

10	11	12	13	14
	•	••	•••	••••
15	16	17	18	19
	•	••	•••	••••

W tym systemie liczbę dwanaście zapisujemy za pomocą jednej cyfry:  $\equiv$ , gdyż dwanaście to po prostu  $\equiv$  jedności. Liczba 35 w systemie dwudziestkowym Majów to  $\bullet \equiv$ , bo trzydzieści pięć to piętnaście jedności i jedna dwudziestka, czyli  $\bullet \cdot 20 + \equiv$ .

Na marginesie mówiąc, Majowie zapisywali kolejne cyfry w pionie, a nie w poziomie, ale to oczywiście nie gra większej roli. Ważna jest koncepcja, która praktycznie niczym nie różni się od naszego systemu dziesiętkowego, nieprawdaż?

Takie systemy liczbowe, w których liczba reprezentowana jest przez ciąg cyfr stojących na konkretnych miejscach, nazywamy systemami pozycyjnymi. W zapisie liczby pozycja cyfry bowiem dokładnie mówi, o którą potęgę podstawy chodzi. Pozycja cyfry w liczbie liczona od prawej strony jest kolejną potęgą (zaczynając od zerowej) podstawy systemu liczbowego. Dla wprawy podajmy jeszcze „wygląd” liczby trzydzieści pięć np. w systemie piątkowym. W tym celu rozkładamy tę liczbę na kolejne potęgi liczby pięć. Mamy:  $35 = 1 \cdot 25 + 2 \cdot 5 + 0 \cdot 1 = 1 \cdot 5^2 + 2 \cdot 5^1 + 0 \cdot 5^0$ . Zatem trzydzieści pięć to w systemie piątkowym 120. W systemie czwórkowym to natomiast 203, a w trójkowym 1022.

No i teraz konkluzja tej sporej dygresji. Liczba trzydzieści pięć w systemie dwójkowym! W systemie dwójkowym jest to: 100011. Jak widać, jest to pewien ciąg „zer” i „jedynek”. W systemie dwójkowym używamy bowiem tylko dwóch cyfr, które umownie możemy oznaczyć 0 i 1. To oznacza, że system dwójkowy jest tym, czego szukamy. Pozwala on każdą, nawet najbardziej wymyślną, liczbę zapisać jako pewien, dobrze określony ciąg „zer” i „jedynek”. W ten właśnie sposób każdą informację możemy przystosować do przetworzenia przez komputer. Najpierw zamieniamy ją na liczbę, a później tę liczbę przepisujemy do systemu dwójkowego. Napis „Młody Technik” od teraz to taki ciąg: 01001101 01101100 01101111 01100100 01111001 01010100 01100101 01100011 01101000 01101110 01101001 01101011.

### PRZETWARZANIE INFORMACJI CYFROWEJ

Urządzenie, które powszechnie nazywamy komputerem, to po prostu maszyna, która ma dwa porty: jednym pobiera ciąg „zer” i „jedynek” (informacja wejściowa) i na tej podstawie na drugim generuje inny ciąg „zer” i „jedynek” (informacja wyjściowa). Taki proces nazywamy właśnie przetwarzaniem informacji. To, jaka informacja zostanie wygenerowana przez komputer, zależy od dwóch rzeczy: od informacji wejściowej i oczywiście od budowy samego komputera.

Najprostszym komputerem jest komputer trywialny, który nic nie robi z informacją wejściową, a przesyła ją od razu do wyjścia. Jeśli na wejściu dostanie „jedynekę”, to na wyjściu „jedynekę” wygeneruje. Jeśli dostanie „zero”, to wygeneruje „zero”. Innym, troszkę bardziej zaawansowanym komputerem jest taki, który zamienia sygnał na odwrotny. Jeśli dostanie „zero”, to generuje „jedynekę” i vice versa. Takie najprostsze komputery nazywa się bramkami logicznymi. Ten pierwszy to tzw. bramka identyczności, bo nic nie zmienia, a ta druga to bramka negacji (ang. NOT), bo zamienia sygnał na przeciwny.

WEJŚCIE 1	WYJŚCIE
0	1
1	0



Bramki logiczne możemy ze sobą łączyć i budować coraz bardziej skomplikowane komputery. Niemniej jednak mając komputer złożony tylko z tych dwóch bramek, nie możemy za dużo od niego wymagać. Nie możemy bowiem nim sterować. Dlatego do bramek logicznych zalicza się też takie komputery, które mają dwa wejścia i jedno wyjście. Najprostszą z takich bramek jest bramka iloczynu (ang. AND), która na wyjściu zawsze podaje iloczyn sygnałów wejściowych. Krótko mówiąc, na wyjściu generuje ona „jedynekę” tylko wtedy, gdy na obu wejściach równocześnie otrzymuje „jedynekę”. W przeciwnym razie na wyjściu generuje „zero”. Inną bramką tego typu jest bramka sumy (ang. OR). Ta bramka w przeciwieństwie do poprzedniej generuje na wyjściu „jedynekę”, jeśli na choćby jednym z wejść otrzyma „jedynekę”. Daje w odpowiedzi „zero” tylko wtedy, gdy na obu wejściach jest taka wartość.

WEJŚCIE 1	WEJŚCIE 2	WYJŚCIE
0	0	0
0	1	0
1	0	0
1	1	1



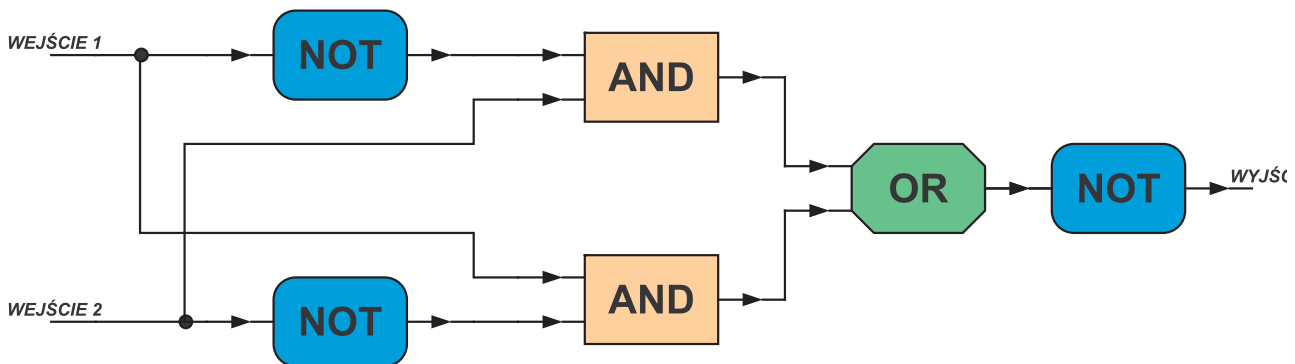
WEJŚCIE 1	WEJŚCIE 2	WYJŚCIE
0	0	0
0	1	1
1	0	1
1	1	1





## PRZYKŁADOWY KOMPUTER

Choć takie bramki mogą wydawać się bardzo trywialne, to praktycznie dają możliwość dowolnego przetworzenia informacji. Dla przykładu zbudujemy teraz komputer, który będzie miał dwa porty wejściowe i jeden wyjściowy. Jego zadaniem będzie porównywanie dwóch zero-jedynkowych informacji. Jedną z nich będziemy kierować na jeden port wejściowy, a drugą na port drugi. Na wyjściu naszego komputera ma się pojawić „jedynek”, jeśli obie informacje się takie same, oraz „zero”, jeśli są różne. Aby tego dokonać, użyjemy sześciu bramek logicznych: trzech bramek NOT, dwóch bramek AND i jednej bramki OR. Schemat znajduje się poniżej. Bardzo łatwo można się przekonać, że niezależnie od tego, jakie informacje prześlemy naszemu komputerowi na wejściach, to zadziała on tak, jak powinien.



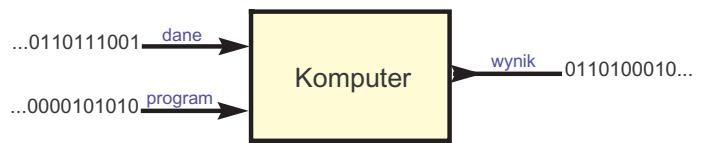
Komputer ten jest bardzo prosty i mogłoby się wydawać, że porównuje on jedynie „zero” z „jedyką”. Jednak jak pamiętamy każdą, nawet najbardziej skomplikowaną informację możemy zapisać jako ciąg „zer” i „jedynek”. To oznacza, że nasz komputer tak naprawdę może porównać ze sobą dowolne dwie informacje. Wystarczy porównywać je bit po bicie. Jeśli choć raz na wyjściu pojawi się „zero”, to znaczy, że informacje nie były równe. Jeśli będą pojawiały się same jedynki, będzie to oznaczało, że obie informacje są tożsame.

## MASZINY PROGRAMOWALNE

Oczywiście każdy zdaje sobie sprawę z tego, że budowanie komputera tylko do jednego konkretnego zadania byłoby bardzo uciążliwe. Jeśli chcielibyśmy, aby komputer robił coś innego, musielibyśmy zbudować maszynę od początku. Dlatego w praktyce lepiej jest skonstruować maszynę uniwersalną, w której będzie można w jakiś sposób powiedzieć, jakie połączenia bramek ma zrealizować. Można to zrobić, dodając do tej maszyny dodatkowy, specjalny port wejściowy, przez który będziemy przesyłać w postaci „zer” i „jedynek” odpowiednią informację o połączeniu bramek dla danych wejściowych przesyłanych na inne porty wejściowe. Tę dodatkową informację zwyczajnie nazywamy PROGRAMEM. Program to nic innego jak informacja przesłana do komputera, którą wykorzystuje on do skonstruowania odpowiednich połączeń pomiędzy portami wejściowymi i wyjściowymi. Oczywiście komputer niczego nie łączy. Jest on po prostu zbudowany z bramek w taki sposób, aby mógł

symulować działanie różnych, dowolnie wymyślonych połączeń bramek. Z tego punktu widzenia program to zatem nic innego jak pewna specyficzna, dodatkowa informacja wejściowa, która steruje pracą bramek, z których składa się sam komputer.

Tak właśnie powstaje maszyna PROGRAMOWALNA, tzn. urządzenie, które w zależności od potrzeb można odpowiednio zaprogramować. Najczęściej, gdy mówimy „komputer”, to mamy właśnie na myśli maszynę programowalną.



Klasyczny komputer to maszyna całkowicie deterministyczna. Wynik jego działania jest jednoznacznie określony przez dane wejściowe oraz program, który mówi, jak te dane przetworzyć. Lekcją pokory dla każdego użytkownika komputera powinna być zatem świadomość, że jeśli komputer robi coś wbrew naszej woli, to jest to jedynie nasza wina. Albo przekazujemy mu błędne dane, albo program, którego użyliśmy do ich przetworzenia jest źle opracowany. Komputer bowiem jedynie przesyła „zera” i „jedynki” i wykonuje na nich działania logiczne tak jak mu każemy. I tak właśnie działa każdy znany ludzkości komputer. Zmieniają się sposoby ich technicznej realizacji, ale koncepcja wciąż taka sama... czy naprawdę nie da się z tym nic zrobić? Cdn. •

- 1) Zero bezwzględne to temperatura, w której zamiera wszelki ruch. Wynosi ona  $-273,16^{\circ}\text{C}$ . Współczesne metody doświadczalne pozwalają w sposób kontrolowany ochładzać niewielkie liczby atomów do temperatur rzędu  $0,01^{\circ}$  powyżej zera bezwzględnego!
- 2) Trurl i Klapucjusz to bohaterowie słynnych opowiadań Stanisława Lema. Byli oni genialnymi inżynierami, którzy potrafili konstruować najbardziej wyrafinowane urządzenia.

