

# Ukryty język komputerów

K

O

D



---

W Y D A N I E I I

CHARLES PETZOLD

Tytuł oryginału: Code: The Hidden Language of Computer Hardware and Software, 2<sup>nd</sup> Edition

Tłumaczenie: Krzysztof Sawka

ISBN: 978-83-8322-168-7

Authorized translation from the English language edition, entitled Code: The Hidden Language of Computer Hardware and Software, 2<sup>nd</sup> Edition by Charles Petzold, published by Pearson Education, Inc, publishing as Addison-Wesley Professional, Copyright © 2023 Charles Petzold.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Polish language edition published by Helion S.A., Copyright © 2023.

Microsoft and the trademarks listed at <http://www.microsoft.com> on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/koduk2>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

	<i>Informacje o autorze</i>	v
	<i>Przedmowa do wydania drugiego</i>	vii
Rozdział 1.	Najlepsi kumple	1
Rozdział 2.	Kody i kombinacje	7
Rozdział 3.	Kod Braille'a a kod binarny	13
Rozdział 4.	Anatomia latarki	21
Rozdział 5.	Komunikacja zza winkła	30
Rozdział 6.	Logika przełączników	38
Rozdział 7.	Telegrafy i przekaźniki	53
Rozdział 8.	Przekaźniki i bramki	59
Rozdział 9.	Naszych dziesięć cyfr	84
Rozdział 10.	Alternatywne dziesiątki	92
Rozdział 11.	Bit bita bitem pogania	110
Rozdział 12.	Bajty i system szesnastkowy	130
Rozdział 13.	Od systemu ASCII do Unicode	140
Rozdział 14.	Dodawanie za pomocą bramek logicznych	159
Rozdział 15.	Czy tak jest naprawdę?	174
Rozdział 16.	A co z odejmowaniem?	187
Rozdział 17.	Sprzężenie zwrotne i przerzutniki	202
Rozdział 18.	Zbudujmy zegar!	229
Rozdział 19.	Czas sklecić pamięć	253
Rozdział 20.	Automatyzacja obliczeń arytmetycznych	275
Rozdział 21.	Jednostka arytmetyczno-logiczna	298
Rozdział 22.	Rejestry i magistrale	315
Rozdział 23.	Sygnały sterujące procesora	332

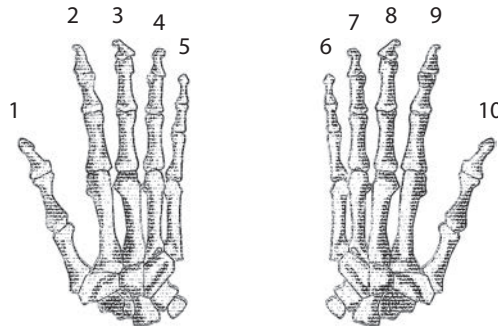
Rozdział 24.	Pętle, skoki i wywołania	353
Rozdział 25.	Urządzenia peryferyjne	375
Rozdział 26.	System operacyjny	384
Rozdział 27.	Pisanie kodu	395
Rozdział 28.	Mózg świata	415
	<i>Kolofon</i>	426



## Rozdział 10.

# Alternatywne dziesiątki

**D**la nas, ludzi, dziesiątka jest niezwykle ważną liczbą. Większość z nas ma dziesięć palców u rąk i u nóg i jest to dla nas całkiem naturalny stan rzeczy. Nasze palce nadają się znakomicie do liczenia, dlatego opracowaliśmy system liczbowy bazujący na dziesięciu cyfrach.



Jak już wspomniałem w poprzednim rozdziale, nasz konwencjonalny system liczbowy jest nazywany *dziesiętnym* lub *decymalnym*. Liczby dziesiętne wydają się nam tak intuicyjne, że początkowo aż trudno sobie wyobrazić inne systemy liczbowe. Rzeczywiście gdy widzimy liczbę 10, z miejsca przychodzi do głowy następująca liczba kaczek:



Jedynym powodem tego, że liczba 10 oznacza dla nas tyle kaczek, jest tylko i wyłącznie taka sama liczba palców u naszych rąk lub nóg. Gdyby w wyniku ewolucji człowiek miał inną liczbę palców, nasz sposób liczenia mógłby być inny, a liczba 10 oznaczałaby coś innego. Ta sama liczba mogłaby oznaczać tyle kaczek:

$$10 = \text{[8 ducks]}$$

Albo tyle:

$$10 = \text{[4 ducks]}$$

Albo nawet tyle kaczuch:

$$10 = \text{[2 ducks]}$$

Gdy dotrzemy do punktu, w którym liczba 10 oznacza dwie kaczki, będziemy mogli przyjąć się sposobowi reprezentowania liczb za pomocą przełączników, przewodów i żarówek, a także temu, jak przekaźniki oraz bramki logiczne (a zatem i komputery) wykorzystują te liczby.

Jak potoczyłaby się historia systemów liczbowych, gdyby człowiek dysponował jedynie czterema palcami u każdej ręki, na podobieństwo bohaterów kreskówek? Prawdopodobnie nie zaistniałaby potrzeba, aby kiedykolwiek wymyślić system dziesiętny. Natomiast byłoby dla nas całkowicie naturalne, normalne, rozsądne, nieuniknione, niepodważalne i niewątpliwie właściwe opracować system liczbowy bazujący na ósemce. Jest to tak zwany system *ósemkowy* (lub *oktalny*).

Gdyby nasz system liczbowy bazował nie na dziesiątce, lecz na ósemce, nie potrzebowalibyśmy już tego symbolu:

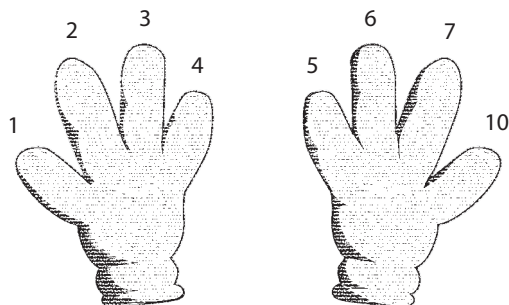
9

Pokaż go dowolnej postaci kreskówek, to zapyta Cię: „A co to? Czym to się je?”. Jeśli zastanowić się nieco dłużej, doszlibyśmy do wniosku, że nie potrzebujemy również następującego znaku:

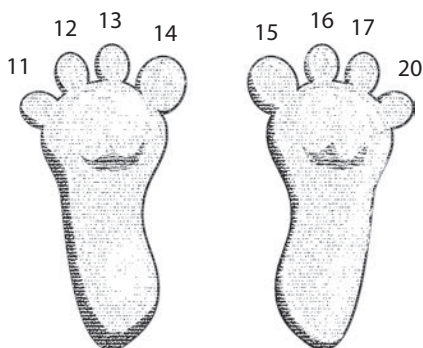
8

W systemie dziesiętnym nie używamy osobnego symbolu reprezentującego liczbę 10, zatem w systemie ósemkowym nie potrzebujemy cyfry 8.

W systemie dziesiętnym występuje następująca numeracja: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Z kolei w systemie ósemkowym występują liczby 0, 1, 2, 3, 4, 5, 6, 7 i co dalej? Skończyły nam się symbole. Jedyną logiczną możliwością jest 10 i tak też jest w istocie. W systemie ósemkowym liczbą występującą po siódemce jest 10. Liczba ta nie oznacza jednak liczby palców zwykłego człowieka. Określa ona w tym przypadku liczbę palców u bohaterów kreskówek.



Doliczmy do tego jeszcze palce u kreskówkowych stóp:



Podczas korzystania z systemów innych od dziesiętkowego możesz (w celu unikania pomyłek) wymawiać liczbę 10 jako *jeden zero*. Na drodze analogii 13 to *jeden trzy*, a 20 to *dwa zero*. Żeby być bardziej precyzyjnym i *naprawdę* uniknąć niejednoznaczności, możesz mówić *dwa zero w systemie ósemkowym* albo *dwa zero oktalne*.

Mamy tylko po osiem palców rąk i nóg, mimo to możemy dalej liczyć w systemie ósemkowym. Nie różni się to tak bardzo od liczenia w systemie dziesiętnym, musimy jedynie pomijać każdą liczbę zawierającą cyfrę 8 lub 9:

0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22,  
 23, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 36, 37, 40, 41, 42, 43,  
 44, 45, 46, 47, 50, 51, 52, 53, 54, 55, 56, 57, 60, 61, 62, 63, 64,  
 65, 66, 67, 70, 71, 72, 73, 74, 75, 76, 77, 100...

Ostatnią liczbę wymawiamy jako *jeden zero zero*. Jest ona równa kwadratowi liczby palców obu dłoni u postaci kreskówkowych.

Przez całe życie korzystamy z systemu dziesiętnego, dlatego oczekujemy, że pewne sekwencje liczb będą odpowiadać określonym liczebnościom w prawdziwym świecie. Liczenie za pomocą innego systemu liczbowego przypomina wkroczenie do zupełnie odmiennego świata. Poniżej prezentuję kilka przykładów liczb ósemkowych:



W bajce o Królowie Śnieżce występuje 7 krasnoludków, zupełnie jak w systemie dziesiętnym.

Bohaterowie kreskówki mają 10 palców.

Beethoven skomponował 11 symfonii.

Ludzie mają 12 palców.

Rok składa się z 14 miesięcy.

Jeżeli przeliczanie w pamięci liczb ósemkowych na dziesiętne nie sprawia Ci problemu, to świetnie. Jest to bardzo dobre ćwiczenie. W przypadku dwucyfrowej liczby ósemkowej, której pierwsza cyfra zaczyna się od 1, jej równoważnik w systemie dziesiętnym wynosi 8 plus cyfra na drugiej pozycji. Liczba miesięcy w systemie oktalnym wynosi 14, a zatem w systemie decymalnym jest to  $8 + 4 = 12$ . Idźmy dalej:

Tuzin „piekarski” wynosi 15.

Dwa tygodnie to inaczej 16 dni.

Człowiek osiąga dorosłość w wieku 22 lat.

W jednej dobie mieści się 30 godzin.

Alfabet łaciński zawiera 32 litery.

Gdy dwucyfrowa liczba ósemkowa zaczyna się od cyfry innej niż 1, konwersja do liczby dziesiętnej wygląda nieco inaczej: musisz pomnożyć pierwszą cyfrę przez 8 i dodać do wyniku drugą cyfrę. W systemie oktalnym liczba liter alfabetu łacińskiego wynosi 32, zatem w systemie dziesiętnym otrzymujemy  $3 \cdot 8 + 2 = 26$ .

Jedna kwarta amerykańska płynu składa się z 40 uncji.

W jednej talii znajdziemy 64 karty.

Szachownica składa się z  $10 \cdot 10$ , czyli 100 pól.

W systemie dziesiętnym liczba pól szachowych to  $8 \cdot 8$ , czyli 64 pola.

Długość pola gry na boisku futbolu amerykańskiego wynosi 144 jardy.

Liczba zawodniczek przystępujących do zawodów singlowych w Wimbledonie jest równa 200.

Liczba znaków w ósmiopunktowym kodzie Braille’a wynosi 400.

Zwróć uwagę, że na powyższej liście znajdziemy kilka ładnych okrągłych liczb ósemkowych, takich jak 100, 200 czy 400. Pod pojęciem *ładna okrągła liczba* mamy najczęściej na myśli wartość zawierającą na końcu co najmniej jedno 0. Dwa zera na końcu liczby decymalnej oznaczają, że jest ona wielokrotnością liczby 100, czyli iloczynu  $10 \cdot 10$ . W systemie ósemkowym dwa zera na końcu liczby oznaczają, że stanowi ona wielokrotność liczby 100, ale w systemie dziesiętnym jest odpowiednikiem liczby 64. Liczba zawodniczek przystępujących do zawodów w Wimbledonie wynosi 128 w systemie dziesiętnym, a liczba znaków w ósmiopunktowym kodzie Braille’a jest równa 256.

W pierwszych trzech rozdziałach analizowaliśmy zależność kodów dwójkowych od potęg cyfry 2. Liczba kombinacji kodu Morse’a przy użyciu czterech kropek oraz kresek wynosi 2 do potęgi czwartej, czyli 16. Liczba kombinacji w sześciopunktowym kodzie Braille’a jest równa 2 do potęgi szóstej, czyli 64. W przypadku ósmiopunktowego kodu Braille’a liczba ta

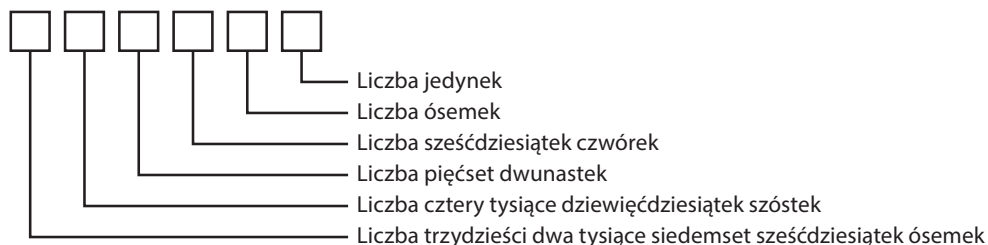
wzrasta do 2 do potęgi ósmej, czyli 256. Za każdym razem, gdy mnożymy potęgę cyfry 2 przez inną potęgę cyfry 2, otrzymujemy również liczbę stanowiącą potęgę cyfry 2.

Poniższa tabela przedstawia pierwszych 12 potęg liczby 2, a także ich reprezentacje w systemach dziesiętnym i ósemkowym:

Potęga liczby 2	Wartość w systemie dziesiętnym	Wartość w systemie ósemkowym
$2^0$	1	1
$2^1$	2	2
$2^2$	4	4
$2^3$	8	10
$2^4$	16	20
$2^5$	32	40
$2^6$	64	100
$2^7$	128	200
$2^8$	256	400
$2^9$	512	1000
$2^{10}$	1024	2000
$2^{11}$	2048	4000
$2^{12}$	4096	10 000

Ósemka jest potęgą dwójki, dlatego w kolumnie z wartościami ósemkowymi widzimy wiele ładnie zaokrąglonych liczb, co sugeruje bliższe powiązanie z kodami binarnymi niż z dziesiętnymi.

Struktura systemu ósemkowego nie różni się zbytnio od systemu decymalnego. Diabeł tkwi jedynie w szczegółach. Na przykład każda pozycja w liczbie zapisanej systemem ósemkowym stanowi kolejną potęgę liczby 8:



Zatem przykładową liczbę  $3\,725_{(8)}$  możemy rozbić na następujące czynniki:

$$3725 = 3000 + 700 + 20 + 5$$

Liczbę tę można również wyrazić jako poszczególne cyfry pomnożone przez potęgę liczby 8 w zapisie ósemkowym:

$$\begin{aligned} 3725 &= 3 \cdot 1000 + \\ &+ 7 \cdot 100 + \\ &+ 2 \cdot 10 + \\ &+ 5 \cdot 1 \end{aligned}$$

Albo jeszcze inaczej:

$$\begin{aligned} 3725 &= 3 \cdot 8^3 + \\ &+ 7 \cdot 8^2 + \\ &+ 2 \cdot 8^1 + \\ &+ 5 \cdot 8^0 \end{aligned}$$

Po obliczeniu powyższej operacji w systemie dziesiętnym otrzymasz wartość 2005. W taki właśnie sposób można przekształcać liczby systemu ósemkowego w wartości systemu decymalnego.

Możesz dodawać i mnożyć liczby w systemie ósemkowym tak samo, jak robione jest to w systemie dziesiętnym. Jedyna różnica polega na zawartości tabliczek dodawania i mnożenia poszczególnych cyfr. Oto tabliczka dodawania liczb systemu oktalnego:

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Na przykład  $5 + 7 = 14$ . Można dodawać dłuższe liczby tak samo jak w systemie dziesiętnym. Poniżej prezentuję proste ćwiczenie, w którym operacja przypomina dodawanie liczb dziesiętnych, ale zostały w rzeczywistości użyte liczby ósemkowe. Skorzystaj z powyższej tabeli, aby obliczyć wynik:

$$\begin{array}{r} 135 \\ + 643 \\ \hline \end{array}$$

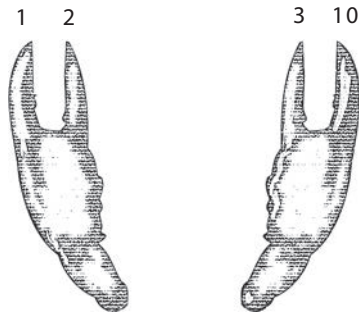
W każdej kolumnie poszczególne cyfry są sumowane do wartości przekraczającej siódmkę w systemie ósemkowym, dlatego musimy przenosić wartość z każdej jednej kolumny do następnej. Otrzymujemy w konsekwencji liczbę 1000.

Mnożenie także wygląda podobnie. Dwa razy dwa nadal daje cztery w systemie ósemkowym, ale trzy razy trzy to już nie jest dziewięć. Jak to możliwe? Zamiast tego otrzymujemy 11, co stanowi odpowiednik wartości 9. Poniżej widzimy pełną tabliczkę mnożenia w systemie ósemkowym:

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

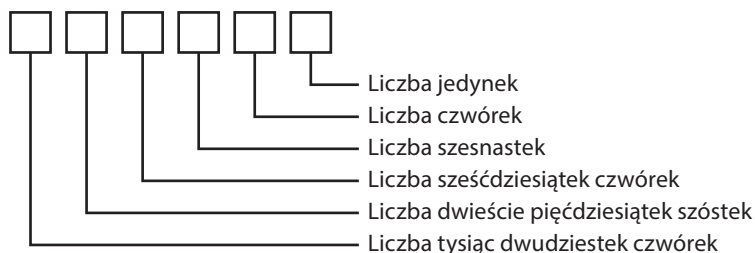
Zgodnie z tą tabelą iloczyn  $4 \cdot 6 = 30$ , czyli 24 w systemie dziesiętnym.

System ósemkowy jest równie prawidłowy jak system dziesiętny, pójdźmy jednak dalej. Skoro udało nam się stworzyć system liczbowy odpowiedni dla postaci z kreskówek, przygotujmy teraz coś bardziej pasującego do homara. Homary w zasadzie nie mają palców, ale przedstawiciele gatunku *Homarus americanus* mają szczypce na długich przednich odnóżach. W takim przypadku najodpowiedniejszy będzie system *czwórkowy*:



Kolejne liczby w systemie czwórkowym wyglądają tak: 0, 1, 2, 3, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33, 100, 101, 102, 103, 110, 111, 112, 113, 120 i tak dalej.

Nie zamierzam poświęcać wiele miejsca systemowi czwórkowemu, ponieważ niebawem przejdziemy do znacznie ważniejszego zagadnienia. Możesz jednak przekonać się tutaj, że każda pozycja w systemie czwórkowym odpowiada tym razem potęgom liczby *cztery*:



Liczbę 31 232 w systemie czwórkowym możemy rozisać następująco:

$$\begin{aligned}
 31\ 232 &= 3 \cdot 10\ 000 + \\
 &+ 1 \cdot 1000 + \\
 &+ 2 \cdot 100 + \\
 &+ 3 \cdot 10 + \\
 &+ 2 \cdot 1
 \end{aligned}$$

Każda cyfra jest mnożona przez potęgę liczby 4 w systemie czwórkowym:

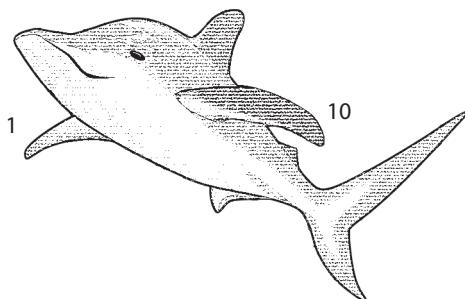
$$\begin{aligned}
 31\ 232 &= 3 \cdot 4^4 + \\
 &+ 1 \cdot 4^3 + \\
 &+ 2 \cdot 4^2 + \\
 &+ 3 \cdot 4^1 + \\
 &+ 2 \cdot 4^0
 \end{aligned}$$

Po wykonaniu obliczeń w systemie dziesiętnym okazuje się, że odpowiednikiem liczby 31 232 w systemie czwórkowym jest 878 w systemie dziesiętnym.

Pozostał nam jeszcze jeden przeskok i tym razem czeka nas jazda bez trzymanki. Przyjmijmy, że jesteśmy delfinami i możemy liczyć jedynie za pomocą dwóch płetw. Taki system liczbowy jest znany jako *dwójkowy* albo *binarny* (od łacińskiego słowa *binarius* oznaczającego *podwójny*). Wydaje się, że jako delfiny znalibyśmy tylko dwie cyfry: 0 i 1.

Przekonałeś się już wcześniej, że wartości 1 i 0 można użyć w algebrze boolowskiej do określania prawdy i fałszu, potwierdzenia lub zaprzeczenia, kotka idealnego lub niezbyt nam odpowiadającego. Tych samych dwóch cyfr możesz używać także do liczenia.

Dwie cyfry (0 i 1) to niewiele, a przyzwyczajenie się do systemu binarnego wymaga wprawy i czasu. Problem polega na tym, że bardzo szybko kończy nam się pula dostępnych cyfr. Na przykład delfin może liczyć za pomocą płetw w następujący sposób:



Tak, w systemie dwójkowym następną liczbą po 1 jest 10. Trochę to przerażające, ale nie powinno być zaskoczeniem. Bez względu na stosowany system liczbowy zawsze po wyczerpaniu liczb jednocyfrowych pierwszą liczbą dwucyfrową jest 10. W systemie binarnym kolejne liczby są takie:

0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100,  
1101, 1110, 1111, 10 000, 10 001...

Liczby te wydają się duże, ale tak nie jest. Precyzyjniejsze byłoby stwierdzenie, że liczby binarne bardzo szybko stają się *długie*, a nie duże:

Istota ludzka posiada 1 głowę.

Delfiny mają 10 płetw.

Objętość jednej łyżki stołowej jest równa objętości 11 łyżeczek.

Kwadrat ma 100 boków.

Człowiek ma 101 palców u każdej ręki.

Owady mają 110 odnóży.

Tydzień liczy 111 dni.

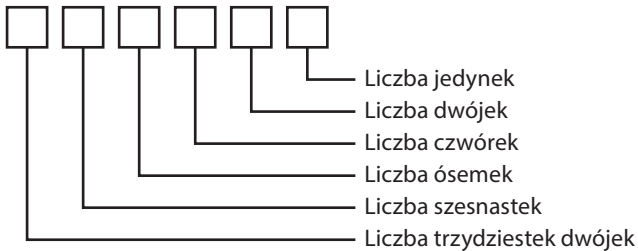
Na oktet składa się 1000 muzyków.

Na jeden mecz bejsbolowy składa się 1001 rund.

Kapelusz kowbojski mieści w sobie 1010 galonów.

itd.

W wielocyfrowej liczbie dwójkowej poszczególne pozycje symbolizują kolejne potęgi liczby 2:



Zatem gdy mamy jakąś liczbę binarną, której pierwsza cyfra ma wartość 1, a po niej następują zera, to ta cyfra jest potęgą liczby 2, a wykładnik tej potęgi jest równy liczbie pozycji występujących po danej jedynce. Reguła ta została zaprezentowana w znanej nam, ale rozszerzonej tabeli:

Potęga liczby 2	Wartość w systemie dziesiętnym	Wartość w systemie ósemkowym	Wartość w systemie czwórkowym	Wartość w systemie dwójkowym
$2^0$	1	1	1	1
$2^1$	2	2	2	10
$2^2$	4	4	10	100
$2^3$	8	10	20	1000
$2^4$	16	20	100	10 000
$2^5$	32	40	200	100 000
$2^6$	64	100	1000	1 000 000
$2^7$	128	200	2000	10 000 000
$2^8$	256	400	10 000	100 000 000
$2^9$	512	1000	20 000	1 000 000 000
$2^{10}$	1024	2000	100 000	10 000 000 000
$2^{11}$	2048	4000	200 000	100 000 000 000
$2^{12}$	4096	10 000	1 000 000	1 000 000 000 000

Załóżmy, że natrafiasz na liczbę binarną 101 101 011 010. Możesz ją rozbić na czynniki pierwsze:

$$\begin{aligned}
 101\ 101\ 011\ 010 &= 1 \cdot 100\ 000\ 000\ 000 + \\
 &+ 0 \cdot 10\ 000\ 000\ 000 + \\
 &+ 1 \cdot 1\ 000\ 000\ 000 + \\
 &+ 1 \cdot 100\ 000\ 000 + \\
 &+ 0 \cdot 10\ 000\ 000 + \\
 &+ 1 \cdot 1\ 000\ 000 + \\
 &+ 0 \cdot 100\ 000 + \\
 &+ 1 \cdot 10\ 000 + \\
 &+ 1 \cdot 1000 + \\
 &+ 0 \cdot 100 + \\
 &+ 1 \cdot 10 + \\
 &+ 0 \cdot 1
 \end{aligned}$$

Tę samą liczbę możemy też zapisać prościej za pomocą potęg liczby 2:

$$\begin{aligned}
 101\ 101\ 011\ 010 &= 1 \cdot 2^{11} + \\
 &+ 0 \cdot 2^{10} + \\
 &+ 1 \cdot 2^9 + \\
 &+ 1 \cdot 2^8 + \\
 &+ 0 \cdot 2^7 + \\
 &+ 1 \cdot 2^6 + \\
 &+ 0 \cdot 2^5 + \\
 &+ 1 \cdot 2^4 + \\
 &+ 1 \cdot 2^3 + \\
 &+ 0 \cdot 2^2 + \\
 &+ 1 \cdot 2^1 + \\
 &+ 0 \cdot 2^0
 \end{aligned}$$

Po dodaniu składowych w systemie dziesiętnym otrzymujesz  $2048 + 512 + 256 + 64 + 16 + 8 + 2$ , co daje w rezultacie 2906, czyli wartość liczby binarnej równoważną w zapisie dziesiętnym.

Aby móc przekształcać liczby binarne w decymalne w bardziej zwięzły sposób, być może spodoba Ci się przygotowana przeze mnie metoda:

$$\begin{array}{cccccccc}
 \square & \square & \square & \square & \square & \square & \square & \square \\
 \cdot 128 & \cdot 64 & \cdot 32 & \cdot 16 & \cdot 8 & \cdot 4 & \cdot 2 & \cdot 1 \\
 \square + \square + \square + \square + \square + \square + \square + \square = \square
 \end{array}$$

Powyższy szablon umożliwia przekształcanie ośmiocyfrowych liczb dwójkowych, ale można go z łatwością rozwinąć. Żeby z niego skorzystać, wstaw na górze osiem cyfr binarnych, po jednej na każde pole. Wykonaj osiem operacji mnożenia i wyniki wstaw w osiem dolnych polach. Zsumuj otrzymane rezultaty, aby otrzymać wynik końcowy. Poniżej pokazuję przykład przekształcenia liczby  $10\ 010\ 110_{(2)}$ :

$$\begin{array}{cccccccc}
 \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{1} & \boxed{0} \\
 \cdot 128 & \cdot 64 & \cdot 32 & \cdot 16 & \cdot 8 & \cdot 4 & \cdot 2 & \cdot 1 \\
 \boxed{128} + \boxed{0} + \boxed{0} + \boxed{16} + \boxed{0} + \boxed{4} + \boxed{2} + \boxed{0} = \boxed{150}
 \end{array}$$

Operacja przekształcania z liczb dziesiętnych na dwójkowe jest nieco bardziej skomplikowana, ale za pomocą poniższego szablonu możesz dokonywać tego na liczbach dziesiętnych w zakresie od 0 do 255:

$$\begin{array}{cccccccc}
 \square & \square & \square & \square & \square & \square & \square & \square \\
 /128 & /64 & /32 & /16 & /8 & /4 & /2 & /1 \\
 \square & \square & \square & \square & \square & \square & \square & \square
 \end{array}$$

Proces konwersji jest trudniejszy, niż się na pozór wydaje. Najpierw wstaw całą liczbę (mniejszą lub równą 255) w pierwszym polu na górze:

$$\begin{array}{cccccccc}
 \boxed{150} & \square & \square & \square & \square & \square & \square & \square \\
 /128 & /64 & /32 & /16 & /8 & /4 & /2 & /1 \\
 \square & \square & \square & \square & \square & \square & \square & \square
 \end{array}$$



Podziel tę liczbę (dzielną) przez znajdujący się bezpośrednio pod nią dzielnik (128) aż do momentu uzyskania ilorazu i reszty: 150 dzielone przez 128 daje 1 i resztę równą 22. Umieść iloraz w pierwszym polu na dole, a resztę z dzielenia w sąsiednim polu na górze.

150	22						
/128	/64	/32	/16	/8	/4	/2	/1
1							

Teraz podziel 22 przez 64, ale tym razem pozostań przy pierwszym etapie: 22 jest mniejsze od 64, iloraz wynosi 0, a reszta 22. Umieść wartość 0 w drugim polu na dole, a resztę wpisz w kolejnym wolnym polu na górze:

150	22	22					
/128	/64	/32	/16	/8	/4	/2	/1
1	0						

W analogiczny sposób wypełnij resztę szablonu. Każdy iloraz wynosi 0 lub 1, dlatego finalnie pola w dolnym rzędzie utworzą sekwencję cyfr binarnych:

150	22	22	22	6	6	2	0
/128	/64	/32	/16	/8	/4	/2	/1
1	0	0	1	0	1	1	0

Binarnym odpowiednikiem liczby 150 jest 10 010 110.

Przekształcenia między liczbami dziesiętnymi a binarnymi wyglądają zdecydowanie dziwnie, dlatego jeżeli będziesz chciał kiedykolwiek przeprowadzać je w jakichś rzeczywistych zastosowaniach, to ucieszy Cię wiadomość, że kalkulatory zarówno w systemie Windows, jak i macOS zawierają tryb programisty ułatwiający Ci to zadanie.

Na początku ery komputerów cyfrowych korzystanie z liczb binarnych wcale nie było takie powszechne. Niektóre z pierwszych komputerów zaprojektowano i skonstruowano z myślą o znanym nam doskonale systemie dziesiętnym. Zaprojektowana w latach 30. XIX wieku przez angielskiego matematyka Charlesa Babbage'a ([https://en.wikipedia.org/wiki/Charles\\_Babbage](https://en.wikipedia.org/wiki/Charles_Babbage), 1791 – 1871) maszyna analityczna przechowywała liczby dziesiętne za pomocą odpowiedniego położenia kół zębatych (niestety nie był w stanie dokończyć konstruowania tej maszyny). Niektóre z pierwszych komputerów cyfrowych, takie jak Harvard Mark I (uruchomiony w 1944 r.) czy ENIAC (1946 r.) również wykorzystywały do działania liczby dziesiętne. Poniemkóre komputery firmy IBM tworzone w latach 60. XX wieku również miały architekturę bazującą na systemie dziesiętnym.

To kodowanie binarne stanowi jednak cechę wyróżniającą rewolucję cyfrową. Prostota liczb dwójkowych jest prawdopodobnie najwyraźniej widoczna w podstawowych operacjach dodawania i mnożenia. Ta część *naprawdę* Ci się spodoba. Wyobraź sobie, jak szybko opanowałbyś dodawanie, gdybyś musiał zapamiętać tylko taką tabliczkę:

+	0	1
0	0	1
1	1	10

Skorzystajmy z tej tabeli i dodajmy dwie liczby binarne:

$$\begin{array}{r} 1100101 \\ + 0110110 \\ \hline 10011011 \end{array}$$

Począwszy od pierwszej kolumny z prawej: 1 dodać 0 daje 1. W kolejnej kolumnie: 0 dodać 1 daje 1. Trzecia kolumna: 1 dodać 1 daje 0 i przenosimy 1. Czwarta kolumna: przeniesiona jedynka dodać 0 i dodać 0 daje 1. Piąta kolumna: 0 dodać 1 daje 1. Szósta kolumna: 1 dodać 1 daje 0 i przenosimy 1. Siódma kolumna: przeniesiona jedynka dodać 1 i dodać 0 daje 10.

Tabliczka mnożenia okazuje się jeszcze łatwiejsza od tabliczki dodawania, ponieważ bazuje na dwóch bardzo prostych regułach mnożenia: iloczyn dowolnej liczby i zera daje zawsze 0, natomiast iloczyn dowolnej liczby i jedynki nie wpływa w żaden sposób na tę liczbę.

·	0	1
0	0	0
1	0	1

Pomnóżmy liczbę dziesiętną 13 (binarnie 1101) przez 11 (1011 w zapisie binarnym). Nie pokażę wszystkich etapów, ale operacja ta jest realizowana tak samo jak przy mnożeniu liczb dziesiętnych:

$$\begin{array}{r} 1101 \\ \cdot 1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1010 \\ \hline 10001111 \end{array}$$

W rezultacie otrzymujemy liczbę 143.

Osoby pracujące na liczbach binarnych często zapisują je z zerami wiodącymi (to znaczy z zerami występującymi na lewo od pierwszej jedynki), na przykład 0011 zamiast 11. W żadnym wypadku nie zmienia to wartości liczby; służy to wyłącznie celom kosmetycznym. W poniższej tabeli widzimy pierwszych szesnaście liczb binarnych wraz z ich decymalnymi odpowiednikami:

Zapis w systemie dwójkowym	Zapis w systemie dziesiętnym
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Zatrzymajmy się na chwilę przy tej liście liczb binarnych. Przyjrzyj się każdej z czterech kolumn zer i jedynek i zwróć uwagę, jak zmieniają się w nich cyfry:

- Cyfra w pierwszej kolumnie od prawej naprzemiennie przyjmuje wartość 0 i 1.
- W drugiej kolumnie od prawej wartości zmieniają się co dwa zera/dwie jedynki.
- W trzeciej kolumnie od prawej następuje zmiana co cztery zera/jedynki.
- W ostatniej kolumnie od prawej zmiana pojawia się co osiem zer/jedynek.

Jest to *bardzo* powtarzalne, prawda? Jest to w rzeczywistości tak powtarzalne, że istnieje możliwość stworzenia układu generującego automatycznie sekwencje liczb binarnych. Wróćmy do tego zagadnienia w rozdziale 17.

Co więcej, możemy z łatwością zapisać kolejnych 16 liczb binarnych, powtarzając sekwencję pierwszych szesnastu cyfr i wstawiając jedynkę na pierwszej pozycji:

Zapis w systemie dwójkowym	Zapis w systemie dziesiętnym
10 000	16
10 001	17
10 010	18
10 011	19
10 100	20
10 101	21
10 110	22
10 111	23
11 000	24
11 001	25
11 010	26
11 011	27
11 100	28
11 101	29
11 110	30
11 111	31

Można spojrzeć na to w inny sposób: podczas liczenia w systemie dwójkowym cyfra znajdująca się skrajnie po prawej stronie (jest to tak zwana najmniej znacząca cyfra) naprzemiennie przyjmuje wartość 0 i 1. Przy każdej zmianie z jedynki na zero cyfra na lewej pozycji (bardziej znacząca cyfra) również się zmienia, albo z 0 na 1, albo z 1 na 0. Mówiąc ogólniej, gdy cyfra binarna przechodzi z wartości 1 do 0, zmienia się również cyfra znajdująca się na pozycji po lewej, albo z 0 na 1, albo z 1 na 0.

Liczby binarne bardzo szybko stają się długie. Na przykład w zapisie binarnym dwanaście milionów przyjmuje postać 101 101 110 001 101 100 000 000. Jedną z metod bardziej zwięzłego przedstawiania liczb binarnych jest przedstawianie ich w formie ósemkowej. Rozwiązanie to sprawdza się dobrze, ponieważ trzy cyfry binarne odpowiadają jednej cyfrze ósemkowej:

Zapis binarny	Zapis ósemkowy
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Wróćmy do tej długiastej liczby binarnej odwzorowującej liczbę 12 milionów i podzielmy ją na trzejelementowe grupy, počząwszy od prawej strony:

101 101 110 001 101 100 000 000

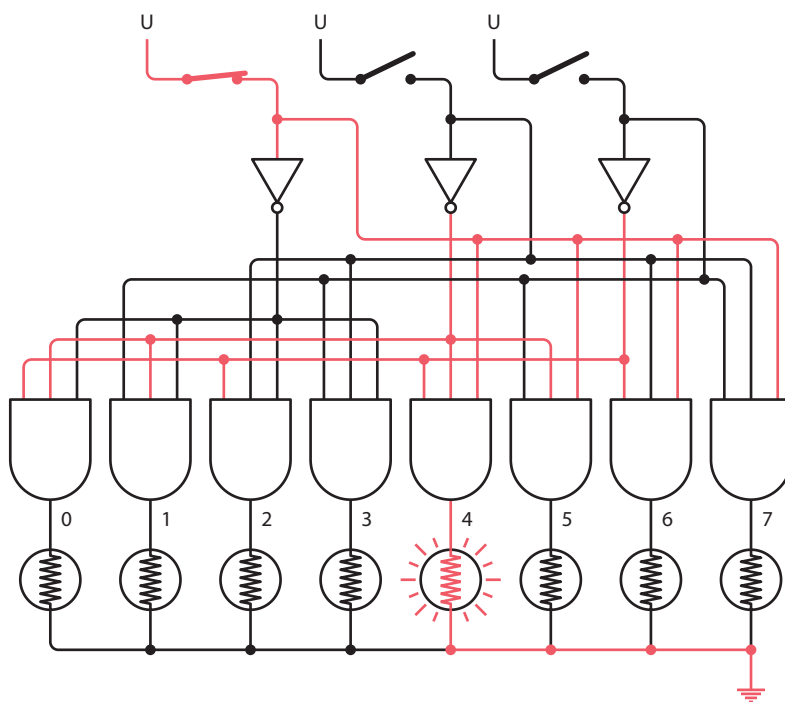
Każda grupa trzech cyfr binarnych odpowiada jednej cyfrze ósemkowej:

$\underbrace{101}_5 \underbrace{101}_5 \underbrace{110}_6 \underbrace{001}_1 \underbrace{101}_5 \underbrace{100}_4 \underbrace{000}_0 \underbrace{000}_0$

Liczba dziesiętna 12 milionów przyjmuje wartość 55615400 w systemie ósemkowym. W rozdziale 12. poznasz jeszcze zwięźlejszy sposób wyrażania liczb binarnych.

Poprzez zredukowanie systemu liczbowego do zaledwie dwóch cyfr, 0 i 1, osiągnęliśmy szczyt naszych możliwości. Prościej już się nie da, chyba że powrócilibyśmy do prymitywnych kresek naskalnych. Najważniejsze jednak, że dwójkowy system liczbowy stanowi pomost pomiędzy arytmetyką a elektrycznością. Przełączniki, przewody i żarówki mogą reprezentować binarne cyfry 0 i 1, a po dołączeniu bramek logicznych można tymi cyframi manipulować. Dlatego właśnie liczby binarne mają tak *wiele* wspólnego z komputerami.

Powyżej widziałeś niewielką tabelę ukazującą relację między trzycyfrowymi liczbami binarnymi a ich ósemkowymi odpowiednikami. Za pomocą przełączników, żarówek i bramek logicznych możesz zbudować obwód realizujący tę konwersję za nas:



Obwód ten niewątpliwie na pierwszy rzut oka wygląda niezwykle złowrogo i bardziej przypomina koszmarny zlepek krzyżujących się autostrad w obcym mieście, gdzie wszystkie znaki drogowe są nieczytelne. W rzeczywistości jednak schemat ten jest całkiem logiczny. Małe kropki wyznaczają miejsca połączenia przewodów. W pozostałych przypadkach nie są one połączone i jedynie się na siebie nakładają.

Obwód rozpoczyna się na górze trzema przełącznikami symbolizującymi trzycyfrowe liczby binarne. Zamknięte przełączniki mają wartość 1, a otwarte — 0. W tym przykładzie widoczna jest liczba binarna 100. Na dole widzimy osiem żarówek oznaczonych wartościami od 0 do 7. Tylko jedna z nich świeci się w zależności od konfiguracji przełączników.

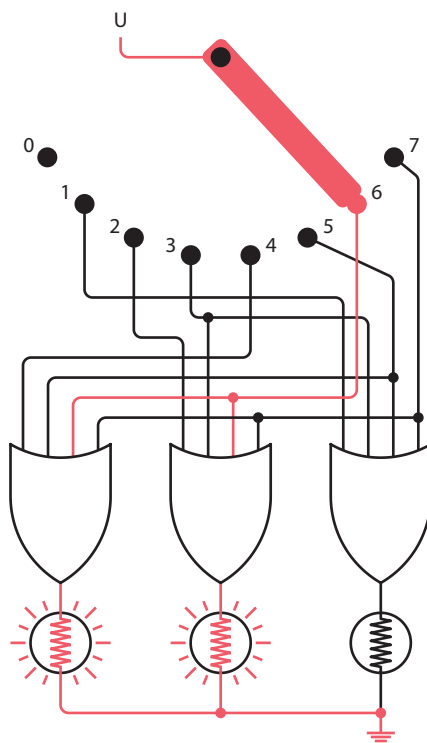
Prawdopodobnie łatwiej zrozumieć ten obwód „od dołu”: każda z żarówek jest zasilana przez trójwejściową bramkę AND. Na wyjściu tej bramki występuje sygnał 1 tylko wtedy, gdy do wszystkich jej wejść dociera sygnał 1. Każde z tych wejść odpowiada poszczególnym przełącznikom, czasami bezpośrednio, a czasami za pomocą sygnału odwróconego przez trzy inwertery umieszczone tuż pod przełącznikami. Jak wiemy, jeżeli na wejście inwertera dociera sygnał 1, to będzie z niego wychodził sygnał 0 i odwrotnie.

Widoczne na schemacie przełączniki przedstawiają konfigurację: zamknięty, otwarty, otwarty, co odpowiada liczbie binarnej 100. Jeżeli prześledzisz czerwone linie, to najbardziej znacząca cyfra (1) stanowi jedno z wejść do bramki AND symbolizującej liczbę ósemkową 4. Sygnał następnej cyfry (środkowy przełącznik) zostaje odwrócony, a następnie przekazany na wejście tej samej bramki AND. Najmniej znacząca cyfra (prawy przełącznik) również zostaje potraktowana w podobny sposób, zanim stanie się trzecim wejściem do interesującej nas bramki AND. Zatem w bramce AND odpowiadającej cyfrze ósemkowej 4 wszystkie sygnały wejściowe wynoszą 1, dlatego na jej wyjściu również występuje sygnał 1.

Na podobnej zasadzie każda z pozostałych siedmiu bramek AND ma na wejściu odmienną kombinację sygnałów pochodzących z przełączników lub inwerterów.

Takie małe urządzenie nazywane jest *dekoderem 3 na 8 linii*. Nazwa sugeruje, że trzycyfrowa liczba binarna jest *kodelem* określającym jedną z ośmiu kombinacji.

Inny obwód, zwany *dekoderem 8 na 3 linie*, realizuje odwrotne zadanie. Teraz będziemy potrzebować nieco innego typu przełącznika, umożliwiającego wybór jednej z ośmiu pozycji. W rzeczywistości możemy użyć w tym celu pinezek bądź gwoździ i fragmentu metalowej puszki:



Każda z cyfr binarnych u dołu jest wyświetlana za pomocą żarówki zasilanej za pomocą czterowejściowej bramki OR. Na jej wyjściu występuje wartość 1 wtedy, gdy na *którymkolwiek* z wejść występuje sygnał 1. Gdy przełącznik na górze zostaje umieszczony na pozycji symbolizującej cyfrę ósemkową 6, pierwsza i druga bramka OR mają na wejściu sygnał 1, co sprawia, że również na ich wyjściu pojawia się sygnał 1, dzięki czemu zostaje odkodowana liczba binarna 110. Zwróć uwagę, że pozycja 0. w górnym lewym rogu przełącznika nie jest do niczego podłączona. Wynika to z faktu, że odpowiednikiem liczby ósemkowej 0 w zapisie binarnym jest 000, więc nie trzeba zapalać żadnej żarówki.

Mniej więcej w 1948 r. amerykański matematyk John Wilder Tukey (<http://www.ams.org/notices/200202/fea-tukey.pdf>; 1915 – 2000) zrozumiał, że pojęcie *cyfra binarna* (ang. *binary digit*) nabierze o wiele większego znaczenia w nadchodzących latach popularyzacji komputerów. Postanowił ukuć nowe słowo w zastępstwie niewygodnych pięciu sylab *binary digit*. Rozważał słowa *bigit* i *binit*, jednak zwyciężył krótki, prosty, elegancki i przeuroczy wyraz *bit*.





# PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA  
**Helion** 

# Chcesz zrozumieć, jak działa komputer? Kod jest źródłem!

Mało kto się zastanawia, jak działa komputer. Jeśli ktoś chce pisać aplikacje, najczęściej wybiera sobie język programowania i koduje, jednak bez zagłębiania się w niuanse pojedynczych bitów pamięci. Tymczasem tajemniczy świat komputerów warto poznać nieco lepiej. Zrozumienie zasad działania procesora ułatwia pisanie wysoce efektywnego kodu. A nawet zwykły użytkownik powinien rozumieć różnicę między pamięcią operacyjną a masową, znać podstawy działania sieci i internetu czy też mieć choćby pobieżną wiedzę z zakresu grafiki komputerowej.

To drugie, gruntownie przejrane i starannie uzupełnione wydanie fascynującej opowieści o tajemnym życiu toczącym się wewnątrz komputerów i innych inteligentnych urządzeń. Pozwala zdobyć wiedzę o rzeczywistych mechanizmach działania komputerów osobistych, multimediów cyfrowych i internetu. Opowiada też o historii poszczególnych wynalazków i przystępnie wyjaśnia wiele pojęć, które warto przyswoić. W tym wydaniu szczególnie pieczołowicie opisano wewnętrzny świat procesorów, decydujących o działaniu komputerów i innych urządzeń. Co najważniejsze: lektura nie wymaga wiedzy technicznej, a opisy i wyjaśnienia są zrozumiałe i angażujące.

## W książce między innymi:

- różne rodzaje kodów
- system binarny i szesnastkowy, ASCII i Unicode
- bramki logiczne, pamięć i magistrala
- procesor: serce i mózg komputera
- system operacyjny a pisanie kodu

**Charles Petzold** napisał wiele książek, które dziś są uważane za klasyczne pozycje poświęcone komputerom i programowaniu. W 1994 roku otrzymał nagrodę Windows Pioneer z rąk Billa Gatesa. Zdobył też tytuł MVP. Mieszka w Nowym Jorku z żoną i dwoma kotami.

	<b>KOD KORZYŚCI</b> Sięgnij po więcej! ▶	
 <a href="http://helion.pl">helion.pl</a>	ISBN 978-83-8322-168-7	
 <b>HELION SA</b> ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	 9 788383 221687	
Cena: 99,00 zł		

Microsoft Press