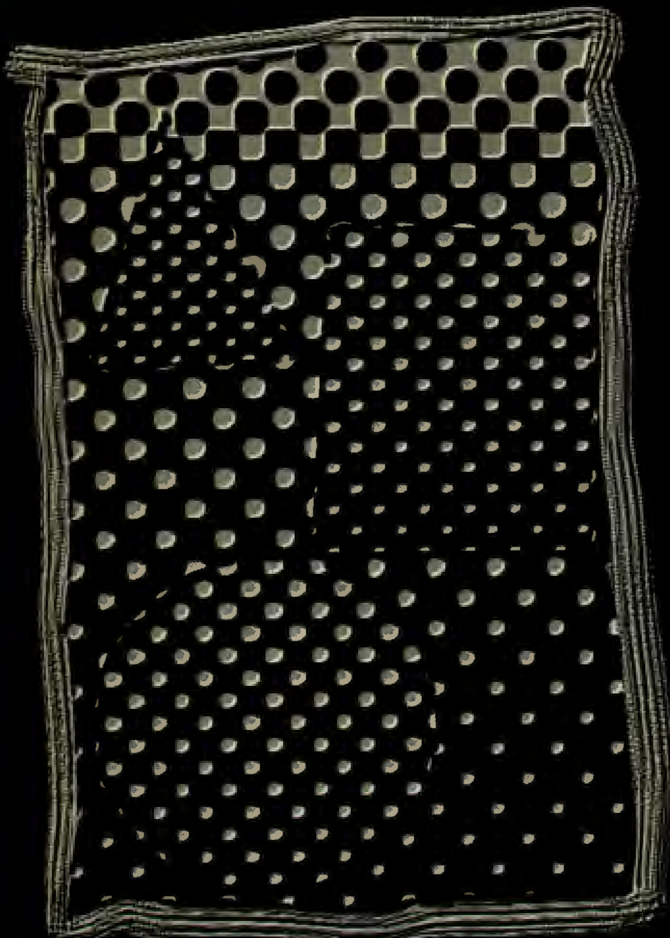
WYŻSZA SZKOŁA
INFORMATYKI STOSOWANEJ
I ZARZĄDZANIA



Henryk Spustek

ELEMENTY INFORMATYKI

WARSZAWA 2000

13,-

Seria: Skrypty WSISiZ

Skrypt zgłoszony przez
Dziekana Wydziału Zarządzania i Marketingu
dr Barbarę Mażbic-Kulmę

WYŻSZA SZKOŁA
INFORMATYKI STOSOWANEJ I ZARZĄDZANIA

Henryk Spustek

ELEMENTY INFORMATYKI

Warszawa 2000

ISBN 83-88311-17-4

Projekt graficzny okładki: Jan Młynarczyk

Druk:
Zakład Poligraficzny Jerzy Kosiński
Warszawa

3. SYSTEMY LICZBOWE - ARYTMETYKA MASZYNY CYFROWEJ

Wśród systemów liczbowych wyróżniamy systemy pozycyjne oraz niepozycyjne. Systemy pozycyjne to takie w których wartość znaku zależy od położenia (pozycji) w danej liczbie. W systemach niepozycyjnych wartości poszczególnych znaków są niezależne od pozycji jaką zajmują w zapisie liczby. Jako przykład może tu posłużyć system rzymski. System dziesiętny (przedstawiciel systemu pozycyjnego) oparty jest na podstawie 10, tzn. każdą dowolną liczbę N można zapisać w postaci:

$$N = \sum_i a_i 10^i \text{, gdzie } 0 \le a_i < 10 .$$

W przypadku ogólnym, gdy podstawą systemu jest liczba p, to N przyjmuje postać:

$$N = \sum_i a_i p^i \text{, gdzie } 0 \le a_i < p .$$

Najczęściej stosowanym systemem pozycyjnym w maszynie cyfrowej jest system dwójkowy (binarny). Jest to uzasadnione naturalną reprezentacją dwustanową jaka tworzą cyfry 0 i 1.

Postać liczb dwójkowych

Liczby przechowywane w rejestrach mikroprocesora lub w pamięci mogą mieć następującą postać:

- liczb dwójkowych bez znaku,
- liczb 8–bitowych z zakresu [0,255],
- liczb 16-bitowych z zakresu [0,65535],

- liczb dwójkowych ze znakiem (w kodzie uzupełnień do 2) przyjmujących wartość z zakresu [-128,+127] lub [-32768,+32767],
- liczb w tzw. rozpakowanym kodzie BCD – jedna cyfra dziesiętna na bajt,
- liczb w tzw. upakowanym kodzie BCD – dwie cyfry dziesiętne na bajt,

3.1. Działania arytmetyczne na liczbach binarnych

- *DODAWANIE*

$$\begin{aligned} 0+0&=0\\ 1+0&=1\\ 0+1&=1\\ 1+1&=10 \end{aligned}$$

- *ODEJMOWANIE*

$$\begin{aligned} 1-0&=1\\ 10-1&=1\\ 0-0&=0 \end{aligned}$$

- *MNOŻENIE* (wykonujemy analogicznie jak dla liczb dziesiętnych)

Przykład 1

$$\begin{array}{cccccccccccc} & & & & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1\\ & & \times & & & & & & 1 & 0 & 0 & 1\\ \hline & & & & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1\\ + & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & & & \\ \hline & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

Przykład 2

$$
\begin{array}{rccccccc}
 & & & & 1 & 0 & 1 & 0 \\
\times & & & & 1 & 0 & 1 & 1 \\
\hline
 & & & & 1 & 0 & 1 & 0 \\
 & & & 1 & 0 & 1 & 0 & \\
 & & 0 & 0 & 0 & 0 & & \\
+ & 1 & 0 & 1 & 0 & & & \\
\hline
 & 1 & 1 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

$1010_{(b)} = 10_{(d)}$

$1011_{(b)} = 11_{(d)}$

$10_{(d)} \times 11_{(d)} = 110_{(d)} = 1101110_{(b)}$

- *DZIELENIE*

Przykład 3

$$
\begin{array}{rccccc}
 & 1 & 1 & 1 & 1 & 0 \\
- & \underline{1} & \underline{1} & \underline{0} & & \\
 & & & 1 & 1 & 0 \\
 & & - & \underline{1} & \underline{1} & \underline{0} \\
 & & & - & - & -
\end{array}
\quad : \quad 1\ 1\ 0 = 1\ 0\ 1
$$

Dla porównania przedstawimy jeszcze operacje dodawania i odejmowania w systemach: dziesiętnym i dwójkowym.

LICZBA

dziesiętna	dwójkowa
$0 + 0 = 0$	$0 + 0 = 0$
$0 + 1 = 1$	$0 + 1 = 1$
$1 + 1 = 2$	$1 + 1 = 10$
$1 + 1 + 1 = 3$	$1 + 1 + 1 = 11$

Przykład 4

Przykład 4

$29_d = 11101_b$ zaś $31_d = 11111_b$

suma: $60_d = 111100_b$, co możemy wykonać pisemnie(patrz niżej).

$$
\begin{array}{r}
\underline{1111} \leftarrow \text{to reszty z przeniesień} \\
11101 \\
+\,11111 \\
\hline
111100
\end{array}
$$

Dwa algorytmy zamiany z systemu binarnego na decymalny[9]

(System binarny → System decymalny)

ALGORYTM I

1. ZAPISZ LICZBĘ BINARNĄ
2. DOKŁADNIE POD KOLEJNYMI CYFRAMI, ZACZYNAJĄC OD KOŃCA ZAPISZ POTĘGI LICZBY DWA
3. DODAJ TE POTĘGI LICZBY DWA, NAD KTÓRYMI JEST CYFRA JEDEN

Przykład 5

$$1010110_{(b)} = ?_{(d)}$$

[9] A.Andriusz, M.Sokołowski Mapa pamięci IBM/PC w przykładach Lynx-SFT, Warszawa 1999

0	1	0	1	1	0		
32	16	8	4	2	1		
+	16	+	4	+2		=	86

Wobec tego: $1010110_{(b)} = 64_{(d)}$.

(System binarny ⟶ System decymalny)

;ORYTM II

Ź PIERWSZĄ CYFRĘ LICZBY BINARNEJ

)WOJ JĄ

)AJ DO NIEJ KOLEJNĄ CYFRĘ LICZBY BINARNEJ, OTRZYMASZ
ЛĘ

VRÓĆ DO PUNKTU 2

Y DODASZ JUŻ WSZYSTKIE CYFRY, OTRZYMANA SUMA
)ZIE WARTOŚCIĄ DZIESIĘTNĄ PRZELICZANEJ LICZBY
ARNEJ

,kład 6

$)110_{(b)} = ?_{(d)}$

LICZBA BINARNA	**1**	**0**	**1**	**0**	**1**	**1**	**0**
)OBIERAMY PIERWSZĄ CYFRĘ	1	2	5	10	21	43	**86**
1 • 2 = 2 2 + 0 = 2							
2 • 2 = 4 4 + 1 = 5							
5 • 2 = 10 10 + 0 = 10							
)0 • 2 = 20 20 + 1 = 21							
21 • 2 = 42 42 + 1 = 43							
3 • 2 = 86 86 + 0 = 86							

BCD (Binary Coded Decimal) - cyfry dziesiętne zakodowane w syste-
binarnym

W przypadku zapisu jednej cyfry na czterech bitach, zmiennym logicznym A, B, C, D przyporządkowane są sygnały We/Wy reprezentujące potęgi liczby 2.

Algorytm zapisu liczby dziesiętnej w kodzie BCD jest następujący:

1. Liczbę dziesiętną podziel na osobne cyfry,

   np. 94 $\rightarrow$ 9 4

2. Każdą cyfrę zamień na czterobitową liczbę binarną

   9 $\rightarrow$ 1001, 4 $\rightarrow$ 0100

3. Otrzymany bajt (1001 0100) jest liczbą 94 zapisaną w kodzie BCD

Zalety stosowania kodu BCD:

- łatwe przeliczanie na kod dziesiętny,
- łatwa zamiana liczb większych od 255.

W kodzie BCD zapisana jest w komputerze informacja dotycząca daty i czasu.

Reprezentacja liczb ujemnych - Kod uzupełnień do dwu *(U2-two's complement).*

Przykład 7

Przedstawić liczbę $\mathbf{29_d}$ i $\mathbf{-29_d}$ w systemie binarnym.

$\mathbf{29_d = 11101_b}$

Krok 1: Negacja wyniku

Otrzymujemy: **00010**

Krok 2: do wyniku dodajemy liczbę **1** Stąd: $\mathbf{-29_d = (1)00010_b}$.

Reprezentacja liczby rzeczywistej

Część całkowita . część ułamkowa

zapisane binarnie

każda część oddzielnie

Np.

$0.6875_d = 0.1011_b$

$29.6875_d = 11101.1011_b$

System szesnastkowy (heksadecymalny)

Pomimo tego, że system dwójkowy najwierniej odzwierciedla sposób zapisu liczb w pamięci komputera, to o wiele wygodniejszy w użyciu jest zapis szesnastkowy (heksadecymalny). Podstawą tego systemu jest liczba 16, a cyframi symbole 0 ÷ 9 oraz litery A ÷ F (patrz tabela niżej).

cyfra szesnastkowa	wartość dziesiętna	wartość binarna
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Przykład 8

62450_d = $F3F2_h$= 1111001111110010_b

3.2 Zamiana podstawy systemu

Aby przejść z jednego dowolnego systemu na drugi można wykorzystać następujący sposób. Załóżmy, że liczbę N należy wyrazić używając nowej podstawy q. Liczbę tę można zapisać ogólnie w postaci:

$$N = a_m \cdot q^m + a_{m-1} \cdot q^{m-1} + \ldots + a_1 \cdot q + a_0 .$$

Naszym zadaniem jest wyznaczenie współczynników a_i nowej postaci liczby N. Dzieląc liczbę N przez podstawę q otrzymujemy całkowity iloraz S oraz resztę R.

$$S = \frac{N}{q} = a_m \cdot q^{m-1} + a_{m-1} \cdot q^{m-2} + \ldots + a_2 \cdot q + a_1 ,$$

$$R = a_0 .$$

Reszta a_0 jest pierwszą cyfrą na lewo od kropki dziesiętnej liczby N wyrażonej w nowym systemie. Dzieląc iloraz ponownie przez q otrzymujemy kolejną cyfrę liczby N w nowej postaci. Oczywiście, działanie wykonywane jest w tym samym systemie w którym zapisana jest liczba N.

W podobny sposób dokonuje się zamiany ułamków. Działanie dzielenia zastępowane jest mnożeniem. Załóżmy, że ułamek N należy wyrazić używając nowej podstawy q. Ułamek ten zapisujemy w postaci:

$$N = a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \ldots + a_m \cdot q^{-m} .$$

Kolejne cyfry a_i otrzymujemy poprzez kolejne mnożenie powyższego wyrażenia przez q:

$$q \cdot N = a_1 + a_2 \cdot q^{-1} + \ldots + a_m \cdot q^{-m+1} .$$

Częścią całkowitą jest a_1, czyli pierwsza cyfra na prawo od kropki dziesiętnej. Wykonując kolejne mnożenia dla pozostałej części ułamkowej otrzymujemy wszystkie cyfry ułamka zapisanego w nowym systemie o podstawie q[10].

Przykład 9

$27_{(10)} = ?_{(2)}$

$27 : 2 = 13$ R = 1

$13 : 2 = 6$ R = 1

$6 : 2 = 3$ R = 0

$3 : 2 = 1$ R = 1

$1 : 2 = 0$ R = 1, stąd $27_{(10)} = 11011_{(2)}$

Sprawdzimy jeszcze otrzymany wynik:

$$11011_{(2)} = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 1 + 2 + 8 + 16 = 27_{(10)}$$

Przykład 10

$0{,}6875_{(d)} = ?_{(b)}$ - uwaga -

oznaczenia (d), (b) używane
są tu wymiennie z (10), (2)

0,6875 x 2

1,3750 x 2

0,7500 x 2

1,5000 x 2

1,0000 stąd $0{,}6875_{(d)} = 0{,}1011_{(b)}$

---

[10] J. Bromirski Podstawy systemów informatycznych, PWN Warszawa 1982

Sprawdzenie:

$$0{,}1011_{(b)} = 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = \frac{11}{16} = 0{,}6875_{(d)}.$$

Przykład 11

$0{,}8125_{(d)} = ?_{(b)}$

0,8125 x 2

1,6250 x 2

1,2500 x 2

0,5000 x 2

1,0000 stąd $0{,}8125_{(d)} = 0{,}1101_{(b)}$

Sprawdzenie:

$$0{,}1101_{(b)} = 1\cdot 2^{-1} + 1\cdot 2^{-2} + 0\cdot 2^{-3} + 1\cdot 2^{-4} = \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16} = 0{,}8125_{(d)}$$

Przykład 12

$0{,}4375_{(d)} = ?_{(b)}$

0,4375 x 2

0,8750 x 2

1,7500 x 2

1,5000 x 2

1,0000 stąd $0{,}4375_{(d)} = 0{,}0111_{(b)}$

$$0{,}0111_{(b)} = 0\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3} + 1\cdot 2^{-4} = \frac{1}{4} + \frac{1}{8} + \frac{1}{16} = \frac{7}{16} = 0{,}4375_{(d)}$$

Przykład 13

$62450_{(10)} = ?_{(16)}$

$62450 : 16 = 3903 \qquad R = 2$

$3903 : 16 = 243 \qquad R = 15 \leftarrow F$

$243 : 16 = 15 \quad R = 3$

$15 : 16 = 0 \qquad R = 15 \leftarrow F$

stąd $\quad 62450_{(10)} = F3F2_{(16)}$.

Sprawdzenie:

$$F3F2_{(16)} = 2\cdot16^0 + 15\cdot16^1 + 3\cdot16^2 + 15\cdot16^3 = 2 + 240 + 768 + 61440 = 62450$$

Przykład 14

$103_{(10)} = ?_{(2)}$

$103 : 2 = 51 \qquad R = 1$

$51 : 2 = 25 \qquad R = 1$

$25 : 2 = 12 \qquad R = 1$

$12 : 2 = 6 \qquad R = 0$

$6 : 2 = 3 \qquad R = 0$

$3 : 2 = 1 \qquad R = 1$

$1 : 2 = 0 \qquad R = 1$ $\quad$ stąd $\quad 103_{(10)} = 1100111_{(2)}$

Przykład 15

$0{,}1_{(3)} = ?_{(10)}$

Wykonujemy mnożenie $\cdot 10$ ale w systemie trójkowym !

Czyli $10_{(d)} = 101_{(3)}$, więc:

0,1 x 101

... itd.

Stąd $0{,}1_{(3)} = 0{,}333\ldots_{(10)}$.

Przykład 16

$(1)1101_{(b)} = ?_{(d)}$

Liczba jest ujemna (jedynka na najstarszym bicie).

Krok 1 - odejmujemy 1

$$\begin{array}{rccccc} & 1 & 1 & 1 & 0 & 1 \\ - & & & & & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Krok 2 - negacja wyniku

$$\begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \end{array}$$

$$0011_{(b)} = 3_{(d)}$$

Wobec tego: $(1)1101_{(b)} = -3_{(d)}$.

Liczby rzeczywiste x zapamiętywane są w maszynie cyfrowej w tzw. postaci zmiennopozycyjnej: $x = M \cdot N^W$,

gdzie M - mantysa liczby,

W - wykładnik części potęgowej,

N - podstawa.

W systemie dwójkowym podstawa jest równa 2 zaś w dziesiętnym 10.

Widać zatem, że w zapisie zmiennopozycyjnym liczba rzeczywista przedstawiana jest za pomocą dwóch grup bitów. Pierwsza grupa, tworząca mantysę M, jest interpretowana jako część ułamkowa i spełnia warunek: $\frac{1}{2} \le M < 1$.

Druga grupa, tworząca wykładnik W jest interpretowana jako liczba całkowita, przy czym zakres liczbowy wykładnika W decyduje o zakresie liczb rzeczywistych przechowywanych w maszynie.

Przykład 17

Załóżmy, że liczbę M w zapisie dwójkowym określa 5 bitów, zaś liczbę W trzy bity, przy czym pierwsze bity oznaczają znaki tych liczb (liczbę ujemną rozpoczyna bit o zawartości 1), to zapis:

$$x = (1)1101 \quad (0)10 \ ,$$

oznacza liczbę równą liczbie dziesiętnej:

$$-0{,}1101_{(b)} \cdot 2^{10_{(b)}} = -\left(\frac{1}{16}+\frac{1}{4}+\frac{1}{2}\right)\cdot 2^2 = -\frac{13}{4} = -3{,}25 \, .$$

Wykonamy teraz obliczenia odwrotne, tzn. mając dana liczbę $-3{,}25_{(d)}$ znajdziemy jej postać dwójkową w zapisie jak wyżej.

<u>Krok 1</u>

Znajdujemy mantysę $M = \dfrac{3{,}25}{4} = 0{,}8125_{(d)}$

<u>Krok 2</u>

Zamieniamy mantysę na postać binarną: $0{,}8125_{(d)} = 0{,}1101_{(b)}$

<u>Krok 3</u>

Zapisujemy $-3{,}25_{(d)} = (0)1101 \quad (0)10_{(b)}$.

3.3 Działania logiczne na liczbach binarnych - przykłady

1. AND

$$\begin{array}{r} 11100111 \\ \underline{11000011} \\ 11000011 \end{array}$$

2. OR

$$\begin{array}{r} 11100111 \\ \underline{11000011} \\ 11000011 \end{array}$$

3. NAND (NOT END)

$$\begin{array}{r} 11100111 \\ \underline{11000011} \\ 11000011 \end{array}$$

4. NOT

$$\begin{array}{r} 11100111 \\ \underline{11000011} \\ 11000011 \end{array}$$

5. NOR (NOT OR)

$$\begin{array}{r} 11100111 \\ \underline{11000011} \\ 11000011 \end{array}$$

6. XOR - (eXclusive OR)

$$\begin{array}{r} 11100111 \\ \underline{11000011} \\ 11000011 \end{array}$$

ZADANIA

1. Sprawdź poprawność poniższych obliczeń:

a) $29{,}6875_{(10)} = 11101.1011_{(2)}$,

b) $0{,}25_{(10)} = 0.01_{(2)}$,

c) $0{,}75_{(10)} = 0.11_{(2)}$,

d) $14{,}5_{(10)} = 111.1_{(2)}$.

2. Obliczyć:
   a) $-0{,}1101_{(b)} = ?_{(d)}$,

Odp. -3,25.

b) $-29_{(10)} = ?_{(2)}$.

Odp. (1)00011

Dotychczasowe wydawnictwa
WYŻSZEJ SZKOŁY INFORMATYKI STOSOWANEJ I ZARZĄDZANIA

- Z. Stachowiak: *Ekonomia. Zarys podstawowych problemów.* 1998; Wyd. 2. 2000.
- Z. Mikołejko: *Elementy filozofii.* 1998; Wyd. 2 popr. i rozsz. 1998; Wyd. 3 popr. i rozsz. 1999.
- W. Arczewska: *Bazy danych Oracle* 1998; Wyd. 2 popr. i rozsz. 1999; Wyd. 3 popr. 1999.
- S. Bożek, P. Cholajda, G. Szkatuła: *Wstęp do bazy danych MS Access dla Windows 95.* 1998.
- T. Łuba: *Podstawy układów logicznych.* 1998; Wyd. 2 popr. 1999.
- G. Szkatuła, A. Pogorzelec: *Ćwiczenia z bazy danych Microsoft Access 97.* 1999; Wyd. 2 rozsz. 1999.
- A. Żochowski: *L E M Laboratorium eksperymentów matematycznych.* 1999; Wyd. 2. popr. 2000.
- J. Hołubiec, red.: *Analiza systemowa w finansach i zarządzaniu. Wybrane problemy.* 1999.
- M. Doros: *Przetwarzanie obrazów. Materiały pomocnicze.* Cz.1, 2. 1999; Wyd. 2 popr. 1999.
- L. Oleksyn: *Istota, zakres i cechy rachunku kosztów.* 1999.
- L. Oleksyn: *Zadania rachunku kosztów w zarządzaniu.* 1999.
- L. Oleksyn: *Ekonomia - zarys wykładu.* 1999.
- Z. Nahorski: *Metoda najmniejszych kwadratów.* Cz. 1, 2. 1999.
- O. Hryniewicz: *Wykłady ze statystyki.* 1999.
- P. Cholajda: *Systemy informatyczne w MS ACCESS 97 PL.* 1999.
- K. Liderman: *Bezpieczeństwo informacji w systemach informatycznych.* 2000.
- M. Barszczewski: *Zarządzanie sieciami telekomunikacyjnymi.* 2000.
- J. Borkowski, M. Dyrda, L. Kanarski, B. Rokicki: *Wybrane problemy psychologii organizacji. O konflikcie i negocjacjach.* 2000.
- J. Jarmakiewicz: *Sieci teleinformatyczne.* Cz. 1, 2. 2000.
- T. Łuba: *Synteza układów logicznych.* 2000.
- H. Spustek: *Elementy informatyki.* 2000.

WYŻSZA SZKOŁA
INFORMATYKI STOSOWANEJ
I ZARZĄDZANIA

pod auspicjami
Polskiej Akademii Nauk

ZAŁOŻYCIELEM
Wyższej Szkoły Informatyki Stosowanej i Zarządzania
jest
Fundacja Krzewienia Nauk Systemowych
powołana z inicjatywy
Prezesa
POLSKIEJ AKADEMII NAUK

FUNDATOREM
Fundacji Krzewienia Nauk Systemowych
jest
POLSKA AKADEMIA NAUK

ORGANEM
sprawującym nadzór jest
MINISTERSTWO EDUKACJI NARODOWEJ

Wyższa Szkoła Informatyki Stosowanej i Zarządzania
prowadzi studia wyższe na kierunkach:
INFORMATYKA
ZARZĄDZANIE I MARKETING

SIEDZIBA
Instytut Badań Systemowych
Polskiej Akademii Nauk
ul. Newelska 6, 01-447 Warszawa

ISBN 83-88311-17-4