

Techniki multimedialne



Digitalizacja

podstawą rozwoju systemów multimedialnych.

Digitalizacja czyli obróbka cyfrowa oznacza ***przetwarzanie*** wszystkich typów informacji - słów, dźwięków, ilustracji, wideo i liczb - na ***kod cyfrowy*** (w praktyce zero-jedynkowy), rozpoznawany i właściwie interpretowany przez odpowiednie urządzenia elektroniczne.



SYSTEMY LICZENIA



Systemy liczbowe

- System dziesiętny jest systemem pozycyjnym, co oznacza, że wartość liczby zależy od pozycji na której się ona znajduje np. w liczbie 333 każda cyfra oznacza inną wartość bowiem:

$$333 = 3 * 100 + 3 * 10 + 3 * 1$$

- każdą z cyfr mnożymy przez tzw. wagę pozycji, która jest kolejną potęgą liczby 10 będącej podstawą systemu liczenia co możemy zapisać jako:

$$333_{(10)} = 3 * 10^2 + 3 * 10^1 + 3 * 10^0$$

a dowolną liczbę dziesiętną można zapisać jako:

$$L_{(10)} = a_n * 10^n + a_{n-1} * 10^{n-1} + a_{n-2} * 10^{n-2} + \dots + \\ + \dots a_2 * 10^2 + a_1 * 10^1 + a_0 * 10^0$$

Przy czym współczynniki a_n mogą mieć wartość 0,1,...,9



Inne systemy

Można stworzyć dowolny pozycyjny system liczenia o podstawie np. 2, 3, 4, 7, 8, 16.

W technice komputerowej praktyczne zastosowanie znalazły systemy:

- o podstawie 2 - tzw. **system binarny** (dwójkowy) używany do przechowywania i przetwarzania danych przez układy elektroniczne komputera
- o podstawie 16 - tzw. **system heksadecymalny** (szesnastkowy), używany głównie do prezentacji niektórych danych m.in adresów komórek pamięci



System binarny

Zgodnie z pokazanym poprzednio rozwinięciem (na przykładzie systemu dziesiętnego) liczbę w systemie o podstawie 2 możemy więc przedstawić jako:

$$L_{(2)} = a_n * 2^n + a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} \dots + \dots a_2 * 2^2 + a_1 * 2^1 + a_0 * 2^0$$

a współczynniki a_n mogą przybierać tylko dwie wartości: **0** lub **1**

Uwaga! *Ilość dostępnych cyfr w systemie jest równa podstawie systemu, a więc w systemie dziesiętnym – 10, w systemie dwójkowym – 2 itd.*



System szesnastkowy

Analogicznie do systemu dziesiętnego czy binarnego liczbę w systemie szesnastkowym (o podstawie 16) możemy przedstawić jako:

$$L_{(16)} = a_n * 16^n + a_{n-1} * 16^{n-1} + a_{n-2} * 16^{n-2} \dots + \\ \dots + a_2 * 16^2 + a_1 * 16^1 + a_0 * 16^0$$

natomiast współczynniki a_n mogą być liczbami:

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15



System szesnastkowy

Aby zapis liczby był jednoznaczny, na każdej pozycji powinna być umieszczona tylko 1 cyfra.

I tak np. pisząc 145 nie można mieć wątpliwości czy kolejne cyfry tak zapisanej liczby to:

1 4 5 czy 14 5

Dlatego też cyfry od 10 do 15 zastąpiono w zapisie literami:

10 - A, 11 - B, 12 - C, 13 - D, 14 - E, 15 - F




Konwersja liczb

Posługiwanie się różnymi systemami liczbowymi wymaga umiejętności:

- ✱ **przedstawiania liczb** w różnych systemach
- ✱ **konwersji** (zamiany) liczby przedstawionej w jednym systemie na liczbę w innym systemie.



Zamiana liczby dziesiętnej na binarną

69		
34	1	Najmłodszy bit  Najstarszy bit
17	0	
8	1	
4	0	
2	0	
1	0	
0	1	

Podstawowy sposób polega na kolejnym **dzieleniu liczby dziesiętnej przez 2 z resztą** i zapisaniu liczby od najstarszego do najmłodszego bitu więc:

$$69_{(10)} = 1000101_{(2)}$$

Każdą pozycję liczby binarnej nazywamy **bitem** (binary digit) i jest to najmniejsza jednostka ilości informacji



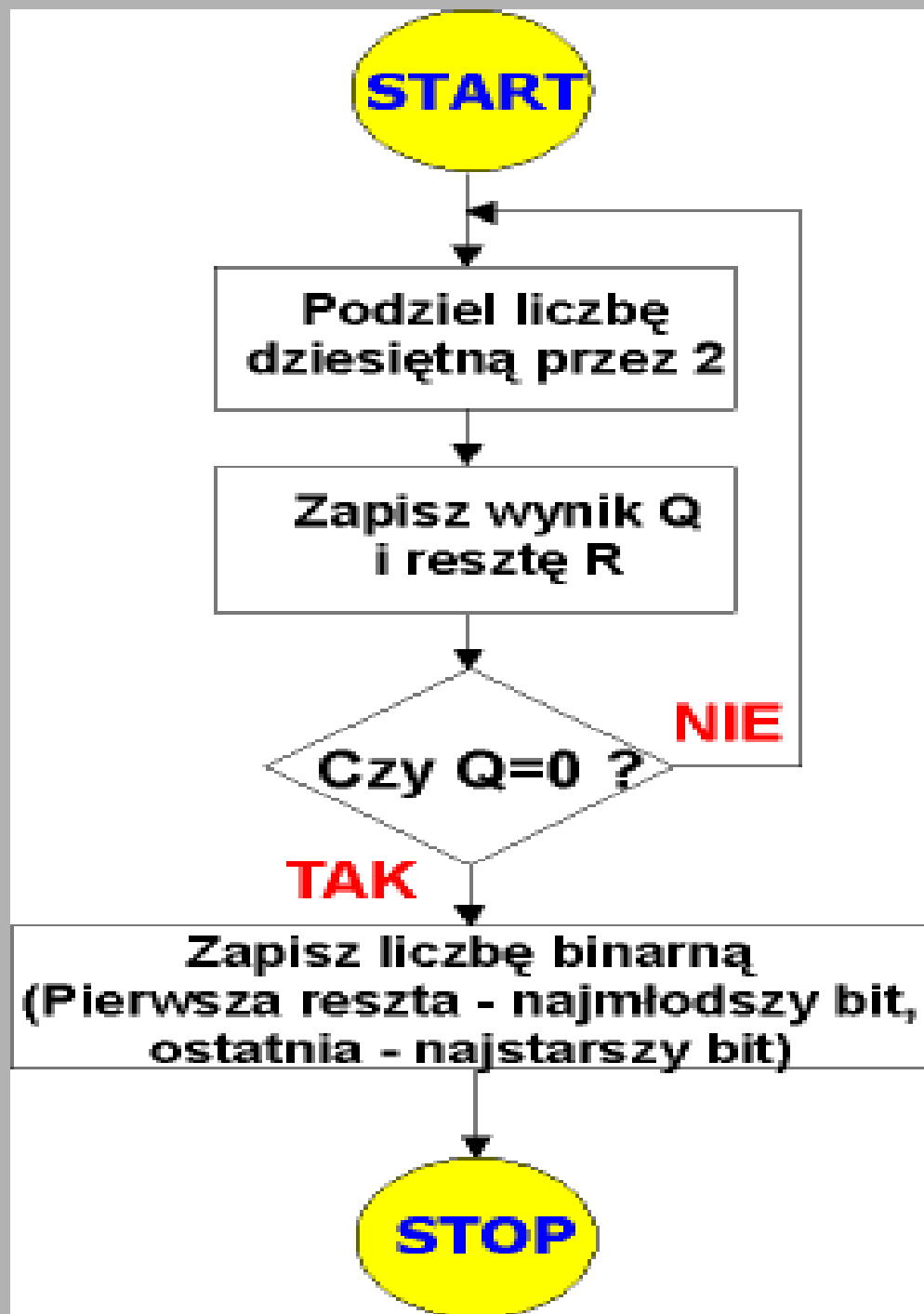
Zamiana liczby binarnej na dziesiętną

Aby obliczyć dziesiętną wartość naszej liczby binarnej mnożymy cyfrę stojącą na każdej pozycji przez jej wagę, czyli kolejną potęgę liczby **2** będącej podstawą systemu

$$\begin{aligned} &1000101_{(2)} = \\ &= 1*2^6 + 0*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = \\ &= 64 + 0 + 0 + 0 + 4 + 0 + 1 = 69 \end{aligned}$$



- Algorytm zamiany liczby dziesiętnej na binarną



Konwersja liczby dziesiętnej do systemu heksadecymalnego

Liczba dziesiętna 69 zapisana binarnie: **1000101**

◆ **Algorytm zamiany liczby binarnej na heksadecymalną jest następujący:**

□ **dzielimy liczbę binarną na tzw. **kęsy** o długości 4 bity (licząc od ostatniej pozycji) czyli:**

100 0101

□ **Dla każdego kęsa znajdujemy wartość dziesiętną i zapisujemy ją w postaci heksadecymalnej**
binarnie

100 0101



dziesiętnie

4

5

heksadecymalnie

45

tak więc: $45_{(16)} = 4 * 16^1 + 5 * 16^0 = 64 + 5 = 69$



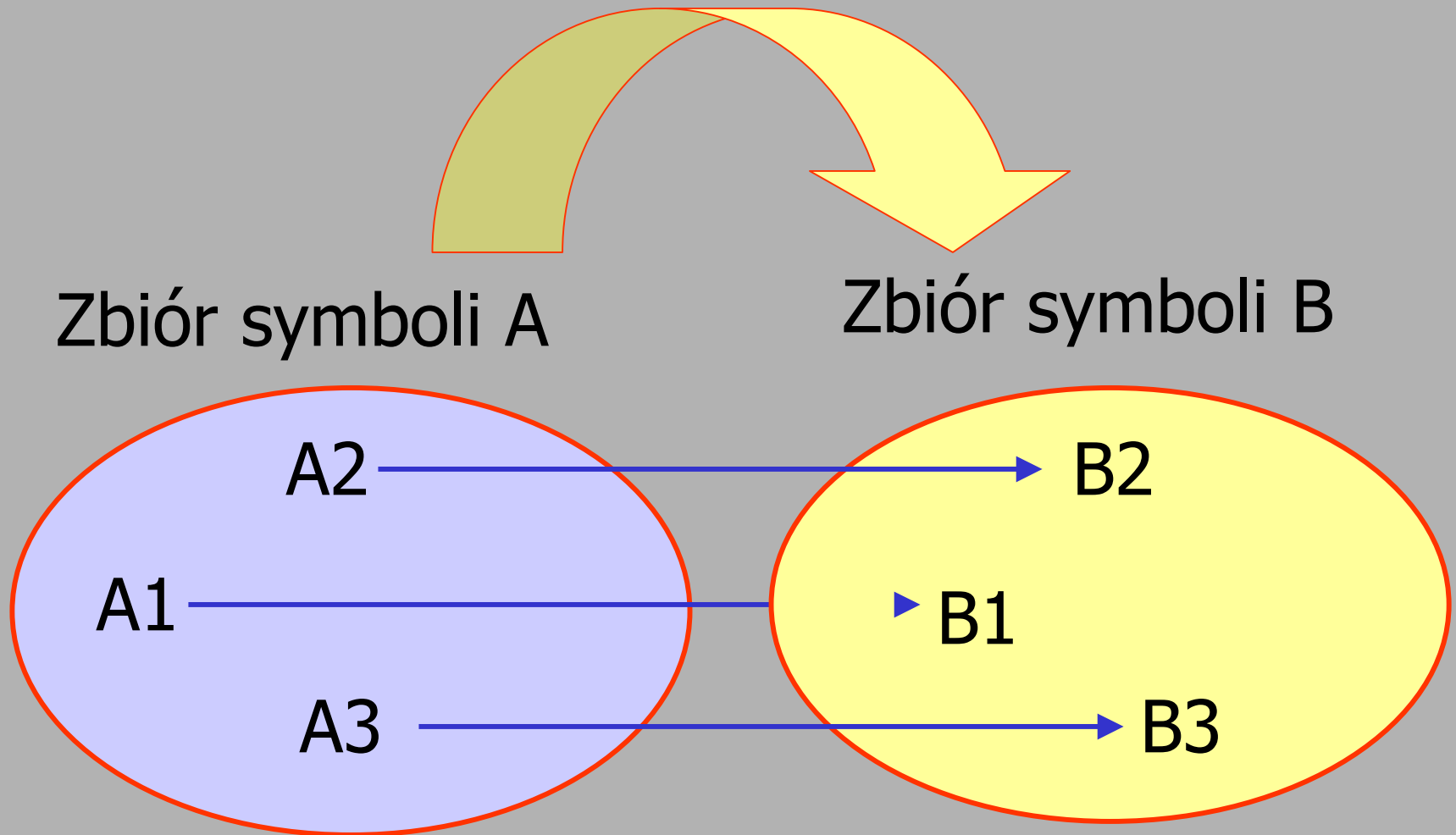
Kodowanie informacji

Przedstawiając liczbę dziesiętną w systemie binarnym lub heksadecymalnym należy pamiętać, że w dalszym ciągu jest to ta sama liczba lecz przedstawiona za pomocą innego zestawu znaków.

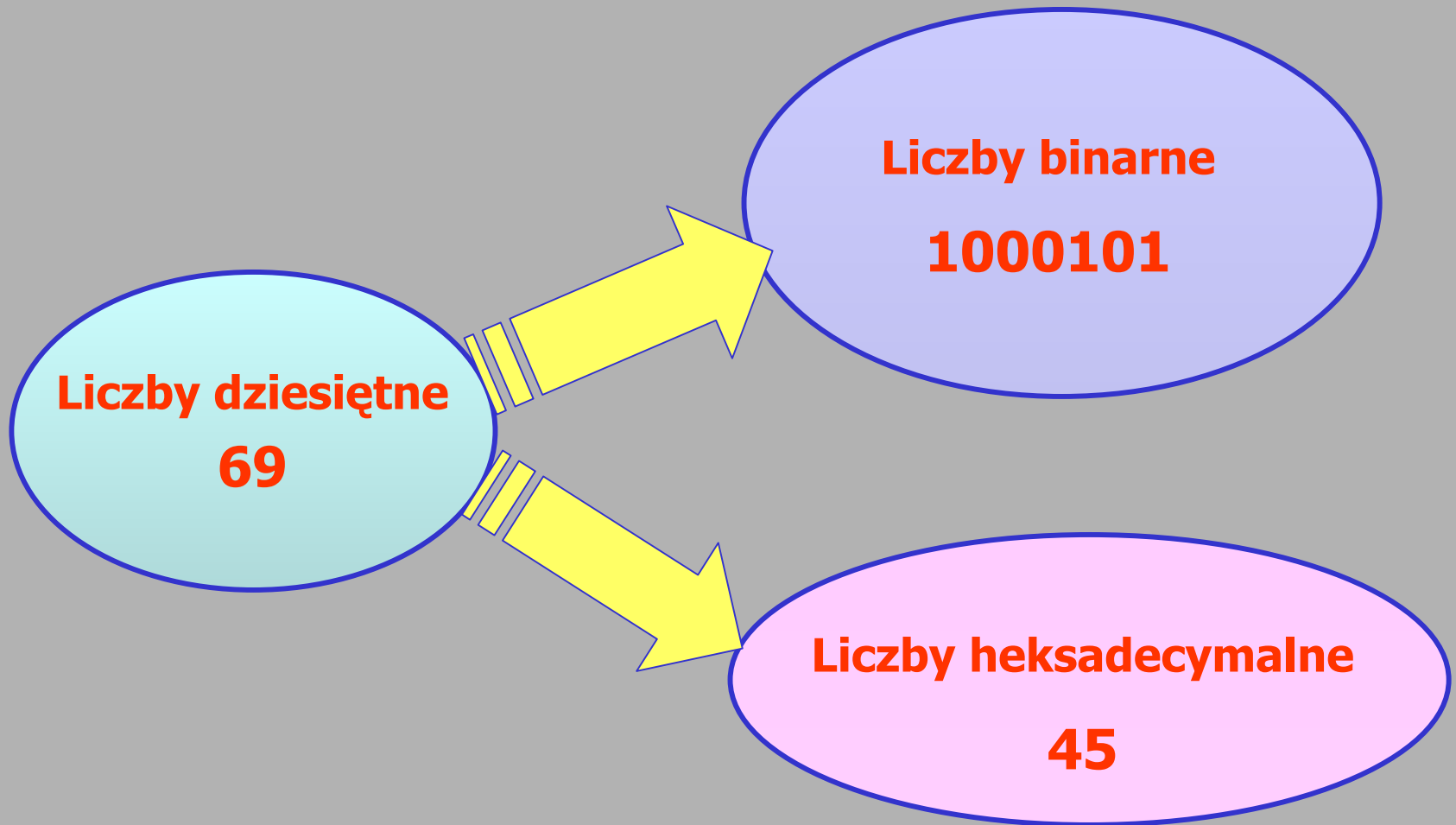
Można więc mówić o **kodzie binarnym** czy też **kodzie heksadecymalnym**.



Zasada tworzenia kodu



Kodowanie liczb



Kod ASCII

Do przechowywania i przetwarzania danych przez układy elektroniczne komputera używany jest **system binarny**.

Konieczne więc jest przedstawienie tekstu za pomocą liczb czyli jednoznaczne przyporządkowanie literom i innym znakom alfanumerycznym liczb (numerów).

W ten sposób powstał w 1965 r. **kod ASCII** (**American Standard Code for Information Interchange**).



Kod ASCII

Kod ASCII jest kodem 7 bitowym, za pomocą którego można przedstawić:

$$2^7=128$$

znaków.

W 1981 r. Firma IBM wprowadziła rozszerzony do 8 bitów kod, co pozwala na przedstawienie 256 znaków (w tym znaki specjalne, graficzne, matematyczne i diakrytyczne znaki narodowe).



Fragment tabeli kodu ASCII

Znak	Kod dzies.	Kod binarny	Znak	Kod dzies.	Kod binarny
A	65	01000001	a	97	00110001
B	66	01000010	b	98	00110010
C	67	01000011	c	99	00110011
K	75	01001011	k	107	01101011
L	76	01001100	l	108	01101100
Ż	171	10101011	Ž	189	10111101
ı	179	10110011	Ǻ	198	11000110
+	188	10111100	-	196	11000100

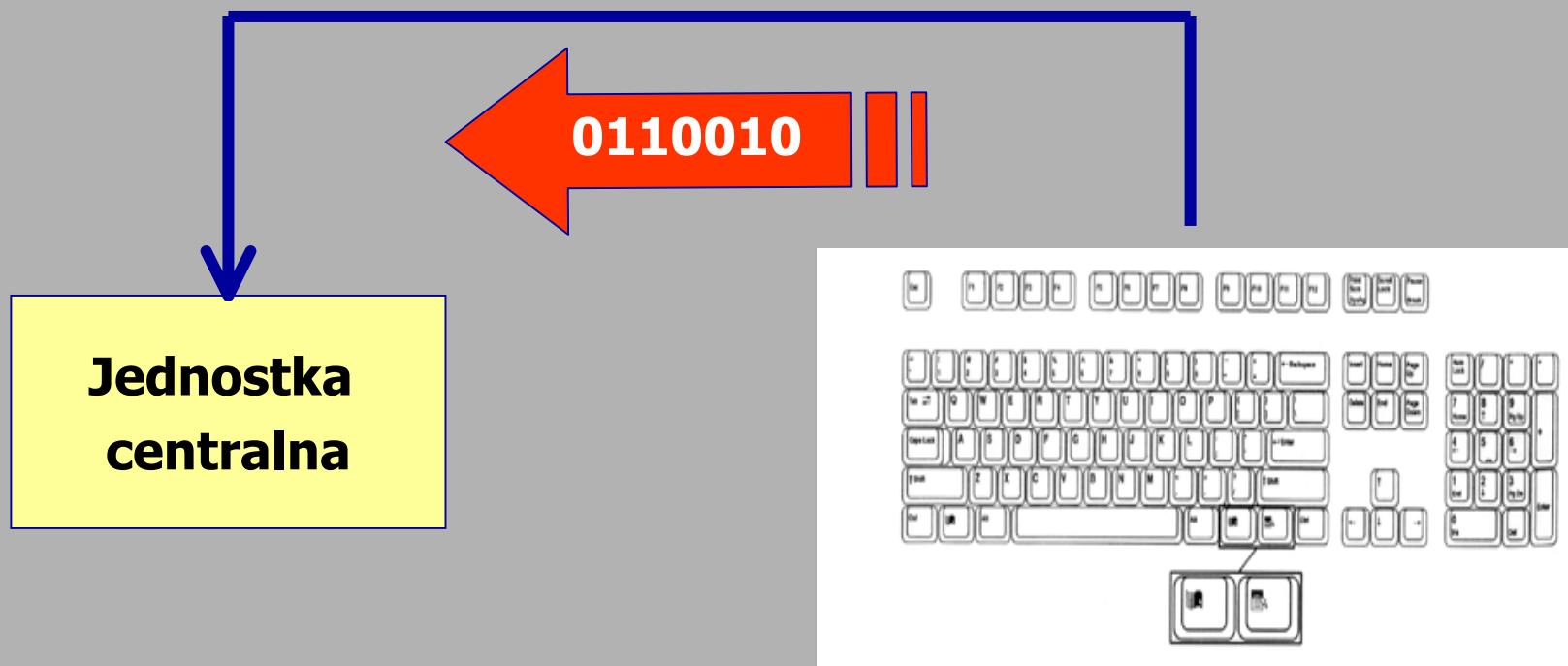


Kod UNICODE

- ▶ 256 znaków alfanumerycznych jakie można zakodować za pomocą rozszerzonego kodu ASCII nie dawało możliwości zakodowania znaków diakrytycznych wielu języków np.:japońskiego, arabskiego, hebrajskiego itp.
- ▶ Stworzono kod o nazwie **UNICODE** o długości 16 bitów dla każdego znaku, a to daje już możliwość zakodowania 2^{16} czyli 65536 znaków



Kodowanie w praktyce



Jednostki informacji

1kbit [Kb]= 2^{10} b=1024 bity

1Mbit[Mb]=1024 Kb=1048576 bity

1 byte(bajt)=8 bitów

1kB = 2^{10} bajtów=1024 B

1MB=1024 KB=1048576 B

