# INFORMATYKA

SYSTEM DWÓJKOWY

## CEL LABORATORIUM

Celem zajęć jest zapoznanie uczestniczki/uczestnika z systemem dwójkowym. Podczas zajęć przedstawiony zostanie system dwójkowy, metody konwersji liczb dziesiętnych na binarne i odwrotnie, operacje binarne („nie", „i", „lub", „albo"), metody dodawania i odejmowania liczb binarnych, system uzupełnień do dwóch i standard IEEE 754.

## WAŻNE INFORMACJE O ZAJĘCIACH

Niniejsza instrukcja oraz instrukcje przedstawione na kolejnych zajęciach zawierają kilka elementów pozwalających lepiej zorientować się w treści tychże.

- Sekcje oznaczone na **CZERWONO** (tak jak ta) są **ważne** i nie powinny zostać pominięte
- Sekcje oznaczone na CZARNO (takie jak sekcja „CEL LABORATORIUM" powyżej) pełnią funkcję informacyjną i mogą zostać pominięte, **jeżeli** uczestniczka/uczestnik zajęć nie uzna ich za interesujące.
- Sekcje oznaczone na **NIEBIESKO** zawierają materiały dodatkowe, których umieszczenie w niniejszej instrukcji było niemożliwe lub niepraktyczne. Mogą zostać pominięte, lecz zapoznanie się z nimi może znacznie przyspieszyć pracę.

Z treścią instrukcji (jeżeli jest dostępna) należy zapoznać się najpóźniej w **dniu poprzedzającym** zajęcia. Podczas zajęć mogą zostać zadane pytania (zarówno w formie ustnej, jak i pisemnej), których celem będzie stwierdzenie stopnia przygotowania uczestników do zajęć.

Podczas zajęć uczestniczka/uczestnik powinien wykonać **obowiązkowe** zadania znajdujące się w instrukcji. Niewykonanie zadań podczas trwania laboratorium może skutkować **oceną negatywną**.

W instrukcji znajdują się również zadania dodatkowe, których wykonanie podczas zajęć jest **nieobowiązkowe**. Są one oznaczone symbolem "Ⓟ". Pierwsza uczestniczka/pierwszy uczestnik zajęć, który zgłosi wykonanie takiego zadania, otrzyma plusa. Ilość plusów jakie można zdobyć podczas zajęć jest nieograniczona. Zadania wykonane przed rozpoczęciem zajęć nie będą nagradzane, chyba że nikt inny nie zgłosi ich wykonania. W przypadku konfliktów, nagrodzona zostanie osoba z mniejszą liczbą plusów.

Jeżeli uczestniczka/uczestnik natrafi na problem, powinien spróbować go rozwiązać samodzielnie (sięgając po informacje zawarte na Internecie lub bazując na własnej wiedzy i doświadczeniu). Jeżeli rozwiązanie nie zostanie znalezione w sensownym[1] czasie, należy skonsultować się z innymi, pobliskimi uczestnikami, a w razie dalszego braku rozwiązania,

---

[1] Za sensowny czas uznaje się czas nie krótszy niż 5 minut i proporcjonalny do problemu oraz pozostałej do realizacji ilości materiału.

z prowadzącym zajęcia. Brak zrozumienia nie jest jednoznaczny z brakiem przygotowania (chyba, że bezpośrednio z niego wynika).

Powyższe informacje dotyczą **wszystkich** zajęć laboratoryjnych i instrukcji. **Nie będą one powtarzane** w kolejnych instrukcjach.

## SYSTEM DZIESIĘTNY

Z liczbami człowiek ma do czynienia od najwcześniejszych lat. Najpierw uczy się rozpoznawać ich wartość, a w dalszym etapie poznaje stopniowo kolejne operacje, począwszy od podstawowych (jak dodawanie), a skończywszy na bardziej wymagających (jak silnia). Operowanie liczbami z czasem staje się tak powszednie, że człowiek przestaje dopuszczać do siebie możliwość, że można się bez nich obejść. Taki dysonans spotyka większość osób na studiach, które pierwszy raz widzą pochodne i całki. Ta powszechność sprawia również, że przestaje się zwracać uwagę na znaczenie systemu, w którym liczby zapisujemy.

Na co dzień operujemy wartościami zapisanymi w dziesiętnym systemie liczbowym. W systemie tym istnieje dziesięć cyfr (0, 1, 2, 3, 4, 5, 6, 7, 8 i 9), a dokładna wartość liczby, wynika nie tylko z tego, z jakich cyfr się ona składa, ale także z tego, na jakiej pozycji poszczególne cyfry się znajdują. Wartość dowolnej liczby można więc poznać, mnożąc poszczególne jej cyfry, przez 10 podniesione do odpowiedniej potęgi. Dla liczb jednocyfrowych, będzie to $10^0$, czyli 1. Dla liczb dwucyfrowych, pierwszą cyfrę mnożymy przez $10^1$, a drugą przez $10^0$. Analogicznie postępujemy dla liczb posiadających więcej cyfr.

Dla przykładu, wartość liczby 2394, zapisać można w następujący sposób:

$$2394 = 2 * 10^3 + 3 * 10^2 + 9 * 10^1 + 4 * 10^0$$

## SYSTEM DWÓJKOWY

Wartości w urządzeniach elektronicznych, nie są jednak przechowywane w systemie dziesiętnym. Byłoby to niezwykle niepraktyczne, gdyż wymagałoby od urządzeń rozpoznanie 10 różnych poziomów napięcia. Projektując urządzenie, o wiele łatwiej jest rozróżnić dwa poziomy napięcia – brak napięcia i napięcie różne od zera. Dlatego też większość urządzeń korzysta z systemu dwójkowego (binarnego).

W systemie tym istnieją wyłącznie dwie cyfry – 0 i 1. Tak jak w systemie dziesiętnym, wartość liczby wynika z tego, z jakich cyfr się składa oraz z tego, na jakiej pozycji poszczególne cyfry się znajdują. Z tą różnicą, że cyfry nie są mnożone przez 10 podniesione do potęgi, a 2 podniesione do potęgi. Poszczególne cyfry liczby binarnej nazywane są **bitami**.

Przykładem liczby zapisanej w systemie dwójkowym, może być liczba $10011_{(2)}$. Liczba w indeksie dolnym wskazuje, w jakim systemie została zapisana liczba. Żeby odczytać wartość takiej liczby, należy **pomnożyć** wartość poszczególnych cyfr, przez kolejne potęgi dwójki:

$$10011_{(2)} = 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 16 + 2 + 1 = 19_{(10)}$$

Praca z liczbami binarnymi jest o wiele szybsza i łatwiejsza, gdy pamięta się kolejne potęgi dwójki. Dla ułatwienia, pierwsze 13 potęg dwójki przedstawiono w tabeli poniżej.

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^n$ | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 |

# ZADANIE 1

Podaj wartość (w systemie dziesiętnym) następujących liczb binarnych:

a) $100_{(2)}$
b) $101010_{(2)}$
c) $11001110_{(2)}$

## ZAMIANA LICZB W SYSTEMIE DZIESIĘTNYM NA LICZBY BINARNE

Istnieje metoda pozwalająca zamienić liczbę w systemie dziesiętnym, na liczbę binarną. Metoda polega na dzieleniu liczby przez 2 i zapisywaniu reszty z dzielenia. Metodę należy następnie powtarzać na ilorazie, dopóki iloraz będzie różny od 1. Potem należy metodę powtórzyć dla jedynki, a później odczytać wynik **od końca**.

Przykładowo, aby zamienić $19_{(10)}$ na liczbę w systemie dwójkowym, należy postępować w następujący sposób:

| Liczba | /2 | Reszta |
|---|---|---|
| 19 | 9 | 1 |
| 9 | 4 | 1 |
| 4 | 2 | 0 |
| 2 | 1 | 0 |
| 1 | 1 | 1 |

Stąd, liczbie $19_{(10)}$, odpowiada liczba 10011 w systemie dwójkowym.

# ZADANIE 2

Podaj wartość (w systemie dwójkowym) następujących liczb:

a) $66_{(10)}$
b) $112_{(10)}$
c) $234_{(10)}$

## OPERACJE NA BITACH

Jako, że poszczególne bity przyjmują wyłącznie wartość zero lub jeden, na liczbach binarnych można wykonywać operacje jak na zdaniach logiki matematycznej. W wypadku liczb, wartość „prawda" można utożsamić z wartością „1", a „fałsz" z „0".

Najbardziej podstawową operacją jest negacja binarna. Oznaczana jest ona przez znak tyldy („~", znak ten powinien znajdować się na lewo od „1", żeby go wpisać, należy trzymać wciśnięty klawisz SHIFT) lub wykrzyknika. Tabelę wartości przedstawiono poniżej:

| X | ~X |
|---|---|
| 0 | 1 |
| 1 | 0 |

Negację stosuje się oddzielnie dla każdego bitu. Dla przykładu, negacją liczby $100101_{(2)}$, będzie liczba $011010_{(2)}$.

Istnieją także funktory dwuargumentowe, z których najbardziej podstawowymi jest koniunkcja binarna („i", oznaczana przez „&") oraz alternatywa binarna („lub", oznaczana przez „ | " – SHIFT+\). Tabele wartości przedstawiono poniżej:

| X | Y | X&Y | X\|Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Dla przykładu weźmy liczby $1010_{(2)}$ i $10011_{(2)}$. Należy zauważyć, że posiadają one różną liczbę bitów. Żeby wykonać na nich operacje koniunkcji lub alternatywy, należy „krótszą" z wartości **uzupełnić zerami od najbardziej znaczącego bitu**. W naszym wypadku, „krótsza" jest wartość $1010_{(2)}$, którą uzupełnimy, dodając jedno „0" z przodu. W efekcie otrzymamy wartość $01010_{(2)}$.

Następnie, na wartościach $01010_{(2)}$ i $10011_{(2)}$ , możemy wykonać koniunkcję ($01010_{(2)}$ & $10011_{(2)} = 00010_{(2)} = 2_{(10)}$) i alternatywę ($01010_{(2)} \mid 10011_{(2)} = 11011_{(2)} = 27_{(10)}$).

Często stosowaną w automatyce operacją jest alternatywa wykluczająca (ang.: „exclusive or", **XOR**, po polsku „albo"), którą oznacza się znakiem „^" (SHIFT+6). Tabela wartości XOR jest podobna jak zwykłej alternatywy, z tą różnicą, że dla pary „1" i „1", zwraca ona wartość „0" (stąd „wykluczająca" w nazwie). Tabelę przedstawiono poniżej:

| X | Y | X^Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# ZADANIE 3

Podaj wynik (w systemie dwójkowym) następujących operacji:

a) $\sim 10011010_{(2)}$
b) $10011_{(2)}$ & $10110_{(2)}$
c) $100000_{(2)}$ | $1011_{(2)}$
d) $10010_{(2)}$ ^ $101110_{(2)}$

## DODAWANIE LICZB BINARNYCH

Liczby binarne można do siebie również dodawać analogicznie, jak robi się to z liczbami w systemie dziesiętnym. Dodając do siebie $0_{(2)}$ i $0_{(2)}$, otrzymamy $0_{(2)}$. Dodając $0_{(2)}$ do $1_{(2)}$ (lub $1_{(2)}$ do $0_{(2)}$), otrzymamy $1_{(2)}$. Kiedy dodamy $1_{(2)}$ do $1_{(2)}$, otrzymamy $10_{(2)}$. Na pierwszy rzut oka, może się to wydawać dziwne, ale podobnie dzieje się przecież w systemie dziesiętnym. Dodając $5_{(10)}$ do $5_{(10)}$, otrzymamy $10_{(10)}$. Nie możemy bowiem zapisać wyniku na jednej cyfrze, dlatego dodajemy kolejną.

Spróbujmy dodać do siebie liczby $1100_{(2)}$ i $10101_{(2)}$. Żeby tego dokonać, dodajemy po jednym bicie, rozpoczynając od najmniej znaczącego:

| | | | |
|---|---|---|---|
| | | $1100_{(2)}$ | |
| | + | $10101_{(2)}$ | |
| KROK 1 | | $1_{(2)}$ | $0_{(2)} + 1_{(2)} = 1_{(2)}$ |
| KROK 2 | | $01_{(2)}$ | $0_{(2)} + 0_{(2)} = 0_{(2)}$ |
| KROK 3 | | $001_{(2)}$ | $1_{(2)} + 1_{(2)} = 0_{(2)}$, $1_{(2)}$ w pamięci. |
| KROK 4 | | $0001_{(2)}$ | $1_{(2)} + 0_{(2)} + 1_{(2)}(z\ pamięci) = 0_{(2)}$, $1_{(2)}$ w pamięci. |
| KROK 5 | | $00001_{(2)}$ | $0_{(2)} + 1_{(2)} + 1_{(2)}(z\ pamięci) = 0_{(2)}$, $1_{(2)}$ w pamięci. |
| KROK 6 | | $100001_{(2)}$ | |
| | = | $100001_{(2)}$ | |

Skąd wiadomo, że dodaliśmy liczby poprawnie? Wystarczy zamienić je na system dziesiętny. $1100_{(2)} = 12_{(10)}$, $10101_{(2)} = 21_{(10)}$ i $100001_{(2)} = 33_{(10)}$.

# ZADANIE 4

Podaj wynik (w systemie dwójkowym) następujących operacji:

a) $1000_{(2)} + 100_{(2)}$
b) $10001_{(2)} + 1001_{(2)}$

## LICZBY UJEMNE I KOD UZUPEŁNIEŃ DO DWÓCH

Jak na razie potrafimy zapisać wyłącznie liczby dodatnie. Co jednak w momencie, gdy chcemy liczby od siebie odjąć lub zapisać liczbę ujemną za pomocą systemu dwójkowego?

Z pomocą przychodzi kod uzupełnień do dwóch. Jest to system zapisu liczb binarnych, w którym najbardziej znaczący bit przechowuje informacje o znaku liczby. Jeżeli wartość najbardziej znaczącego bitu wynosi „0", to liczba jest dodatnia. Jeżeli wartość najbardziej znaczącego bitu wynosi „1", to liczba jest ujemna.

Żeby odróżnić liczby zapisane za pomocą kodu uzupełnień do dwóch, od zwykłych liczb binarnych, w indeksie dolnym, zamiast (2), umieszcza się „U2" (np. $010011_{U2}$).

Liczby zapisane za pomocą kodu uzupełnień do dwóch, mają bardzo interesujące właściwości. Po pierwsze, wartość każdej liczby binarnej, którą do tej pory zapisaliśmy w tej instrukcji, jest **identyczna** w kodzie uzupełnień do dwóch. Wystarczy dopisać „0" na najbardziej znaczącym bicie, co nie zmieni wartości samej liczby.

Żeby utworzyć liczbę ujemną w kodzie uzupełnień do dwóch, należy na liczbie przeciwnej, wykonać operacje negacji binarnej i dodać 1. Dla przykładu, żeby otrzymać zapis liczby -42 w U2, należy wykonać następujące operacje:

$$-42_{(10)} = \sim\left(42_{(10)} \to U2\right) + 1 = \sim 0101010_{U2} + 1 = 1010101_{U2} + 1 = 1010110_{U2}$$

Gdy potrafimy zapisać liczby ujemne, możemy zająć się odejmowaniem liczb binarnych. W matematyce, operację odejmowania można zastąpić dodawaniem liczby przeciwnej. Na przykład, $60 - 42 = 60 + (-42) = 18$.

Żeby odjąć dwie liczby binarne, należy odjemnik sprowadzić do liczby przeciwnej zapisanej w systemie U2, a następnie dodać odjemną i odjemnik oraz odrzucić najbardziej istotny bit, którego wartość wynika z operacji na bitach znaków i przepełnienia. Przykłady:

$$60_{(10)} - 42_{(10)} = 60_{(10)} + \left(-42_{(10)}\right) = 0111100_{U2} + 1010110_{U2} = \cancel{1}0010010 = 010010_{U2} = 18_{(10)}$$

$$42_{(10)} - 60_{(10)} = 42_{(10)} + \left(-60_{(10)}\right) = 0101010_{U2} + 1000100_{U2} = \cancel{1}101110 = 101110_{U2} = -18_{(10)}$$

# ZADANIE 5

Podaj wynik (w kodzie uzupełnień do dwóch) następujących liczb i operacji:

a) $91_{(10)}$
b) $-73_{(10)}$
c) $1000_{(2)} - 100_{(2)}$ Ⓟ
d) $11001_{(2)} - 1001110_{(2)}$ Ⓟ

## IEEE 754

Po zapoznaniu się z kodem uzupełnień do dwóch, możliwe staje się zapisanie całego zbioru liczb całkowitych, w postaci binarnej. Żaden z poznanych dotychczas sposobów zapisu, nie pozwala pracować z liczbami zmiennoprzecinkowymi[2].

[2] Liczby zmiennoprzecinkowe to liczby rzeczywiste, zapisane za pomocą notacji naukowej.

Żeby zaradzić temu problemowi, organizacja IEEE (Instytut Inżynierów Elektryków i Elektroników), zaproponowało standard IEEE 754. Każda liczba zapisana w tym standardzie, składa się z 32 bitów, a każdy z bitów ma jasno przypisane znaczenie.

| Znak | Wykładnik | | | | | | | | Mantysa | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

Wartość liczby jest określona według wzoru: $(-1)^{Znak} * (1.Mantysa) * 2^{(Wykładnik-127)}$. Tak, jak w przypadku liczb zapisanych w kodzie uzupełnień do dwóch, pierwszy bit przechowuje znak liczby. 0 oznacza liczbę dodatnią, a 1 liczbę ujemną.

Wykładnik jest przechowywany na 8 bitach, za pomocą zwykłego kodowania. Przy przeliczaniu wartości liczby, odejmujemy od wykładnika $127_{(10)}$.

Mantysa jest przechowywana na 23 bitach a jej bity reprezentują **część ułamkową**, co znaczy, że wartości poszczególnych bitów są **odwrócone**. Przy interpretowaniu mantysy, **przyjmuje się, że jej długość wynosi 24 bity, ale najistotniejszy bit jest ukryty**. Znaczy to, że kolejne (licząc od najbardziej znaczącego) bity mantysy mają wartość $2^{-1}$, $2^{-2}$, $2^{-3}$, i tak dalej. Przykładowo, jeżeli mantysa przyjmie wartość $101\ 0000\ 0000\ 0000\ 0000\ 0000$ (czyli, licząc „ukryty bit”: $0101\ 0000\ 0000\ 0000\ 0000\ 0000$), oznacza to, że jej wartość wynosi $2^{-1} + 2^{-3} = 0.625_{(10)}$.

Połączmy te informacje w całość i spróbujmy zinterpretować konkretną liczbę:

$$1100\ 0011\ 0011\ 0000\ 0000\ 0000\ 0000\ 0000_{IEEE\ 754}$$

Zacznijmy od podzielenia bitów na części o określonym znaczeniu:

| Znak | Wykładnik | | | | | | | | Mantysa | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Teraz podstawmy wartości do wzoru:

$$(-1)^{Znak} * (1.Mantysa) * 2^{(Wykładnik-127)} =$$

$$= (-1)^1 * \left(1.011_{(2)}\right) * 2_{(10)}^{10000110_{(2)}-127_{(10)}} =$$

$$= -1 * (1 + 2^{-2} + 2^{-3})_{(10)} * 2_{(10)}^{(134-127)_{(10)}} =$$

$$\left(-1 * 1.375\right)_{(10)} * \left(2^7\right)_{(10)} =$$

$$(-1.375 * 128)_{(10)} = \underline{\underline{-176_{(10)}}}$$

# ZADANIE 6 Ⓟ (NAGRADZANE DWOMA PLUSAMI)

Oblicz wartość dziesiętną liczby $0100\ 0001\ 1010\ 0110\ 0000\ 0000\ 0000\ 0000_{IEEE\ 754}$.

| | |
|---|---|
| Autor: | Mgr inż. Paweł Stawarz, 29.09.2021 |
| Korekta: | - |