

Architektura systemów komputerowych

Laboratorium 4

Operacje arytmetyczne w systemie dwójkowym

Marcin Stępnia

1 Informacje

Podstawowe operacje arytmetyczne w systemie binarnym wykonuje się w analogiczny sposób jak w systemie dziesiętnym.

1.1 Dodawanie liczb binarnych

Dodawanie dwóch jedynek w systemie binarnym powoduje przeniesienie jedynki na pozycję wyższą.

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 1 \\ \hline 1 \text{ } 0 \end{array}$$

Przykłady:

$$\begin{array}{r} 11111,010 \\ + 00001,000 \\ \hline 100000,010 \end{array} \quad \begin{array}{r} 10011,010 \\ + 00001,011 \\ \hline 10100,101 \end{array}$$

1.2 Odejmowanie liczb binarnych

Gdy wynik odejmowania dałby liczbę ujemną, to należy dokonać "pożyczki" z pozycji wyższej. W takim przypadku na pozycji wyższej odejmujemy jeden, a do pozycji niższej przenosimy $10_{(2)}$ (podstawę systemu, czyli 2 dla systemu binarnego).

$$\begin{array}{r}
 \begin{array}{r}
 0 \\
 - 0 \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{r}
 (-1) \quad (10) \\
 \rightarrow \quad \downarrow \\
 0 \\
 - 1 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 - 0 \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 - 1 \\
 \hline
 0
 \end{array}
 \end{array}$$

Przykłady:

$$\begin{array}{r}
 11111,010 \\
 - 00001,000 \\
 \hline
 11110,010
 \end{array}
 \quad
 \begin{array}{r}
 10,010100 \\
 - 01,101001 \\
 \hline
 00,101011
 \end{array}$$

1.3 Mnożenie liczb binarnych

Każdą cyfrę mnożnej mnożymy przez poszczególne cyfry mnożnika zapisując wyniki mnożeń w odpowiednich kolumnach. Działania są nawet prostsze niż w systemie dziesiętnym ponieważ wynik mnożenia jest zawsze jednocyfrowy. Na końcu dodajemy do siebie wszystkie cyfry w kolumnach. Liczby z przecinkiem mnożymy z pominięciem przecinków, a w wyniku przecinek wstawiamy, licząc od prawej strony, na pozycji, która jest sumą pozycji ułamkowych mnożnika i mnożnej.

$$\begin{array}{r}
 0 \quad 0 \quad 1 \quad 1 \\
 \cdot 0 \quad \cdot 1 \quad \cdot 0 \quad \cdot 1 \\
 \hline
 0 \quad 0 \quad 0 \quad 1
 \end{array}$$

Przykłady:

$$\begin{array}{r}
 \begin{array}{r}
 1 \ 0, \ 1 \ 0 \\
 \cdot \ 0 \ 1, \ 0 \ 1 \\
 \hline
 1 \ 0 \ 1 \ 0 \\
 0 \ 0 \ 0 \ 0 \\
 1 \ 0 \ 1 \ 0 \\
 + \ 0 \ 0 \ 0 \ 0 \\
 \hline
 0 \ 1 \ 1, \ 0 \ 0 \ 1 \ 0
 \end{array}
 \quad
 \begin{array}{r}
 1, \ 1 \ 1 \ 1 \\
 \cdot \ 1 \ 1 \ 0, \ 1 \\
 \hline
 1 \ 1 \ 1 \ 1 \\
 0 \ 0 \ 0 \ 0 \\
 1 \ 1 \ 1 \ 1 \\
 + \ 1 \ 1 \ 1 \ 1 \\
 \hline
 1 \ 1 \ 0 \ 0, \ 0 \ 0 \ 1 \ 1
 \end{array}
 \end{array}$$

1.4 Dzielenie liczb binarnych

Przy dzieleniu liczb binarnych należy brać pod uwagę brak możliwości dzielenia przez zero.

$$\begin{array}{r}
0 \\
: 0 \\
\hline
error
\end{array}
\quad
\begin{array}{r}
0 \\
: 1 \\
\hline
0
\end{array}
\quad
\begin{array}{r}
1 \\
: 0 \\
\hline
error
\end{array}
\quad
\begin{array}{r}
1 \\
: 1 \\
\hline
1
\end{array}$$

Dzielenie w naturalnym systemie dwójkowym jest podobne do dzielenia w systemie dziesiętnym. Dzielimy dzielną przez dzielnik i wynik zapisujemy nad kreską. Następnie mnożymy otrzymany wynik przez dzielnik i dokonujemy odejmowania pod kreską. Obrazują to poniższe przykłady.

Przykłady:

$$\begin{array}{r}
1 \ 0 \ 1 \ 1 \\
\hline
1 \ 0 \ 1 \ 1 \ 0 \quad : 100 \\
- 1 \ 0 \ 0 \\
\hline
1 \ 1 \ 0 \\
- 1 \ 0 \ 0 \\
\hline
1 \ 0 \ 0 \\
- 1 \ 0 \ 0 \\
\hline
0 \ 0 \ 0
\end{array}$$

$$\begin{array}{r}
1 \ 0 \ 1 \\
\hline
1 \ 0 \ 0 \ 0 \ 1 \ 1 \quad : 111 \\
- 1 \ 1 \ 1 \\
\hline
1 \ 1 \ 1 \\
- 1 \ 1 \ 1 \\
\hline
0 \ 0 \ 0
\end{array}$$

1.5 Działania na liczbach w systemie U2

Liczby U2 dodajemy i odejmujemy wg poznanych zasad dla naturalnego systemu dwójkowego. Przeniesienia poza bit znaku i pożyczki do tej pozycji ignorujemy.

W trakcie wykonywania działań arytmetycznych wynik operacji może przekroczyć dozwolony zakres liczb zarówno powyżej górnej granicy (nadmiar) jak i poniżej dolnej (niedomiar). W takiej sytuacji następuje zmiana znaku, gdy w rzeczywistości nie powinna ona nastąpić. Założmy, iż operujemy na 4 bitowych liczbach w kodzie U2 i chcemy wykonać proste dodawanie:

$$\begin{array}{r}
0 \ 1 \ 1 \ 1 \\
+ 0 \ 0 \ 0 \ 1 \\
\hline
1 \ 0 \ 0 \ 0
\end{array}
\quad
\begin{array}{r}
7 \\
+ 1 \\
\hline
-8
\end{array}$$

Po sumowaniu 7 i 1 otrzymujemy liczbę w kodzie U2, której wartość jest równa -8, co nie jest poprawnym wynikiem operacji sumowania. Spowodowane to jest tym, iż liczba 8, będąca sumą 7 i 1, wykracza poza górny zakres wartości 4 bitowego kodu U2, którym jest 7. Aby poprawnie przedstawić liczbę 8 należy przeznaczyć na zapis liczby więcej bitów. Podobną sytuację zastaniemy przy próbie dodania dwóch liczb ujemnych, np. -6 i -3:

$$\begin{array}{r}
 1\ 0\ 1\ 0 \\
 +\ 1\ 1\ 0\ 1 \\
 \hline
 1\ 0\ 1\ 1\ 1
 \end{array}
 \qquad
 \begin{array}{r}
 -6 \\
 +\ -3 \\
 \hline
 7
 \end{array}$$

W tym przypadku otrzymujemy prawidłowy wynik na 5 bitach, ale przeniesioną „1” poza bit znaku musimy odrzucić, a tak otrzymana liczba w systemie U2 ma wartość 7. Problemem w tym przypadku jest to, że liczba -9 jest mniejsza od dolnego krańca 4-bitowych liczb w kodzie U2 (równego -8) i z tego powodu nie może być poprawnie przedstawiona w tym kodzie.

Tabela 1: Reguły powstawania nadmiaru/niedomiaru w kodzie U2

Operacja	Pierwszy argument	Drugi argument	Wynik
Dodawanie	0	0	1
	1	1	0
Odejmowanie	0	1	1
	1	0	0

Tabela 1 zawiera reguły powstawania nadmiaru i niedomiaru przy operacjach dodawania i odejmowania na liczbach w kodzie U2. Przedstawione „0” i „1” to stany bitów znakowych.

Dla operacji mnożenia i dzielenia liczb w systemie U2 istnieje wiele różnych algorytmów. Najprostszym sposobem (niestety nie najszybszym) jest zamiana liczb, na których działamy, na liczby dodatnie. Jeżeli obie liczby są dodatnie lub ujemne, to wynik pozostawiamy bez zmian. W przypadku, gdy mnożymy lub dzielimy liczby o różnych znakach, to po zamianie liczby ujemnej na dodatnią i wykonaniu wybranej operacji należy wynik zamienić na liczbę ujemną.

Więcej informacji można znaleźć:

1. w książce Barczak A., Florek J., Sydoruk T.: Elektroniczne techniki cyfrowe, Wyd.: VIZJA PRESS&IT Sp. z o.o, Warszawa 2006 (Rozdział 1 dostępny pod adresem: <http://mirek.ii.uph.edu.pl/ask/files/rozd1.pdf>)
2. na stronie <https://pl.wikipedia.org/wiki/Dodawanie>
3. na stronie <https://pl.wikipedia.org/wiki/Odejmnowanie>
4. na stronie <https://pl.wikipedia.org/wiki/Mno%C5%BCenie>
5. na stronie <https://pl.wikipedia.org/wiki/Dzielenie>
6. na stronie http://www.medianauka.pl/dzielenie_pisemne

2 Zadania

2.1 Działania na liczbach bez znaku

2.1.1 Dodać następujące liczby binarne:

1. $1010101 + 10101$
2. $1110011 + 110$
3. $101,0101 + 1010,11$
4. $11001 + 11010101010$

2.1.2 Odjąć następujące liczby binarne:

1. $1110011 - 110$
2. $10101,01 - 10,1011$
3. $11001011 - 1100111$
4. $11110000 - 11$

2.1.3 Pomnożyć następujące liczby binarne:

1. $1110011 \cdot 101$
2. $101,0110 \cdot 11,0$
3. $101011101101010 \cdot 10$
4. $110010001011,10101 \cdot 100$

2.1.4 Podzielić następujące liczby binarne:

1. $10101 : 111$
2. $110101,001 : 101$
3. $101011101010 : 10$
4. $1100100011,10101 : 100$
5. $11110,1 : 100$

2.2 Działania na liczbach ze znakiem

2.2.1 Wykonać działania w zapisie U2:

1. $7-8$
2. $-9-9$

3. $-27+12$

4. $-17-1$

5. $56-64$

6. $128+5$

Poprawność wykonanych obliczeń można sprawdzić w widoku programisty kalkulatora systemowego. Można wspomagać się konwerterem pod adresem:

<http://baseconvert.com/>