

Лекція № 5 з навчальної дисципліни

“Архітектура обчислювальних систем”.

Розділ 1. Організація апаратної частини комп'ютерів

Тема лекції:

Алгоритми виконання арифметичних операцій над числами з фіксованою та рухомою комою

План лекції

1. *Натуральний, обернений, прямий і доповняльний коди двійкових чисел.*
2. *Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою.*
3. *Особливості виконання арифметичних операцій із рухомою комою .*

1. **Натуральний, прямий, обернений і доповняльний коди двійкових чисел**

Натуральним кодом називають подання числа як цілого беззнакового у двійковій системі числення. Діапазон подання чисел у натуральному кодi для n-розрядної сітки становить від 0 до $2^n - 1$.

Для подання цілих знакових чисел використовують прямий, обернений і доповняльний коди. **Старший розряд сітки є знаковим. Значення цього розряду дорівнює 0 для додатних чисел і 1 – для від'ємних.** В інших розрядах розміщується модуль числа.

Якщо до натурального коду цілого числа додати знаковий розряд, то одержуємо запис числа у **прямому кодi**. Домовимося знаковий розряд розташовувати зліва і відокремлювати від розрядів модуля числа точкою, наприклад: $+6(10) = 0.110$ (ПК); $-6(10) = 1.110$ (ПК).

Використання ПК забезпечує виконання операції додавання двох додатних чисел звичайним способом – не варто лише робити перенос одиниці старшого розряду модуля суми у знаковий розряд, тобто при виконанні арифметичних операцій над ПК двійкових чисел знаковий розряд і розряди модуля не можна розглядати як єдине ціле.

	<u>Правильно:</u>	<u>Неправильно:</u>
<u>Приклад:</u>	0.0110	0.0110
	$+ 0.1010$	$+ 0.1010$
	0.10000	1.0000
	$(+6) + (10) = (+16)$	$(+6) + (+10) = (-0)$

1. *Натуральний, прямий, обернений і доповняльний коди двійкових чисел*

Однак, виконання операції віднімання одного числа від іншого шляхом безпосереднього додавання їхніх ПК неможливо. Також в ПК нуль має два можливі зображення: $-0 = 1.000\dots$ і $+0 = 0.000\dots$, що ускладнює інтерпретацію результатів виконання арифметичних операцій у ЕОМ.

Іншою формою запису двійкових чисел є **обернений код**. ОК двійкового від'ємного числа утворюється з ПК, рівного йому за модулем додатнього числа, шляхом інвертування значень усіх його розрядів. Або: ОК від'ємного числа утворюється шляхом інверсії всіх розрядів модуля цього числа, записаного у ПК. Знаковий розряд при цьому зберігає значення 1.

Наприклад, $-6 (10) = 1.110 (ПК) = 1.001 (ОК)$.

При виконанні арифметичних операцій над двійковими числами, поданими в ОК, знаковий розряд і розряд модуля числа можна розглядати як єдине ціле (перенос одиниці зі старшого розряду модуля суми в знаковий розряд не приводить до помилкового результату), але нуль, як і раніше, має два зображення – «додатне» і «від'ємне». Слід зазначити, що отриманий при додаванні від'ємний результат також утворюється в ОК. У цьому випадку число може бути перетворене у ПК інверсією всіх значущих розрядів (розрядів модуля).

$$\begin{array}{r} \mathbf{0.110} \\ \mathbf{+1.001} \\ \hline \mathbf{1.111}_{(ОК)} = \mathbf{1.000}_{(ПК)} \end{array} \quad \mathbf{(+6) + (-6) = (-0)}$$

1. *Натуральний, прямий, обернений і доповняльний коди двійкових чисел*

Найбільше поширення в обчислювальних пристроях одержало подання від'ємних двійкових чисел за допомогою **доповняльного коду**.

ДК від'ємного числа утворюється з його прямого коду за правилом:

- у знаковому розряді залишається одиниця;
- розряди модуля числа інвертуються;
- до молодшого розряду додається одиниця.

Очевидно, що ДК від'ємного числа утворюється з його ОК додаванням одиниці до молодшого розряду.

Наприклад, $-6(10) = 1.010$ (ДК).

Дійсно, для числа -6 маємо:

$$\begin{array}{r} 1.110_{(ПК)} \\ 1.001_{(ОК)} \\ + \quad 1 \\ \hline 1.010_{(ДК)} \end{array}$$

Зворотний перехід від ДК до ПК або ОК відбувається за тими ж правилами.

Головною перевагою ДК є те, що цифра 0 у ньому має єдине подання: 0.000...

Неправильний дріб (число, що має цілу частину) зі знаком записують у різних кодах за допомогою традиційного роздільника – коми між цілою і дробовою частиною. Наприклад: $-118,375(10) = 1.0001,101$ (ДК).

Слід пам'ятати, що для кодування додатних чисел застосовується тільки ПК, хоча можна сказати, що для таких чисел ДК і ОК збігаються з прямим.

1. *Натуральний, прямий, обернений і доповняльний коди двійкових чисел*

Операція одержання ДК від'ємного числа з ПК, рівного йому за модулем додатнього числа, називається ***операцією доповнення до двох***. Ця операція полягає в інвертуванні всіх розрядів вихідного коду (включаючи знаковий) і додавання до молодшого розряду одиниці.

Таким чином, сформулюємо наступне правило: ***у системі двійкових чисел зі знаком заміна додатнього числа на рівне йому за модулем від'ємне і навпаки, від'ємного на додатне, здійснюється шляхом застосуванням до коду даного числа операції доповнення до двох***.

Ця властивість подання від'ємних чисел у ДК дозволяє при виконанні арифметичних операцій взагалі відмовитися від операції віднімання, замінивши її операцією додавання з числом, що має знак, протилежний знаку числа, яке віднімається.

2. Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою

Додавання двійкових чисел із знаком

Очевидно, що при додаванні чисел із знаком можуть виникати переноси одиниці із старшого розряду модуля суми до знакового розряду (домовимося позначати його **$P1$**) та із знакового розряду – ліворуч за межі розрядної сітки, у розряд переповнення (**$P2$**). Через використання розглянутих раніше кодів, у яких знак числа позначається тими ж цифрами, що і розряди модуля, переповнення розрядної сітки може виникати навіть у випадку додавання чисел із різними знаками, коли модуль результату не перевищує модуля будь-якого операнда. При додаванні ж двох від'ємних чисел перенесення одиниці до розряду переповнення відбувається завжди.

При виникненні переповнення розрядної сітки для одержання правильного результату додавання необхідно застосовувати таке правило:

- якщо $P1 \oplus P2 = 0$, одиниця в розряді переповнення ігнорується (відкидається);***
- якщо $P1 \oplus P2 = 1$, необхідно зсунути число на один розряд праворуч (або зсунути позицію точки на один розряд ліворуч).***

Додавання дробових і цілих двійкових чисел, поданих у формі з фіксованою комою, відбувається однаково, тобто порядок додавання не залежить від розташування коми. Тому операцію додавання розглянемо на прикладі додавання цілих чисел.

2. Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою

Приклади:

- 1) Додавання двох додатних чисел (без переповнення розрядної сітки).
- | | |
|--------------|-------|
| 0.100111 | 39 |
| $+ 0.001101$ | $+13$ |
| <hr/> | |
| 0.110100 | 52 |
- $P1 \oplus P2 = 0$ – результат коректний і остаточний.
- 2) Додавання двох додатних чисел (з переповненням розрядної сітки).
- | | |
|-------------|-------|
| 0.01101 | 13 |
| $+ 0.10011$ | $+19$ |
| <hr/> | |
| 1.00000 | 32 |
- $P1 \oplus P2 = 1$. Результат некоректний, тому що відбулося переповнення розрядної сітки. Зсуваючи число на один розряд праворуч, остаточо маємо 0.100000 (ПК) = 32 (10).
- 3) Додавання двох чисел із різними знаками (без переповнення розрядної сітки).
- | | |
|--------------|--------|
| 1.001100 | - 52 |
| $+ 0.001101$ | $+ 13$ |
| <hr/> | |
| 1.011001 | - 39 |
- $P1 \oplus P2 = 0$. Результат коректний, але тому що він є від'ємним, для перевірки правильності розв'язання необхідно перетворити його у прямий код. Остаточо маємо 1.100111 (ПК) = 39 (10).
- 4) Додавання двох чисел, рівних за модулем і різних за знаком.
- | | |
|--------------|--------|
| 1.011001 | - 39 |
| $+ 0.100111$ | $+ 39$ |
| <hr/> | |
| 10.000000 | 0 |
- $P1 \oplus P2 = 0$. Результат коректний, якщо не брати до уваги одиницю у розряді переповнення.

2. Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою

Висновок

Правильність виконання операцій додавання обов'язково повинна перевірятися шляхом аналізу значення виразу $P1 \oplus P2$, щоб уникнути одержання некоректного результату, що виникає при переповненні розрядної сітки, при цьому: якщо $P1 \oplus P2 = 0$, одиниця в розряді переповнення ігнорується (відкидається); якщо $P1 \oplus P2 = 1$, необхідно зсунути число на один розряд праворуч.

Правило перевірки коректності результату додавання двійкових чисел також можна сформулювати в такий спосіб: якщо знак операндів однаковий, а знак суми протилежний, результат є некоректним. При додаванні двох операндів із різними знаками результат завжди коректний, якщо не брати до уваги одиницю у розряді переповнення.

2. Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою

Приклад:

Множення та ділення двійкових чисел із фіксованою комою

$$\begin{array}{r} 1101 \\ * 1011 \\ \hline 1101 \\ 1101 \\ +0000 \\ 1101 \\ \hline 10001111 \end{array} \quad \begin{array}{r} 13 \\ * 11 \\ 13 \\ + 13 \\ \hline 143 \end{array}$$

Множення двійкових чисел виконують у прямому коді. Знак добутку визначають за знаковими розрядами множників згідно з загальновідомим правилом: якщо знаки операндів однакові, то результат додатний, у протилежному випадку – знак добутку від’ємний. Цій алгоритм реалізується за допомогою суматора за модулем 2. Знак добутку двох чисел не впливає на алгоритм виконання операції множення модулів цих чисел.

Часто використовують спосіб множення, процедура якого аналогічна процедурам множення вручну. У цьому випадку результат одержують додаванням часткових добутків. Кожен частковий добуток удвічі перевищує попередній, що відповідає його зсуванню ліворуч на один розряд.

Характерно, що у загальному випадку розрядність добутку двійкових чисел удвічі перевищує розрядність співмножників. Якщо у множенні беруть участь мантиси, тобто правильні дроби, то молодші розряди, що виходять за межі розрядної сітки, можуть бути відкинуті без округлення або з округленням.

2. Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою

Приклад множення двох чисел з витискуванням другого операнда:

Rg1:	$\begin{array}{r} 1101 \\ \times 1011 \\ \hline 0000 \\ 1101 \\ \hline 1101 \end{array}$	Rg2:	$\begin{array}{r} 1101 \\ \times 1101 \\ \hline 0110 \\ 1001 \\ \hline 10011 \end{array}$	Rg3:	$\begin{array}{r} 1101 \\ \times 1110 \\ \hline 1001 \\ 0100 \\ \hline 1001 \end{array}$	Rg3:	$\begin{array}{r} 1101 \\ \times 1111 \\ \hline 0100 \\ 1000 \\ \hline 10001 \end{array}$	Rg3 : Rg2	$\begin{array}{r} 1101 \\ \times 1111 \\ \hline 1000 \\ 1111 \\ \hline 1000 \end{array}$
	1		2		3		4	Результат	

В кожному такті множення виконуються наступні операції:

1. Отримання чергового часткового добутку та додавання його до вмісту регістра – акумулятора Rg3.
2. Зсування другого операнда (вмісту Rg2) на один (використаний) розряд праворуч. Витиснутий розряд втрачається.
3. Зсування вмісту Rg3 на один розряд праворуч з записом витиснутого розряду в старший розряд Rg2.

Результат множення буде в парі регістрів Rg3: Rg2. При цьому другий співмножник втрачається повністю.

2. Алгоритми виконання арифметичних операцій над двійковими числами із фіксованою комою

Операція ділення також виконується способом, аналогічним застосовуваному при діленні вручну, що ілюструє приклад ділення двох чисел

$506 : 23 = 22$, тобто $0.111111010 : 0.10111 = 0.10110$.

Знак частки визначають аналогічно до знака добутку. Застосоване при діленні віднімання дільника виконують шляхом додавання його додаткового коду.

$$\begin{array}{r}
 0.111111010 \\
 + 1.01001 \\
 \hline
 10.010001010 \\
 0.10001010 \\
 + 1.01001 \\
 \hline
 1.11010010 \\
 1.1010010 \\
 + 0.10111 \\
 \hline
 10.0101110 \\
 0.101110 \\
 + 1.01001 \\
 \hline
 10.000000
 \end{array}$$

– ділене додатне
перше віднімання дільника
1 – результат додатний
зсув ліворуч
– друге віднімання дільника
0 – від’ємний результат
зсув ліворуч
– додавання дільника
1 – результат додатний
зсув ліворуч
– третє віднімання дільника
1 – остача дорівнює нулю

У прикладі використаний **алгоритм без відновлення остачі**:

- із діленого віднімається дільник (додається дільник, записаний у ДК);
- якщо остача додатна, перша цифра частки дорівнює одиниці, у протилежному випадку – 0;
- остача зсувається ліворуч, і до неї додається дільник із знаком, протилежним знаку остачі;
- знак наступної остачі визначає наступну цифру частки;
- ці дії повторюють доти, поки не утвориться необхідне число розрядів частки або нульова остача. Всі розряди проміжних сум, старші за знаковий, слід ігнорувати.

3. Особливості виконання арифметичних операцій із рухомою комою

Операція додавання у пристроях із рухомою комою відбувається у чотири етапи:

- Порівнюються порядки доданків: менший порядок збільшується до більшого. При цьому відповідним чином корегується мантиса числа, яке перетворюється.
- Виконується перетворення мантис у доповняльні коди.
- Виконується додавання мантис за правилами, розглянутими вище для чисел із фіксованою комою.
- До суми приписується порядок доданків i , в разі необхідності, виконується нормалізація результату.

Операція множення чисел, поданих у формі з рухомою комою також виконується у чотири етапи:

- Визначається знак добутку.
- Перемножуються мантиси співмножників за правилами для чисел із фіксованою комою.
- Обчислюється порядок добутку алгебраїчним додаванням порядків співмножників за правилами додавання цілих чисел із знаком.
- Виконується нормалізація отриманого результату у випадку її необхідності.

Ділення чисел у пристроях з рухомою комою виконується так само, як і множення.

3. Особливості виконання арифметичних операцій із рухомою комою

Приклад (додавання чисел з рухомою комою):

$$X=68_{10}=1000100_2; Y=29,125_{10}=11101,001; S=X+Y=97,125_{10}=1100001,001_2$$

1. Якщо розрядність мантиси – 1 байт:

$$M_x=1,0001000; P_x=0110_2=6_{10}; M_y=1,1101001; P_y=0100_2=4_{10}$$

2. Оскільки $X > Y$, порядок числа Y приводиться до порядку числа X (зсувом мантиси M_y на $P_x - P_y = 2$ розряди ліворуч): $M_y=0,0111010; P_y=0110_2=6_{10}$

3. Додавання мантис: $M_s = M_x + M_y =$

$$\begin{array}{r} 1,0001000 \\ \underline{0,0111010} \\ 1,1000010 \end{array}$$

4. Якщо необхідно, виконується операція нормалізації результату – приведення мантиси до стандартного вигляду змінюванням порядку (в даному прикладі результат вже є нормалізованим).

Результат: $M_s=1,1000010; P_s=0110_2=6_{10}; S=1100001,0$

Висновки:

Операція віднімання в обчислювальних пристроях завжди заміняється операцією додавання з числом протилежного знака (доповненням модуля від'ємника).

Операції множення і ділення в обчислювальних пристроях виконуються у вигляді послідовності операцій додавання і зсування.

Таким чином, виконання усіх арифметичних операцій у ЕОМ так чи інакше зводиться до виконання послідовності дій, серед яких основною є додавання чисел, допоміжними – зсування, інверсія та ін. Тому всі ЦА, що виконують арифметичні операції, містять у своєму складі один або декілька суматорів.

Арифметичні операції з рухомою комою, порівняно з операціями з фіксованою комою, складніше, тому виконуються значно повільніше.

Література

1. Мельник А.О. Архітектура комп'ютера. Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – с.56 – 62, 210 – 238.