

Лабораторна робота №3

Тема: ДОСЛІДЖЕННЯ НА МІКРОПРОЦЕСОРНИХ ПРИСТРОЯХ АЛГОРИТМІВ АРИФМЕТИЧНИХ ОПЕРАЦІЙ З ОДНОБАЙТОВИМИ І МНОГОБАЙТОВИМИ ЦІЛИМИ ЧИСЛАМИ (МНОЖЕННЯ ТА ДІЛЕННЯ)

Мета роботи: ознайомлення з принципами програмування на мові АСЕМБЛЕР МП 1810 із використанням налагоджених засобів, з дослідженням алгоритмів арифметичних операцій множення та ділення одно та многобайтових цілих чисел. Ознайомитися з принципами введення та виведення арифметичних операцій множення та ділення однобайтових і многобайтових цілих чисел на екран на мові Turbo Assembler.

1 Теоретичні відомості

1.1 Двійкове множення

Двійкове і десяткове множення також, як і двійкове і десяткове додавання або віднімання, виконують за схожими алгоритмами обчислення.

Множення - це швидкий засіб додавання декількох однобайтових чисел.

Наприклад, множення 7 на 5 зводиться до додавання п'ятох однакових однобайтових чисел, кожне з котрих дорівнює семи.

Недолік такого засобу множення - час обчислення добутку залежить від значення множника, чим більше множник, тим більше час обчислення. Однією з ідей зменшення часу обчислення добутку є виняток множень на нуль.

При множенні одного числа на інше одне з чисел називають множене, інше - множником. Множення виконують поразрядно. Часто виникає необхідність переносу до наступного за старшинством розряду. множеного і розряду множника, значимість якого в 10 разів більше значимості розряду, використаного в попередній операції множення, то всі цифри отриманого добутку зсовують вліво на одну позицію (розряд). Для одержання результату, усунуті щодо один одного часткові добутки складаються. Виникаючі при

додаванні переноси повинні бути обчислені при формуванні остаточного результату. Перемножуючи десяткові числа, звичайно "вирішуємо в розумі" виникаючі при цьому проблеми переносу. По завершенні множення множеного на значення молодшого розряду множника утворюється перший частковий добуток. В результаті множення множеного на значення наступного за старшинством розряду множника формується другий частковий добуток. Подібна процедура повторюється з метою одержання всіх необхідних часткових добуток. Оскільки черговий частковий добуток - результат перемножування.

Розглянемо як приклад множення числа 17 на число 12:

17 множене
* 12 множник
34 перший частковий добуток
17 другий частковий добуток
100 перенос
204 результуючий добуток

Оскільки множник складається з двох розрядів, одержуємо два часткових добутки, додавання яких викликає перенос у розряд сотень.

Тепер перейдемо до розгляду двійкового множення, схематично поданого за допомогою наступних правил:

$0 \times 0 = 0$;
 $0 \times 1 = 0$;
 $1 \times 0 = 0$;
 $1 \times 1 = 1$.

Скористаємося цими правилами для обчислення добутку 8-розрядних двійкових еквівалентів десяткових чисел 17 і 12:

00010001 множене 17
00001100 множник 12
00000000 перший частковий добуток
00000000 другий частковий добуток
00010001 третій частковий добуток
00010001 четвертий. частковий добуток
00000000 п'ятий частковий добуток
00000000 шостий частковий добуток
00000000 сьомий частковий добуток

00000000 восьмий частковий добуток
0000000000000000 перенос
0000000011001100 результат 204

Отже в результаті множення отримано вісім часткових добутоків, оскільки множник складається з восьми розрядів. Перші два часткових добутки включають тільки нулі, такі множники - значення першого і другого розрядів двійкового еквівалента числа 12 рівні 0. Третій частковий добуток - копія множеного. Різниця між ними полягає в тому, що копія зсунута щодо множеного на два розряди вліво, оскільки для одержання цього часткового добутку в якості множника використовується значення третього розряду. Четвертий частковий добуток також є копією множеного, зсунутою щодо останнього на три двійкових розряди вліво. Частковий добуток із п'ятого по восьме складається тільки з нулів, тому що відповідні множники, що беруть участь у формуванні добутку, - двійкові нулі. Додавання всіх часткових добутоків у даному прикладі не супроводжується переносом.

Створений простий засіб виконання двійкового множення, що одержав назву множення шляхом зсуву і додавання.

Перерахуємо основні правила цього засобу.

1. Формування першого часткового добутку. Якщо значення молодшого значущого розряду множника дорівнює 1, то і результат є копією множеного.
2. Правило зсуву. При використанні чергового розряду множника для формування часткового добутку провадиться зсув множеного на один розряд уліво.
3. Правило додавання. Щораз, коли значення розряду множника дорівнює 1, до результату необхідно додати множене, розташоване в визначеній правилом зсуву позиції.
4. Визначення результуючого добутку. Шуканий добуток є результатом виконання всіх операцій зсуву і додавання.

Даний спосіб дозволяє робити операції множення не тільки над однобайтовими числами, але і над многобайтовими. показує застосування способу для множення однобайтових чисел із фіксованою комою.

Система команд МП 1810 містить вмонтовані команди множення, що дозволяють значно скоротити час перемноження двох чисел і зменшити код програми.

Команда MUL може множити тільки 8- або 16-бітові значення, але нею можна скористатися і для множення чисел підвищеної точності без знака. Наприклад, за її допомогою можна перемножити два 32-бітових числа. Для цього треба обчислити серію 32-бітових перехресних добутків, а потім скомбінувати з них 64-бітовий результат з використанням такої схеми:

$$\begin{array}{r}
 AB \\
 - \underline{CD} \\
 \hline
 BD \quad (B \times D) \text{ Перший частковий результат} \\
 + AD \quad (A \times D) \\
 + BC \quad (B \times C) \text{ Зсув на одну позицію вліво} \\
 + \underline{AC} \quad (A \times C) \\
 \hline
 MNOD \text{ (кінцевий результат)}.
 \end{array}$$

1.1 Двійкове ділення

Ділення - це операція, обернена множенню. При діленні операцію віднімання повторюють доти, поки зменшуване не стане менш за від'ємник. Число цих повторювань показує, скільки разів від'ємник укладається в зменшуваному.

Процедура ділення декілька складніше процедури множення.

Роздивимося, наприклад, ділення числа 204 на число 12, користуючись десятковою та двійковою арифметикою:

$$\begin{array}{r}
 204 \overline{) 12} \\
 \underline{12} \\
 84 \\
 \underline{84} \\
 0
 \end{array}$$

$$\begin{array}{r}
 1100 \ 1100 \overline{) 1100} \\
 \underline{1100} \\
 0 \ 1 \\
 \underline{0} \\
 0 \ 11 \\
 \underline{0} \\
 0 \ 110 \\
 \underline{0} \\
 0 \ 1100 \\
 \underline{1100} \\
 0
 \end{array}$$

Двійкове ділення починається з аналізу діленого 11001100 і дільника

1100. Відразу ж виявляємо, що дільник 1100 точно вкладає в 1100, і тому записуємо цифру 1 у подане для формування частки поле. Множимо дільник на 1 і віднімаємо результат із 1100. Різниця дорівнює нулю, тобто менше дільника, а тому процес ділення можна продовжувати. Об'єднуємо нуль залишку із значенням наступного розряду діленого, рівним 1. Оскільки число 1100 укладається 0 разів у числі 1, записуємо 0 у частці, а число 1 об'єднуємо з наступною цифрою діленого і т.д., процедура, що описується, продовжується доти, поки ділене не виявляється вичерпаним.

Таким чином, реалізувати операцію ділення на мікропроцесорі (МП) не настільки просто, як операцію множення. Занадто багато зусиль потрібно для з'ясування того, скільки разів дільник укладається у визначеному числі. Процедура двійкового ділення в дійсності проста, тому що кожний біт частки приймає одне з двох можливих значень: 1 або 0. Як у випадку двійкового множення, зручно виявляється використовувати операції зсуву.

Продемонструємо засіб двійкового ділення на розглянутому вище прикладі, попередньо представивши дільник 1100 у додатковому коді. Це дозволить обмежитися двійковим додаванням в усіх випадках, коли потрібно виконати віднімання.

Як і у випадку вище розглянутого так названого довгого двійкового ділення, необхідно визначити скільки разів дільник укладається в числі, утвореному відповідною кількістю старших значущих бітів діленого. Мікропроцесор не може будувати здогадок стосовно того, скільки разів дільник укладається в указаному числі. У дійсності МП починає віднімати дільник із цього числа.

Якщо дільник не буде укладатися в згаданій частині діленого, завжди можна повернути біти, які віднімали, назад діленому. Про те, що дільник не укладається, свідчить поява від'ємного результату віднімання (біт знака дорівнює 1).

Почнемо спробу виконати віднімання перший раз:

01100 1100 ділене

10100 0000 від'ємник число 12

00000 1100 перший результат

↑----- наявність тут нуля означає, що 1-й біт шуканої частки дорівнює 1 частка 1XXXX.

Якщо дільник укладається у відповідну частину діленого, біт знака дорівнює 0. Це значить, що результат ділення є позитивне число. У даному прикладі це так. А тому перший біт шуканого результату (частки) дорівнює 1.

Другий крок процедури ділення. Слід ще раз спробувати виконати віднімання дільника. Але попередньо необхідно зсунути перший результат. Зсув повинний бути таким, щоб при останній операції віднімання був сформований другий біт частки. У результаті зсуву отримуємо:

000001100 перший результат до зсуву

00001100 перший результат після зсуву

Тепер можна виконати другу операцію віднімання:

00001100 зсунутий перший результат

10100000 від'ємник 12

10101100 другий результат

↑----- наявність тут 1 означає, що 2-й біт шуканої частки дорівнює 0, частка: 10XXX.

Результат цього віднімання містить 1 у позиції знака, тобто отримано від'ємне число. Отже, дільник не вкладається у відповідному числі. Тому, насамперед у другу (по старшинству) позицію поля представлення частки варто записати 0. Крім того, оскільки віднімання не відбулося, необхідно повернути всі біти дільника назад першому результату:

10101100 другий результат

01100000 що повертається число 12

00001100 зсунутий перший результат

Тепер настає черга наступного зсуву:

0000100 зсунутий перший результат

0001100 перший результат після другого зсуву

Виконує третю операцію віднімання:

0001100 двічі зсунутий перший результат

1010000 від'ємник число 12

1011100 третій результат

↑----- наявність тут 1 означає, що 3-й біт шуканого дорівнює 0, частка:

100XX.

Третій результат - негативне число, тобто підданий подвійному зсуву перший результат виявився менше дільника, отже, третій біт шуканої частки дорівнює 0. Тому число 12 (дільник) слід повернути зсунутому двічі першому результату:

1011100 третій результат

0110000 дільник 12

0001100 двічі зсунутий перший результат

Виконуємо черговий зсув:

0001100 двічі зсунутий перший результат

001100 перший результат після третього зсуву

Проведемо віднімання

001100

101000

110100 четвертий результат

↑----- наявність тут одиниці означає, що 4-й біт частки дорівнює 0, частка:

1000X.

Повернемо дільник четвертому результату:

110100

011000

001100

Виконуємо четвертий зсув

001100 до зсуву

01100 після зсуву

Проведемо віднімання:

01100 результат після четвертого зсуву

10100 число 12, що віднімається

00000 п'ятий результат

↑----- наявність тут нуля означає, що 5-й битий частки дорівнює 1, частка: 10001.

На цьому виконання процедури закінчується. Таким чином, при діленні числа 11001100 на число 1100 результат дорівнює 10001.

Розглянута процедура двійкового ділення трохи складніша процедури множення. Проте її неважко здійснити на практиці, якщо послідовно виконувати запропоновані операції. Оскільки правила ділення чітко сформульовані, вони можуть бути реалізовані в МП.

У МП 1810 для реалізації процедури ділення використовуються вмонтовані команди. Програма 2.5 показує використання цих команд для ділення 32-бітного числа на 16-бітне число. Числа розташовуються в DX: AX ділене в регістрі або в пам'яті дільника. Результат ділення: у AX частка, у DX залишок.

Програма 2.5

TITLE ДІЛЕННЯ ЦІЛИХ БЕЗЗНАКОВИХ ЧИСЕЛ

TITLE 3 ВИКОРИСТАННЯМ КОМАНД ДІЛЕННЯ МП 1810

; Визначення сегмента стека

SSEG SEGMENT PARA STACK 'STACK'

DB 128 DUP(0)

SSEG ENDS

;

; Визначення сегмента даних

DSEG SEGMENT PARA PUBLIC 'DATA'

DELIM DD 1 DUP (0) ; Ділене, довжина подвійне слово

DELIT DW 1 DUP (0) ; Дільник, довжина слово

REZLT DW 1 DUP (0) ; Результат, довжина слово

OSTAT DW 1 DUP (0) ; Залишок, довжина слово


```

DSEG ENDS
;
; Визначення сегмента коду програми
;
CSEG SEGMENT PARA PUBLIC 'CODE'
ASSUME CS:CSEG,DS:DSEG,SS:SSEG
;
DIVD PROC FAR
mov ax,DSEG ; Визначення адреси сегмента DATA
mov ds,ax ; Пересилка в сегментний регістр DS
Start: lea bx,DELIM ; Занесення адреси діленого в регістр BX
mov ax,[bx]
mov dx,[bx+2]
div DELIT ; Ділення двох чисел (DXAX:DELIT=AX, DX)
mov REZLT,ax ; Зберігання результату
mov OSTAT,dx
jmp Start
DIVD ENDP
CSEG ENDS
        END DIVD

```

2 Порядок виконання роботи

2.1 Напишіть текст програми згідно свого варіанту представлений в Таб.2.1. Якщо $N \geq 10$, $V=N$ (де N - номер списку в журналі, V – номер варіанту), а $10 \leq N \leq 20$ $V=N-10$. A, B, C – вважати в десятковій формі.

2.2 Повторіть повний цикл створення програми для Турбо Асемблера див. Лабораторну роботу 1 рис.1.1.

2.3 Запустіть програму на виконання.

2.4 Проаналізуйте та збережіть результат програми.

Таблиця 2.1

№ варіанту	завдання
1	Написати програму для обчислення $C=A \times B$, де A, B – додатні числа, A та B ввести з клавіатури, C – вивести на екран.
2	Написати програму для обчислення $C=A \times B$, де A, B – можуть бути, як від’ємні, так і додатні, A та B ввести з клавіатури, C – вивести на екран.
3	Написати програму для обчислення $C=A : B$, де A, B – додатні числа, A та B ввести з клавіатури, C – вивести на екран.
4	Написати програму для обчислення $C=A : B$, де A, B – можуть бути, як від’ємні, так і додатні, A та B ввести з клавіатури, C – вивести на екран.
5	Написати програму для обчислення $C=4 \times (A \times B)$, де A, B – додатні числа, A та B ввести з клавіатури, C – вивести на екран.
6	Написати програму для обчислення $C=4 \times (A \times B)$, де A, B – можуть бути, як від’ємні, так і додатні, A та B ввести з клавіатури, C – вивести на екран.
7	Написати програму для обчислення $C=4 \times (A : B)$, де A, B – додатні числа, A та B ввести з клавіатури, C – вивести на екран.
8	Написати програму для обчислення $C=4 \times (A : B)$, де A, B – можуть бути, як від’ємні, так і додатні, A та B ввести з клавіатури, C – вивести на екран.

3 Зміст звіту

3.1 Назва та мета роботи

3.2 Блок-схема алгоритму програми згідно свого варіанту Turbo Assembler

3.3 Текс програми

3.4 Результат виконання програми

3.5 Висновки по роботі

4 Контрольні питання

- 4.1 Які відмінності команд без знакового і знакового множення і ділення даних МП 1810ВМ86?
- 4.2. Які є способи адресації даних?
- 4.3. Які функції виконують регістри AX, BX, CX, DX, SP, BP, DI, SI, IP, F, CS, DS, SS, ES ?
- 4.4 Як відбувається множення та ділення в двійковій системі числення?
- 4.5 Які вмонтовані команди використовуються у МП 1810 для реалізації процедури ділення та множення?