

Лабораторная работа №3

Арифметические команды центрального процессора

Цель работы: изучение арифметических команд центрального процессора для работы с целыми числами.

Теоретические сведения

Арифметико-логическое устройство центрального процессора содержит следующие основные команды для работы с целочисленной арифметикой:

ADD <операнд_1>, <операнд_2>. Команда складывает два числа, помещённых в регистры или память. Результат записывается на место первого операнда: **операнд_1 = операнд_1 + операнд_2**.

ADC <операнд_1>, <операнд_2>. Сложение чисел с учётом знака переноса. Команда складывает значения двух операндов со значением флага переноса **CF**.

XADD <операнд_1>, <операнд_2>. Команда сначала обменивает содержимое двух операндов, а затем складывает их и пересылает результат на место первого операнда.

INC <операнд>. Команда выполняет инкремент содержимого регистра или ячейки памяти.

SUB <операнд_1>, <операнд_2>. Команда выполняет вычитание значения второго операнда из первого и записывает результат на место первого операнда.

SBB <операнд_1>, <операнд_2>. Команда вычитает из значения первого операнда значение второго операнда и флага CF: **операнд_1 = операнд_1 – (операнд_2 + CF)**.

DEC <операнд>. Выполняет уменьшение на единицу значения регистра или ячейки памяти.

MUL <операнд>. Команда выполняет умножение содержимого регистра **AL/AX/EAX** на значение операнда **без учёта знака**. Местоположение результата зависит размерности операнда и представлено в следующей таблице:

Размер операндов	Первый множитель	Результат	Пример
байт	AL	AX	MUL BL
слово	AX	DX:AX	MUL CX
двойное слово	EAX	EDX:EAX	MUL ESI

IMUL. Команда выполняет умножение чисел **с учётом знака** и имеет три формы:

1. **IMUL** <операнд_множитель>. Механизм работы данной команды похож на работу команды **MUL** с тем отличием, что произведение выполняется с учётом знака. Местоположение множителей и результата смотреть в вышеприведённой таблице.

2. **IMUL** <операнд_1>, <операнд_2>. Команда выполняет умножение значения первого операнда на значение второго. Результат записывается на место первого: **операнд_1 = операнд_1 * операнд_2**.
3. **IMUL** <операнд_1>, <операнд_2>, <операнд_3>. Команда выполняет умножение второго и третьего операндов: **операнд_1 = операнд_2 * операнд_3**. Результат записывает на место первого операнда. Третий операнд может иметь только непосредственную адресацию.

DIV <операнд>. Команда выполняет деление без учёта знаков. Делимое задаётся неявно. Местоположение делимого и результат зависят от размерности операнда-делителя и определяется следующей таблицей:

Размер операнда	Делимое	Частное	Остаток	Максимальное частное
байт	AX	AL	AH	255
слово	DX:AX	AX	DX	65535
двойное слово	EDX:EAX	EAX	EDX	$2^{32}-1$

IDIV <операнд>. Команда выполняет деление с учётом знаков. Делимое задаётся неявно. Механизм работы данной команды похож на механизм работы команды **DIV**.

Команды изменения размерности и знака числа

Следующая группа команд расширяет число в два раза, сохраняя при этом его знак. Знак сохраняется за счёт копирования старшего (знакового) бита числа в старшую половину результирующего числа.

CBW. Команда расширяет байт до размерности слова, копируя старший бит регистра **AL** во все биты регистра **AH**.

CWD. Команда расширяет слово до размерности двойного слова, копируя старший бит регистра **AX** во все биты регистра **DX**.

CWDE. Команда расширяет слово до размерности двойного слова, копируя старший бит регистра **AX** во все биты старшей половины регистра **EAX**.

CDQ. Команда расширяет двойное слово до размерности учетверённого слова, копируя старший бит регистра **EAX** во все биты регистра **EDX**.

NEG <операнд>. Команда изменяет знак числа.

Задания для выполнения к работе

1. Написать программу для вычисления значения арифметического выражения согласно варианту задания. Все переменные, используемые в программе, требуется использовать как знаковые и расширять до размерности двойного слова. Результат должен быть записан в регистр **EAX**. Если результат содержит остаток от деления, оставить его в регистре **EDX**. Подобрать набор тестовых данных (не менее 3).
2. Написать программу для сложения или вычитания целых беззнаковых чисел большой размерности (размерность и операция зависят от варианта задания). Младшие байты при этом хранить по младшему адресу. Подобрать наборы тестовых данных (не менее 3). Для выполнения этого задания изучить теоретический материал главы «Вычитание и сложение операндов большой размерности», начиная со страницы 176 учебника Юрова «Assembler».

Пример выполнения первого задания:

#	$\frac{g^2 + \frac{r}{5} - 9^4}{h}$	g – word r – byte h – word	сложение 7 байт
---	-------------------------------------	--	------------------------

Переменные g , r , h разместим в сегменте данных. Программа, вычисляющая значение этого выражения, имеет вид:

```
.386
.model flat, stdcall
option casemap: none

include d:\masm32\include\kernel32.inc
includelib d:\masm32\lib\kernel32.lib

.data
    g dw -102
    r db -1
    h dw 15
.code
start:
    MOV AX, g          ; AX = g
    CWDE               ; Расширение в регистре AX слова до двойного в EAX
    IMUL EAX            ; EAX = EAX * EAX = g^2
    MOV EBX, EAX        ; EBX = EAX
    MOV AL, r           ; AL = r
    CBW                ; Расширение до слова
    CWDE               ; Расширение до двойного слова
    MOV ECX, 5          ; ECX = 5
    CDQ                ; Расширение перед делением до двойного слова, т.к.
                       ; следующая команда оперирует содержимым EDX.
    IDIV ECX            ; EAX = EDX:EAX / 5 = r / 5
    ADD EAX, EBX        ; EAX = EAX + EBX = r / 5 + g^2
    SUB EAX, 9*9*9*9    ; EAX = EAX - 9^4 = r / 5 + g^2 - 9^4
    MOV EBX, EAX        ; Копирование в EBX
    MOV AX, h           ; AX = h
    CWDE               ; Расширение h до двойного слова
    XCHG EAX, EBX       ; Обмен местами содержимого EAX и EBX
    CDQ                ; Расширение до двойного слова
    IDIV EBX            ; EAX = EDX:EAX / EBX = (r / 5 + g^2 - 9^4) / h

    push 0
    call ExitProcess ; Выход из программы
end start
```

Тестовые данные

g	r	h	Частное (EAX)	Остаток (EDX)
-1000	50	100	000026CEh= 9934	00000031h= 49
10	-100	4	FFFFFF9ACh в дополни- тельном коде соответствует -1620	FFFFFFFFFh в дополни- тельном коде соответствует -1

300	60	-1000	00000053h= 83	000001C3h= 451
-----	----	-------	------------------	-------------------

При выполнении второго задания числа требуется хранить в виде последовательности байт следующим образом:

```
.386
.model flat, stdcall
option casemap: none

include d:\masm32\include\kernel32.inc
includelib d:\masm32\lib\kernel32.lib

.data
a db 2Ah, 03h, 12h, 0DE, 43h, 0E2h, 34h ; 7 байт
b db 15h, 0DDh, 34h, 4Bh, 57h, 7Fh, 0CDh ; 7 байт
r db 8 dup(?) ; Для результата резервируется на один байт больше
.code
start:
...
push 0
call ExitProcess ; Выход из программы
end start
```

В некоторых случаях целесообразно складывать сразу по 2 или 4 байта для уменьшения количества операций.

Вариант	Выражение	Размер входных параметров	Операция Размерность (2-е задание)
1	$\frac{a+a^2+bd-1}{a-d/8}$	a – dword b – byte d – word	сложение 17 байт
2	$ab+ad+bd-\frac{a+1}{d}-1$	a – word b – word d – byte	вычитание 14 байт
3	$\frac{(x-3^4)^2+(y-4)^2+(z+5)^2}{4}$	x – word y – word z – word	сложение 30 байт
4	$\frac{abe+ab-\frac{b}{e}-1}{a+1+3^3}$	a – byte b – dword e – byte	вычитание 17 байт
5	$\frac{t+10^5}{s-2^5}+(r+1)^2$	t – dword r – word s – byte	сложение 16 байт
6	$((k+1)^2+1)^2-l/m+1$	k – word l – dword m – byte	вычитание 15 байт
7	$i^3+j^3-k^3+10^7/i$	i, j, k – word	сложение 21 байт
8	$(x+10)(y-5)\left(z-\frac{z}{3}\right)-7^4$	x, y, z – word	вычитание 15 байт

9	$t_1(x-a) + \frac{t_2(x-a)^2}{2} + \frac{t_3(x-a)^3}{6}$	t – массив из 3 чисел типа byte x, a – word	сложение 17 байт
10	$\frac{\left(\frac{k}{m}+1\right)^3}{5} + 9^4$	k – dword m – byte	вычитание 22 байта
11	$(r/s+5^5)^4 - 1$	r – dword s – byte	сложение 20 байт
12	$\frac{50i+170j+200k}{50-100k} + 11^3$	i, j, k – word	вычитание 19 байт
13	$ax + bx^2 + dx^3 - 14^3$	a, b, d – byte x – word	сложение 14 байт
14	$\frac{100}{a^2} - \frac{7^5}{b^2} + \frac{3^7}{d^2} - 1$	a, b, d – byte	вычитание 18 байт
15	$\frac{x^2 - y^2}{y^2} + 14^3 + xz$	x, y, z – byte	сложение 27 байт
16	$\frac{15^4}{x^2 + \frac{y}{z}} - \frac{9^5}{xz} - 1$	x, y, z – byte	вычитание 25 байт
17	$\left(\frac{p}{q} + 1\right)^3 - ps$	p – dword q – byte s – word	сложение 26 байт
18	$\frac{(n+500)^2}{(m-10)^3} r + 1$	n – word m – byte r – word	вычитание 25 байт
19	$vt + \frac{gt^2}{2} - 5^6$	v – word t – dword g – word	сложение 25 байт
20	$fg + \frac{f}{10^6 h} - \frac{g^3}{h^2}$	f – word g – dword h – byte	вычитание 16 байт
21	$i^4 - j^4 + \frac{k^4}{2^4} + 1$	i, j, k – byte	Сложение 14 байт
22	$(x-500)^2 + \left(\frac{y}{4} + 200\right)^2 + z^2$	x, y, z – word	вычитание 15 байт
23	$(l - (m+1)^2)^2 + \frac{m}{n} + 7^6$	m – byte l – word n – byte	сложение 19 байт
24	$(7+a)^2 + ab - \frac{12c}{d} - 1$	a – dword b – word c – word d – byte	вычитание 26 байт
25	$\frac{(ce+30k)^2}{(ce-40k)^2} + \frac{5^{10}}{e}$	c, e, k – byte	сложение 28 байт

