Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт

по лабораторной работе № 1

на тему:

“Изучение арифметических команд”

по дисциплине “Организация ЭВМ и Систем”

Вариант 4

Выполнил студент гр. 4306: Табаков А.В.

Принял: Манирагена Валенс

Санкт-Петербург  
2016

Цель

Ознакомиться с арифметическими операциями на языке ассемблера intel 8086. Получить знания по этапам разработки на ассемблере, а также навыки работы с программами tasm, tlink, turbo debugger.

Текст программы с данными из табл. 1.5

TITLE LAB1.1

.Model Tiny

.STACK 100h

.Data

.Code

begin:

mov al, 100d

mov bl, 76d

add ax, bx

mov bl, 20d

sub al, bl

mov bl, 65d

mul bl

xor ax, ax

mov al, 100d

mov bl, 66d

add al, bl

mov bl, -30d

sub al, bl

mov bl, 97d

mul bl

mov ax, 2047d

mov bx, 255d

add ax, bx

mov bx, 1040d

sub ax, bx

mov bx, 8193d

mul bx

mov ah, 4Ch

int 21h

END begin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Трассировка программы из табл. 1.5** | | | | |
| Адрес | Мнемокод | Двоичный код | Изменения регистров | Комментарий |
| 0000 | mov al, 100d | Байт 1: 10110000  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  000 –код регистра AL  Байт 2: Число | AL=64h |  |
| 0002 | mov bl, 76d | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=4Сh |  |
| 0004 | add al, bl | Байт 1: 00000010  000000 – операция сложения  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  011 – код регистра источника (BL) | AL=B0h  S=O=A=1 | Флаги установлены:  Переполнения, Полупереноса,  Знака (ст. бит). |
| 0006 | mov bl, 20d | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=14h |  |
| 0008 | sub al, bl | Байт 1: 00101010  001010 – операция вычитания  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  011 – код регистра источника (BL) | DL=9Ch  S=P=1 | Флаги установлены:  Паритета,  Знака |
| 000A | mov bl, 65d | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=41h |  |
| 000C | mul bl | Байт 1: 11110110  1111011 – Вызвать операции по адресу  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11100011  11 – указываем что работаем с регистрами  100 – (№ команды) без знаковое умножение  011 – код регистра источника (BL) | AX=279C  C=O=1 | Флаги установлены:  Переноса,  Переполнения |
| 000E | xor ax, ax | Байт 1: 00110011  001010 – операция вычитания  1 – сначала приёмник потом источник  1 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000000  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AX)  000 – код регистра источника (AX) | AX=0000  Z=0 | Обнуляем AX для дальнейшей работы  Флаг установлен:  Нуля |
| 0010 | mov al, 100d | Байт 1: 10110000  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  000 – код регистра AL  Байт 2: Число | AL=64h |  |
| 0012 | mov bl, 66d | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=42h |  |
| 0014 | add al, bl | Байт 1: 00000010  000000 – операция сложения  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  011 – код регистра источника (BL) | AL=A6h  S=O=1 | Флаги установлены:  Переполнения,  Знака (ст. бит). |
| 0016 | mov bl, -30d | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=E2h |  |
| 0018 | sub al, bl | Байт 1: 00101010  001010 – операция вычитания  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  011 – код регистра источника (BL) | AL=C4h  C=1 | Флаг  установлен:  Переноса |
| 001A | mov bl, 97d | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=61h |  |
| 001C | mul bl | Байт 1: 11110110  1111011 – Вызвать операции по адресу  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11100011  11 – указываем что работаем с регистрами  100 – (№ команды) без знаковое умножение  011 – код регистра источника (BL) | AX=4A44h  O=1 | Флаг  установлен:  Переполнения |
| 001E | mov ax, 2047d | Байт 1: 10111000  1011 – операция занесения числа в регистр  1 – слово (0 - 8 бит, 1 - 16 бит)  000 – код регистра AX  Байт 2: Число | AX=07FFh |  |
| 0021 | mov bx, 255d | Байт 1: 10111011  1011 – операция занесения числа в регистр  1 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BX  Байт 2: Число | BX=00FFh |  |
| 0024 | add ax, bx | Байт 1: 00000011 – операция сложения  000000 – операция сложения  1 – сначала приёмник потом источник  1 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AX)  011 – код регистра источника (BX) | AX=08FEh  A=1 | Флаг  установлен:  Доп. переноса |
| 0026 | mov bx, 1040d | Байт 1: 10111011  1011 – операция занесения числа в регистр  1 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BX  Байт 2: Число | BX=0410h |  |
| 0029 | sub ax, bx | Байт 1: 00101011  001010 – операция вычитания  1 – сначала приёмник потом источник  1 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AX)  011 – код регистра источника (BX) | AX=04EEh  P=1 | Флаг  установлен:  Паритета |
| 002B | mov bx, 8193d | Байт 1: 10111011  1011 – операция занесения числа в регистр  1 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BX  Байт 2: Число | BX=2001h |  |
| 002E | mul bx | Байт 1: 11110110  1111011 – Вызвать операции по адресу  1 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11100011  11 – указываем что работаем с регистрами  100 – (№ команды) без знаковое умножение  011 – код регистра источника (BX) | AX=C4EEh  DX=009Dh  C=O=1 | Флаги  установлены:  Переноса,  Переполнения |
| 0030 | mov ah, 4Ch | Байт 1: 10110100  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  100 – код регистра AH  Байт 2: Число | AH=4Ch |  |
| 0032 | int 21h | Байт 1: 11001101 – вызов прерывания  Байт 2: Номер прерывания | - |  |

Текст программы с данными из табл. 1.6

TITLE LAB1.2

.Model Tiny

.STACK 100h

.Data

.Code

begin:

mov al, 37h

mov bl, 68h

add al, bl

daa

mov ah, al

mov al, 10h

mov bl, 20h

adc al, bl

daa

mov bh, al

mov al, ah

sub al, 96h

das

mov ah, al

mov al, bh

sbb al, 35h

das

mov bl, al

mov al, ah

mov ah, bl

mov ah, 4ch

int 21h

END begin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Трассировка программы из табл. 1.6** | | | | |
| Адрес | Мнемокод | Двоичный код | Изменения регистров | Комментарий |
| 0000 | mov al, 37h | Байт 1: 10110000  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  000 – код регистра AL  Байт 2: Число | AL=37h |  |
| 0002 | mov bl, 68h | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=68h |  |
| 0004 | add al, bl | Байт 1: 00000010  000000 – операция сложения  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  011 – код регистра источника (BL) | AL=9Fh  S=O=P=1 | Флаги установлены:  Переполнения, Паритета,  Знака (ст. бит). |
| 0006 | daa | Байт 1: 00100111 – операция корректировки после сложения | AL=05h  C=A=1 | Флаги установлены:  Перенос,  Доп. перенос |
| 0007 | mov ah, al | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11100000  11 – указываем что работаем с регистрами  100 – код регистра приёмника (AH)  000 – код регистра источника (AL) | AH=05h |  |
| 0009 | mov al, 10h | Байт 1: 10110000  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  000 – код регистра AL  Байт 2: Число | AL=10h |  |
| 000B | mov bl, 20h | Байт 1: 10110011  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  011 – код регистра BL  Байт 2: Число | BL=29h |  |
| 000D | adc al, bl | Байт 1: 00010010  000100 – операция сложения с переносом  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000011  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  011 – код регистра источника (BL) | AL=31h |  |
| 000FE | daa | Байт 1: 00100111 – операция корректировки после сложения | - |  |
| 0010 | mov bh, al | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11111000  11 – указываем что работаем с регистрами  111 – код регистра приёмника (BH)  000 – код регистра источника (AL) | BH=31h | Запомнили старший байт |
| 0012 | mov al, ah | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000100  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  100 – код регистра источника (AH) | AL=05h |  |
| 0014 | sub al, 96h | Байт 1: 00101100 - операция вычитания числа из регистра AL  Байт 2: Число | AL=6Fh  C=P=A=1 | Флаги установлены:  Переноса,  Паритета,  Доп. переноса |
| 0016 | das | Байт 1: 00101111 – операция корректировки после вычитания | AL=09h |  |
| 0017 | mov ah, al | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11100000  11 – указываем что работаем с регистрами  100 – код регистра приёмника (AH)  000 – код регистра источника (AL) | AL=09h |  |
| 0019 | mov al, bh | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000111  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  111 – код регистра источника (BH) | AL=31h |  |
| 001B | sbb al, 35h | Байт 1: 00101100 - операция вычитания числа из регистра AL с заимствованием  Байт 2: Число | AL=FBh  S=1 | Флаг  установлен:  Знака |
| 001D | das | Байт 1: 00101111 – операция корректировки после вычитания | AL=95h |  |
| 001E | mov bl, al | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11011000  11 – указываем что работаем с регистрами  011 – код регистра приёмника (BL)  000 – код регистра источника (AL) | BL=95h |  |
| 0020 | mov al, ah | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11000100  11 – указываем что работаем с регистрами  000 – код регистра приёмника (AL)  100 – код регистра источника (AH) | AL=09h |  |
| 0022 | mov ah, bl | Байт 1: 10001010  100010 – операция переноса из регистра в регистр  1 – сначала приёмник потом источник  0 – слово (0 - 8 бит, 1 - 16 бит)  Байт 2: 11100011  11 – указываем что работаем с регистрами  100 – код регистра приёмника (AH)  011 – код регистра источника (BL) | AH=95h |  |
| 0024 | mov ah, 4Ch | Байт 1: 10110100  1011 – операция занесения числа в регистр  0 – слово (0 - 8 бит, 1 - 16 бит)  100 – код регистра AH  Байт 2: Число | AH=4Ch |  |
| 0026 | int 21h | Байт 1: 11001101 – вызов прерывания  Байт 2: Номер прерывания | - |  |

Текст программы с вводом-выводом

.Model tiny

.Data

greeting db "This programm do (A+B-D)xC", 0dh, 0ah, "$"

help db "Please, input number from -9000 to 9000", 0dh, 0ah, "$"

textA db "A = $"

textB db "B = $"

textC db "C = $"

textD db "D = $"

textRes db "Result = $"

endl db 0ah, 0dh, "$"

pkey db "Press any key...$"

negative db 0

arr dw 4 dup(?)

buffer db 6 ;max num with 4 symbols

blength db ?

bconteg: ;consistance of buf is over of prog

hexstring equ bconteg

.Stack 0100h

.Code

start:

mov ax, @data

mov ds, ax

mov es, ax

call setDisp

lea dx, greeting ;greeting message

mov ah, 09h

int 21h

lea dx, help ;help message

mov ah, 09h

int 21h

lea dx, textA ;A=

mov ah, 09h

int 21h

call input ;input A

mov arr, ax

call endlp

lea dx, textB ;B=

mov ah, 09h

int 21h

call input ;input B

mov arr+2, ax

call endlp

lea dx, textC ;C=

mov ah, 09h

int 21h

call input ;input C

mov arr+4, ax

call endlp

lea dx, textD ;D=

mov ah, 09h

int 21h

call input ;input D

mov arr+6, ax

call endlp

mov ax, arr

mov bx, arr+2

add ax, bx ;A+B

mov bx, arr+6

sub ax, bx ;(A+B)-D

mov bx, arr+4

mul bx ;(A+B-D)xC

jns printing

mov negative, 1

printing:

push ax

lea dx, textRes ;Result=

mov ah, 09h

int 21h

pop ax

call print ;output result

call endlp

call quit

proc setDisp

xor dx,dx ;cursor's position

mov ah,02h ;set at (0,0)

int 10h

mov bl,00001010b ;colors green on black

mov cx,25\*80 ;count of simbols on display

mov ax,0920h ;printing 25\*80 spaces

int 10h

ret

endp

proc quit

lea dx, pkey

mov ah, 9h

int 21h ; output string at ds:dx

; wait for any key....

mov ah, 1h

int 21h

mov ax, 4c00h ; exit to operating system.

int 21h

endp

proc endlp ;press enter

push dx

push ax

lea dx, endl

mov ah, 09h

int 21h

pop ax

pop dx

ret

endp

proc input

lea dx,buffer ;buffer's address

mov ah,0ah ;write in buffer

int 21h

;from string to bin

xor di,di ;start of buffer

xor ax,ax ;clear ax

mov cl,blength

xor ch,ch

xor bx,bx

mov si,cx ;buffer's length

mov cl,10 ;multiplier

mov bl,byte ptr bconteg[di]

cmp bl, '-'

jnz toHex

mov negative, 1

inc di

toHex:

mov bl,byte ptr bconteg[di]

sub bl,'0' ;num = num's code - 30h

jb badInp ;if symbol not a num

cmp bl,9 ;same

ja badInp ;try input again

mul cx ;multiply on 10

add ax,bx ;+new num to ax

inc di ;next symbol

cmp di,si ;if di<blength + 1

jb toHex

cmp negative, 1 ;num is negative

jnz nM

neg ax

mov negative, 0

nM:

jmp endInp

badInp:

jmp start

endInp:

ret

endp

proc print

cmp negative, 1 ;if num<0

jnz stPrint

push ax

mov dl, '-'

mov ah, 02h

int 21h

pop ax

neg ax

stPrint:

mov bx,0ah ;divider

xor cx,cx ;clear count

divloop:

xor dx,dx ;clear dx

div bx ;divide on 10

add dx,'0' ;make a symbol from num

push dx ;save dx

inc cx

test ax,ax ;if ax!=0

jnz divloop ;continue to divide

restore:

;pop ax ;read from stack

pop ax

mov dx, ax ;

mov ah,2 ;print symbol from al

int 21h ;

loop restore

ret

endp

end start ; set entry point and stop the assembler.

Вывод

Я ознакомился с арифметическими операциям на языке ассемблера intel 8086.

Получил знания по этапам разработки на ассемблере, а также навыки работы с программами tasm, tlink, turbo debugger.