

Лабораторная работа №1

**Освоение инструментальной среды
для выполнения лабораторных
работ**

Структура процессора i8086 для программиста и обзор системы команд

Для программиста при программировании на языке ассемблера микропроцессор i8086 имеет в своём составе 14 шестнадцатиразрядных регистров:

- регистры общего назначения (РОН) – AX, BX, CX, DX, SI, DI ;
- сегментные регистры – CS, DS, ES, SS;
- специальные регистры – IP, SP, BP, FLAGS.

Регистры общего назначения

Регистры общего назначения (16 разрядные):
AX(АН, AL), BX(ВН, BL), CX(СН, CL), DX(DH, DL)
делятся программно на пары однобайтных регистров
и используются для хранения данных (8 или 16
битных);

SI, DI – шестнадцатиразрядные регистры для
хранения данных.

AX (16 бит)	АН (8 бит)	AL (8 бит)
BX (16 бит)	ВН (8 бит)	BL (8 бит)
CX (16 бит)	СН (8 бит)	CL (8 бит)
DX (16 бит)	DH(8 бит)	DL (8 бит)
SI (16 бит)		
DI (16 бит)		

Сегментные регистры

Регистры CS, DS, ES, SS – хранят адреса сегментов в памяти (кодирового, данных, дополнительных данных, стека) и **не могут использоваться для хранения данных.**

CS (16 бит)	Начало сегмента кода (программы) в ОП
DS (16 бит)	Начало сегмента данных программы в ОП
ES (16 бит)	Начало дополнительного сегмента в ОП
SS (16 бит)	Начало сегмента стека программы в ОП

Специальные регистры

SP, BP – указатель и база стека, соответственно, обеспечивают доступ к данным в стеке, могут использоваться для хранения данных, но делать это не рекомендуется, так как при этом возможно нарушение адресации в стеке, особенно при использовании SP.

IP – регистр инструкций (счетчик команд) – хранит адрес следующей исполняемой команды (относительно смещения).

FLAGS – регистр флагов содержит набор битовых флагов, определяющий результат выполнения предыдущей команды.

SP (16 бит)	Указатель начала стека в ОП
BP (16 бит)	Указатель начала базы в ОП
IP (16 бит)	Указатель (счетчик) команд ОП
FLAGS (16 бит)	Регистр флагов

Флаги

Регистр флагов процессора

Флаг	Название	Назначение	
OF	O	Переполнение	Переполнение при выполнении арифметических операций
	D	Направление	Направление пересылки данных при выполнении строковых команд
	I	Прерывание	Разрешает/Запрещает внешние прерывания
	T	Пошаговый режим	Останов после выполнения каждой команды(используется отладчиками)
SF	S	Знак	Знак результата выполненной команды(0 – плюс, 1 – минус)
ZF	Z	Ноль	Значение результата выполненной команды(0 – ненулевой, 1 – нулевой)
	A	Внешний перенос	Используется для специальных арифметических операций
	P	Контроль чётности	Число единиц в операнде(0 – нечётное, 1 – нечётное)
CF	C	Перенос	Содержит перенос из старшего бита при выполнении арифметических операциях

Оперативная память

Память, с которой взаимодействует процессор при выполнении программы, называется Оперативным Запоминающим Устройством (**ОЗУ**) или Random Access Memory (**RAM**). Также используется аббревиатура – ОП.

Память состоит из набора однобайтных ячеек, обращение к которым происходит по их номерам (физическим адресам). Адресация начинается с 0000

Исполняемая программа (загруженный exe-файл) в ОП состоит из трех последовательно расположенных сегментов (частей). Начало каждого сегмента определяется значением регистров DS (данные), CS (код программы), SS (стек программы).

Значение этих регистров определяется при линковке программы (Tlink).

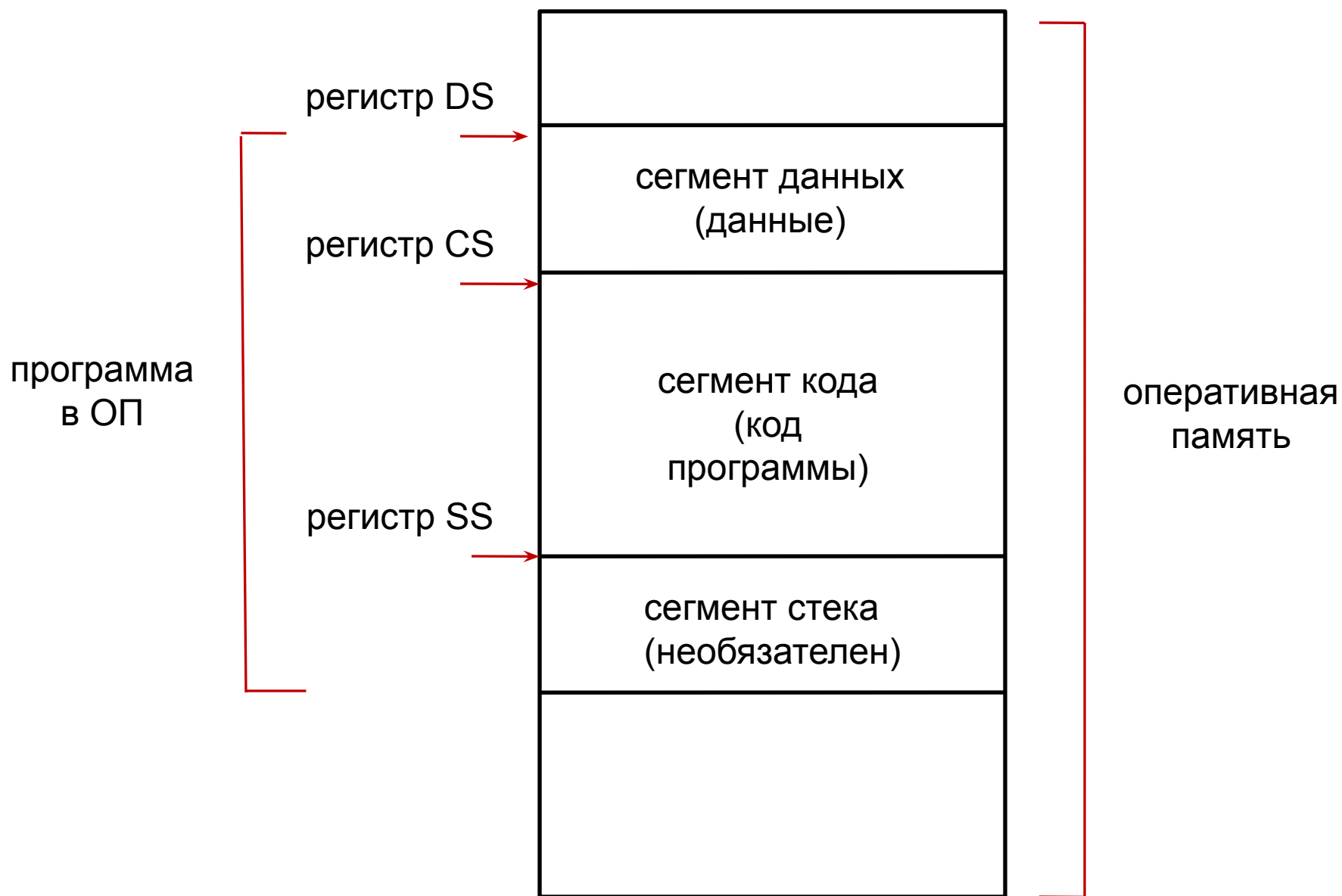
Используемый инструментарий

Для получения exe-файла и его отладки будут нужны:

- текстовый редактор – Notepad, Notepad++ и др. -> file.asm
- компилятор языка ассемблера – Tasm -> file.obj
- линковщик – Tlink -> file.exe
- отладчик - Td

Tasm, Tlink, Td – это 16-ти разрядные приложения и для их работы в Windows 7, 8, 10 потребуется приложение "Dosbox".
Внутри этого приложения будет использоваться файловый менеджер Norton Commander (NC)

Структура данных и кода в оперативной памяти



Turbodebugger (отладчик)

окно процессора
(код программы)

окно регистров процессора

окно флагов процессора

The screenshot shows the Turbodebugger interface with the following components:

- Assembly Window (CPU 80486):** Displays assembly instructions with their addresses and operands. The current instruction is `mov ax, [0000]` at address `cs:0005A10000`.
- Registers Window:** Shows the current values of the 80486 registers: `ax 5B42`, `bx 0000`, `cx 0000`, `dx 0000`, `si 0000`, `di 0000`, `bp 0000`, `sp 0000`, `ds 5B42`, `es 5B32`, `ss 5B42`, `cs 5B43`, and `ip 0005`.
- Flags Window:** Shows the status of the processor flags: `c=0`, `z=0`, `s=0`, `o=0`, `p=0`, `a=0`, `i=1`, and `d=0`.
- Memory Window (OP):** Displays the memory contents at the current instruction address, showing the instruction bytes: `ds:0000 0A 00 14 00 05 00 00 00`.
- Stack Window:** Shows the stack contents, with the current stack pointer at `ss:0000 000A`.

The status bar at the bottom indicates the current state: `READY`. The menu bar includes: `E File View Run Breakpoints Data Options Window Help`. The keyboard shortcuts are: `F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu`.

окно ОП
(данные)

окно стека
(стек)

Окно процессора

столбец адресов
команд

столбец кодов
команд

два столбца
мнемоники команд

```
C:\WINDOWS\system32\cmd.exe - td ke
E File View Run Breakpoints Data Options Window Help
[ ]=CPU 80486
cs:0000 B8425B  mov ax,5B42
cs:0003 8ED8    mov ds,ax
cs:0005 A10000    mov ax,[0000]
cs:0008 D1E0    shl ax,1
cs:000A 03060000  add ax,[0000]
cs:000E 8B1E0200  mov bx,[0002]
cs:0012 83C305    add bx,0005
cs:0015 D1FB    sar bx,1
cs:0017 03C3    add ax,bx
cs:0019 2B060400  sub ax,[0004]
cs:001D 48      dec ax
cs:001E A30600    mov [0006],ax
cs:0021 B8004C    mov ax,4C00
cs:0024 CD15    int 15
cs:0026 0000    add [bx+si],al
```

команда `mov ax,5B42` находится в ОП по адресам `0000 – 0002` (занимает три байта) и имеет код - `B8425B`

Окно ОП (ОЗУ)

столбец адресов
байт ОП

значение байт
по этим адресам

символьное отображение
байтов

ds:0000	0A	00	14	00	05	00	00	00	☐	¶	♣			
ds:0008	00	00	00	00	00	00	00	00						
ds:0010	B8	42	5B	8E	D8	A1	00	00	3	B	[0	1	6	
ds:0018	D1	E0	03	06	00	00	8B	1E	7	p	♥	♣	♠	
ds:0020	02	00	83	C3	05	D1	FB	03	0	Г		♣	7	♥

значение ds=5B42

по адресу ОЗУ = 5B42+0000=5B42 значение байта=0Ah

по адресу ОЗУ = 5B42+0001=5B42 значение байта=00h

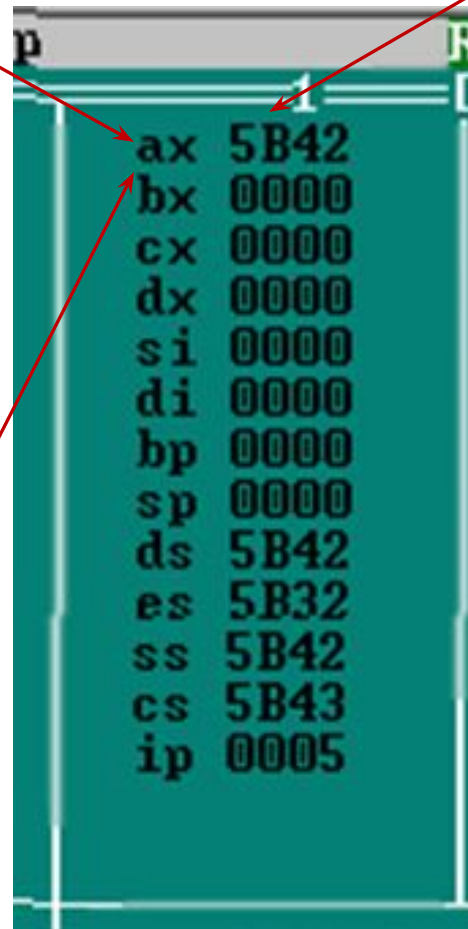
по адресу ОЗУ = 5B42+0002=5B42 значение байта=14h

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step

Окно регистров

столбец имен
регистров

столбец значений
регистров



The image shows a screenshot of a debugger's register window. The window has a dark green background and a white border. At the top, there are labels 'p' and 'R'. The registers and their values are listed as follows:

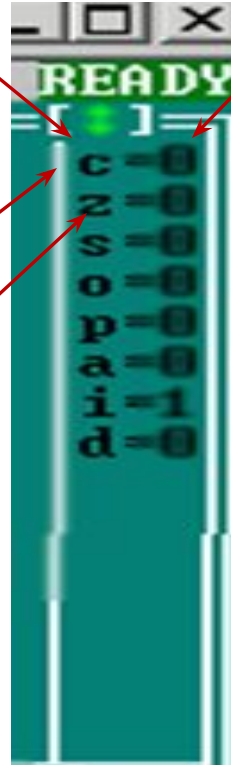
ax	5B42
bx	0000
cx	0000
dx	0000
si	0000
di	0000
bp	0000
sp	0000
ds	5B42
es	5B32
ss	5B42
cs	5B43
ip	0005

регистр AX=5B42h=0101101110000010b

Окно флагов

имя флага

значение флага



CF=0, ZF=0

окно стека: адрес байта, значение байта

The screenshot shows a DOS debugger window with the following content:

```
C:\WINDOWS\system32\cmd.exe - td ke
E File View Run Breakpoints Data Options Window Help
[ ]=CPU 80486
cs:0000 B8425B   mov ax,5B42
cs:0003 8ED8       mov ds,ax
cs:0005 A10000     mov ax,[0000]
cs:0008 D1E0       shl ax,1
cs:000A 03060000  add ax,[0000]
cs:000E 8B1E0200  mov bx,[0002]
cs:0012 83C305     add bx,0005
cs:0015 D1FB       sar bx,1
cs:0017 03C3       add ax,bx
cs:0019 2B060400  sub ax,[0004]
cs:001D 48         dec ax
cs:001E A30600     mov [0006],ax
cs:0021 B8004C     mov ax,4C00
cs:0024 CD15     int 15
cs:0026 0000       add [bx+si],al

ds:0000 0A 00 14 00 05 00 00 00  9  4
ds:0008 00 00 00 00 00 00 00 00
ds:0010 B8 42 5B 8E D8 A1 00 00  7 B[0]6
ds:0018 D1 E0 03 06 00 00 8B 1E  7  4  1  4
ds:0020 02 00 83 C3 05 D1 FB 03  8  7  4  7

ax 5B42
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0000
ds 5B42
es 5B32
ss 5B42
cs 5B43
ip 0005

ss:0008 0000
ss:0006 0000
ss:0004 0005
ss:0002 0014
ss:0000 000A

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu
```

Red arrows in the image point from the title to the register `ax` (value 5B42) and the stack address `ss:0002` (value 0014).

Задание на лабораторную работу и порядок её выполнения смотри в файле:

Лабораторная работа №1 часть 2 Задание.docx