

Тема 3. Регистры процессора.

Учебные вопросы:

1. Введение в регистры микропроцессора.
2. Команды сложения и вычитания.

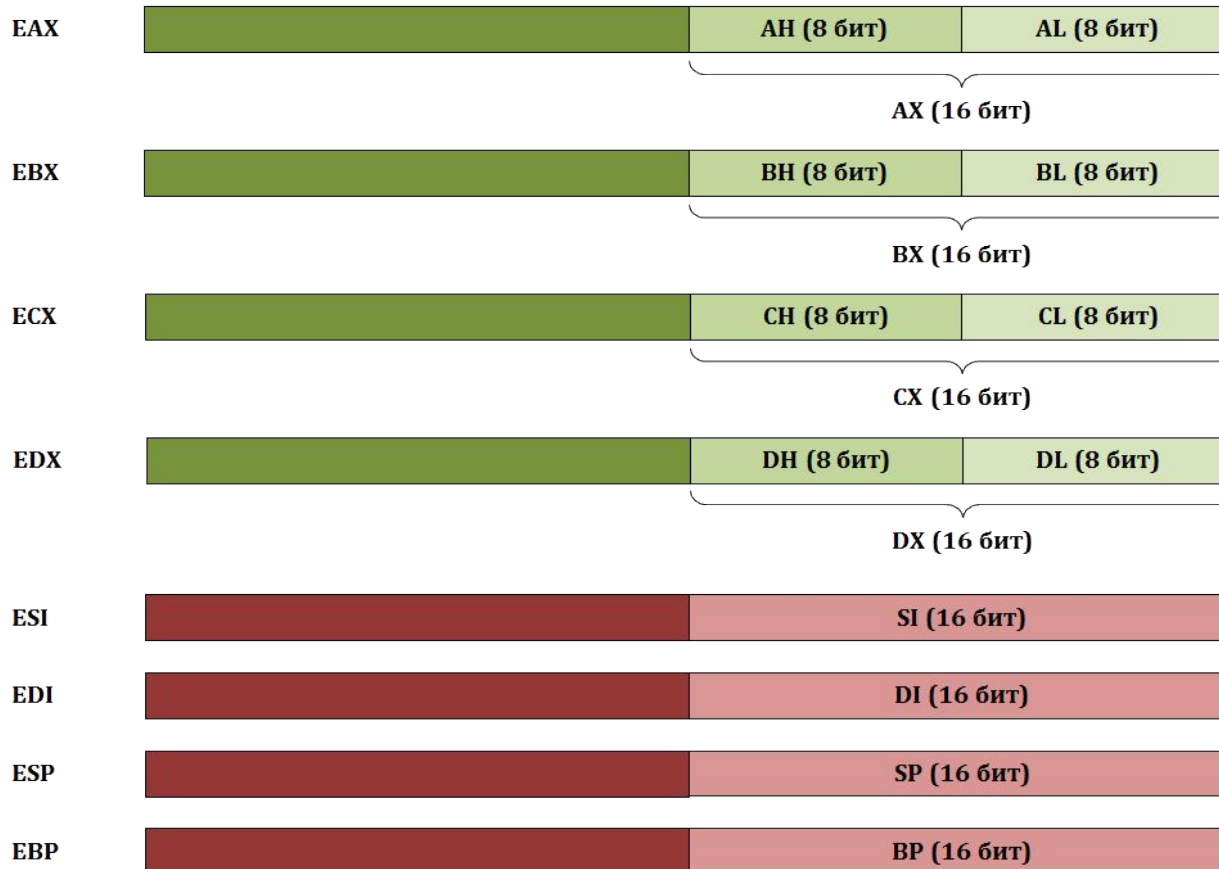
Учебная литература:

1. Калашников О. А. Ассемблер? Это просто! Учимся программировать. — СПб.: БХВ-Петербург, 2006.
2. Пирогов В. Ю. Ассемблер и дизассемблирование. — СПб.: БХВ-Петербург, 2006.
3. Абель П. Язык Ассемблера для IBM PC и программирования. / Перевод. с англ.- М.: Высшая школа, 1992.

1. Введение в регистры микропроцессора.

Регистры можно разделить на:

- регистры общего назначения (регистры данных — EAX, EBX, ECX и EDX; индексные — ESI, EDI; и указательные — EBP,ESP);
- указатель команд — EIP;
- регистр флагов;
- сегментные регистры — CS, DS, SS, ES, FS и GS.



Названия регистров происходят от их назначения:

- **EAX/AX/AH/AL** (*accumulator register*) – аккумулятор;
- **EBX/BX/BH/BL** (*base register*) – регистр базы;
- **ECX/CX/CH/CL** (*counter register*) – счётчик;
- **EDX/DX/DH/DL** (*data register*) – регистр данных;
- **ESI/SI** (*source index register*) – индекс источника;
- **EDI/DI** (*destination index register*) – индекс приёмника (получателя);
- **ESP/SP** (*stack pointer register*) – регистр указателя стека;
- **EBP/BP** (*base pointer register*) – регистр указателя базы кадра стека.

- 32-разрядные регистры могут хранить числа от 0 до 4 294 967 295 (0FFFFFFFFh);
- 16-разрядные - могут хранить числа от 0 до 65 535 (FFFFh);
- 8-разрядные регистры - максимальное число 255 (FFh).

MOV приемник, источник

```
mov al, 35h  
mov ax, 346Ah
```

```
mov ah, 123h - максимум FFh  
mov bx, 12345h - максимум FFFFh  
mov dl, 100h - максимум FFh
```

Команда	Результат
mov ax, 1234h	ax = 1234h, ah = 12h, al = 34h
mov bx, 5678h	bx = 5678h, bh = 56h, bl = 78h
mov cx, 9ABCh	cx = 9ABCh, ch = 9Ah, cl = 0BCh
mov dx, 0DEF0h	dx = 0DEF0h, dh = 0DEh, dl = 0F0h

dh — это старшие (High) половинки dx

dl — это младшие (Low) половинки dx

Примеры применения команды mov

```
mov ax,[number] ; заносим значение переменной number в регистр AX  
mov [number],bx ; загрузить значение регистра BX в переменную number  
mov bx,cx ; занести в регистр BX значение регистра CX  
mov al, 1 ; занести в регистр AL значение 1  
mov dh,cl ; занести в регистр DH значение регистра CL  
mov esi,edi ; копировать значение регистра EDI в регистр ESI
```

```
mov [number_two], [number_one] ;НЕПРАВИЛЬНО!!!
```



```
mov ax, [number_one] ;загружаем в AX 16-битное значение "number_one"  
mov [number_two], ax ;а затем копируем его в переменную "number_two"
```

```
mov ax, bl ;НЕПРАВИЛЬНО!
```



```
mov ax, bx ;загружаем BX в AX  
mov ah, 0 ;"сбрасываем" старшую часть AX — записываем в нее 0
```

ИЛИ

```
mov ax, 0 ;AH = 0, AL = 0  
mov al, bl ;вносим в AL значение BL
```

C:\ CV.EXE

File Edit Search Run Data Options Calls Windows Help

[3] source1 CS:IP

1293:0100	B409	MOV	AX,09
1293:0102	BA0D01	MOV	DX,010D
1293:0105	CD21	INT	21
1293:0107	B410	MOV	AX,10
1293:0109	CD16	INT	16
1293:010B	CD20	INT	20
1293:010D	50	PUSH	AX
1293:010E	7265	JB	0175
1293:0110	7373	JNB	0185
1293:0112	20616E	AND	BYTE PTR [BX+DI+6E],AH
1293:0115	7920	JNS	0137
1293:0117	6B65793A	IMUL	SP,[DI+79],3A
1293:011B	2467	AND	AL,67
1293:011D	668916F827	MOV	DWORD PTR [27F8],EDX
1293:0122	5F	POP	DI
1293:0123	5E	POP	SI
1293:0124	5D	POP	BP
1293:0125	C20800	RET	0008
1293:0128	6657	PUSH	EDI
1293:012A	6656	PUSH	ESI
1293:012C	1E	PUSH	DS
1293:012D	B91000	MOV	CX,0010
1293:0130	8B1EE227	MOV	BX,WORD PTR [27E2]
1293:0134	8EC3	MOV	ES,BX
1293:0136	BF0020	MOV	DI,2000
1293:0139	C1EB08	SHR	BX,08
1293:013C	87CB	XCHG	BX,CX
1293:013E	B806DE	MOV	AX,DE06

[7] register

EAX	=	00000000
EBX	=	00000000
ECX	=	00000000
EDX	=	00000000
ESP	=	0000FFFE
EBP	=	00000000
ESI	=	00000000
EDI	=	00000000
DS	=	1293
ES	=	1293
FS	=	0000
GS	=	0000
SS	=	1293
CS	=	1293
EIP	=	00000100
EFL	=	00000200

NU UP EI PL
MZ NA PO NC

[5] memory1 b DS:0

1293:0000	CD 20 FF 9F 00 9A F0 FE 1D F0 96 02	= Я.бЕИ++ЕЦ
1293:000C	8A 10 97 03 8A 10 56 01 15 04 41 12	К>4*К>U@S+A
1293:0018	01 01 01 00 01 01 FF FF FF FF FF FF	000.00
1293:0024	FF FF FF FF FF FF FF FF 52 12 E8 36	Rm6
1293:0030	31 05 14 00 18 00 93 12 FF FF FF FF	1*9.↑.9
1293:003C	00 00 00 00 00 05 00 00 00 00 00 00
1293:0048	00 00 00 00 00 00 00 00 00 CD 21 CB 00=?
1293:0054	00 00 00 00 00 00 00 00 00 00 20 20 20
1293:0060	20 20 20 20 20 20 20 20 00 00 00 00
1293:006C	00 20 20 20 20 20 20 20 20 20 20 20
1293:0078	00 00 00 00 00 00 00 00 00 0D 00 00P..

[9] command

```

CV1053 Warning: TOOLS.INI not found
CV0101 Warning: no CodeView information for 'c:\source\prog02.com'
>

```

<F8=Trace> <F10=Step> <F5=Go> <F3=S1 Fmt> <Sh+F3=M1 Fmt> DEC

2. Команды сложения и вычитания.

add, sub, inc, dec

Команда (применение)	Перевод	Назначение	Процессор
add приемник, источник	Addition — сложение	Сложение	8086

Примеры использования оператора add:

```
mov al,10      ;загружаем в регистр al число 10
add al,15      ;al = 25; al — приемник, 15 — источник

mov ax,25000   ;загружаем в регистр ax число 25000
add ax,10000   ;ax = 35000; ax — приемник, 10000 — источник

mov cx,200     ;загружаем в регистр cx число 200
mov bx,760     ;a в регистр bx — 760
add cx,bx      ;cx = 960, bx = 760 (bx не меняется)
               ;cx — приемник, bx — источник
```

Команда	Перевод	Назначение	Процессор
sub приемник, сточник	Subtraction — вычитание	Вычитание	8086

Примеры использования оператора **sub**:

```

mov al,10
sub al,7           ;al = 3; al — приемник, 7 — источник

mov ax,25000
sub ax,10000      ;ax = 15000; ax — приемник, 10000 — источник

mov cx,100
mov bx,15
sub cx,bx         ;?

```


Команда	Перевод	Назначение	Процессор
inc приемник	Increment — инкремент	Увеличение на единицу	8086

Примеры использования оператора `inc`:

```
mov al,15
inc al           ;теперь al = 16 (эквивалентна add al,1)
```

```
mov dh,39h
inc dh          ;dh = 3Ah (эквивалентна add dh,1)
```

```
mov cl,4Fh
inc cl          ;cl = 50h (эквивалентна add cl,1)
```

Команда	Перевод	Назначение	Процессор
dec приемник	Decrement — декремент	Уменьшение на единицу	8086

Примеры использования оператора dec:

```
mov al,15  
dec al           ;теперь al = 14
```

```
mov dh,3Ah  
dec dh          ;dh = 39h
```

```
mov cl,50h  
dec cl          ;cl = 4Fh
```