

Тема 3. Регистры процессора.

Учебные вопросы:

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

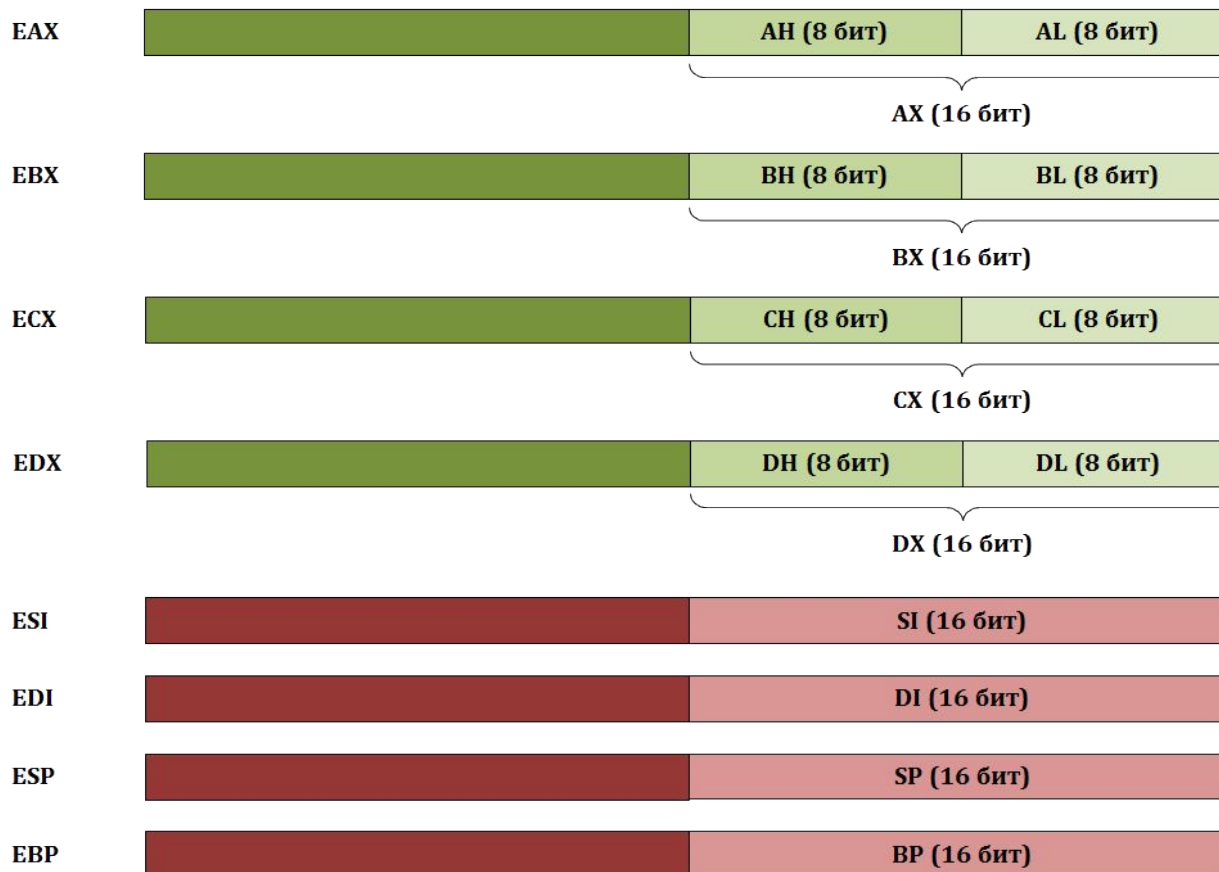
Учебная литература:

- 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 (**H**igh) половинки dx

dl — это младшие (**L**ow) 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"

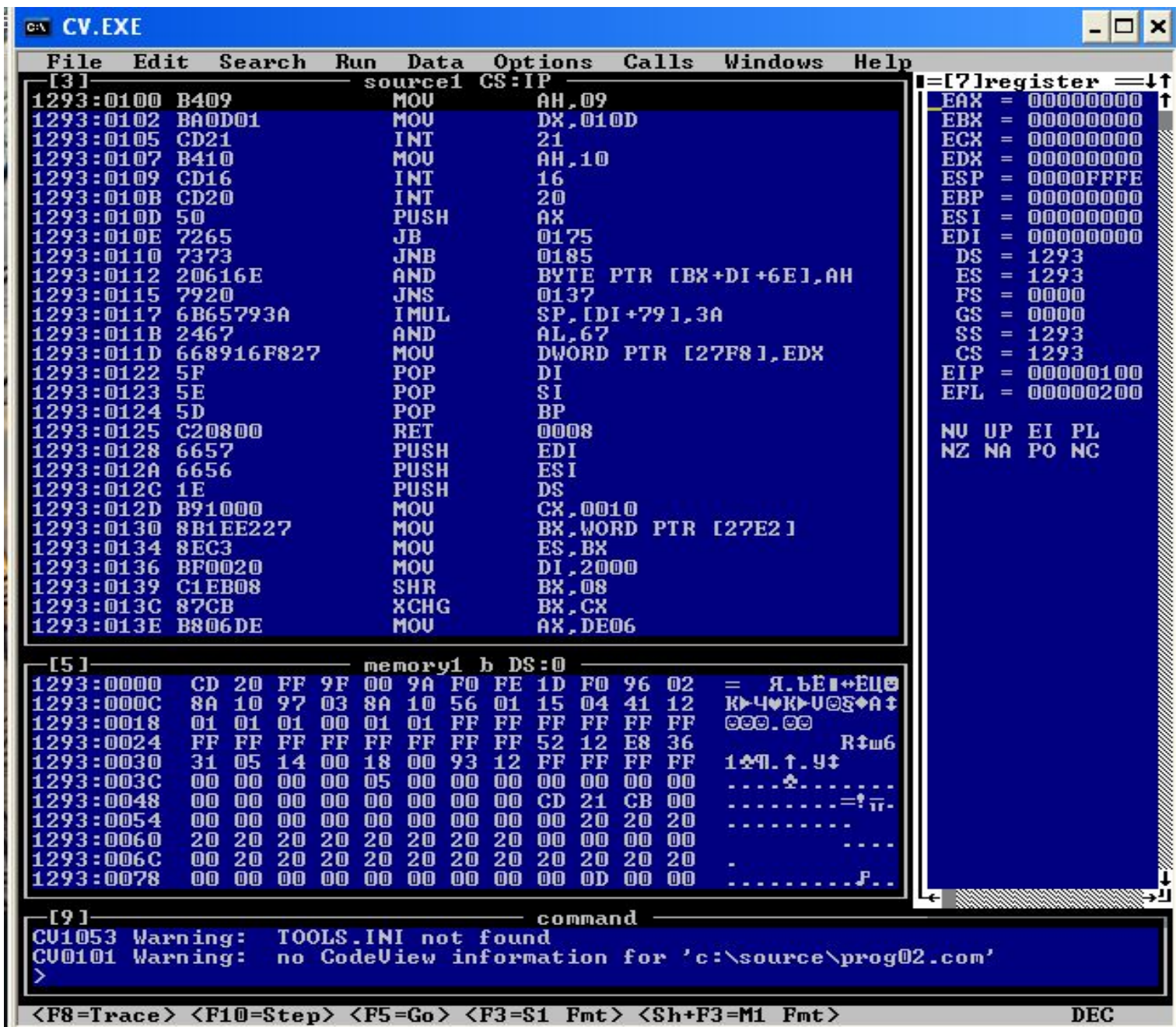
mov ax, bl ;НЕПРАВИЛЬНО!



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 ;записываем в AL значение BL



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     ;а 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
```