

Лекция 6

Раздел 2. Команды МП 8088/86

2.1. Состав команд

Микропроцессор имеет **92 команды**, которые можно разделить на 7 групп:

1. Команды пересылки данных между регистрами, ячейками и портами ввода/вывода;
2. Арифметические команды;
3. Команды над битами, которые осуществляют сдвиги и логические операции;
4. Команды передачи управления, вызова процедур и возвращения из процедуры;
5. Команды обработки строк;
6. Команды прерывания для обработки специфических событий;
7. Команды управления процессором - установление и сброс флагов состояния, изменения режима функционирования МП.

2.1.1. Команды пересылки данных

Команда **MOV** - чаще всего употребляется в программе. Ее можно применять для пересылки:

MOV AX, CX ; из регистра в регистр
MOV AX, TABLE ; из памяти в регистр
MOV TABLE, AX ; из регистра в память
MOV DS, AX ; из регистра в регистр сегмента
MOV AH, AL ; обмен байтами
MOV AX, -40 ; константу в регистр
MOV BETA, 2Fh ; константу в память

НЕЛЬЗЯ осуществлять такие пересылки:

Из памяти в память. Как отмечалось, в двухадресных командах нельзя использовать прямую адресацию в двух операндах. Поэтому пересылку память-память можно осуществить двумя командами:

~~MOV ALPHA, BETA~~

MOV AX, BETA
MOV ALPHA, AX

Содержание ячеек нельзя пересылать непосредственно в регистр сегмента, а только через регистр общего назначения:

~~MOV DS, BETA~~

MOV AX, BETA
MOV DS, AX

Нельзя пересылать данные из одного регистра сегмента в другой, а только через регистр общего назначения:

MOV AX, DS
MOV ES, AX

~~MOV ES, DS~~

Нельзя использовать регистр CS как приемник, то есть, **НЕЛЬЗЯ**:

~~MOV CS, AX~~

Стек автоматически создается для работы с подпрограммами. Но поскольку в МП лишь 4 регистра общего назначения, то часто придется запоминать содержание регистров, чтобы освободить регистры и выполнить другие действия. Именно для этого используются команды:

PUSH источник ; заслат в стек

POP приемник ; считать из стека

Например:

PUSH SI
PUSH DS
PUSH CX
PUSH ALPHA
PUSH DELTA [BI + SI]

Как отмечалось, данные засылаются на вершину стека, поэтому при их считывании необходимо сдерживаться соответствующей последовательности. Например:

PUSH AX
PUSH BX
PUSH DS

SP - 2 SP -	DS	SS:02F8
	BS	SS:02FA
	AX	SS:02FC
		SS:02FE

Поскольку сверху находится **DS**, а ниже - **BX** и **AX**, то возобновление регистров нужно осуществить в противоположном порядке

POP DS
POP BX
POP AX

Команды PUSH-POP можно использовать для обмена между сегментными регистрами:

PUSH DS
POP ES

При этом не используется РЗП. Но выполнение команд будет дольше: пара PUSH-POP реализуется за 26 тактов, а две команды MOV за 4 такта.

2.1.2. Команда обмена XCHG

Название происходит от английского слова exchange - обменять. Используется для обмена содержанием двух регистров или регистра и памяти.

XCHG CS, DS

XCHG BX, AX
XCHG AH, BL
XCHG AX, TABLE
XCHG TABLE, AX

НЕЛЬЗЯ использовать для обмена между сегментными регистрами.

2.1.3. Команды обмена с портами

IN аккумулятор, порт

OUT порт, аккумулятор

Аккумулятор - это регистр **AX** при обмене **словами** и **AL** при обмене **байтами**. Порт определяется своим номером от 0 до 256. Можно непосредственно отмечать номер порта или дать ему имя.

```
PORT_NUM EQU 210
IN AX, 200
IN AL, PORT_NUM
OUT DX, AX
OUT 200, AL
```

Также номер порта можно записать в регистр **DX**.

2.1.4. Команда *LEA* - загрузки эффективного адреса

LEA (load effective address - загрузить исполнительный адрес) пересылает смещение ячейки памяти в:

1. определенный 16-битовый регистр ЗУ
2. регистр указателя
3. индексный регистр

LEA регистр 16, память 16

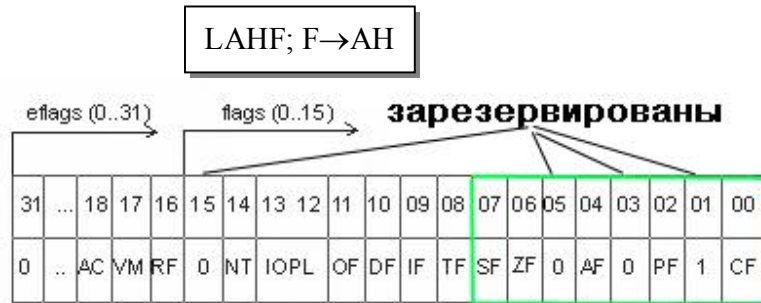
В отличие от команды **MOV** с операцией **OFFSET**, операнд память 16 может быть индексированным, что обеспечивает гибкость адресации. Например:

```
LEA BP, TABLE [DI]
```

Если в **DI** содержится 8, то в **BP** будет заслан адрес **TABLE + 8**.

2.1.5. Команды пересылки флагов

Можно пересылать в регистр **АH** младший байт регистра флагов **F** командой **LAHF**;



Обратная пересылка с **АH** в младший байт регистра **F** - **SAHF**:

SAHF; AH→F

Содержание регистра флагов можно также пересылать в стек командой **PUSHF**, а в обратном направлении - **POPF**.

Это нужно для защиты регистра флагов от изменения при вызове процедур. Если нет уверенности, что процедура не изменит регистр флагов, то его нужно защитить (сохранить).

Пример:

```

PUSH AX
PUSH DI
PUSHF
CALL SORT
POPF
POP DI
POP AX
        
```