**ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ.**

Команда ***MOV*** замещает операнд-приемник операндом-источником. При этом исходное значение операнда-приемника теряется. Операнд-источник не изменяется. В зависимости от описания операндов, пересылается слово или байт. Команда не воздействует на флаги процессора! В зависимости от используемых режимов адресации команда может осуществлять пересылки следующих видов:

* из РОН в РОН;
* из РОН в память;
* из РОН в сегментные регистры DS, ES, SS;
* из памяти в РОН;
* из памяти в сегментный регистр;
* из сегментного регистра в РОН;
* из сегментного регистра в память;
* непосредственный операнд в РОН;
* непосредственный операнд в память.

**ЗАПРЕЩЕНО:**

* пересылка из ячейки памяти в ячейку памяти (для этого предусмотрена команда MOVS);
* загрузка сегментного регистра непосредственным значением, которое загружается через РОН. Необходимо поступать так:

MOV AX, seg mem ; сегментный адрес ячейки памяти mem

MOV DS, AX ; загрузка его в регистр DS;

* пересылка содержимого одного сегментного регистра в другой. Такие операции выполняются с использованием стека.

MOV – MOVe operand – копирование содержимого операнда *источник* в операнд *приемник*.

## Действие: приемник = источник.

|  |  |  |  |
| --- | --- | --- | --- |
| 16-ричный код (1 байт) | MOD Reg/OPC Reg/Mem (2-ой байт) | смещение  disp\_Lo, disp\_Hi | формат операндов:  приемник, источник |
| 88 | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg8/Mem8, Reg8 |
| 89 | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg16/Mem16, Reg16 |
| 8A | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg8, Reg8/Mem8 |
| 8B | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg16, Reg16/Mem16 |
| 8C | MOD 0SegReg Reg/Mem | Disp\_Lo, Disp\_Hi | Reg16/Mem16, SegReg |
| 8E | MOD 0SegReg Reg/Mem | Disp\_Lo, Disp\_Hi | SegReg, Reg16/Mem16 |
| A0 | Disp\_Lo | Disp\_Hi | AL, Mem8 |
| A1 | Disp\_Lo | Disp\_Hi | AX, Mem16 |
| A2 | Disp\_Lo | Disp\_Hi | Mem8, AL |
| A3 | Disp\_Lo | Disp\_Hi | Mem16, AX |
| 10110Reg | Data8 | Не используется | Reg8, Imm8 |
| 10111Reg | Data16 (2 байта) |  | Reg16, Imm16 |
| C6 | MOD **000** Reg/Mem | Disp\_Lo, Disp\_Hi | Reg8/Mem8, Imm8 |
| C7 | MOD **000** Reg/Mem | Disp\_Lo, Disp\_Hi | Reg16/Mem16, Imm16 |

**000** - расширение кода операции при работе с непосредственным операндом, размещается в поле Reg/Opc в байте адресации.

Если MOD=11 форматами C6h и C7h не пользуются.

***Таблица кодировки сегментных регистров:***

|  |  |
| --- | --- |
| Поле SegReg | Сегментный регистр |
| 000 (00) | ES |
| 001 (01) | CS |
| 010 (10) | SS |
| 011 (11) | DS |

***Примеры:***

*;в сегменте данных:*

pam1 db 5 ; смещение в сегменте 1580h

pam2 dw 0 ; смещение в сегменте 1A20h

pam3 dd 0 ; смещение в сегменте 1FF0h

*;в программном сегменте:*

*MOV DX, AX ; из регистра в регистр*

Машинный код команды: 1000 1001 1100 0010, команда занимает 2 байта

16-ричный код команды: 89С2h.

*MOV AL, pam1 ; из памяти в регистр*

Машинный код команды: 1010 0000 1000 0000 0001 0101, команда занимает 3 байта

16-ричный код команды: A08015h.

*MOV AX, 0B800h ; непосредственное значение в регистр*

Машинный код команды: 1011 1000 0000 0000 1011 1000, команда занимает 3 байта

16-ричный код команды: B800B8h.

*MOV ES, AX ; из РОН в сегментный регистр*

Машинный код команды: 1000 1110 1100 0000, команда занимает 2 байта

16-ричный код команды: 8EC0h.

*MOV byte ptr pam2+1, BH ; из регистра в память*

Машинный код команды: 1000 1000 0011 1110 0010 0001 0001 1010 команда занимает 4 байта

16-ричный код команды: 883E211Ah.

*MOV word ptr pam3, 5000h ; непосредственное значение в память*

Машинный код команды: 1100 0111 0000 0110 1111 0000 0001 1111 0000 0000 0101 0000, команда занимает 6 байт.

16-ричный код команды: C706F01F0050h.

*MOV SI, pam2 ; из памяти в регистр*

Машинный код команды: 1000 1011 0011 0110 0010 0000 0001 1010, команда занимает 4 байта

16-ричный код команды: 8B36201Ah.

**CMP – CoMPare two operands – сравнение двух операндов.**

**CMP *операнд1, операнд2***

### Действие: *операнды операнд1 и операнд2 сравниваются методом вычитания, при этом сами операнды не изменяются.*

Флаги: OF, SF, ZF, AF, PF, CF

|  |  |  |  |
| --- | --- | --- | --- |
| 16-ричный код (1 байт) | MOD Reg/OPC Reg/Mem (2-ой байт) | смещение  disp\_Lo, disp\_Hi | формат операндов:  приемник, источник |
| 38 | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg8/Mem8, Reg8 |
| 39 | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg16/Mem16, Reg16 |
| 3A | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg8, Reg8/Mem8 |
| 3B | MOD Reg/OPC Reg/Mem | Disp\_Lo, Disp\_Hi | Reg16, Reg16/Mem16 |
| 3C | Data8 (непосредств.операнд) | отсутствует | AL, Immed8 |
| 3D | Data16 (непосред.операнд) | отсутствует | AX, Immed16 |
| 80 | MOD **111** Reg/Mem | Disp\_Lo, Disp\_Hi  Data Lo | Reg8/Mem8, Immed8 |
| 81 | MOD **111** Reg/Mem | Disp\_Lo, Disp\_Hi  Data Lo, Data Hi | Reg16/Mem16, Imm16 |
| 83 | MOD **111** Reg/Mem | Disp\_Lo, Disp\_Hi  Data SX | Reg16/Mem16, Imm8 |

**111** - расширение кода операции при работе с непосредственным операндом, размещается в поле Reg/Opc в байте адресации.

Команда CMP выполняет вычитание *операнд1 - операнд2*. В соответствии с результатом вычитания устанавливаются флаги. Сами операнды не изменяются ! Нельзя использовать в качестве операндов сегментные регистры, определять оба операнда как ячейки памяти. Операнды могут быть байтами или словами и представлять числа со знаком и без знака. Обычно вслед за командой CMP стоит одна из команд условных переходов, анализирующая состояние флагов процессора.

**Например,** 1) *CMP AX, 10000 ; AX-10000*

*JE L1 ;переход на метку L1, если AX=10000*

*2) MOV DX, 0AF00h ; DX=0AF00h, CF=0, PF=0, AF=1, ZF=0,*

*CMP DX, 8002h ; SF=0, OF=0*

**ЗАДАНИЕ 1.** Записать машинные коды команд, дать их 16-ричное представление, если

;в сегменте данных:

op1 db 34h ; размещается в сегменте по смещению 0004h

op2 dw 0A10h

mas1 db 5, 0, 117, 30

mas2 dw 240, 550, 1034

;в программном сегменте:

|  |  |  |  |
| --- | --- | --- | --- |
| ***№ вар.*** | ***Команды*** | | |
| 1,2,3 | MOV BL, [SI] | MOV CX, 3212 | MOV op2, DS |
| 4,5,6 | MOV DS,SI | MOV [BX][DI+05h], AX | MOV DX, ES |
| 7,8,9 | MOV AL, 30h | MOV mas1[SI], 25h | MOV ES, op2 |
| 10,11,12 | MOV byte ptr mas2+4, DL | MOV SS, DI | MOV DH, [BP+4520] |
| 13,14,15 | MOV SS, mas2 | MOV DL, 9 | MOV [DI+4]mas1, CL |
| 16,17,18 | MOV DH, 17 | MOV BX, SI | MOV DS, mas2+2 |
| 19,20,21 | MOV AL, 4 | MOV ES, AX | MOV word ptr [DI], 0Ah |
| 22,23,24 | MOV SS, DX | MOV BX, 0A4Fh | MOV AL, [BX+920] |
| 25,25,27 | MOV word ptr [DI], AX | MOV word ptr [DI], SS | MOV AL, 0AAh |
| 28,29,30 | MOV DS, AX | MOV DX, 0ABh | MOV AX, 0AAh |

**ЗАДАНИЕ 2.** Получить машинные коды команд, выполнить их и установить флаги.

|  |  |  |  |
| --- | --- | --- | --- |
| ***№ вар.*** | ***Команды*** | | |
| 1,2,3 | CMP DH, 17 | CMP BX, SI | CMP word ptr [DI], 0DBBh (3) |
| 4,5,6 | CMP AL, 4 | CMP SI, AX | CMP word ptr [DI], 0Ah (4) |
| 7,8,9 | CMP CX, DX | CMP BX, 0A4Fh | CMP AL, [BX+920] |
| 10,11,12 | CMP [DI], AX (1) | CMP AH, 5 | CMP AL, 0AAh |
| 13,14,15 | CMP SI, AX | CMP DX, 0ABh | CMP AX, 0AAh |
| 16,17,18 | CMP BX, [BP+3Fh] (2) | CMP DX, [BX+SI] (5) | CMP BL, 5[DI] |
| 19,20,21 | CMP BL, BH | CMP CX, 0F0Dh | CMP 34[DI], BL (5) |
| 22,23,24 | CMP DL, 37 | CMP DX, DI | CMP SI, 0ABh |
| 25,26,27 | CMP BL, 8 | CMP SI, 85h | CMP byte ptr 5[SI], 0FFh (4) |
| 28,29,30 | CMP AX, CX | CMP DX, 0AFh | CMP CL, [BX+DI] |

***Содержимое регистров:***

(AX) → 88BAh (SI) → 00DDh

(BX) → 5F5Eh (DI) → 753Ah

(CX) → 3399h (BP) → 0016h

(DX) → F070h

***Содержимое ячеек памяти:***

1. → 6Bh
2. → E10Fh
3. → 5F7Fh
4. → 0000h
5. → F5h

**КОНТРОЛЬНЫЕ ВОПРОСЫ.**

1. Какие группы команд вы можете назвать в системе команд процессора?
2. Что такое мнемоническая запись команды?
3. Какова структура команды на языке ассемблера?
4. Какие поля в структуре команды не являются обязательными?
5. Каким образом можно записывать числовые значения в сегментные регистры?
6. Почему команды пересылки имеют большее количество кодов по сравнению с остальными командами?
7. Объясните, каким образом выполняется команда сравнения кодов?
8. Влияет ли команды пересылки на флаги, а команда сравнения кодов?
9. Как осуществляется расширение восьмибитного операнда до размеров 2-х байт?
10. Какие режимы адресации возможны в командах пересылки?

### ПРИЛОЖЕНИЯ

### П.1 *Таблица 1 Кодирование регистров (при MOD = 11)*

|  |  |  |  |
| --- | --- | --- | --- |
| REG | Разрядность операндов | | |
| 8-миразрядные | 16-тиразрядные | 32-хразрядные  (для CPU i80386+) |
| 000 | AL | AX | EAX |
| 001 | CL | CX | ECX |
| 010 | DL | DX | EDX |
| 011 | BL | BX | EBX |
| 100 | AH | SP | ESP |
| 101 | CH | BP | EBP |
| 110 | DH | SI | ESI |
| 111 | BH | DI | EDI |

# П.2 *Таблица 2 Формирование 16-тиразрядного исполнительного адреса EA операнда*

|  |  |  |  |
| --- | --- | --- | --- |
| REG/MEM | MOD | | |
| 00 | 01 | 10 |
| 000 | DS:[BX+SI] | DS:[BX+SI+d8] | DS:[BX+SI+d16] |
| 001 | DS:[BX+DI] | DS:[BX+DI+d8] | DS:[BX+DI+d16] |
| 010 | SS:[BP+SI] | SS:[BP+SI+d8] | SS:[BP+SI+d16] |
| 011 | SS:[BP+DI] | SS:[BP+DI+d8] | SS:[BP+DI+d16] |
| 100 | DS:[SI] | DS:[SI+d8] | DS:[SI+d16] |
| 101 | DS:[DI] | DS:[DI+d8] | DS:[DI+d16] |
| 110 | DS:[d16] | SS:[BP+d8] | SS:[BP+d16] |
| 111 | DS:[BX] | DS:[BX+d8] | DS:[BX+d16] |