x86 Instruction Set Reference

MOVS/MOVSB/MOVSW/MOVSD

Move Data from String to String

Opcode	Mnemonic	Description
A4	MOVS m8, m8	Move byte at address DS:(E)SI to address ES:(E)DI.
A5	MOVS m16, m16	Move word at address DS:(E)SI to address ES:(E)DI.
A5	MOVS m32, m32	Move doubleword at address DS:(E)SI to address ES:(E)DI.
A4	MOVSB	Move byte at address DS:(E)SI to address ES:(E)DI.
A5	MOVSW	Move word at address DS:(E)SI to address ES:(E)DI.
A5	MOVSD	Move doubleword at address DS:(E)SI to address

Description

Moves the byte, word, or doubleword specified with the second operand (source operand) to the location specified with the first ope operand). Both the source and destination operands are located in memory. The address of the source operand is read from the DS registers (depending on the address-size attribute of the instruction, 32 or 16, respectively).

The address of the destination operand is read from the ES:EDI or the ES:DI registers (again depending on the address-size attribute The DS segment may be overridden with a segment override prefix, but the ES segment cannot be overridden.

At the assembly-code level, two forms of this instruction are allowed: the "explicit-operands" form and the "no-operands" form. The expl (specified with the MOVS mnemonic) allows the source and destination operands to be specified explicitly. Here, the source and des should be symbols that indicate the size and location of the source value and the destination, respectively. This explicit-operands form is documentation; however, note that the documentation provided by this form can be misleading.

That is, the source and destination operand symbols must specify the correct type (size) of the operands (bytes, words, or doublewords have to specify the correct location.

The locations of the source and destination operands are always specified by the DS:(E)SI and ES:(E)DI registers, which must be loade the move string instruction is executed.

The no-operands form provides "short forms" of the byte, word, and doubleword versions of the MOVS instructions. Here also DS:(E)SI assumed to be the source and destination operands, respectively. The size of the source and destination operands is selected with the m (byte move), MOVSW (word move), or MOVSD (doubleword move).

After the move operation, the (E)SI and (E)DI registers are incremented or decremented automatically according to the setting of the DF fl register. (If the DF flag is 0, the (E)SI and (E)DI register are incremented; if the DF flag is 1, the (E)SI and (E)DI registers are decrement are incremented or decremented by 1 for byte operations, by 2 for word operations, or by 4 for doubleword operations.

The MOVS, MOVSB, MOVSW, and MOVSD instructions can be preceded by the REP prefix (see "REP/REPE/REPZ/REPNE /REP Operation Prefix" in Chapter 4) for block moves of ECX bytes, words, or doublewords.

ES:(E)DI.

Operation

```
Destination = Source;
if(IsByteMove()) {
        if(DF == 0) {
                (E)SI = (E)SI + 1;
                (E)DI = (E)DI + 1;
        }
        else {
                (E)SI = (E)SI - 1;
                (E)DI = (E)DI - 1;
        }
}
else if(IsWordMove()) {
        if(DF == 0) {
                (E)SI = (E)SI + 2;
                (E)DI = (E)DI + 2;
        }
        else {
                (E)SI = (E)SI - 2;
                (E)DI = (E)DI - 2;
        }
}
else { //doubleword move
        if(DF == 0) {
                (E)SI = (E)SI + 4;
                (E)DI = (E)DI + 4;
        }
        else {
                (E)SI = (E)SI - 4;
                (E)DI = (E)DI - 4;
        }
}
```

Flags affected
None.

Protected Mode Exceptions	
#GP(0)	If the destination is located in a non-writable segment. If a memory operand effective address is outside the CS, DS, ES, FS, limit. If the DS, ES, FS, or GS register contains a null segment selector.
#GP(0)	If the destination is located in a non-writable segment. If a memory operand effective address is outside the CS, DS, ES, FS, limit. If the DS, ES, FS, or GS register contains a null segment selector.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#PF(fault-code)	If a page fault occurs.

Real-Address Mode Exceptions	
#GP	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#GP	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.

Virtual-8086 Mode Exceptions	
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#PF(fault-code)	If a page fault occurs.

Instruction	Latency	Throughput	Execution Unit
CPUID	0F3n/0F2n	0F3n/0F2n	0F2n
MOVSB/MOVSW	1/0.5	0.5/0.5	ALU