



# **String Instructions**

- **String instructions were designed to operate on large data structures.**
- **The SI and DI registers are used as pointers to the data structures being accessed or manipulated.**
- **The operation of the dedicated registers stated above are used to simplify code and minimize its size.**



# String Instructions

- The registers(DI,SI) are automatically incremented or decremented depending on the value of the direction flag:
  - DF=0, increment SI, DI.
  - DF=1, decrement SI, DI.
- To set or clear the direction flag one should use the following instructions:
  - CLD to clear the DF.
  - STD to set the DF.



# String Instructions

- The REP/REPZ/REPNZ prefixes are used to repeat the operation it precedes.
- String instructions we will discuss:
  - LODS
  - STOS
  - MOVS
  - CMPS
  - SCAS



# **LODS/LODSB/LODSW/LODSD**

- **Loads the AL, AX or EAX registers with the content of the memory byte, word or double word pointed to by SI relative to DS. After the transfer is made, the SI register is automatically updated as follows:**
  - SI is incremented if DF=0.
  - SI is decremented if DF=1.



# LODS/LODSB/LODSW/LODSD

## ■ Examples:

- **LODSB**

**AL=DS:[SI]; SI=SI ± 1**

- **LODSW**

**AX=DS:[SI]; SI=SI ± 2**

- **LODSD**

**EAX=DS:[SI]; SI=SI ± 4**

- **LODS MEAN**

**AL=DS:[SI]; SI=SI ± 1 (if MEAN is a byte)**

- **LODS LIST**

**AX=DS:[SI]; SI=SI ± 2 (if LIST is a word)**

- **LODS MAX**

**EAX=DS:[SI]; SI=SI ± 4 (if MAX is a double word)**

# LODS/LODSB/LODSW/LODSD

## Example

Assume:

Location	Content
Register SI	500H
Memory location 500H	'A'
Register AL	'2'

After execution of **LODSB**

If **DF=0** then:

Location	Content
Register SI	501H
Memory location 500H	'A'
Register AL	'A'

Else if **DF=1** then:

Location	Content
Register SI	4FFH
Memory location 500H	'A'
Register AL	'A'



# **STOS/STOSB/STOSW/STOSD**

- **Transfers the contents of the AL, AX or EAX registers to the memory byte, word or double word pointed to by DI relative to ES. After the transfer is made, the DI register is automatically updated as follows:**
  - **DI is incremented if DF=0.**
  - **DI is decremented if DF=1.**



# STOS/STOSB/STOSW/STOSD

## ■ Examples:

- STOSB

ES:[DI]=AL; DI=DI  $\pm$  1

- STOSW

ES:[DI]=AX; DI=DI  $\pm$  2

- STOSD

ES:[DI]=EAX; DI=DI  $\pm$  4

- STOS MEAN

ES:[DI]=AL; DI=DI  $\pm$  1 (if MEAN is a byte)

- STOS LIST

ES:[DI]=AX; DI=DI  $\pm$  2 (if LIST is a word)

- STOS MAX

ES:[DI]=EAX; DI=DI  $\pm$  4 (if MAX is a double word)





# STOS/STOSB/STOSW/STOSD

## Example

Assume:

Location	Content
Register DI	500H
Memory location 500H	'A'
Register AL	'2'

After execution of STOSB

If DF=0 then:

Location	Content
Register DI	501H
Memory location 500H	'2'
Register AL	'2'

Else if DF=1 then:

Location	Content
Register DI	4FFH
Memory location 500H	'2'
Register AL	'2'



# **MOVS/MOVSB/MOVSW/MOVSDB**

- **Transfers the contents of the the memory byte, word or double word pointed to by SI relative to DS to the memory byte, word or double word pointed to by DI relative to ES. After the transfer is made, the DI register is automatically updated as follows:**
  - **DI is incremented if DF=0.**
  - **DI is decremented if DF=1.**

# MOVS/MOVS<sub>B</sub>/MOVSW/MOVS<sub>D</sub>

## ■ Examples:

- MOVS<sub>B</sub>

ES:[DI]=DS:[SI]; DI=DI ± 1; SI=SI ± 1

- MOVSW

ES:[DI]= DS:[SI]; DI=DI ± 2; SI=SI ± 2

- MOVS<sub>D</sub>

ES:[DI]=DS:[SI]; DI=DI ± 4; SI=SI ± 4

- MOVS MEAN

ES:[DI]=DS:[SI]; DI=DI ± 1; SI=SI ± 1 (if MEAN is a byte)

- MOVS LIST

ES:[DI]=DS:[SI]; DI=DI ± 2; SI=SI ± 2 (if LIST is a word)

- MOVS MAX

ES:[DI]=DS:[SI]; DI=DI ± 4; SI=SI ± 4 (if MAX is a double word)

# MOVS/MOVSB/MOVSW/MOVSDB

## Example

Assume:

Location	Content
Register SI	500H
Register DI	600H
Memory location 500H	'2'
Memory location 600H	'W'

After execution of MOVSB

If DF=0 then:

Location	Content
Register SI	501H
Register DI	601H
Memory location 500H	'2'
Memory location 600H	'2'

Else if DF=1 then:

Location	Content
Register SI	4FFH
Register DI	5FFH
Memory location 500H	'2'
Memory location 600H	'2'



# **CMPS/CMPSB/CMPSW/CMPSD**

- **Compares the contents of the the memory byte, word or double word pointed to by SI relative to DS to the memory byte, word or double word pointed to by DI relative to ES and changes the flags accordingly. After the comparison is made, the DI and SI registers are automatically updated as follows:**
  - **DI and SI are incremented if DF=0.**
  - **DI and SI are decremented if DF=1.**



# **SCAS/SCASB/SCASW/SCASD**

- **Compares the contents of the AL, AX or EAX register with the memory byte, word or double word pointed to by DI relative to ES and changes the flags accordingly. After the comparison is made, the DI register is automatically updated as follows:**
  - **DI is incremented if DF=0.**
  - **DI is decremented if DF=1.**



# REP/REPZ/REPNZ

- These prefixes cause the string instruction that follows them to be repeated the number of times in the count register ECX or until:
  - ZF=0 in the case of REPZ (repeat while equal).
  - ZF=1 in the case of REPNZ (repeat while not equal).



# **REP/REPZ/REPNZ**

- **Use REPNE and SCASB to search for the character 'f' in the buffer given below.**
- **BUFFER DB 'EE3751'**
- **MOV AL,'f'**
- **LEA DI,BUFFER**
- **MOV ECX,6**
- **CLD**
- **REPNE SCASB**
- **JE FOUND**





# REP/REPZ/REPNZ

- Use REPNE and SCASB to search for the character '3' in the buffer given below.
- BUFFER DB 'EE3751'
- MOV AL,'f'
- LEA DI,BUFFER
- MOV ECX,6
- CLD
- REPNE SCASB
- JE FOUND