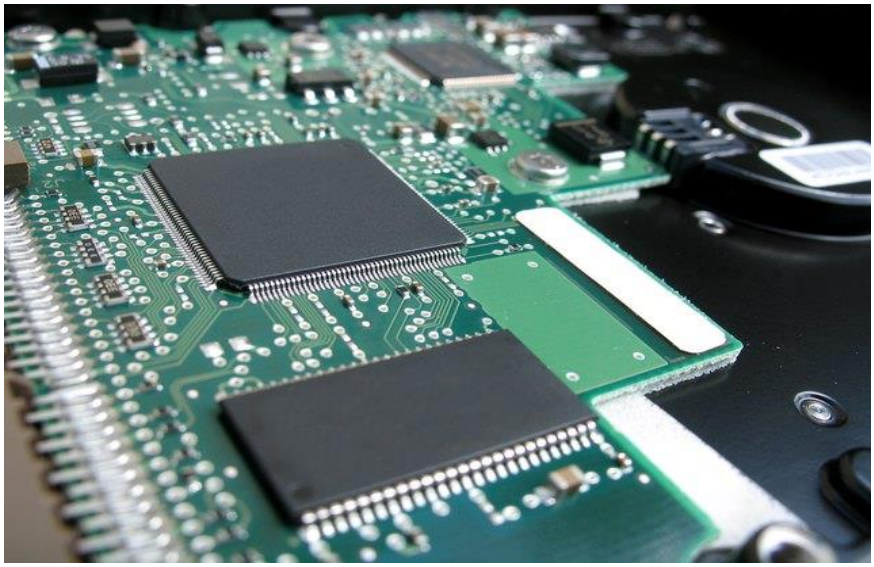
جامعة الحمدانية
كلية التربية

Lecture 18

Microprocessors



Prepared by:

Firas Abdulrahman Yosif

String Instructions

The string instructions operate on strings of bytes. Operations include storing strings in memory, loading strings from memory, comparing strings, and scanning strings for substrings.

- 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.

عمل تعليمات السلسلة على سلاسل من البايت. تشمل العمليات تخزين السلاسل في الذاكرة ، وتحميل السلاسل من الذاكرة ، ومقارنة السلاسل ، ومسح السلاسل بحثًا عن السلاسل الفرعية.

• تم تصميم تعليمات السلاسل لتعمل على هياكل البيانات الكبيرة.

• يتم استخدام سجلات SI و DI كمؤشرات لهياكل البيانات التي يتم الوصول إليها أو معالجتها.

• يتم استخدام تشغيل السجلات المخصصة المذكورة أعلاه لتبسيط التعليمات البرمجية وتقليل حجمها.

- 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.

ملاحظة:

- ❖ زيادة مستمرة في عنوان SI, DI (clear direction flag) CLD: DF=0
- ❖ نقصان مستمر في عنوان SI, DI (set direction flag) STD: DF= 1

- **String instructions include:**

❖ MOVS

❖ CMPS

❖ SCAS

❖ STOS

❖ LODS

Now we discuss these instructions in detail:

1. **MOVS:**

A. MOVS B (نسخ بايت بين ذاكرتين بشكل مباشر)

ES: [DI] ⟵ DS: [SI]

SI ⟵ SI-1

DI ⟵ DI-1 } STD

SI ⟵ SI-1

DI ⟵ DI-1 } CLD

B. MOVS W (نسخ بايتين بين ذاكرتين)

ES: [DI] ⟵ DS: [SI]

ES: [DI+1] ⟵ DS: [SI+1]

SI ← SI-1
SI ← SI-2 } STD
DI ← DI-1
DI ← DI-2

OR

SI ← SI+1
SI ← SI+2 } CLD
DI ← DI+1
DI ← DI+2

2. CMPS (مقارنة بين ذاكرتين ومحتويات الذاكرة لاتتأثر)

A. CMPSB (مقارنة بايت بين ذاكرتين)

Format) **DS: [SI] – ES: [DI]**

DI ← DI-1
SI ← SI-1 } STD

OR

DI ← DI+1
SI ← SI+1 } CLD

B. CMPSW (مقارنة بين بايتين في كل ذاكرتين)

Format
DS: [SI] – ES: [DI]
DS: [SI+1] – ES: [DI+1]

DI ← DI-1
DI ← DI-2
SI ← SI-1
SI ← SI-2
} STD

DI ← DI+1
DI ← DI+2
SI ← SI+1
SI ← SI+2
} CLD

3. SCAS (مقارنة ذاكرة مع سجل AX او مع AL)

A. SCAB (مقارنة بين بايت في ذاكرة مع سجل AL)

AL – ES : [DI]

DI ← DI+1 } CLD

DI ← DI-1 } STD

B. SCAW (مقارنة بين بايتين في الذاكرة مع سجل AX)

AX – ES : [DI]

DI ← DI+1
DI ← DI+2
} CLD

DI ← DI-1
DI ← DI-2
} STD

4. <u>STOS</u> : (تحميل محتويات AX او AL الى الذاكرة)

A. STOSB (تحميل محتويات AL الى الذاكرة)

ES: [DI] ← AL

DI ← DI+1 } CLD

DI ← DI-1 } STD

B. STOSW (تحميل محتويات AX الى الذاكرة)

ES : [DI] ← AX

DI ← DI+1
DI ← DI+2
} CLD

DI ← DI-1
DI ← DI-2
} STD

3. **LODS** : (تحويل محتويات ذاكرة الى سجل AL او AX)

ملاحظة: السجل DI لايستخدم ومحتويات AL و AX لاتتغير.

A. LODS B

AL ← DS: [SI]

(CLD) SI ← SI+1

(STD) SI ← SI – 1

B. LODS W

AX ← DS: [SI]

SI ← SI+1 } CLD

SI ← SI+2

or

SI ← SI – 1 } STD

SI ← SI – 2

EX1)

```
CLD
MOVSB
HLT
```

⟺

```
MOV AL, [SI]
ES: MOV[DI], AL
INC SI
INC DI
HLT
```

EX1) Write a program in assembly language to transfer 10 bytes from location in DS to location his address 2010 in ES, if DS=C00H and ES=D00H (using string instruction).

Sol.)

```
CLD

MOV AX, C00H

MOV DS, AX

MOV BX, D00H

MOV ES, BX

DS: MOV SI, 1000H

ES: MOV DI, 2010H

MOV CX, A

**: MOVESB

DEC CL

JNZ **

HLT
```

EX2) Write a program in assembly language to put 50h in ten locations of memory using string instruction (2100: 100) in ES, if DF=0.

Sol.)

```
        CLD
        MOV DI, 100H
        MOV AX, 2100H
        MOV ES, AX
        MOV AL, 50H
        MOV CL, 0AH
Loop: STOSB ;          (ES: [DI] ←—— AL )
        DEC CL
        JNZ Loop
        HLT
```

EX3) Write a program in assembly language to search for number 32H in a string that begins with address 102H and 10 bytes long using string instructions.

```
CLD
MOV AL, 32H
MOV CL, A
ES: MOV DI, 102H
X1: SCASB;      (AL- ES:[DI])
JNZ X2
INC DI
X2: DEC CL
JNZ X1;      (ZF ≠ 0 اذا اقفز)
HLT
```

EX4) Gives examples for these commands:

LODSW ⟺ DS:MOV AX, [SI]

MOVSW ⟺
- MOV AL, [SI]
- ES: MOV [DI], AL

CMPSB ⟺
- DS: MOV AL, [SI]
- ES: CMP AL, [DI]

SCASW ⟺ ES: CMP AX. [DI]

STOSB ⟺ ES: MOV [DI], AL

ملاحظة: وضع ES: معنى هذا ان [DI] يتعامل مع ES وليس مع DS .