

Laboratory Manual

for

Computer Organization and Assembly Language

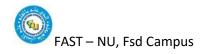
Course Instructors

Lab Instructor(s)

Section

Semester

Department of Computer Science



COAL Lab 13 Manual

Objectives:

- String Instructions
- Direction Flag and REP Prefix
- Problems & Assignments

13.1 String Instructions:

String instructions perform following functions:

Instruction	Description
MOVSB, MOVSW, MOVSD	Move string data: copy data from memory addressed by ESI to memory addressed by EDI .
LODSB, LODSW, LODSD	Load accumulator from string: Load memory addressed by ESI into the accumulator.
STOSB, STOSW, STOSD	Store string data: Store the accumulator contents into memory addressed by EDI.
SCASB, SCASW, SCASD	Subtract the contents of accumulator register into memory location addressed by DI
CMPSB, CMPSW	Compare the contents of memory addressed by SI to memory addressed by DI

Table 13.1: String Instructions

Direction Flag:

It is a control flag and works in accordance with following instructions:

Format	Description	Mode	Effect
CLD		Auto Increment	SI→ SI+1
CLD	Clear DF, i.e DF=0	Auto increment	$DI \rightarrow DI + 1$
CITID			SI →SI – 1
STD	Set DF, i.e: DF=1	Auto Decrement	$DI \rightarrow DI - 1$

Table 13.2: Direction Flag

EL 213 – Lab 13 Manual Page 2



Instruction	Mnemonic	DF	Oper	ation	Flag affected	Example Comments
	MOVSB	0	ES:EDI ← DS:ESI	; ESI ← ESI + 1 ; EDI ← EDI + 1		EX # 1: Moving first two bytes of string . DATA
		1	ES:EDI ← DS:ESI	; ESI ← ESI - 1 ; EDI ← EDI - 1		String1 BYTE 'FAST-NU' String2 BYTE 5 DUP(?)
	MOZZGE	0	ES:EDI ← DS:ESI	; ESI ← ESI + 2 ; EDI ← EDI + 2		.CODE LEA ESI,String1
Move string	MOVSW	1	ES:EDI ← DS:ESI	; ESI ← ESI - 2 ; EDI ← EDI - 2	None	LEA EDI, String2 CLD ; DF = 0
	MOVSD	0	ES:EDI ← DS:ESI	; ESI ← ESI + 4		MOVSB ; move 1st byte i.e. H
		1	ES:EDI ← DS:ESI	; EDI \leftarrow EDI + 4 ; ESI \leftarrow ESI - 4 ; EDI \leftarrow EDI - 4		MOVSB ; move 2nd byte i.e. E ;String1 = FAST-NU ;String2 = FA?????
	LODSB	0	AL ← DS:ESI	$; ESI \leftarrow EDI + 1$		EX # 2: Loading 1st two bytes of string in AL
Load String	TODGH	1 0		; EDI ← EDI -1 : ESI ← EDI +2 No	None	LEA ESI, String1 CLD ; DF = 0
	LODSW	1 0	$AX \leftarrow DS:ESI$ $EAX \leftarrow DS:ESI$; EDI ← EDI - 2 ; ESI ← EDI + 4	- , , , , ,	LODSB ; AL = F LODSB ; AL = A
		0		; EDI ← EDI - 4 ; ESI ← EDI + 1		EX # 3: Storing two L's in String1
	STOSB	1	DS:ESI ← AL	; EDI ← EDI -1	None	LEA EDI, String1
Store String	STOSW	0 1	DS:ESI ← AX	; ESI \leftarrow EDI + 2 ; EDI \leftarrow EDI - 2		CLD ; DF = 0 MOV AL, 'L'
	STOSD	0	DS:ESI← EAX	; ESI ← EDI +4		STOSB ; String1 = LAST- NU
		1		; EDI ← EDI - 4		STOSB ; String1 = LLST- NU
	SCASB	0	DS:ESI AL	; $ESI \leftarrow EDI + 1$		EX # 4: Search 'N' in 1st two bytes of String1
Scan String	SCASW	1 0 1	DS:ESI AX	; EDI \leftarrow EDI -1 ; ESI \leftarrow EDI +2 ; EDI \leftarrow EDI -2	CF, AF, PF, ZF,	LEA DI, String2-1 STD; DF = 1 MOV AL, 'N'
Scan Sunig		0	DS:ESI EAX	; ESI \leftarrow EDI + 4	SF, OF	SCASB ; Scan 1st byte ZF=0
	SCASD	1		; EDI ← EDI - 4		SCASB ;Scan 2nd byte ZF=1
		0	ES:EDI DS:ESI	; ESI ← ESI + 1		EX # 5: Compare 1st two bytes of given Strings
Compare	CMPSB			; EDI \leftarrow EDI + 1 ; ESI \leftarrow ESI - 1	CF, AF, PF, ZF,	.DATA String3 DB 'ACD'
String		1	ES:EDI DS:ESI	; EDI \leftarrow EDI - 1 ; ESI \leftarrow ESI + 2	SF, OF	String4 DB 'ABC'
	CMPSW	0	ES:EDI DS:ESI	; EDI \leftarrow EDI + 2		LEA SI, String3

EL 213 – Lab 13 Manual Page 3



			1	ES:EDI DS:ESI	; ESI ← ESI - 2	LEA DI, String4	
			ES:EDI DS:ESI ; ED	; EDI ← EDI - 2	CLD	; $DF = 0$	
	C	CMPSD	0	ES:EDI DS:ESI	; ESI ← ESI + 4	CMPSB	;Comp 1st byte ZF=1
					; EDI \leftarrow EDI + 4	CMPSB	;Comp 2nd byte ZF=0
		CMF 5D	1	ES:EDI DS:ESI	; ESI ← ESI - 4		
			I ES:ED	E9:ED1 D9:E91	; EDI ← EDI - 4		

Table 13.3: String Instructions with Examples

REP Prefix:

Prefix	Description	Examples
REP	Repeat while CX>0	EX#6: Comparing two given Strings .DATA String5 DB 'HELLO' St5_Len = \$ - String5 String6 DB 'WORLD' .CODE LEA SI, String5 LEA DI, String6 MOV CX, St5_Len CLD ; DF = 0 REP CMPSB; Compare string bytes while CX > 0
REPZ, REPE	Repeat while ZF=1 and CX>0	See Yourself
REPNZ, REPNE	Repeat while ZF=0 and CX>0	See Yourself

Table 13.4: REP Instruction with example

Estimated completion time: 1 hr, 30 mins

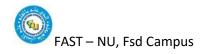
Problem(s) / Assignment(s)

Discussion & Practice

PROBLEM 13.1: SEARCHING Write a program that performs following tasks using String instructions only:	Estimated completion time:30mins
1. Prompt the user to enter a string terminated by Enter using STR_IN procedure.	
2. Search for vowels and consonants in the string using SEARCH procedure.	

EL 213 – Lab 13 Manual Page 4

3. Display vowels and consonants in alphabetical order using



DIS STR.

PROBLEM 13.2: *Finding relative prime numbers*

Estimated completion time:30mins

Harry wants to check the relative primality of 2 numbers. For this purpose, he checks the GCD (Greatest Common Divisor) of the numbers. If GCD comes out 1 then numbers are relative prime to each other. Harry requirements are as follows:

- 1. Procedure **DEC_IN** should load two registers (BX and DX) with two numbers. Numbers should be a 2 digit decimal ranging from (01 99).
- 2. Procedure GCD_AB apply the logic for GCD of two numbers. GCD of two numbers is performed by dividing the greater number (in BX) by the smaller number (in DX) till the remainder is zero. If it is zero, the divisor is the GCD if not the remainder and the divisor of the previous division are the new set of two numbers. The process is repeated by dividing greater of the two numbers by the smaller number till the remainder is zero and GCD is found.
- 3. Also check if,
 - a. The numbers are equal then GCD would be BX,
 - b. BX<DX then exchange the contents of 2 register.
- 4. Procedure **DEC_OUT** should display the GCD on screen in decimal.

Sample 1:

```
Enter 1<sup>st</sup> Number: 20
Enter 2<sup>nd</sup> Number: 09
GCD is: 1
```

Numbers are relative prime

Sample 2:

```
Enter 1<sup>st</sup> Number: 09
Enter 2<sup>nd</sup> Number: 03
GCD is: 03
```

Numbers are not relative prime

EL 213 – Lab 13 Manual Page 5