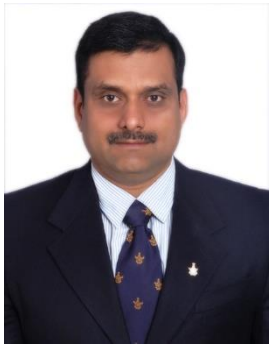
Microprocessors and Interfaces: 2021-22
Lab 7
ALP for Reversal of String

By Dr. Sanjay Vidhyadharan



ALPs to be completed

Problem 1: Write a program to reverse the given string and store at the same location.

Problem 2: Write a program using the LOOP instruction with indirect addressing that copies a string from source to target, reversing the character order in the process.

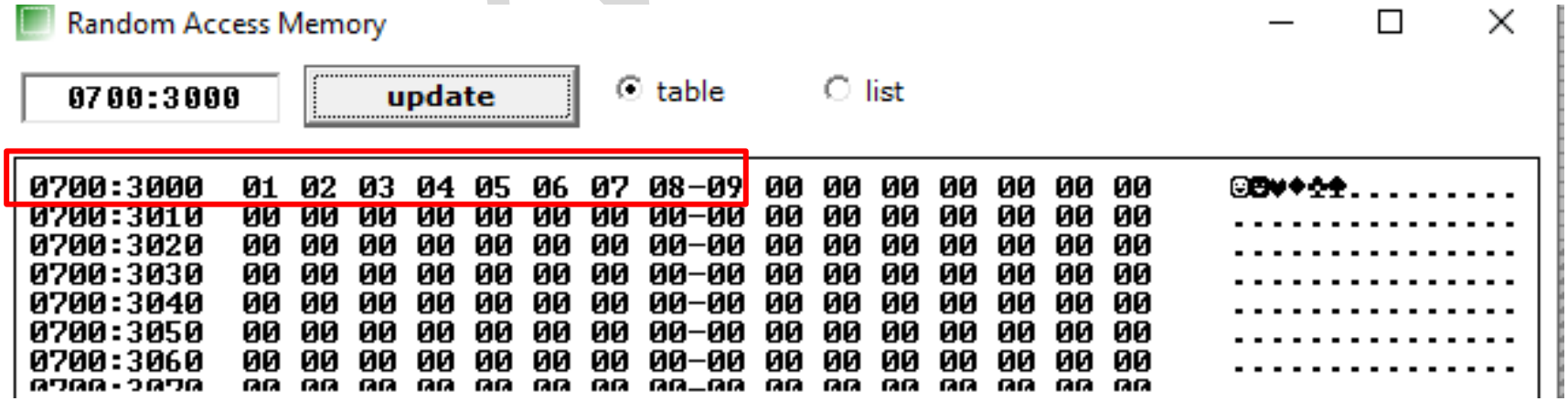
7.1 Reversal of String

Objective: Reversal of string.

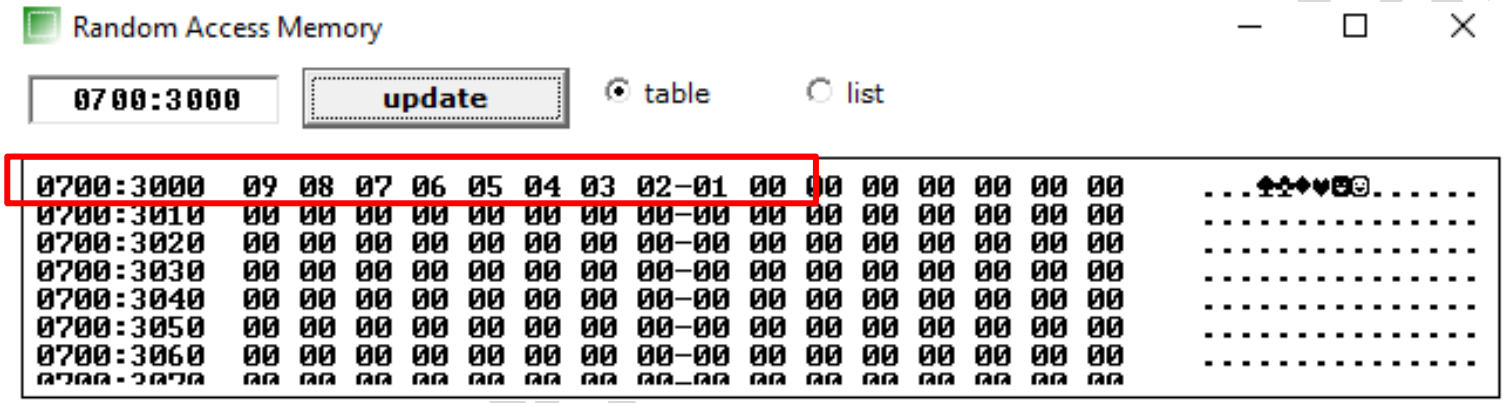
Restrictions: No other memory location to be used.

Solution: Use of XCHG command.

Example:



7.1 Reversal of String



7.1 Reversal of String: Pseudocode


```
org 100h

MOV SI,xxxxH
MOV CL,xxH
MOV CH,00H
MOV DI,yyyyH
L1: MOV AL,[SI]
    MOV BL,[DI]
    XCHG [xx],BL
    XCHG [yy],AL
    INC SI
    DEC DI
    INC zz
    DEC ww
    CMP CL,CH
    JG xx
    HLT

ret
```

```
Assume: SI = 4000h
        DI = Dependent on length of string

Data string: 1, 2, 3, 4, 5, 6, 7h
```

Change here to complete the code.

```
CL,CH - 07,00
1-7
CL,CH - 06,01
2-6
CL,CH - 05,02
3-5
CL,CH - 04,03
4-4
CL,CH - 03,04
```

```
CL,CH - 08,00
1-8
CL,CH - 07,01
2-7
CL,CH - 06,02
3-6
CL,CH - 05,03
4-5
CL,CH - 04,04
```

7.1 Reversal of String: Pseudocode


```
org 100h

MOV SI,xxxxH
MOV CL,xxH
MOV CH,00H
MOV DI,yyyyH
L1: MOV AL,[SI]
    MOV BL,[DI]
    XCHG [xx],BL
    XCHG [yy],AL
    INC SI
    DEC DI
    INC zz
    DEC ww
    CMP CL,CH
    JG xx
    HLT

ret
```

```
Assume: SI = 4000h
        DI = Dependent on length of string

Data string: 1, 2, 3, 4, 5, 6, 7h
```

Change here to complete the code.

```
CL,CH - 06,00
1-7
CL,CH - 05,01
2-6
CL,CH - 04,02
3-5
CL,CH - 03,03
```

```
CL,CH - 07,00
1-8
CL,CH - 06,01
2-7
CL,CH - 05,02
3-6
CL,CH - 04,03
4-5
CL,CH - 03,04
```

Review Questions

1. Repeat the problem with data ranging from 00h to 0Fh.

2. What is the role of JGE instruction?

3. Which addressing mode is used here?

7.2 Reversing Character Order


```
org 100h

MOV SI,xxxxh
MOV DI,yyyyh
MOV CX,zzzzh
L1: cld
    lodsb
    std
    stosb
    loop zz

Hlt

ret
```

```
Assume: SI = 3000h
        DI = 2000h

Data to be reversed:
00,01,02,03,04
```

Change here to complete the code.

Review Questions

1. Repeat the problem with data ranging from 00h to 0Fh

2. What are the roles of CLD and STD instructions?

7.1 Pseudocode: Method 2

Algorithm

1. Create a string like: “I am a student of BITS-Pilani”
2. Traverse through the string
3. Push the characters in the stack
4. Count the number of characters
5. Load the staring address of the string
6. POP the top character of the stack until count is not equal to zero
7. Put the character and reduce the count and increase the address
8. Continue until the count is greater than zero
9. Load the effective address of the string in dx using LEA command
10. Print the sting by calling the interrupt with 9H in AH
11. The string must be terminated by ‘$’ sign

7.3 Home Assignment


```
org 100h

.DATA
  ; The string to be printed
STRING DB 'I am a student of BITS-Pilani', '$'

.CODE
MAIN PROC FAR
MOV AX,@DATA
MOV DS,AX

; call reverse function
CALL REVERSE

; load address of the string
LEA DX,STRING

; output the string
; loaded in dx
MOV AH, 09H
INT 21H  ; interrupt to exit
MOV AH, 4CH
INT 21H
```

Continued…

7.3 Home Assignment


```
MAIN ENDP
REVERSE PROC
    ; load the offset of
    ; the string
    MOV SI, OFFSET STRING
      ; count of characters of the;
    ;string
    MOV CX, 0H
    LOOP1:
    ; compare if this is;
    ;the last character
    MOV AX, [SI]
    CMP AL, '$'
    JE LABEL1
     ; else push it in the;
    ;stack
    PUSH [SI]
     ; increment the pointer;
    ;and count
    INC SI
    INC CX
```


Continued…

7.3 Home Assignment


```
JMP LOOP1
     LABEL1:
    ; again load the starting;
    ;address of the string
    MOV SI, OFFSET STRING
        LOOP2:
        ;if count not equal to zero
        CMP CX,0
        JE EXIT

        ; pop the top of stack
        POP DX

        ; make dh, 0
        XOR DH, DH

        ; put the character of the;
        ;reversed string
        MOV [SI], DX

        ; increment si and;
        ;decrement count
        INC SI
        DEC CX
        JMP LOOP2

    EXIT:
    ; add $ to the end of string
    MOV [SI],'$ '
    RET

REVERSE ENDP
END MAIN

ret
```

No need to initialize a memory.
Output will be shown automatically.

Thankyou