CSE 331L: Microprocessor Interfacing & Embedded System Lab

Faculty: DR. DIHAN MD. NURUDDIN HASAN

Instructor: Moin Shahriyar

EEE 332/ CSE 331

Lab 6

Topic: String

Topics to be covered in class today:

- Conditional Jumps/Unconditional Jumps
- Procedures
- Instructions:
  - CMP = Compare
  - AND/OR = Logic AND/OR operation
  - JZ = Jump if Zero
  - JNZ = Jump if not Zero
  - JMP =(Unconditional) Jump
  - INT = Interrupt

Instruction	Operands	Description

CMP	REG, memory
	memory, REG
	REG, REG
	memory, immediate
	REG, immediate
		Compare.
		
		Algorithm:
		
		operand1 - operand2
		
		Result is not stored anywhere, flags are set (OF, SF, ZF, AF, PF, CF) according to result.
		
		Example:
		
		MOV AL, 5
		MOV BL, 5
		CMP AL, BL; (AL = 5, ZF = 1 so equal!)
		
		RET

JZ	Label	Short Jump if Zero (equal). Set by CMP, SUB, ADD, TEST, AND, OR, XOR instructions.
		
		Algorithm:
		
		if ZF = 1 then jump (ZF=Zero Flag. So, ZF=1 means it is 0)
		
		Example:
		
		include 'emu8086.inc'
		
		ORG 100h
		
		MOV AL, 5
		CMP AL, 5
		JZ label1
		PRINT 'AL is not equal to 5.'
		JMP exit
		label1:
		PRINT 'AL is equal to 5.'
		exit:
		RET

JNZ	Label	Short Jump if NOT Zero (equal). Set by CMP, SUB, ADD, TEST, AND, OR, XOR instructions.
		
		Algorithm:
		
		if ZF = 0 then jump (ZF=Zero Flag. So, ZF=0 means it is 1[NOT ZERO])
		
		Example:
		
		include 'emu8086.inc'
		
		ORG 100h
		MOV AL, 5
		CMP AL, 5
		JNZ label1
		PRINT 'AL is equal to 5.'
		JMP exit
		label1:
		PRINT 'AL is not equal to 5.'
		exit:
		RET

<table>
<tr>
<td>JMP</td>
<td>Label</td>
<td>Unconditional Jump. Transfers control to another part of the program. 4-byte address may be entered in this form: 1234h:5678h, first value is a segment second value is an offset.

Algorithm:

always jump

Example:

include 'emu8086.inc'
ORG 100h
MOV AL, 5
JMP label1 ; jump over 2 lines!
PRINT 'Not Jumped!'
MOV AL, 0
label1:
PRINT 'Got Here!'
RET</td>
</tr>
<tr>
<td>INT</td>
<td>Label</td>
<td>Interrupt, used to take input or to show output.

Algorithm:
Halt the program to fulfill the interrupt depending on “ah” register value.

Example:
org 100h
mov ah,1
int 21h
ret

<table>
<tr><td>Single Input</td><td>ah=1
int 21h (al=input)</td></tr>
<tr><td>Single Output</td><td>ah=2
int 21h (print dl as ascii)</td></tr>
<tr><td>Single Message/String Print:</td><td>ah=9
dx->offset “string name”
int 21h</td></tr>
</table>
</td>
</tr>
</table>

Practice 1 (Length of String)

Copy the code in emulator and run:

```
org 100h

lea si,str1

search:
  mov bl,[si]
  inc si
  inc count
  cmp bl,"$"
  jnz search

ret

str1 db "Hello$"
count db -1
```

Practice 2 (Find key in string)

Copy the code in emulator and run:

```
org 100h

lea si,str1

search:
  mov bl,[si]
  inc si
  cmp bl,key
  jz save
back:
  cmp bl,"$"
  jnz search
ret

save:
  inc count
  jmp back

ret

str1 db "Hello$"
count db 0
key db "l$"
```

Practice 3 (Reverse String)

Copy the code in emulator and run:

```
org 100h

lea si,str1
lea di,str2
add di,4   ;(string length-1)

search:
  mov bl,[si]
  mov [di],bl
  inc si
  dec di
  cmp bl,"$"
  jnz search

  mov dx,offset msg1
  mov ah,9
  int 21h

lea di,str2
mov cx,5   ;(string length)
print:

  mov ah,2
  mov dl,[di]
  int 21h
  inc di
  loop print

ret

str1 db "hello$"
str2 db 5 dup (0)

msg1 db "Reverse string is:$"
```

Practice 4 (String Match)

Copy the code in emulator and run:

```
org 100h

lea si,str1
lea di,str2

search:
  mov bl,[si]
  mov bh,[di]
  cmp bl,bh
  jnz result
  inc si
  inc di
  cmp [si],"$"
  jnz search

  mov dx,offset msg1
  mov ah,9
  int 21h
  ret


result:
  mov dx,offset msg2
  mov ah,9
  int 21h


ret

str1 db "Hello$"
str2 db "World$"

msg1 db "Same String$"
msg2 db "Not Same String$"
```