**Assembly Language – Conditional Processing (1)**

6-1 Boolean and Comparison Instructions

Objective: Understanding Usage of Boolean operators *AND* and *XOR*.  
The code below is to count the odd numbers in “myArray”.

| .data  myArray BYTE 1, 2, 4, 7, 10, 11  .code  main PROC  mov ecx, LENGTHOF myArray ; Set the number for LOOP executions  xor ebx, ebx ; ebx = 0  xor esi, esi ; esi = 0  L1:  movzx eax, myArray[esi]  and eax, 1 ; if *eax* equals to 1,  ; *myArray[esi]* is an odd number,  ; if *eax* is 0, *myArray[esi]* is an even number  add ebx, eax  L2:  inc esi  loop L1  exit  main ENDP  END main |
| --- |

1. Assume that the 0th row shows the value of the registers when the program executes to L1, and fills the value of the registers in the i-th row of the table when the "loop L1" is executed to the L2.

|  | EAX | EBX |
| --- | --- | --- |
| 0 | 00000000h | 00000000h |
| 1 | 00000001h | 00000001h |
| 2 | 00000000h | 00000001h |
| 3 | 00000000h | 00000001h |
| 4 | 00000001h | 00000002h |
| 5 | 00000000h | 00000002h |
| 6 | 00000001h | 00000003h |

Assembly Language – Conditional Processing

| Group: | Student ID: | Name: |
| --- | --- | --- |
| Student ID: | Name: |

6-2 Conditional Processing

Objective: Understanding the conditional jump

1. The purpose of code below is let you know how “cmp” and “jump” instructions work with “myString” (lowercase saved to *edx* ; uppercase saved to *ebx*).

Complete the code below:

| .data  myString BYTE "aBCd"  .code  main PROC  mov ecx, LENGTHOF myString ; Set the iterations of loop  xor ebx, ebx ; ebx = 0  xor edx, edx ; edx = 0  mov esi, OFFSET myString ; esi = address of myString  L1:  mov al, [esi]  cmp al, 'a' ; if al < 'a',jump to L2  \_\_\_\_\_\_\_\_\_jb L2\_\_\_\_\_\_\_\_\_\_\_\_  mov dl,[esi] ; dl放入小寫字母  \_\_\_\_\_\_\_\_\_jmp L3\_\_\_\_\_\_\_\_\_\_\_  L2:  mov bl,[esi] ; bl放入大寫字母  L3:  inc esi  loop L1  exit  main ENDP  END main |
| --- |

1. Assume that the 0th row shows the value of the registers when the program executes to L1 the first time. The values in the ith row of the table represent the content when this program executes to L3 the ith time.

|  | AL | ZF | CF | EBX | EDX |
| --- | --- | --- | --- | --- | --- |
| 0 | 33h | 0 | 0 | 00000000h | 00000000h |
| 1 | 61h | 1 | 0 | 00000000h | 00000061h |
| 2 | 42h | 0 | 1 | 00000042h | 00000061h |
| 3 | 43h | 0 | 1 | 00000043h | 00000061h |
| 4 | 64h | 0 | 0 | 00000043h | 00000064h |

# ’a’:61h

’B’:42h

‘C’:43h

‘d’:64h