Assembly Language – Conditional Processing

|  |  |  |
| --- | --- | --- |
| Group:36 | Student ID:109502517 | Name:洪啟貿 |
| Student ID:109502555 | Name:蔡明翰 |

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 ; *myArray[esi]* is an odd number  ; *after and*, it will be 1;  ;*myArray[esi]* is an even number  ;*after and*, it will be 0  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 |