**Assembly Language – Data Transfers, Addressing, and Arithmetic (4)**

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

* 1. Indirect Addressing、JMP and LOOP Instruction

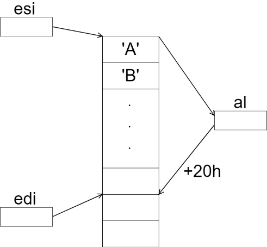
Objective: Understanding indirect addressing for LOOP application.

1. Using indirect addressing and loop, store five letters in ChStr1 to ChStr2 to change capital letter to its lowercase by adding 20h!

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .data  ChStr1 BYTE 'A', 'B', 'C', 'D', 'E'  ChStr2 BYTE LENGTHOF ChStr1 DUP(?)  .code   |  |  |  | | --- | --- | --- | | mov ecx, | 5 | ; set number of LOOP execution | |  | | ; for ChStr1 using LENGTHOF |   mov esi, OFFSET ChStr1 ; esi is pointing to ChStr1[0]  mov edi, OFFSET ChStr2 ; edi is pointing to ChStr2[0]  L: ; LOOP starting point   |  |  |  | | --- | --- | --- | | mov | al,[esi] | ; set value of al with address |   ; value in esi  add al,20h ; 'A' -> 'a'   |  |  |  | | --- | --- | --- | | mov | [edi],al | ; set value of al with address | |  | | ; value in edi |   L2:  inc esi ; esi is pointing to the next  ; point of ChStr1  inc edi ; edi is pointing to the next    ; point of ChStr2 Element   |  |  | | --- | --- | | LOOP | L | |

Whenever a program visits at Label L2 the k-th time, what value in each register becomes?

|  |  |  |  |
| --- | --- | --- | --- |
| k | [esi] | [edi] | ecx |
| 0 | 41h | 00h | 5 |
| 1 | 41h | 61h | 5 |
| 2 | 42h | 62h | 4 |
| 3 | 43h | 63h | 3 |
| 4 | 44h | 64h | **2** |
| 5 | 45h | 65h | 1 |



|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | hex | Symbol | Hex |
| A | 41h | a | 61h |
| B | 42h | b | 62h |
| C | 43h | c | 63h |
| D | 44h | d | 64h |
| E | 45h | e | 65h |