Assembly Language – Data Transfers, Addressing, and Arithmetic (1)

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

4-1 Integer Arithmetic, Data Transfer Instructions

Objective： Familiar with the instructions MOV and SHIFT.

a. The following codes inteㄠㄩ25nd to set Rval = 19\*Val1. Please complete the codes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .data  Val1 SBYTE -1 ; Val1 = 0ffh  Rval SWORD ?  .code  main PROC  L1:  movsx ax, Val1 ;BaseValue  movzx bx, Val1  mov cl, Val1  mov ch, 10h **你看不到我~ 鍋貼好吃**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | L2: |  |  |  | AX | DX | | mov dx, | 0 | ;1. dx = 0 | 1 | **0ffffh** | **0000h** | | add dx, | ax | ;2.*dx*=*dx*+*ax* | 2 | **0ffffh** | 0ffffh(9) | | shl ax, | 1( 1 ) | ;3.*ax*\*2 | 3 | 0fffeh(54)  1fffeh(55)  0fffeh(37) | 0ffffh(41)  教授好帥  我也很帥 | | add dx, | ax | ;4.dx=2Val1+Val1=3Val1 | 4 | 1fffeh(33)  0fffeh(79) | 2fffdh(29)  2fffdh(50)  0fffdh(54) | | shl ax, | 3（38） | ;5.*ax*\*8, ax = 16Val1 | 5 | 0fff0h(30)  ffff0h(55)  0ffff0h(28) | 0fffdh(76)  0fffd(37)  2fffdh(28) | | add dx, | ax | ;6.*dx*=19\**Val1* | 6 | 0ffff0h(36)  0ffff0h(50)  0ffff0h(7)  0fff0h(81) | 0ffedh(16)  12fffdh(50)  0ffedh(7)  0ffedh(44)  12ffedh(28) |   L3: exit  main ENDP  END main |

1. Based the code above, when is not any command executed at L1 position, registers values will be as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| EAX | 00000000h | EBP | 0012fb50h |
| EBX | 00000000h | ESP | 0012fb24h |
| ECX | 0012fb08h | ESI | fffffffeh |
| EDX | 772070b4h | EDI | 00000000h |

When all the command have been executed before L3 position, what is registers values?

|  |  |  |  |
| --- | --- | --- | --- |
| EAX | 0000fff0h(10) | EBP | 0012fb50h |
| EBX | 000000ffh(3) | ESP | 0012fb24h |
| ECX | 001210ffh(60) | ESI | fffffffeh |
| EDX | 7720ffedh(57)  7720ffedh(37)  7720ffedh(36) | EDI | 00000000h |