**Assembly Language – Data Transfers, Addressing, and Arithmetic (2)**

|  |  |  |
| --- | --- | --- |
| 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 intend 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   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | L2: |  |  |  | AX | DX | | mov dx, | 0 | ;1. dx = 0 | 1 | **0ffffh** | **0000h** | | add dx, | ax | ;2.*dx*=*dx*+*ax* | 2 | **0ffffh** | 0ffffh | | shl ax, | 1 | ;3.*ax*\*2 | 3 | 0fffeh | 0ffffh | | add dx, | ax | ;4.dx=2Val1+Val1=3Val1 | 4 | 0fffeh | 1fffdh | | shl ax, | 3 | ;5.*ax*\*8, ax = 16Val1 | 5 | 0fff0h | 1fffdh | | add dx, | ax | ;6.*dx*=19\**Val1* | 6 | 0fff0h | 1ffedh |   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 commands have been executed before L3 position, what is the register's values?

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

4-2 Data Transfer Instructions, Addition and Subtraction

Objective: Understanding the instructions (ADD, SUB, and NEG)

1. Complete the following code to implement this equation:

Rval = -(Val3 - (Val1 - Val2))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .data  Val1 SBYTE 03h  Val2 SBYTE 02h  Val3 SBYTE 8fh  Rval SWORD ?  .code  movsx ax, Val1  movsx bx, Val2  movsx cx, Val3  ; Rval = -(Val3 - (Val1 - Val2))   |  |  |  |  |  | | --- | --- | --- | --- | --- | | sub | ax, | bx | ; ax= (Val1 – Val2) | | | sub | cx,ax | | ; cx = Val3 – (Val1 – Val2) | | | neg cx | | | ; cx = -cx | |  | | |  | | | mov | Rval,cx | | ; Rval = cx | | |