PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 4.

ASSEMBLY PROGRAMMING LANGUAGE LABORATORY PROBLEM SHEET 1

Context: 8086 Emulator Practice

***Data transfer instructions I***

|  |  |
| --- | --- |
| Step 1 | Using emu8086 to assemble the instructions   1. MOV AX,BX 2. MOV AX,0AAAAh 3. MOV AX,[BX] 4. MOV AX,[4] 5. MOV AX,[BX+SI] 6. MOV AX,[SI+4] 7. MOV AX,[BX+SI+4] |
| Step 2 | Initializing the internal registers of the 80x86 as follows:  (AX) = 0000H (BX) = 0001H (CX) = 0002H (DX) = 0003H (SI) = 0010H (DI) = 0020H (BP) = 0030H (DS) = 0B60H  Verify the initialization by displaying the new content of registers |
| Step 3 | Fill all memory locations in the range DS:00 through DS:1F with 00H and then initialize the following storage locations:  (DS:0001H) = BBBBH (DS:0004H) = CCCCH (DS:0011H) = DDDDH  (DS:0014H) = EEEEH (DS:0016H) = FFFFH |
| Step 4 | Trace the execution of the instructions (a) through (g). Explain the execution of **each** instruction, including addressing mode, physical address for memory addressing mode, value in AX. Fill the table below. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction | Addressing Mode | Physical Address | AX | Function |
| a | | | | |
| b | | | | |
| c | | | | |
| d | | | | |
| e | | | | |
| f | | | | |
| g | | | | |

***Data transfer instructions II***

|  |  |
| --- | --- |
| Step 1 | Assemble the instruction MOV SI, [0ABCh] to memory at address CS:100 and verify loading of the instruction. How many bytes does the instruction take up?  ……………………………………………… |
| Step 2 | Initialize the word of memory starting at DS:0ABC with the value FFFFH , and DS register to 0B60H |
| Step 3 | Clear the SI register, verify by displaying its content. |
| Step 4 | Trace the execution. Describe the operation performed by the instruction. |
| Step 5 | Assemble the instruction MOV WORD PTR [SI], ABCD into memory at  Address CS: 100 and then verify loading of the instruction. How many bytes does it take? ............................................................................... |
| Step 6 | Initialize SI register with the value 0ABCH |
| Step 7 | Clear the word of memory starting at DS:0ABC |
| Step 8 | Trace the execution. Describe the operation performed by the instruction. |

Step 8 trace

Step 4 trace

***Arithmetic instructions***

|  |  |
| --- | --- |
| Step 1 | Assemble the instruction ADC AX, [0ABC] to memory at address CS:100 and verify loading of the instruction. How many bytes does the instruction take up? …………………………………………………… |
| Step 2 | Initialize the word of memory starting at DS:0ABC with the value FFFFH, and DS register to 0B60H |
| Step 3 | Initialize AX with the value 0001H. Verify by displaying register contents. |
| Step 4 | Clear the carry flag |
| Step 5 | Trace the execution. Describe the operation performed by the instruction and what happened to the carry flag? |

Step 5 trace

|  |  |
| --- | --- |
| Step 1 | Assemble the following instruction sequence into memory at address CS:100 and then verify loading of the instruction. How many bytes do they take?  …………………………………………..   1. LAHF 2. MOV BH, AH 3. AND BH, 1FH 4. AND AH, 0E0H 5. MOV [200H], BH 6. SAHF |
| Step 2 | Initialize DS register to 0B60H and the byte of memory starting at DS:200H with the value 00H |
| Step 3 | Clear register AX and BX |
| Step 4 | Display the current state of flags, make sure the status flags equal NG, ZR, AC, PE and CY |
| Step 5 | Trace the execution. Describe the operation performed by each instruction. What value is read out of the flags register? What value is saved in memory? What value is reloaded into flags register? |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | Flags before execution | Memory address affected | Memory contents | Flags after execution | Function |
| a | | | | | |
| b | | | | | |
| c | | | | | |
| d | | | | | |
| e | | | | | |
| f | | | | | |