|  |  |
| --- | --- |
| **Microprocessors Systems**  **CMPS-201** | **Cairo University**  **Faculty of Engineering**  **Computer Engineering Department** |

**Sheet 2 Solution Manual**

**Question 3**

|  |
| --- |
| **1278** |

After the execution of **PUSHF**, the **SP** =

The contents of the stack are after **PUSHF** are:

|  |  |
| --- | --- |
| **Address** | **Content** |
| 1288 | N/A |
| 1287 | 04 |
| 1286 | 53 |
| 1285 | AH |
| 1284 | AL |
| 1283 | BH |
| 1282 | BL |
| 1281 | CH |
| 1280 | CL |
| 127F | DH |
| 127E | DL |
| 127D | SI High Byte |
| 127C | SI Low Byte |
| 127B | DI High Byte |
| 127A | DI Low Byte |
| 1279 | FR High Byte |
| 1278 | FR Low Byte |

To restore the contents of the registers, we use these instructions:

|  |  |  |
| --- | --- | --- |
| Instruction | **SP** after the execution of the instruction | |
| POPF | 127A | |
| POP DI | 127C | |
| POP SI | 127E | |
| POP DX | 1280 | |
| POP CX | 1282 | |
| POP BX | 1284 | |
| POP AX | 1286 | |
| RET | 1288 | |
| The instruction that pushes all registers in one go is | | |  | | --- | | PUSHA | |
| The instruction that pops all registers at once | | |  | | --- | | POPA | |

**Question 4**

|  |  |
| --- | --- |
| The syntax errors in the program are:   1. DE6H => 0DE6H 2. No “:” after START 3. SUM => RES 4. STRT => START | The logical errors in this program are:   1. MOV AX, DATA => MOV AX, @DATA 2. INC DI => ADD DI, 2 3. DEC CX 4. MOV [SI], BX => MOV [SUM], BX |

The program now is

|  |
| --- |
| .MODEL SMALL  .STACK 32  ;--------------------  .DATA      DATA DW  234DH,0DE6H,3BC7H,566AH           ORG 10H      SUM  DW  ?  ;--------------------  .CODE  START PROC  FAR            MOV AX,@DATA            MOV DS,AX            MOV CX,04             *;LOOP COUNTER = 4*            MOV BX,0              *;INITIALIZE BX TO ZERO*            MOV DI,OFFSET DATA *;SET UP DATA POINTER BX*      LOOP1:ADD BX,[DI]       *;ADD CONTENTS* POINTED *AT BY [DI] TO BX*            add di, 2         *;POINT DI TO NEXT* BYTE            dec cx            JNZ LOOP1         *;JUMP IF COUNTER NOT ZERO*            MOV [SUM],BX      *;STORE THE SUM*            HLT  START ENDP  END START |

**Question 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Code | Result Stored In | Result | CF | ZF | AF |
| MOV BH,3FH  ADD BH,45H | BH | **84** | **0** | **0** | **1** |
| MOV BX,0FF01H  ADD BL,BH | **BL** | **1 + 00** | **1** | **1** | **1** |
| MOV AH,0FEH  STC  ADC AH,00 | **AH** | **FF** | **0** | **0** | **0** |
| MOV DX,4599H  MOV CX,3458H  ADD CX,DX | **CX** | **79F1** | **0** | **0** | **1** |
| MOV AX,255  STC  ADC AX,00 | **AX** | **0100** | **0** | **0** | **1** |
| MOV CX,0FFFFH  STC  ADC CX,00 | **CX** | **1 + 0000** | **1** | **1** | **1** |

**Question 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Result Stored In | Result (HEX) | ZF | CF |
| AND DX, AX | **DX** | **E000** | **0** | **0** |
| XOR AL, 76H | **AL** | **76** | **0** | **0** |
| XOR AX, AX | **AX** | **0000** | **1** | **0** |
| AND AH, 0FFH | **AH** | **F0** | **0** | **0** |
| XOR DX, 0EEEEH | **DX** | **0D7E** | **0** | **0** |
| MOV CL, 04  SHL AL, CL | **AL** | **00** | **1** | **0** |
| MOV CL, 3  SHR DL, CL | **DL** | **12** | **0** | **0** |
| MOV CL, 6  SHL DX, CL | **DX** | **E400** | **0** | **0** |

**Question 7**

|  |  |  |
| --- | --- | --- |
| **Code** | **ZF** | **CF** |
| MOV BX,2500  CMP BX,1400 | **0** | **0** |
| SUB AX, AX  CMP AX,0000 | **1** | **0** |
| MOV DL,34  CMP DL,88 | **0** | **1** |
| MOV AL,0FFH  CMP AL,6FH | **0** | **0** |
| XOR DX, DX  CMP DX,0FFFFH | **0** | **1** |
| MOV BX,2378H  MOV DX,4000H  CMP DX, BX | **0** | **0** |
| SUB CX, CX  DEC CX  CMP CX,0FFFFH | **1** | **0** |
| MOV AL,0AAH  AND AL,55H  CMP AL,00 | **1** | **0** |

**Question 8**

Case A: **Jump happens.**

Case B: **Jump does not happen.**

Case C: **Jump does not happen.**