|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMP201A** | **Total: 20 Points** |
| **Faculty of Engineering** | **Microprocessor Systems I** | **2018-2019** |
| **Computer Eng. Department** | **Midterm Exam** | **One Hour** |

**This is an open-book, open notes exam. All electronic devices - Except calculators - are forbidden.  
Make any reasonable assumptions (if necessary)**

Name: Section: B.N.:

1. **[16] Identify the choice that best completes the statement or answers the question**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | **1- The addressing mode of mov ax,[bx] is** | | | | | | | | | | | |
| a) register | | | b) direct | | | c) register indirect | | | | d) none of the above | |
| B | **2- To add the content of al to the content of bl** | | | | | | | | | | | |
| A) add al, bl | | | b) add bl, al | | | c) sum al, bl | | | | | d) sum bl, al |
| A | **3- Increasing the data bus size increases** | | | | | | | | | | | |
| a) the bus bandwidth | b) addressable locations | | | c) available memory size | | | | d) none of the above | | | |
| D | **4- To copy the content of bl to ax** | | | | | | | | | | | |
| a) copy ax, bl | | | b) mov ax, bl | | | c) mov bl, ax | d) none of the above | | | | |
| B | **5- What are the names of the 4 segment registers?** | | | | | | | | | | | |
| a) Data, Index, Code, Stack | | b) Stack, Extra, Code, Data | | | c) Stack, Data, Base, Counter | | | | d) Stack, Index, Extra, Code | | |
| D | **6-Assuming that AL=0, then number of times the instruction sequence below will loop is**  **T: INC AL**  **JNZ T** | | | | | | | | | | | |
| a) 0 | | | b) 1 | | | c) 255 | | | d) 256 | | |
| C | **7-The instruction, CMP to compare source and destination operands it performs** | | | | | | | | | | | |
| a) Addition | | | b) Division | | | c) Subtraction | | | d) Multiplication | | |
| D | **8- The main enhancement of 8088 over 8086 is** | | | | | | | | | | | |
| a) Speed | | | b) registers size | | | c) accessible memory size | | | | | d) cost |

1. **[14] Using one line [without any other changes]**

|  |  |  |
| --- | --- | --- |
| Invert bits D3,D4 of AL | XOR AL,00011000 b | |
| Set bits D5,D7 of bl | OR BL,10100000 b | |
| Swap the nibbles of dh | ROR DH,4 - ROL DH,4 | |
| Decrease the content of ah by one  [Two methods] | DEC AH | SUB AH,1 |
| Declare an uninitialized word labeled “STD” | STD DW ? | |
| Declare a string variable named SysData containing the word “TEST” repeated 500 times. | SysData DB 500 DUP(‘TEST’) | |

1. **[4] Complete the following**
2. **For the following code, the final value of AX= 0B H**

mov ax, 7H

mov cx, 4H

L1: inc ax

loop L1

1. **In the following data definition, assume that STD begins at offset 2002h. The offset of the byte containing the value (6) =** 2006 **H**

STD DD 1,6,5,8,9

**4- Two system designers decided to try solving the branch penalty problem. The first designer suggested adding another BIU queue to fetch and prepare code lines for both cases (JUMP and Not JUMP). The second designer suggests adding another very simplified EU to be responsible for pre-evaluating JUMP conditions lines only. Other than design cost, what is the main disadvantage of each solution?**

1. [2] The first design:
2. [2] The second design:
3. [5][Bonus] In no more than three lines, suggest your own solution:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**5- [6] what is the content of AX and BX after the following code**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| .model small  .stack 64  .data  a label byte  b label byte  c db 12h | d db 14h  e dw 30h  .code  main proc  mov ax,@data  mov ds,ax | mov a+2,20h  mov ax, word ptr d+1  mov bl, c  mov cx,0  shl bx,8  push cx | popf  adc bx, ax  hlt  main endp  end main |  | AX= 0020 H  BX= 1220 H |

**6- [4] Assuming that all variables are declared. Put (√) in front of only four illegal commands**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Illegal?** |  | **Illegal?** |
| PUSH AX |  | MOV AX, 5000H |  |
| ADD DX,[CX+WORDVAL] | **√** | MOV DS, 5000H | **√** |
| INC CX,2 | **√** | MOV AX, MARK[SI] |  |
| ADD AX,[BX] |  | MOV [BX],[SI] | **√** |
| MOV AX, BX |  | MOV SI,CS |  |

**7- [4] Suppose we have declared**

A DW 0A00h, 1A01h, 2A02h

B DW 0B00h, 1B01h, 2B02h, 3B03h

C DB 0Ch, 1Ch, 2Ch, 3Ch, 4Ch, 5Ch

**What are the contents of ax or al after execution of each of the following instructions?**

Mov al, BYTE PTR [A + 3] ; AL = 1A H

MOV ax, WORD PTR [C - 11] ; AX = 021A H

**8- [6] The output of the following program is** : [KoKoWaWa]

|  |  |  |  |
| --- | --- | --- | --- |
| .model small  .data  A db "Khaled"  B Db "What?"  S EQU 3  .code  main proc far  mov bx,@data | mov DS,bx  mov al,a  mov ah,b  mov bh,0h  mov a[2],al  mov a[4],ah  mov ah,3h  int 10h | mov ah,1  mov al,b+S  sub al,2\*S-1  int 16h  mov dl, byte ptr b+2  mov B+S-4,dl  mov ah,9  mov byte ptr a+1,al | mov b[1],dl  lea dx,A  mov byte ptr a+S,al  mov B+S-1,'$'  int 21h  hlt  main endp  end main |