|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMPN201** | **Total:100 Points** |
| **Faculty of Engineering** | **Microprocessor Systems** | **2019-2020** |
| **Computer Eng. Department** | **Final Exam**  | **Two Hour** |

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

**Q1 – A - [12] Indicate if the statement is True or False by writing √ or X**

|  |  |
| --- | --- |
| ADD command changes flag register the same as MOV command do |  |
| Loop command uses CX register |  |
| Parallel communication is used for small distances only |  |
| SI and DI are used for string operations |  |
| One word = four nibbles |  |
| Data bus, address bus and control bus are bidirectional buses |  |

**Q1 – B - [5] What does the following abbreviations stands for:**

|  |  |
| --- | --- |
| **UART** |  |
| **TTL** |  |
| **Modem** |  |
| **DTE** |  |
| **DCE** |  |

**Q1 – C - [10] What does the following programs do? [Results saved in RESULT variable]**

|  |  |
| --- | --- |
| .MODEL SMALL.DATAX DB 2RESULT DB ?.CODEMAIN PROC FARMOV AX,@DATAMOV DS,AXMOV AL, XAND AL,ALJZ T MOV AH,1JMP ENDT: MOV AH,0END: MOV RESULT,AH MAIN ENDPEND MAIN | .MODEL SMALL.DATAX DB 7Y DB 9 RESULT DB ? .CODEMAIN PROC FAR MOV AX,@DATAMOV DS,AXMOV AL, XMOV AH,YADD AL,AHMOV RESULT,ALMAIN ENDPEND MAIN |
|  |  |

**Q1 – D - [6] Trace the following code to find AX after each command.**

|  |  |
| --- | --- |
|  | **AX** |
|  | **0** | **0** | **0** | **0** |
| MOV AX,4CACH |  |  |  |  |
| ADD AX,0BACFH |  |  |  |  |
| SUB AX,0EDACH |  |  |  |  |
| ADC AX,12H |  |  |  |  |
| SBB AX,768H |  |  |  |  |
| XCHG AL,AH |  |  |  |  |

 **Q2 – A - [15] Write (in HEX) the value passed over Data bus, Address bus and Control bus for each of the following INDEPENDENT cases Assuming DS=2318H, Var1 logical address =2318:0300,Var1 =60H,Var2 logical address =2318:0200,Var2 =52H,MEMR=1,MEMW=2,IOR=4 and IOW=8,AX=50H**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data Bus (Hex)** | **Address Bus (Physical - Hex)** | **Control Bus** |
| 1. Add AX, [200H]
 |  |  |  |  |  |  |  |  |  |  |
| 1. Mov VAR1, 200H
 |  |  |  |  |  |  |  |  |  |  |
| 1. Add AX,Var2
 |  |  |  |  |  |  |  |  |  |  |
| 1. IN AX,20H
 |  |  |  |  |  |  |  |  |  |  |
| 1. OUT 30H,AX
 |  |  |  |  |  |  |  |  |  |  |

**Q2 – B - [28] Trace the following code to find AX, BX and flags values after each command. Assume that flags values initially are SF=0,CF=0, PF=0, AC=0, OF=0 and ZF=0 and all registers are 0**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AX** | **BX** | **SF** | **CF** | **PF** | **AC** | **OF** | **ZF** |
|  | **0000 H** | **0000 H** | **0** | **0** | **0** | **0** | **0** | **0** |
| MOV AX, 2FF5H |  |  |  |  |  |  |  |  |
| MOV BX, 0F2ADH |  |  |  |  |  |  |  |  |
| ADD AX, BX |  |  |  |  |  |  |  |  |
| SUB AL, BL |  |  |  |  |  |  |  |  |
| ADC AH, AL |  |  |  |  |  |  |  |  |
| SHL AH,1 |  |  |  |  |  |  |  |  |
| RCL AL,1 |  |  |  |  |  |  |  |  |

**Q3 - [12] - *Hany strikes again***: after he has been fired, he decided to modify his strategy. He is still paid by the number of lines he wrote. Adding useless lines is not valid yet. Therefore, in this code, if you removed any single line, it will not work correctly. However, this program could be simplified to a much smaller version.

[6] a. What does this program do, knowing that, the final result is saved in AL only (nothing else matters)?

[ …………….…………………………………………………………. ]

[6] b. Rewrite the same functionality with the minimum number of lines [each extra line more than optimal for a working program = -1 – Non working program =0 ]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |
| --- |
| .MODEL SMALL |
| .STACK 64 |
| .DATA |
| VAR1 DD 4,2,4,5 |
| VAR2 DW 5,5,3,6,7 |
| VAR3 DW 6,1,0,2 |
|  DB 3,2,4,5,1 |
| VAR5 DB 2 |
| VAR6 DW 5 |
| VAR7 DB 2 |
| .CODE |
| MAIN PROC FAR |
| MOV AX,@DATA |
| MOV BX,AX |
| MOV DS,BX |
| MOV BL,VAR5 |
| MOV CX,2345H |
| SUB CH,01CH |
| XCHG CH,DL |
| MOV DI,DX |
| MOV VAR6,DI |
| MOV SI,VAR6 |
| MOV VAR3[2],SI |
| MOV AL,BYTE PTR VAR3+2 |
| MOV CL,AL |
| SHL BX,CL |
| MOV DH,VAR7 |
| MOV VAR2[4],DX |
| MOV CX,VAR2[5] |
| SHL BX,1 |
| PUSH CX |
| POP AX |
| XCHG AH,AL |
| ADD AH,BH |
| MOV AL,AH |
| MAIN ENDP |
| END MAIN |

 |

|  |
| --- |
| .MODEL SMALL |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| MAIN ENDP |
| END MAIN |

 |

**Q4 - [12] An attacker intercepted a serial transmission for 12.5 milliseconds. The transmission was as the following table. Complete the following transmission configurations**

1. **The word size is (7 – 8) : ………. Bits**
2. **Parity (Odd – Even – No parity) :………..**
3. **Number of stop bits: …….. bits**
4. **Number of transmitted words = ………**
5. **The max length between transmitter and receiver =……………ft**

**[Note: header numbers are for the transmission order. Meaning that, the first, second and third transmitted bits are zeros, the fourth is one and the fifth is 0 and so on]**

 First transmitted bit 8/no parity/1/11/500

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***10*** | ***11*** | ***12*** | ***13*** | ***14*** | ***15*** | ***16*** | ***17*** | ***18*** | ***19*** | ***20*** |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|  |
| ***21*** | ***22*** | ***23*** | ***24*** | ***25*** | ***26*** | ***27*** | ***28*** | ***29*** | ***30*** | ***31*** | ***32*** | ***33*** | ***34*** | ***35*** | ***36*** | ***37*** | ***38*** | ***39*** | ***40*** |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
|  |
| ***41*** | ***42*** | ***43*** | ***44*** | ***45*** | ***46*** | ***47*** | ***48*** | ***49*** | ***50*** | ***51*** | ***52*** | ***53*** | ***54*** | ***55*** | ***56*** | ***57*** | ***58*** | ***59*** | ***60*** |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
|  |
| ***61*** | ***62*** | ***63*** | ***64*** | ***65*** | ***66*** | ***67*** | ***68*** | ***69*** | ***70*** | ***71*** | ***72*** | ***73*** | ***74*** | ***75*** | ***76*** | ***77*** | ***78*** | ***79*** | ***80*** |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  |
| ***81*** | ***82*** | ***83*** | ***84*** | ***85*** | ***86*** | ***87*** | ***88*** | ***89*** | ***90*** | ***91*** | ***92*** | ***93*** | ***94*** | ***95*** | ***96*** | ***97*** | ***98*** | ***99*** | ***100*** |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
|  |
| ***101*** | ***102*** | ***103*** | ***104*** | ***105*** | ***106*** | ***107*** | ***108*** | ***109*** | ***110*** | ***111*** | ***112*** | ***113*** | ***114*** | ***115*** | ***116*** | ***117*** | ***118*** | ***119*** | ***120*** |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |