|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMPN201** | **Total: 20 Points** |
| **Faculty of Engineering** | **Microprocessor Systems I** | **2017-2018** |
| **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)**

1. **[10] Identify the choice that best completes the statement or answers the question**

|  |  |
| --- | --- |
|  D | **1- A 32-bit address bus allows access to a memory of capacity** |
| a) 64 MB | b) 16 MB | c) 1 GB | d) 4 GB |
| B  | **2- The system bus is made up of** |
| A) data bus | b) data bus, control bus and address bus | c) data bus and address bus | d) data bus and control bus |
|  C | **3- Which of the following is the pair of register used to access memory in string instructions?** |
| a) DI and BP  | b) SI and BP | c) DS and SI | d) DI and SI |
|  B | **4- Pipelining improves CPU performance due to** |
| a) reduced memory access time | b) the introduction of parallelism | c) increased clock speed | d) additional functional units |
|  D | **5- Which flags are NOT used for mathematical operations?** |
| a) Carry, Interrupt and Trap flag  | b) Direction, Interrupt and Sign flag  | c) Direction, Overflow and Trap flag  | d) Direction, Interrupt and Trap flag |
|  D | **6- If CS = 0701H, SS = 0801H, SI = 0100H and IP = 0108H the address of the next instruction is:** |
| a) 07090H | b) 07811H | c) 07110H | d) 07118H |
|  D | **7- Which of the following, when used in the data section of a MASM program, reserves 40 bytes of RAM(memory)?** |
| a) db 20 DUP (2) | c) db 20 DUP (20) | b) dw 40 DUP (1) | d) dw 20 DUP (1) |
|  B | **8- The bus that defines the “size” of the processor is --------------** |
| a) The system bus | b) The data bus | c) The address bus | d)The size bus |
|  D | **9- Which flag(s) does the 80x86 use to check for unsigned arithmetic overflow?** |
| a) Overflow | b) Direction | c) Interrupt | d) Carry |
|  D | **10- The sp register is typically used for accessing** |
| 1. strings
 | 1. data segment
 | 1. memory
 | 1. stack
 |
|  A | **11- What will be the contents of register AL after the following has been executedMOV BL, 8CMOV AL, 7EADD AL, BL** |
| 1. 0A and carry flag is set
 | 1. 6A and carry flag is set
 | 1. 0A and carry flag is reset
 | 1. 6A and carry flag is reset
 |
|  A | **12- What will be the values of the Sign, and Zero flags after the following instructions have executed?mov ax,620hsub ah,0F6h** |
| 1. S=0,Z=0
 | 1. S=0,Z=1
 | 1. S=1,Z=0
 | 1. S=1,Z=1
 |
|  B | **13- The BX register is used in multiplication operation to hold the upper 16-bits of the result.** |
| 1. Always
 | 1. Never
 | 1. Usually
 | 1. Rarely
 |
| C  | **14- What is the largest signed integer that may be stored in 32 bits?** |
| 1. 232 - 1
 | 1. 232
 | 1. 231 - 1
 | 1. 231
 |
|  C | **15- Which of the following will generate assembly errors?** |
| a) var1 db 1101b, 22, 35 | b) var2 db "ABCDE", 18 | c) var4 db 256, 19, 40 | d) None of the above |
|  D | **16- To reserve 32-bits in memory \_\_\_ directive is used.** |
| 1. db
 | 1. dn
 | 1. dw
 | 1. dd
 |
|  D | **17 - In the following data definition, assume that Count begins at offset 2002h. What is the offset of the third value (5)?Count DD 1,6,5,8,9** |
| 1. 2004h
 | 1. 2006h
 | 1. 2008h
 | 1. 200Ah
 |
|  C | **18- One of the following is not a valid segment address** |
| 1. 00000
 | 1. 8CE90
 | 1. 8CE91
 | 1. E0840
 |
|  A | **19- What is the result in AL after executing the following instructions?XOR AL, ALOR AL, 80H** |
| 1. 80H
 | 1. 88H
 | 1. 00H
 | 1. None of the above
 |
|  A | **20 - Which of the following are performed when an interrupt occurs:(I) FLAGS register is pushed to the stack(II) CS register is pushed to the stack(III) IP register is pushed to the stack** |
| a) (I) and (II) and (III)  | b) (I) and (II) only | c) (II) and (III) only  | d) (I) and (III) only |

**2- [2] If the content of the register SS = 3500H and the content of the register SP= FFFEH, then**

1. Calculate the physical address.

35000+FFFE = 44FFE

1. Calculate the lower range of the stack.

35000+0000 = 35000

1. Calculate the upper range of the stack segment.

35000+FFFF = 44FFF

1. Show the logical address of the stack.

3500:FFFE

**3- [2] Using one line**

1. Declare a byte containing the value 64. Label the memory location “var”.

var DB 64 / var DB 100 / var DB 40

1. Declare an unlabeled byte initialized to 10.

DB 10

1. Set the most significant nibble of CX

OR CX,1111 000 0000 0000 b / OR CX,0F000 H

1. Declare 10 uninitialized bytes starting at the address label “bytes”.

bytes DB 10 DUP(?)

**4- [2] Use the following data definitions to answer this question:**

.DATA

num1 DW 100

num2 DB 225

char1 DB ’Y’

num3 DD 0, 10,30,40,50

Identify whether the following instructions are **legal** or **illegal**. Explain the reason for each illegal
instruction.

|  |  |  |
| --- | --- | --- |
| Instruction  | Answer  | Reason |
| MOV num2, num1 | **Illegal** | Memory to Memory |
| MOV DS, CS | **Illegal** | Segment to Segment / [*Legal is also accepted*] |
| SUB char1, ’A’ | **Legal** |  |
| INC num3, 1 | **Illegal** | Syntax Error |

**5- [4] Write complete assembly program that computes the following equations**

L=J+K and U3=U1-U2

In your code:
• Define the 2-bytes variables J, K and L. (; Initialize J to 3 and K to -2)
• Define the 2-bytes variables U1, U2 and U3. (; Initialize U1 to 254 and U2 to 22)
• Get J, K, U1, and U2 from the user and display L and U3

Assume that all inputs and results are one digit numbers

*; Author: Khaled Mohamad*

*; Date:11-11-2017*

*; This program calculates L=J+K and U3=U1-U2*

;---------------------------

**GetNum MACRO NUM**

MOV AH,07

INT 21H

MOV DL, AL

MOV AH,0

SUB AL,'0'

MOV NUM, AX

MOV AH,2

INT 21H

**ENDM GetNum**

**ShowNUM MACRO NUM**

MOV AH,2

MOV DX, NUM

ADD DX,'0'

INT 21H

**ENDM ShowNUM**

 **.MODEL SMALL**

 **.DATA**

J dw 3

K dw -2

L dw ?

U1 dw 254

U2 dw 22

U3 dw ?

 **.code**

**MAIN PROC FAR**

 MOV AX, @DATA

 MOV DS, AX

 GetNum J

 GetNum K

 GetNum U1

 GetNum U2

 MOV AX, J

 ADD AX, K

 MOV L, AX

 MOV AX, U1

 SUB AX, U2

 MOV U3, AX

 ShowNUM L

 ShowNUM U3

**MAIN ENDP**

 **END MAIN**