|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **[60] Answers for question 1 [Put √ for the correct answer]** | | | | | | | | | | |
|  | **A** | **B** | **C** | **D** |  |  | **A** | **B** | **C** | **D** |
| 1 |  |  |  |  | 31 |  |  |  |  |
| 2 |  |  |  |  | 32 |  |  |  |  |
| 3 |  |  |  |  | 33 |  |  |  |  |
| 4 |  |  |  |  | 34 |  |  |  |  |
| 5 |  |  |  |  | 35 |  |  |  |  |
| 6 |  |  |  |  | 36 |  |  |  |  |
| 7 |  |  |  |  | 37 |  |  |  |  |
| 8 |  |  |  |  | 38 |  |  |  |  |
| 9 |  |  |  |  | 39 |  |  |  |  |
| 10 |  |  |  |  | 40 |  |  |  |  |
| 11 |  |  |  |  | 41 |  |  |  |  |
| 12 |  |  |  |  | 42 |  |  |  |  |
| 13 |  |  |  |  | 43 |  |  |  |  |
| 14 |  |  |  |  | 44 |  |  |  |  |
| 15 |  |  |  |  | 45 |  |  |  |  |
| 16 |  |  |  |  | 46 |  |  |  |  |
| 17 |  |  |  |  | 47 |  |  |  |  |
| 18 |  |  |  |  | 48 |  |  |  |  |
| 19 |  |  |  |  | 49 |  |  |  |  |
| 20 |  |  |  |  | 50 |  |  |  |  |
| 21 |  |  |  |  | 51 |  |  |  |  |
| 22 |  |  |  |  | 52 |  |  |  |  |
| 23 |  |  |  |  | 53 |  |  |  |  |
| 24 |  |  |  |  | 54 |  |  |  |  |
| 25 |  |  |  |  | 55 |  |  |  |  |
| 26 |  |  |  |  | 56 |  |  |  |  |
| 27 |  |  |  |  | 57 |  |  |  |  |
| 28 |  |  |  |  | 58 |  |  |  |  |
| 29 |  |  |  |  | 59 |  |  |  |  |
| 30 |  |  |  |  | 60 |  |  |  |  |
| **[5] Answers for question 2 [Fill in each location with 0 or 1]** | | | | | | | | | | |
| **CF** | **ZF** | **DF** | **OF** | **PF** |  | | | | | |
|  |  |  |  |  |
| **[10] Answers for question 3** | | | | | | | | | | |
|  | **V/I** |  | | |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |
| **7** |  |
| **8** |  |
| **9** |  |
| **10** |  |

|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMPN201** | **Total:120 Points** |
| **Faculty of Engineering** | **Microprocessor Systems** | **2016-2017** |
| **Computer Eng. Department** | **Final Exam [Answer Sheets]** | **Two Hour** |

**Please fill in your answers in the following four pages. Questions starts at page five**

**[6] Answers for question 4**

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
|  |
|  |
| 3 |  |
| 4 |  |

**[5] Answers for question 5 [Put only one digit at each answer location]**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | **AL=** |  |  | H |  | | | | | |
| 2 | **AL=** |  |  | H |
| 3 | **AX=** |  |  |  |  | H |  | | | |
| 4 | **EDX=** |  |  |  |  |  |  |  |  | H |
| 5 | **EDX=** |  |  |  |  |  |  |  |  | H |

**[6] Answers for question 6**

**[2] Answers for question 7 – A [Put only one digit at each answer location]**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AX= |  |  |  |  | **H** |  | BX= |  |  |  |  | **H** |

**[2] Answers for question 7 – B [Put only one digit at each answer location]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | AX= |  |  |  |  | **H** |
| 2 | AX= |  |  |  |  | **H** |

**[2] Answer for question 7 – C**

N=

**[2] Answers for question 7 – D**

The code purpose:

Affecting the result (Yes/No) =

|  |  |
| --- | --- |
| **[4] Answer for question 8-1** | **[4] Answer for question 8-2** |
|  |  |
| **[6] Answer for question 8-3** | **[6] Answer for question 8-4** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMPN201** | **Total:120 Points** |
| **Faculty of Engineering** | **Microprocessor Systems** | **2016-2017** |
| **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 at the answer sheet**

**Q1 - [60] Choose the correct answer**

|  |  |  |
| --- | --- | --- |
| 1. **Operation code field is present in:** | | |
| a) programming language instruction | | b) assembly language instruction |
| c) machine language instruction | | d) none of the mentioned |
| 1. **The length of the one-byte instruction is** | | |
| a) 2 bytes | | b) 1 byte |
| c) 3 bytes | | d) 4 bytes |
| 1. **The instruction, MOV AX, 1234H is an example of** | | |
| a) register addressing mode | | b) direct addressing mode |
| c) immediate addressing mode | | d) based indexed addressing mode |
| 1. **The instruction, MOV AX, [BX] is an example of** | | |
| a) direct addressing mode | | b) register addressing mode |
| c) register relative addressing mode | | d) register indirect addressing mode |
| 1. **The contents of a base register are added to the contents of index register in** | | |
| a) indexed addressing mode | | b) based indexed addressing mode |
| c) relative based indexed addressing mode | | d) based indexed and relative based indexed addressing mode |
| 1. **The instruction that subtracts 1 from the contents of the specified register/memory location is** | | |
| a) INC | | b) SUBB |
| c) SUB | | d) DEC |
| 1. **The flag that acts as Borrow flag in the instruction, SBB is** | | |
| a) direction flag | | b) carry flag |
| c) parity flag | | d) trap flag |
| 1. **In general, the destination operand of an instruction can be** | | |
| a) memory location | | b) register |
| c) immediate data | | d) memory location and register |
| 1. **During comparison operation, the result of comparing or subtraction is stored in** | | |
| a) memory | | b) registers |
| c) stack | | d) no where |
| 1. **The operands, source and destination in an instruction cannot be** | | |
| a) register, register | | b) memory location, memory location |
| c) memory location, register | | d) immediate data, register |
| 1. **The instruction that is not possible among the following is** | | |
| a) MOV AX, [BX] | | b) MOV AX, 5555H |
| c) MOV AX, [SI] | | d) MOV [SI], [DI] |
| 1. **The registers that cannot be used as operands for arithmetic and logical instructions are** | | |
| a) general purpose registers | | b) pointers |
| c) index registers | | d) segment registers |
| 1. **The disadvantage of machine level programming is** | | |
| a) time consuming | b) chances of error are more | |
| c) debugging is difficult | | d) all of the mentioned |
| 1. **The stack is useful for** | | |
| a) storing the register status of the processor | | b) temporary storage of data |
| c) storing contents of registers temporarily | | d) all of the mentioned |
| 1. **The extension file that is must for a file to be accepted by the LINK as a valid object file is** | | |
| a) .OBJ file | | b) .EXE file |
| c). MASM file | | d) DEBUG file |
| 1. **If the processor is executing a main program that calls a subroutine, then after executing the main program up to the CALL instruction, the control will be transferred to** | | |
| a) address of main program | | b) subroutine address |
| c) address of CALL instruction | | d) none of the mentioned |
| 1. **The Stack is accessed using** | | |
| a) SP register | | b) SS register |
| c) SP and SS register | | d) none |
| 1. **While retrieving data from the stack, the stack pointer is** | | |
| a) incremented by 1 | | b) incremented by 2 |
| c) decremented by 1 | | d) decremented by 2 |
| 1. **The reverse process of transferring the data back from the stack to the CPU register is known as** | | |
| a) pulling out the stack | | b) pushing out the stack |
| c) pulling off the stack | | d) popping off the stack |
| 1. **In the RCL instruction, the contents of the destination operand undergoes function as** | | |
| a) carry flag is pushed into LSB & MSB is pushed into carry flag | | b) carry flag is pushed into MSB & LSB is pushed into carry flag |
| c) auxiliary flag is pushed into LSB & MSB is pushed into carry flag | | d) parity flag is pushed into MSB & LSB is pushed into carry flag |
| 1. **Match the following** | | |
| **a) MOVSB/SW** | | **1) loads AL/AX register by content of a string** |
| **b) CMPS** | | **2) moves a string of bytes stored in source to destination** |
| **c) SCAS** | | **3) compares two strings of bytes or words whose length is stored in C X register** |
| **d) LODS** | | **4) scans a string of bytes or words** |
| a) a-3, b-4, c-2, d-1 | | b) a-2, b-1, c-4, d-3 |
| c) a-2, b-3, c-1, d-4 | | d) a-2, b-3, c-4, d-1 |
| 1. **The instructions that are used to call a subroutine from a main program and return to the main program after execution of called function are** | | |
| a) CALL, JMP | | b) JMP, IRET |
| c) CALL, RET | | d) JMP, RET |
| 1. **In register address mode, the operand is stored in** | | |
| a) 8-bit general purpose register | | b) 16-bit general purpose register |
| c) SI or DI | | d) all of the mentioned |
| 1. **In which of the following addressing mode, the offset is obtained by adding displacement, with the contents of SI?** | | |
| a) direct mode | | b) register mode |
| c) based mode | | d) indexed mode |
| 1. **The statement that is true for the instruction POPA is** | | |
| a) flags are unaffected | | b) no operands are required |
| c) exceptions generated are same as that of PUSHA | | d) all of the mentioned |
| 1. **The instruction that represents the ‘rotate source, count’ is** | | |
| a) RCL | | b) RCR |
| c) ROR | | d) all of the mentioned |
| 1. **While executing the instruction, OUTSW, the SI is incremented by** | | |
| a) 1 | | b) 2 |
| c) 3 | | d) 4 |
| 1. **The stack segment register contains** | | |
| a) address of the stack segment | | b) base address of the stack segment |
| c) pointer address of the stack segment | | d) data in the stack segment |
| 1. **POP operation** | | |
| a) decrements SP | | b) increments SP |
| c) decrements SS | | d) increments SS |
| 1. **When a stack segment is initialized then** | | |
| a) SS and SP are initialized | | b) only SS is initialized |
| c) only SP is initialized | | d) SS and SP need not be initialized |
| 1. **For 8086 microprocessor, the stack segment may have a memory block of a maximum of** | | |
| a) 32K bytes | | b) 64K bytes |
| c) 16K bytes | | d) NONE |
| 1. **The technique that is used to pass the data or parameter to procedures in assembly language program is by using** | | |
| a) global declared variable | | b) registers |
| c) stack | | d) all of the mentioned |
| 1. **If a number of instructions are repeating through the main program, then to reduce the length of the written program without reducing speed, ……. Is used.** | | |
| a) procedure | | b) subroutine |
| c) macro | | d) none of the mentioned |
| 1. **The time required for execution of a macro is ……… that of procedure.** | | |
| a) greater than | | b) less than |
| c) equal to | | d) none of the mentioned |
| 1. **The beginning of the macro can be represented as macro name followed by…. keyword** | | |
| a) START | | b) BEGIN |
| c) MACRO | | d) none of the mentioned |
| 1. **If the data is transmitted only in one direction over a single communication channel, then it is of** | | |
| a) simplex mode | | b) duplex mode |
| c) semi duplex mode | | d) half duplex mode |
| 1. **TXD (Transmitted Data Output) pin carries serial stream of the transmitted data bits along with** | | |
| a) start bit | | b) stop bit |
| c) parity bit | | d) all of the mentioned |
| 1. **In a 32-bit register, ESP, the lower 16-bits of the register can be represented by** | | |
| a) LSP | | b) FSP |
| c) SP | | d) none of the mentioned |
|  | | |
| 1. **The directive used to inform the assembler, the names of the logical segments to be assumed for different segments used in the program is** | | |
| a) ASSUME | | b) SEGMENT |
| c) SHORT | | d) DB |
| 1. **The directive that marks the end of a logical segment is** | | |
| a) ENDS | | b) END |
| c) ENDS & END | | d) None of the mentioned |
| 1. **The recurrence of the numerical values or constants in a program code is reduced by** | | |
| a) ASSUME | | b) LOCAL |
| c) LABEL | | d) EQU |
| 1. **An assembly language program is translated to machine code by** | | |
| a) an assembler | | b) an interpreter |
| c) a compiler | | d) a linker |
| 1. **Number of the times the instruction sequence below will loop before coming out of loop is**   **MOV AL, 00h**  **A1: INC AL**  **JNZ A1** | | |
| a) 00 | | B) 01 |
| C) 255 | | D) 256 |
| 1. **What will be the values of the Sign, and Zero flags after the following instructions have executed?**   **mov ax,620h**  **sub ah,0F6h** | | |
| a) S=0,Z=0 | | b) S=0,Z=1 |
| c) S=1,Z=0 | | d) S=1,Z=1 |
| 1. **Direction flag is used with** | | |
| A) String instructions. | | B) Stack instructions. |
| C) Arithmetic instructions. | | D) Branch instructions. |
| 1. **If AX = 6A7BH what will be the contents of AX after ADD AL, AH** | | |
| a) AX = E56AH | | b) AX = 6AE5H |
| c) AX = E57BH | | d) The instruction is wrong |
| 1. **From quickest to slowest instruction execution time, order these three addressing modes:** | | |
| a) Immediate, indirect, direct | | b) Direct, indirect, immediate |
| c) Immediate, direct, indirect | | d) Indirect, immediate, direct |
| 1. **the register that contains the base address for the code segment is:** | | |
| a) SS | | b) DS |
| c) CS | | d) BX |
| 1. **IF CX =1234H and BX=75FDH what is the value stored in CX after the execution of the following instruction.**   **TEST CX, BX** | | |
| a) 1234H | | b) 77FDH |
| c) 75FDH | | d) 1032H |
| 1. **Given that the subprogram WriteChar displays the character in al, the effect of the following instructions:**   **mov al, ‘c’**  **sub al, 2**  **call WriteChar**  **is to** | | |
| a) display 2 | | b) display 'a' |
| c) display 'c' | | d) display a blank |
| 1. **Let W be an array of Words, one of the following is a correct code to set the fifth element in W to 27** | | |
| a) mov [W+10], 27 | | b) mov [W+2\*4], 27 |
| c) mov [W+5], 27 | | d) None of the above |
| 1. **What will be the final value of ax?**   **mov ax, 6**  **mov ecx, 4**  **L1:inc ax**  **loop L1** | | |
| a) 0B H | | b) 0A H |
| c) 9H | | d) None of the above |
| 1. **Assume that the AX register contains the value 6521 H*.* What will be the contents of AX after execution the instruction:**   **SUB AL, AH** | | |
| a) 65BC H | | b) BC65 H |
| c) BC21 H | | d) 6544 H |
| 1. **The instruction PUSH AL** | | |
| a) Decrement SP by 2 and push a word to stack | | b) Increment SP by 2 and push a word to stack |
| c) Decrement SP by 1 and push a AL to stack | | d) Illegal |
| 1. **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 |
| 1. **If CS = 020AH, SS = 0801H, SI = 0100H and IP = 1BCDH the address of the next instruction is:** | | |
| a) 03C6D | | b) 03C70 |
| c) 03D5D | | d) None of the above |
| 1. **Since the x86 has an address bus of 20-bits, its memory is segmented into 1M segments (i.e. 220).** | | |
| a) True | | b) False |
| 1. **If CS = 7FA2H, SS = 0801H, SI = 0100H and IP = 438EH the address of the next instruction is:** | | |
| a) 83DAEH | | b) 438EH |
| c) 83DA0H | | d) None of the above |
| 1. **One of the following is not a valid segment address** | | |
| a) 00000 | | b) E0840 |
| c) 8CE90 | | d) 8CE91 |
| 1. **Pipelining improves CPU performance due to** | | |
| a) increased clock speed | | b) the introduction of parallelism |
| c) reduced memory access time | | d) additional functional units |

**Q2 – [5] Assume that all flags and registers initialized by zero, show the flag settings for CF, ZF, DF, OF and PF after the following code execution**

MOV AX,1c18H

MOV BX,0A1FFH

ADD BX,AX

XOR AX,AX

STD

ADC SI,0

**Q3- [10] Allocate syntax errors and write V(Valid) or I (Invalid) for each of the following instructions**

|  |  |
| --- | --- |
| 1 | MOV BL,DL |
| 2 | mov [bx],[si] |
| 3 | add dx, [cx+ Yword] |
| 4 | MOV DS, BX |
| 5 | Add AL , BX |
| 6 | POP AH |
| 7 | Mov cx, [BX+SI] |
| 8 | INC CX,2 |
| 9 | INT 1FFH |
| 10 | mov IP, num1 |

**Q4-[6] Write a single instruction for each of the following operations. Note that no other changes should occur.**

1. [1] Multiply AX by 8
2. [3] Three different instructions that will clear the contents of register CL
3. [1] Sets the right most five bits of DI without changing the remaining bits of DI.
4. [1] Invert the seventh bit in DX (note: the LSB is the first bit)

**Q5-[5] Use the following data definitions:**

X1 DB FEh, 33h, 30h, 21h

X2 DW 8h, 3ah, 7fh, 43h, 66h

X3 DD 1, 2, 3, 4, 5

**Fill in the requested register values on the right side of the following instruction sequence:**

MOV ESI, OFFSET X1

1. MOV AL, [ESI] ; **AL** =?
2. MOV AL, [ESI+3] ; **AL** =?

MOV ESI, OFFSET X2+ 2

1. MOV AX, [ESI] ; **AX** =?

MOV EDI, 12

1. MOV EDX, [X3+ EDI] ; **EDX** =?
2. MOV EDX, X3 [EDI] ; **EDX** =?

**Q6- [6] What does the following programs do**

|  |  |
| --- | --- |
| **1** | **2** |
| MOV AH,0  MOV AL,13H  INT 10H  MOV CX,50  MOV AL,5  MOV AH,0CH  BACK: MOV DX,50  INT 10H  MOV DX,100  INT 10H  LOOP BACK  MOV DX,100  BACK2: MOV CX,50  INT 10H  MOV CX,0  INT 10H  DEC DX  CMP DX,50  JNZ BACK2 | .MODEL SMALL  .DATA  V DB 1,4,5,3,2  N DB ?  .CODE  MAIN PROC FAR  MOV AX,@DATA  MOV DS,AX  MOV DL,V[0]  MOV CX,OFFSET N  DEC CX  MOV BX,0  LBL: INC BX  MOV AL,V[BX]  CMP DL,AL  JA LBL2  MOV DL,AL  LBL2: LOOP LBL  MOV N,DL  HLT  MAIN ENDP  END MAIN |

**Q7- A) - [2] What are the contents of AX and BX as exactly four hex digits after executing the following instructions?**

mov ax, 1234h

mov bx, 5678h

xchg ah, bl

**B) - [2] If we have declared**

Alpha DB "ABCDEFG"

**What are the contents of ax as exactly four hex digits after executing the instructions?**

1. mov ax, WORD PTR Alpha
2. mov ax, WORD PTR Alpha+1

(Hint: "A" = 41h, "B" = 42h, etc.)

**C) - [2] What is the value of N after executing the following program**

.MODEL SMALL

.DATA

V DB 1,4,5,3,2

N DB ?

.CODE

MAIN PROC FAR

MOV AX,@DATA

MOV DS,AX

MOV DL,V[0]

MOV CX,OFFSET N

DEC CX

MOV BX,0

LBL: INC BX

MOV AL,V[BX]

CMP DL,AL

JA LBL2

ADD DL,AL

LBL2: LOOP LBL

MOV N,DL

HLT

MAIN ENDP

END MAIN

**D) - [2] What does the following code do? Does the instructions reordering affecting the result?**

XCHG AX,BX

XCHG AX,CX

XCHG AX,DX

**Q8- [Total: 20] Write the following programs**

1. [4] Write a program that takes a number and determine if it's prime or not
2. [4] Write a program that shifts a five elements list and reverse the result. For example, it converts a list from 1,2,3,4,5 to 4,3,2,1,0 and converts 2,3,5,8,7 to 8,5,3,2,0. List is defined at the data segment.
3. [6] Reads two points from the user and draws a line connecting them.
4. [6] Read from user baud rate, file size (in B, KB, MB or GB) and calculate transmission time. Assume 1 stop bit and no parity bits. The program output should be as the following:

Baud rate: 80

File size: 2KB

Transmission time = 256 seconds