Mohamed Magdy Hebashy

802323192

IT

1. What do the following MOV instructions accomplish?

- (a) MOV AX,BX: Copies the contents of register BX into register AX.

- (b) MOV BX,AX: Copies the contents of register AX into register BX.

- (c) MOV BL,CH: Copies the contents of the higher byte of register CX into the lower byte of register BX.

- (d) MOV ESP,EBP: Copies the contents of register EBP into register ESP.

- (e) MOV RAX,RCX: Copies the contents of register RCX into register RAX.

2. List the 8-bit registers that are used for register addressing.

- AH, AL, BH, BL, CH, CL, DH, DL, R8B, R9B, R10B, R11B, R12B, R13B, R14B, R15B, SPL, BPL, SIL, DIL

3. List the 16-bit registers that are used for register addressing.

- AX, BX, CX, DX, SI, DI, BP, SP, R8W, R9W, R10W, R11W, R12W, R13W, R14W, R15W

4. List the 32-bit registers that are used for register addressing in the 80386 through the Core2 microprocessors.

- EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP, R8D, R9D, R10D, R11D, R12D, R13D, R14D, R15D

5. List the 64-bit registers available to the 64-bit mode of the Pentium 4 and Core2.

- RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, R8, R9, R10, R11, R12, R13, R14, R15

6. List the 16-bit segment registers used with register addressing by MOV, PUSH, and POP.

- CS, DS, SS, ES, FS, GS

7. What is wrong with the MOV BL,CX instruction?

- The instruction attempts to move a 16-bit register (CX) into an 8-bit register (BL), which is not allowed.

8. What is wrong with the MOV DS,SS instruction?

- Directly moving between segment registers is not allowed.

9. Select an instruction for each of the following tasks:

- (a) copy EBX into EDX: MOV EDX, EBX
- (b) copy BL into CL: MOV CL, BL
- (c) copy SI into BX: MOV BX, SI
- (d) copy DS into AX: MOV AX, DS
- (e) copy AL into AH: MOV AH, AL
- (f) copy R8 into R10: MOV R10, R8

10. Select an instruction for each of the following tasks:

- (a) move 12H into AL: MOV AL, 12H
- (b) move 123AH into AX: MOV AX, 123AH
- (c) move 0CDH into CL: MOV CL, 0CDH
- (d) move 1000H into RAX: MOV RAX, 1000H
- (e) move 1200A2H into EBX: MOV EBX, 1200A2H

11. What special symbol is sometimes used to denote immediate data?

- The special symbol is the (#)

12. What is the purpose of the .MODEL TINY statement?

- The .MODEL TINY statement indicates that the program is a small, single-segment executable (.COM file).

13. What assembly language directive indicates the start of the CODE segment?

- The CODE segment is typically indicated by the .CODE directive.

14. What is a label?

- A label is a symbolic name for a memory address used to mark a location in the code or data segment.

15. The MOV instruction is placed in what field of a statement?

- The MOV instruction is placed in the opcode field of a statement.

16. A label may begin with what characters?

- A label may begin with a letter (A-Z, a-z) or an underscore (_),@,$ and ?.

17. What is the purpose of the .EXIT directive?

- The .EXIT directive indicates the end of the program and causes the assembler to generate the appropriate code to terminate the program.

18. Does the .MODEL TINY statement cause a program to assemble as an execute (.EXE) program?

- No, the .MODEL TINY statement causes the program to assemble as a .COM file, not a .EXE file.

19. What tasks does the .STARTUP directive accomplish in the small memory model?

- The .STARTUP directive initializes the stack pointer and sets up the data segment registers, preparing the program to start execution.

20. What is a displacement? How does it determine the memory address in a MOV DS:[2000H],AL instruction?

- A displacement is an offset value used to calculate an effective address. In MOV DS:[2000H],AL, the displacement is 2000H, and the effective address is determined by adding this displacement to the base address of the DS segment.

21. What do the symbols [ ] indicate?

- The symbols [ ] indicate indirect addressing, meaning the enclosed value is an effective address to be accessed in memory.

22. Suppose that DS = 0200H, BX = 0300H, and DI = 400H. Determine the memory address accessed by each of the following instructions, assuming real mode operation:

- (a) MOV AL,[1234H]: The effective address is2000+1234= 3234H.

- (b) MOV EAX,[BX]: The effective address is 02000+0300H=2300H.

- (c) MOV [DI],AL: The effective address is 02000+0400H=2400H.

23. What is wrong with a MOV [BX],[DI] instruction?

- The instruction is incorrect because it attempts to move data directly between two memory locations, which is not allowed.

24. Choose an instruction that requires BYTE PTR.

- MOV BYTE PTR [BX], AL

25. Choose an instruction that requires WORD PTR.

- MOV WORD PTR [BX], AX

26. Choose an instruction that requires DWORD PTR.

- MOV DWORD PTR [BX], EAX

27. Select an instruction that requires QWORD PTR.

- MOV QWORD PTR [BX], RAX

28. Explain the difference between the MOV BX,DATA instruction and the MOV BX,OFFSET DATA instruction.

- MOV BX,DATA loads the value stored at the memory location labeled DATA into BX. MOV BX,OFFSET DATA loads the address (offset) of the memory location labeled DATA into BX

29. Suppose that DS = 1000H, SS = 2000H, BP = 1000H, and DI = 0100H. Determine the memory address accessed by each of the following instructions, assuming real mode operation:

- (a) MOV AL,[BP+DI]: The effective address is 10000+1000H +0100=11100H(since BP is in SS,DI is in DS use DS).

- (b) MOV CX,[DI]: The effective address is 10000+0100H=10100H.

- (c) MOV EDX,[BP]: The effective address is 20000:1000H=21000H.

30. What, if anything, is wrong with a MOV AL,[BX][SI] instruction?

- The instruction is syntactically correct. It is equal MOV AL, [BX+SI].

31. Suppose that DS = 1200H, BX = 0100H, and SI = 0250H. Determine the address accessed by each of the following instructions, assuming real mode operation:

- (a) MOV [100H],DL: The effective address is 12000+0100=12100H.

- (b) MOV [BX+100H],EAX: The effective address is 12000+0100+0100=12200H.

- (c) MOV DL,[SI+100H]: The effective address is 12000+0250+0100=12350H.

32. Suppose that DS = 1100H, BX = 0200H, LIST = 0250H, and SI = 0500H. Determine the address accessed by each of the following instructions, assuming real mode operation:

- (a) MOV LIST[SI],EDX: The effective address is 11000+0250+0500=11750H.
- (b) MOV CL,LIST[BX]: The effective address is 11000+0250+0200=11450H.
- (c) MOV CH,[BX+SI]: The effective address is 11000+0200+0500=11700H.

33. Suppose that DS = 1300H, SS = 1400H, BP = 1500H, and SI = 0100H. Determine the address accessed by each of the following instructions, assuming real mode operation:

- (a) MOV EAX,[BP+SI]: The effective address is 13000+1500+0100=14600H.
- (b) MOV AL,[BP+SI+100H]: The effective address is 13000+1500+0100+100=14700H.
- (c) MOV AL,[SI+100H]: The effective address is 13000+0100+100=13200H.

34. Which base register addresses data in the stack segment?

- The BP (Base Pointer) register addresses data in the stack segment.

35. Suppose that DS = 1300H, SS = 1400H, BP = 1500H, and SI = 0100H. Determine the addresses accessed by the following instructions, assuming real mode operation:

- (a) MOV ECX,[BP+SI]: The effective address is 14000+1500+0100=15600H.
- (b) MOV [BP+SI],CL: The effective address is 14000+1500+0100=15600H.
- (c) MOV DH,[BP+100H]: The effective address is 14000+1500+100=15600H.

36. Develop a data structure that has five fields of one word each named F1, F2, F3, F4, and F5 with a structure name of FIELDS.

FIELDS STRUC F1 DW ? F2 DW ? F3 DW ? F4 DW ? F5 DW ? FIELDS ENDS

37. Show how field F3 of the data structure constructed in question 36 is addressed in a program.

If FIELDS starts at 1000H and SI is used as the base register:

MOV AX, [SI + FIELDS.F3]

38. What are the three program memory-addressing modes .?

Direct -

Indirect -

Indexed -

39.How many bytes of memory store a far direct jump instruction? What is stored in each of the byte

A far direct jump instruction typically stores 5 bytes: 1 byte for the opcode and 4 bytes for the - segment and offset

40. What is the difference between an intersegment and intrasegment jump?

Intrasegment jump: A jump within the same code segment. •

Intersegment jump: A jump to a different code segment, changing both the segment and offset. •

41. If a near jump uses a signed 16-bit displacement, how can it jump to any memory location within the current code segment?

A near jump with a signed 16-bit displacement can jump within a range of -32,768 to +32,767 bytes relative to the instruction pointer.

42. The 80386 and above use a 32-bit displacement to jump to any location within the 4G-byte code segment.

43. What is a far jump?

A far jump is a jump to a different segment and offset, allowing the CPU to change both the segment and the instruction pointer.

44. If a JMP instruction is stored at memory location 100H within the current code segment, it cannot be a short jump if it is jumping to memory location 200H within the current code segment.

45. Show which JMP instruction assembles (short, near, or far) if the JMP THERE instruction is stored at memory address 10000H and the address of THERE is:

(a) 10020H: Short jump (within -128 to +127 bytes) (b) 11000H: Near jump (within -32,768 to +32,767 bytes) (c) 0FFFEH: Near jump (within -32,768 to +32,767 bytes) (d) 30000H: Far jump (requires changing the segment)

46. Form a JMP instruction that jumps to the address pointed to by the BX register.

47. Select a JMP instruction that jumps to the location stored in memory at the location TABLE. Assume that it is a near JMP.

48. How many bytes are stored on the stack by a PUSH AX?

The PUSH AX instruction stores 2 bytes (the size of the AX register) on the stack.

49. Explain how the PUSH [DI] instruction functions.

The PUSH [DI] instruction is used to push the value located at the memory address specified by the DI register onto the stack. Here’s a step-by-step breakdown of how it works:

The current value of the stack pointer (SP or ESP) is decremented by 2 (in 16-bit mode) or 4 (in 32-bit mode) to make space on the stack. .1

The value located at the memory address pointed to by DI is read. .2

This value is then stored at the new address pointed to by the stack pointer.

?What registers are placed on the stack by the PUSHA instruction? In what order .50

.The registers are pushed in the following order: AX, CX, DX, BX, SP, BP, SI, DI -

51. What does the PUSHAD instruction accomplish ?

PUSHAD pushes all the general-purpose registers (EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI) onto the -
.stack in that order

52. Which instruction places the EFLAGS on the stack in the Pentium 4 microprocessor?

PUSHFD -

53. Is a PUSHA available in the 64-bit mode of the Pentium 4 or the Core2?

.No, PUSHA is not available in 64-bit mode -