Nicholas Martinez

Cosci 416 - M 5:45 - 10:00PM

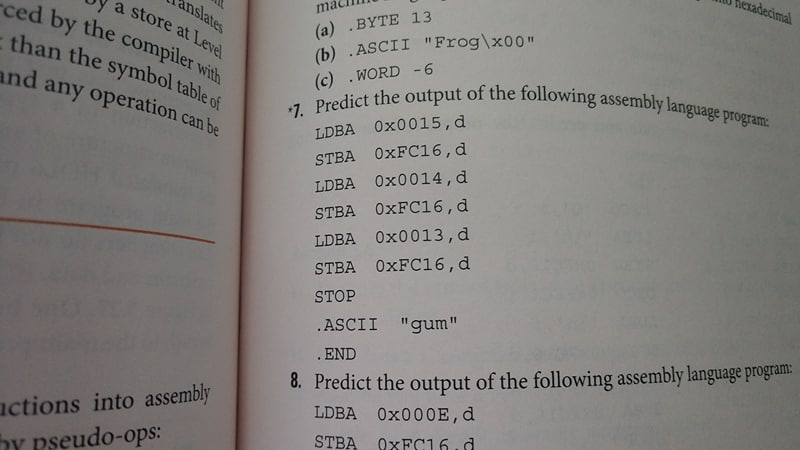
Mari Rettke

April 15th 2018

Section 5.1 Exercises 1-9

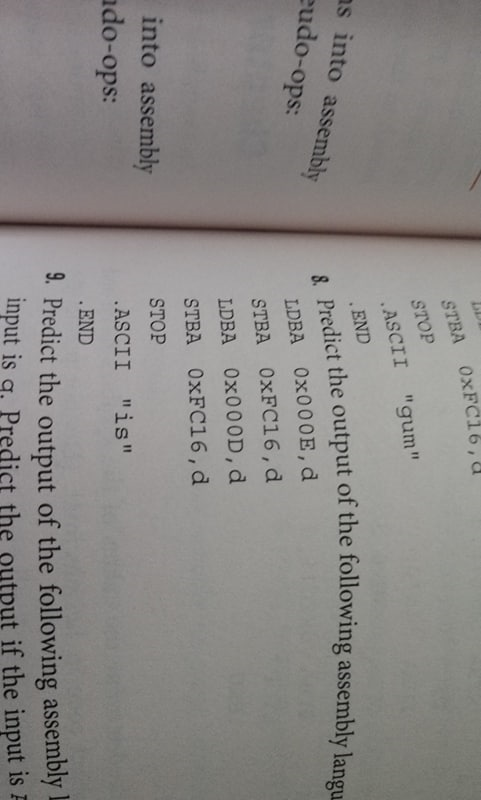
1. Convert the following machine language instructions into assembly language, assuming that they were not generated by pseudo ops.
2. 9AEF2A -> 10011010 -> ORX 0xEF2A, n
3. 03 -> 00000011 -> MOVSPA
4. D7003D -> 11010111 -> LDBA 0x003D, sfx
5. Convert the following machine language instructions into assembly language, assuming that they were not generated by pseudo ops.
6. 82B7DE -> 10000010 -> ANDA 0xB7DE, n
7. 04 -> 00000100 -> MOVFLGA
8. DF63DF -> 11011111 -> LDBX 0x63DF, sfx
9. Convert the following assembly language instructions into hexadecimal machine language:
10. ASLA -> 00001010 -> 0A
11. DECI 0x000F, s -> 00110000 -> 30000F
12. BRNE 0x01E6, I -> 00011010 -> 1A01E6
13. Convert the following assembly language instructions into hexadecimal machine language:
14. ADDA 0x01FE. I 01100000 -> 6001FE
15. STRO 0x000D, sf -> 01001100 -> 4C000D
16. LDWX 0x01FF, s -> 11001011 -> CB01FF
17. Convert the following assembly language pseudo-ops into hexadecimal machine language:
18. .ASCII “Bear\x00” -> 42 65 61 72 00 -> 01000010 01100101 01100001 01110010 00000000
19. .BYTE 0xF8 -> 11111000
20. .WORD 790 -> 0316 -> 0000001100010110
21. Convert the following assembly language pseudo-ops into hexadecimal machine language
22. .BYTE 13 -> D -> 1101
23. .ASCII “Frog\x00” -> 46 72 6F 67 00 -> 01000110 01110010 01101111 01100111 00000000
24. .WORD -6 -> FFFA -> 11111111 11111010

1. Predict the output of the following assembly language program:



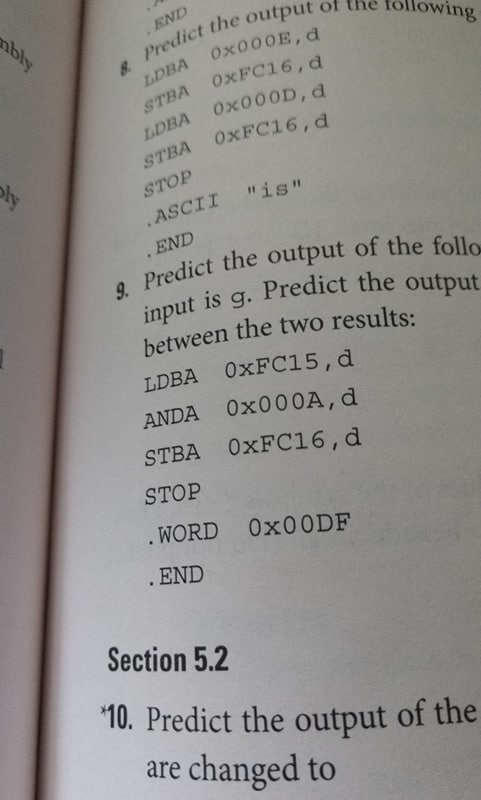
Prints the ASCII text “gum” in reverse order. Since its loading the bytes from the ASCII from descending addresses.

1. Predict the output of the following assembly language program:



Does exactly the same thing as the previous program, reversing the order of the ASCII text to the output device.

1. Predict the output of the following assembly language program:

 If the input of the program is “g”, then “g” will be AND’d with 00DF, which will produce “G”.

If the input of the program is “A”, “A” will be AND’d with 00DF and produce “A”, since the bit patterns of A remain unchanged from the AND operation.