

Extra Credit #2

Out: April 25, 2023

Due: May 2, 2023 @ 3:00 PM (start of lecture)

Code Optimization (30 points)

For this assignment there is no collaboration allowed

For the following code fragments, rewrite or write them to minimize the number of lines and/or bytes of code. That is, the number of lines of code and/or the number of bytes the assembled code generates. The operation of the code is correct (there are no errors) and must not be changed, nor may any additional registers be used or changed (except additional registers may be used in problem #4). The code is only required to have the effects listed in the comments. Other "side effects" do not need to be replicated and no new "side effects" should occur. Be sure to test the code syntax in the assembler.

Assume each problem is independent and assume `Buffer` is the label of a byte in the data segment and `WIN_DIGIT` is a constant between 0 and 7.

1) Negation (5 points)

```
;R16 <- -R16  
COM R16  
INC R16
```

2) Reading Memory (5 points)

```
;R16 <- value stored at Buffer + WIN_DIGIT  
LDI YL, LOW(Buffer)  
LDI YH, HIGH(Buffer)  
LDI R17, WIN_DIGIT  
LDI R18, 0  
ADD YL, R17
```

```
ADC YH, R18  
LD R16, Y
```

3) Set SREG bit (5 points)

```
;set bit 7 in SREG  
IN R16, SREG  
ORI R16, 0b10000000  
OUT R16, SREG
```

4) Shifting R17 | R16 by 4 bits (15 points)

Given: R17 | R16 - 16-bit unsigned value (FEDCBA98 | 76543210) to shift right by 4 bits

Result: R17 | R16 shifted right by 4 bits (xxxxFEDC | BA987654, the upper nibble of R17 can be any value)

Code: must use fewer than 8 instructions

Resources

- [Homework Q&A](#)

Last updated April 12, 2023 03:35 PM by glen@caltech.edu

copyright © 2019 - 2023, [Glen George](#). All rights reserved. Reproduction of all or part of this work is permitted for educational or research use only, provided that this copyright notice is included in any copy.