**51LAB105 test-3.1** 2016-10-20

= = = = = = = = = = = = = = = = = = = = = = = = =

Consider a block of 24 data bytes, BLK24, starting at 30H of built-in data space of 51 microprocessor. Let ***addrj***, j=0-23, be the addresses, in ascending order, of the 24 data bytes.

[Q] Let k = mod(***addrj***, 3), j=0-23;

a) write a short 51-code for fulfilling all the

requests below.

\* saving those data bytes, for which k=0, of BLK24 in an 8-byte area, BLKk0, starting at 50H; and bit[1:0] of all data bytes in BLKk0 should be marked with 3.

\* saving those data bytes, for which k=1, of BLK24 in an 8-byte area, BLKk1, starting at 58H; and bit[7:6] of all data bytes in BLKk1 should be marked with 2.

\* saving those data bytes, for which k=2, of BLK24 in an 8-byte area, BLKk2, starting at 60H; and bit[4:3] of all data bytes in BLKk2 should be marked with 1.

b) tabulate the source code lines of yours from a) and corresponding machine code bytes in a neat, clean way, as done in lecture materials, with starting address specified for every instruction in your code.

[ a line in the tabulation may look like

. . .

source-line addr: code-byte

. . .

]

; a plausible coding for a)

ORG 0000H

mov SP, #70H

again:

mov R0, #30H

mov R1, #50H

mov R2, #8

mov\_next:

push 01H

mov A, @R0 ;;; mod-0 transfer

orl A, #3 ;;; 3-marking

mov @R1, A

inc R0 ; next source

xch A, R1

add A, #8

xch A, R1

mov A, @R0 ;;; mod-1 transfer

orl A, #80H ;;; 2-marking

anl A, #0BFH

mov @R1, A

inc R0; ; next source

xch A, R1

add A, #8

xch A, R1

mov A, @R0 ;;; mod-2 transfer

orl A, #08H ;;; 1-marking

anl A, #0E0H

mov @R1, A

inc R0; ; next source

pop 01H

inc R1

djnz R2, mov\_next

jmp again

END