

# EE333 Microcontroller Engineering

## Oregon Tech Portland, Fall 2012

### Homework Assignment #2 Due October 9

Each of the problems in this assignment use the following data declarations:

```
    org  $1000      ; They all start at location $1000
P:   db  $10        ; byte variable at location $1000
Q:   db  $F0        ; byte variable at location $1001
R:   dw  $1234      ; word variable at location $1002
S:   dw  $fedc      ; word variable at location $1004
T:   dw  R          ; address of variable R is here
    org  $2000      ; Code starts at location $2000
; code to implement answer goes here
```

In each of the following problems, answer with the code necessary to solve the question. You may use any registers not mentioned in the problem statement for anything you wish. You may not alter any of the variables except as mentioned in the problem statement, nor may you alter any other data memory location. You can take the data declarations above and add your code and then test for the correct result using the simulator. No partial credit will be given on any answer. Check your work!

Two problems have been done for you.

#### 1. Exchange the two bytes in variable R

```
    org  $1000      ; They all start at location $1000
P:   db  $10        ; byte variable at location $1000
Q:   db  $F0        ; byte variable at location $1001
R:   dw  $1234      ; word variable at location $1002
S:   dw  $fedc      ; word variable at location $1004
T:   dw  R          ; address of variable R is here
    org  $2000      ; Code starts at location $2000
    ldd  R
    exg  A B
    std  R
```

When this program is run (starting at location \$2000), variable R gets the value \$3412.

- Using a single instruction with immediate addressing mode, store the value \$3355 in variable S.
- Using a single instruction, copy the contents of variable S to variable R.
- Put the address of variable R in register X, then using an indexed addressing mode on register X, load register Y with the contents of variable R.

5. Put the address of variable R in register X, then using an indexed addressing mode on register X, load register Y with the contents of variable S.
6. Put the address of variable R in register X and the address of variable S in register Y, then using only the movb instruction and the auto post-increment indexing mode copy the contents of variable R to variable S.
7. Copy the contents of unsigned byte variable Q to variable S.

```

    org  $1000      ; They all start at location $1000
P:   db  $10       ; byte variable at location $1000
Q:   db  $F0       ; byte variable at location $1001
R:   dw  $1234     ; word variable at location $1002
S:   dw  $fedc     ; word variable at location $1004
T:   dw  R         ; address of variable R is here
    org  $2000     ; Code starts at location $2000
    ldab Q
    clra
    std  S

```

When this program is run (starting at location \$2000), variable S gets the value \$00f0.

8. Copy the contents of signed byte variable Q to variable S.
9. Exchange the values stored in variables R and S.
10. Copy the least significant byte of R to P.
11. Load the value \$6791 into accumulator D. Load the value 0 into register X. Then store that value at the location which is in variable T. (Use indirect addressing mode.)
12. Copy the unsigned value in P into both both R and S using the movb and movw instructions each once, and a single additional instruction.