

**ELEC 2220 Computer Systems**  
**Homework #7**  
**Due: Wednesday, June 9**

It is assumed that you have already obtained and installed a copy of the *CodeWarrior* integrated development environment (IDE) on your PC from the CD in your text book (or from my office). For your convenience, *CodeWarrior* is also installed on the ECE Department PCs in Broun Hall rooms 308 and 310, and the PCs in the College of Engineering computing labs.

1. Write and test a short assembly language program to perform the following computation. Use direct/extended addressing for all memory references.

$$A1 = (B1 - C1) - (D1 + 8)$$

- a. In *CodeWarrior*, create a new “project”.
  - (1) Select **File > New** from the menu bar.
  - (2) Select the “HC(S)12 New Project Wizard” and enter a project name (example: homework1).
  - (3) Select the MC9S12C32 derivative from the list of microcontrollers.
  - (4) Check “Assembly Language” (and not C or C++).
  - (5) Check “Absolute Assembly” (not Relocatable Assembly).
  - (6) Check “Full Chip Simulation”.You should see a project summary window to the left of an edit window.
- b. In the project window, expand “Sources” and double-click on “main.asm” to open the “skeleton” assembly language file.
  - (1) At the top, replace the “comment block” with one that lists your name, date, and brief description of the program.
  - (2) Delete the sample data definitions and code, and replace them with your program. Make sure the “org” statements define the start of RAM and ROM for the MC9S12C32.
  - (3) Define variables A1, B1, C1, and D1 as 8-bit integer constants in the RAM section. You will provide initial values in the debugger.
  - (4) Enter your program in the ROM area.
- c. Instruct the assembler to produce a “listing file”, showing assembled code, memory addresses, etc.
  - (1) Click on the “Full Chip Simulation Settings” button at the top of the project summary window.
  - (2) Select “Assembler for HC12”.
  - (3) Click on “Options”.
  - (4) Check “Generate a listing file”.
  - (5) Click OK in both windows.
- d. Assemble the program with the “make” command. Click on the “Make” button at the top of the project summary window or from the menu bar, select Project > Make, or press key F7. If any errors are reported by the assembler, fix them and reassemble the program.
- e. Open and examine the listing file. This file will have the name *main.lst*, and can be found in “bin” directory of the project. Open this file by selecting File > Open from the menu bar, and locating the bin directory.
- f. Open the debugger to test the program. Click on the “Debug” button at the top of the project summary window or from the menu bar, select Project > Debug, or press key F5.
- g. In the memory window, scroll down to the data area beginning at \$0800, so you can view changes to data variables.
- h. Provide the following initial values: A1 = \$12, B1 = \$45, C1 = \$FB, D1 = \$EF by entering them in the memory window (click on each current value to select it, and enter the new value.)
- i. Execute the program one step at a time, watching the values of the registers and memory locations. (Click on the “step” button to single-step.)
- j. After executing the last instruction, copy and paste the debugger window to a Word document (ALT-Print Screen will capture the debug window to the Windows “clip board”, and then “paste” in a Word document). In both the “Data” window and the “Memory” window, highlight or circle the final value of A1 (use a yellow highlighter or some other means to point out the results.)
- k. Print and turn in your assembly language source program and the captured screen.

2. Rewrite the program from part 1, modifying it to use indexed addressing for all memory references. Repeat steps a through k above.