ELT 220
Investigation of Addressing Modes Using Move Instructions, rev. 1

Objectives:

This experiment should familiarize the student with the basic addressing modes of the 8051 family of microprocessors. This will be accomplished using the mov instruction.

Procedure:

You will write a section of code to accomplish each of the tasks below. In each case test out your program as appropriate to assure that it works. As appropriate, list the contents of the registers and/or memory locations at each step. Have your data initialed by the instructor before unplugging your 8051 board.

After you write the program and single step through it it, prepare a report of what you have done and what data was taken. Include a listing of your program showing all op-codes and explain. You must clearly explain what you did and what you saw as a result.

Using the 8051IDE and the template you made in the first IDE lab, write six (6) programs to do the tasks given below. Note that each task must be taken separately and sequentially. That is you should write and test each one (see Procedure and Report section) **before** you go on to the next.

ORG your code at 0000h

1) As an example of an *immediate addressing mode* command, put the data OFFh into the A register.

1. Move the literal value 55h into the accumulator. As an example of *register addressing mode* move the byte located in accumulator into register R1.
2. Equate a memory location 60h to the label *data1.* Equate a memory location 61h to the label *data2.* Move a literal value of your choice into the accumulator. As an example of a *direct addressing mode* command, move the byte in the accumulator to location 70h. As an example of a *direct addressing mode* command using a label, move the byte in the accumulator to location 61h using the label *data2.*
3. Move a literal value of your choice into the accumulator. As an example of an *indirect addressing mode* command use the R1 register as a pointer to move the data in the accumulator into the location 55h. Note that this may involve several steps to initialize the registers before the actual command is executed.
4. Write a program that will assign the value 22h to the label datal, assign the value 33h to the label data2, assign the value 44h to the label data3, store the value datal in register R0, store the value data2 in register R1, store the value data3 in internal memory location 30h. The program should then add the contents of R0, RI. and internal memory location 30h. The result of the addition should be stored in memory location 40h. In writing this program think about how to write the program in order to minimize the changes needed to add different data values.
5. Modify the program in (5) to have three different data values, of your choice, for datal, data2 and data3. Run the modified program and verify its operation.

**Procedure** for each program:
For EACH of the six programs, Assemble your programs and print a listing (.LST) before you run them. Load them into EdSim51 and step through them noting the contents of relevant registers and memory locations before and after each step. Note the discussion points in the Report part (below) and *be sure you have enough data to illustrate them*. At a minimum you must show the relevant registers and memory locations before the program was stepped through and after to see the changes in the registers and memory locations. You may wish to include additional screen captures showing intermediate steps.

REPORT
For programs 1 through 4, Write a report commenting on how your assembly language commands were translated into machine language (refer to the ".LST" file). Be sure to include in your discussion:

1. How the assembler assigns addresses relative to the instruction size in bytes.
2. How are the op-codes formed; where do the addresses and/or data of the immediate and/or direct commands appear in the op-code. Comment on the differences in the form of the op-codes produced by the various addressing modes.
3. What happened when you ran each step? Comment on the contents of the register(s) and memory before and after. State whether or not you felt each program ran successfully and refer to your data to verify your statements.

For programs 5 and 6, Include the .LST files and comment on each line, as appropriate, saying the effect of it. It may be appropriate to put those comments on each line as non-assembling comments.

Report Format

1. Cover Page — (Sample downloadable from the ELT220 Lab page, www.donwade.us/220)

1. Four program pages following the format on the annotation handout. 1-3 should be answered on the program pages.
2. For programs 5 and 6, answer questions asked in the procedure and screen shots showing and commenting on the results of each step.
3. Original data (screen captures) if not pasted directly to the program pages.

4/2/20