Laboratory 5

Subroutines

Due Date: Beginning of Week 8 Lab

Introduction

Subroutines are useful for a variety of reasons. This lab will explore some useful subroutines and different methods of passing parameters to subroutines in general.

Assignment

The follow section walks though a main program that calls existing subroutines. Note that the three subroutines used are supplied only as .s19 files with instructions on how to use them. This is a common method of distributing intellectual property, much like buying a program to run on your home PC. You have the executable code and a manual, but you are not given the source code. Most End User License Agreements, which you are legally required to agree to before using the program, often prohibit reverse-engineering the executable (i.e. converting the machine code to assembly code to see how the program actually works).

- Download the files 7segment.s19 to the Dragon12+ board.
- Download the file 7segdemo.asm, assemble it, and download the resulting .s19 file to the Dragon12+ board.
- Run the program beginning at $2200.
  Question 1: What do you see on the 7-segment display?
- Reset the board (there is no way to stop this program) and set a breakpoint for line 34 of the .lst file
- Run the program from the beginning.
  Question 2: What do you see on the 7-segment display?
- Remove the breakpoint, set a new one at line 37 of the .lst file, and continue execution.
  Question 3: What do you see on the 7-segment display?
- Remove the breakpoint, set a new one at line 40 of the .lst file, and continue execution.
  Question 4: What do you see on the 7-segment display?
- Remove the breakpoint, set a new one at line 43 of the .lst file, and continue execution.
  Question 5: What do you see on the 7-segment display?

Question 6: Based on what you've seen in the previous questions, describe briefly how the 7-segment display works and why the demo program loops.

Write a flowchart and the corresponding assembly subroutine that meets the following requirements. Demonstrate the working program to the instructor.

1. The subroutine converts a 2-byte hexadecimal number to a 4-byte ASCII string (This is NOT a null-terminated string).
2. The subroutine must start at address $2C00.
3. The hexadecimal number is supplied by reference and is passed on the stack. The hexadecimal number is limited to the range 0 to 9999_{10}. (Your subroutine may assume the value is in this range and does not need to check bounds.)
4. The ASCII string is returned by reference on the stack.
5. The subroutine is responsible for preserving all registers that it affects.
6. No addresses between $3000 and $34FF may be used.

Write a main program that meets the following requirements. You should refer to the program and subroutines from the first section as a reference.

1. The program should take the two-byte unsigned hexadecimal number supplied in $1000 and $1001 and output it to the Dragon12+ 7-segment display.
2. No addresses between $3000 and $34FF may be used.

Extra Credit: In addition to the above assignment, write a subroutine and main program with the previous requirements, but interpret the hexadecimal value as a signed value within the range of -999 and 999. As part of this, you will have to modify the toLED.asm file that generates the 7-segment display patterns.

What to Demonstrate/Submit

- Typed answers to questions.
- Listing files.
- Instructor check-off.