Laboratory 6

Input and Output

Due Date: Beginning of Week 8 Lab

Introduction

Microcomputers must often interact with the “outside world”, meaning that they must receive inputs and generate outputs. This lab will walk you through three of the general-purpose I/O ports of the MC9S12DG256 used on the Dragon12+ evaluation board.

Assignment

The follow section walks though Ports B, H, and P on the Dragon12+ board.

- Reset the board, then change addresses $258 and $25A to $00.
- Change the value of DDRB ($0003) to $FF.
- Change the value of PortB ($0001) to $FF.
  Question 1: What do you see on the 7-segment displays?
- Change the value of PortB to $4F.
  Question 2: What do you see on the 7-segment displays?
- Change the value of PortB back to $FF, then change the value of DDRB to $71.
  Question 3: What do you see on the 7-segment displays, and why is this displayed if PortB is $FF?
- Change DDRB to $FF, then change PortB to $FF. Change DDRP ($025A) to $0F, and change PTP ($0258) to $00.
  Question 4: What do you see on the 7-segment displays?
- Change PTP to $07.
  Question 5: What do you see on the 7-segment displays?
- Change PTP to $0B.
  Question 6: What do you see on the 7-segment displays?
- Change PTP to $0D.
  Question 7: What do you see on the 7-segment displays?
- Change PTP to $0E.
  Question 8: What do you see on the 7-segment displays?
- Change PTP to $05.

Question 9: What do you see on the 7-segment displays?

- Change DDRH ($0262) to $00, and make sure that all eight DIP switches are in the upper position.

  Question 10: What value is in PTH ($0260)?
- While holding down the pushbutton "SW5", do a memory display for PTH.

  Question 11: What value is in PTH?
- Release the pushbutton SW5, slide the right-most DIP switch (switch 1) into the lower (i.e. "on") position, and then do another memory display for PTH.

  Question 12: What value is in PTH?

  While switch 1 is still in the "on" position, press and hold pushbutton SW5, and do a memory display for PTH.

  Question 13: What value is in PTH?

  Question 14: Describe the behavior of bit 0 of PTH in terms of switch 1 and pushbutton SW5.
- Experiment with the pushbuttons and the DIP switches to observe their effects on PTH.
- Save the file Lab6Prog1.asm to your Flash drive, assemble it, and download it to the Dragon12+ board. Run the program at $2000.
- Press the pushbutton SW4 several times and notice the effect.

  This program uses several common concepts to interface with a mechanical switch. The first concept is debouncing. When a physical switch is pressed, the mechanical contacts act like a spring. The contacts literally "bounce" several times, thus making and breaking the electrical path, before coming to rest. This causes the electrical signal to toggle several times before reaching a steady value. This "bounce" may be incorrectly interpreted as several presses instead of one noisy one. To "debounce" the input signal, the supplied program generates a pause after seeing a signal transition. If the signal has the same new value after the delay, the transition is registered. Otherwise, it is treated like noise and ignored.

- Based on the .lst file, comment out lines 37 through 41 and 48 through 52. These lines generate the pause.
- Assemble the modified program, download it, and run it. Press the button SW4 several times.

  Question 15: Is there a noticeable change in the behavior? If so, what?
- Comment out line 47, then recompile, download, and run the program. Press the button SW4 several times.

  Question 16: Is there a noticeable change in the behavior? If so, what?

The previous question should point out the second concept. A program should monitor a button to see that it has been released after it has been pressed. Otherwise, the program will perform the action once for each time the button is checked, not just once for each time the button is pressed.

The next program uses all four push buttons to control the different LED 7-segment displays. The approach used in the first program won't quite work. It continuously polled the pushbutton to look for a change. With four digits, the S12 needs to repeatedly display each digit (called time division multiplexing), and the program can't get stuck in a tight polling loop staring at one button, much less four. In the second program, the processor stores the value of a button each time it is examined. If the button was not pressed in the previous loop and it is pressed in the current loop, the processor has detected a new button press, and the corresponding digit will be updated.

- Download Lab6prog2.asm, assemble it, download it, and run it at $2000. Experiment with the four pushbuttons and notice the behavior of the program.
  Examine the assembly code for the second program. Notice that there is a "JSR pause" subroutine call after each digit is displayed. One reason for this is to make the main loop consume enough time between iterations to debounce the buttons.
- Comment out the four lines containing "JSR pause", reassemble, download, and run the program. Experiment with the pushbuttons.
  Question 17: Explain the second reason for the pause after each 7-segment pattern is output.
  Question 18: Other than generating a pause after updating the 7-segment displays, describe a second method to eliminate the effect observed above.

What to Demonstrate/Submit

- Answers to questions.