LAB #4: Keypad 32-Bit RPN Calculator

The Lab write-up is due to your TA at the beginning of your next scheduled lab. Don't put this off to the last minute! There is pre-lab work to complete before the start of the next lab. **NO LATE LAB REPORTS WILL BE ACCEPTED.**

1 Objectives

- Design the hardware interface between a keypad and microcomputer.
- Create the low-level device driver that can be used in other applications.
- Implement the 32-bit Reverse Polish Notation Calculator with Keypad input.

2 Background

This lab demonstrates how a microcomputer can be used to control a keypad matrix via a parallel interface. Your low-level software should input, scan and debounce keystrokes. Additionally, this lab demonstrates how 32-bit arithmetic operations are implemented on a native 16-bit extended architecture.

In this lab, you will expand the functionality of your Reverse Polish Notation calculator from Lab3. However, you will need to extend the functionality of your calculator to support 32-bit add, 32-bit subtract, and 32-bit multiply. You will not need to implement a divide function at all, so you may ignore divide for this calculator. You will still need to handle all the error conditions from the previous lab. Overflow should be detected as an overflow for a 32-bit result, and Underflow should be detected as an underflow for a 32-bit result.

Your 32-bit calculator will also accept input from the keypad instead. You will need only five push buttons for input for this lab. One for "push", one for "pop", one for "add", one for "subtract", and one for "multiply". You may use any of the available push buttons for input. You must use the keypad as your numerical input values. Your inputs will be interpreted as 32-bit **unsigned** integers and you will be performing 32-bit **signed** arithmetic. Since your input should be 32-bits, your implementation should handle an input of "131071" for instance when pushed onto your stack. Finally, you should implement loop polling as a Gadfly loop when reading input from the keypad.

As mentioned before, you will need to handle all error conditions from the previous lab, Lab #3. It is also recommended that you use Port T as the interface to your keypad, as it contains an internal pull-up, pull-down network.

3 Parts

This lab will require a 12-key matrix keypad (Grayhill 96AB2-102-R) which you can purchase in the ECE lab.

4 Pre-lab

Complete tasks 1 and 2 before coming to lab.

5 Tasks

1. Prepare a schematic for your design including all components used. (Note: if you do not use Port T, then you will need pull-up resistors on all the keypad inputs).
2. Write your low-level keypad device driver.
3. Extend your Lab#3 functionality by implementing 32-bit add, 32-bit subtract, and 32-bit multiply arithmetic operations instead.

6 Writeup

Include the following items:

1. Your hardware schematic
2. A description of your design and why you chose to implement it that way.
3. A printout of your well-commented code.