Experiment 4 Fall 2012

What's My UIN? with improved output

Due: Week 6 Lab Seeions (10/02)

Go over experiment 1 for a complete explanation of delay loops. You will need to implement delays in experiment 4.

Design a system that will output the decimal digits of your UIN on a Seven Segment display.

A START/STOP DISPLAY signal is required to begin displaying the digits. The output should be blank until the START/STOP DISPLAY signal is pulsed LOW.

When the START/STOP DISPLAY signal is pulsed LOW the digits should display one digit at a time starting with the most significant digit.

After the digits begin being displayed the START/STOP DISPLAY signal should "toggle" the display output. So, after digits begin displaying if START/STOP DISPLAY is pulsed LOW momentarily the display should hold the current digit. If START/STOP DISPLAY is again pulsed LOW momentarily the digits continue to be displayed one second each as before.

Each output digit should display for about 1 second followed by a short time (.1sec?) of blank display. After the last digit is displayed (for one second) the display should go blank and the system should go into an idle state and do nothing.

The RESET signal of the NanoCore12 should be implemented so that the program can be re-started.

Design and implement the system with the following specification:

I. Input:

1. One line to RESET (close) the system.

2. One line to implement the START/STOP DISPLAY key.

Use the "push button switch" and 10K "Pull-up" resistors to create these input lines.
Use PORT M pin PM0 for the START/STOP DISPLAY input signal.
Use the NanoCore12 RESET input to implement the system RESET.

II. Output:

1. Use a Seven Segment display to show the decimal digit output digit.

   Use PORT T bits PT6-0 for the signals to a data latch that will then drive the seven segment display (you need 7 lines of input to the data latch and seven segment display.)

III. Operation:

1. At power up or RESET pulse (signal that is pulled LOW and then HIGH) the seven segments should go blank and wait for the START DISPLAY signal.
2. When the START/STOP DISPLAY key is pulsed LOW the UIN digits begin to display.
3. If START/STOP DISPLAY is pulsed LOW again display should hold the current digit and not change.
4. If START/STOP DISPLAY is pulsed LOW again the display should resume displaying the digits from the current digit.

Use "polling" to check for the START/STOP DISPLAY key. Assuming START/STOP DISPLAY is held HIGH by the pull-up resistor (switch closed) then the following code can check for HIGH (switch open)

HERE: BRSET PORTM,X $01 HERE ; Branch if PM0 is SET to HERE
code to run when PM0 = LOW ; If we get here PM0 is LOW

Note that your code will run VERY FAST! It will be done before you open the START/STOP DISPLAY switch again! So, you must check for "key release" before you check for the next PA0 LOW pulse. Use this to check for key release

HERE2: BRCLR PORTM,X $01 HERE2 ; Branch if PM0 is CLEAR to HERE2
code to run when PM0 = HIGH ; If we get here PM0 is HIGH

Now the problem is where do you put the above code slice and how do we restart the display?

Your UIN number is "HARD CODED." The UIN number is permanently in the code. The only way to change it will be to re-assemble the code after you change those digits. Note however that the code is for the bar segments of the seven segment display such that the stored values, when sent to the display, will have the display show the correct decimal digit. See your text.

You MUST use a "look-up" table to store the digits for your UIN. Have the table start at $5200. See examples from class.

The 7-segment displays are Common Cathode type. See this old experiment for the correct connection of the current limiting resistors and display pin-out. You will need seven 470 ohm resistors. You have them in your kits. DO NOT run the seven segment displays without the current limiting resistors. The display should be connected to (i.e. driven by) a 74HC373 8-bit D_TYPE latch. The latch will be driven by NanoCore12 output pins of PortT. Use PM1 for latch enable (LE) control signal. The latch tri-state output control (OE)' is active LOW. So, tie it to ground. We do want output from the latch and not High-Z! Here is the 74HC373 data sheet. The pins are defined on page 1.

EXTRA Credit - 5 Points: Low Power Operation. Can you get this to work with just one segment (LED) on at a time yet changing very fast so that all segment appear to be on at once?

Experiment 4 is due WEEK 6 (10/02)

Report: Your lab report must include the following:
1. A cover page with: Experiment Name and number, ECE 367 Spring 2012, Your name, Your UIN, the date

submitted, and your TA's name.
2. Your assembler code with your name, course, date, experiment number, program explanation and comments for every line of code. The program explanation should include the NanoCore12 pin assignments and an explanation of the organization of the data table, etc.
3. A logic diagram of the circuit
4. An electric circuit diagram of the complete system
5. A user manual to explain how to use the system.
6. Conclusions: Does your circuit meet the specifications and function properly? What problems did you encounter during the coding or construction of the circuit? How would you do this project differently? What extra features or functionality could you have included? What did you learn from working on this project?

Last modified: Sun Sep 23 22:05:08 2012

ECE 367: Experiment 4

Project: What's My UIN with Improved Output and Power Saving

Semester: Fall 2012

Name: Kai Zhao

Signature: _____

UIN: 670720413

Due Date: 2012 October 2

Lab Section: T11

TA: Chenjie Tang

User Manual:
To use this microprocessor, first supply power by USB and press the reset button. After pressing the reset button, press the start/stop button to start displaying my UIN. In order to pause the display of my UIN, press the start/stop button. In order to resume the display of my UIN, press the start/stop button again. After my UIN has finished displaying, press the start/stop button to display my UIN again from the begining.

Conclusion:
Yes, my circuit meets the specifications and function properly.

I had trouble figuring where to branch to after the start/stop button was pressed or released and then figuring how to display the proper display since implementing the extra credit was a challenge. For example, the start/stop button was supposed to stop the display, but the display had to continue if the extra credit was implemented. I had trouble figuring out how to use variables as Booleans and counter since declaring variable is different from other assembly languages. I had trouble between subroutines and branches since subroutines were used to display digits while branches are used when the start/button was pressed, so combining the two was a challenge.

I would do this project differently by using more option variables to reduce the repetitive of my code.

The extra feature I have implemented is the extra credit of power saving mode.

I learned about the 7-segment LED display, 74HC373 latch, variables for counters and Booleans, start/stop button, combining branches with subroutines, and using clock cycles for delays.