Experiment 3 Fall 2012

What's My UIN? - Plus (with power saving)

Go over experiment 1 for a complete explanation of delay loops. You will need to implement delays in experiment 3.

Design a system that will read a 3-bit binary input and based on the input will perform one of eight possible tasks. The tasks are:

1. For input 000 output is blank
2. For input 001 output is the same as experiment 2 - UIN in 4-bit binary and 6-bit ASCII in one second intervals until complete then display goes blank
3. For input 010 output is the same as experiment 2 but with the 6-bit ASCII on the left and 4-bit binary on the right. Repeat 4 times then go blank.
4. For input 011 output is one LED on at a time moving from right to left in half second intervals. Repeat 4 times then go blank.
5. For input 100 output is one LED on at a time moving from left to right in half second intervals. Repeat 4 times then go blank.
6. for input 101 output is LED's go ON from right to left and stay ON until all ten are ON then go blank. Repeat 4 times then go blank.
7. for input 110 output is LED's go ON from left to right and stay ON until all ten are ON then go blank. Repeat 4 times then go blank.
8. for input 111 (input 110 with power saver output) output is LED's appear to go ON from right to left and appear to stay ON until all ten are ON then go blank. Repeat 4 times then go blank. For power save only one LED can be on at any instant in time. HINT: If an LED does an ON then OFF cycle at least 60 times a second it will appear to be ON continuous to the eye. Design a system that will meet the above output requirements for the given input.

   A START DISPLAY signal is required. When the START DISPLAY pulse (HIGH - LOW - HIGH) signal occurs the system should READ the 3-bit input and output according to the table above. Changes made to the 3-bit input are ignored until the START DISPLAY signal occurs. The START DISPLAY signal should be monitored at all times and the system should respond immediately to the signal.

   On power RESET the display should be blank until the START DISPLAY signal occurs.

   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 the system.

   2. One line to implement the START DISPLAY key.

Use the "push button switch" and 10K "Pull-up" resistors that came with your ECE 367 parts kit to create these input lines.
Use PORT M pin PM5 for the START DISPLAY input signal.
Use the NANOCORE12 RESET input to implement the system RESET. There must be a PULL-UP resistor from RESET (pin 30) to Vcc. When pin 30 is connected to ground the NANOCORE12 will re-start running the code from the Start label location.

3. Three input lines for the display choice control (PM2,3,4 are recommended.)

II. Output:

1. Use the LED BAR GRAPH DISPLAY (with a 470 Ohm current limiting resistor for each LED bar - use the 470 Ohm resistors from your ECE 367 parts kit) to show the BINARY and ASCII output.
2. Use PORT T bits , and if necessary, Port M bits PM0-1 for the signals to the LED BAR GRAPH

III. Operation:

1. At power up or RESET pulse (signal that is pulled LOW and then HIGH) the LED's should go blank and wait for the START DISPLAY signal.
2. The user now sets the 3-bit input choice
3. When the START DISPLAY key goes LOW the LED begin to display in the appropriate manner.

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

```
HERE: BRSET PortM, $20, HERE ; Branch if PM5 is SET (1) to HERE - i.e repeat this line
; i.e check PM5 again. Hence polling! Assumes PORTM = $250 and the DDRM = $252
code to run when PM5 = LOW ; If we get here PM5 is LOW
```

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.

You MUST use "look-up" tables to store the digits for your UIN. See examples from class.

Experiment 3 is due WEEK 5 (9/25)

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: Mon Sep 17 20:57:14 2012

ECE 367: Experiment 3

Project: **What's My UIN with Power Saving**

Semester: Fall 2012

Name: Kai Zhao

UIN: 670720413

Due Date: 2012 Sept 25

Lab Section: T11

TA: Chenjie Tang

User Manual:
To use this microprocessor, first supply power by USB and press the reset button. Adjust the settings of the switch for difference output. Inputting binary 0 will result in nothing, inputting binary 1 will show my UIN once in binary then ASCII binary, inputting binary 2 will show my UIN four times in ASCII binary then binary, inputting binary 3 will show one LED on at a time from right to left, inputting binary 4 will show one LED on at a time from right to left, inputting binary 5 will show LEDs on and stay on from right to left, inputting binary 6 will show LEDs on and stay on from left to right, and inputting binary 7 will show LEDs on and appear to stay on from right to left even though only 1 LED is on at a time. Press the start button on the right of the bar graph when ready. After the output, the input options can be adjusted for a different mode by using the switches and then pressing the start button.

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

I had trouble figuring out how to poll for inputs, which was fixed by polling for certain input at certain times. I had trouble using loops to repeat most outputs 4 times due to the lack of registers since A is used for portT, B is used for portM, X is used for traversing through the table, and Y is used for delay. This problem was fixed by using a stack for registers. I had trouble with the delays between 0.5 seconds and 1 second. This problem was due to a copy and paste error. I also had trouble with output of the input of binary 7 since it seems like a loop within a loop within a loop, which was handled by listing out all the tables of input of binary 7.

I would do this project differently by using yet another loop for input of binary 7.

I learned about multiple program inputs, look-up tables, loops, stack for registers, using clock cycles for delays, and subroutines.