Lab Overview

In this lab assignment, you will do the following:

- Plan the layout of your development board and obtain your parts kit.
- Build your basic hardware, consisting of a perf board, standoffs, power connector, voltage regulator, power on LED, power switch, microcontroller, reset circuit, oscillator, Port 0 pull-up resistors, 74LS08 AND gate, 74LS373 latch, and RS-232 connector.
- Learn how to use the ASM51 assembler and Emily52 simulator.

This lab assignment is due by **Saturday, September 18, 2004.**
The deadline for Lab #1 is **Wednesday, September 22, 2004.**

This lab is easier than the following labs and it is weighted as a small part of your course grade. The reason for making this lab easier is to give you a chance to come up to speed on basic hardware and firmware concepts, to make sure that you have all the parts you need to do the assignment, and to make sure that you have access to the lab and the lab equipment is all functional. The following lab assignments will be more involved and more time consuming, so you will have to plan your time wisely.

Lab Details

1. Obtain a parts kit for the lab (available by 9/08/2004).
   - Wire wrap wire may be obtained in the laboratory. Please only take what is required to implement your microcontroller board (please only take 2-3 yards of each color needed at a time).
   - Soldering irons and solder will be available in the laboratory.
   - Please leave the lab equipment in the lab, make sure to turn off the soldering irons, and lock up the room when you leave.
   - You may sign out a tool kit with wire wrap tool, cutters, needle nose mini-pliers, power supply, and RS-232 cable for the semester. Some digital logic probes are also available.

2. Obtain access to the lab. You will need to obtain an electronic key or to get your current CU electronic key programmed for the laboratory door. The keys for this lab will be managed by the TA.

3. If you don't have experience with soldering electronics, read the article "A Guide to Better Soldering", available in the laboratory or visit [http://www.metcal.com](http://www.metcal.com) and read about soldering techniques and tips. Answer the following questions for yourself:
   - Why is flux used? What type of flux should be used with electronic circuits? Will flux remove grease from a connection point?
   - What is a cold solder joint, and how is it created?

4. Skim the article "Handling of Power Plastic Transistors", available in the laboratory. You should always take care not to stress the pins or packaging of any electronics.

5. Your initial hardware assignment will be to implement your core microcontroller design. The basic circuit elements consist of an 80C51 microcontroller with power-on and run-time reset circuitry and an 11.0592MHz crystal oscillator, a 7805 (or 340T5) +5V voltage regulator, and hardware, such as your perf board, standoffs, power connector, 9-pin RS-232 connector, etc.

6. Design your power supply circuit and draw a schematic using Orcad Capture. You should include the 2.1mm power jack, power switch, the 7805 voltage regulator, and a power-on LED which glows whenever power is applied to your board. Some starter schematics are on the course web site.

7. Design your oscillator circuit using the 11.0592 MHz crystal and the 27pF capacitors and add this circuit to your schematic.
8. Design your power-on reset circuit with run-time reset capability. Include an RC circuit and a pushbutton. You may choose to mount the pushbutton on the top or the bottom of your board.

9. You will be using the 6"x8" piece of perf board from your parts kit. Now is the time for you to decide whether you will want to put your development board in an enclosure. If so, you will need to determine the dimensions of the enclosure so that you can trim your perf board down to size. To help in development, you will probably use standoffs to support your board in its four corners. Drill any necessary mounting holes for your standoffs—do not drill within 0.5" of the edge of your board, or the board may crack. In addition, it is suggested that you leave a keep-out area (perhaps 0.5" to 0.8" wide) around the edge of your board, so that you can add connectors or other parts later in the semester. **NOTE:** Parts to be added to your board this semester include a 32KB EPROM; a 32KB SRAM; supporting glue logic, such as your decoding circuits, pull-up resistors and an address/data line demultiplexer; a MAX232 RS-232 line driver/receiver; 8-pin serial EEPROM, and an LCD. You need to leave room for these parts on your board, so keep your regulator and basic microcontroller circuitry confined to one corner or edge of your board. If you do not have a saw and drill, we will try to provide you with access to these tools. You will also want to mount your power jack, power switch, and RS-232 connector at this time. Finish all drilling and sawing before mounting chips.

10. Spend some time planning your layout (see the **Example Board Layout and Tips for Board Construction documents provided**). A lot can be learned from laying out a circuit poorly, but it's usually much easier to correct an error at the layout stage of the game rather than after you have already wrapped 50 wires. You will want to try to keep your wires short to minimize noise problems.

11. You will need to wire up the 2.1mm power jack, a master on/off switch and +5V regulator. If you plan to use a heat sink on your regulator, be sure to leave enough space on the board. You can drill a hole in your board for the power jack. Pay close attention to the polarity of the wall adapter plug. Different plugs have different polarities, and if you're not careful and use the wrong polarity, you can destroy your components on your board. You may also want to include a diode bridge rectifier to protect against damage caused by using a power supply with the opposite polarity. Use heavier wire (e.g. 22 AWG) for your main power connections or double up your wire wrap connections. Remember, when you solder circuitry such as the 7805 regulator, be careful about applying too much heat, or you'll end up damaging or destroying the circuitry. If you don't use a diode bridge, clearly label the polarity of your power connection so that you'll always know what kind of adapter to use.

Example:  

12. In the next lab assignment, you will add your decode logic and EPROM to your base board. If you can picture how you will hook these elements up to your 8051, create a wiring diagram of your basic circuit and label the signal names so that you can visualize the connections you will need to make between chips on your board; otherwise, you can skip to the next step in this assignment. Determine the size of the socket to be used for each chip, and use the socket or the wire wrap guides available on the course web site to plan your layout. Remember, these guides are numbered as if you're looking at the bottom side of your boards, so the pin numbers are reversed from what you'll see when looking at the top side of the board. You will want to place your chips close together to maximize your use of board space and to minimize wire lengths, but far enough apart so that you have room to work, move wires, and see the signal names which will be printed on your boards. A good rule of thumb is to leave 3 empty rows of holes (0.3") between IC sockets in your perf board. Don't spread your chips too far apart or you'll run out of space later in the semester. Don't forget to include space for your decoupling capacitors, and leave plenty of space for more components to be added later in the semester. Several methods can be used for adding decoupling capacitors. One is to use a socket which is one space longer than the chip, and use this space for the decoupling cap (however, you need to be very careful to get the IC pin numbering correct if you use a socket which is larger than the IC). Another way is to use individual T44 wire wrap pins, and solder your decoupling cap to those pins before wire wrapping the pins to your socket. You could also solder the capacitor directly to the chip.
Do not skip the following labeling steps—if you do, you may save a few minutes now, but you'll waste a lot of time later when you are debugging your circuits and have to probe many signals.

13. Cover the bottom of your bare perf board with a piece of white paper. A thin layer of Glue Stic or some transparent tape around the edges will hold your paper down. Neatly print your name and the text “ECEN 5613 F2004” in some unoccupied and visible area on the paper and the top of your circuit board in permanent ink. A fine point marker will work nicely.

14. Make your final decision regarding your circuit layout and gently push your wire wrap socket pins through the paper at the appropriate locations while supporting the back of the paper so that it doesn't rip. Remember that you have capacitors and other support circuitry to locate. Note that the four capacitors for the MAX232 chip take up more space if you choose to use 10uF caps instead of 0.1uF.

15. Remove the sockets and outline each chip's location in ink on the paper. Decide where you are going to put decoupling capacitors and stick holes in the paper at those places. Some of these can be placed in the extra space provided in the wire wrap sockets, but the caps for some chips (e.g. EPROM and 80C51) may need the T44 wire wrap pins.

16. Now neatly number the pins of each chip on the paper (for future labs, you will use the same technique for each new IC that you add to the board). Remember, you are looking at the bottom side of the board, so the pin numbers will be in the opposite order than if you were looking at the top of the board. If you want, use the wire wrap IDs for help in numbering the paper. Your numbers can either be inside the outline of the chip or outside; just be consistent.

17. Now that you've got the pin numbers done, go back and label the signal for each pin (e.g. RST for the reset pin). This will help you greatly when wire wrapping and debugging. Label the positive sides of polarized caps with a ‘+’ sign. Remember, it is much easier to label the paper before you start wiring up the sockets, since at that point you'll have pins and wires in your way.

18. Insert the wire wrap socket(s). If you're pretty sure that you won't have to remove the socket again, you can use a small drop of glue to hold the socket. If you use this approach, be very careful not to get any glue on the socket pins, thus insulating the pins from the wire you will attach later.

19. You must choose wire colors to use when implementing your circuits. For this class, red is always used for power and black is always used for ground. Your other connections must be colors other than red or black. You are free to choose whatever other colors you wish; however, remember that when debugging, it is easier when particular wire colors signify a particular type of signal, and when the chosen colors contrast with each other and the background to a great degree. For example, it is very difficult to debug a board when the background is white and when only white wire wrap wire is used for all connections. One color scheme is shown below:

<table>
<thead>
<tr>
<th>Signal Type</th>
<th>Color</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>Red</td>
<td>Must be red</td>
</tr>
<tr>
<td>Ground</td>
<td>Black</td>
<td>Must be black</td>
</tr>
<tr>
<td>Multiplexed Address/Data</td>
<td>Gray</td>
<td>Your choice of color</td>
</tr>
<tr>
<td>Buffered Data</td>
<td>Yellow</td>
<td>Your choice of color</td>
</tr>
<tr>
<td>Latched Address</td>
<td>Blue</td>
<td>Your choice of color</td>
</tr>
<tr>
<td>Control Signals</td>
<td>Orange</td>
<td>Your choice of color</td>
</tr>
<tr>
<td>Serial Port Signals</td>
<td>Green</td>
<td>Your choice of color</td>
</tr>
</tbody>
</table>

20. After you have become proficient at basic wire wrapping technique, it is time to start wire wrapping your base microcontroller board. The strategy often used is first to lay down the wires which are least likely to move or change, so that any wires you have to adjust later are laying on top of the other wires. First hook up all your ground connections. Remember what you learned about loop size and placement of the decoupling caps—keep the caps close to the IC power and ground pins and keep the leads as short as possible. If you need to solder your decoupling caps into the T44 pins, now is a good time to do this. When wiring, make sure you get good connections so that the sockets are not terribly loose. Leave just a little slack in the wire connections so that you can debug easier later. After you have completed wiring up your ground connections, wire all your +5V connections.
21. Once you have finished wiring up all your ground and +5V connections and before turning on power, you need to test to make sure that you do not have any short circuits. Using a multimeter, test between your +5V and ground connections to verify that there is an open circuit between power and ground. If you catch a mistake at this point, there are fewer wires to search through to find the problem. Use this approach as you are wiring up your board—take small steps, and verify your work often. This will save time in the long run.

22. Now, before putting in any chips, turn on power and verify that you have +5V and ground at the correct pins on your sockets. Measure the voltage with a digital multimeter.

23. Wire up your reset circuitry and your oscillator circuitry, and tie the 8051 EA line low. Keep the oscillator circuitry close to the microcontroller and use very short wires. Pull-up your Port 0 pins to VCC so that the processor doesn't go into power-down mode. You can use a resistor network in a SIP package or discrete resistors. You may not need to pull up all 8 data lines.

24. If you have verified all the basic connections and circuitry, you can now insert your microcontroller (with the power off). Then, turn on power, and use an oscilloscope verify that your ALE line is oscillating at the correct frequency (1/6 the XTAL2 frequency). Use an oscilloscope to view the XTAL2 waveform and check its frequency. If you want, use a logic analyzer to verify that a fetch from address 0000h is occurring immediately after the processor comes out of reset. You should make sure that your oscillator starts reliably every time power is turned on and after a reset. You may need to adjust the oscillator load capacitance. Be aware that the oscilloscope probe capacitance can affect your oscillator startup when you probe the XTAL1 and XTAL2 pins.

25. Using the oscilloscope, verify that the peak to peak noise between VCC and GND on your C501 is less than ~800mV. If necessary, add more 0.1uF/1.0uF bypass capacitors across the C501 power pins.

26. Wire up the '373 to the microcontroller. Also, wire up the '08 to the microcontroller to generate READ from RD and PSEN. Think carefully about the steps you will need to take to complete your base board, and formulate a plan for incrementally wiring and testing parts of your circuit.

27. For the next part, you will need to use the ASM51 or AS31 assemblers and the Emily52 simulator. You may use the tools installed on the computers in the lab, or you can obtain a copy of the Emily52 demo kit from http://www.dunfield.com/software.htm. This demo kit includes a version of both the ASM51 assembler and the Emily52 simulator, as well as three simple example files. Refer to the assembler and simulator homework assignments given previously for more information.

28. Write an assembly language program which meets the following criteria:
   - The program starts at address 0000h.
   - If at any time the accumulator reaches a value of greater than 50h (including the starting value), then the program jumps to the END label and no further values are written to external memory.
   - The program loops a maximum of 10 times. The first time through the loop, the value in the accumulator is written to external data memory starting at location 0100h. In each successive loop, the accumulator is incremented and the new value is written to the next successive location in external memory. After the last loop is executed, the program jumps to a label named 'END' and writes the number of times the loop executed into internal memory (IRAM) at address 20h. The program should then terminate, by using an illegal instruction which will stop Emily52.

Refer to the test1.asm and test3.asm files distributed with the demo version of Emily52 for hints. Test your program with Emily52 by setting the accumulator to various values. Note: Use a combined code and data space in the 8051 (/overlap option). Practice using the various capabilities of Emily52.

29. Q: How many bytes of code space does your program require? (don't count the illegal instruction)
   Q: How long did your program take to execute, assuming it completed 10 writes to external memory? Assume an 11.0592 MHz clock. Don't include the illegal instruction. Show your detailed calculations on the code listing that you submit with the signoff sheet.
   Q: What would happen if your loop started writing to external memory at location 0008h when running in the simulator? What would happen if the code was running in actual hardware?

30. For the demo, be able to change the starting value in the accumulator using Emily52.
You will need to obtain the signature of your TA or instructor on the following items in order to receive credit for your Lab #1 assignment. Your assignment is due on **Saturday, September 18, 2004**. Labs completed after the due date or submitted after the deadline date will receive grade reductions.

The software portion of Lab #1 should be completed by **September 8th** in order to give you time to complete the hardware portion upon receipt of your parts kit. Note that Lab #2 is more time consuming.

Print your name below, sign the honor code pledge, and then demonstrate your working hardware/firmware in order to obtain the necessary signatures. All items must be completed to get a signature. Separate this sheet from the rest of the lab and turn in this signed form, a copy of your schematic, a printout of your full listing file (printed legibly and complete with comments), and the answers to any applicable lab questions to the instructor in order to receive credit for your work. Make sure your name is on each item and staple the items together, **with this signoff sheet as the top item**.

**Student Name:** ______________________________________

**Honor Code Pledge:** "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work."

**Student Signature:** _________________________________

**Checklist**

- Schematic of acceptable quality:
- Student name on board/paper backing in permanent ink:
- Pins and signals labeled on paper backing:
- Mounting hardware present (e.g. standoffs or an enclosure):
- Power switch and LED:
- Voltage regulator functional, power jack present and labeled:
- Power-on Reset (RC) and Run-time Reset (pushbutton):
- RS-232 connector mounted:
- 74LS373 transparent latch wired:
- 74LS08 AND gate generation of \( R_{\text{EAD}} \):
- Student displays good knowledge of oscilloscope:
- C501 bypass cap is present. Peak to peak noise measured across C501 VCC and GND is < 800mV.
- Oscillator functional (check for correct ALE/XTAL2 signals after power on-off cycles):

**Student Answers to Lab Questions (see lab item 29)**

Code Size? ________________  Execution Time? ________________

Consequences of writing to 0008h when code is running in simulator? When code running in hardware?

- Student demonstrates detailed knowledge of Emily52 (including editing data memory, breakpoints, single stepping, uses /overlap option, etc.):
- Student assembly program works correctly:

**Instructor/TA Comments:**

**Instructor/TA signature and date**