
Lab Overview

In this lab assignment, you will do the following:

- Learn how to use the ASM51 (or AS31) assembler and Emily52 simulator.
- Learn how to use the WinCUPL/WinSim design suite for the Atmel AT16V8C SPLD.
- Plan the layout of your development board and obtain your parts kit.
- Build your basic hardware, consisting of a board, standoffs, power connector, voltage regulator, power on LED, power switch, wire wrap sockets, microcontroller, reset circuit, oscillator, Port 0 pull-up resistors, SPLD, 74LS373 latch, and RS-232 connector.
- Start learning schematic capture and gain skills in soldering.

The software portion of this lab assignment should be done by **Wednesday, September 9, 2009**.

The signature due date for the entire lab assignment is **Tuesday, September 15, 2009**.

The submission due date (when the report must be turned in) is **Wednesday, September 16, 2009**.

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. If you do not already have one, obtain a Buff OneCard, needed for access to the engineering center and the 2B37 laboratory. CAETE students not seeking a degree can obtain a card for \$25. You will need to get this card from the Buff OneCard office on campus. For more information, see <http://buffonecard.com>. **When you get your card, please provide the card number to the instructor via e-mail so that you can get access to the lab areas.**
2. For the software part of this lab, you will need to use the ASM51 or AS31 assemblers and the Emily52 simulator. You may use the licensed tools installed on the computers in the lab, or you can obtain a copy of the Emily52 demo kit from <http://www.dunfield.com/products/emily52.htm>. This demo kit comes as a self-extracting .exe file, and includes a version of both the ASM51 assembler and the Emily52 simulator, as well as three example files. Refer to the assembler and simulator homework assignments for more information.
3. Write a single 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 (/overlay option). Practice using the various capabilities of Emily52.

4. For the demo, be able to change the value in the accumulator using Emily52 (during program execution). Make sure your program can correctly handle different starting values in the accumulator.

▪ **Submit a listing (.LST file) of your final commented firmware.**

5. For the SPLD design suite part, you will need the Atmel WinCUPL tools, which are available on the Atmel web site and through a link on our course web site. These tools are also installed on the computers in our lab.
6. Learn how to use Atmel WinCUPL and WinSim. Review the tutorials and examples.
7. Using WinCUPL, develop code for the Atmel AT16V8C SPLD. Assign A15, A14, A13, A12, /RD, and /PSEN from the processor to six of the SPLD inputs. Generate two outputs: /READ = /RD & /PSEN, and /CSPERIPH = !(A15 & A14 & A13 & A12).
8. Verify your logic with WinSim, using at least six test vectors to show correct logic functionality for a well-chosen subset of all possible input conditions. The /READ output of the SPLD should be toggling when either /RD or /PSEN toggles. The /CSPERIPH OUTPUT should be high most of the time, and should be low only for addresses in the range of F000h-FFFFh.

- As part of the signoff procedure, show the TA your commented .PLD source file and .SI simulator input file. Explain how these files are structured.

9. Obtain a parts kit for the lab (available by **Saturday 9/05/2009**).
 - 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 make sure the door to the lab is closed and locked when you leave.
 - You may sign out a tool kit with wire wrap tool, cutters, needle nose mini-pliers, power supply, digital logic probe, 28-pin ZIF socket, NVRAM, floppy diskette, and RS-232 cable for the semester. Take good care of these items, as you are financially responsible for them.
10. If you don't have experience with soldering electronics, read the article "A Guide to Better Soldering", available in the laboratory or visit a web site (e.g. <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?
11. [Optional] 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.

12. You will be using the 6"x8" printed circuit board (PCB) from your parts kit. Study the schematics included in the document "**ECEN 5613 PCB Layout and Partial Schematics**", which is available on the course web site. The schematics show you some of the circuit connections that are present on your PCB. Note that not all of the components shown in the partial schematics will end up being populated on your board; the components are in the schematics so that pads (through hole or surface mount) would be present in the PCB design. You will need to determine which components you will need to load during the semester.
13. 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 PCB, standoffs, power connector, 9-pin RS-232 connector, etc.

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 (in OrCad .DSN and Adobe .PDF formats).

Design your oscillator circuit using the 11.0592 MHz crystal and the 27pF capacitors and add this circuit to your schematic.

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.


- **Include a printout of your schematic as part of your lab submission.**

14. **Before soldering in any components and before using your bare PCB, you must test to make sure that you do not have any short circuits. Using a multimeter, test between your +5V and ground connection on the PCB to verify that there is an open circuit between power and ground.** If you catch a PCB flaw at this point, you won't waste time and money debugging your board after you've added your circuits.

Study your PCB carefully and compare it to the partial schematics. Use a multimeter to learn how the power and ground planes on your board are designed. Are they connected directly to the VCC and GND pins on each chip? Are they connected directly to the VCC and GND headers distributed around the board? Which of the capacitors in the partial schematics are through hole and which are surface mount technology (SMT)?

15. Spend some time planning your layout (**see the "Example Board Layout" document 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 soldered in chips and wrapped 50 wires. You will want to try to keep your wires short to minimize noise problems. To help in development, you will probably use standoffs to support your board in its four corners. It is recommended not to drill any additional holes in your PCB, as you take the chance of damaging existing traces or shorting together the VCC and GND planes. 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; (optionally) a 32KB NVRAM/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.
16. Using a marker or pen, neatly print **your name** and the text "**F2009**" in the big white space on the bottom side of the PCB.
17. **If you are new to soldering, you may want to complete the SMT soldering lab and also do some practice soldering on a scrap PCB before doing any soldering on your board for this course. Ask the instructor for details during lecture.**


18. Insert the wire wrap socket(s). Carefully solder the wire wrap sockets to the PCB (be careful not to use too much solder, as you don't want to solder together your power and ground planes by mistake). If you examine the PCB carefully, you'll notice that not all the pins for each chip have electrical connections on the board - the VCC pin, GND pin, and some signal pins on the chips have connections to traces on the PCB. **At a minimum, the VCC and GND pins of each socket must be soldered to the board, as well as the pins which have traces on the board. Make sure you've completed all the soldering before you start wire wrapping to a socket!** It's easy to melt the insulation on the wire wrap wires if you have to come back and solder to the socket after you've already finished your wire wrapping. **Make sure the wire wrap sockets are tight against the board before you start soldering.**
19. You will need to wire up the 2.1mm power jack, a master on/off power switch and +5V regulator. If you plan to use a heat sink on your regulator, be sure to leave enough space on the board. 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 will use the diode bridge rectifier on your PCB 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 the diode bridge, clearly label the polarity of your power connection so that you'll always know what kind of adapter to use.

Example: + —  — -

20. You will also want to mount your RS-232 connector at this time. Solder in the 9 pins of the RS-232 connector (reference designator J4). You can also solder in an 8-pin or 9-pin header so that you'll be able to wire wrap to the RS-232 pins later in the semester. If using an 8-pin header, make sure to solder it in to positions 1-8 on the header outline for reference designator J5.
21. The individual T44 wire wrap pins can be used to hold discrete components as necessary.
22. 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:

Signal Type	Color	Comment
Power	Red	Must be red
Ground	Black	Must be black
Multiplexed Address/Data	Gray	Your choice of color
Buffered Data	Yellow	Your choice of color
Latched Address	Blue	Your choice of color
Control Signals	Orange	Your choice of color
Serial Port Signals	Green	Your choice of color

23. 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 capacitors—**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 in the semester. If the wires are too tight, then it's difficult to move them when you need to see

silkscreen labels or other wires underneath. After you have completed wiring up your ground connections, wire all your +5V connections. Remember to strip at least 3/4" of bare wire in order to wire wrap. 0.75 inches is about this long: 

24. 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.
25. 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. Using the oscilloscope, verify that the VCC voltage is close to 5.0V and looks stable at 5.0V, without big oscillations.
26. Wire up your reset circuitry and your oscillator circuitry, and tie the 8051 \overline{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.
27. 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 need to 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.
28. Using the oscilloscope, verify that the VCC voltage is close to 5.0V and that the peak to peak noise between VCC and GND on your C501 is less than ~800mV. If necessary, add more bypass capacitors (1.0uF/4.7uF/etc.) across the C501 power pins.
29. Wire up the '373 latch to the microcontroller.
30. Wire up your Atmel AT16V8C SPLD socket. Attach A15, A14, A13, A12, /RD, and /PSEN from the processor to six of the SPLD inputs (matching the assignments in your WinCUPL code).
31. Program the SPLD to generate two outputs:
/READ = /RD & /PSEN, and /CSPERIPH = !(A15 & A14 & A13 & A12).
32. Verify the SPLD outputs. Examine the /READ output of the SPLD to verify that it is toggling when /PSEN toggles. The /CSPERIPH OUTPUT should be high most of the time, and should have a 15/16 duty cycle (it will be low only for addresses in the range of F000h-FFFFh, while the processor continuously cycles through addresses 0000h to FFFFh).
33. All ICs you add to your board must be labeled with identification of the IC name, pin numbers, and all signal names, similar to the information present on the PCB silkscreen. Wire wrap label templates are available on the course web site (search for "Wire Wrapping ID"), and may be helpful to you.
34. 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.

You will need to obtain the signature of your TA on the following items in order to receive credit.

The software portion of Lab #1 should be completed and signed off by **Wednesday, September 9, 2009** in order to give you time to complete the hardware portion upon receipt of your parts kit. Both signoffs are due by **Tuesday, September 15, 2009**. You need to submit both of your signoff sheets and lab report by **Wednesday, September 16, 2009**. Labs completed after the signature due date or submitted after the submission due date will receive grade reductions.

Print your name below, circle your course number, 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 the signoff sheets from the rest of the lab and turn in the signed forms, a copy of your schematic, a printout of the full listing file (.LST 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.

- Signed and dated software and hardware signoff sheets (No cover sheet please)
- Full copy of complete and accurate schematic of acceptable quality (all components shown).
- Printout of fully, neatly, clearly commented code in .LST file. Ensure your printout is easy to read.

Make sure your name is on each item and staple the items together, with this signoff sheet as the top item.

Student Name: _____ **4613** or **5613** (circle one)

Honor Code Pledge: "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work. **I have clearly acknowledged work that is not my own."**

Student Signature: _____

Checklist

- Student demonstrates detailed knowledge of Emily52 (including changing register values, editing data memory, using breakpoints, single stepping, uses /overlay option, etc.)
- Student assembly program works correctly
- Student demonstrates detailed knowledge of WinCUPL and WinSim, logic equations correct

Student Answers to Lab Questions

1. **How many bytes of code space does your program require?**
(Don't count the illegal instruction. Show how you arrived at your answer.)
Code Size? _____
2. **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 (\$A5). Show your detailed calculations on the code listing that you submit with the signoff sheet.
Execution Time? _____
3. **What would happen if your loop started writing to external memory at location 0008h when running in the simulator?**
4. **What would happen if your loop started writing to external memory at location 0008h if the code was running in actual hardware?**

Instructor/TA Comments:

Instructor/TA signature and date

Demo Below Average / Average / Above Average / Superior

Print your name below, sign the honor code pledge, and then demonstrate your working hardware in order to obtain the necessary signatures. All items must be completed to get a signature.

Student Name: _____

Checklist

- Schematic of acceptable quality
- Student name on board in permanent ink
- Pins and signals labeled, decoupling capacitors, and two 28-pin wire wrap sockets present on board:
- Mounting hardware present (e.g. standoffs or an enclosure)
- Power switch and LED
- Voltage regulator functional, power jack present
- Power-on Reset (RC) and Run-time Reset (pushbutton)
- RS-232 connector mounted
- 74LS373 transparent latch wired
- Logic outputs correct (e.g. SPLD generation of /READ and /CSPERIPH; view SPLD code)
- Student displays good knowledge of oscilloscope
- C501 bypass cap (4.7uF) is present
- Peak to peak noise measured across processor VCC and GND is < 800mV
- Oscillator functional (check for correct ALE/XTAL2 signals after power on-off cycles)

Student Answers to Lab Questions

1. **What voltage is present at the regulator input?** Use a digital multimeter. _____
2. **What voltage is present at the regulator output?** Use a digital multimeter. _____
3. **How much power is dissipated in the regulator, assuming a load current of 100mA?** Assume that the regulator is drawing the max quiescent current shown in the data sheet. Show all your work.

Calculated value: _____

4. **What peak to peak noise is present across the processor VCC and GND?** Use an oscilloscope.

Measured value at processor package pins on top side of board: _____

Measured value at wire wrap socket pins on bottom side of board: _____

5. **How long is the processor held in reset after the run-time reset pushbutton is released?** Use an oscilloscope and try to measure the time between the release of the pushbutton and the time when noise from ALE is observed on the RST signal.

Measured value: _____

6. **What frequency is present at the ALE pin?** Use an oscilloscope. _____

Instructor/TA Comments:

Instructor/TA signature and date

Demo Below Average / Average / Above Average / Superior