CS/EE 3720 Handout #2
Spring 2006
Myers

LAB #2: A Command Prompt Shell for the 68HC11

Lab writeup 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

- Learn how to use a couple simulators for the 68hc11.
- Gain experience with interrupt handlers and serial interface drivers.
- Complete a 68hc11 command prompt shell that will be useful for future labs.

2 Tasks

Note: In order to use lab time efficiently, try to complete as much of Task 4 as possible before the lab.

1. Simulate the count_a.asm program from Lab 1 using the Wookie simulator.
2. Simulate the count_a.asm program from Lab 1 using the TExaS simulator.
3. Assemble, load, and run the lab2.asm program. Use the oscilloscope to observe characters as they are echoed.
4. Extend the SCI receive interrupt handler in the program lab2.asm to buffer characters. Use carriage-returns and a max buffer size of 16 to terminate commands. Interpret "U", "D", and "Q" commands to make the LED counter count up, count down, and turn off, respectively.

3 Resources (available in lab)

- Lab2.asm source code (download source from course website)
- The Wookie 68HC11 simulator (download executable from course website)
- Hyperterminal (available at http://www.hilgraeve.com/htpe/)
- RS232/TTL converter cable (from www.kevinro.com). Available for checkout.
- For PCs with D25 serial connectors, a D25-D9 adapter is required.
- Bench cable set. Available for checkout.
- AS11.exe assembler (download executable from course website)
- DL11.exe loader (download executable from course website)
- load_e1.bat batch file (download batch file from course website)
- Test program to count on port A (download source from course website)

Students should also bring a diskette for saving work completed during the lab.

4 Parts (available for purchase in lab)

Should have your 68HC11 microcontroller circuit from Lab 1 working, since this will be used extensively in Lab 2.

5 Procedures

See Lab 1 for procedures on using AS11 and DL11.

1. Set up the Wookie simulator to run the Count_a.asm program.
   (a) Make sure the Wookie simulator is available on your lab PC. (If not, create a D:\Wookie directory and place the executable there.)
   (b) Using an MS-DOS command prompt, execute the command:
   "as11 count_a.asm -l > count_a.lst"
   to create an .s19 file and a .lst file.
   (c) Launch the Wookie simulator.
   (d) On the File menu, select "Load .s19 File...". Find and open your "count_a.s19" file. In the "Set HC11 Mode" dialog box that pops up, choose "Rug Warrior Bootstrap". Don't worry about the start address.
   (e) Click on the "View Code" button.
   (f) On the "Code View" window, click on "Load LST File".
   (g) Find and select the "count_a.lst" file that you created using AS11.
   (h) In the "Choose LST File Format" dialog box, select "AS11M". Leave the offset as it is.
   (i) Move and resize the "Code View" window as necessary.
   (j) Click on the "M68HC11 CPU" button. Move the "MCU" window to a convenient place.
   (k) In the "MCU" window, click on PC, and push Modify. Set the PC (program counter) to the address of the first instruction in your program.
   (l) Press the single-step button (with the picture of the walking man) to step through the code. The highlighted line in the "Code View" window should change, and the registers in the "MCU" window should update appropriately.
   (m) Experiment with the simulator features. Notice which instructions affect the N, Z, V, and C flags in the CCR (condition code register). Display Port A, and simulate the program at full speed by clicking on the red stoplight button. When single-stepping, how do you get past the long busy wait?
2. Use the TExaS simulator to run the Count_a.asm program.
   (a) Insert the CD that comes with the textbook, and watch the movie named README in the root directory of the CD. This movie is a tutorial of how to use TExaS.
   (b) If you are working on your own computer, then you should next install and upgrade TExaS. To install run setup.exe in the TExaS subdirectory, then follow the link on the course website and download the upgrade zip file.
   (c) Start up the TExaS simulator which should be located under: start → programs → Test Execute and Simulate → microcomputer simulator.
   (d) Select File → Open, select all files from the list of types, and browse to find where you saved count_a.asm, and open it.
   (e) Find the line:
   ```
       ldaa   #$80
   ```
   And change it to:
   ```
    Go   ldx   #$80
   ```
   This creates a label for the start of your program.

(f) Find the line:

```
ldx  #$F000
```

And change it to:

```
ldx  #$000F
```

This will reduce the delay to something shorter otherwise the simulation takes a very long time.

(g) Finally go to the bottom of your file, and add these two lines:

```
org  $FFFE
fdb  Go
```

This makes your program start at the label Go on reset.

(h) Select File → New, then select microcomputer.

(i) In the new window, click on registers to see the register values.

(j) Select File → New, then select I/O device.

(k) While in this new window, select IO → LED.

(l) In the new window, enter PA4 through PA7 in the port fields and press OK to create four LEDs.

(m) Go back to your assembly code window, and select Assemble → Assemble.

(n) Check the window labeled TheList to make sure that there were no errors.

(o) Select Action → Go.

(p) Use the Window menu to go look at each window to see what happens during the simulation.

(q) When you are done, select Action → Halt.

(r) Continue to play around with the simulator till you are comfortable with it.

3. Run and test the lab2.asm program on your 68hc11 circuit.

   (a) First, launch the Hyperterminal program. Use "Properties" on the File menu to configure this program for:
   COM1, 9600 bits per second, 8 data bits, no parity, 1 stop bit, no flow control. Under "ASCII Setup" deselect everything except "Wrap lines that exceed terminal width".
   Make sure Hyperterminal is not connected before proceeding with the next step. (Use "Disconnect" on the "Call" menu, if necessary.)

   (b) Using AS11 and DL11, assemble and load the lab2.asm program on your 68hc11 circuit. After the program is loaded, you'll need to disconnect the serial cable in order to boot your program. Once the program is running (your LEDs on Port A should count), reconnect the serial cable. This process can be simplified by using a DIP switch to control whether the RX signal of the 68hc11 is connected. The TX signal can be permanently connected to the serial cable, since it won't affect the boot behavior.

   (c) Using Hyperterminal, type some characters. This should force the program to open COM1, transmit the characters, and display any received characters. If you've set up Hyperterminal program as recommended, your typed characters will not be echoed by Hyperterminal directly; they will only be displayed if they are echoed by the device at the the other end of the serial cable. In fact, the lab2.asm program does echo characters, so you should see characters displayed as you type.

   (d) Using both channels of the oscilloscope, probe the RxD and TxD signals on the 68hc11. Observe the delay between the characters received by the 68hc11 and the echoed characters. Type the numbers 0 through 9, and notice how the pattern on the oscilloscope changes.

4. Extend the SCI receive interrupt handler in the program lab2.asm to buffer characters and interpret commands.

(a) Study the lab2.asm code carefully to make sure you understand how the transmit buffer command works. Note that it is interrupt-driven, and contains no busy waits.

(b) Copy the lab2.asm program to a new "shell.asm" file. Add code where you see the word "Pending" (and modify code if necessary) to implement the following requirements:

  i. The shell will accept commands of up to 16 characters.
  ii. Commands are terminated by a carriage-return.
  iii. A 17th transmitted character will terminate the command as if it were a carriage-return.
  iv. The "Q" command will cause the LEDs to turn off until a "U" or "D" command is received.
  v. The "U" command will cause the LEDs to count up.
  vi. The "D" command will cause the LEDs to count down.
  vii. A command starting with "Q", "U", or "D" may be treated as a "Q", "U", or "D" command, respectively.
  viii. Any command not recognized will cause the string "Commands: U, D, Q" to be displayed.
  ix. After a command is received and responded to, the prompt "HC11> " will be displayed.

(c) Demonstrate your working shell program to your TA.

(d) Copy your programs to your own diskette, and delete from the lab PC.

6 Prelab

1. You should complete simulation steps 1 and 2 above before coming to your lab section and be prepared to show this simulation to your TA.

2. You should write the code for step 4 before coming to lab and be prepared to show this to your TA at the beginning of your lab section. This code does not need to be 100 percent debugged, but getting it close to working will allow your lab section to go more smoothly.

7 Writeup

1. Include a printout of your final command prompt program.

2. Describe any problems you encountered.

3. Answer the following questions:

   (a) Based on your oscilloscope measurements, what is a typical value for the delay between the time the 68hc11 receives a character and the time it retransmits it? Describe what happens during this time.

   (b) How might the lab2.asm program fail if the instruction "lds #STACK" were omitted?

   (c) In the initsci initialization code in lab2.asm, what would happen if the instruction "ldaa #$2c" was replaced with "ldaa #$ac"?

   (d) Explain why the instruction "brclr SCSR,Y #SCSR_TDRE txchr_" at the beginning of the txchr interrupt handler is necessary. Will this interrupt handler even be called if the TDR is still full? Explain.