CSCI 4415 Lab # 5

Serial Memory

Created: January 30, 2011

Objective:

- Learn how to read and write from serial (I2C and SPI) memory chips..

Time Estimate:

- 10 hours.
- You should be prepared to start this after class 7 (Serial Interfaces) and this will be due by Midnight class 11.

Specific Requirements:

For this lab you will take 2 memory chips and connect them to the Z16. You will need to find the proper datasheet for each memory so that you can read the details on how to connect it and how to writ software to read and write from the memory. You will then add commands to your command line interpreter for accessing these memories.

1. Connect your Z16 contest kit console port to your computer's serial port (you may need a USB-serial dongle for this). A serial cable is included in the lab kit.
2. In the lab kit you will have 2 8-pin DIP memory chips.
    - The **24LC16B** is a 16 Kbit Electrically Erasable Programmable Read-Only Memory device. The device memory is accessed with a 2-wire (I2C) serial interface
    - The **25LC040A** is a 4 Kbit Electrically Erasable Programmable Read-Only Memory. The memory is accessed though a simple Serial Peripheral Interface (SPI) compatible serial bus.
3. You will need to read the marking on the chips, determine which is which, and then find the detailed datasheet for each. The data sheet will identify the pins and explain how to connect the memory to a microcontrioller and how a software applications needs to interact with it.
4. Insert the memory chips into the prototype board and connect them to the appropriate pins on the Z16.
5. Write a library for each I2C and SPI for reading and writing to/from these memory chips.
6. You will need to consult the ZNEO product specification (PS220) chapter on the "Enhanced Serial Peripheral Interface".
7. Modify your CLI from the previous lab to include the following new commands (and appropriate actions).

```
i2c [fill|dump [start address, [length]]
```

```
        - fill memory with incremental data (0x00, 0x01 …) starting
          at address, continue for length bytes.
        - read from memory starting at address, continue for length
          bytes. display output line in common hexdump
          (http://en.wikipedia.org/wiki/Hex_dump) format.

    spi [fill|dump [start address, [length]]
        - fill memory with incremental data (0x00, 0x01 …) starting
          at address, continue for length bytes.
        - read from memory starting at address, continue
          for length bytes. display output line in common hexdump
          (http://en.wikipedia.org/wiki/Hex_dump) format.
```

8. Create a file named readme.txt file that includes any notes about the lab and answers to the questions below.
9. In the main.c file include the following information in a comment block

```
Author: [Your name]
Email:  [Your email]
Class:  [CSCI-4415]
Date:
Lab:    [Lab number]
Description: [short description of program. Include changes you made and
note any enhancements that you made to the lab]
Other files: [list other files that are necessary for this program]
Compile: [provide any special instructions for compiling. Only necessary if
there are special instructions]
Problems: [explaining what you did, what problems you had, how you solved them,
and what you might do differently if you had to do it again]
Comments: [Feel free to provide comments on how this lab went, what you think
is good or bad about it and how it could be improved or anything else you want
to say.]
Enhancements: [describe the enhancements you added to this lab]
```

10. Turn in a zip file (lastname-lab5.zip) of the entire project directory (include the entire ZDSII project directory and the readme.txt). Use BlackBoard to submit assignments (or if thats not working for some reason email the ZIP file).

References

1. ZNEO product specification (PS220) chapter on the "Enhanced Serial Peripheral Interface".

Grading

- On time, in proper format (Zip file), named <lastname>-lab1.zip
- Compiles and runs without problems
- Meets lab requirements
- Reasonable code (partitioning of function, coding standard, readable
- Reasonable comments
- I2C function
- SPI function

- Modular code (no side effects)
- Error detection and handling
- Proper use of Z16
- Answered questions

Enhancements:

- Better (fault tolerant) SPI, I2C library
- Add command to CLI

```
i2c|spi [erase [ALL | address,length]]
      - write 0x00 to memory

spi [lock|unlock [block]]

i2c|spi write [address] [“text”]
      - write “text” to memory starting at address

i2c|spi [search “text”]
     - starting at address 0x00 read from memory looking for all
       occurrences of “text”, print out starting address of each
       occurrence.

i2c|spi [strings]
     - search memory starting at 0x00 and display occurrences and
       starting address of 3 or more printable (ASCII 32-126)
       characters.
```

- Modify the dump command, add the ASCII column (as show here http://en.wikipedia.org/wiki/Hex_dump)
- Something you thought of

Questions

Include with your lab submission (in the comments of your main.c file):

1. What is the full part number for each device?
2. Give me a simple schematic (ASCII art is good) showing how you connected each memory to the Z16. Label all pins with signal name an pin numbers.
3. Did you have any problems finding the datasheets or getting the necessary information?
4. What techniques did you to use help you debug your programs?
5. Did you need any pull up resistors? Why?