Lab 11a 6811 Construction and Testing

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, Second edition, by Jonathan W. Valvano, published by Thomson, copyright © 2006.

Goals
- Build and test the 6811 system,
- Obverse the read/write cycles of a computer/memory interface.

Review
- Valvano Chapter 9 on interfacing to your microcontroller,
- Data sheets for your microcontroller,
- Data sheets for your memory.

Starter files
- none

Background

This experiment includes the sequence of steps required to test an embedded system. Verification will involve many interrelated steps. The first step is to verify power and ground signals are properly applied to all the chips. The second step is to test the ability to download and run a program. It is important that this program be so simple that if the system doesn't work, then there is a hardware fault and not a software bug. One method to diagnose hardware failures is to do a side-by-side comparison with an operational system. Next, you run a program to test the microcomputer-memory interface.

Some students plan to put extra electronics off the PCB, because it doesn't all fit on the PCB. If you do have off-board electronics, then you will need a connector or something to create the bridge. You can get a good grade in Labs 8,11,12 with off board electronics, but you will not be eligible to win "best design". In particular, your grade depends on if the required tasks are on time and if the 6811/memory interface works, and if the eventual project (I/O software computer) works. However to win "best design" you will need meet the following restrictions

1) runs using the 6811 and external EEPROM/RAM memory on the PCB
2) all electronics (resistors, capacitors, ICs) are on the Miniboard PCB
3) your team of 2 spends less than $20 on extra components (which are readily available to all students).

It is possible to have external I/O devices, like speakers switches thermistor LCD, off the PCB and still win "best design".

Preparation (do this before your lab period)

1: Recheck the PCB layout on the actual board looking for dropout (missing traces) and inadvertent shorts. Follow the +5V and ground signals around the board verifying with an ohmmeter than there are no shorts to inappropriate places. Assemble the components required to complete the interface, and solder them to the PC board.

2: Verify that Read Data Available overlaps Read Data Required and that Write Data Available overlaps Write Data Required. Draw the combined READ timing diagram and draw the combined WRITE timing diagram, as described in Chapter 9 of the book.

3: Before plugging the microcomputer and memory chips into their sockets, check the power supply circuit. Look on the oscilloscope to verify the absence of noise in the power signal. Using a DVM make sure the voltage level is within tolerance of the microcomputer and memory.

4: Next, write a very simple program that interacts with the switch input and LED output on your system.

Procedure (do this during your lab period)

1: First test your system with a very simple main program (preparation step 4). This will verify you can download and run software on your system.

2: Experimentally measure the actual memory interface timing diagrams. For example if location **0xC000** is external EEPROM on your computer. Run programs like this (add code to initialize the interface if needed)

```
#define memory *(unsigned char *)(0xC000)
void WriteTest(void){
  while(1){
```

```
    memory = 0xC3;     // write data to external RAM
    Timer_msWait(20);  // wait 20ms (should be done in 10ms)
  }
}
void ReadTest(void){  unsigned char data;    // internal RAM
  while(1){
    data = memory;     // read from external RAM
  }
}
```

While running these simple programs over and over, the memory interface signals can be observed on the oscilloscope using the dual channel capability of the scope. A pulse should be received from the /**CE /OE** or **/WE** line during accesses to the memory. Observe one signal such as /**CE /OE** or **/WE** as appropriate with the following signals in a pair-wise fashion. E.g.,

- /**CE** and **E**
- /**CE** and **A15**
- /**CE** and **R/W**
- /**CE** and **/OE**
- /**CE** and **/WE**

Trigger on the falling edge of /**CE**. Using the dual channel capability, verify the proper relation between **E**, **A15**, **R/W /WE /OE** and /**CE**. In this way you should be able to verify the timing diagram created as part of the lab preparation. Draw the read and write timing diagrams as actually measured. Include both data available and data required. Which one did you measure and which one did you derive? Verify that all the microcomputer and EEPROM timing restrictions are met. Perform this measurement for both the read and write cycles.

Deliverables (exact components of the lab report)

A) Objectives (1/2 page maximum)
B) Hardware Design
 Detailed circuit diagram of the system (from Lab 8)
C) Software Design (no software printout in the report)
 none
D) Measurement Data
 Draw the theoretical read and write timing diagrams (preparation 2)
 Draw the measured read and write timing diagrams (procedure 2)
E) Analysis and Discussion (1 page maximum)

Checkout (show this to the TA)

You should demonstrate your very simple main program and the ability to download and run software on the 6811 to the TA. At this point you and TA should agree about exactly what your final system is supposed to do.

Your software files will be copied by the TA, and graded for style at a later time.

Hints:

1. If you need help soldering, please ask the TA or one of the 2nd floor lab staff. There will be a +5 point bonus if you show your soldering job to the 2nd floor lab staff, and they think your soldering work is high quality.

2. The read access time (delay from address available to the start of read data available) can be determined by the number after the dash on the chip. For example, –15 means 150 ns, –20 means 200 ns, and –25 means 250 ns. The read access time is also the delay from the fall of **/CE** to the start of read data available.

3. Common mistakes that could cause the board not to work
 Cold solder joints look dull and round (proper solder joints are shiny and shaped like a cone)
 Wrong component, many components look the same
  the MC34164 and the 2N2222
  a 10mA LED and a 1mA LED
 Wrong resistor values (use 2.7k with 1mA LED and 220 with 10mA LED)

Backwards components
The LEDs and 0.47uF do not operate backwards
The 6811 PLCC socket can be rotated to one of four positions (only one orientation is correct)
Damaged PLCC socket (do not probe the top side of the 6811 placed in the PLCC socket)

4. Add a delay while transferring object code from the PC to the 6812 to the 6811. In Hyperterminal,
Execute File->Properties
1) Click the settings tab
2) Click the ASCII Settings button
3) Add 1ms delay after each character