CMPE 121
Microprocessor Design

Darrell Ross
sdr85@cats.ucsc.edu

Table of Contents

Part 1 Introduction.....2

Part 2 Basic Design 2

1. Memory Map 2
2. Layout 3

Part 3 Physical Board Work 3

1. Power Supply Issues 3
2. Clock and Reset Issues.....4
3. Troubleshooting 4

Part 4 Enhancement.....4

1. Ideas 5
2. Equalizer 5

Part 5 Programming.....6

1. External SRAM Tester.....6
2. RTI Timer 6
3. Full Duplex Interrupt 6

Part 6 Conclusion 6

1. Introduction

My initial understanding of this class was from its title, "Introduction to Microprocessor Design." I thought that we would be building a Microprocessor or at least learning how to do so. However, I quickly learned that we were using a Microprocessor and therefore probably were not building a Microprocessor but building a Microcontroller instead. I was further mixed up when I learned that the 68HC11A8 is actually a Microcontroller itself. So, I finally understood, we were using a Microcontroller to build a Microcontroller. The 68HC11A8 cannot do everything we need it to do by itself and we certainly cannot learn how to build a Microprocessor just by staring at it. I have noticed now that the Course description on Petersen's web site calls the class "Microprocessor System Design" which is a much better description of what we are doing.

I thought signing up for a 9:30am lab was a good idea because it would guarantee low attendance and it was not too early for me. It would have worked great, except that lab got changed to 8:00am. Ouch. After the change, I rarely made it to lab on time. I think the only instance I was on time was when I pulled an all-nighter starting on a Tuesday.

2. Basic Design

I was totally lost in this class for about two weeks. I was baffled by the memory map and timing diagrams. How was I supposed to arrange it? How the heck did people understand those massively complex timing diagrams?

2.1 Memory Map

My first memory map was just plane silly (Appendix A, 6). I had not understood that the EEPROM must go on top. I had also not refreshed my memory on logic design. When I did redesign my memory map the way I wanted to (see Appendix A, 19), I decided that I would need a 2x4 decoder to implement it. I saw that the map was split into four 16K sections and by running address pins 14 and 15 into a 2x4 decoder, I could address each of those chunks easily. BELS did not agree with me. When I asked them for a 2x4 decoder, they did not give me one. Not only did they not give it to me, they did not give me a reason why. I asked Mike Fahmie and he said that that usually means there is an easier way to go about the problem that I had not noticed. After working on it for ten minutes, we discovered that my memory map could be done with one NAND gate (Appendix A, 12). A major benefit of using only one NAND gate was that there were four NAND gates total: 1 for my memory map and 3 for switching from Motorola to Intel timing. So I only had to use one NAND chip.

2.2 Layout

When I got hold of the chips and the board, I immediately tried to find a nice way to arrange them. My first layout was shot down by a TA who explained that making the wires travel down "streets" to make them look neat was lower priority than getting them to go the shortest distance possible. I came up with two new layout ideas(Appendix A, 5). They both used the same concepts but one left lots of space on one side of the board and the other left lots of space on the bottom of the board. I decided the one that left space on the side was better.

After showing my idea to several people in lab, they pointed out the errors of my ways. I had been lining up chips so that they were aesthetically pleasing to the eye. Ajay pointed out to me that if I lined up my EEPROM and SRAM, the address bus could run straight across. The concept of arranging the chips to make the wiring easier made me redesign most of my layout.

The power supply was an interesting issue. I did not like the idea of drilling a permanent whole in my board. I noticed during Banana Slug Spring Fair while giving tours of Baskin Engineering, that one of the TA's used the support peg to attach the power supply at an angle. It was perfect for me. I no longer had to hurt my board and as Mike said, the peg added to the heat sink for heat draw. Recently, while running my board, I have felt the peg and found it warm! The design that I got checked off can be found in Appendix A on page 9. The corresponding wiring diagram is in Appendix A on page 17.

3. Physical Board Work

Building the board was fun. I am a perfectionist when it comes to arranging things. If I messed up soldering a socket onto the board, I removed it and soldered it again. I got very good at removing parts using the solder "sponge." I was also a perfectionist on the wiring. If a wire was too loose or had too much slack, I rewired it. Unfortunately, this added a lot of work for me. I also ran into two major problems.

3.1 Power Supply Issues

When I hooked up the power supply, I placed the ground and Vcc right next to each other. I was assured that this was okay and would not hinder my progress. That was correct, but having the ground and Vcc so close together slowed down a lot of my work. I also did not use long enough rails and had to add extras. On my board, it looks like I added the rails the way I did on purpose because it looks cool, but I had to add them because I ran out of space.

3.2 Clock and Reset Issues

The clock caused me the most trouble in this entire project (second only to procrastination). When I was ready to test my reset switch and clock, I plugged in my HC11, hooked up the power, and hooked up the oscilloscope. The reset stayed low. It was supposed to default high and only go low when I pressed the reset button. I spent about 40 hours troubleshooting my reset switch and clock. I reasoned that if my HC11 was in a permanent state of reset, then the clock would not work. In that time, I changed the wires about ten times, replaced the resistor and capacitor that the reset was using, and re-soldered the clock twice.

Petersen said that if I drew up a wiring diagram, he would help me but not until then. I wasted 20 hours trying to find the problem. Then I drew a wiring diagram (Appendix A, 17) and Petersen spent about 30 seconds looking at it before he pointed out that the capacitors on my clock (0.1 pF) were too small. I changed the capacitors and everything worked great. My reasoning had been flawed. The reset switch worked on the edge when it came up, not transparently as I had thought, so the reset could not work without the clock.

3.3 Troubleshooting

With the clock working, I was ready to test the Hello World program. This section is about the hardware troubleshooting I went through to get the Hello World program working, not about Hello World and programming(See Part 5).

The Hello World program did not work my first try. I spent several hours tracing every connection on my board to match my wiring diagram. I was delighted to find two mistakes but quite annoyed that I had to remove ten wires in order to fix them. My second test failed as well. This time, I spent three hours making sure that all my pin alignments matched those in the printouts I had on my EEPROM, SRAM, and other chips. I asked Chris (dreads) if he saw anything wrong with my design at a glance and he noticed that the top three pins of my EEPROM (pins 14, 15, 16) were not tied anywhere. I tied the pins through three dip switches effectively paging my EEPROM eight times. It worked! Yay!

4. Enhancement

As I have mentioned, I spent far too much time researching the enhancement side of this project. At first, I had no idea what qualified as an enhancement. After I learned what qualified and what a lot of students were doing, I decided that I didn't want to do

anything main-stream. I wanted to come up with something all my own. I came up with something, but as it is with many of us, I bit off more than I could chew.

4.1 Ideas

The first ideas I saw were:

- ATMEL thumb print device using a 1152 bit word
- using arrays of LED's to scroll words or characters across
- wireless communication
- combining wireless with a computer to simulate remote control
- digital signal processing with sound to do sound effects or manipulate sound
- decibel LED's

I liked a few of the ideas namely the remote control and the decibel LED's. I decided very quickly that the remote would be too much work. Not only would I have to build the remote, but I would have to interface it with either USB or PS/2 or something and then write software for whatever OS I wanted the remote to work with.

4.2 Equalizer

The decibel LED's became my main idea, only I referred to them as an Equalizer. I had misinterpreted the meaning of Equalizer. I only meant the LED frequency and decibel representation found on many equalizers but Petersen thought I meant a real equalizer with the ability to change volumes of different frequencies. I decided only to do the decibel LED's and work on DSP of the frequencies if I had time.

My main idea for the decibel LED's consisted of finding some green, yellow, and red LED's and putting them in rows to represent how loud the sound was: maybe five green then five yellow, then five red. I could make several columns like this to represent different frequencies.

I began my search for supplies. I found a perfect unidirectional microphone about 0.5cm across and then some LED Dot Matrices. The LED's Matrices were 5x8 and I could get one of red, one of green, and one of yellow. Mike Fahmie and I went over the schematics of the LED Dot Matrices I was planning to buy and found that there was no way to control them to do what I wanted.

I began looking at other ways to build the project. In the end, the best way looked like building my own array of LED's. Unfortunately, I ran out of time to design my equalizer. I have decided to continue work on it this summer if I can obtain access to the right equipment I will need to do so.

5.Programming

As seen in Part 3.3, Hello World helped me significantly debugging my board. I have run into a few recurring issues with other programs:

- moving the internal resources
- getting the reset to work on all programs

5.1 External SRAM Tester

I spent about ten hours on this program. At least half of that time was spent trying to solve the two points I just mentioned. I designed two SRAM tests. The first one moved all the internal resources up to 0x8000 (64 bytes of registers) and 0x9000 (256 bytes of internal SRAM) and used a loop to test my 32K x 8 of external SRAM from 0x0000 to 0x7000. Unfortunately, the move of the resources does not work quite right. I did not find the problem. Instead, I made a second program that carefully tests around the internal resources. It works fine. Part of my SRAM Test programming process can be found in Appendix A on pages 22 to 25.

5.2 RTI Timer

I spent about ten hours on this section as well. I got a lot of help from Chris ____, Joseph ____, Arnell ____, and Jason Holbrook. Most of my time was spent looking for and then finding small errors. For instance, I forgot to set the flag back to 1 each time the clock incremented by 1 second.

5.3 Full Duplex Interrupt

As of yet, I have not started this program. If this is all that exists on this page when I turn in this report, then there is a good chance that I did not get it working.

6. Conclusion

For the most part, my design(s) did not exist until I ran into problems along the way to a finished board. I did not read the documentation well enough, I never made it on time to lab, and I put too much time into my Enhancement idea without first finishing my basic board. Now, in the last week of the quarter, I finally understand how to do it. How to keep up with the work, how to take my engineering notes. I really wish I knew then what I know now. I would have taken much better and far more detailed notes.

I now have the engineering bug. This class was fun and got me hooked. I will be taking both CE 123 and CE 126 next year to continue learning in this field. Though my performance during the quarter in the lab was not the best, I feel that I learned more from

this class than I have in almost any of my engineering courses in the past. The only comparable class so far is CE 100.