CSCE 312, Section 501

4 November 2014

Nicholas Warner

With Jennings Fairchild

We each had a similar role. Much of our research was done together, and we validated each other’s processes. I did problem 2, and half of problem 3. Jennings did problem 1 and half of 3. As addressed, we each helped each other with our individual parts – there was a fair bit of bleed over.

In example: though I may design something, Jennings would look over it and we would both debug any problems. We aimed for a quality control environment that was highly adaptable to change, since each of us is incredibly busy.

Because of that mentality, the team based approach was highly beneficial. We could flex workloads around as need dictated. I don’t believe there were really any disadvantages – since we were only two people, accountability for workload was a no-issue. If I had to find a disadvantage, it would be in our confusion over what Problem 2 was asking. I’m not sure this was because there were two minds on it though – if anything that opened the door to what we did complete.

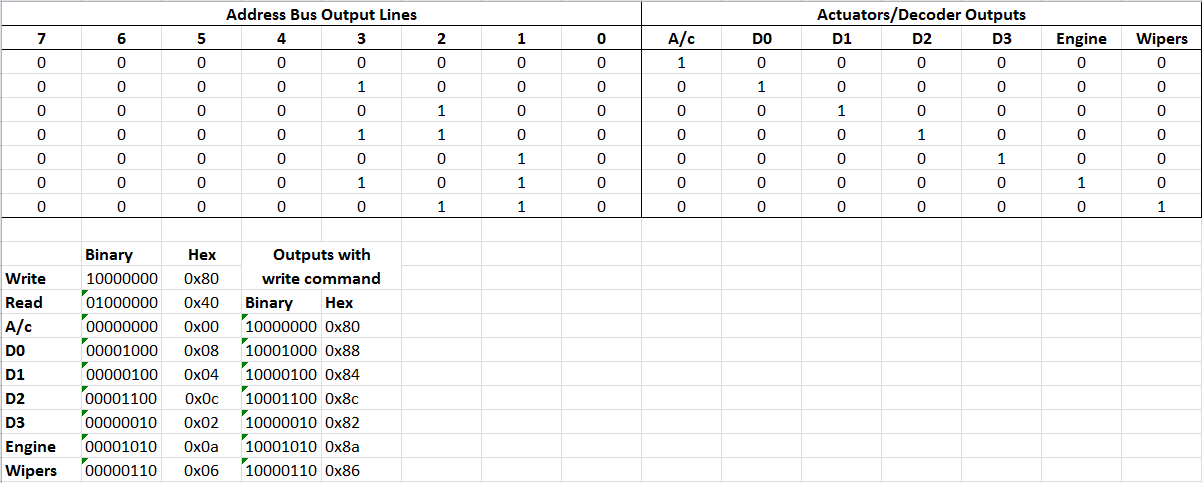
Also due to being only two people, there was no superior design. We combined and edited as needed to reach a suitable design, as opposed to having a run off. In many ways, we took a page from the Lean Six Sigma methodology in trying to eliminate any of the eight wastes as relevant to us (a graphic and general overview can be viewed [here](http://en.wikipedia.org/wiki/Lean_Six_Sigma)). In a team as small as ours, with limited time, making separate designs would be considered overproduction.

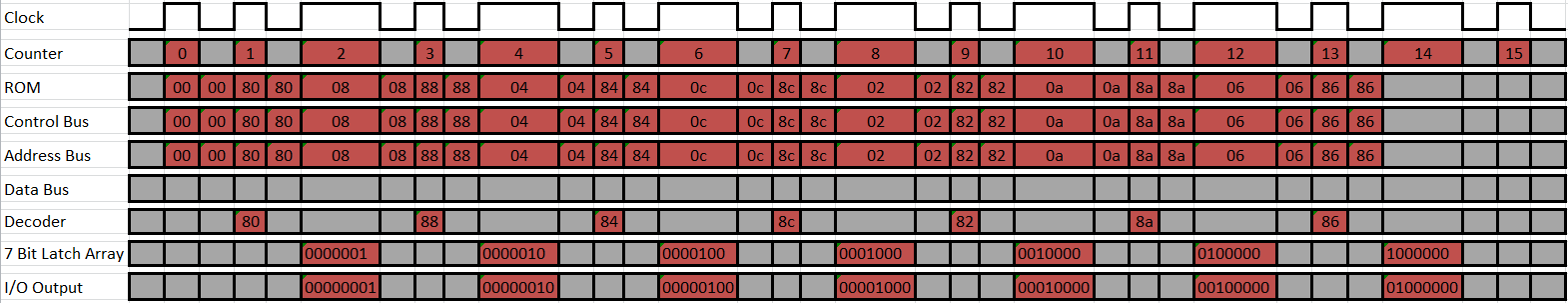
Problem 1

1. See [prob1 FSM.png](prob1%20FSM.png)
2. See [prob1.2.circ](prob1.circ)
3. See [prob1.3.circ](prob1.circ)
4. N/A (submitted along with this report)

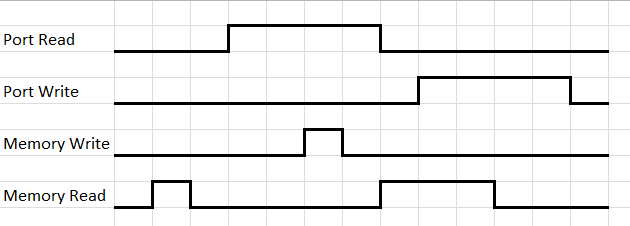
Problem 2

1. See prob2.circ
2. See prob2.circ
3. Run the TESTRUN file in the ROM of prob2.circ to verify the working of the IO System.
4. See prob2.circ
5. The only sub-systems implemented thus far are the IO System and ROM, which follows the Truth Table and instruction set [here](TruthTable.xlsx) and below, and the following signal timing diagram [here](TimingDiagrams.xlsx) and below.





1. Operations
   1. Pseudo-code
      1. Read ROM
      2. Read digitized analog input
      3. Read from digital input address 0xFF
      4. Write to RAM to memory at “x”
      5. Read from RAM value at memory location “x”
      6. Write to digital output address 0x0A
      7. Write analogized digital output
   2. Task Timing Diagram



1. C-code template
   1. unsigned int x;
   2. #define INPORT 0xFF
   3. #define OUTPORT 0x0A
   4. for ( ; ; )
   5. {
   6. x = inb( INPORT );
   7. outb( x, OUTPORT );
   8. }
2. Synchronous and asynchronous control IO methods, advantages and disadvantages
   1. Essentially, synchronous I/O waits until the I/O is complete to move on to the next step. With asynchronous I/O, the process doesn’t have to wait for the I/O to complete to move on to the next step. This method can be used to increase program response time, especially when executing methods known to take a large amount of processing. The only downside to this is more complex programming required to handle the data flow, especially when considering the execution of a process is not necessarily known. From the User standpoint, synchronous is more secure, since the I/O results appear “instant” to the user. Asynchronous is considered less secure since user processes execute in parallel with the I/O processed, allowing the user access to the data before processing is complete.

Problem 3

1. See <prob3.circ>