Gabriel Emerson

ELEC 4200 – Lab 10

11/4/21

Lab 10: Sequential System Design Using ASM Charts

**Goals:**

The goal for this lab is to design and create a sequential system using ASM charts and methods. ASM is a new type of state machine that differs from the previous finite state machines that we have been using for the past few weeks. This is important because these machines are frequently used to make underlying systems in real world machines. This also helps the mindset of designing a machine based on a previously written chart or diagram.

**Design Process:**

Task 1 and 2 are the same program but doing two different things. The program is to design a 3 bit x 3 bit binary multiplier. This will use an accumulator, multiplier register, counter, and shifter. Task 3 will do a similar thing, but instead we will create a 4 bit x 4 bit multiplier using a ROM file. The first thing to do is create an ASM chart and then a block diagram. Then since we create a testbench for the first task, we can compare it to the simulation given in the lab writeup. An example chart and diagram are shown in Figure 1 and 2, then the given simulation is shown in Figure 3.

Diagram

Description automatically generated

Figure 1

Diagram

Description automatically generated

Figure 2

Graphical user interface, application

Description automatically generated

Figure 3

**Detailed Design:**

Since Task 1 and 2 are the same program, the design should be the same. However, when trying to design this myself, I could accomplish task 1, but not task 2. The code I designed in shown in Figure 4. However, since I could not get the second task to work, the code given that does work in shown in Figure 5. And since we simulated task 1, the testbench used for that simulation is shown in Figure 6.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Background pattern

Description automatically generated with low confidence

Figure 4

Graphical user interface, text, application, email

Description automatically generated

A picture containing application

Description automatically generated

Figure 5

Graphical user interface, text, application, email

Description automatically generated

Figure 6

Task 3 was different by adding in a bit to make it a 4x4 bit multiplier, however, it also changed how the multiplier worked. Instead of using multiple lower level systems to add/shift bits into the correct place, task 3 uses a ROM file and the clock wizard to get the correct product. The code is shown below in Figure 7.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

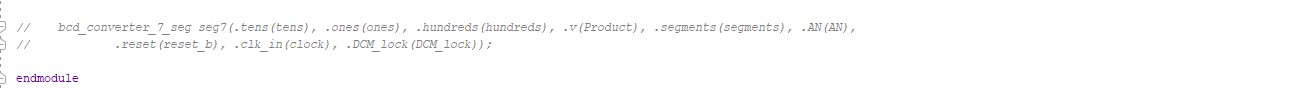


Figure 7

**Verification:**

Task 1 firstly is tested by using a testbench to simulate the results, then both of these tasks are verified by programming them to the board and visually verifying the results. When verifying, we simply load the multiplicand and multiplier in the inputs and watch to see what the output is. In the case of task 1 and 2, we can also be sure to check if the done signal has gone high when the output number is correct. The simulation results for task 1 is shown below in Figure 8.

Graphical user interface

Description automatically generated

Figure 8

**Conclusion:**

What I learned in this lab is how ASM charts can be used to design complex control units. I also designed digital system to perform binary multiplication using the ASM chart technique to develop the control unit which interfaced to the datapath processing unit. I also learned more about testbenches, the clock\_wizard, and ROM files. This lab did not go very well in comparison to the few weeks before but since this is the last lab, after learning this last bit about ASM’s, we do not need these tasks to build off another.