

[23%] Problem 1 – **ASM Charts, Data Processing & Circuit Implementation**

In this problem we will build a reaction-time testing circuit. The problem text is quite long because I will be explaining things in considerable detail; however the final circuit will *not* be particularly complicated. Read the problem very carefully, and try to proceed as systematically as possible.

Basically the circuit will wait for a few seconds, then turn on a light, and then count how many clock cycles it takes before you push a switch.

There are two inputs from the user:

1. Reset (i.e., start over)
2. React.

There are also four outputs from the circuit:

1. Wait (don't react yet)
2. Go (react as quickly as you can)
3. Error (you responded inappropriately)
4. Outnum — an 8-bit number (your reaction time)

There is one clock supplied; it runs at 256 Hz (i.e., 256 cycles per second); note that  $256 = 2^8$ .

Here's how the circuit will operate:

- Wait for the user to press “Reset”.
- Turn on “Wait”; turn off “Go” and “Error”. Wait for four seconds (in a good reaction-time testing circuit this waiting period should be random or something, but to keep things simple we'll just wait for four seconds).

If the user presses “React” any time during these four seconds, turn on the “Error” output.

- After the four seconds pass, turn off “Wait” and turn on “Go”. Wait for the user to press “React”. Count clock cycles while waiting.
- If the user presses “React” before one second is up, then output the number of clock cycles on “Outnum,” the eight-bit output (you don't have to show any LEDs or seven-segment displays or anything; just have an eight-bit quantity leaving your circuit and label it as “Outnum”).

If a whole second passes and the user hasn't pressed “React” then output zero on “Outnum”.

Basically we have three states here (although you may find using four states to be more convenient; the ASM Chart in my solution had four states). I'll get you started on your ASM Chart:

**State 0:** When user presses "Reset" go to State 1.

**State 1:** Count seconds using an eight-bit counter.

If the user presses "React" turn on "Error" and return to State 0.

Once four seconds are up, go to State 2.

**State 2:** Count clock cycles until the user presses "React".

If the user presses "React", let "Outnum" equal the number of clock cycles and go back to State 0.

If one second passes, let "Outnum" equal zero and go back to State 0.

Here's what you need to do:

- [6] a) Write down an ASM chart for this problem.
- [6] b) Draw the data-processing logic. Identify clearly all controls from the control logic, all status lines going back to the control logic, and any inputs or outputs to the outside world.
- [4] c) Write down the current-state — next-state table.
- [7] d) Implement the circuit, using whatever method you prefer. I recommend the MUX / Reg / Decoder approach, but any of the methods which we discussed in class will work fine.