**Lab 3 – Multiple Elevators – Test Plan**

The test cases start simple and only test for single elevator cases or only handle floor calls. Then they get more complex with multiple floor calls and passenger destinations. For the sake of simplicity in simulation we choose to have 2 elevators in a 10 floor building.

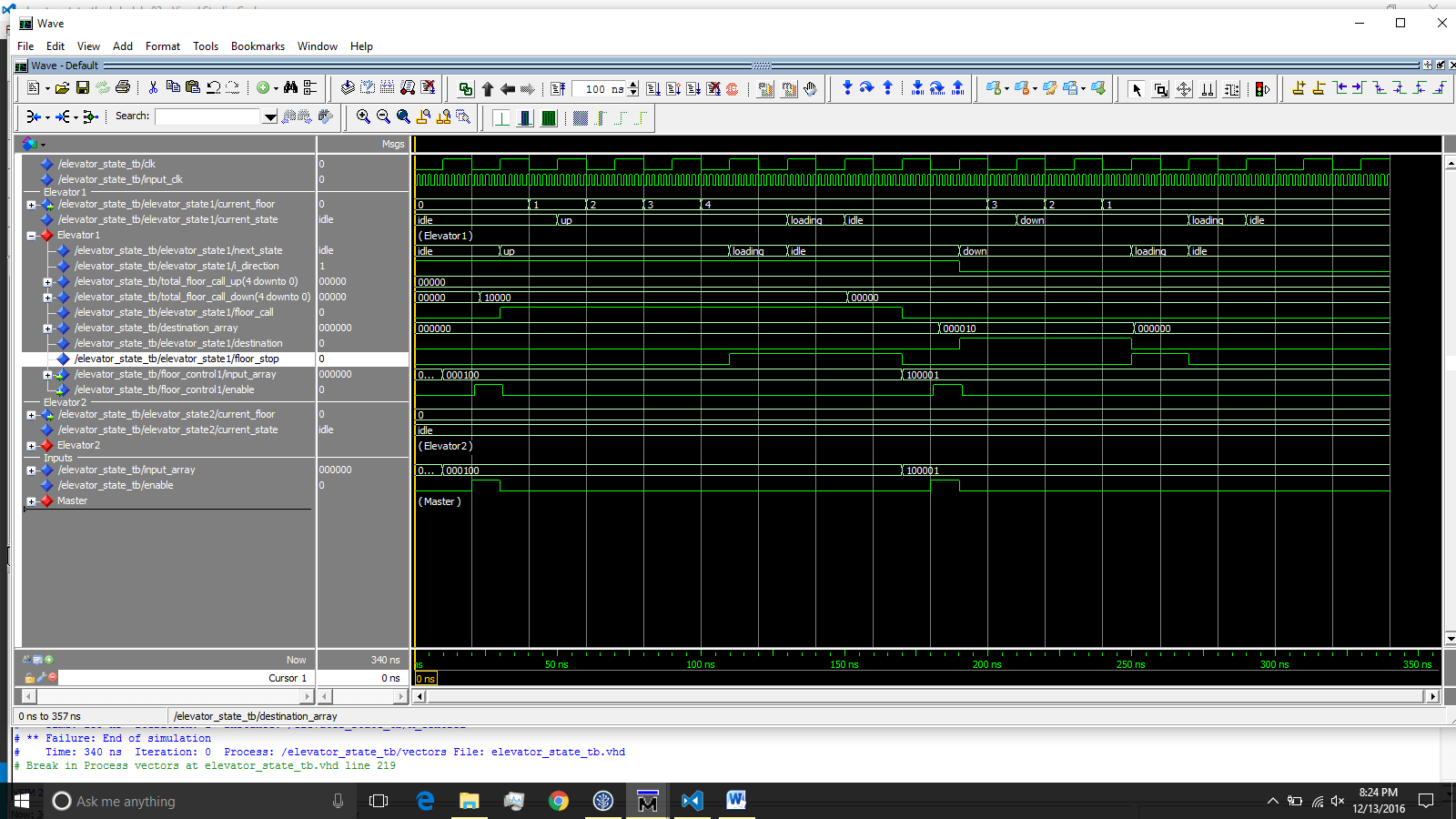
Case 1: Both elevators at idle on ground floor. The following inputs are pressed:

* Floor call down at floor 4
* Passenger presses floor 1 as their destination

Expected result:

* Elevator 1 receives the floor call and heads up to floor 4
* A Passenger enters Elevator 1
  + Presses floor 1 as destination
* Elevator 1 heads down to floor 1
* Elevator 2 remains idle this whole time

Output Wave:



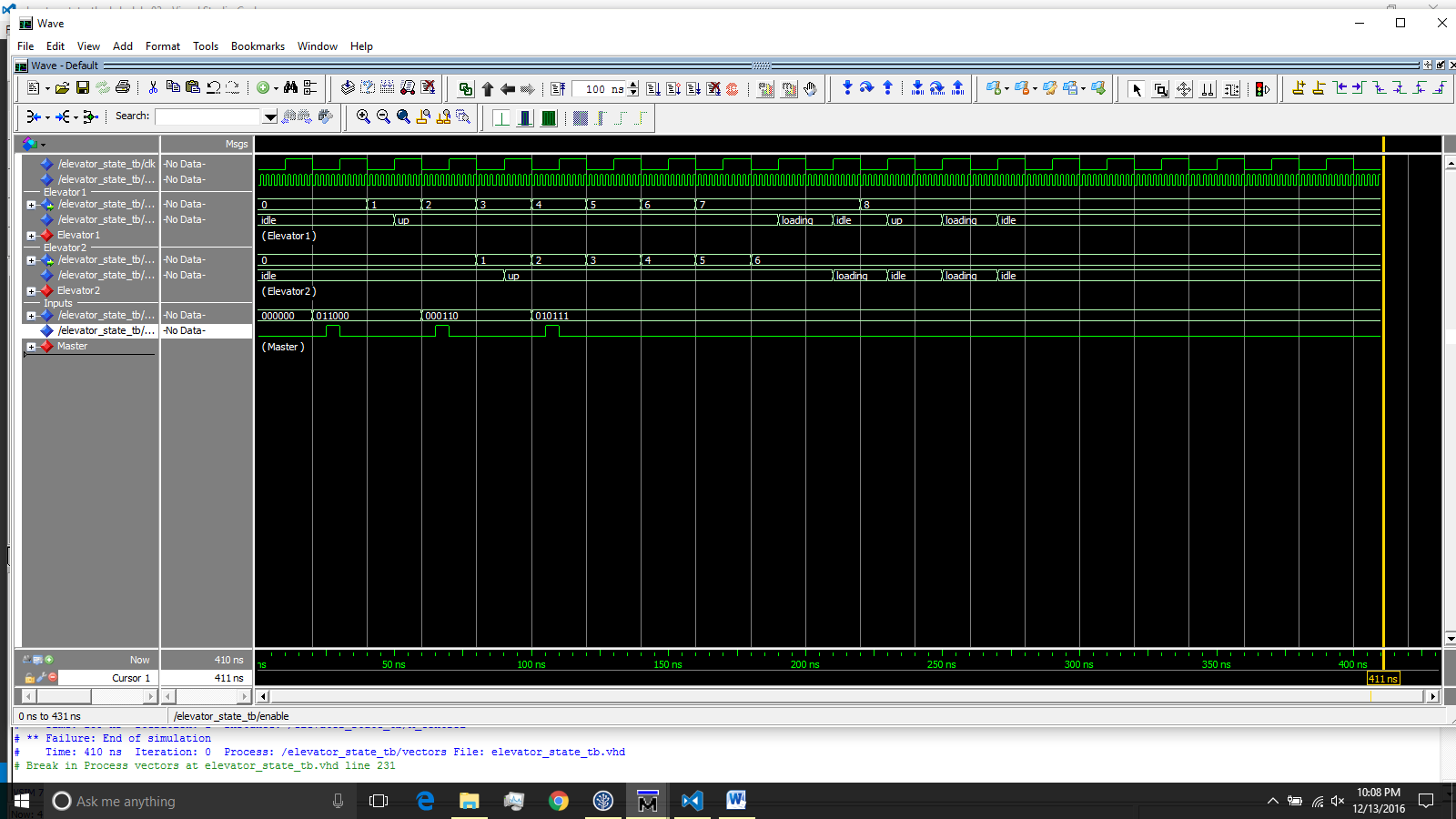
Case 2: Both elevators initially idle on floor 0. The following inputs are pressed:

* Floor call up at floor 8
* Floor call down at floor 6
* Floor call up at floor 7

Expected result:

* Elevator 1 heads towards floor 8
* Elevator 2 heads toward floor 6 and stops there.
* Elevator 1 stops at floor 7 on it’s way to floor 8

Output Wave:



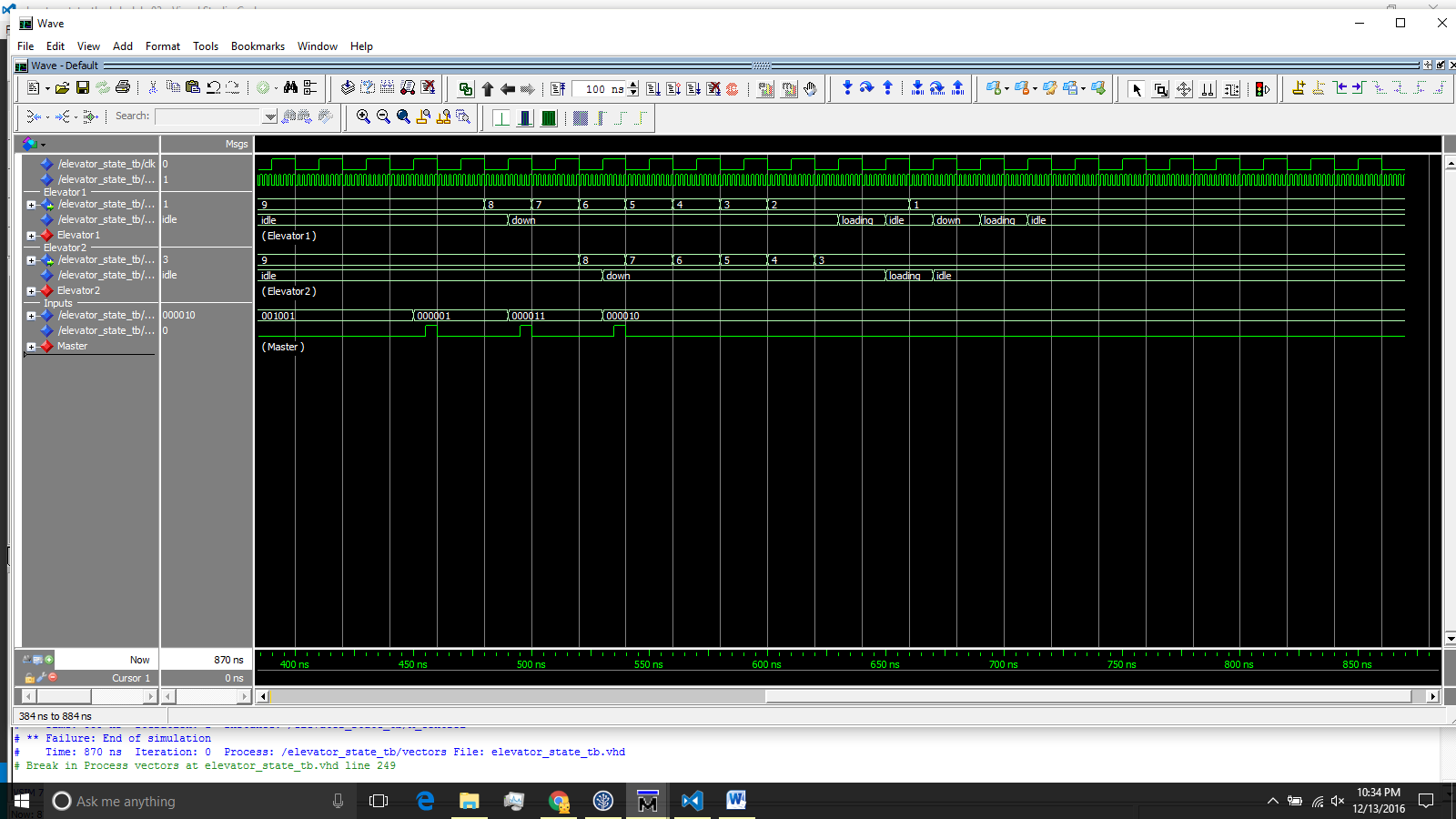
Case 3: Both elevators start at the top floor. The following inputs are pressed:

* Floor call down at floor 1
* Floor call down at floor 3
* Floor call down at floor 2

Expected Result:

* Elevator 1 heads to floor 1
* Elevator 2 heads to floor 3 and stops there
* Elevator 1 stops at floor 2 on its way to floor 1
* Then both elevators rest in the idle state

Output Wave:



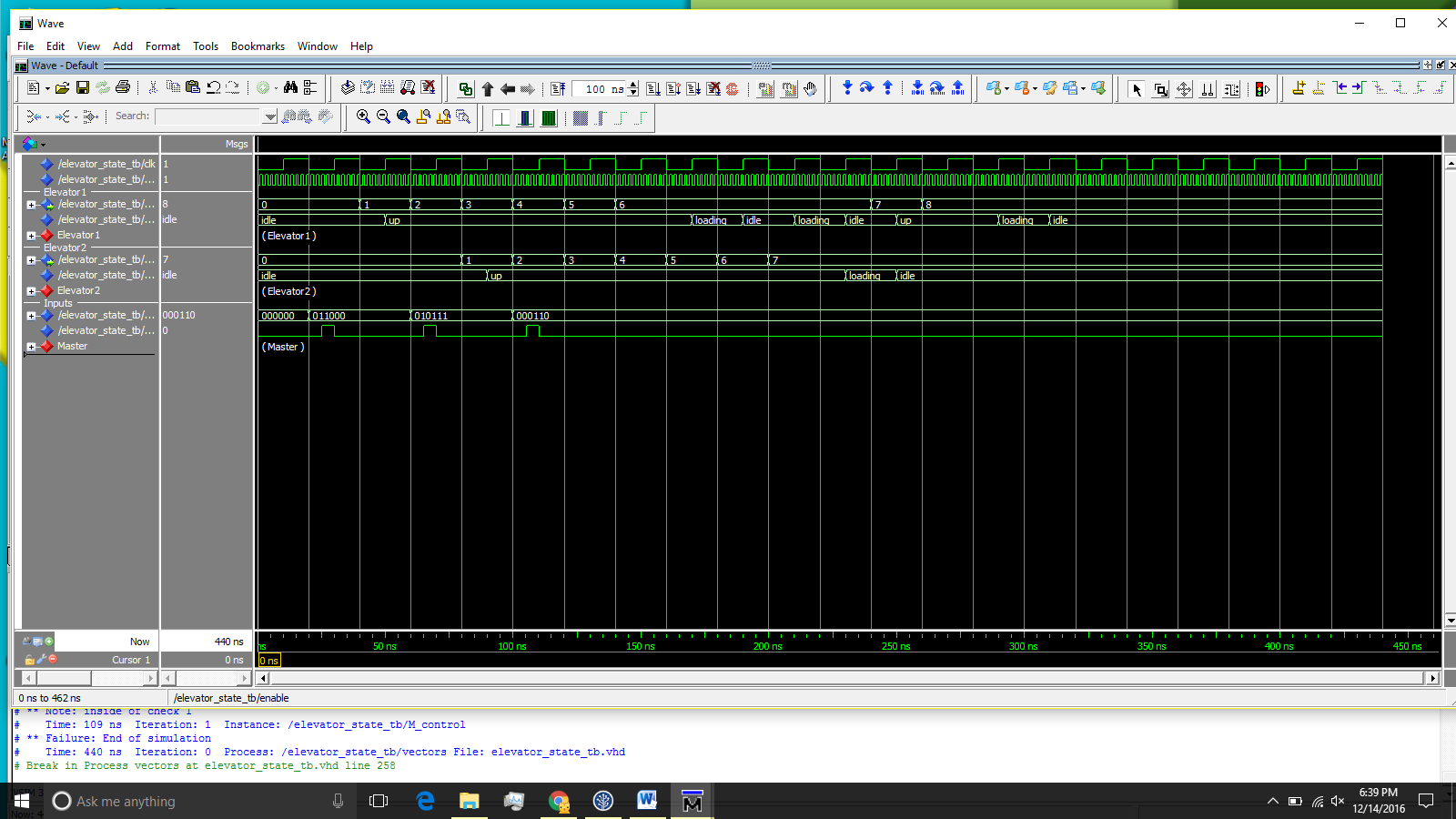
Case 4: Both elevators start at Ground Floor. The following inputs are pressed:

* Floor call up at floor 8
* Floor call up at floor 7
* Floor call down at floor 6

Expected Result:

* Elevator 1 heads towards floor 8
  + Passes floor call at floor 6 because it is a call in the opposite direction
* Elevator 2 heads to floor 7
* Elevator 2 stops at floor 7
* Elevator 1 stops at floor 8
* Elevator 1 heads to and stops at floor 6

Output Wave:



Actual Result:

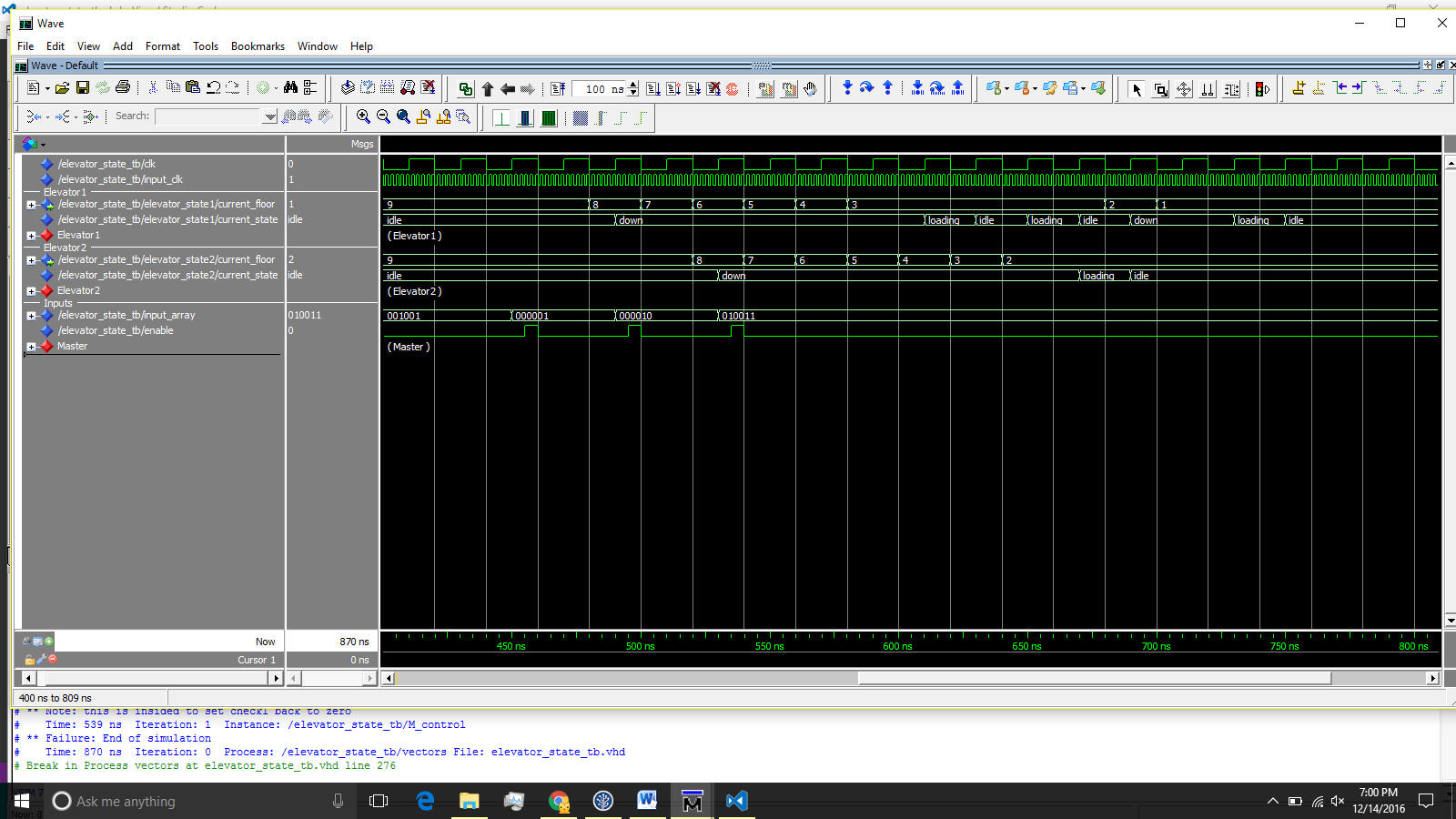
* Elevator 1 heads to floor 8
* Elevator 1 stops to pick up the call at floor 6
* Elevator 1 continues up to floor 8
* Elevator 2 stops at floor 7

What went wrong: Elevator 1 was supposed to skip the floor call down at floor 6 because it was already heading up to floor 8. We don’t want an elevator to change direction before it reaches its initial destination. This could have been fixed if we had more time to add “smart” logic.

Case 5: Both elevators start at top floor. The following inputs are pressed:

* Floor call down at floor 1
* Floor call down at floor 2
* Floor call up at floor 3

Expected Result:

* Elevator 1 heads to floor 1
* Elevator 2 heads to floor 2
* Elevator 2 stops at floor 2
* Elevator 1 loads passengers at floor 1
* Elevator 1 continues up to floor 3

Output Wave:

Actual Result:

* Elevator 1 heads to floor 1
* Elevator 1 stop to pick up the call at floor 3
* Elevator 1 continues down to floor 1
* Elevator 2 heads to floor 2 and stops there

What went wrong: Elevator 1 was supposed to skip the floor call at floor 3 because it was to go in the opposite direction that the elevator was heading. This is basically the same case as Test Case 4 but in the opposite direction.

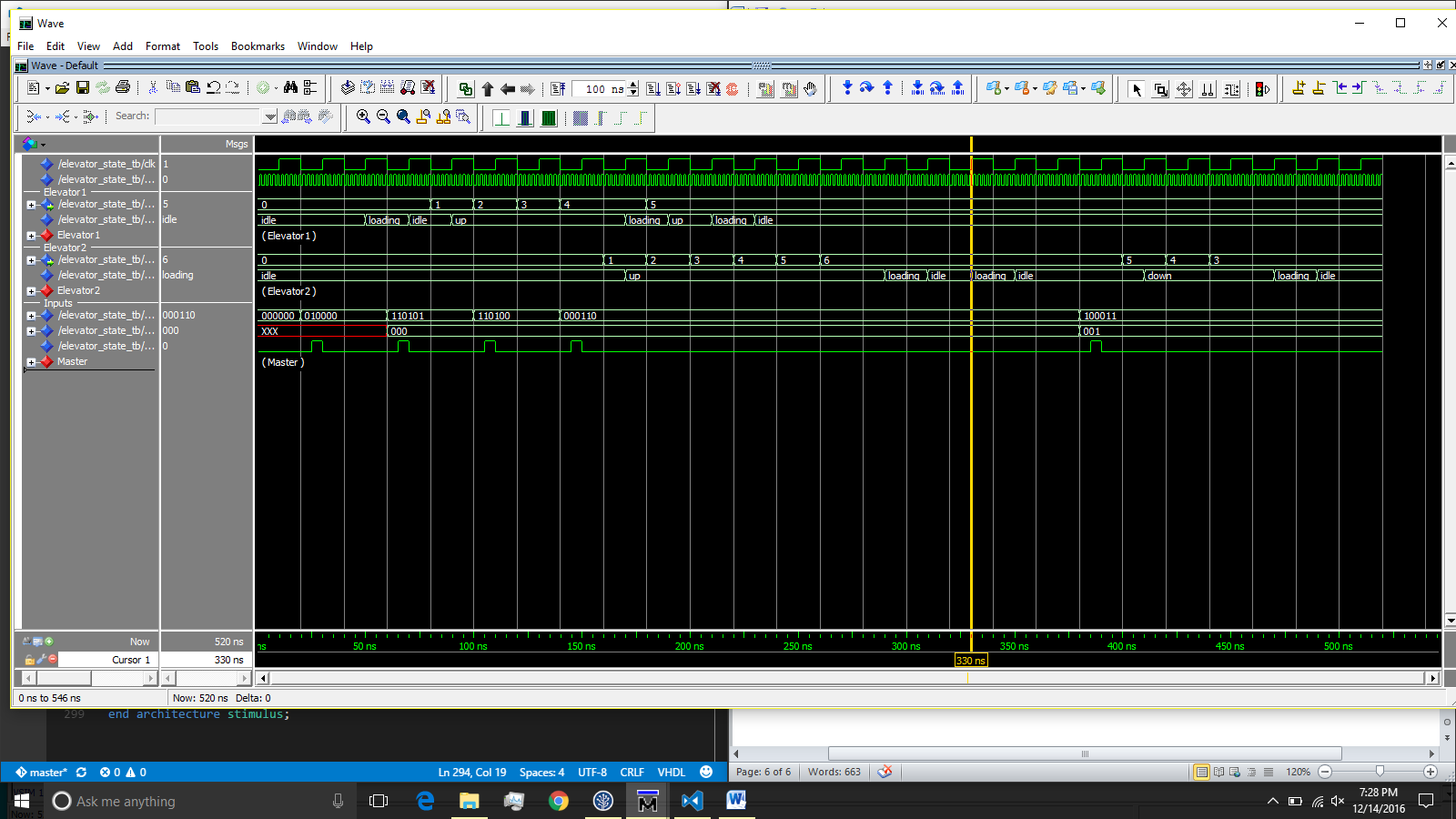
Case 6: This test case tests how multiple elevators can handle floor calls with passengers selecting destinations. Both elevators start at ground floor. Following inputs are entered:

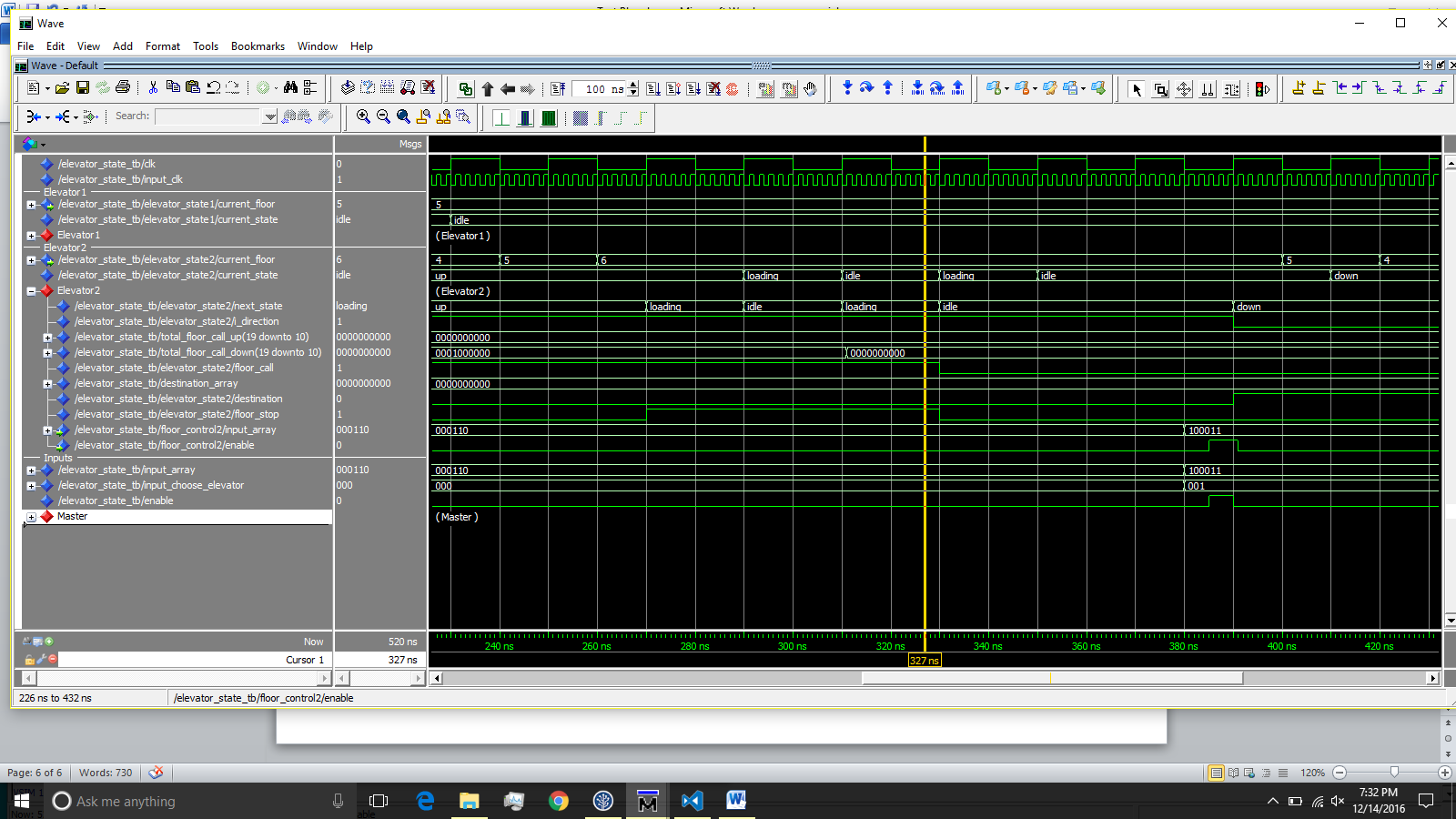
* Floor call up at Ground Floor
  + Passenger 1 presses 5
  + Passenger 2 pressed 4
* Floor call down at floor 6
  + Passenger presses floor 3

Expected Result:

* Elevator 1 picks up the floor request at the ground floor
* Two passengers enter Elevator 1
  + Passenger 1 presses 5
  + Passenger 2 presses 4
* Elevator 1 heads to floor 4 drops to drop off passenger
* Elevator 1 goes up another floor to drop off passenger 2
* Meanwhile Elevator 2 responds to the floor call at floor 6
* Once Elevator 2 arrives at floor 6 a Passenger enters
  + Passenger presses 3
* Elevator 2 heads down to floor 3

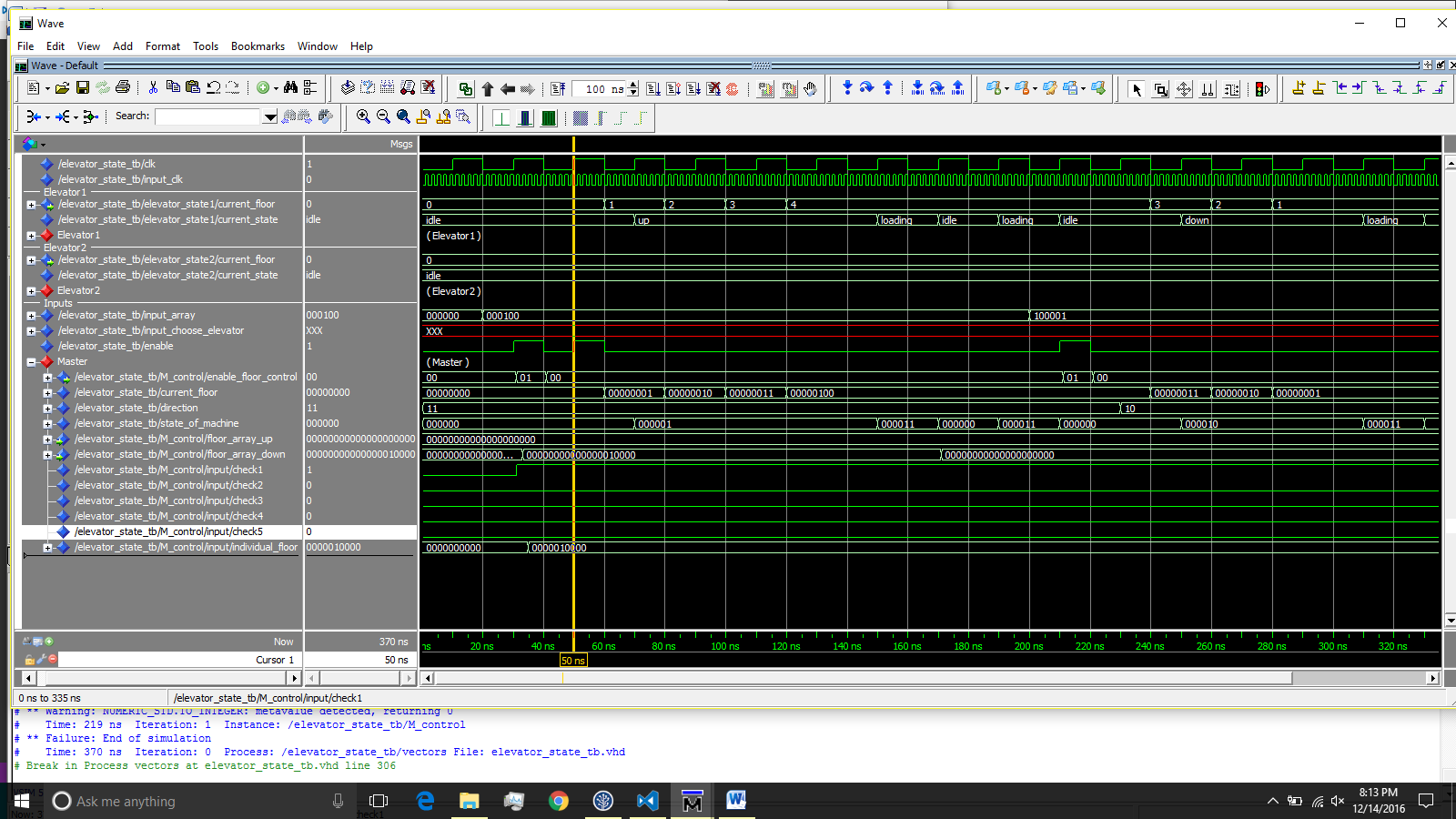
Output Wave:



Note: Sometimes the elevator goes into the loading state twice once it arrives at a floor. An example of this is seen at the yellow line above. This is a timing bug due to how we turn off floor call bits when an elevator arrives at a floor. The timing doesn’t line up in certain cases. This is also something we would have fixed if we had more time. The bits that signal for the elevator to load passengers (floor\_stop & floor\_call) go low too late which is seen just after the yellow line below.

Case 7: Same as test case 1 but the input is entered twice.

Expected Result: The elevator should behave the same as it did in the first test case. The second input should be ignored.



Output Wave:

Notes: This test case showcases the Master Control component and how it is smart enough to recognize that an elevator already is heading towards an existing floor call before sending a second elevator. The Master Control component handles which inputs go to which elevators, this is where all the “smarts” happen. This is where we could improve the most with more time.