ECE 156a HW2
Blake Johnson
Thursday 5-8pm

Introduction:

In this lab I made three main modules, a D flipflop, a 4 bit upcounter, and a 7segment display decoder. In addition to these modules, i made an SR latch and a D latch to facilitiate the D flipflop, as well as test files for each module.

Procedure:

D flipflop:
First, I designed the SR latch and the D latch. As i had seen these two latches before, the underlying concept was not difficult to understand. From there, I instantiated all the modules, and combined them to create the Dflipflop. I added an and gate to control the flipflop with a reset, and used the master/slave method of combining two flip-flops and inverting the clock between them. I ran into trouble with the inverted clock because I mixed up the ports but was able to figure it out.

Upcounter:
When I made the upcounter, I converted the D flip flops to T flip flops, and used AND gates to determine when each flipflop is incremented. The reset is passed directly to the D flip flops.

7 Segment Display:
To determine the logic for each output of the display, I made kmaps to determine the minimum SOP for each light within the display.

Testing:

D flipflop:
To test the D flip flop, I tested both the SR latch and the D latch i made because both made up my D flipflop. I tested the flip flop and latches by changing the inputs and checking the corresponding Q value to make sure it was correct.

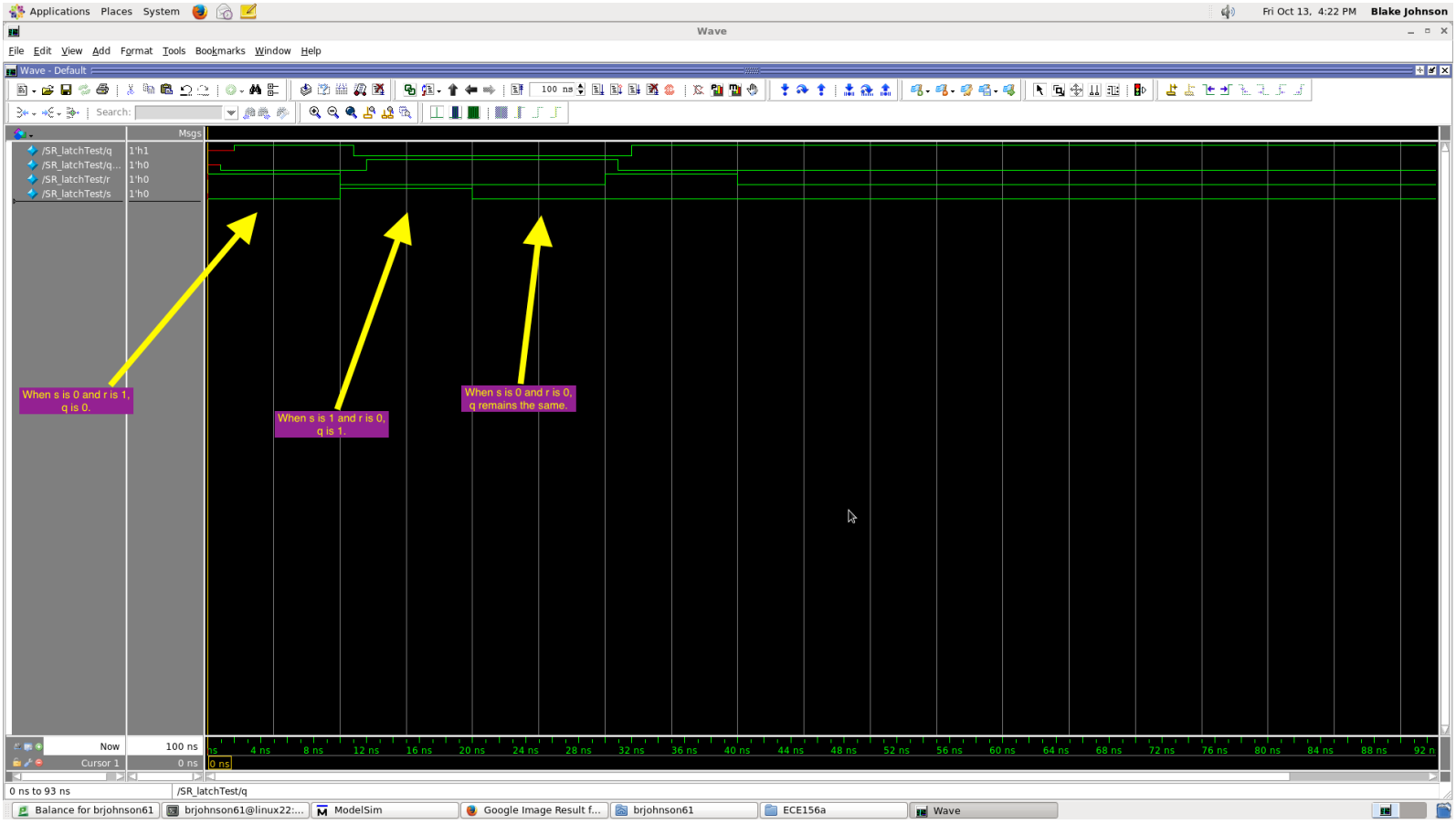
Upcounter:
To test the Upcounter, I created different scenarios by changing the enable and reset inputs. My Upcounter correctly increments from 0000 to 1111 and resets immediately when reset is high.

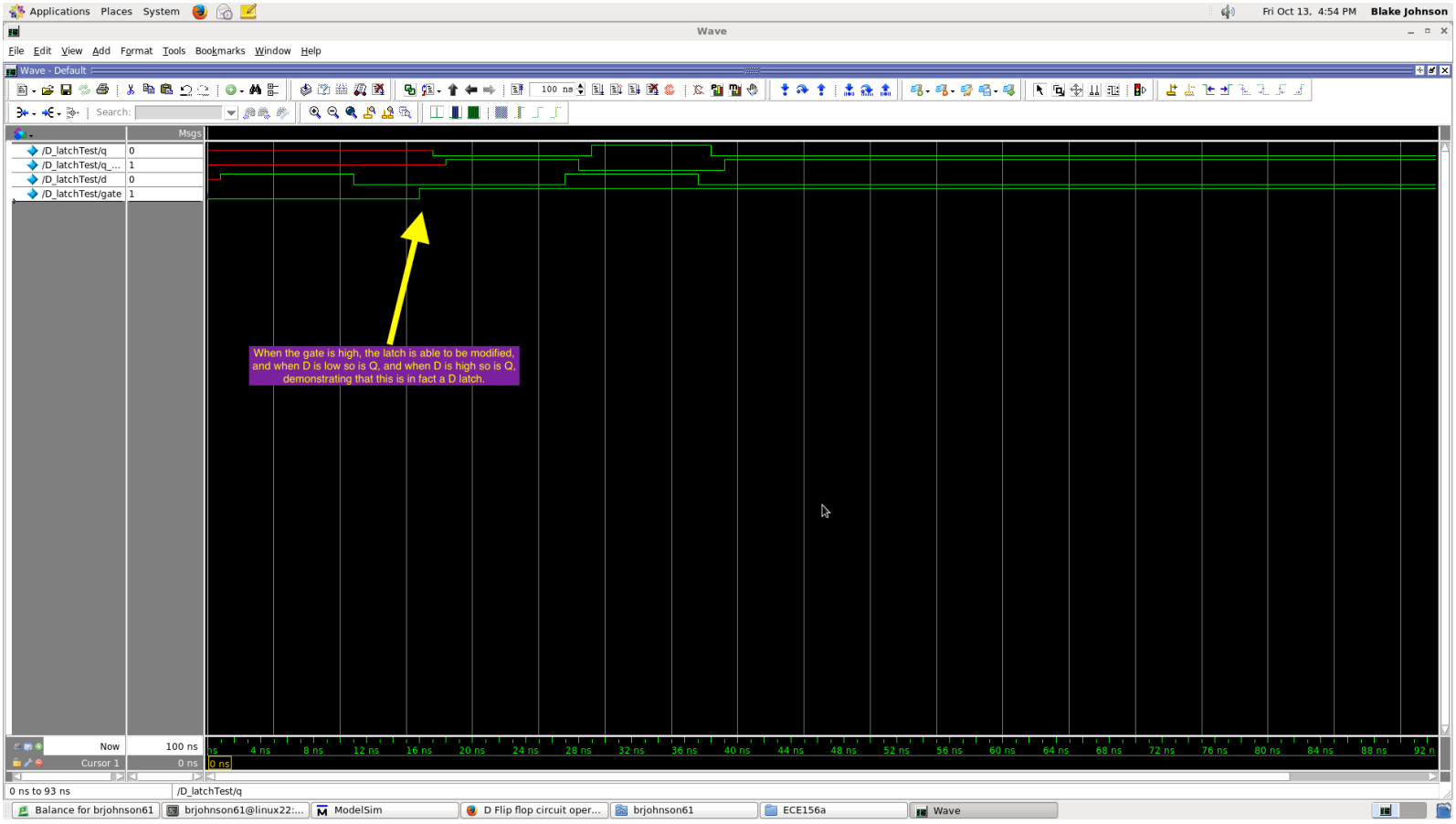
7 Seg Display:
To test the decoder I looped the control input through all possible values and showed how each corresponding display output was active low, and showed the correct output.

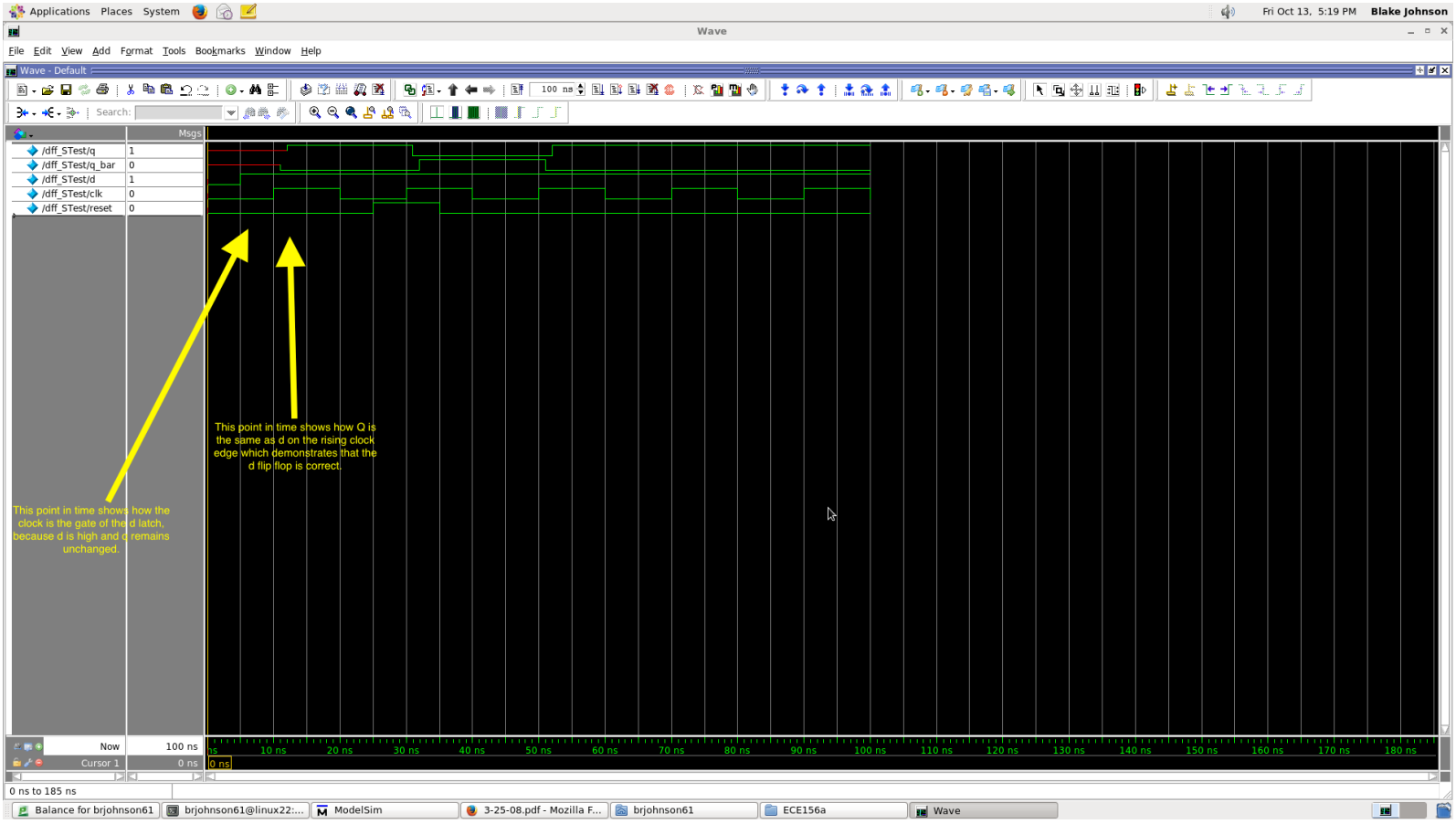
**For each test I included annotated waveforms shown below. Order of screenshots is RS latch, D latch, D flipflop, Upcounter, and finally 7 Segment Decoder.

Conclusion:

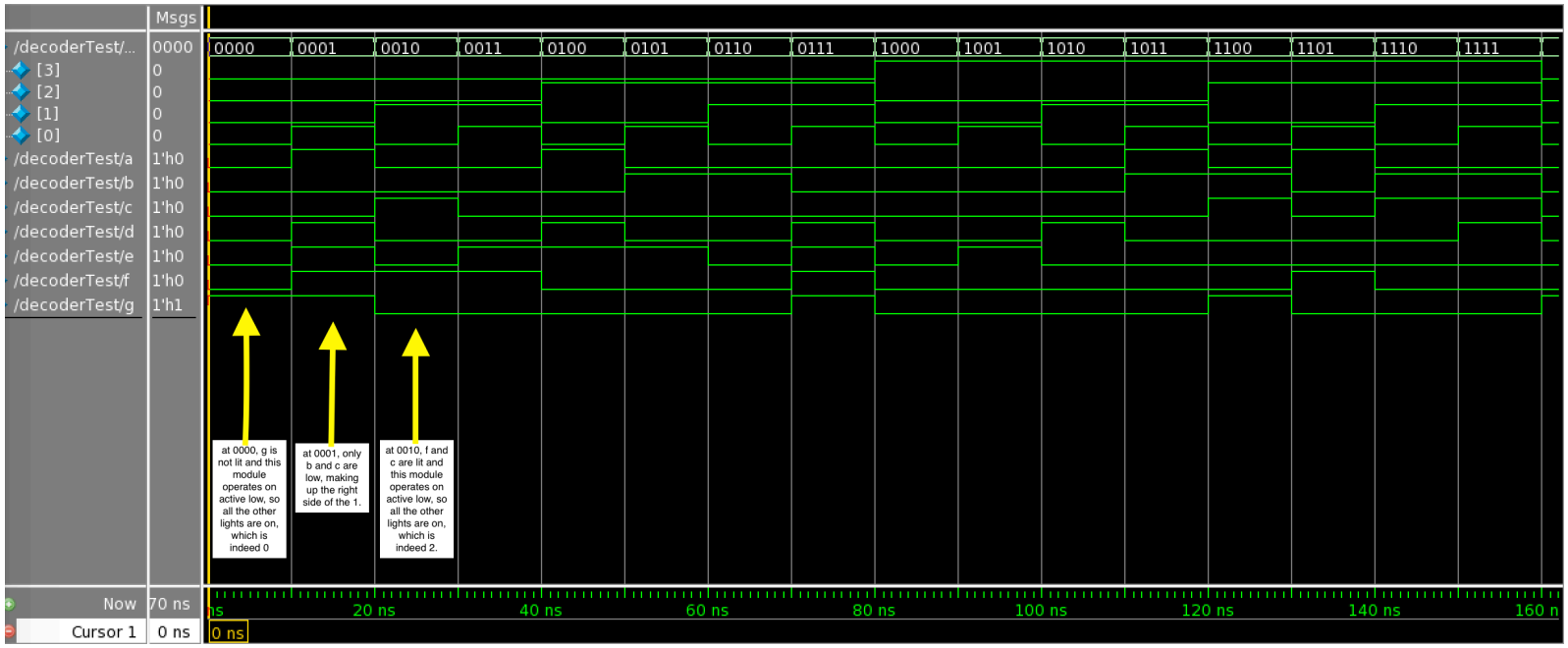
This lab/homework has taught me a lot about how to use verilog and how to compile, and effectively simulate my modules to show their correctness as well as to diagnose any problems when I have them.


Applications Places System
Fri Oct 13, 4:22 PM Blake Johnson
Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/SR_latchTest/q
/SR_latchTest/q...
/SR_latchTest/r
/SR_latchTest/s
1'h1
1'h0
1'h0
1'h0
When s is 0 and r is 1, q is 0.
When s is 1 and r is 0, q is 1.
When s is 0 and r is 0, q remains the same.
Now
100 ns
Cursor 1
0 ns
0 ns to 93 ns
/SR_latchTest/q
Balance for brjohnson61
brjohnson61@linux22:...
ModelSim
Google Image Result f...
brjohnson61
ECE156a
Wave


Applications Places System
Fri Oct 13, 4:54 PM Blake Johnson
Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/D_latchTest/q 0
/D_latchTest/q_... 1
/D_latchTest/d 0
/D_latchTest/gate 1
When the gate is high, the latch is able to be modified, and when D is low so is Q, and when D is high so is Q, demonstrating that this is in fact a D latch.
Now 100 ns
Cursor 1 0 ns
0 ns 4 ns 8 ns 12 ns 16 ns 20 ns 24 ns 28 ns 32 ns 36 ns 40 ns 44 ns 48 ns 52 ns 56 ns 60 ns 64 ns 68 ns 72 ns 76 ns 80 ns 84 ns 88 ns 92 n
0 ns to 93 ns
/D_latchTest/q
Balance for brjohnson61
brjohnson61@linux22:...
ModelSim
D Flip flop circuit oper...
brjohnson61
ECE156a
Wave


Applications Places System
Fri Oct 13, 5:19 PM Blake Johnson
Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/dff_STest/q 1
/dff_STest/q_bar 0
/dff_STest/d 1
/dff_STest/clk 0
/dff_STest/reset 0
This point in time shows how the clock is the gate of the d latch, because d is high and q remains unchanged.
This point in time shows how Q is the same as d on the rising clock edge which demonstrates that the d flip flop is correct.
Now 100 ns
Cursor 1 0 ns
0 ns 10 ns 20 ns 30 ns 40 ns 50 ns 60 ns 70 ns 80 ns 90 ns 100 ns 110 ns 120 ns 130 ns 140 ns 150 ns 160 ns 170 ns 180 ns
0 ns to 185 ns
Balance for brjohnson61
brjohnson61@linux22:...
ModelSim
3-25-08.pdf - Mozilla F...
brjohnson61
ECE156a
Wave


Applications Places System
Fri Oct 13, 6:31 PM Blake Johnson
Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/UcTest/clk 0
/UcTest/reset 1
/UcTest/enable 1
/UcTest/number 0000
[3] 0
[2] 0
[1] 0
[0] 0
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111 0000 0001 0010 0011 0000
The upcounter needs a reset to "initialize" the counter, and since enable is 1, the counter begins to count.
The counter wraps around when it gets to 1111.
The counter resets to zero correctly as shown at this point in time.
Now 720 ns
Cursor 1 0 ns
50 ns 100 ns 150 ns 200 ns 250 ns 300 ns 350 ns 400 ns 450 ns
0 ns to 496 ns
/UcTest/clk
Balance for brjohnson61
brjohnson61@linux22:...
[3-25-08.pdf - Mozilla ...
brjohnson61
ECE156a
ModelSim
Wave


Msgs
/decoderTest/...
[3]
[2]
[1]
[0]
/decoderTest/a
/decoderTest/b
/decoderTest/c
/decoderTest/d
/decoderTest/e
/decoderTest/f
/decoderTest/g
0000
0
0
0
0
1'h0
1'h0
1'h0
1'h0
1'h0
1'h0
1'h1
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
at 0000, g is not lit and this module operates on active low, so all the other lights are on, which is indeed 0
at 0001, only b and c are low, making up the right side of the 1.
at 0010, f and c are lit and this module operates on active low, so all the other lights are on, which is indeed 2.
Now
70 ns
Cursor 1
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
160 n
0 ns