8. /EBICS0 to $CE2_{propmax}$ = 12.594ns
   /EBICS0 to $CE2_{propmin}$ = 4.626ns

   For a write cycle, changing tAD will affect tCW:
   - tEBI-AS + tEBI-WP > $tCW_{min}$ + $tAD_{max}$
   - tEBI-AS + tEBI-WP > 50ns + 12.6ns = 62.6ns
   - 62.6ns / 20.833ns = 3.005
   - 3.005 means that tEBI-AS and tEBI-WP will still be 1 and 3 cycles respectively.

   For a read cycle, changing tAD will affect tEBI-RC:
   - tEBI-RC – $tEBICO_{max}$ – $tAD_{max}$ – $tACE_{max}$ >= $tEBIDS_{min}$
   - tEBI-RC – 5ns – 12.6ns – 55ns >= 5ns
   - tEBI-RC – 72.6 >= 5ns
   - tEBI-RC >= 77.6ns
   - 77.6ns / 20.833ns = 3.725
   - 3.725 means that tEBI-RC needs to be at least 4 cycles, so the parameter needs to be set to at least 3.

9.

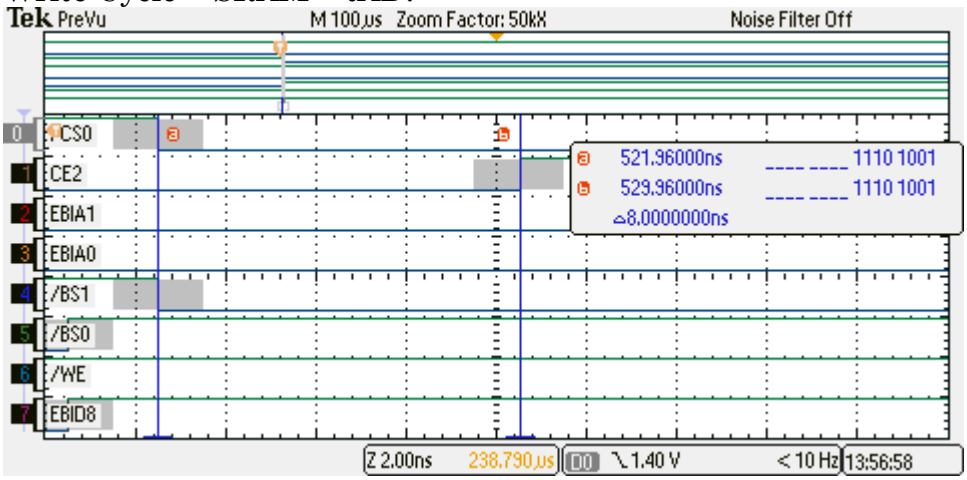
Write cycle timing:

| EBI Parameter: | Measured time: | SRAM Parameter: | Measured time: |
|---|---|---|---|
| AS | Can't see PBCLK8 | AD | 8ns (+/- 1ns) |
| WP | Can't see PBCLK8 | AS | 20ns (+/- 1ns) |
| WR | Can't see PBCLK8 | AW | 106ns (+/- 1ns) |
| ICO | Can't see PBCLK8 | CW | 98ns (+/- 1ns) |
| IDO | Can't see PBCLK8 | WP | 86ns (+/- 1ns) |
| | | WR | 18.04ns (+/- 1ns) |

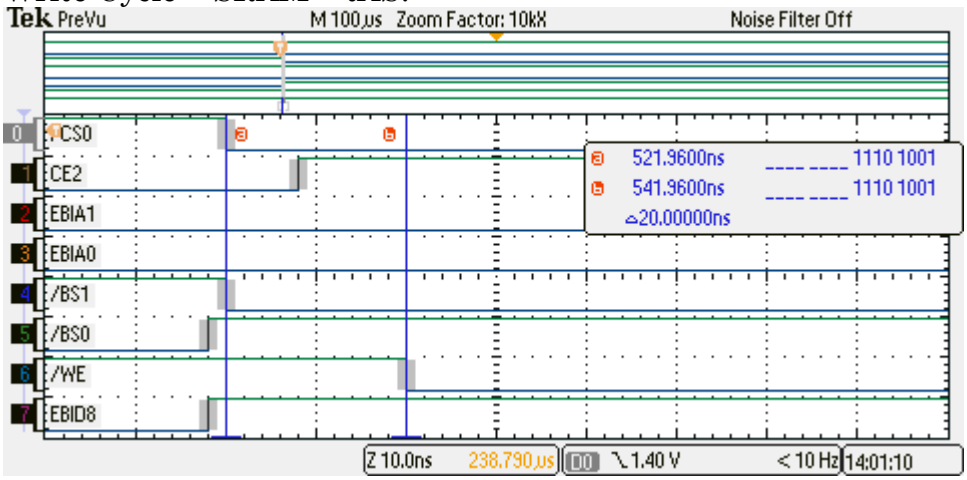
Read cycle timing:

| EBI Parameter: | Measured time: | SRAM Parameter: | Measured time: |
|---|---|---|---|
| RC | Can't see PBCLK8 | AD | 8ns (+/- 1ns) |
| ICO | Can't see PBCLK8 | AA | 29.96ns (+/- 1ns) |
| $DS_{(max)}$ | 53.96ns (+/- 1ns) | ACE | 22ns (+/- 1ns) |
| $DH_{(max)}$ | 187.96ns (+/- 1ns) | CLZ | Can't see Lo-Z |
| | | OLZ | Can't see Lo-Z |
| | | OE | 30ns (+/- 1ns) |
| | | OH | Can't see Hi-Z |
| | | CHZ | Can't see Hi-Z |
| | | OHZ | Can't see Hi-Z |

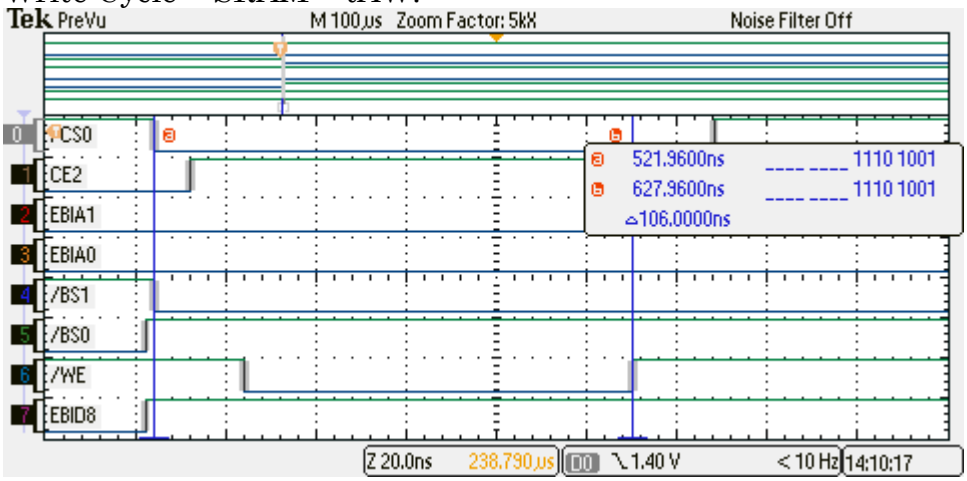
Write Cycle – SRAM – tAD:



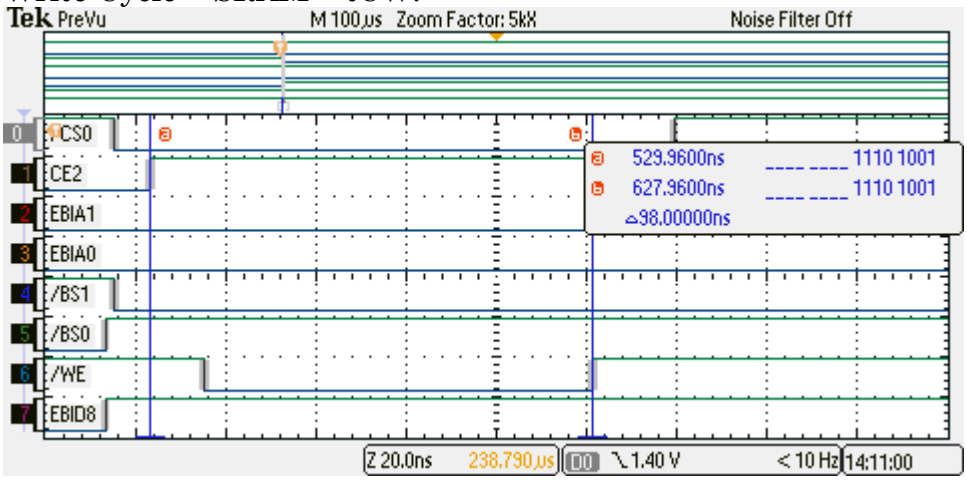

Write Cycle – SRAM – tAS:



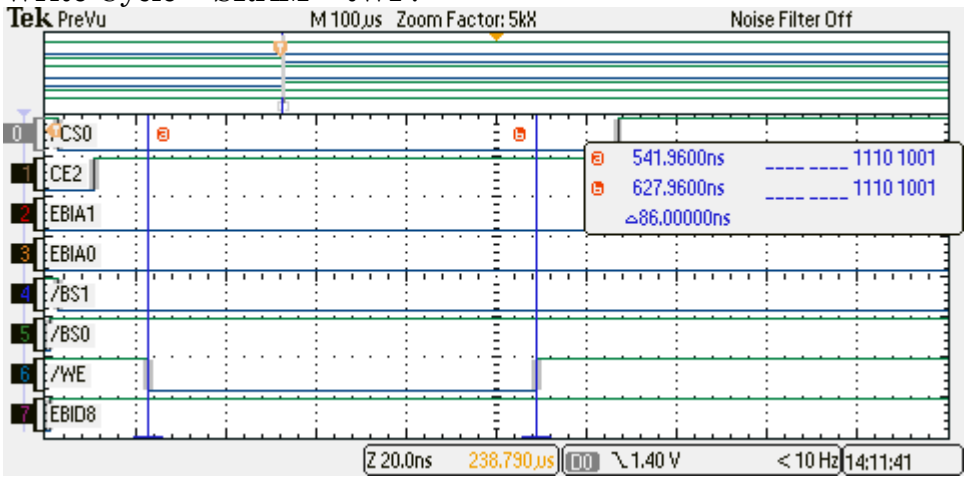

Write Cycle – SRAM – tAW:



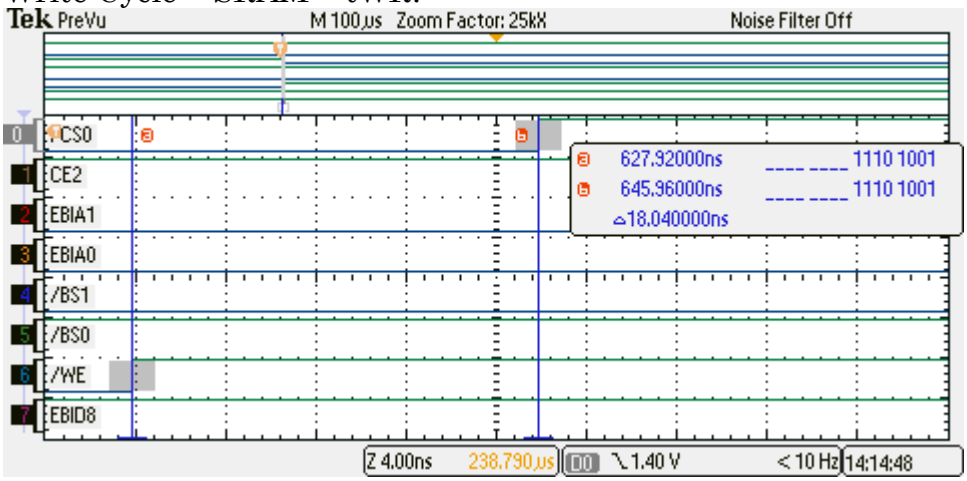
Write Cycle – SRAM – tCW:



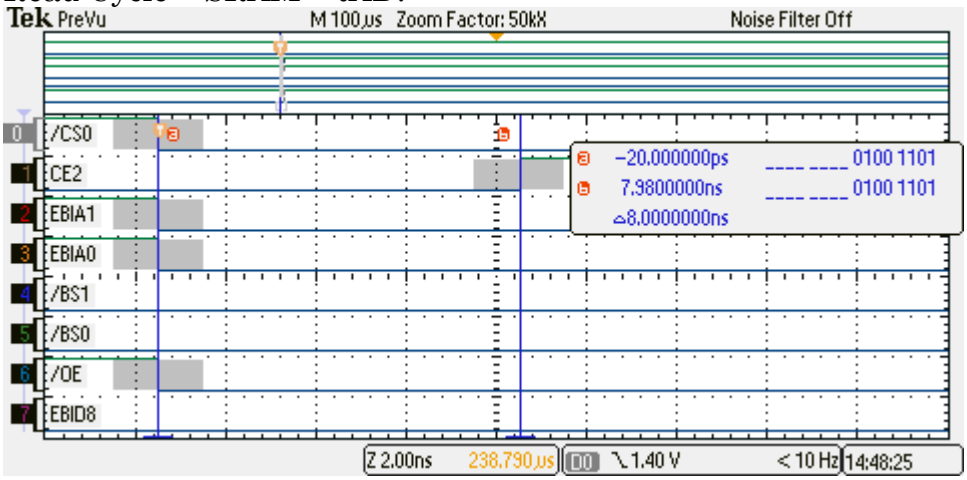

Write Cycle – SRAM – tWP:



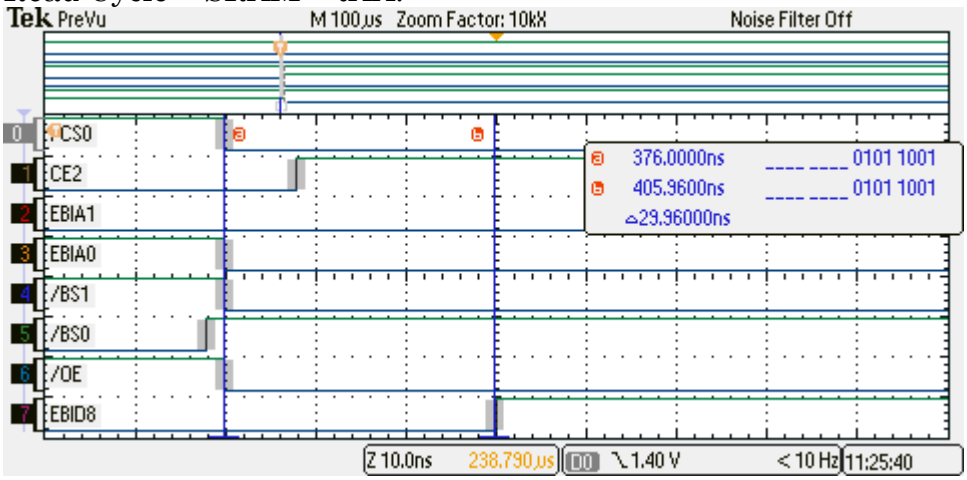

Write Cycle – SRAM – tWR:



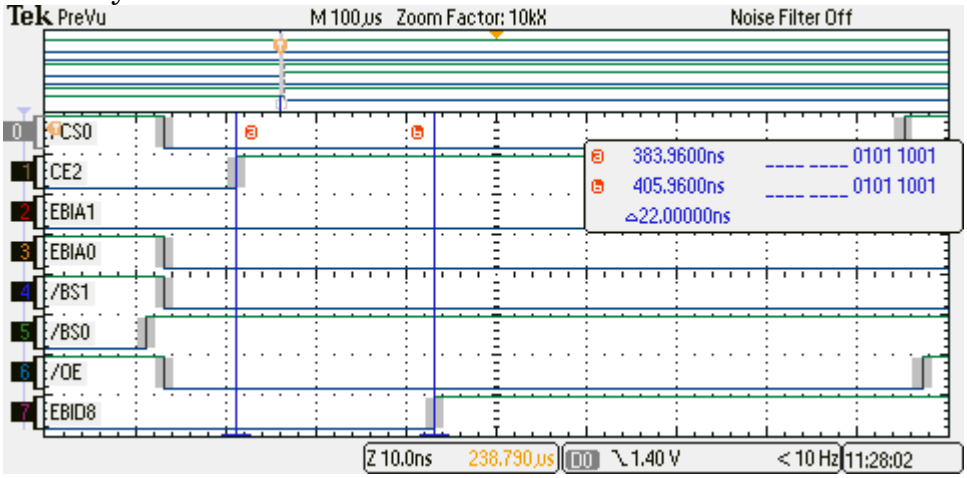
Read Cycle – SRAM – tAD:



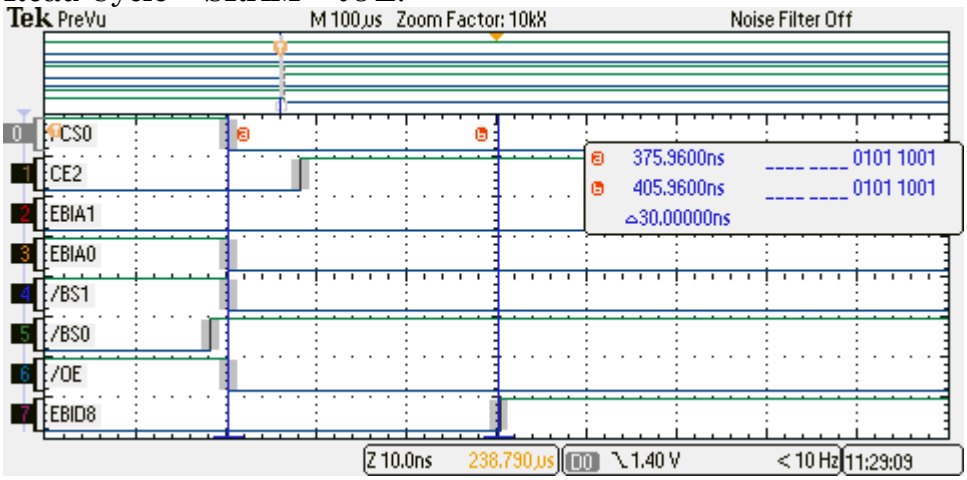

Read Cycle – SRAM – tAA:



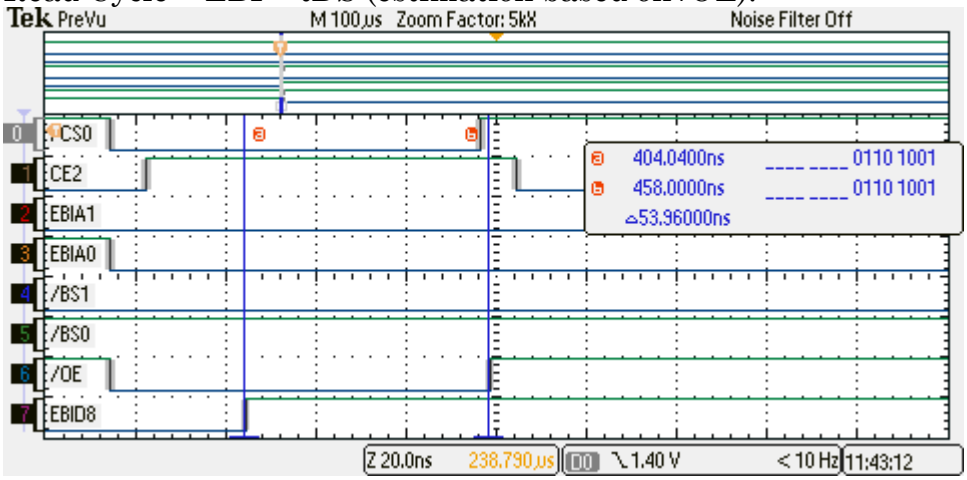

Read Cycle – SRAM – tACE:



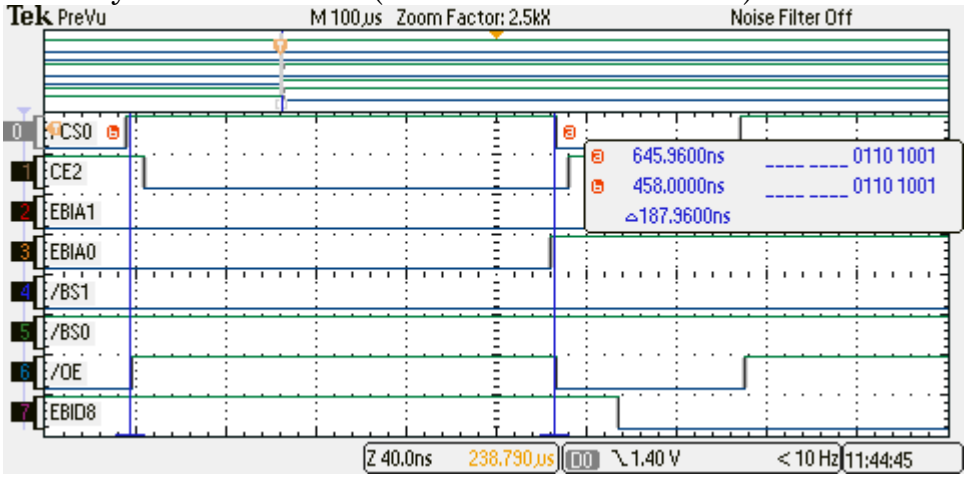
Read Cycle – SRAM – tOE:




Read Cycle – EBI – tDS (estimation based on /OE):




Read Cycle – EBI – tDH (estimation based on /OE):



Lab Report Section:

1. Code printouts included.

2. Circuit diagram included.

3. Construction and testing steps:
   - Created circuit diagram showing all pin connections before wiring
   - Wire wrapped connections based on circuit diagram and test continuity
   - Programmed FPGA code before connecting to the rest of the system
   - Tested functionality of the FPGA code by using the oscilloscope and tying /EBICS0 to ground to see CE2 go high, then tying /EBICS0 high to see CE2 go low
   - Added code to exercise the EBI functionality and used the oscilloscope to check for proper signals at the U3 and U4 memory pins before adding the modules
   - Added memory modules and tested them with the various requirements described in the lab assignment

4. TimingQuest analysis included on Page 1.

5. Screenshots included from Page 1 to Page 5.
   - tRC cycles: 4
   - tAS cycles: 1
   - tWR cycles: 1
   - tWP cycles: 4

   All measureable timing requirements are met. Measured timing parameters included on Page 1.

6. This lab seemed to be more time consuming than difficult, so I didn't have many problems other than allotting time to work on the lab. Once wire wrapping was complete, I tested continuity and had no apparent issues. While the FPGA code isn't complex, it was nice to see the CS signal being inverted on the scope. When testing the memory pin signals, I thought my bank selects might be backwards so I swapped them. However, once I created some test code, my memory wasn't functioning as expected. This was because I swapped my bank selects, so after swapping them back to their original pins my program worked as expected. I also ran into some confusion when measuring the timing parameters; for example, trying to find a way to measure the parameters that are based off the impedance of the data bus when we can't actually measure them.

7. In order to prevent aliasing, EBIA13 and EBIA14 would enabled and connected to the FPGA. In the Verilog code, there would be a check if EBIA13 and EBIA14 are anything but 0. If they are anything but 0, don't assert the CE2 line. The EBI will still run bus cycles because it only cares if EBICSx is asserted (not CE2), but nothing will happen with the connected device since the chip enable isn't asserted.