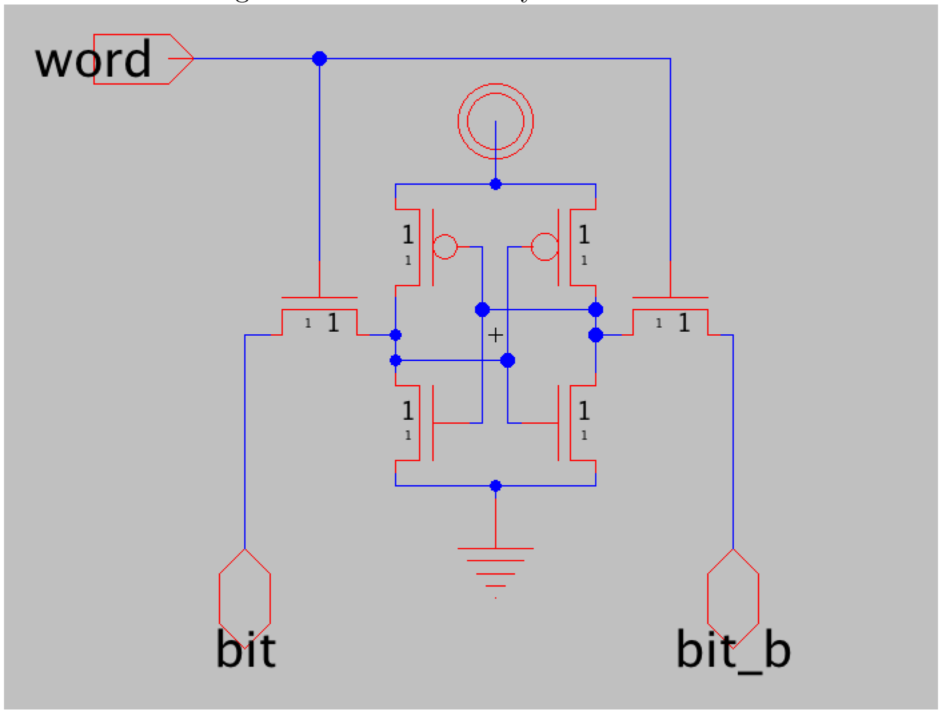
1. **Bit Line Capacitance**
   We estimated our bit line capacitance using the basic circuit for an SRAM memory cell. By looking at this schematic, we see that our bit line is connected to a pass transistor, which is connected to two of the gates of the coupled inverter. To estimate the bit line capacitance, we used the Elmore delay method to calculate the capacitance in terms of C_o, which we found to be $(4\gamma + 2)C_o$. This takes into consideration that our $C_{diff} = \gamma * C_o$.

Figure 1: SRAM Memory Cell Schematic



2. Column Driver and Memory Cell Schematics

Figure 2: Basic Cell Test Schematic

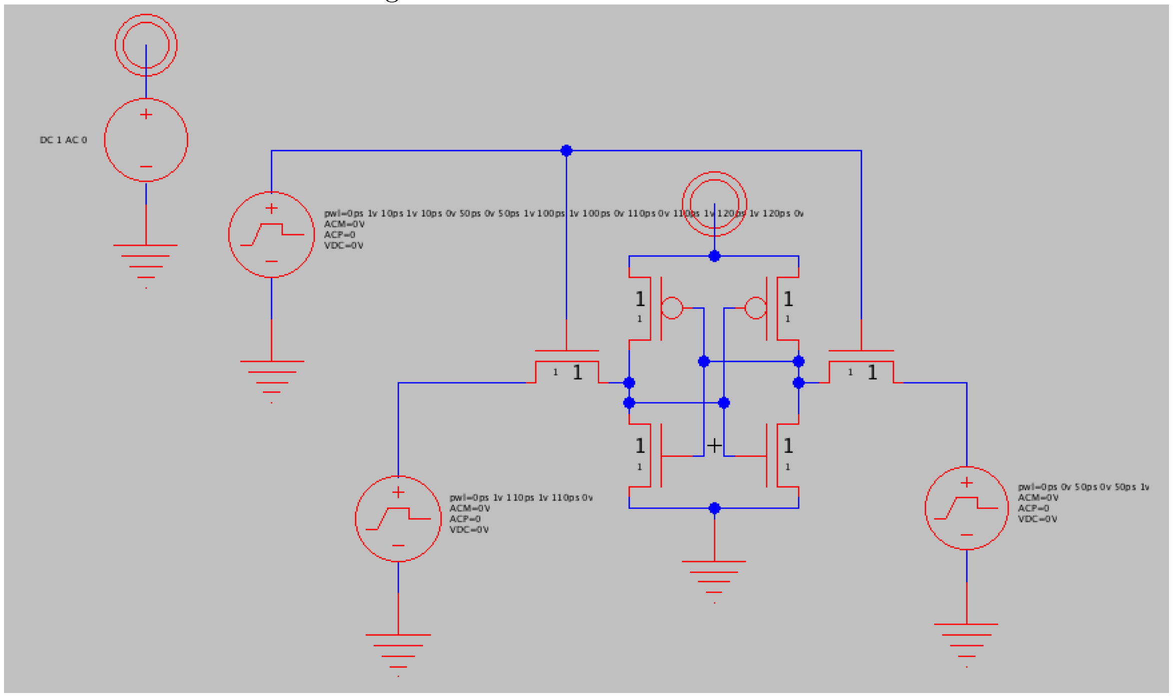

DC 1 AC 0
pwl=0ps 1v 10ps 1v 10ps 0v 50ps 0v 50ps 1v 100ps 1v 100ps 0v 110ps 0v 110ps 1v 120ps 1v 120ps 0v
ACM=0V
ACP=0
VDC=0V
pwl=0ps 1v 110ps 1v 110ps 0v
ACM=0V
ACP=0
VDC=0V
pwl=0ps 0v 50ps 0v 50ps 1v
ACM=0V
ACP=0
VDC=0V

Figure 3: Basic Cell with Tristate Configuration Schematic

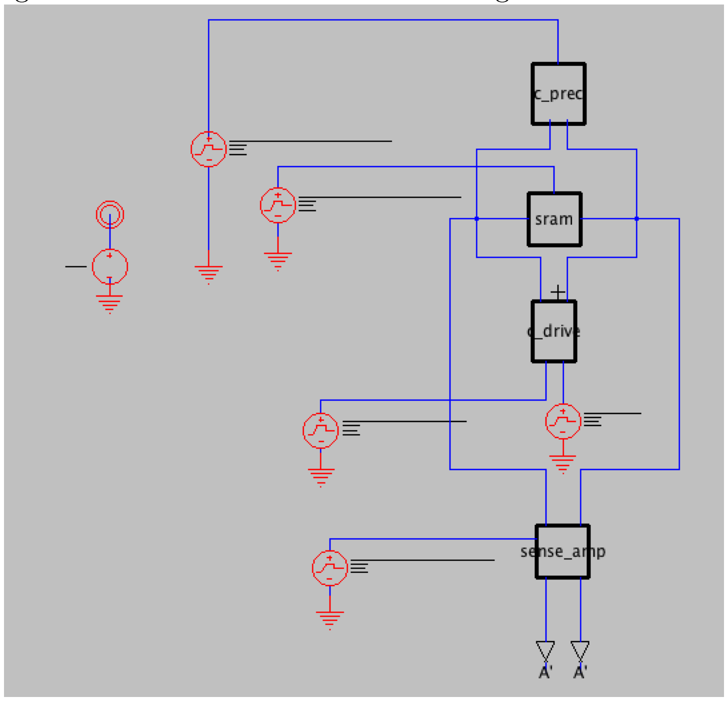


Figure 4: Column Driver Schematic

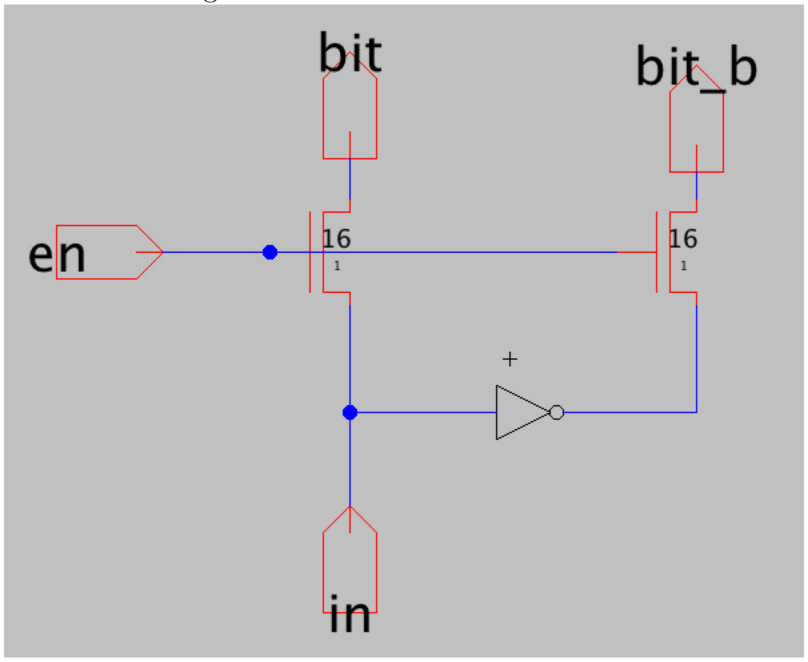


Figure 5: Column Precharger Schematic

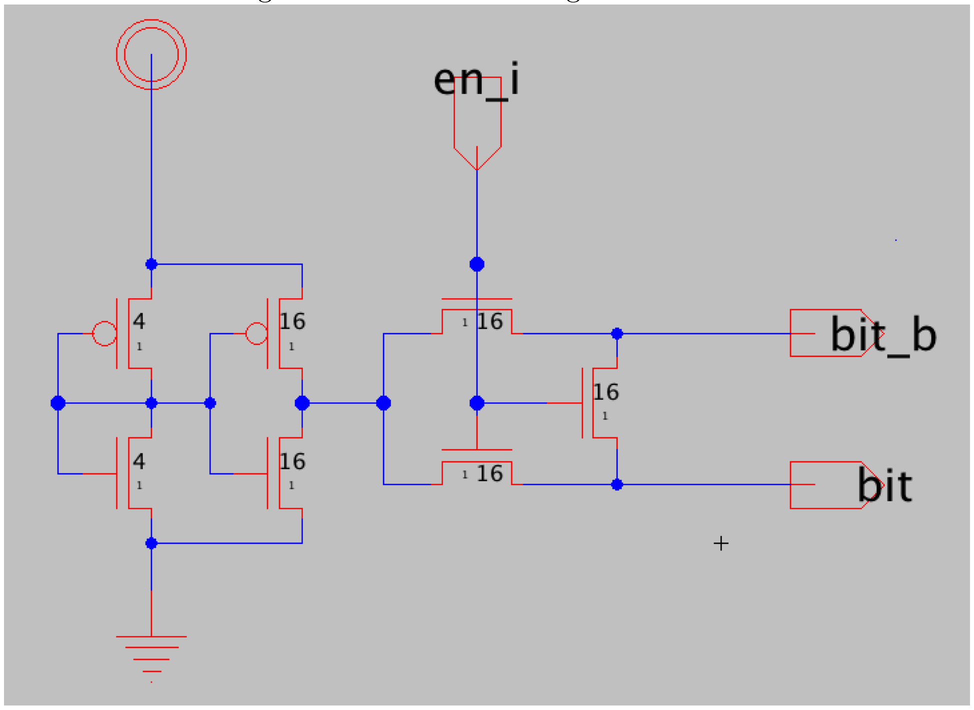


Figure 6: Sense Amp Schematic

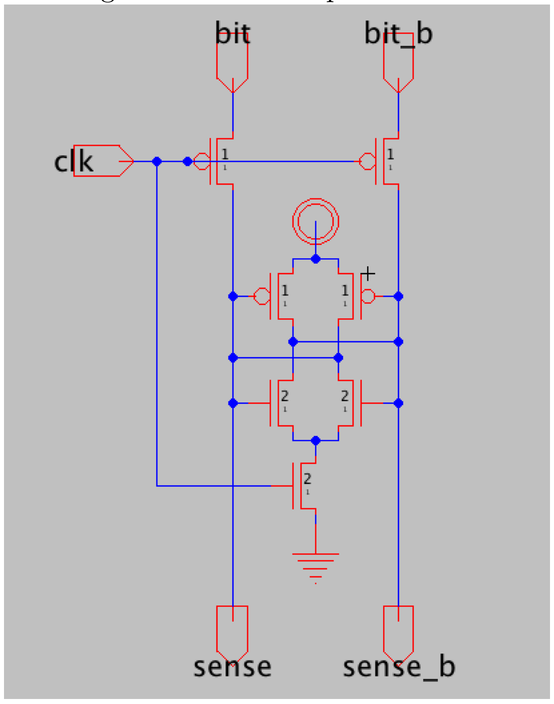
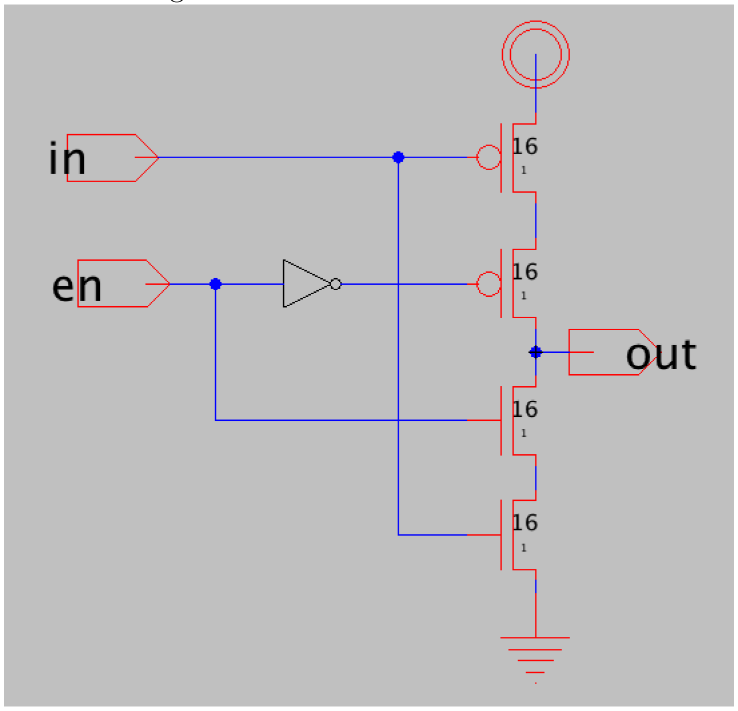


Figure 7: Tristate Buffer Schematic



3. **Test Cases to Validate Operation**
   To validate correct operation of our memory cell, we set up a few test cases to make sure that we can read and write. For writing, we had to make sure that we could write the following scenarios:

   (a) Write high to a cell containing a low value
   (b) Write high to a cell containing a high value
   (c) Write low to a cell containing a low value
   (d) Write low to a cell containing a high value

   These test cases cover the four possible scenarios of a writing to a memory cell, and are therefore sufficient to demonstrate correctness. To properly test our reading ability, we used these test cases:

   (a) Read from a high cell
   (b) Read from a low cell
   (c) Read from a low cell while writing high
   (d) Read from a low cell while writing low
   (e) Read from a high cell while writing high
   (f) Read from a high cell while writing low

   Similar to the cases listed for writing, these six cases encompass all possible read situations, and therefore can be used to show that our memory cell can properly read out values. However, for this milestone, we are not required to test simultaneous reads and writes. That being said, this is definitely an area we will be exploring with our full FIFO design for the final report.

   For testing reading and writing, we used Figure 3 as our test schematic, and we just changed values for the different VPWL sources to test different behavior scenarios.

4. **Demonstration of Writing to Cell**

Figure 8: Unamplified Bit Line Voltages

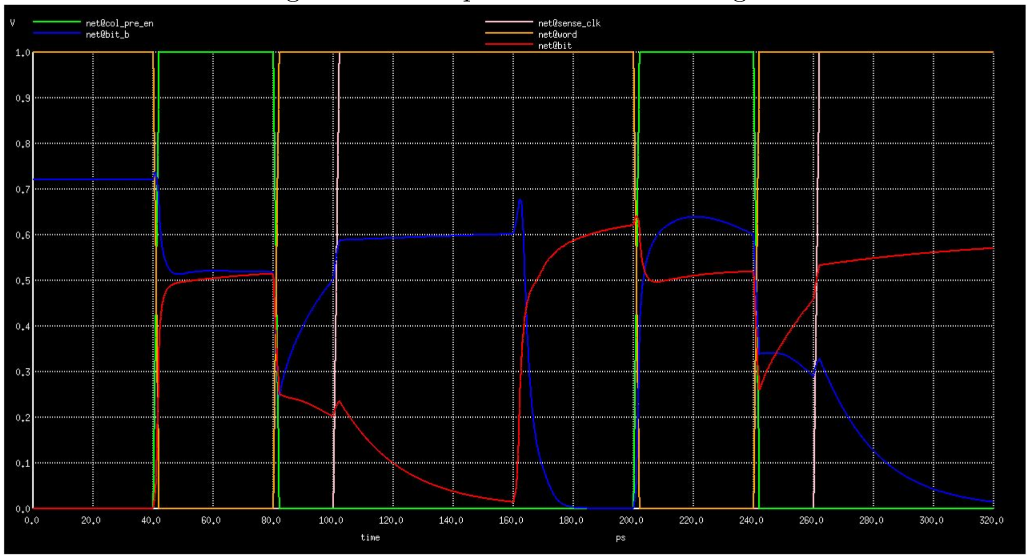
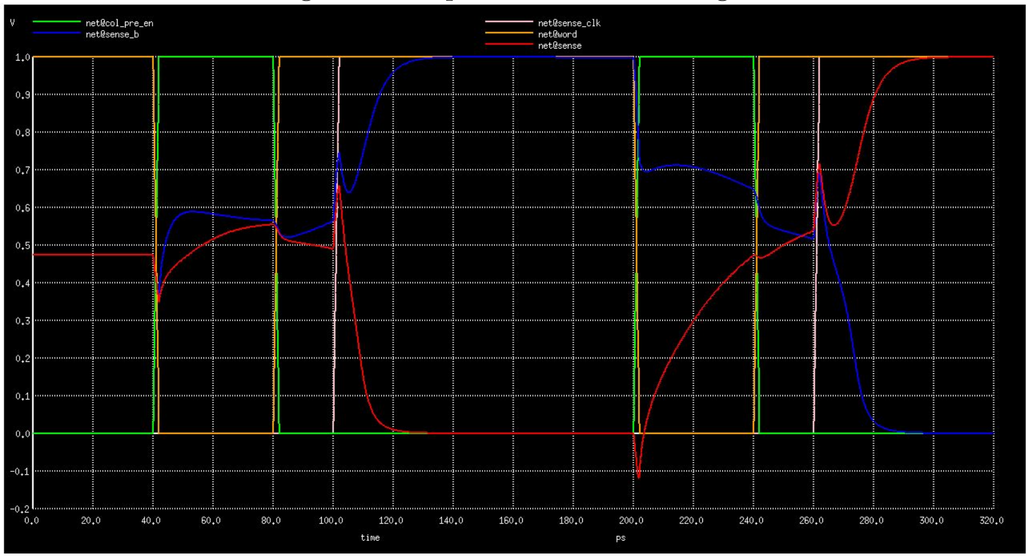


Figure 9: Amplified Bit Line Voltages



We see from these two graphs that our write function performs exactly as expected. Let us first take a look at our unamplified graph. We see around 80 ps that our word line goes high, and we are initially trying to write low to our memory cell. At the 80 ps mark, the word line switches, and we see that shortly after, our bit line goes to almost 0 V and our bit-b line goes to around 0.6 V. This shows that we can properly write low to the cell. Then again at 240 ps, we see the second instance of our word line going high, but this time we are trying to write high to our memory cell. After this instance, we see that our bit line goes to 0.6 V and our bit-b line goes to 0 V. Thus we can write both low and high to our memory cell. We were also able to reverse the order of writing high and low to satisfy all four of our test conditions, and those writes were also successful.

Now we can turn our attention to Figure 9. Here, we see the same behavior as in Figure 8, except we see both the bit line and the bit-b line being driven to rail. This demonstrates the functionality of our sense amplifier. It picked up that there was a minor change on the two lines and then drives the two lines to rail. Hence forth, we will only include the amplified versions of our graphs, as we have successively demonstrated that our sense amplifier is doing exactly what it should be.

5. **Demonstration of Reading from Cell**

Figure 10: Reading High from Memory Cell

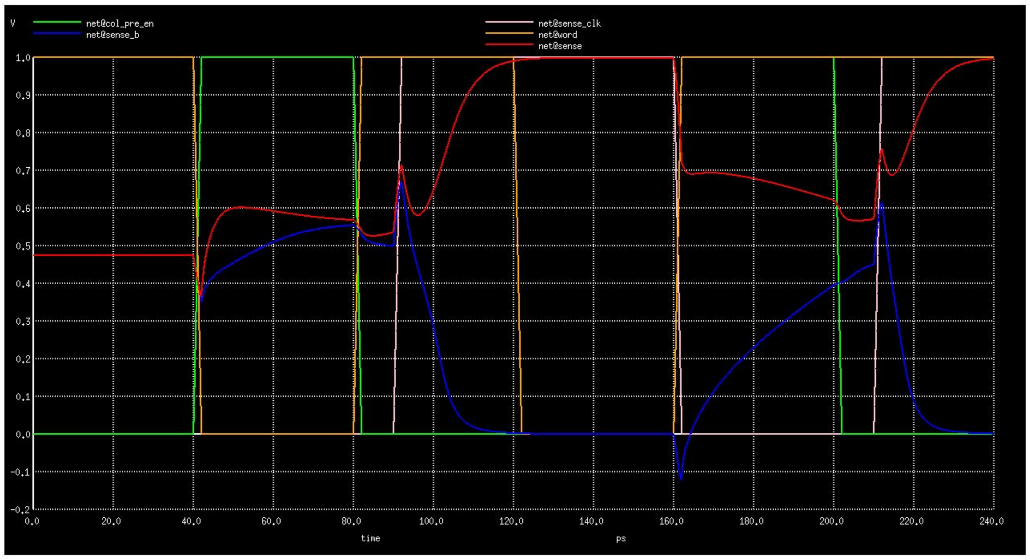
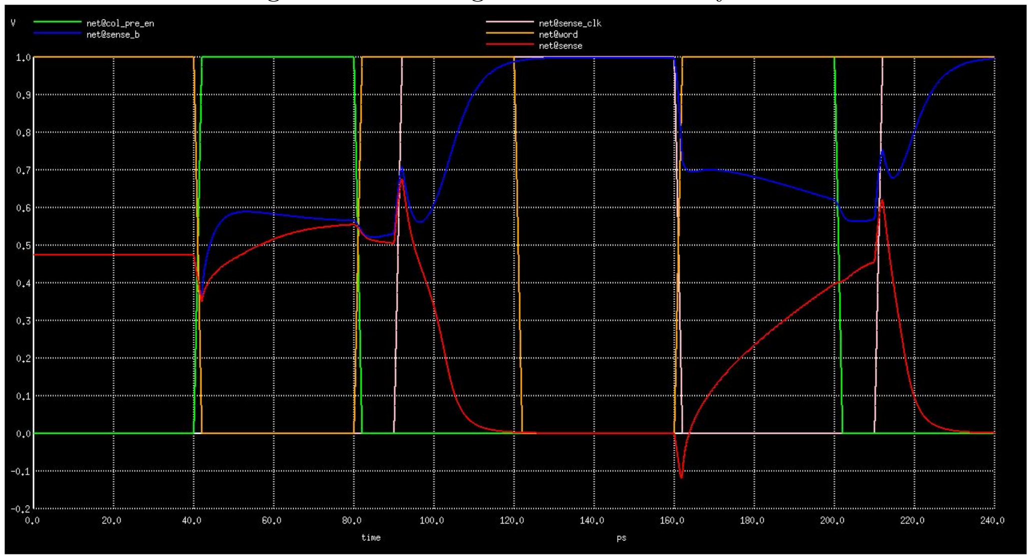


Figure 11: Reading Low from Memory Cell



These two test cases are very similar to our write cases. At around 80 ps the word line goes high, indicating that we are ready to perform a read operation. At around 90 ps, the sense clock goes high, which activates our sense amplifier. From 90 ps to the next rising edge of the word line (approximately 160 ps) and the sense clock (approximately 210 ps), we see that the value that we are reading is held strong at rail for both the bit line and the bit-b line. The only differences between the two simulations is whether the bit line goes high or low, which depends on whether we are reading a high value from the memory cell or a low value.

6. **Full Design Write Timing Constraints**
   There are a few constraints on write timing that we need to employ in order to ensure correct operation of our FIFO design. First, we are no longer dealing with a single memory cell in the full design. Rather, we are dealing with 4, so we need to make sure that our bit line and bit-b line are properly charged for each of the memory cells, and we also need to make sure that our word line stays high long enough for all of the memory cells to simultaneously ascertain their new values. Second, in order to read and write, we need to manipulate the word and bit lines, so we cannot read and write exactly at the same time. But we should be able to read and write within the same clock cycle. In order to make this functionality work, we need to slightly offset our read and write operations for the memory cells. Also, for this milestone, we were able to use VPWL sources to generate our signals, but in the case of the full FIFO design, we will be using actual control architecture, which has some inherent delay that needs to be taken into account. Along that same line of thinking, we are also storing our input into registers before writing it into the proper line in memory. Because of this, we need to allow enough time for the registers to properly hold the given input value (i.e. account for setup and hold time for the register), and only then can we proceed to read from the register and write out to the desired location in memory.

   In order to properly test the write ability for our full FIFO design, we need to formulate some new test cases. Some of the things we should be testing, aside from testing normal operation under "ideal" conditions, is making sure that our write timing waits the sufficient amount of time for the input registers to settle and hold the correct input value before being read from. We should also test writing when our write/enqueue signal comes extremely late, as well as testing writing if we get a new input but we get a signal to read/dequeue rather than write/enqueue, or no control signal at all. One of the other things we should be testing is to make sure that our control logic unit properly charges the individual bit and bit-b lines for each column of memory cells, and that the word line for each row of cells is held long enough for each cell to take the new value.