ECE/CSE 371 HW3

Elijah Melton, 2164822
Ankith Tunuguntla, 2334509

Problem 1

I ended up making the following changes:

- `fifo`: added upper signal and changed w_data to be twice the width of r_data
- `fifo_ctrl`: added upper signal and next state logic so that when reads are made, first upper is high and the upper half of the data is read, then upper becomes low and the lower half of the data is read. This logic also included incrementing the read address, and handling empty/full edge cases
- `reg_file`: added upper signal and an always_comb block to assign r_data to either the upper or lower half of the next element depending on the upper signal. also updated the write width to be double the read width

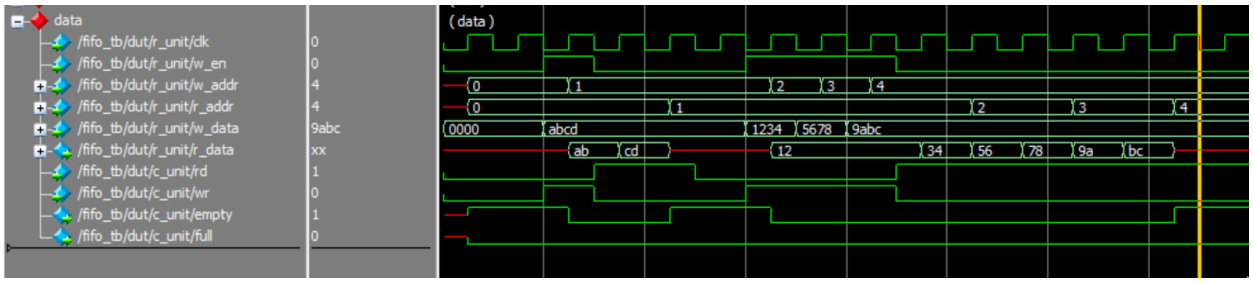
To verify our changes, we tested a few cases:

sequential write then read

- i.e. below you can see we write abcd, then read ab, then cd

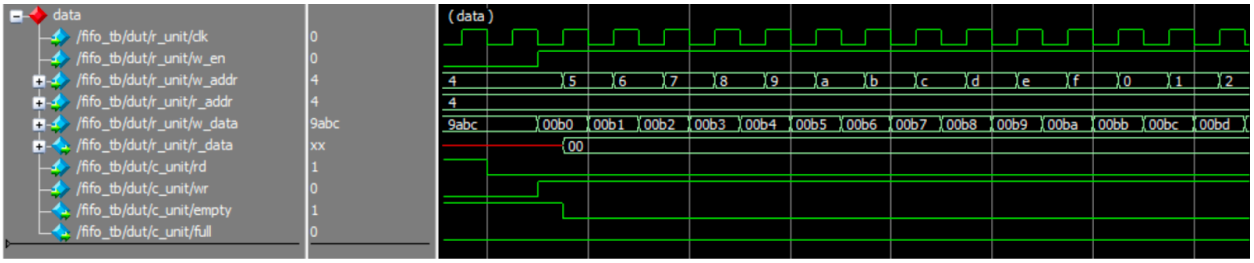
reading until empty

- i.e. we read until there are no more elements, and empty is asserted



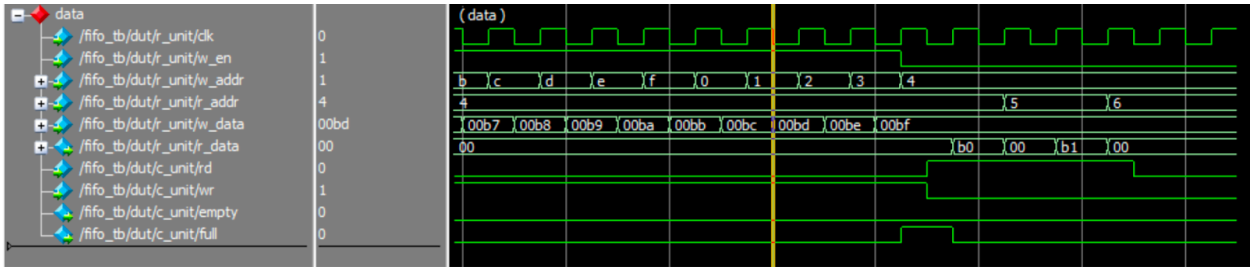
write until full

- i.e. below we write 16 elements, after which full is asserted



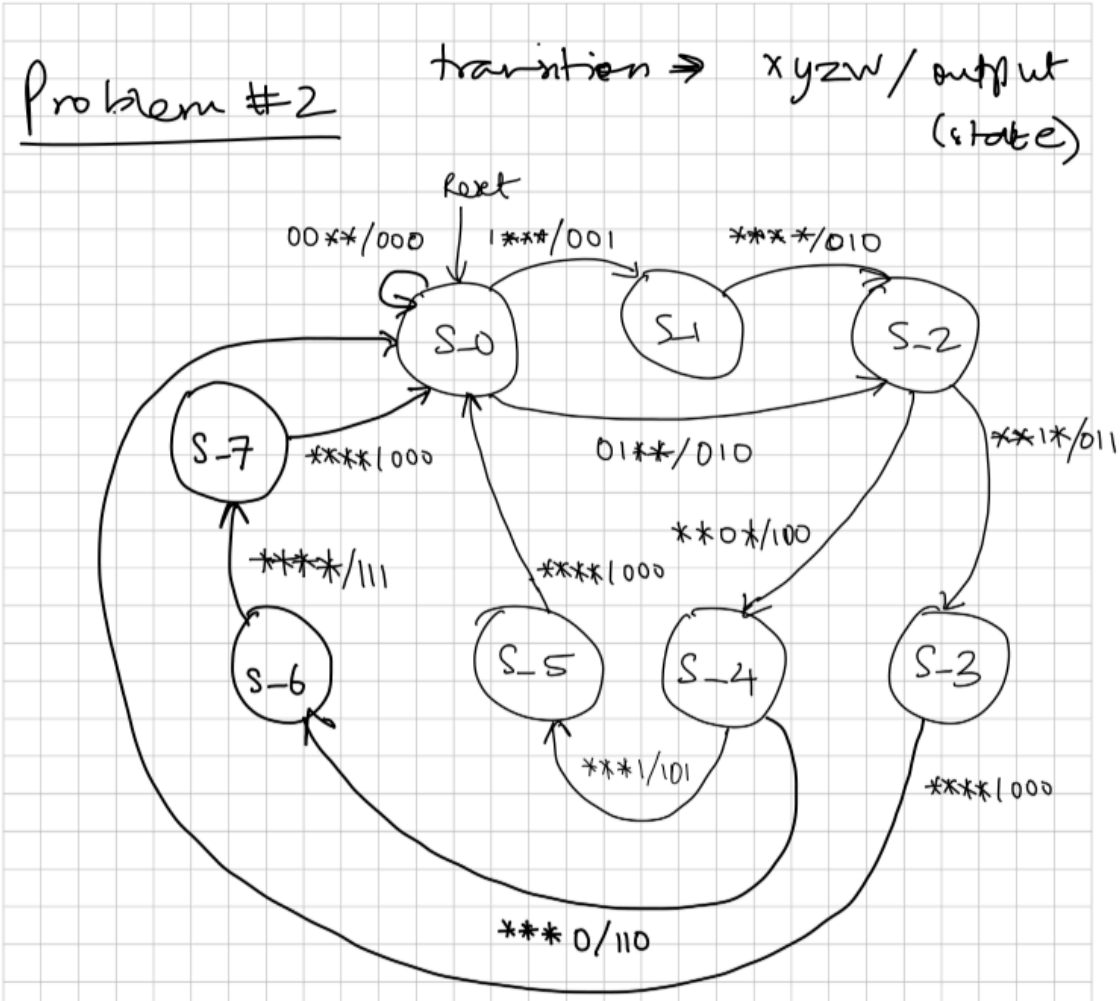
then we also try reading back these elements to make sure nothing was evicted

- i.e. the first thing we wrote was 00b0, and then the first thing we read is 00, followed by b0



Problem 2

State diagram for the ASM chart is:



Problem 3

Moore output: A new value can't appear until the next clock edge, so once it changes it stays steady for the whole cycle. The input seen currently only affects the output after one clock cycle. Glitches are not possible mid-cycle.

Mealy output: Output can change instantly as the input changes with no added delay. Basically the input seen currently can be seen within the same clock cycle. Glitches are possible mid-cycle.

In the ASM, the dashed lines around each component is one clock cycle with the rectangle blocks (Y_a, Y_b, Y_c) being the Moore inputs which stay fixed for the clock cycle. The diamonds are the Mealy inputs (X) which can change immediately. And finally, the arrows exit each block indicating the end of one clock cycle into the next component for a new clock cycle.

Tracing an ASM is much easier than that of a state machine as what happens in a clock cycle is more detailed in an ASM. An FSM is vague as you have to recall that outputs written in the bubble for Moore and those on arcs for Mealy while an ASM mimics how clock cycle in hardware works through a clear diagram of decision diamonds and state rectangles and dashed blocks.

Difficulty rating for each assignment:

Problem 1 - Moderate

Problem 2 - Easy

Problem 3 - Moderate

Time Spent

7 hours