

Connor Aksama
CSE 371
Hw 3
2/2/23

Problem 1

Changes

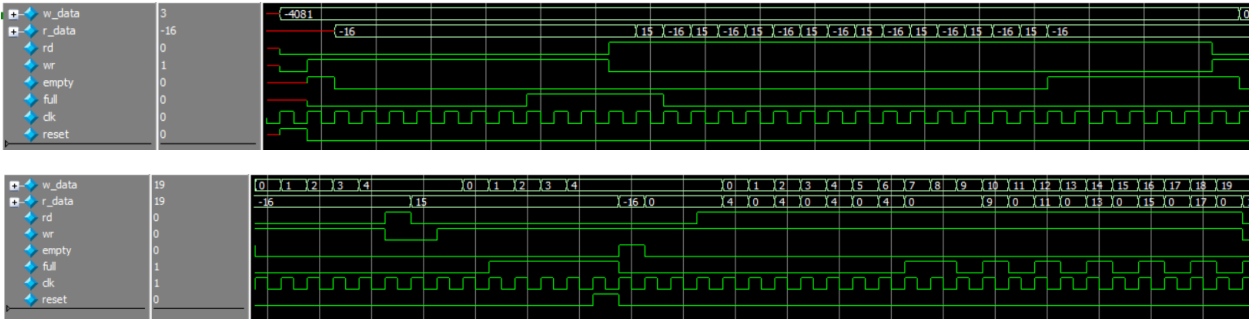
1. `fifo.sv` was changed to take 2 words of write data instead of 1.
2. `fifo_ctrl.sv` was changed to increase the write pointer by 2 locations on a write operation rather than just 1. Additionally, the logic to check if the buffer was full was changed to include the case when only 1 word was open.
3. `reg_file.sv` was changed to take in 2 words of write data, and to write to 2 adjacent words on a write operation.

Verification

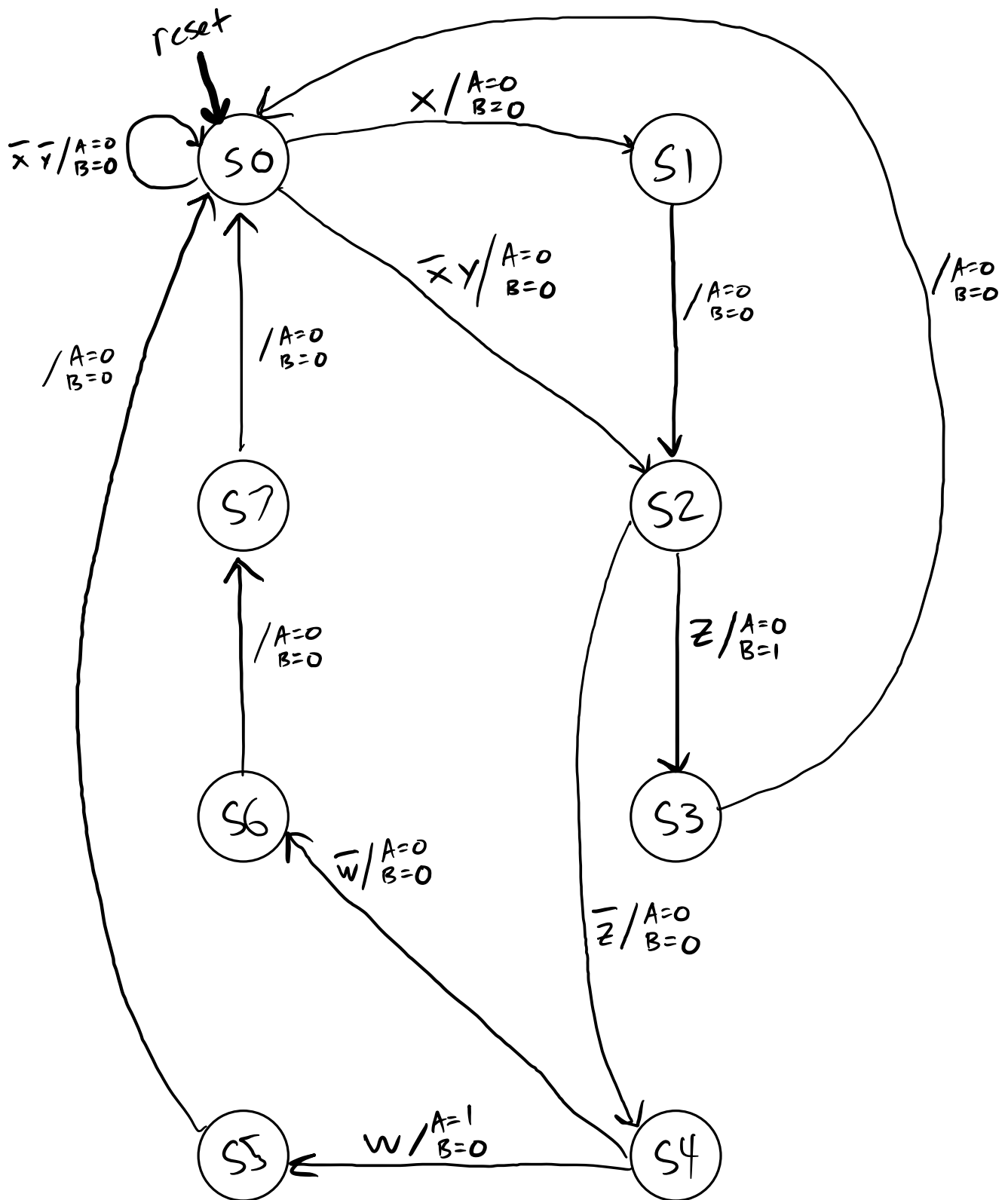
To verify this design, I sequentially pushed 16 bit values into the buffer until the buffer was full, then read 8 bit values until empty to ensure the correct ordering of values.

Next, a few words were written to the buffer, then one was read. Pairs of words were then written until the buffer became full at 15 words. Next, simultaneous read/write was tested. When full, we expect the read to succeed every clock cycle, but the write should only happen every other clock cycle.

Simulation Results



Problem 2



Problem 3

- Both Mealy and Moore actions are executed on the exit out of a state. Mealy actions depend on which transition is taken while Moore actions are not.
- ASM blocks help to indicate this timing by associating the action of an output with its physical location in the diagram (i.e. along a transition edge, or out of a state box).
- I think ASM charts are more convenient in tracing state transitions since the use of decision boxes make determining transitions easier. However, FSMs are easier to determine output values since output values are indicated at every state/ state transition.

Time Spent: 2 hours

Difficulty: Easy