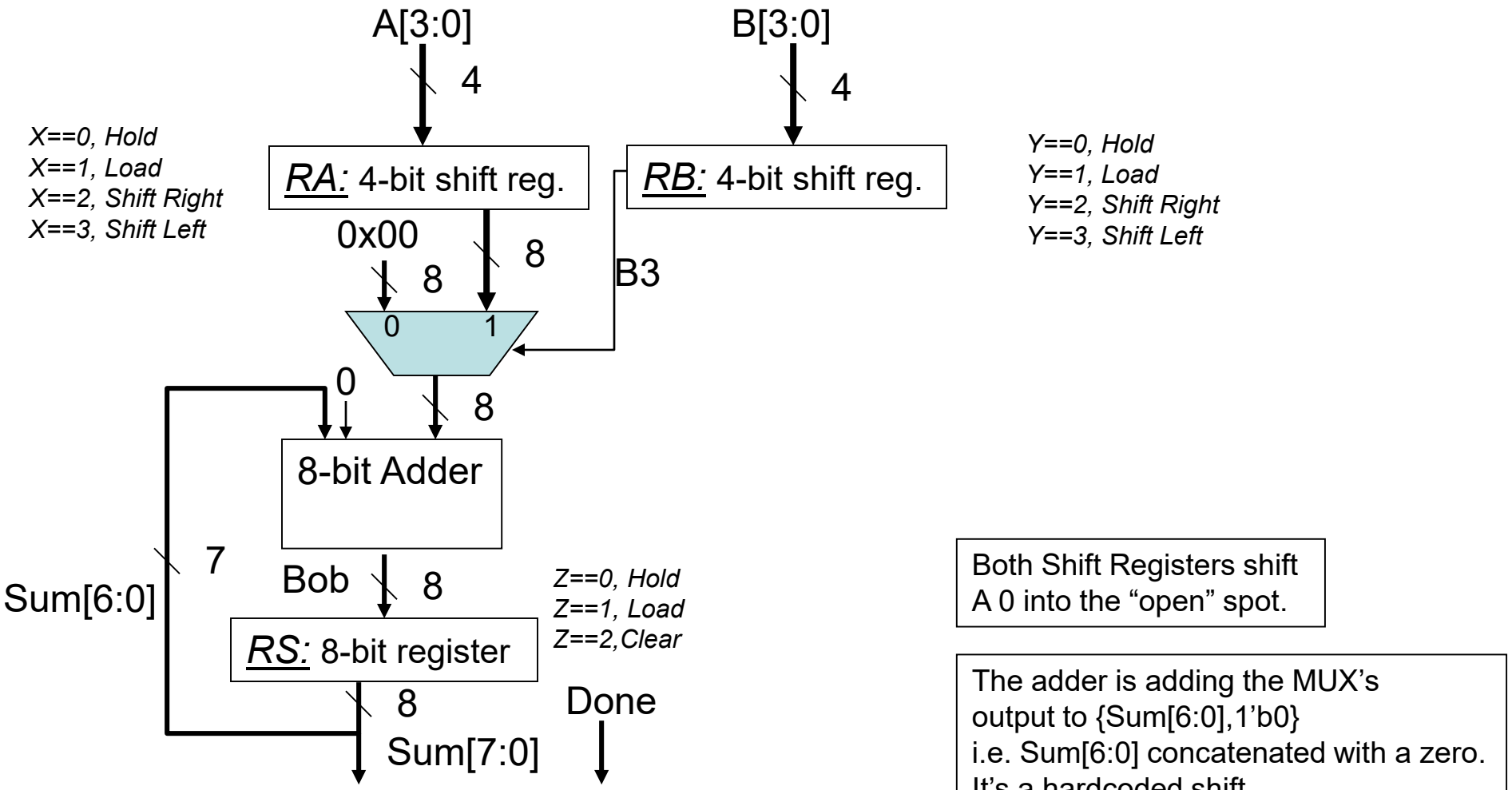
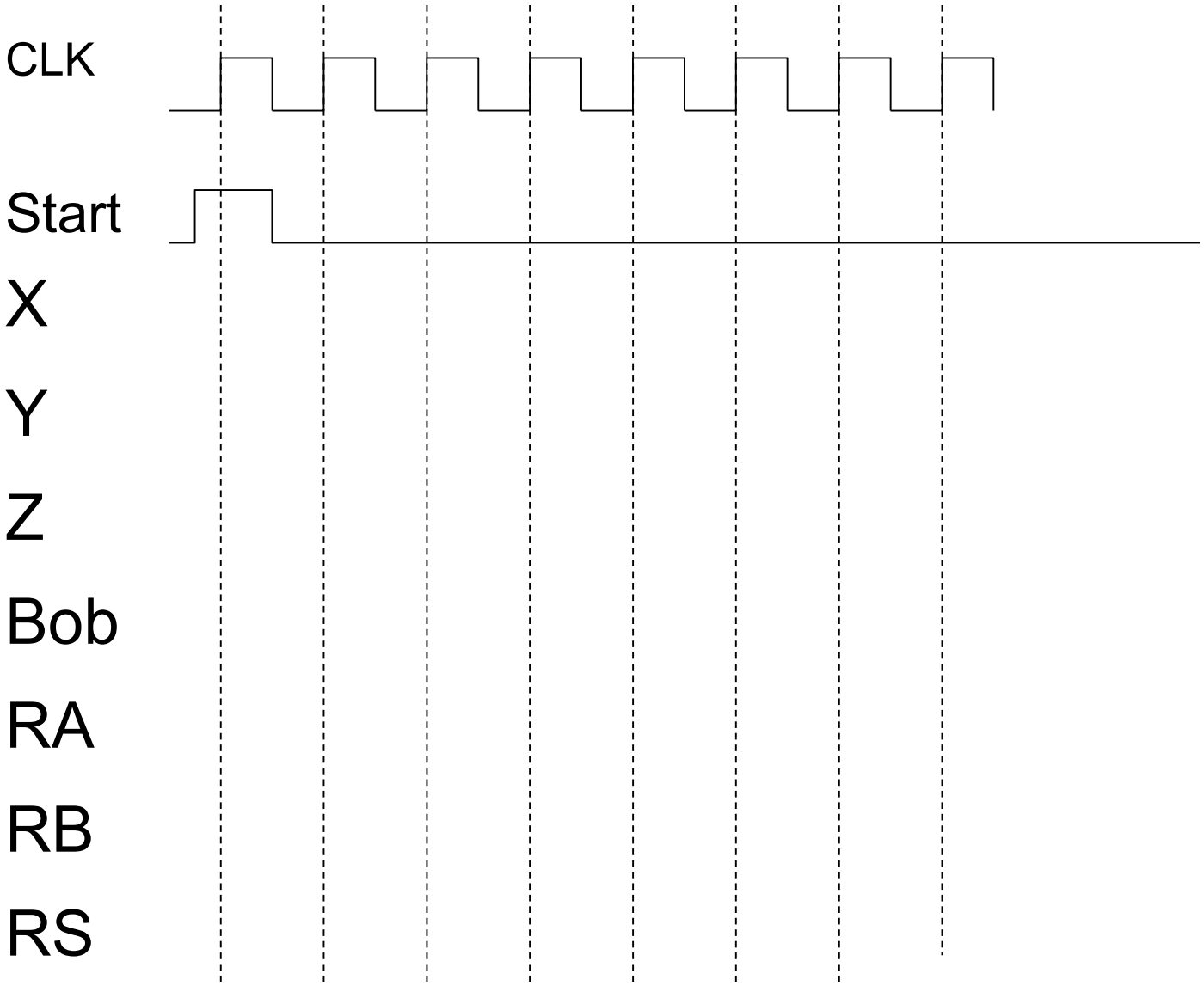


Both Shift Registers shift A 0 into the "open" spot.

The adder is adding the MUX's output to {Sum[6:0],1'b0} i.e. Sum[6:0] concatenated with a zero. It's a hardcoded shift.

You are to assume:

- Start will be asserted for one cycle and that A and B are available on the rising edge that start is asserted
- On any rising edge Done is asserted, Sum should be correct.
- Done need only be held high for 1 cycle.
- Start will not be reasserted to start a new operation until after you assert Done.


CLK
Start
X
Y
Z
Bob
RA
RB
RS