1. **Note that we used two always @ blocks in the file OoE_Divider.v. Can we combine these two blocks into one clock-edge-triggered block? If yes, explain how to implement. If no, explain why.**

We can combine them but we should not. The reason being is because the two always blocks serve two different purposes: the first one is used for sequential logic and the second for combinational logic.

If we merged the two then the “start” and “ack” signals would persist across cycles, causing issues with the “single_divider” modules which need “start” and “ack” to behave like 1-clock pulses.

2. **Note that we instantiated 4 single dividers in the OoE divider. Let’s take the signals Ack (ack_from_OoE_div) and Start (start_from_OoE_div) of single dividers as examples. Explain how we ensure that these two signals are active for exactly one clock when they should be (they are both set in an always @ block implementing a combinational logic in OoE_Divider.v).**

start_from_OoE_div and ack_from_OoO_div are inside a combinational block. In each clock cycle they are both reset to 0 at the start of the block. Since both these signals are not registered and are re-evaluated every clock cycle. This guarantees that they single-cycle pulses.

3. **In this lab, we used 4-bit ROB read/write pointers (RP and WP) for the 8-entry ROB, yet we only used the lower 3-bits to index the ROB. Do you think ignoring the**

MSB would cause any error in some case, explain why. Where did we use the MSB of WP and RP?

Although the indexing is using the lower 3 bits, we use the 4 bit pointers to track wrap-around conditions and distinguish between full and empty ROB.

If we ignore the MSB we wouldn't be able to distinguish between an empty or full ROB, because in a circular buffer when rob_wp == rob_rp and MSB match then ROB is empty and when rob_wp == rob_rp and MSB don't match then ROB is full. The MSB is implicitly used in the calculation of "rob_depth = rob_wp - rob_rp" and in the condition for detecting rob_full, ensuring proper wraparound tracking.