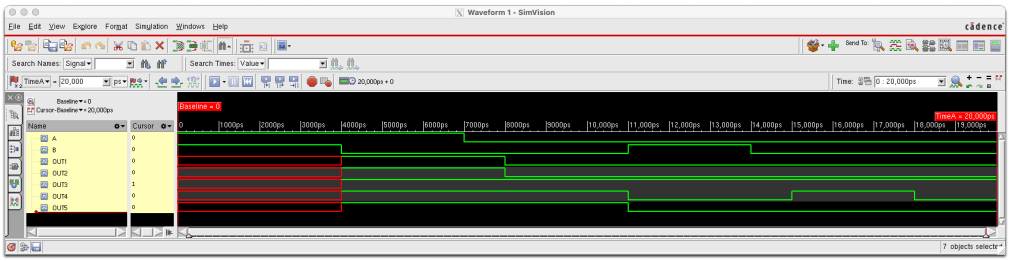
EE577B HW2 – Part B Explanation

Explanation of Delay Models

The five OR gate models exhibit different behaviors due to the type of delay and assignment used in Verilog. The simulation results can be understood as follows:

- **At time 0 ps:** All outputs are unknown (x) because no assignment has occurred yet. The processes only schedule updates after the first input change. The continuous assignment with delay also does not resolve immediately.
- **At time 4000 ps:** Inputs changed at 0 ps and all models update after 4 ns. Since $(A|B) = 1$, all outputs are equal to 1.
- **At time 7000 ps:** Inputs changed at 7 ns but the 4 ns delay has not yet expired. Therefore, all outputs remain at 1.
- **At time 11000 ps:** The effect of the 7 ns input change is visible. The inter-statement inertial models (`or_block_inter`, `or_nba_inter`) still show 1 because the short 0 pulse was filtered out. The intra-assignment transport models (`or_block_intra`, `or_nba_intra`) update to 0 since they propagate every input event after the programmed delay. The continuous assignment with delay (`or_continuous`) also updates to 0, consistent with inertial delay behavior.
- **At time 14000 ps:** Inputs change again but the outputs have not yet been updated because the 4 ns delay has not elapsed. The values remain the same as at 11000 ps.
- **At time 15000 ps:** The non-blocking intra-assignment model (`or_nba_intra`) shows a temporary 1. This is because the input value of 1 at 11 ns was scheduled to appear at 15 ns, reflecting the transport nature of intra-assignment delays. The other models do not show this change.
- **At time 18000 ps:** The effect of the input change at 14 ns is visible. All models update to 0, with the inertial models showing the stable 0 after the full delay and the transport models having also settled to 0.

Waveforms



Summary

Inertial delays (inter-statement assignments and continuous assignments) filter out short pulses, while transport delays (intra-assignment assignments) propagate every event after the delay. Blocking versus non-blocking determines when in the simulation cycle the updates are applied, which explains the subtle differences in timing between the models.