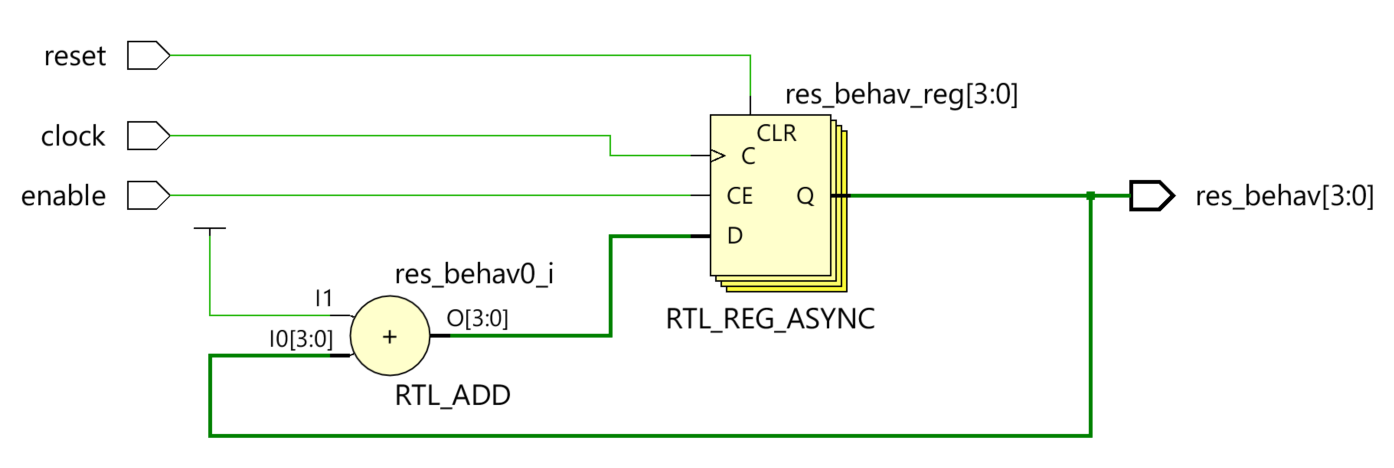
*Comparative Analysis of Synthesis Results: RTL vs. Behavioral Module*

Huzaifah Tariq Ahmed – ha07151

Behavioral Four Bit Counter:

The provided code implements a counter's functionality in a single block, resembling how higher-level languages like C would handle it. While this approach is concise, it doesn't explicitly differentiate between the counter's combinational and sequential logic, which are fundamental concepts in digital design.

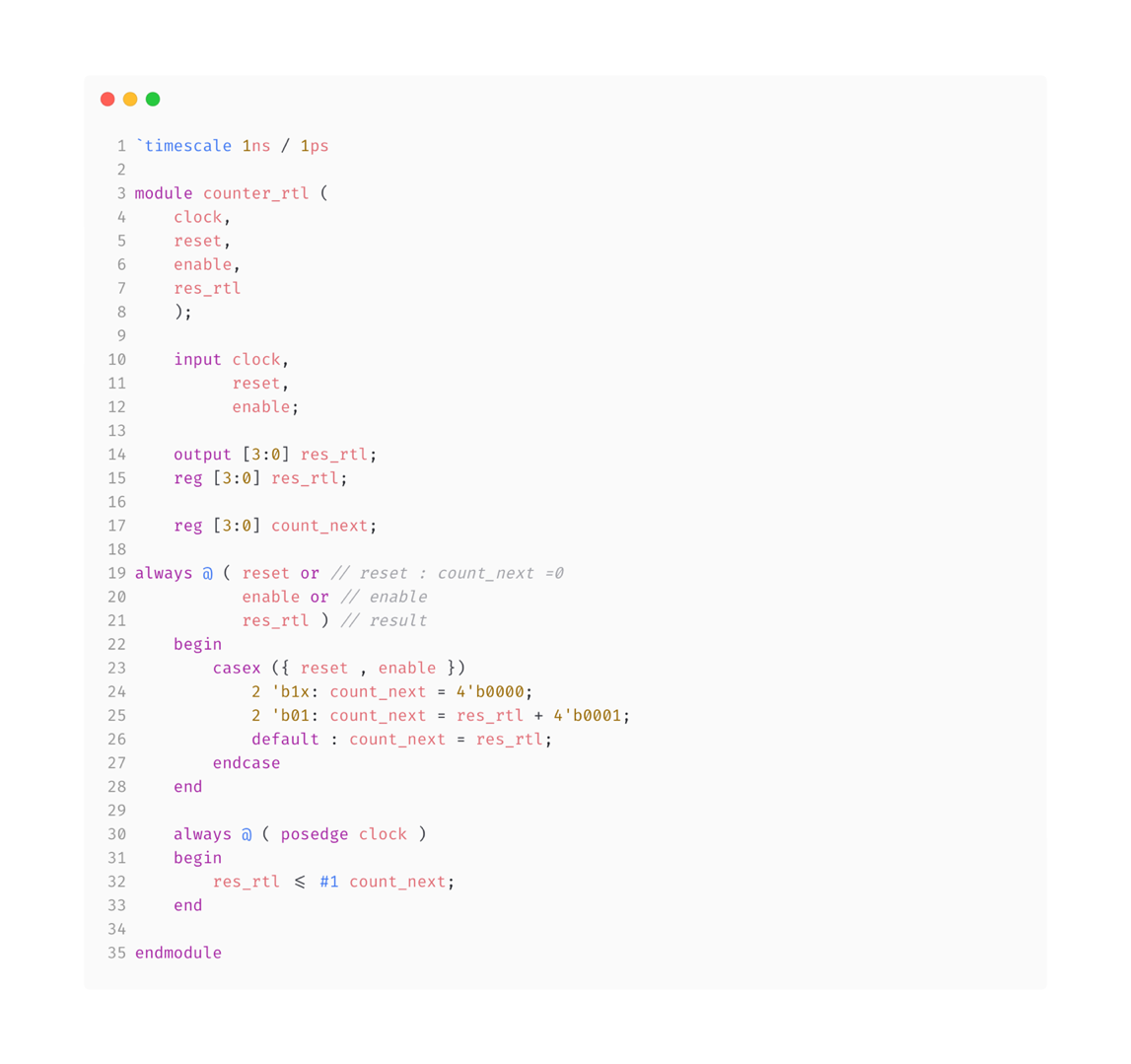




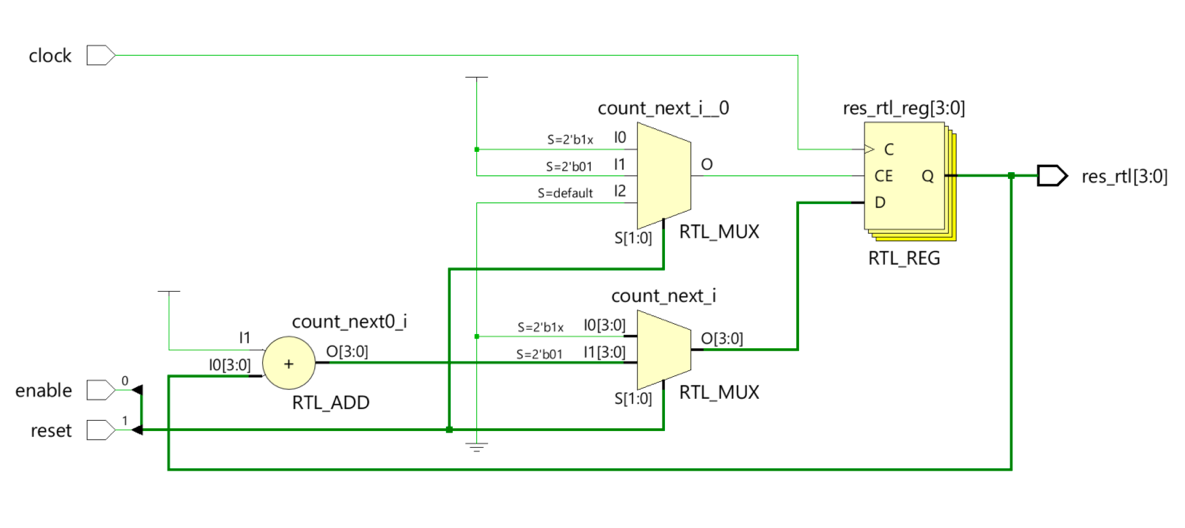
The schematic generated from the provided behavioral code indicates that there is no distinct separation between the combinational and sequential blocks. This is evident as all signals are directly routed to the result register.

RTL Four Bit Counter:

The provided code features an RTL Module with distinct combinational and sequential blocks. The initial always block, responsible for combinational logic, generates an output determined by the reset and enable signals. In the subsequent always block, which operates on the positive edge of the clock signal, only the counter's result is updated.

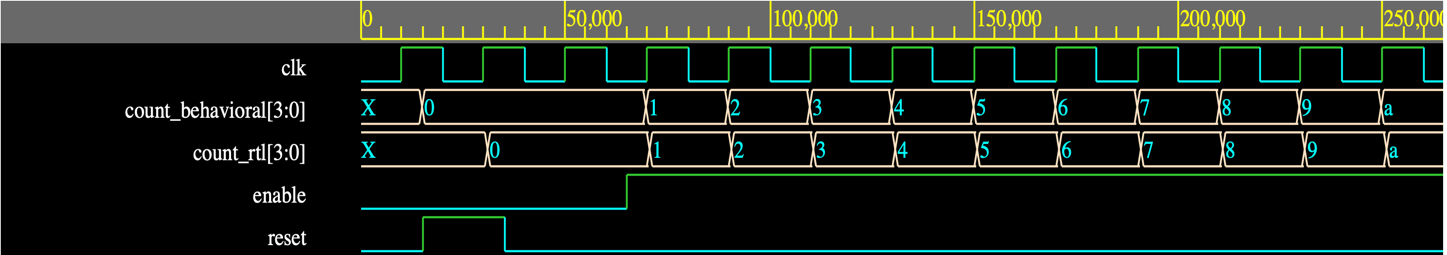


The schematic generated below from the provided RTL code demonstrates a clear distinction between the combinational and sequential blocks. In this design, reset or enable signals are not directly applied to the flip-flop; instead, they are directed to a multiplexer. The multiplexer determines the output based on the incoming signals, effectively controlling the behavior of the flip-flop.



Simulation Comparing RTL and Behavioral Models:

In the provided waveform, the initial states of both 'count rtl' and 'count behav' include don't care values, which are later set to zero when the reset signal is asserted. Notably, 'count behav' promptly transitions to zero as soon as the reset signal changes, whereas 'count rtl' defers its initialization to zero until the positive edge of the clock. Subsequently, both modules exhibit similar behaviors, commencing incrementation upon the activation of the enable signal. However, due to the presence of '#1' in line 32 of the RTL code, 'count rtl' experiences a slight delay compared to 'count behav'.



Top Module:



The schematic depicted illustrates the schematic produced by the top-level module for comparison. In the RTL module, there exist two multiplexers, whose outputs are directed to the flip-flop. Conversely, the behavioral schematic lacks such a multiplexer. The output from “count\_next\_i\_ 0” multiplexer, is directed to the Clock Enable (CE) input of the flip-flop, indicating that data modifications will only occur when the signal is high. Therefore, a high signal is transmitted when either the reset or enable signals are high, signifying the intended moments for data modification.

