Lab 4

Student Name: Dillon Vu
Student NSID: 11100292

Functional coverage

Coverage group:

- Input_data_coverage
  - Cover point: coverage_point_op_code: 1 bin for each op code
  - Cover point: Coverage_point_alu_a : 6 bins from each different range value of input a
  - Cover point: Coverage_point_alu_b : 6 bins from each different range value of input b
- Cg_min_max_value
  - Cover point: coverage_point_alu_a_min_max: 2 bins with value of 0 and 255 for a
  - Cover point: coverage_point_alu_b_min_max: 2 bins with value of 0 and 255 for b
- Transition_cg
  - Cover point: coverage_transition: 256 bins for opcode transition for each range
- Cg_reset
  - Cover point: cover 2 bins reset value 1 and 0
- Cg_alu_co_out
  - Cover point: cover 2 bins alu_co_out value 1 and 0
- Input_alu_in_toggle
  - Cover point bit_val: cover 2 bins value 0 and 1 of each bit
  - Cover point: bit_number: cover 8 bins value from 0 ->7 of each b

To cover the reset coverage group, reset bit was imported into testbench_top to generate reset. Virtual interface with reset object then would pass into coverage class then pass into coverage group.

Initially, when importing the report, a couple of errors appeared to the scoreboard due to reset value change which changed different output. Therefore, import reset==0 to each opcode case would reduce the errors.

There are 13 errors that appear after cover 100% coverage.

Test case:

100% functional coverage has been achieved however there are 13 errors appearing from the scoreboard.