Label	Design Requirement Description	Stimulus Generation	Functionality Check	Functional coverage
Reset_feature	we assert active low Rst_t to get ideal state	we directed at start of simulation then we randomize under constraint that is should be deasserted most of simulation	check with immediate final Assertion for asyn reset function	we cover the values output with right values when reset asserted
Full_feature	we check when Fifo is full if output flag is high or not	we randomize on simulation	we check with immediate final Assertion because its combintional it should be high when count =depth	we cover all values of full with wr_en and read _en with illegal bins check on logic
empty_feature	we check when Fifo is empty if output flag is high or not	we randomize on simulation	we check with immediate final Assertion because its combintional it should be high when size is zero and reset	we cover all values of full with wr_en and read _en with illegal bins check on logic
almostFull_feature	we check when Fifo is full if output flag is high or not	we randomize on simulation	we check with immediate final Assertion because its combintional it should be high when count = depth -1	we cover all values of full with wr_en and read _en
almostEmpty_feature	we check when Fifo is full if output flag is high or not	we randomize on simulation	we check with immediate final Assertion because its combintional it should be high when count 1	we cover all values of full with wr_en and read _en
overflow_feature	we check when Fifo is full if output flag is high or not	we randomize on simulation	check this property with concurrent assertions that check if flag is high when count =depth and write enaple is high	we cover all values of full with wr_en and read _en with illegal bins check on logic
underflow_feature	we check when Fifo is full if output flag is high or not	we randomize on simulation	check this property with concurrent assertions that check if flag is high when count =depth and write enaple is high	we cover all values of full with wr_en and read _en with illegal bins check on logic
wrAck_feature	we check when Fifo is full if output flag is high or not	we randomize on simulation	check this property with concurrent assertions that check if flag is high when count < depth and write enaple is high so it can write successfully	we cover all values of full with wr_en and read _en with illegal bins check on logic
write and read property	we check if FIFO write and read successfully	we randomize on simulation	we check with queue in golden model that we can pop and push on	-

internal_counter	we check on internal counter that it work right on write and read	we randomize on simulation	check this property with concurrent assertions that check counter increase if write is activated and decrement when read is activated and no change if write and read activated or deactivated At the same time	-
Internal_write_read ptr	we check on write and read pointer if working right	we randomize on simulation	check this property with concurrent assertions that check write ptr increase if write is activated and read ptr increment when read is activated	-