Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_1	When the reset_n is asserted, the output count_out value should be low, max_count should be low and zero should be high.	Directed at the start of the simulation, then randomized with constraint that drive the reset to be off(high) most of the simulation time.	-	Immediate assertion to check for the async reset functionality.
FIFO_2	When the rd_en is asserted and wr_en is de-asserted, the output data_out should take the value of the stored current value.	Randomization under constraints on rd_en being on for RD_EN_ON_DISTANCE%.	-	Checked by the reference model. Concurrent assertion to check for the counter decrement.
FIFO_3	When the wr_en is asserted and rd_en is de-asserted, the input data_in should be stored in the fifo, and the output wr_ack should get asserted.	Cyclic randomization on the data_in. Randomization under constraints on wr_en being on for WR_EN_ON_DISTANCE% of the time.	Cover all values of data_in. Cover the crossing of the wr_ack signal with rd_en and wr_en.	Concurrent assertion to check for the wr_ack signal, can also be checked by the reference model. Concurrent assertion to check for the counter increment.
FIFO_4	When the wr_en and rd_en are asserted without empty or full being asserted, the input data_in should be stored in the fifo, and the output wr_ack should get asserted and the current stored value to be read will be output on data_out.	Randomization under constraints on the ce signal to be on(high) 70% of the time.	-	Checked by the reference model. Concurrent assertion to check for the wr_ack signal, can also be checked by the reference model. Concurrent assertion to check for the counter remaining unchanged.
FIFO_5	When the wr_en and rd_en are asserted with empty being asserted, the input data_in should be stored in the fifo, and the output wr_ack should get asserted.	Randomization	Cover the crossing of the wr_ack signal with rd_en and wr_en.	Concurrent assertion to check for the wr_ack signal, can also be checked by the reference model. Concurrent assertion to check for the counter increment.
FIFO_6	When the wr_en and rd_en are asserted with full being asserted, the current stored value to be read will be output on data_out.	Randomization	-	Checked by the reference model. Concurrent assertion to check for the counter decrement.
FIFO_7	When the fifo is full, output flag full must be asserted.	Randomization	Cover the crossing of the full signal with rd_en and wr_en.	Concurrent assertion to check for the full signal, can also be checked by the reference model.
FIFO_8	When the fifo is full and wr_en gets asserted, output flag overflow must be asserted.	Randomization	Cover the crossing of the overflow signal with rd_en and wr_en.	Concurrent assertion to check for the overflow signal, can also be checked by the reference model.
FIFO_9	When the fifo is empty, output flag empty must be asserted.	Randomization	Cover the crossing of the empty signal with rd_en and wr_en.	Concurrent assertion to check for the empty signal, can also be checked by the reference model.
FIFO_10	When the fifo is empty and wr_en gets asserted, output flag underflow must be asserted.	Randomization	Cover the crossing of the underflow signal with rd_en and wr_en.	Concurrent assertion to check for the underflow signal, can also be checked by the reference model.
FIFO_11	When the fifo has only 1 data value stored, output flag almostempty must be asserted.	Randomization	Cover the crossing of the almostempty signal with rd_en and wr_en.	Concurrent assertion to check for the almostempty signal, can also be checked by the reference model.
FIFO_12	When the fifo has 1 remaining spot left to store, output flag almostfull must be asserted.	Randomization	Cover the crossing of the almostfull signal with rd_en and wr_en.	Concurrent assertion to check for the almostfull signal, can also be checked by the reference model.