Top-level Module - async_fifo1_tb_uvm:
Responsibilities:

Acts as the top-level module for the testbench.
Defines parameters for the asynchronous FIFO (DSIZE, ASIZE, IDLE_R, IDLE_W).
Generates write (wclk) and read (rclk) clocks.
Instantiates the async_fifo1 module (dut) and the test class (test_inst).

Environment - environment Class:
Components:
generator: Responsible for generating random transactions.
driver: Drives transactions into the virtual interface based on data from the generator.
monitor_in: Monitors the write side of the virtual interface.
monitor_out: Monitors the read side of the virtual interface.
scoreboard: Compares data received from monitors for verification.

Communication:
Utilizes mailboxes (gen2driv, mon_in2scb, mon_out2scb) for communication between components.
Initializes and controls the testbench flow using pre-test, test, and post-test tasks.

Tasks:
pre_test: Initiates the reset task in the driver.
test: Orchestrates the concurrent execution of generator, driver, monitors, and scoreboard.
post_test: Waits for the generator to complete, then verifies transaction counts.

Generator - generator Class:
Responsibilities:
Generates random transactions (tx) with random write data (wdata) and write increments (winc).
Uses a mailbox (gen2driv) to send generated transactions to the driver.
Models the asynchronous behavior of write and read clocks.
Implements the main task that generates transactions, verifies read data against the expected write data, and signals completion.

Driver - driver Class:
Responsibilities:
Drives transactions into the virtual interface (vif) based on data received from the generator.
Monitors the write clock (wclk), writes data (wdata), and increments (winc).
Uses a mailbox (gen2driv) to receive transactions from the generator.
Implements the reset and main tasks for initialization and driving transactions, respectively.

Monitors - monitor_in and monitor_out Classes:

Responsibilities:

monitor_in captures data from the write side of the virtual interface (vif).
monitor_out captures data from the read side of the virtual interface (vif).
Utilizes mailboxes (mon_in2scb and mon_out2scb, respectively) to send captured transactions to the scoreboard.
Implements the main task to monitor and display data.

Scoreboard - scoreboard Class:
Responsibilities:

Compares data received from monitor_in and monitor_out with the expected data.
Implements tasks (get_data_w, get_data_r, get_output) to gather and compare data.
Displays assertion checks, including expected and received data.

Async FIFO - async_fifo1 Module:
Responsibilities:

Represents the asynchronous FIFO being tested.
Takes the virtual interface (bus_tb) as an input.
Generates asynchronous behavior based on write (wclk) and read (rclk) clocks.

Transaction - transaction Class:
Responsibilities:

Represents a transaction with detailed information about write and read operations.
Includes a print function to display transaction information.

Top-level Module - test:
Responsibilities:

Initiates the environment and runs the testbench.
Displays a summary of the total number of transactions and finishes the simulation.