Verification Plan for the MOD 10 Loadable UP/DOWN Counter

MOD 10 Loadable up/down counter:

```
  Clock->        ┌─────────────────┐
                 │                 │
  Reset->        │   Modulo 10     │
                 │   UP/DOWN       │    ->Data_out [3:0]
  Load->         │   counter       │
                 │                 │
Mode [UP/DOWN]-> │                 │
                 │                 │
Data_in [3:0]->  └─────────────────┘
```
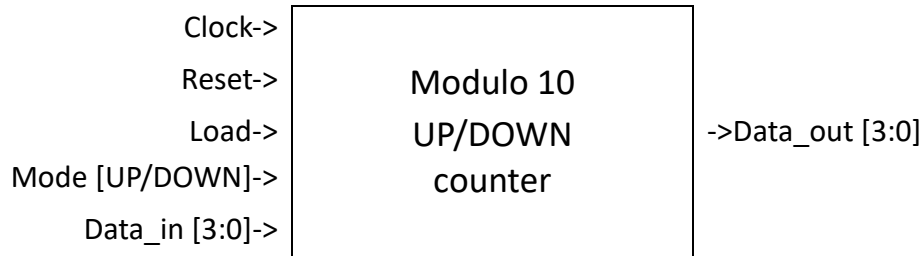
An active high loadable up/down counter is a digital circuit that can be programmed to count either up or down based on user input. The counter can be loaded with an initial value, which sets the starting point for counting.

When counting up, the counter increments by one on each clock cycle, and when counting down, the counter decrements by one on each clock cycle. The direction of counting is determined by a control signal, which can be set to either "up" or "down" mode.

TB Environment:
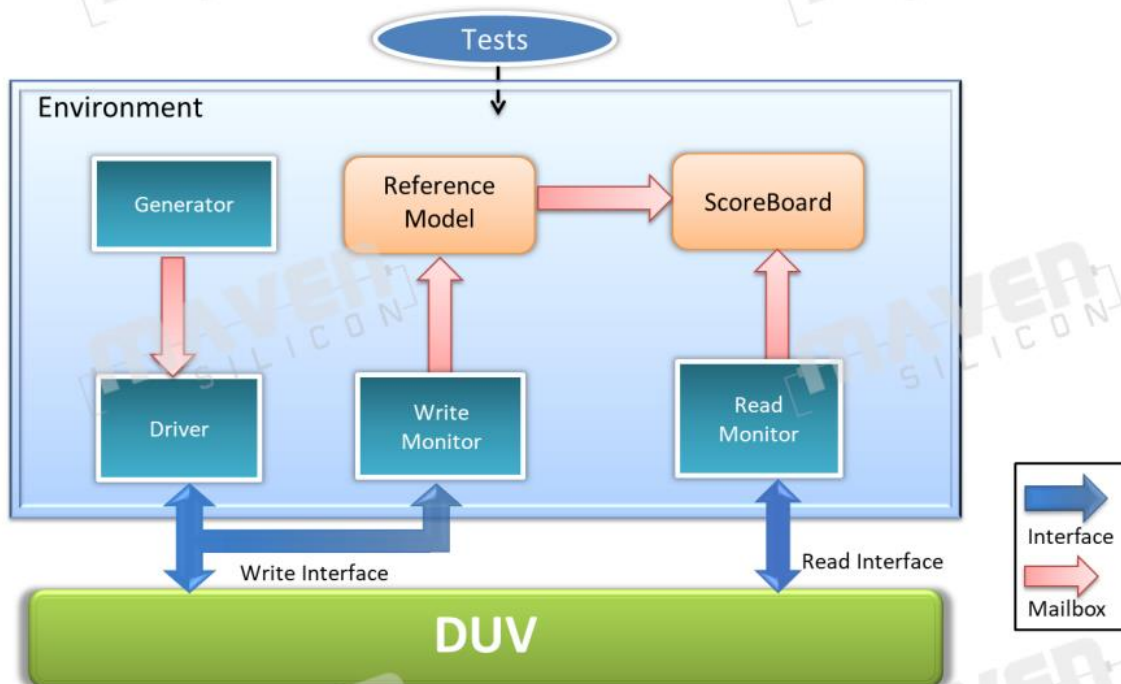
Features:

- Reset
- Mode
  - ❖ UP: Increment by one on each clock cycle.
  - ❖ DOWN: Decrement by one on each clock cycle.
- Load
- Data_in
  - ❖ When Load is high, Counter can be loaded with the initial value data_in, which sets the starting point for counting.
- Data_out

Strategies:

- Reset
  - ❖ Reset should be distributed such that low value should occur number of times than high.
  - ❖ Reference Model: if reset is "1" then output variable made equal to "0".
- Load
  - ❖ Load should be distributed such that low value should occur number of times than high.
  - ❖ Reference Model: If Load is high output variable should be data_in.
- Mode
  - ❖ Mode can be equally distributed.
  - ❖ Reference Model:
    - ▪ UP: if mode is low then if output variable is "9" then output variable made equal to "0" otherwise increment by "1".
    - ▪ DOWN: if mode is high then if output variable is "0" then output variable made equal to "9" otherwise decrement by "1".
- Data_in
  - ❖ Data_in is randomized such that value of data_in should be in the range of 0 to 9.
- Data_out
  - ❖ Score board: Data_out is compared with the output variable in reference model.

Transaction:

- Random: Reset, Mode, Load, Data_in

Transactors:

- Generator: Generates random transaction.
- Driver: Drives Reset, Load, Mode and Data_in signals.
- Monitor: collects the Data_out and random transaction.
- Reference Model: Mimic the design inside the environment from transaction data which is collected in the monitor.
- Scoreboard: Compares the data in reference model and Data_out generated in the design and generates coverage.

Coverage Model:

- Reset {implicit bins}
- Load {implicit bins}
- Mode {implicit bins}
- Data_in {explicit bin [bin range [0:9]]}
- Data_out {explicit bin [bin range [0:9]]}
- Load X Data_in
- Mode X Load X Data_in

Callbacks:

- Scoreboard : Callback trigger the coverage model.