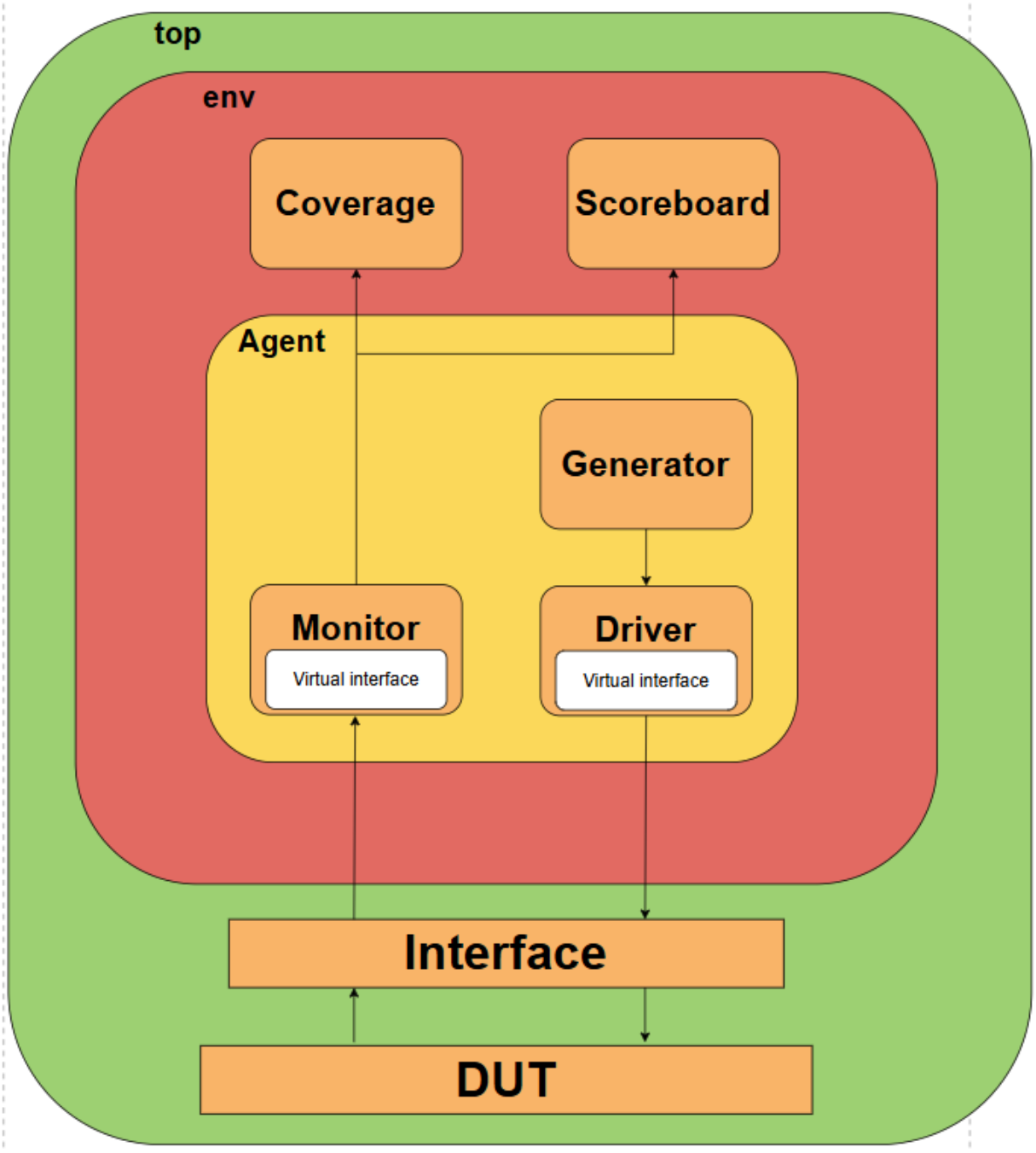
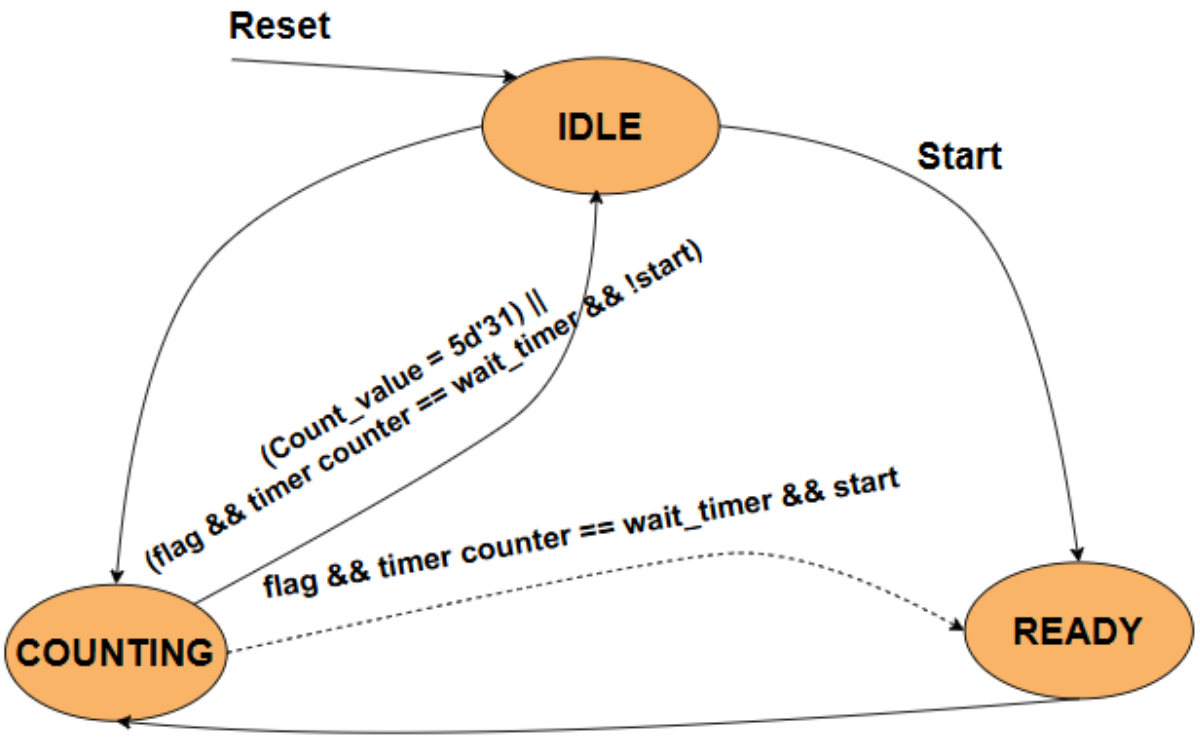
Abdelrahman Khaled Fouad

Verification Documentation of Counter_FSM

Verification Diagram



Scoreboard vs Design FSM



- **Design supports consecutive transactions; however, this behavior is not explicitly specified in the design specification. Therefore, consecutive transactions were not modeled on the scoreboard and were treated as out-of-scope (represented as dashed transitions).**

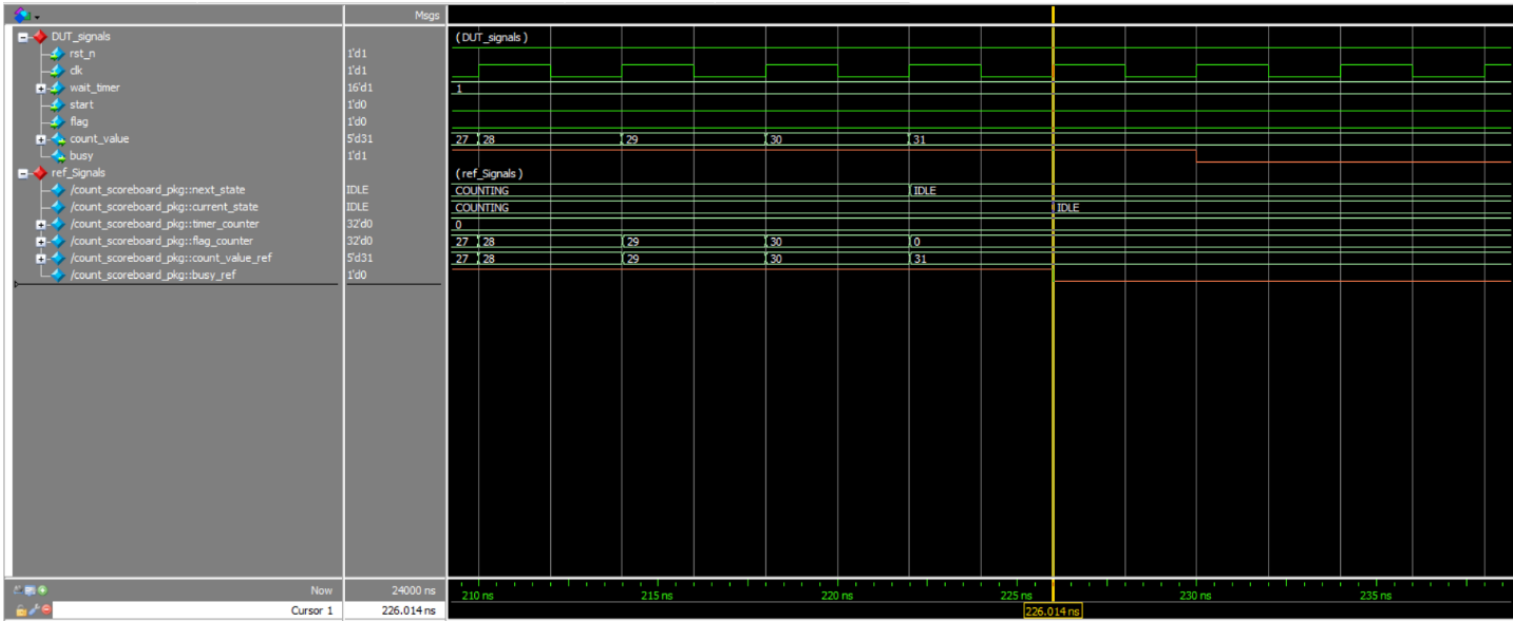
Verification plan

Label	Description	Stimulus generation	Functional Coverage
counter_1	**Testing reset signal**	**Directed testing at start of the simulation and randomized throughout the simulation**	**Bin for reset signal**
counter _2	**Testing flag signal and its effect on DUT**	**Directed testing throughout the simulation**	**Bin for flag signal**
counter _3	**Testing counter is reaching max value**	**Directed testing throughout the simulation**	**bins for count value signals**
counter _4	**The counter DUT was exercised using constrained-random tests with a fixed wait_timer, start asserted 60% of the time, and flag deasserted in 85% of the cases**	**Randomized testing throughout the simulation**	**Bin for wait timer, bin for start, bin for busy**

Errors detected

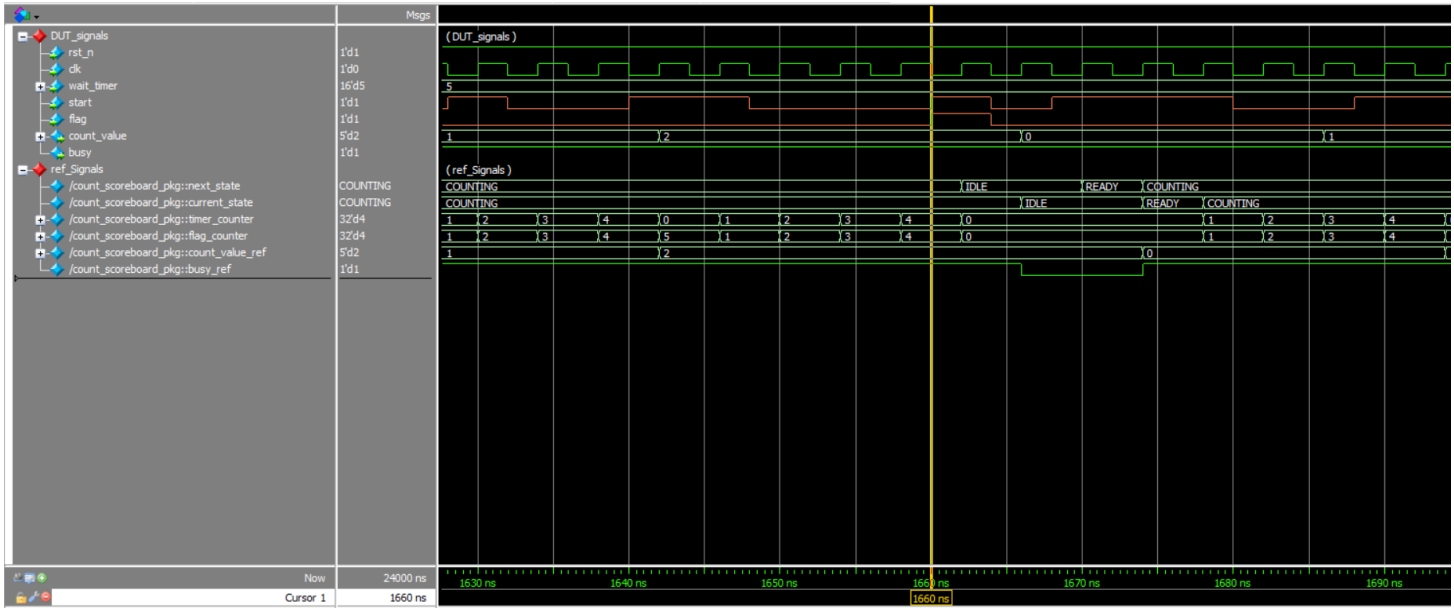
1) Busy Deassertion Timing

When the count value reaches its maximum value, the busy signal is deasserted with a one-cycle delay instead of being deasserted immediately.



2) Consecutive Transaction Handling

The DUT specification does not support consecutive transactions. When start and flag are asserted simultaneously, the DUT transitions through the READY state instead of returning to the IDLE state. This behavior is not modeled or handled by the scoreboard reference model



3) Start Signal Behavior

Keeping the start signal continuously asserted does not initiate a new transaction. To trigger a new transaction, start must be deasserted and then asserted again.

Report summary

```
# Simulation summary: Correct=5928 Errors=72
# ** Note: $stop    : count_top.sv(56)
#    Time: 24 us  Iteration: 0  Instance: /count_top
# Break in Module count_top at count_top.sv line 56
```

Functional Coverage

Covergroups

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances	Ge
/count_coverage_pkg/count_coverage		100.00%						
TYPE count_cg		100.00%	100	100.00...			auto(1)	
CVP count_cg::{#coverpoint__0#}		100.00%	100	100.00...				
bin start_0		5698	1	100.00...				
bin start_1		302	1	100.00...				
CVP count_cg::{#coverpoint__1#}		100.00%	100	100.00...				
bin flag_0		5924	1	100.00...				
bin flag_1		76	1	100.00...				
CVP count_cg::{#coverpoint__2#}		100.00%	100	100.00...				
bin all_values		6000	1	100.00...				
CVP count_cg::{#coverpoint__3#}		100.00%	100	100.00...				
bin busy_0		5372	1	100.00...				
bin busy_1		628	1	100.00...				
CVP count_cg::{#coverpoint__4#}		100.00%	100	100.00...				