# Test Plan for “2-input OR gate with interfaces”

1. Design specification
2. Logistics
3. Environment
4. Test Cases
5. Goals
6. Schedule
7. Protocol

# TEST PLAN

## DESIGN SPECIFICATION

* Implements a 2 input OR Gate
* Interface: Simple RDY/EN protocol
  + Producers assert RDY when they have data to offer and keep it asserted until EN is asserted
  + Consumers assert RDY when they can consume data and keep it asserted until EN is asserted
  + EN is asserted when both Producer. RDY and consumer. RDY are asserted.
* Can take 0-20 cycles to produce/consume data.

## LOGISTICS

* Machine: 1 Laptop
* Repository: Github ( <https://github.com/learn-cocotb/assignment-2-interfaces-AhmedHany1212> )
* Regression: Github Actions
* Issue tracking: Github Issues ( <https://github.com/learn-cocotb/assignment-2-interfaces-AhmedHany1212/issues> )
* Software: Python >3.6, iverilog, cocotb, notepad++
* Licenses: None
* BFM: None

## ENVIRONMENT

* verification components (Driver, Monitor, Assertions, Scoreboard)

## TEST CASES

* Case 1
  + Feature: OR Gate Datapath test
  + Description: Give inputs to the a and b pins of the DUT and check whether the expected value matches the output of the DUT
* Case 2
  + Feature: OR Gate Datapath Randomize test
  + Description: Give random inputs to the a and b pins of the DUT and check whether the expected value matches the output of the DUT
* Case 3
  + Feature: OR Gate input status test
  + Description: Give inputs to a or b pin of the DUT and check the status of the same pin without reading
* Case 4
  + Feature: OR Gate output status test
  + Description: Give inputs to a and b pin of the DUT and check the status of output pin then read then check the status of output pin again
* Case 5
  + Feature: OR Gate input enable test
  + Description: Give inputs to a or b pin of the DUT without enable then check the status of same pin
* Case 6
  + Feature: OR Gate output enable test
  + Description: Give inputs to a and b pin of the DUT with enable then read output pin without enable then check the status of output pin
* Case 7
  + Feature: OR Gate output status test using one input pin
  + Description: Give inputs to a or b pin of the DUT and check the status of output pin then read then check the status of output pin again
* Case 8
  + Feature: OR Gate Overwrite test
  + Description: Give inputs to a or b pin of the DUT and give different input to same pin again then give input to another pin then read and check the output
* Case 9
  + Feature: OR Gate reversed addresses test
  + Description: Give input enable using output address then check the inputs status
* Case 10
  + Feature: OR Gate unreserved addresses test
  + Description: Give input enable using unreserved address (6 or 7) then check the inputs status
* Case 11
  + Feature: OR Gate reset test
  + Description: Give inputs to a and b pin of the DUT then reset system then read and check the output data

## GOALS

* 100 % Functional coverage
* Schedule

## SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testcase | Environment | Start | End | Status |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## PROTOCOL

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Enable | Ready | Description |
| don’t care | 0 | 0 | Idle |
| valid data | 0 | 1 | Ready |
| valid data | 1 | 1 | Transaction |

* The case where enable is 1 and ready is 0 cannot occur.
* Once ready goes high data cannot change until transaction completes.