# INTERFACES TEST PLAN

## Design Specification

### IP Introduction

A typical DUT has 3 sets of interfaces

1. Data Interface
2. Control Interface
3. Configuration Interface

The data interface is used for Data I/O

The control interface is used by other hardware to control or change the behavior of the IP

The configuration interface is used by the processor to change the behavior of the IP.

### IP Functionality

It uses the RDY EN Protocol on all interfaces

The DUT takes N 8 bit input values. Add them and output the result.

The Value of N can be provided either

* At the Interface via the length port
* Or via the length register accessible via the configuration register.

The decision on which length is used (port or register) is controlled via a bit in the configuration space.

Once the DUT starts accumulating the bytes it ignores changes to the length field or register until it has generated the output.

The generated output is stored in a FIFO . While the FIFO is not empty output can be popped from the queue. When the FIFO gets full the DUT stops accepting more data.

### Interfaces

[TO DO] Add diagram here

| **Port** | **Direction** | **Width** | **Description** |
| --- | --- | --- | --- |
| din\_rdy | out | 1 | RDY for Input data |
| din\_en | in | 1 | EN for Input data |
| din\_value | in | 8 | Data |
| dout\_rdy | out | 1 | RDY for Output data |
| dout\_en | in | 1 | EN for output data |
| dout\_value | out | 8 | Accumulated data output |
| len\_rdy | out | 1 | RDY for Length |
| len\_en | in | 1 | EN for length |
| len\_value | in | 8 | Number of bytes that should be accumulated |
| cfg\_rdy | out | 1 | RDY for Configuration Interface |
| cfg\_en | in | 1 | EN for Configuration Interface |
| cfg\_address | in | 8 | Address of the register in cfg space |
| cfg\_op | in | 1 | Operation type, 0=Read, 1=Write |
| cfg\_data\_in | in | 32 | The data that needs to be written, ignored for read |
| cfg\_data\_out | out | 32 | The data returned by read operation |

### Configuration Space Register Map

| **Address** | **Access** | **Bit map** | **reset value** | **Field** | **Description** |
| --- | --- | --- | --- | --- | --- |
| 0 | R | 7:0 | 0 | current\_count | The count of bytes processed |
| 0 | R | 15:8 | 0 | programmed\_length | The length programmed for this session |
| 0 | R | 16 | 0 | busy | An operation has started and is ongoing |
| 4 | R/W | 0 | 0 | s/w override | 0 => use len from port. 1=>use len from register |
| 4 | R/W | 1 | 0 | pause | 0 => normal mode ( configuration previously stored is used again) 1=>Input Rdy will be deasserted after end of current data set. |
| 8 | R/W | 7:0 | 0 | len | len register |

Interface : Simple RDY/EN protocol

Produces assert RDY when 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/Dyumnin-Interns/interfaces-SiliconMerc> )

Regression : Github Actions

Issues :

* <https://github.com/Dyumnin-Interns/interfaces-SiliconMerc/issues/2>
* <https://github.com/Dyumnin-Interns/interfaces-SiliconMerc/issues/3>

Software : Python > 3.6 , iverilog , xcelium/vcs/questasim,cocotb,cocotb-bus,cocotb-coverage

Licenses : Cocotb (BSD 3-Clause "New" or "Revised" License) , Gtkwave(GNU General Public License v2.0)

BFM : None

## Assertions

Assertion 1 : Length and data can't be read at the same time until the process is complete.

Assertion 2 : When the sum of data values overflows or underflows trigger this assertion.

## Test Cases

### Functional Coverage

##### TC\_1\_FIFO

Feature : Randomised FIFO full test

Description : The output enable pin is set to 0 and input is fed until data and length en gets low which means that the FIFO is full and the design cannot accept more data . Then remove all the data from the output queue and see that the FIFO is empty in the end.

Scenario : Randomised corner case test

Given : Unit Test Environment

##### TC\_2\_LEN\_DATA0

Feature : Randomised LEN and SUM 0 operation

Description : DUT Operation when Data length is 0 or Data is 0

When we feed an invalid length 0 then the data.en pin should remain low i.e the design should not accept data . Although if length is non zero then the data sum of zero should also be allowed.

Scenario : Randomised corner case test

Given : Unit Test Environment

##### TC\_3\_CONFIG\_RW

Feature : Directed Configuration Read and Write Data Interface

Description : The length is set using the length port or written directly to the configuration register . The values set in the register are verified by reading the programmed values .

Scenario : Directed test

Given : Unit Test Environment

#### Datapath Verification

##### TC\_4\_DATAPATH\_VERIF

Cross coverage test

Datapath coverage

Feature : Directed Datapath Randomize test

Description : Give random inputs to the din and length pins of the DUT , cross coverage inputs of configuration parameters and delays and check whether the expected value matches output of the DUT

Scenario : Random test

Given : Unit Test Environment

When : Input len is 10 samples of random.randint(0,)

AND : Input din is 10\*len samples of random.randint(0,1)

AND : din delay is random.randint(0,20)

AND : len delay is random.randint(0,20)

AND : dout delay is random.randint(0,20)

Then : Output is sum(din[0:len-1])

Coverage:

Datapath Cross Coverage

We try all possible combinations of s/w overide , pause bits.

Bins

s/w overide :[0,1] , pause :[0,1]

Cross s/w overide x pause

#### Protocol Verification

##### TC\_5\_PROTOCOL\_VERIF

Feature : Randomised test for Protocol Verification

Description : Protocol verification of Data bus , Len bus , Output Bus, Configuration Bus with Randomised Delays on the interfaces

Protocol Verification : State Diagram , verify all possible state transitions

| Data | En | Rdy | Description |
| --- | --- | --- | --- |
| dont care | 0 | 0 | Idle |
| valid data | 0 | 1 | Ready |
| valid data | 1 | 1 | Transition |

Scenario : Random test

Given : Unit Test Environment

Bins

Current state = [Idle , Ready , Transaction]

Previous state = [Idle,Ready,Transaction]

Cross : Current state x Previous state

## Schedule

| Resource | Testcase | Enviroment | Start Date | End Date | Status |
| --- | --- | --- | --- | --- | --- |
| Pranav | TC\_1\_FIFO | Unit | 6 March 2023 | 6 March 2023 | Pending |
| Pranav | TC\_2\_LEN\_DATA0 | Unit | 6 March 2023 | 6 March 2023 | Pending |
| Pranav | TC\_3\_CONFIG\_RW | Unit | 7 March 2023 | 7 March 2023 | Pending |
| Pranav | TC\_4\_DATAPATH\_VERIF | Unit | 8 March 2023 | 8 March 2023 | Pending |
| Pranav | TC\_5\_PROTOCOL\_VERIF | Unit | 9 March 2023 | 9 March 2023 | Pending |