AEDVICES Consulting
166B Rue du Rocher de Lorzier
38430 Moirans– France
trainings@aedvices.com
http://www.aedvices.com

Training on
Functional Verification Methodology
Using UVM

LAB
Interfaces
In SystemVerilog

Objectives

Interfaces are constructs in SystemVerilog that permit the user to encapsulate the communication between the blocks. It is very useful to describe buses connection and can help the design to drive signals to a DUT by another module.

Global Explanation

In this lab we will modify the Full Adder DUT already used in the last lab to support the interface. The interface we will be created by the student specifying two types of module ports (*modports*) and a driver which, through the interface, will drive the DUT signals and check the result (The role of checker is not usually played by the driver, but in this case, since the student will create a simple environment, it is not a problem). This driver will perform all the possible entries to *data1*, *data2* and *cin*.

Get Started

Untar the training tarball:

```
mkdir <TRAINING_WORKAREA>
cd <TRAINING_WORKAREA>
tar zxvf <TRAINING_NAME>.tgz
```

Lab files can be found under:

<TRAINING_NAME>/labs/<LABNAME>

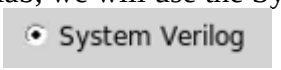
To launch the simulation,

```
cd simulation/<SIMULATOR>
./runsim.sh
```

Note: in order to launch using LSF, you must first setup the variable LAB_LAUNCHER

```
setenv LAB_LAUNCER 'bsub -I -q gui -P mcdverif -R "select[rh60]"'
```

In this lab, we will use the SystemVerilog mode. So the following checkbox should be set:



Browse and load the labs/NN-LABNAME/lab.sv file

Instructions

Step 1: Create the Interface.

- Open the file: <SANDBOX>/labs-Xdays/labNN-systemverilog_interfaces /adder_if.sv
- Search for ***LAB-TODO-STEP1-a***
- Declare the interface ports (The same as the DUT).
- Search for ***LAB-TODO-STEP1-b***
- Declare the *modport* to specify the interface to the Adder.
- Search for ***LAB-TODO-STEP1-c***
- Declare the *modport* to specify the interface to the Driver (Tip: The driver must drive the DUT signals).
    - What is the difference from the Adder one?

Step 2: Adapt Full Adder

- Open the file: <SANDBOX>/labs-Xdays/labNN-systemverilog_interfaces /dut/full_adder.sv
- Search for ***LAB-TODO-STEP2-a***
- Declare the interface previously created as a module port.
    - Do not forget to specify the *modport*.
- Search for ***LAB-TODO-STEP2-b***
- Analyse how the interfaces signals are used.
-

Step 3: Complete the driver

- Open the file: <SANDBOX>/labs-Xdays/labNN-systemverilog_interfaces /driver.sv
- Search for ***LAB-TODO-STEP3-a***
- Declare the interface previously created as a module port.
- Search for ***LAB-TODO-STEP3-b*** to ***LAB-TODO-STEP3-e***
    - See how the signals are accessed and driven
- Search for ***LAB-TODO-STEP3-f***
- Drive the interface to get the *result* and the *cout* (Inspire yourself by the Step-3c).
    - RESULT Address: 0x04
    - COUT Address: 0x05

Step 4: Complete the testbench

- Open the file: <SANDBOX>/labs-Xdays/labNN-systemverilog_interfaces /tb.sv
- Search for ***LAB-TODO-STEP4-a***
- Instantiate the Interface.
- Search for ***LAB-TODO-STEP4-b***
- Instantiate the DUT module.
- Search for ***LAB-TODO-STEP4-c***
- Instantiate the Driver program.
    - The DUT and Driver signals are already connected?