Design and Verification of I²S Protocol using UVM

Vineeth V¹, Sujatha S²
M. Tech Scholar¹, Associate Professor²
Department of Electronics and Communication Engineering
Bangalore Institute of Technology, Bangalore, Karnataka

Abstract:
A present day test bench verification environment can provide stimulus for inputs and checks the results for the test independently, which is controlled by constraints which are user-specified. This is far more efficient than using a standard test bench to write individual tests by hand. To produce a constrained-random test bench, several verification approaches have been developed. Here SystemVerilog based UVM verification has been used. UVM is carried out using SystemVerilog based functional verification which includes SystemVerilog library for easier code generation. Many digital audio systems such as compact discs, digital audiotapes, sound processors, and TV sound are being brought into the consumer audio industry. A Standard Communication interface is essential for both apparatus and IC because they increase the system’s liveness. Thus inter-IC sound (I²S) interface, A bus as a type of serial link has been designed specifically for digital audio. In this paper design of I²S is done using Verilog and verification is done using SystemVerilog and UVM in tools EDA Playground and Questa Sim.

Keywords: I²S; Functional Verification; SystemVerilog; UVM;

I. INTRODUCTION
A number of devices including as ADCs, DACs, DSPs, Digital I/O interfaces, and others, process digital audio signals in most systems. It is vital to have standard communication interface for both the equipment and the silicon developers in order to increase flexibility and interoperability. I²S is sometimes confused with FC. Only similarity between these interfaces is that it’s developed by PHILIPS semiconductors and both interfaces start with the letter “I” to indicate that they are meant for inter-IC communication.

I²S, on the other hand, was released after I²C, and whereas I²C is a general-purpose interface, I²S is meant to convey audio data—the “S” in the acronym stands for “sound.” SystemVerilog has combined the features of Hardware Description Languages (HDLs) like Verilog and VHDL with features from specialized Hardware Verification Languages, as well as features from C, C++. SystemVerilog has been the industry's first Hardware Description and Verification Language (HDVL).

In the areas of compact and productive RTL coding, Assertion Based Verification, constructing the coverage-driven, employing constrained random approaches, SystemVerilog is finding practical application in these areas.

UVM is a SystemVerilog class library that was created with the goal of assisting people in the creation of modular, reusable verification components and test-bench because UVM IP is an industry-standard, we can get it from other sources and utilize it in our environment.

We would have to build everything from scratch if we don't use UVM. The paper outline is as follows: section II describes the specification of I²S protocol, section III describes developing verification environment, section IV describes the results of the verification.

II. I²S PROTOCOL
The bus has three lines[1]:

- Serial clock(SCK)
- Word select(WS)
- Serial data(SD)

A. Serial data
Since transmitter and receiver might have various word lengths serial data is transmitted with MSB first. The transmitter does not need to know how many bits the receiver can handle, and the receiver does not need to know how many bits are being communicated.

The rising edge that is High to Low or falling edge that is low to high of the clock signal can be used to synchronize serial data transmitted by the transmitter when transmitting data that is synched with the leading edge.

B. Word select(WS) or Left-Right clock (LRCLK)
The word select line or left-right clock line indicates the channel being transmitted:

- If WS = 0 left channel is selected.
- If WS = 1 right channel is selected.

WS might alter either on a falling or rising edge of the serial clock, but there is no necessity for it to be symmetrical.
C. Timing

All timing constraints are expressed in terms of a device’s clock period or minimum permissible clock period[1]. As a result, higher data rates will be available in the future.

III. VERIFICATION ENVIRONMENT

A structured verification environment[2] is developed as shown in Fig. 2. The components and their functions as follows:

- Sequence item- A base transaction or Data input is randomized in this class.
- Sequencer- It acts as a handshake signal between Sequence and Driver
  - Driver- Transaction levels are converted to signal level and driven to Design under test(DUT)
  - Monitor- It performs an inverse function of a driver, Signal level are converted to transaction level
- Active monitor- It observes input changes from DUT
- Passive monitor- It observes output changes from DUT
- Active agent- Agent which has a driver is considered active
- Passive agent- Agent without a driver is considered passive
- Interface- Used to drive signals in and out of DUT
- Design under test(DUT)- It consists of an actual I2S design.
- Scoreboard- It has compare logic with respect to input to DUT and output from the DUT.

IV. RESULTS AND CONCLUSION

Fig. 4.1I2S design using Verilog, indicates that a 32-bit data is passed to I2S interface, based on LRCLK right and left channels of I2S bus it transfers bit by bit at every clock pulse and when all the bits are shifted, SD has output data. Fig.4.2 indicates the result of SystemVerilog verification environment. Different layers of class based verification code is written with constrained random stimulus and the results are displayed in transcript window. A base packet is created and randomized. Randomized packets are driven by the driver to DUT through interface, monitor observes the changes in output, Scoreboard compares the data sent to DUT and out of DUT. It is reported using display statements as shown. Fig.4.3 is the result of UVM based verification environment where customization is done using Factory and Config_db mechanisms and display of results are done using reporting mechanism.

Figure.4.1. I2s design waveform using verilog

Figure.4.2. Verification results for i2s using systemverilog architecture
2C Master Controller using System Verilog

- gav Tarpara, “Design
IJESC, Industrial SOC Design” Simulation Strategy after Model Checking: Experience in Thread Player [2015].


Layered testbench of UVM makes testing process more efficient and enhances code reusability from simulation. This project has successfully developed I2S design using Verilog and verification environment is developed using SystemVerilog and UVM. Constrained Randomization techniques have been used to hit the corner testcases making it more effective and efficient. This can be further developed to use different masters and slaves along with I2S.

ACKNOWLEDGEMENT

The Author express sincere gratitude to Mrs. Sujatha S, Department of Electronics and communication, BIT, Bangalore, India for guidance and support. The author is also thankful to other Professors and technical and non-technical staff of Department of Electronics and communication, BIT, Bangalore, India for extending their help & support in giving some ideas about the paper without which it’s not possible to come up with this paper.

IV. REFERENCES


[20] Young-Nam Yun, Jae-Beom Kim, Nam-Do Kim, and Byeong Min. Beyond UVM for practical soc verification. In SoC Design Conference (ISOCC), 2011