Functional and Performance Verification of Bit Move Block on AHB Bus

Sankannavar, Akash (MS Electrical Engineering)
Kanth, Krishna (MS Electrical Engineering)

Methodology

Overview of Verification Plan

To develop a verification environment it is important to understand the basic functionality of the DUT. The basic connection of all the components is shown in the figure below. The bit move block uses the AHB bridge for communication. For the DUT to start working an initial address pointer must be given from the testbench. Once the operation is started the bit move block uses a memory slave to fetch all the source and destination data. In order to verify the functionality of the design the testbench creates a memory slave and a reference model that will in later part will help to verify the functionality of the design. Since performance plays an important role the testbench also monitors the performance of the design by allocating a maximum number of clock cycles for each operation.

Inputs and Stimulus

Sequence generates different testcases which includes source address, destination address and block length. The testcases are checked for overlap and are added to a queue. The memory block is created by placing data randomly in the source and destination address. To create the reference model the source data is transferred to the destination address one at a time using a for loop. For the DUT operation to begin a control block is generated by assigning the source and destination address along with the block length to the control block.

The driver has a case statement controlled by the sequences. The sequences first initialize the memory in the driver and then write the first control block on the AHB bridge. The driver then waits to receive a done signal from the DUT. Once the operations are complete the driver sends the modified data to the scoreboard to verify with the reference model.

To verify the functionality, monitor 1 is used to observe the signals on the AHB master interface and pass it on to scoreboard 1. This scoreboard checks the reference model memory generated in the sequences and the modified model obtained from the driver to verify whether the operation was successful. If there is a data mismatch the scoreboard gives out an error which displays the actual data and the expected data at that particular address. To measure the performance, Monitor 2 is used to observe the signals on the Bit move block. The scoreboard 2 creates the data decoding model of the bitmove and calculates the maximum number of clocks allowed for that particular operation. Scoreboard 3 calculates the total number of clocks taken by the bit move to perform the operation. The result from scoreboard 2 and scoreboard 3 are displayed in scoreboard 3 to measure the performance of the DUT.

Functional verification results

The testbench generated 20 different sequences to verify the functionality of the bit movement block placed on the AHB bus. The verification environment developed was used on different DUT’s. The reference model DUT passed the verification environment. When the testbench was used on a broken DUT data mismatch errors were observed as shown in the figure below. The testbench gives the address of the data mismatch along with the expected and required data to be present in that block. The results of the broken DUT is shown in figure below.

Performance results

Performance is one of the key aspects to be measured for the bit move block. The testbench provided a limited number of clock cycles for each operation to be completed. The clock cycles were limited to ((block length/10) + 15) clocks. The reference DUT successfully completed the operation in the specified clocks. For a DUT with complex design the number of clocks taken to complete the operation exceeded the specified limit. The performance results of the complex DUT is shown in the figure below.

Analysis and Results

The finite state machine coverage report for bit move and AHB bridge is shown below.

Summary/Conclusions

The project demonstrates a robust and reusable testbench for verifying a bit-oriented design on a word-oriented bus. The exhaustive functional verification ensures that a good bit move design operates correctly in all the conditions without breaking down. The stringent performance verification lets the users observe the speed of the bit movement block. The coverage implementation ensures the design has no redundant code, this helps in reducing the number of gates. The test bench successfully detects faulty DUT’s and DUT’s with bad performance.

Key References


Acknowledgements

We would like to thank Prof. Lili He for her guidance and suggestions throughout the project. At the same time, the project would have endured great difficulties without the support and guidance of Prof. Morris Jones. The UVM sequence file provided by Professor Jones has helped us significantly in building the test environment and verifying the DUT. We would like to thank Prof. Morris Jones for all the ideas and support for the successful completion of the project.