

DEVELOPMENT , TESTING AND VERIFICATION OF IP CORE FOR SYNCHRONOUS SERIAL PORT (S-PORT) INTERFACE USING OVM

Bonum Sherley Phebe ¹, J Anusha ²

P.G. Student, Department of Electronics and communication Engineering, JNTU College of Engineering Aantapur, Andhra Pradesh, India¹

Lecturer, Department of Electronics and communication Engineering, JNTU College Of Engineering Anantapur, Andhra Pradesh , India²

Abstract - : *The aim of the project is to develop SPORT Verification IP using open verification methodology to verify SPORT module. Half the full clock rate at the processor, serial communication ports are operated at a maximum data rate of n/2 Mbps, where n-equals the processor core clock frequency. Synchronous serial ports, or SPORTs, support a variety of serial data communication protocols and can provide a direct interconnection between processors in a multiprocessor system. Bidirectional functions can provide flexibility for serial Communication. The motivation for the project is developing an OVM based verification environment which is able to perform the functional verification of the QSPI module and obtain a respectable coverage to avoid costly repines. Without it saving time and improving the verification effort by covering all the difficult corners of design and attaining maximum possible coverage will be difficult.*

Key Words: Serial Port(SPORT) ,Open Verification Methodology(OVM) ,Intellectual Property(IP), System on chip(SoC) , Coverage Driven Verification(CDV).

1.INTRODUCTION

The speedy progress of VLSI technology enables the integration of more than several million transistors in a single chip to make a SoC (System-on-Chip). Synchronous serial ports, SPORTs, support a variety of serial data communications protocols and direct interconnection can be provided between processors in a multiprocessor system.

It provides the high-level data structures available in object-oriented languages, such as C++. These data structures enable a higher level of abstraction and modeling of complex data types. The System Verilog also provides constructs necessary for modeling hardware concepts such as cycles, tri-state values, wires, just like Verilog hardware languages. So System Verilog can be used to simulate the HDL design and verify them by high level test case. OVM is a methodology for the functional verification of digital hardware, primarily using simulation.

The hardware system to be verified would typically be described using Verilog, System Verilog, VHDL or SystemC at any appropriate abstraction level. This can be behavioral, register transfer level, or gate level. OVM is explicitly simulation-oriented, OVM can also be used along side assertion-based verification, hardware acceleration or emulation. If we currently run RTL simulations in Verilog or VHDL. OVM as replacing whatever framework and coding style you use for your test benches. But OVM test benches are more traditional than HDL test benches, which might wiggle a few pins on the design-under-test (DUT) and rely on the designer to inspect a Design and development and testing for verification of IP core for SPORT interface waveform diagram to verify correct operation.

OVM test benches are complete verification environments composed of reusable verification components, and used as part of an overarching methodology of constrained random, coverage-driven, verification. But, UVM(Universal Verification Methodology) is compared to OVM little bit difficult.

2. SERIAL PORT(SPORT)

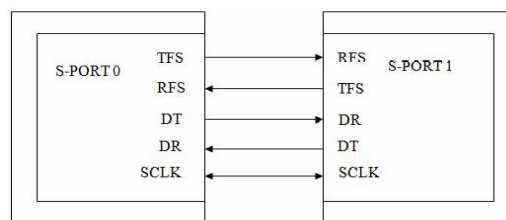


Fig -1: SPORT Interface

Synchronous serial ports, or SPORTs, support a variety of serial data communications protocols and can provide a direct interconnection between processors in a multiprocessor system. A SPORT receives serial data on its DR input and transmits serial data on its DT output. It can receive and transmit simultaneously, for full duplex operation. The data bits

are synchronous to the serial clock SCLK, which is an output if the processor generates this clock or an input if the clock is generated externally. Frame synchronization signals RFS and TFS are used to indicate the start of a serial data word or stream of serial words.

Data to be transmitted is written from an internal processor register to TX FIFO via processor bus interface. This data is then transferred to the transmit shift register. The bits in the shift register are shifted out on the SPORT's DT pin, MSB first, synchronous to the serial clock. The receive portion of the SPORT accepts data from the DR pin, synchronous to the serial clock. When an entire word is received, the data is written to RX FIFO, where it is available to the processor. Enabling the SPORT TX and writing to the TX FIFO, readies the SPORT for transmission. The TFS signal initiates the transmission of serial data. Once transmission has begun, each value written to the TX register is transferred to the internal transmit shift register and subsequently the bits are sent, MSB first. Each bit is shifted out on the rising edge of SCLK.

In the receiving section, bits accumulate as they are received in an internal receive register. When a complete word has been received, it is written to the RX FIFO and the receive interrupt for that SPORT is generated.

2.1 DESIGN PRINCIPLES

Verilog HDL is a hardware description language, it can be used for different levels of logic design, can be used for digital system logic simulation, timing analysis and logic synthesis, a wide range of applications. In this paper, a S-PORT Interface Module is designed using verilog, to achieve a multiprocessor communication within a single PCB(Printed Circuit Board). This happens through the synchronous serial peripheral port where in one serial peripheral port communicates with another serial peripheral port of another processor through a suitable communication channel.

2.1.1 Module design:

The modules which are to be individually coded and to be Instantiated are

- 1) Clock generator
- 2) FIFO
- 3) Transmitter-Receiver logic

4) C.P.U interface

3. OPEN VERIFICATION METHODOLOGY (OVM/OVC)

OVM is a open source methodology for functional verification using system verilog from Mentor + Cadence. OVM combines mentor's AVm and Cadence ERM Methodology features . Its scalable to system level verification. An OVM testbench is composed of reusable verification environments called OVM verification components (OVCs)

3.1.1 Benefits of OVM:

- Improves productivity
- Increases and ensures reusability
- Maintenance of the verification components will be much easier because the components are standardized.
- Independence between components. A test case writer does not need to understand the full flow of the architecture.

3.1.2 Coverage driver verification (CDV) or Constraint random verification :

Automated and random stimulus generation (Reduce the spent in creating 100's of directed test case). Adjust the seeds and constraints, generate new interesting scenarios.

Independent checking or self checking test benches (Generate a valid or invalid scenario. The checking will be done automatically in scoreboards)
Coverage metrics (How much we are done ??)

3.1.3 OVM Test

In the OVM methodology all tests are basically classes. And all test classes are extended by ovm_test class. A OVM test have following properties :

- 1) Unique to OVM.
- 2) Derived from ovm_test base class.
- 3) Tests are top level components that create the dynamic test bench.
- 4) They have all simulation phases.

5) Start_of_simulation(), run() are used to specify the run time parameters of the environment.

Using OVM test cases multiple tests are compiled together and a test case can be chosen at the run time using command line option. The constraint random test cases in the OVM are much shorter than the directed test cases. With ovm_test class we do not need to recompile the environment again to run a new test case which saves a great deal of time for big verification environments.

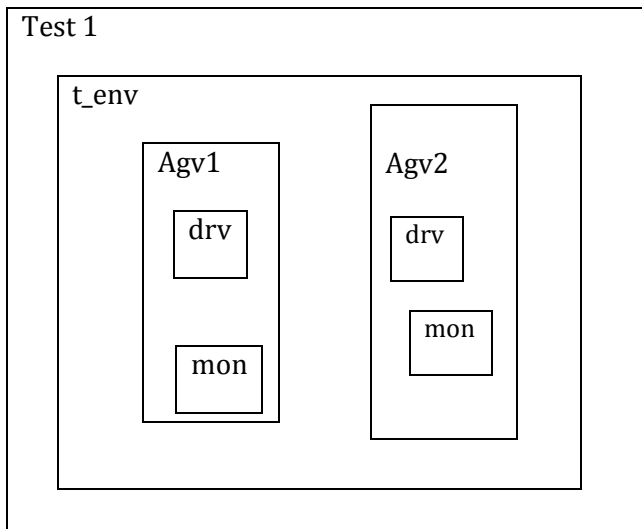


Fig-2 OVM Testbench

4.RESULTS

The alternate framing signal in the alternate framing mode should be asserted in the same SCLK cycle as the first bit of a word. Received data bits are latched on the falling edge of SCLK and transmitted bits are driven on the rising edge of SCLK, but the framing signal is checked only on the first bit. Internally generated frame sync signals remain asserted for the length of the serial word. Externally generated frame sync signals are only checked during the first bit time. In the normal framing mode, the framing signal is checked at the falling edge of SCLK. If the framing signal is asserted, received data is latched on the next falling edge of SCLK and transmitted data is driven on the next rising edge of SCLK. The framing signal is not checked again until the word has been transmitted or received.

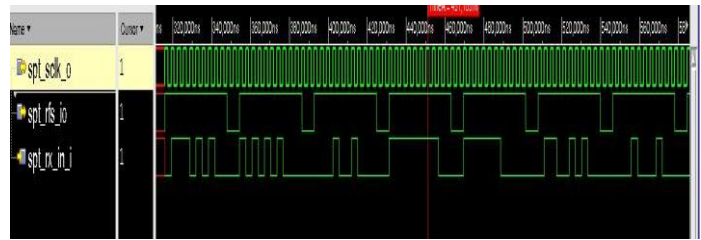


Fig4:Rfs_alt_frame signal

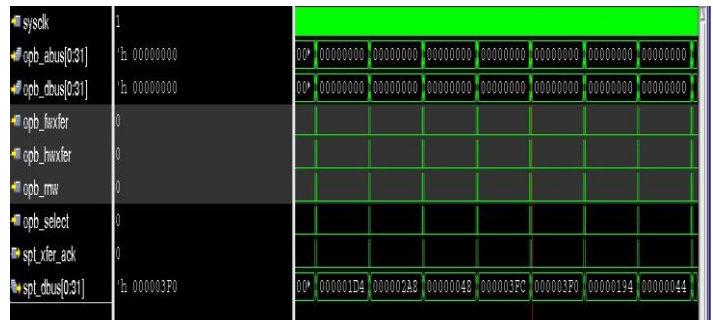


Fig5:Rfs_alt_frame read data signal

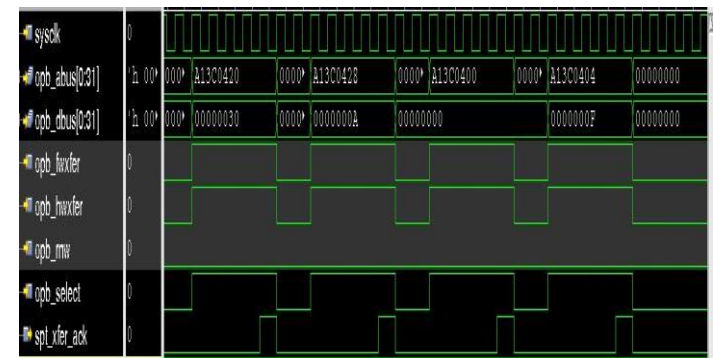


Fig6:Rfs_alt_frame registers

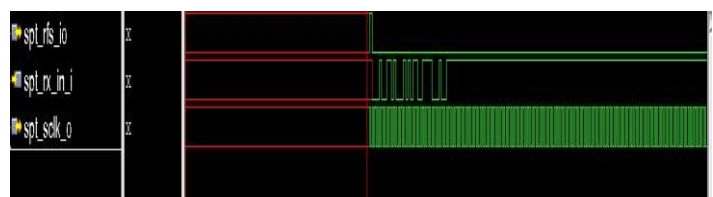


Fig7:Rfs_normal_un frame

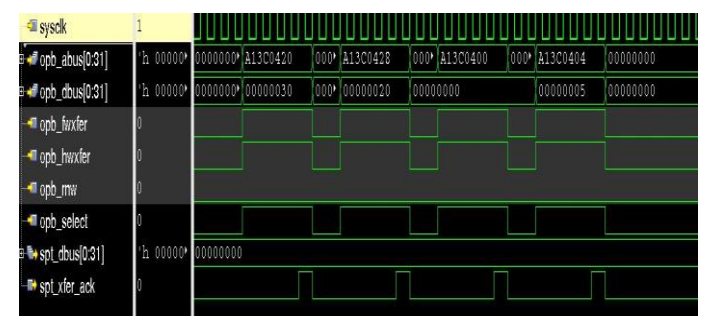


Fig8:Rfs_normal_un frame registers



Fig9: Rfs_un frame read data



Fig 10: S-PORT Results

Table-1: Simulation statistics of SV and OVM Environment

PARAMETER	SV Environment	OVM Environment
Elaboration Time	20.2731	14.4538
Elaboration CPU Time	2.9643	2.6208
CPU Time	0.7213	0.639604
Memory Usage	44MB	25MB

From the above comparison we can reach to a conclusion that OVM based environment is better in many terms than System Verilog environment.

5. CONCLUSIONS

The S-PORT has been implemented in synchronous mode only Even the design can be made more flexible by providing the programmable mode and command word registers that suit the requirements as per the application. The synchronous serial ports are mainly used in multiprocessor communication where the data transmission can takes place at a high speed within the system (i.e., for short distances). The System Verilog verification environment developed along with the complete flow of verification has been discussed. The various classes for driver monitor, stimulus, environment etc. and modules or programs are

compiled and simulated and the outputs observed are shown. The environment created completely wraps the DUT and functional and assertion based coverage have been found. Assertion coverage found is 100% and functional coverage is found as 99.21%.

REFERENCES

- [1] Naresh Patel, Vikaskumar Patel "VHDL Implementation Of UART With Status Register "2012 in (IEEE).
- [2] "TMS320C54X DSP CPU and peripherals: data book" Texas instruments inc.1997
- [3] Z.Navabi. "VHDL Analysis and modeling of digital Systems "MC.GRAWHILLS inc.1998
- [4] Cadence Design Systems, Inc., 2655 Seely Ave., San Jose, CA 95134, USA, Verstion1.1, May2008.
- [5] CHRIS SPEAR "SYSTEMVERILOG FOR VERIFICATION" Synopsys, Inc.
- [6] www.xilinx.com/support/documentation
- [7]www.xilinx.com/support/documentation/sw_manu al/.../x st.pdf
- [8] www.analog.com/processors and DSP/ Blackfin processors/
- [9]www.xilinx.com/products/silicondevices/fpga/virt ex 5/index.htm
- [10] Forums.xilinx.com silicon devices virtex R family FPGAs
- [11] Basic VLSI design by Douglas A.pucknell ESHRAGHIAN, KAMRAN,third edition.
- [12] National Instruments Serial Quick Reference Guide, February 2007.
- [13] Jiang Lidong. VHDL language program design and application. Beijing:Beijing University of Posts and Telecommunications Press, 2004.
- [14] Wang Shiyi. Digital signal processing. Beijing: Beijing Institute of Technology Press, 1997