# Design and Verification of SPI Master Core Using UVM

[1]Dr.Punith Kumar M B, [2]Sreekantesha H N

[1]Associate Professor, [2]PG Student
Department of ECE, PESCE, Mandya

***Abstract:*** Serial Peripheral Interface (SPI) is one of the commonly used bus protocol for connecting peripheral devices to microprocessor .SPI is full duplex, high speed an synchronous bus protocol used for on-board or intra-chip communication In this project the configurable architecture of SPI Protocol with Wishbone Interface has been designed .The main advantage of this design is it overcomes the weaknesses of traditional SPI Bus protocol. As the complexity of the circuit is numerous so there is need of verification methodology to quench the product failure. This project emphasizes on verification of SPI design using Universal Verification Methodology. In this project the verification components and testing environment has been constructed using SystemVerilog. The SPI core has been verified for all its design features with the help of different testcases scenarios also the functionality of the design has been exercised by introducing bugs in the design. The SPI core has been designed using verilog with some modifications to improve the performance of the design and verification has been done using Mentor Graphics Questasim 10.4e. We have achieved 100% functional coverage with no assertion failures in the verification.

***IndexTerms* - SPI, Wishbone, UVM, SystemVerilog.**

## I. INTRODUCTION

The development of advanced integrated circuits not only increased the complexity of IC design but also IC verification equally challenging. Around 70% spend on verification and verification is a critical path and time can be reduced through automation. many protocols used for both long and short distance communication purpose, long distance protocols is ETHERNET, USB, PCI-EXPRESS. SPI and I2C are used for short distance communication protocols SPI is a serial data communication protocol that provides high reliability, compactness and error free data transmission and reception. SPI has been developed by Motorola. The entire SPI communication is divided into master to slave and slave to master communication.SPI protocol is typically used for intra chip communication. In SPI both master and slave performs dual role of transmitter and receiver. The main advantage of SPI protocol speed is very high as compared to the other protocols like I2C. SPI is a serial data communication protocol that provides high reliability, compactness and error free data transmission and reception. SPI has been developed by Motorola. The entire SPI communication is divided into master to slave and slave to master communication.SPI protocol is typically used for intra chip communication. In SPI both master and slave performs dual role of transmitter and receiver. The main advantage of SPI protocol is it's having very high speed as compared to the other protocols like I$^2$C. The Universal Verification Methodology (UVM) consists of set of base class and class library files used for the development of verification environment constructed and UVM is a reusable SystemVerilog based on Verification environment.

## II. SERIAL PERIPHERAL INTERFACE

SPI is full duplex, high speed protocol which works on shifting the content of shift register in both master and slave. The Transmission take place on synchronous clock signal generated by master. The entire SPI is a two way communication where master sends a byte of data to the slave, and slave keeps that data in its shift register. To spare the chip ports and space the ports of the SPI just takes the four lines which are responsible for communication between master and slave and SPI consists of four signals: serial clock (SCLK) output from master and slave select (SS_bar) output from master, master out slave in (MOSI) data output from master, master in slave out (MISO) data output from slave. SPI can operate with a single master device and more than one slave devices. The SPI operates in four different modes, based on the data transmitting and receiving on rising or falling edge of the serial clock.

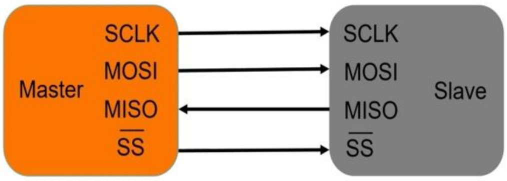


**Figure 1: Serial Peripheral Interface**

## III. SPI DATA TRANSMISSION

Clock polarity and clock phase are the important parameters,used to define a clock format to be used by the SPI bus. Clock polarity is used to determine the idle state of the clock. If idle (or first) state is 0 so the polarity is 0 and idle (or first) state is 1 so the polarity is 1.

Modes in SPI :

The data exchange is defined by two parameters, one is clock polarity (CPOL), onther one is clock phase (CPHA).Figure-2 shows the timing diagram of SPI Modes it consists of four modes.

Mode 0:

The data must be obtained before the rising edge of the first clock signal. The idle clock state is zero and MISO and MOSI lines data must be normal while the clock signal is high and can be changed when the clock signal is low. The data is captured on low to high clock transition and generated on high to low clock transition.

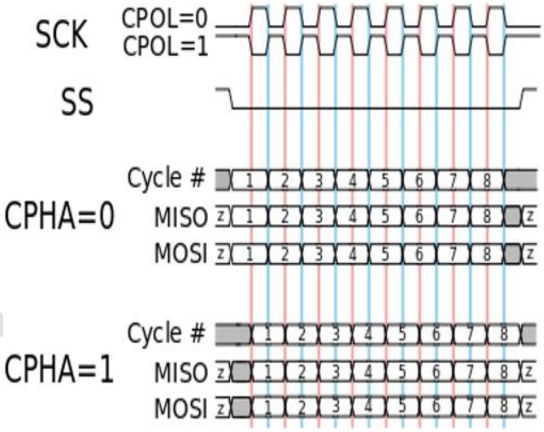


**Figure 2: Timing diagram of SPI modes**

Mode 1:

The data must be obtained before the rising edge of the first clock signal. The idle clock state is zero and MISO and MOSI lines data must be normal while the clock signal is low and can be changed when the clock signal is high. The data is captured on high to low clock signal transition and generated on low to high clock signal transition.

Mode 2:

The data must be obtained before the falling edge of the first clock signal. The idle clock state is one and MISO and MOSI lines data must be normal while the clock signal is low and can be changed when the clock signal is high. The data is captured on high to low clock signal transition and generated on low to high clock transition.

Mode 3:

The data must be obtained before the falling edge of the first clock signal. The idle clock state is one and MISO and MOSI lines data must be normal while the clock is high and can be changed when the clock signal is low. The data is captured on low to high clock signal transition and generated on high to low clock signal transition.

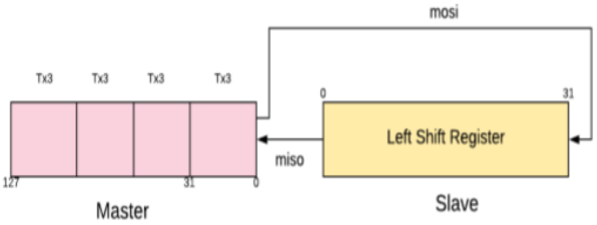


**Figure 3: Shift Register**

Figure 3 shows the shift register and it is acts as a ring buffer. Both SPI master and slave have a shift register.when the master wants to send the data to the slaves, first it loads the data into its shift register and data sent from master most significant bit to slave receives register least significant bit.

## IV.UNIVERSAL VERIFICATION METHODOLOGY(UVM)

The Universal Verification Methodology (UVM) is a set of base class SystemVerilog based on Verification environment. UVM consists of main three types of classes

- **uvm_object:** It is a base class based operational methods; instance identification fields and random seeding were in it. All uvm_component and uvm_transaction derived from uvm_object. uvm_object methods are create, clone, copy, print, record etc.
- **uvm_transaction:** It is used to generate stimulus and analysis. it is the root base class for uvm component.
- **uvm_component:** Uvm Components are quasi static objects that exit throughout simulation.uvm components are UVM Top, Test, Environment, Agent, Sequence Item, Sequence, Driver, Sequencer, Monitor, Scoreboard.

UVM provides the following different phases for all uvm components.

- **Build_phase:** This phase creates and configures additional component hierarchies if required depending on configuration and factory settings.
- **Connect_phase:** Connect ports of different UVM components to each other using TLM port and export.
- **End_of_elaboration_phase**: This phase performs display environment print topology, open files and defines additional configuration settings for components.
- **Start_of_simulation_phase:** This phase performs display environment print topology.
- **Run_phase:** Components implement behaviour that is exhibited for the entire run-time, across the various run time backward compatibility with OVM.
- **Extrat_phase:** Display final state information.
- **Check_phase**: It Checks that no unaccounted for data remain.
- **Report_phase:** This phase reports of all the simulation results.
- **Final_phase**: Close files and terminate simulation.

## V. VERIFICATION ENVIRONMENT

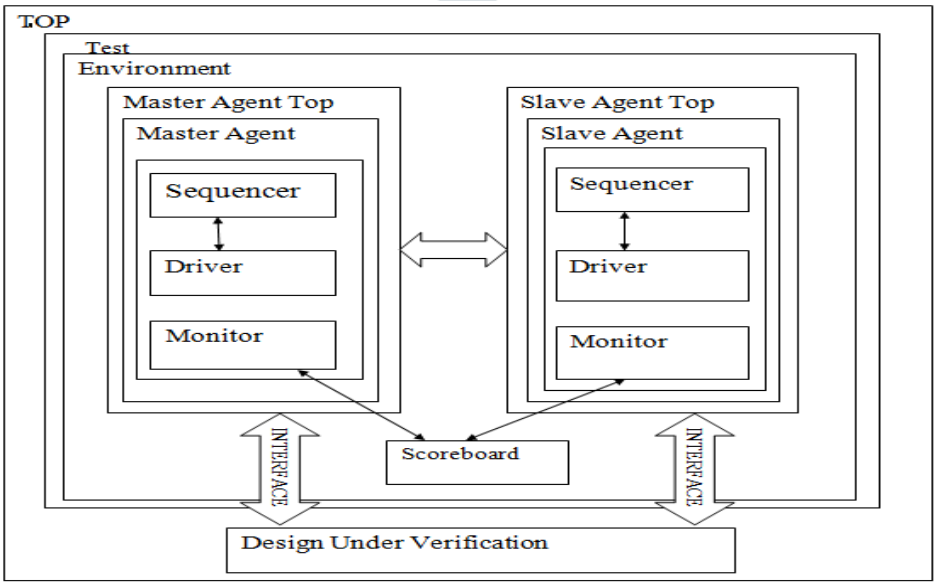


**Figure 4: UVM verification environment**

Figure 4 shows the UVM functional verification environment.
**UVM Test bench Top:** The test bench structure including the test class interfaces and DUT are instantiated in the top module.
**UVM Test:** Using test case checks and verifies specific features of a design.
**UVM Environment:** UVM environment contains multiple and reusable verification environment component and defines their default configuration as required by the application.
**UVM Agent:** UVM agent contains monitor, driver and sequencer into a single entity by instantiating connecting the components together via TLM interfaces.
**UVM Sequence:** UVM sequence is made up of several data items.
**UVM Driver:** UVM driver is used to drive signals to interface of the design.
**UVM Sequencer:** UVM sequencer generates data transactions as class objects and sends it to the driver.
**UVM Monito**r: UVM Monitor is a responsible for capturing design interface signal activity and transfer to transaction level data objects.
**UVM Scoreboard:** UVM scoreboard is a verification component it contains checkers and verifies the functionality of a design.

## VI. WISHBONE INTERFACE

The wishbone interface bus uses both master and slave interfaces. IP cores with master interface generate bus cycle transactions, IP core with slave interface receive the designed bus cycle transactions. Figure 5 shows the wishbone interface for portable and IP cores enables a design methodology for use with semiconductor IP cores. Master and slave IP cores communicate through an interconnection interface called SYSCON and it is a data flow interconnection, point to point interconnection, shared bus interconnection and crossbar interconnection.

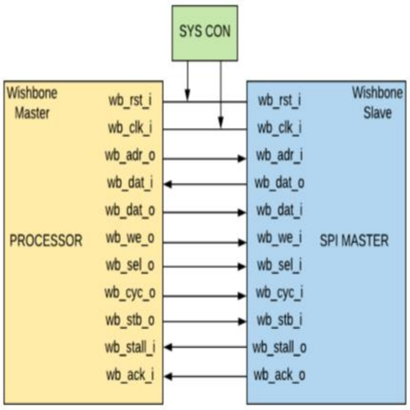


**Figure 5: Wishbone Interface**

## VII. SPI MASTER ARCHITECTURE

The SPI Master IP core design is suitable with the bus principle and SPI protocol. At the host side, the design is compatible with the slave devices of wishbone bus specification complaint. Figure 6 shows the SPI master architecture. The structure of the wishbone complaint SPI master core devices can be divided into three functional units: Serial interface, clock generator and wishbone interface. Clock generator is responsible for the clock signal generation from the external system clock. Serial data transfer module forms the data transfer core module and wishbone interface is a portable and flexible IP cores enables a design methodology for use with semiconductor IP cores.

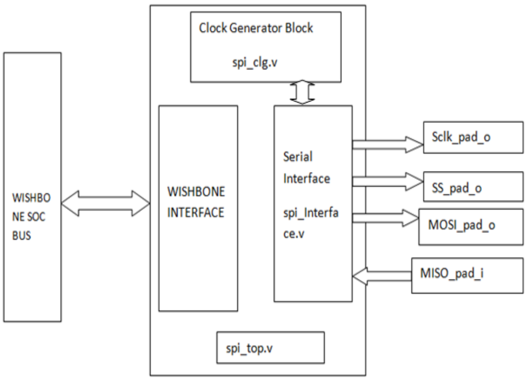


**Figure 6: SPI Master Architecture**

## VIII.MODULE DESIGN

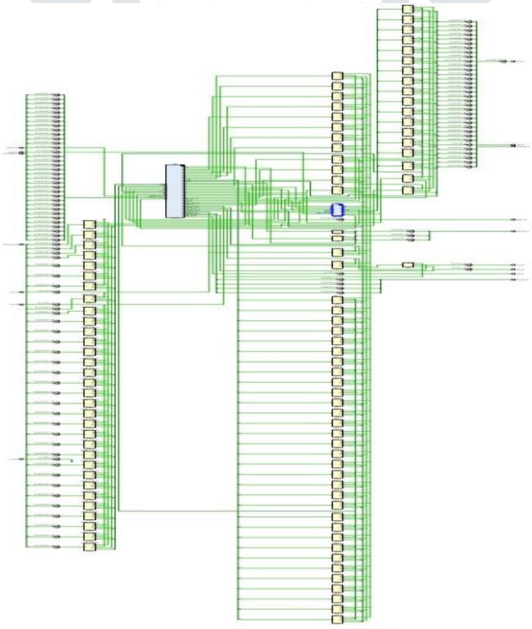


**Figure 7: Top Level Module Design**

Figure 7 shows the synthesis design module f SPI top level circuit. SPI Top module consists of sub modules: clock generation and serial peripheral interface and synthesized by ISE.

## VERIFICATION RESULTS AND DISCUSSIONS

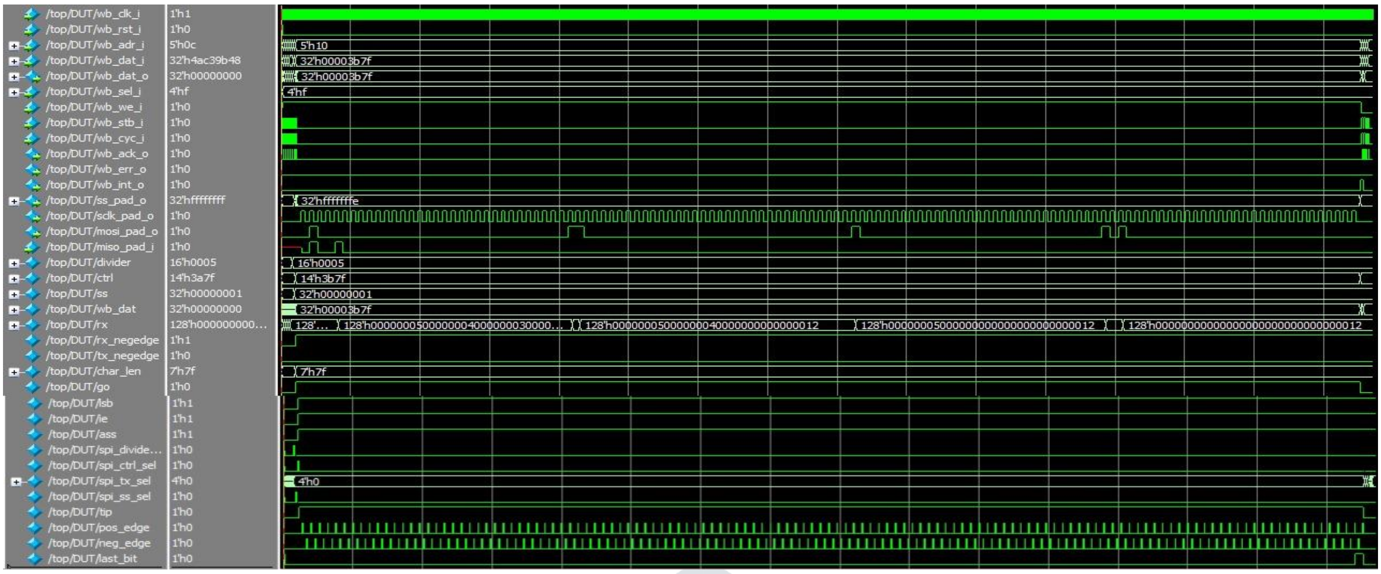


**Figure 8: SPI Master – Slave LSB communication**

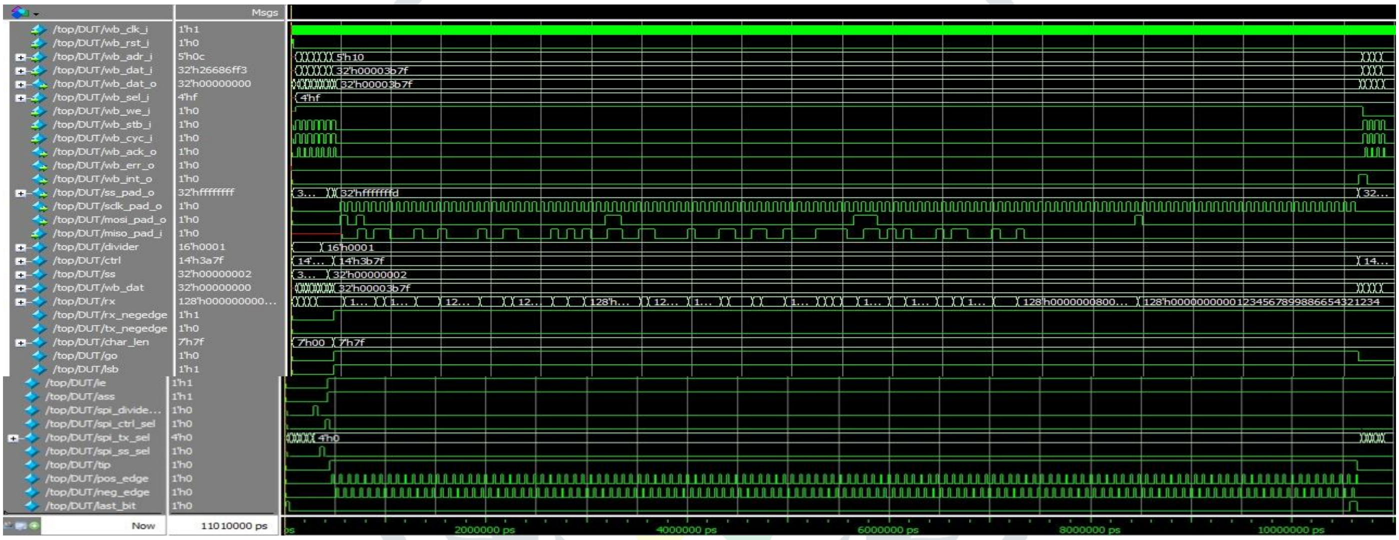
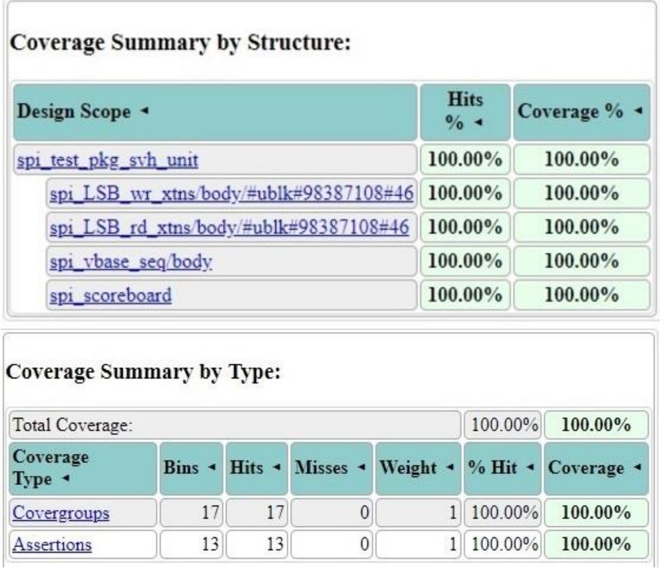


**Figure 9: SPI Master – Slave MSB communication**

Figure 8 and 9 shows the SPI Master slave LSB and MSB communication is synchronized sclk_pad clock signal, which is synchronized to the wb_clk base clock signal Before start the transfer, the master and slave configure its control register. Figure 10 shows the coverage results of SPI master core.



Coverage Summary by Structure:

| Design Scope ◂ | Hits % ◂ | Coverage % ◂ |
|---|---|---|
| spi_test_pkg_svh_unit | 100.00% | 100.00% |
| spi_LSB_wr_xtns/body/#ublk#98387108#46 | 100.00% | 100.00% |
| spi_LSB_rd_xtns/body/#ublk#98387108#46 | 100.00% | 100.00% |
| spi_vbase_seq/body | 100.00% | 100.00% |
| spi_scoreboard | 100.00% | 100.00% |

Coverage Summary by Type:

| Total Coverage: | | | | | 100.00% | 100.00% |
|---|---|---|---|---|---|---|
| Coverage Type ◂ | Bins ◂ | Hits ◂ | Misses ◂ | Weight ◂ | % Hit ◂ | Coverage ◂ |
| Covergroups | 17 | 17 | 0 | 1 | 100.00% | 100.00% |
| Assertions | 13 | 13 | 0 | 1 | 100.00% | 100.00% |

**Figure 10: Coverage Result for SPI Master Core**

## CONCLUSION

In this paper, a reusable SystemVerilog based UVM environment developed for SPI master core. The test bench enable to validate and verify the full duplex serial data transfer between the master agent and slave agent core devices for the various character length and data formats respectively and also done functional and code coverage. The verification result provides the details of the verification environment.

## ACKNOWLEDGEMENT

Centre of Excellence in VLSI design, Department of Electronics and Communication Engineering PESCE, Mandya and VLSIGuru Bangalore. First I would like to express my gratitude to my Guide Dr. Punith Kumar M B and also our HOD Dr. K.A.Radhakrishna Rao who gave me the opportunity to do this project on the topic Design and Verification of SPI master core using UVM, I would also like to thanks to my parents and friends.

## REFERENCES

[1]. Zhili Zhou, Zheng Xie, Xin'an Wang and Teng Wang "Development of verification environment for SPI master interface using SystemVerilog" ICSP,2012 IEEE 11th international Conference.

[2]. T.Liu and Y.Wang, "IP design of universal multiple devices SPI interface," in AntiConterfeiting, Security and identification (ASID), 2011 IEEE International Conference on IEEE,2011,pp. 169-172.

[3]. W.Ni and J. Zhang, "Research of reusability based on UVM verification," in 2015 IEEE 11th International Conference on ASIC, Nov 2015,pp. 1-4.

[4]. A.K. Swain and K. Mahapatra, "Design and Verification of WISHBONE bus interface for System-on-Chip integration," in india Conference (IINDICON),2010 Annual IEEE .IEEE,2010,pp. 1-4.

[5].K. Fathy and K. Salah, "An Efficient Scenario Based Testing Methodology Using UVM," in 2016 17th International Workshop on Microprocessor and SOC Test and Verification (MTV), Dec 2016, pp. 57–60.

[6]. R.Prasad and C.S.Rani, "UART IP CORE VERIFICATION BY USING UVM," IRF International Conferences, 15 2016

[7].K.Aditya, M.Sivakumar,F.Noorbasha, and P.B.Thummalakunta, "Design and Functional Verification of a SPI Master Slave Core Using System Verilog ," International Journal of Computional Engineering Research,05 2018.

[8]. S.Anantha , M.K.Kumar, and J.K. Bhandari, "Design and verification of SPI," International Journal of Engineering Development and Research(IJEDR), vol 1, pp. 130-136,Dec.2014.

[9]. Srot and Simon, SPI Master core Specification, 2004. [online]. Available: https://opencores.org /project/spi

[10]. R.Herveille, SPI Core Specification, 2003.[online]. Available: https://opencores.org/project/simple_spi

## AUTHORS PROFILE

**Dr. Punith Kumar M B** was born in Mandya, India, in 1985. He received the B.E. Degree in Electronics and Communication Engineering from The National Institute of Engineering, Mysore in 2007, and the M.Tech in VLSI Design and Embedded Systems from PES College of Engineering, Mandya under the The Visvesvaraya Technological University (VTU) , Belgaum in 2010 and Ph.D. degrees in Electronics from the University of Mysore (UoM), Mysore, India, in 2017.

In 2007, he joined the Department of Electronics and Communication Engineering as a Lecturer in JVIT, Bidadi, Bangalore. In 2010, he joined the Department of Electronics and Communication Engineering as a Assistant Professor in BGS Institute of Technology, BG Nagar, Mandya. In 2017, he joined the Department of Electronics and Communication Engineering as a Associate Professor in PES College of Engineering Mandya. His current research interests include image processing, video processing, video shot detection etc.

**Mr. Sreekantesha H N** was born in Hassan, India. Pursuing M.TECH (VLSI Design and Embedded System Design) from PESCE College of Engineering, Mandya, Karnataka. Completed BE in ECE from Government Engineering College, K R PET.