



# International Journal for Innovative Engineering and Management Research

A Peer Reviewed Open Access International Journal

www.ijiemr.org

## COPY RIGHT



**2022 IJIEMR.** Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper, all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 5th Mar 2022. Link

[:http://www.ijiemr.org/downloads.php?vol=Volume-11&issue=ISSUE-02](http://www.ijiemr.org/downloads.php?vol=Volume-11&issue=ISSUE-02)

**DOI: 10.48047/IJIEMR/V11/I03/01**

Title An Strategy and Substantiation of Wish bone I2C Protocol

Volume 11, Issue 03, Pages: 1-6

Paper Authors

**P SYAMBABU , Dr. E KRISHNA HARI**



USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per **UGC Guidelines** We Are Providing A Electronic Bar Code

## An Strategy and Substantiation of Wish bone I2C Protocol

P SYAMBABU (PG SCHOLAR), Dr. E KRISHNA HARI (PROFESSOR)

[syam475@gmail.com](mailto:syam475@gmail.com), [rrkrishna12@gmail.com](mailto:rrkrishna12@gmail.com)

HOLY MARY ISTITUTE OF TECHNOLOGY AND SCIENCE

Bogaram(V), Keesara(M), Medchal-Malkajgiri (D) Hyderabad, Telangana, 500034

**ABSTRACT:** On-chip synchronised serial connections allow for realistic on-chip communication between the processor, computerised converters, basic and complex converters, memory, and other building blocks on the chip. Serial interfaces are supported by a wide range of Integrated Circuit (IC) manufacturers. This includes the Serial Peripheral Interface (SPI), Inter-Integrated Circuits (I2C), Universal Asynchronous Receive Transmitter (UART), and Universal Serial Bus (USB), among others (USB). SPI is widely used because of its advantages over other serial interfaces, including its ease of use, low power consumption, synchronised clock, and lack of intrusion on the fast data transmission rate. The host controller is an open source PC transport called Wishbone, which enables parallel information trading for faster communication. A master-slave configuration underpins both equipment transports, making the transport interface simpler to manage.

**KEYWORDS**— *SPI (serial peripheral interface); Wishbone; Verilog HDL.*

### I. INTRODUCTION

Realistic on-chip communication between the CPU, computerised converters, basic and complicated converters, memory, and other chip building blocks is possible thanks to synchronised serial connections on the chip. Many Integrated Circuit (IC) manufacturers provide serial connections. Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), Universal Asynchronous Receive Transmitter (UART), and Universal Serial Bus (USB) are some examples (USB). Due to SPI's low power consumption, synchronised clock and absence of interference with rapid data transfer rate, it is frequently utilised in applications that need a serial interface. An open-source PC transport called Wishbone allows for parallel information swapping to speed up communication. Both equipment transfers are supported by a master-slave arrangement, making the transport interface easier to administer.

### II. SERIAL PERIPHERAL INTERFACE

Microcontrollers and peripherals may communicate with each other via the Serial Peripheral Interface (SPI) created by Motorola. Others in the industry subsequently adopted this technique. Microprocessor/microcontroller peripheral chips utilise the Serial Peripheral Interface (SPI-bus) to communicate with each other through a simple four-wire serial communications interface. SPI may link two processors even though it was originally designed to communicate between the host CPU and peripherals. Using SPI, you may use either a single-master or multi-master protocol. Most of the SPI Bus's functionality is confined to the PCB. The SPI Bus is a high-speed data transmission interface for IC circuits. According to most studies, the

interface can only transmit data in blocks of eight or sixteen bits wide, however Motorola microcontrollers enable transfers in any range between two and sixteen bits wide. It is possible to transport more than sixteen bits of data at a time via control signals because of the serial nature of the interface. An important feature of the SPI protocol is its use of four signal lines (as shown in Fig. 1).

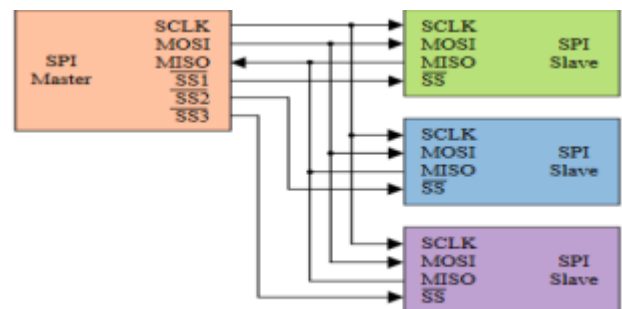


Fig 1 : SPI block diagram

- MOSI - Output data from the master to the inputs of the slaves.
- MISO - Output data from a slave to the input of the master.
- SCLK - Clock driven by the master to slaves, used to synchronize the data bits.

The master sends a choose signal to each slave, which the slaves then utilise to pick the desired one. Master sets a low SS line for the slaves to whom the data must be transferred if there are several slaves. The SPI Master is responsible for driving the SCLK line and controlling the flow of data. To choose a slave, the Slave Select (SS) line must be low. When using SPI, you can only use a single protocol to communicate with the one master. This means that just one central master initiates all communications with the slaves. After

selecting a slave to communicate with, the SPI master pulls the SS line low, which activates the clock recurrence that both master and slave may use to exchange information. Because of this, SPI is the ideal choice for implementing communication between an integrated circuit like a microcontroller and a group of peripheral devices. There's no doubt that SPI will remain a dominant technology in the future of computerised hardware frameworks.

## WISHBONE INTERFACE

The communication of the SPI master and slave devices with the processor on a chip through the Wishbone bus is shown in Fig. 2.

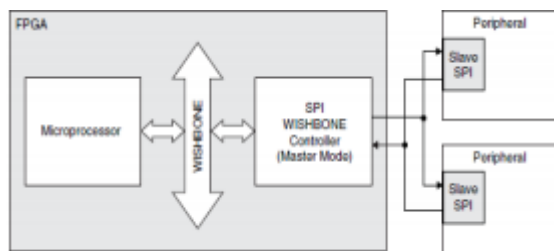


Fig 2 : Processor-SPI Interaction

Using this SPI WISHBONE controller, a microprocessor with a WISHBONE bus may communicate with an SPI device. SPI Master or SPI Slave: The controller may perform either role. Verilog HDL code parameters are used to specify Master or Slave mode. [8] There is just one module used in the design.

### A. SPI DATA TRANSFER MODES:

The peripheral devices and the microcontroller or CPU establish a full-duplex connection. Each clock cycle, the data word or character is moved out sequentially one bit at a time between master and slave. Only once a slave peripheral device is chosen by the microcontroller or CPU can the data sampling and shifting be synced by the SCLK. It's possible to set the clock phase and polarity in the SPI control register with only two bits. There are four ways to use this device. The modalities of data transport are represented in the table.3, as shown. There should be no phase or polarity differences between master and slave throughout the transfer. There are two edges of the serial clock, one for shift and the other for capture, since SPI is synchronised with the clock. Ideally, the clock phase dictates just two data transport forms.

TABLE II. APPLICATIONS OF SPI

Device	Application
Sensors	Temperature, Pressure, Touch Screens, controllers, ADCs
Control Devices	CODECs, DACs, Digital Potentiometers
Communications	Ethernet, USB, CAN, USART, IEEE 802.11, IEEE 802.15.4
Memory	Flash, EEPROM, SD card
Clocks	Real Time Clocks

TABLE III. MODES OF OPERATION

Mode	Polarity (CPOL)	Phase (CPHA)	Data Shift	Data Sample
0	0	0	Falling (Negedge)	Rising (Posedge)
1	0	1	Rising (Posedge)	Falling (Negedge)
2	1	0	Rising (Posedge)	Falling (Negedge)
3	1	1	Falling (Negedge)	Rising (Posedge)

## IV. METHODOLOGY

The design methodology of the SPI follows the below flow

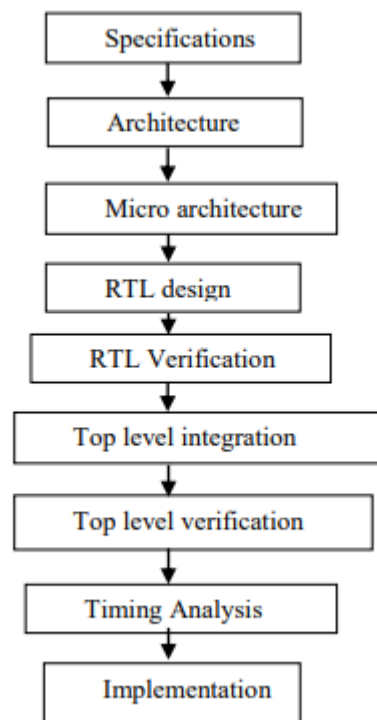


Fig 3 : Flow chart of the VLSI design

## V. ARCHITECTURE

The whole SPI architecture is shown in the diagram below. The RTL design for each of the three blocks has been developed and tested. Then comes the top-level integration and the top-level testing. After that, the integrated architecture's timing analysis is carried out.

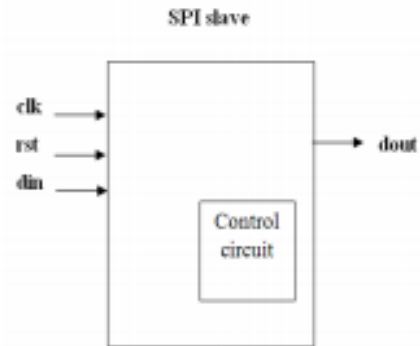
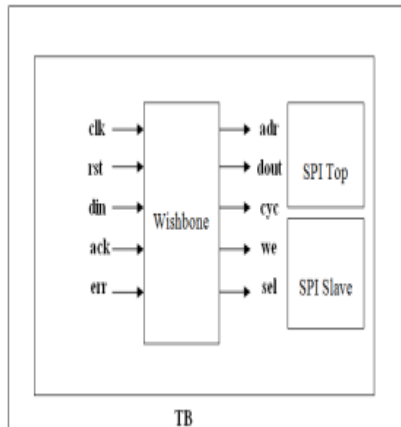


Fig 4 : SPI Slave Mode

## RTL SCHEMATIC AND SIMULATION RESULTS

The Micro-Architecture is shown below is classified into three blocks

- Wishbone
- SPI Master
- Read FIFO
- Write FIFO
- Control Circuits

Read Address Register climbs over the Write Address Register, at which time the FIFO is referred to as "empty."

As long as the read and write address LSBs match, and there are differences in the additional MSBs, the FIFO is full. There are three primary control registers in the control circuit block, which are the SPCR, the SPDCR, and the SPI Data Register (SPDR).

### WISHBONE MASTER RTL SCHEMATIC

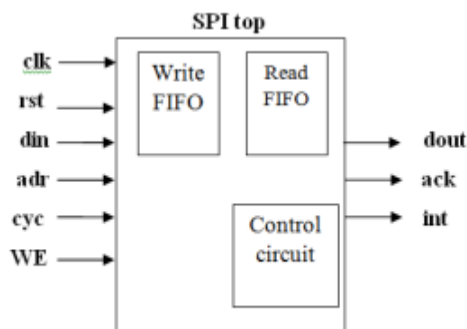
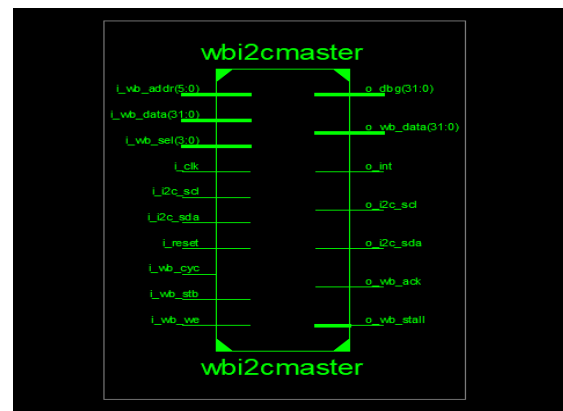
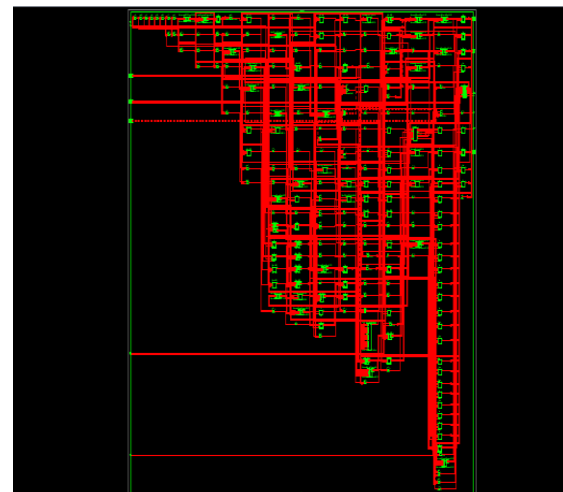
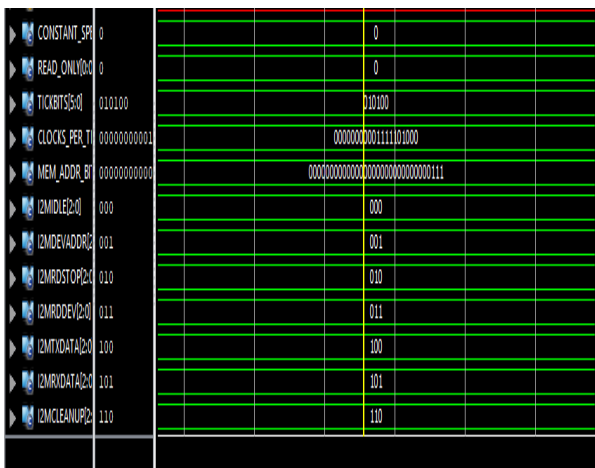
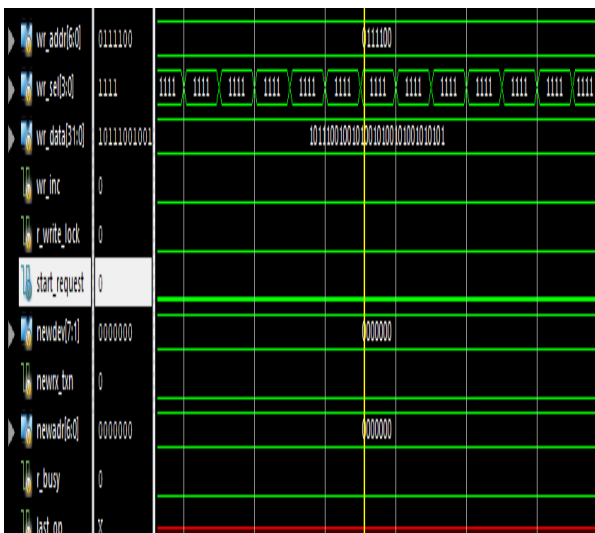
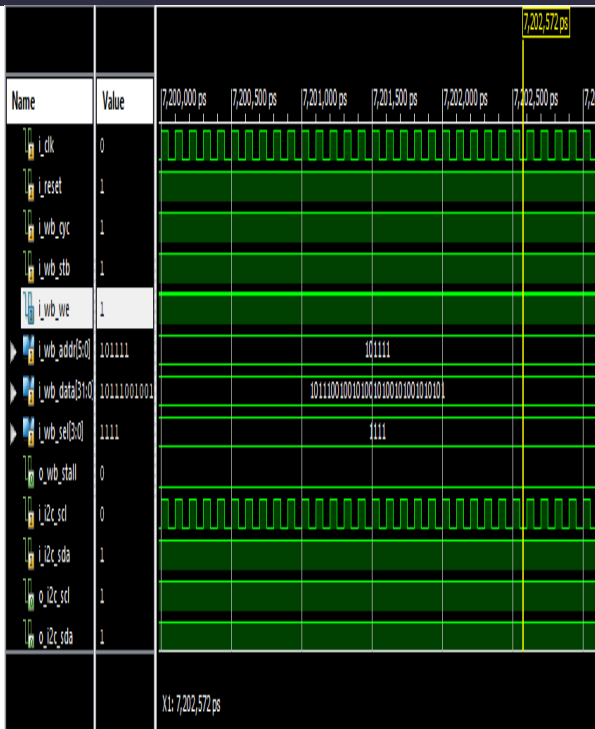


Fig 5 : SPI top module.

SPI Slave contains Control Circuits, the below figure shows the SPI slave block. It contains the control circuit block in which the control registers are there.





## AREA REPORT

Slice Logic Utilization:

Number of Slice Registers:	284	out of	93120	0%
Number of Slice LUTs:	493	out of	46560	1%
Number used as Logic:	427	out of	46560	0%
Number used as Memory:	66	out of	16720	0%
Number used as RAM:	64			
Number used as SRL:	2			

Slice Logic Distribution:

Number of LUT Flip Flop pairs used:	544			
Number with an unused Flip Flop:	260	out of	544	47%
Number with an unused LUT:	51	out of	544	9%
Number of fully used LUT-FF pairs:	233	out of	544	42%
Number of unique control sets:	19			

IO Utilization:

Number of IOs:	118			
Number of bonded IOBs:	117	out of	240	48%
IOB Flip Flops/Latches:	1			

Specific Feature Utilization:

Number of Block RAM/FIFO:	1	out of	156	0%
Number using Block RAM only:	1			
Number of BUFG/BUFGCTRL/BUFGCEs:	1	out of	104	0%

## TIMMING REPORT

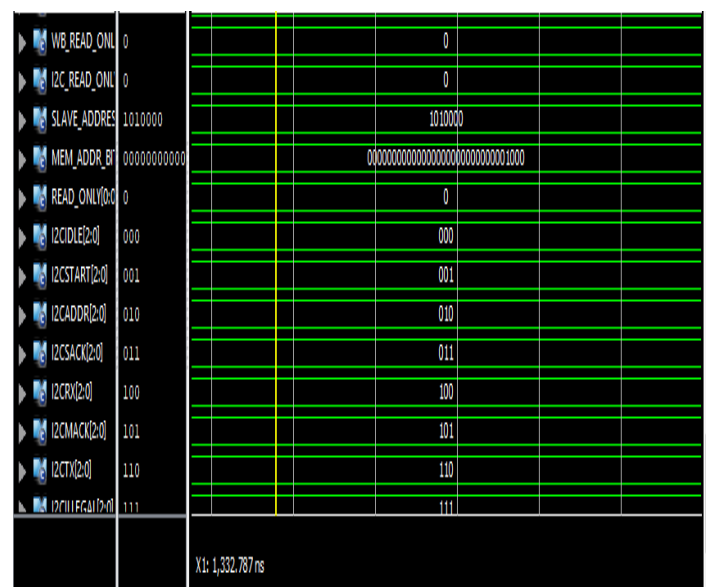
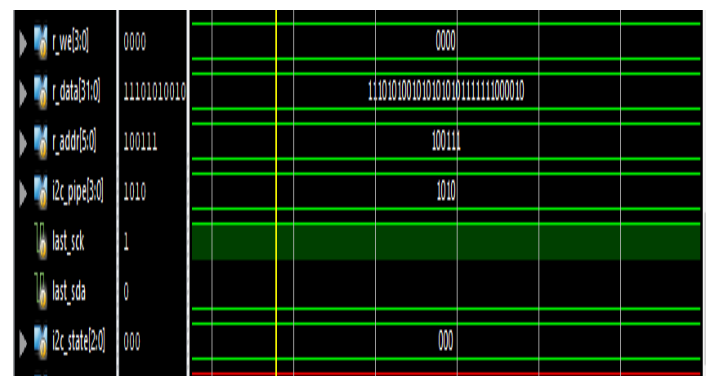
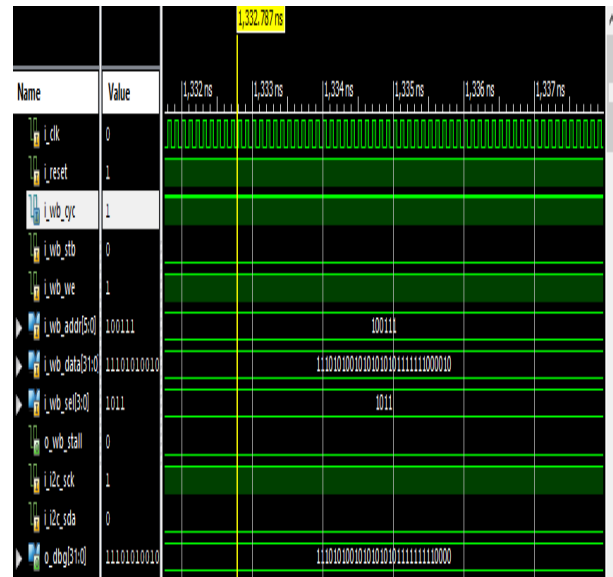
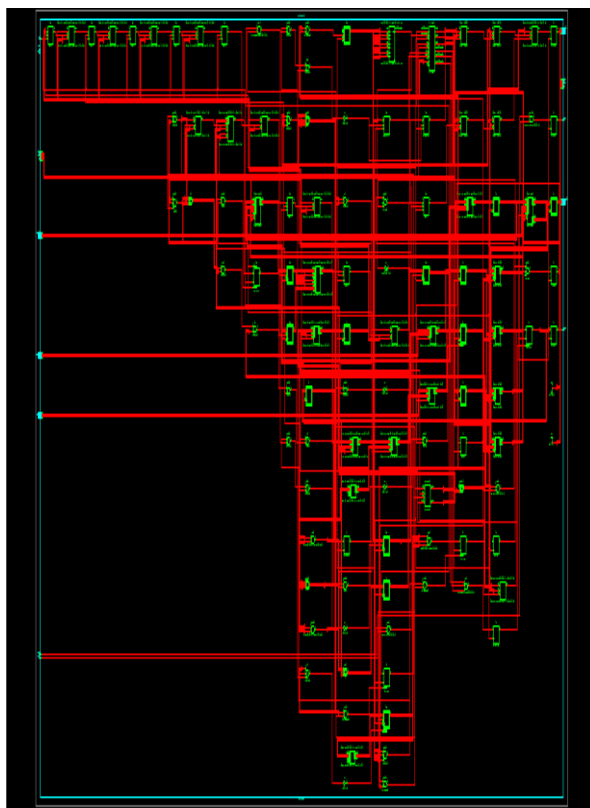
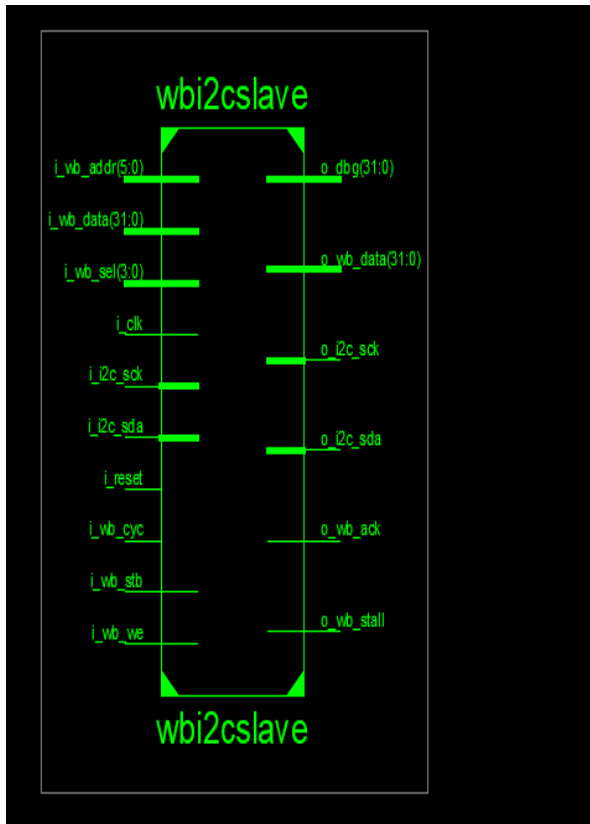
Timing Summary:

Speed Grade: -2

Minimum period: 2.605ns (Maximum Frequency: 383.855MHz)  
 Minimum input arrival time before clock: 2.031ns  
 Maximum output required time after clock: 1.133ns  
 Maximum combinational path delay: No path found

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	284	93120	0%
Number of Slice LUTs	493	46560	1%
Number of fully used LUT-FF pairs	233	544	42%
Number of bonded IOBs	117	240	48%
Number of Block RAM/FIFO	1	156	0%
Number of BUFG/BUFGCTRL/BUFGCEs	1	104	0%

## SLAVE



## Area report

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	144	93120	0%
Number of Slice LUTs	204	46560	0%
Number of fully used LUT-FF pairs	129	219	58%
Number of bonded IOBs	115	240	47%
Number of Block RAM/FIFO	1	156	0%
Number of BUFG/BUFGCTRL/BUFGCE	1	104	0%

Device utilization summary:

Selected Device : 6vcx75tff404-2

```

Slice Logic Utilization:
Number of Slice Registers:      144 out of 93120   0%
Number of Slice LUTs:         204 out of 46560   0%
  Number used as Logic:       137 out of 46560   0%
  Number used as Memory:      67 out of 16720   0%
  Number used as RAM:         64
  Number used as SRL:         3

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 219
  Number with an unused Flip Flop: 75 out of 219   34%
  Number with an unused LUT:      15 out of 219   6%
Number of fully used LUT-FF pairs: 129 out of 219   58%
Number of unique control secs:    12

IO Utilization:
Number of IOs:                 117
Number of bonded IOBs:        115 out of 240   47%
  
```

## Timing report

Timing Summary:

Speed Grade: -2

```

Minimum period: 2.411ns (Maximum Frequency: 414.740MHz)
Minimum input arrival time before clock: 1.216ns
Maximum output required time after clock: 0.791ns
Maximum combinational path delay: 0.350ns
  
```

## APPLICATIONS

- Communication protocol
- Networking applications

## ADVANTAGES

- Low area
- Efficient
- Better timing delay

## CONCLUSION

Our work focuses on the performance analysis of SPI along with the RTL Design work. Design part involves around the specifications and codes written in the Verilog. Serial peripheral interface RTL divided into three blocks i.e. Master, Slave and SPI top module. RTL code written for the

complete architecture using the wishbone interface model which open source, work done using XILINX VIVADO Design suite tool which gives simulation and synthesis validation of specification. The complete RTL has been designed for the complete architecture and design summary reports has to recorded.

## FUTURE SCOPE

The development of verification environment can be extended to the verification other Wishbone-compliant peripherals that support additional communication protocols.

## REFERENCES

- [1] A.K. Oudjida, M.L. Berrandjia, A. Liacha, R. Tiar, K. Tahraoui & Y.N. Alhoumays, "Design and Test of General-Purpose SPI Master/Slave IPs on OPB Bus," 7th International Multi-Conference on Systems, Signals and Devices, 2010.
- [2] Freescale Semiconductor Inc., "MC68HC912D60 manual V 4.0," September 2010.
- [3] "Design and Implementation of Serial Peripheral Interface Protocol Using Verilog HDL", International Journal of Engineering Development and Research, Volume 3, Issue 3, pp 2321-9939, 2015
- [4] S.Sarns and J. Woehr, "Exploring I2C," Embedded Systems Programming, vol. 4, p. 46, Sept. 1991.
- [5] A. K. Oudjida, M. L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, "FPGA implementation of I2C & SPI protocols: A comparative study," in Proc. 16th IEEE International Conference on Electronics, Circuits, and Systems, Dec. 2009, pp.507- 510.
- [6] Konstantin Mikhaylov, J. T. and Fadeev, D. "development of energy efficiency aware applications using commercial low power embedded systems". In Embedded Systems - Theory and Design Methodology, 2012, pages 407–430.
- [7] Žilvinas Nakuti, "Embedded Systems Power Consumption Measurement Methods Overview" . MATAVIMAI, 44(2):29–35, 2013.
- [8] SPI WISHBONE Controller, Reference Design RD1044, Lattice Semiconductor, March 2014.
- [9] F.Leens, "An Introduction to I2C and SPI Protocols," IEEE Instrumentation & Measurement Magazine, February 2009, pp. 8-13.
- [10] N. Anand, G. Joseph, S. S. Oommen, and R. Dhanabal, "Design and implementation of a high speed serial peripheral interface", International Conference on Advances in Electrical Engineering (ICAEE), IEEE, 2014, pp. 1-3.
- [11] K. Aditya, M. Sivakumar, F. Noorbasha, and T. P. Blessington, "Design and functional verification of a spi master slave core using system verilog", International Journal of Soft Computing and Engineering, , vol. 2, no. 2, 2012, pp. 558-563.
- [12] Manish Kundu, Abhijeet Kumar, "A Review on Low Power SPI Protocol", International Journal of VLSI System Design and Communication Systems, Vol. 02, Issue. 03, 2014, pp. 0193-019