

Design of AXI4 Slave Device Using VERILOG

¹Harish T L., ²Dr. Chandrashekhar M C

²Professor

Department of Electronics and Communication Engineering
Sri Siddhartha Institute of Technology, Tumkur-05

Abstract- Validating the essential components of the advanced extensible interface (AXI) is the major focus of the proposed system. It is required to examine all the five channels while validating the memory transactions of the AXI bus. These channels are the read address, read data, read response, write address, and write response. For the purpose of carrying out the verification process in this specific piece of work, a strategy that is built on Verification Intellectual Property cores (VIP) is used as the foundation. Read and write transactions from the same and distinct memory locations have been verified as part of the VIP design using the quantitative values of Busy Count, Valid Count, and its Bus Utilisation. This was done as part of the VIP design. This has been done both for the same memory location and for various memory locations. System Verilog is used to perform a simulation of the whole testing environment. It has been proven that read and write operations from the same memory locations as well as from other memory locations. The new AMBA AXI 4.0 standard would allow for burst durations of up to 32 beats, would update the criteria for write responses, would do away with locked transactions, and would contain information about the compatibility of various components. By using this protocol, it is possible to interface a total of 16 masters and 16 slaves at the same time. In addition to that, the work done on the design and implementation of an AMBA AXI4 Master Model for high performance SoCs that uses Verilog HDL coding is discussed in this paper. The simulation results are shown using a tool built by Xilinx, and the article concludes with a summary of the study's findings.

Keywords: Advanced extensible interface (AXI); Verification Intellectual Property cores (VIP); Advanced Microcontroller Bus Architecture (AMBA); system-on-chip (SoC); Advanced Peripheral Bus (APB); Advanced RISC Machines (ARM);

1. INTRODUCTION

The firm ARM (Advanced RISC Machines) is responsible for the development of the Advanced Microcontroller Bus Architecture (AMBA), of which the AXI is a component. It is a communication protocol that takes place on the chip itself. The AMBA AXI protocol lends its assistance to the development of high-performance and high-frequency system architectures. AMBA is an open protocol that is used as a standard; it is a specification for on-chip interconnects that governs the connecting and administration of functional blocks in a system-on-chip (SoC). AMBA was developed by Advanced Microcontroller Inc. The AMBA bus may be implemented on small size SoCs with relative ease. As a result, due to the AMBA bus's high level of efficiency, it has been chosen to serve as the market representative for the SOC[6]. Within the framework of the AMBA specification are defined three separate buses:

1. Advanced Peripheral Bus (APB).
2. Advanced High-performance Bus (AHB).
3. Advanced extensible Interface Bus (AXI).

Since its introduction, the scope of AMBA has expanded to include much more than just microcontroller devices. It is now employed on a broad variety of ASIC and SoC components, including application processors used in contemporary portable mobile devices such as smartphones. These components may be found on a wide range of ASIC and SoC components[9]. This practise is widespread in the industry at the present time.

When developing the AMBA specification, it is important to take into consideration the following points:

- **Technology independence:** AMBA is an on-chip protocol that is not dependant on any specific technology. The clock cycle level is the sole aspect of the bus protocol that is detailed in the specification.
- **Electrical characteristics:** Within the AMBA specification, there is no information that pertains to the electrical characteristics. This is due to the fact that the information pertaining to the electrical properties will be completely reliant on the manufacturing process technology.
- **Timing Specification:** At the cycle level, the behaviour of a number of different signals is specified by the AMBA protocol. The precise timing requirements will be determined by the kind of process technology that is employed and the number of times the operation[12] is performed.

Since the AMBA protocol does not specify any timing requirements, the system integrator has complete freedom in allocating the available signal timing budget among the many modules that are linked to the bus. This is because the AMBA protocol does not define the precise timing requirements. AXI makes use of master and slave interfaces that are clearly defined and interact with one another through five distinct channels:

- Read address
- Read data
- Write address
- Write data
- Write response

The five AXI channels are seen in Figure 1.

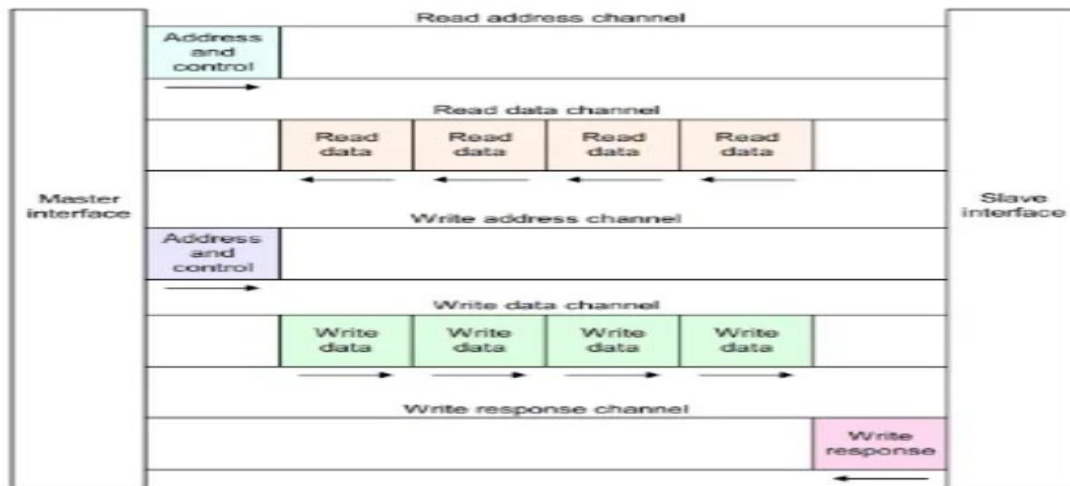


Figure 1: AXI Channels

Only those five channels are required in order to establish a connection between a single master and a single slave. A new strategy is necessary in order to link many masters and/or numerous slaves simultaneously. The following outline constitutes the structure of this paper: In the next section, "Section 2," we will talk about the Related Work. In the third section, we will talk about the activity that has been suggested. Section 4 addresses Simulation Result. Section 5 of the document is where the findings are laid forth for the reader.

2 RELATED WORK

This part provides further information on the extensive analysis of various image improvement algorithms on hardware and software techniques, as well as the discovering and discussion of hardware approaches towards MPSOC via the use of image processing programmes. Using a machine learning technique for training and testing the demographic picture dataset that sits in reconfigurable MPSoC architecture, Dutta et al. [1] present a smart frame design for which they have developed a prototype. This approach was used in order to build the smart frame. The execution time of the FPGA platform is lowered as a result of this technology, and its throughput is increased. According to Wang et al. [2], the FPGA-based MPSoC combines service-oriented architecture on a single chip in order to handle difficulties with programmable interfaces, software chains, and a dedicated module for multiple applications. This was done in order to solve problems with a separate module for different uses. These integrated modules provide modularity, flexibility, and the ability to be modified, but the hardware platform has limitations that prohibit them from being scaled. This prevents the modules from providing a modular solution to the problem. Karim et al. [3] describe an MPSOC-based module for ECG applications on FPGA. The ECG data is processed in software and then shown on the FPGA's input-output switches. This module includes a master and slave processor module.

The hardware design for the image contrast enhancement method is described in detail by Huang et al. [4]. This approach incorporates Gamma correction and luminance modification after statistical computation and cumulative distribution functions. However, the algorithm did not succeed in reducing the computational complexity of the hardware while it did improve image quality. Singh et al. [5] developed a fast technique for picture improvement that makes use of parallel architecture. This approach has a polynomial-based fractional order filter function, and the end result is a better-enhanced image for a different order. Because of the intricacy of the architecture, it cannot be used for any kind of hardware-based modelling. The article by one method of improving satellite images, Pugazhenthil et al. [6] provide automatic multi-histogram equalisation through MATLAB programming. This technique reduces the brightness while simultaneously enhancing the contrast. The findings of the analysis of the picture quality are not satisfactory and are unsuitable for hardware-based techniques. Image contrast was improved by Mahajan et al. [7] by using a Gaussian mixture model (GMM) and a genetic algorithm. Modelling, an expectation-maximization approach, partitioning, and mapping are all components of the GMM, which is used to improve the picture quality of low contrast images by the use of the GMM. Li et al.'s [10] de-noising on low-light images, and Tian et al.'s [11] surface roughness detection system, are extensively evaluated, with improvements to performance metrics. These techniques were developed by Li et al.

3. PROPOSED WORK

Verilog is going to be used to accomplish communication between one master and one slave in this project, and then System Verilog is going to be used to validate the design.

3.1 Design of AXI Protocol

The operating frequency of an AMBA AXI4 slave is intended to be 100 MHz, giving each clock cycle a length of 10 nanoseconds, and it is capable of supporting a maximum of 256 data transfers during a single burst. As can be seen in Figure 2, the AMBA AXI4 system has a master and a slave in its most basic form. There are five distinct channels that may be used to communicate between the AXI master and the AXI slave. These channels are referred to as the read address channel, the write response channel, the read data channel, and the write data channel.

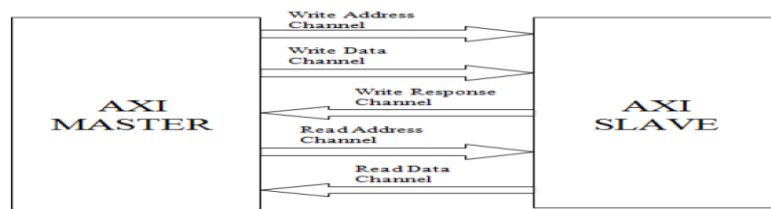


Figure 2: Block Diagram of a system

Every single transmission in the AXI protocol is completed via the handshake process. When it comes to transferring control and data information, every channel employs the same VALID/READY handshake. Both the master and the slave have the ability to change the pace at which data and control information is sent thanks to this technology that permits flow to be controlled in both ways. The VALID signal will be sent out by the source to indicate that it has reached this state once either the data or the control information may be accessed. The READY signal, which indicates that the destination is prepared to accept the data or control information, is produced by the destination. This signal indicates that the destination is ready to receive the data or control information. The VALID and READY signals must both be high in order for the transfer operation to take place. There can be no combinatorial routes between the input signals and the output signals on either the master or the slave interface. This rule applies to both the master and the slave interface.

3.2 Address Write Channel (AW Channel)

Only when ARESETn is HIGH does AXI_MASTER drive the write command signals; in all other circumstances, it drives all signals as zero. Signals AWID, AWADDR, AWBURST, AWLEN, AWSIZE, AWCACHE, AWLOCK, and AWPROT are all address write command signals that must be driven by the AXI_MASTER, with AWVALID set to HIGH to indicate that the driven signals are legitimate. Before the AXI_MASTER drives the AWVALID signal as LOW, it waits for the AWREADY signal, which is driven by the DESTINATION_SLAVE and indicates that it has received the address write instruction signals. This is because it has received the address write instruction signals, as shown by the AWREADY signal. AXI_MASTER will keep its current settings if AWREADY is set to LOW. Figure 3 depicts the states of the address write command signals.

3.3 Write Data Channel (W Channel)

After providing the write address instruction signals, the AXI MASTER is the one responsible for driving these Write Data signals. When ARESETn is HIGH does it cause these signals to be driven; otherwise, it causes all signals to be driven to zero. The WDATA signal is driven by the AXI MASTER with the WVALID bit set to HIGH. When WREADY is HIGH, it causes the subsequent WDATA to be driven. The data for the AWLEN No. is driven by the AXI MASTER. During the process of driving the most recent data, it sets the WLAST to the HIGH position. The state diagram for the WRITE DATA channels may be seen in Figure 4.

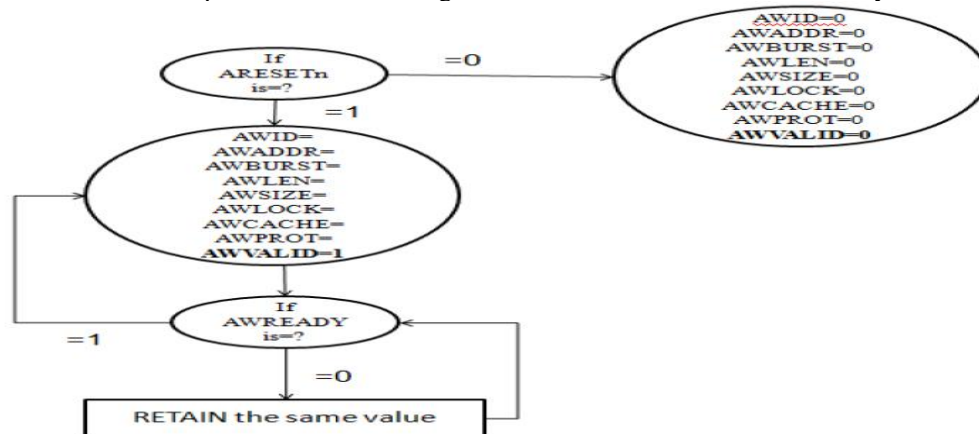


Figure 3: State diagram of Address Write Channel

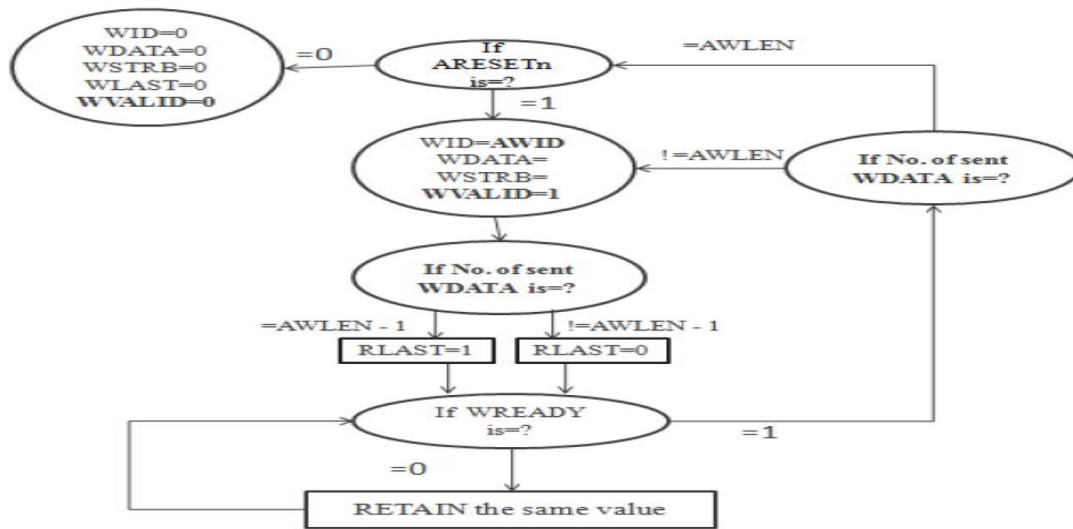


Figure 4: State diagram of Write Data Channel

3.4 Write Response Channel (B Channel)

Only when ARESETn is HIGH does the DESTINATION_SLAVE cause these Write Response signals to be driven; in all other cases, it causes all signals to be driven as zero. DESTINATION_SLAVE is now holding out for the WLAST signal. As soon as it receives the WLAST signal, it starts driving the response signals while keeping BVALID at the HIGH state. It holds steady for certain time, at which point it takes action. At the next positive edge of ACLK, all signals will be reset to zero if BREADY is HIGH; otherwise, they will retain their current values.

3.5 Address Read Channel (AR Channel)

The command signals are driven by AXI_MASTER only when ARESETn is HIGH. When ARVALID has a value of HIGH, it indicates that the driven signals are valid. Before driving the ARVALID signal as LOW, the AXI_MASTER waits until it gets the ARREADY signal. The ARREADY signal is produced by the AXI_MASTER. If ARREADY is LOW, then the values of AXI_MASTER will remain the same.

3.6 Read Data Channel (R Channel)

After receiving the read command signals, it is the SOURCE_SLAVE's job to drive these Read Data signals. These signals are driven only when ARESETn is HIGH; otherwise, they are driven with a value of zero. When the RDATA signal is driven by SOURCE_SLAVE with RVALID set to HIGH, it maintains its previous value until it gets the RREADY signal. If RREADY has a HIGH value, it will drive the subsequent RDATA value. The ARLEN Number of data is driven by the SOURCE_SLAVE. During the process of driving the final data, it sets the RLAST to the HIGH state.

4. SIMULATION RESULT

The verification step of the VLSI design process is highly significant. It is accomplished by constructing test benches, often known as BFM's. Verilog codes known as test benches are used during simulation to impose data onto the different RTL components. Software named Mentor Graphics QuestaSim is used to do the simulation. Then, the simulated waveforms are examined by applying different levels of force.

The Verilog code that is produced is run through the Verilog code simulator, also known as VCS. The name convention for the inputs to the module is internal_awaddr, internal_awdata... etc., while the naming convention for the output parameters is AWADDR, AWDATA etc.

Validating signals like WVALID for data and AWVALID for address are sent after the input parameters have been processed without error. When the correct address and data are received, the Etc signal will go high to indicate this. Following successful completion of the write and read operations, the WLAST and RLAST signals will rise, respectively. The test case is carried out for a number of different processes, and the waveforms may be seen in DVE.

4.1 Write address operation:

The values 23, 30, 20, 50, and 70 are supplied as the input parameter for internal_awaddress, and the resulting output is AWADDR- 23, 30, 20, 50, and 70. And the result is AWID-11, 9, 5, 3, 7, but the internal_awid values are 11, 9, 5, 3, 7. As can be seen in Figure 5, the AWVALID, AWREADY signal gets high whenever it successfully receives the right address.

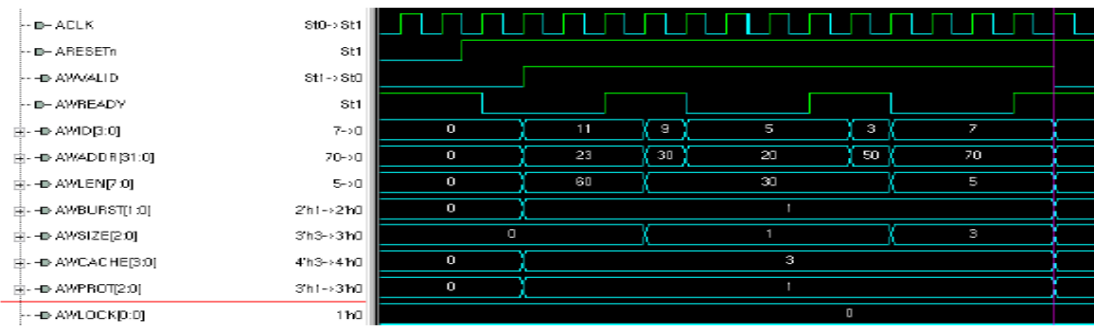


Figure 5: Write Address Waveform

4.2. Write data operation:

The frequency of the clock is now set at 100 MHz. This time around, the input is the value of internal_wdata, which is 24, 25, 26, 27,..44, 45, and the output WDATA value receives the same data. The completion of the write operation is indicated by the WLAST signal becoming high. A successful write response is obtained when the BVALID signal goes from low to high, which happens after the write data operation is complete. Figure 6 depicts an instance of a single-transaction write data operation.

4.3. Read address operation:

The values 30, 20, 50, and 70 are supplied as the input parameter for internal_aaddress, and the corresponding values are produced as ARADDR- 30, 20, 50, and 70. and the internal_arid-9, 5, 3, 7, in addition to the ARID-9, 5, 3, 7. Figure 7 shows that when a valid address is received, the ARVALID, ARREADY signal becomes high.

5. CONCLUSION

The code coverage mode analysis is used in order to do verification and analysis on the AXI protocol verification as well as the signals that are utilised in each channel. The most major advantage of adopting this kind of verification is the use of the pseudo random coverage driven verification, which results in a reduced time to market and is suited for the verification of intricate designs applying system Verilog. This form of verification may also be used to verify software. In the future, we are going to construct a test case in order to validate the read phase and the write phase concurrently from the same location. This will be done using the same location. In addition, we are going to design a test case to validate reading and writing from two distinct places.

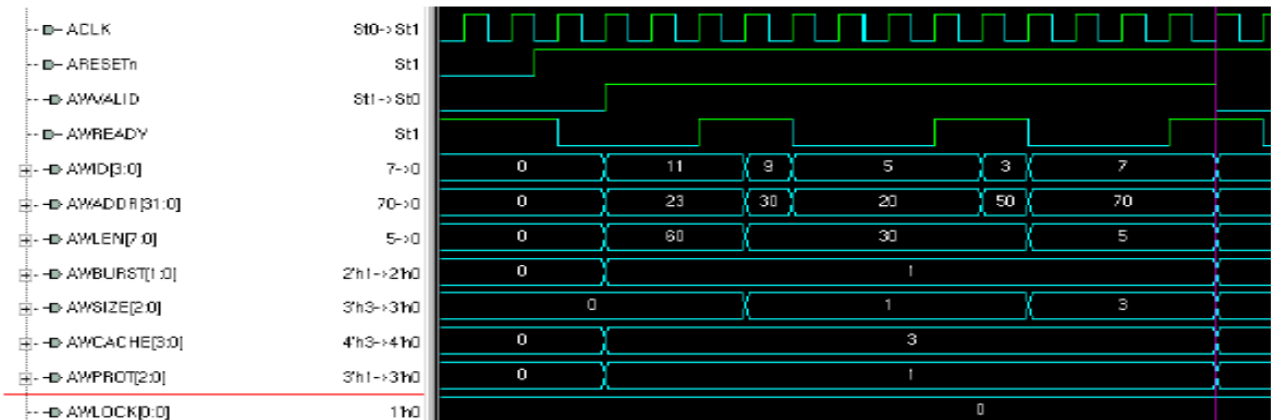


Figure 6: Write Data Waveform

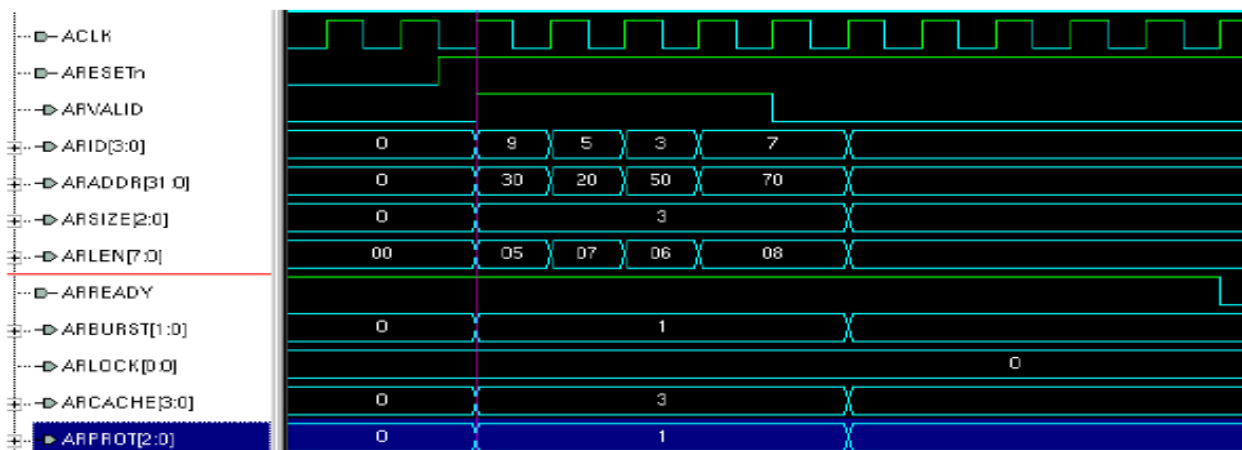


Figure 7: Read Address Waveform

REFERENCES:

1. Dutta, Anandi, and Magdy Bayoumi. "Introducing a Novel Smart Design Framework for a Reconfigurable Multi-Processor Systems-on-Chip (MPSoC) Architecture." In 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1-3. IEEE, 2016.
2. N Ashok Kumar, G Shyni, Geno Peter, Albert Alexander Stonier, Vivekananda Ganji, "Architecture of Network-on-Chip (NoC) for Secure Data Routing Using 4-H Function of Improved TACIT Security Algorithm" Wireless Communications and Mobile Computing Hindawi, 9 pages, March 2022 Wang, Chao, Xi Li, Yunji Chen, Youhui Zhang, Oliver Diessel, and Xuehai Zhou. "Service-oriented Architecture on FPGA-based MPSoC." IEEE Transactions on Parallel and Distributed Systems 28, no. 10 (2017): 2993-3006.
3. Karim, Mohammed, and Mohamed-Yassine Amarouch. "An FPGA-based MPSoC for real-time ECG analysis." In 2015 Third World Conference on Complex Systems (WCCS), pp. 1-4. IEEE, 2015.
4. Huang, Shih-Chia, and Wen-Chieh Chen. "A new hardware-efficient algorithm and reconfigurable architecture for image contrast enhancement." IEEE Transactions on Image Processing 23, no. 10 (2014): 4426-4437
5. Singh, Koushendra Kumar, Durgesh Kumar, Shubham Chauhan, and Manish Kumar Bajpai. "Parallel architecture based fast algorithm for image enhancement." In 2015 IEEE Bombay Section Symposium (IBSS), pp. 1-6. IEEE, 2015.
6. Ashokkumar, N., and A. Kavitha. "Network on Chip: A Framework for Routing in System on Chip." Journal of Computational and Theoretical Nanoscience 12, no. 12 (2015): 6077-6083.
7. Pugazhenti, A., and L. S. Kumar. "Image contrast enhancement by automatic multi-histogram equalization for satellite images." In 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-4. IEEE, 2017.
8. Mahajan, Arushi, and Divya Gupta. "Image contrast enhancement using Gaussian Mixture model and genetic algorithm." In 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), pp. 979-983. IEEE, 2017.
9. N. Ashokkumar, A. Kavitha, "A NOVEL 3D NoC Scheme for high throughput UNICAST and MULTICAST Routing Protocols" Technical Gazette, Vol 23, No.1, Feb 2016, pp 215-219 ISSN 1330-3651 (Print) ISSN 1848-6339 (Online), UDC 62(05)=163.42=111, Tehn. vjesn. Impact Factor : 0.57.
10. Li, Lin, Ronggang Wang, Wenmin Wang, and Wen Gao. "A low-light image enhancement method for both denoising and contrast enlarging." In 2015 IEEE International Conference on Image Processing (ICIP), pp. 3730-3734. IEEE, 2015.
11. Tian, Jie, and Xijie Yin. "Adaptive image enhancement algorithm based on the model of surface roughness detection system." EURASIP Journal on Image and Video Processing 2018, no. 1 (2018): 103.
12. N. Ashok Kumar, Nagarajan, P. Venkataramana, "Design Challenges for 3 Dimensional Network-on-Chip (NoC)" Lecture notes on Data Engineering and Communication Technologies, Springer book series (LNDECT), Volume 39, 773-782, 2020.