© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Design and Analysis of a Novel Low Complexity and Low Power Ping Lock Arbiter by using EGDI based CMOS Technique

Sangeeta Singh[1*], JVR Ravindra[2], B Rajendra Naik[3]

[1]Research Scholar, Dept. of ECE, JNTUK, Kakinada, Andhra Pradesh, INDIA

[1]Dept. of ECE, C-ACRL, Vardhaman College of Engineering, Hyderabad, Telangana, INDIA

[2]Dept. of ECE, C-ACRL, Vardhaman College of Engineering, Hyderabad, Telangana, INDIA

[3]Dept. of ECE, University College of Engineering, Osmania University, Hyderabad, Telangana, INDIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2022.14.01.034
Received 28 June 2021; Accepted 05 October 2021; Available online 07 March 2022

Network-on-chip (NoC) provides solution to overcome the complications of the on-chip interconnect architecture in multi-core systems. It mainly consists of router, links and network interface. An essential component of on-chip router is an arbiter that significantly impacts the performance of the router. The arbiter should provide fast and fair arbitration when it is placed in Critical Path Delay (CPD) systems. The main aim of this research work is to design a novel arbiter for an effective network scheduler in complex real time applications. At the same time resource allocation and power consumption should be very low. Previously, a novel gate level Ping Lock Arbiter (PLA) is designed to overcome the limited fair arbitration in Improved Ping Pong Arbiter (IPPA) with less delay. But the chip size and power consumption are very high. To overcome this problem, an Effective Gate Diffusion Input (EGDI) logic-based CMOS scheme is used to design a novel Compact Ping Lock Arbiter (CPLA). The proposed CPLA is compared with the existing PLA based on static CMOS scheme. The comparison between the conventional and proposed arbiter is carried out to analyze the area, delay and power by using Tanner Tool 14.1 with 250nm and 45nm technology. The proposed CPLA is suitable for compact and smart applications with glitch free and fair arbitration. The proposed CPLA presents 41.2% reduction in total area consumption. It also provides 47.12% average Area and Power Product (APP) reduction when compared to the existing 4-bit PLA. Similarly, the proposed 8-bit CPLA shows 49.48% average APP product shrinkage when compared to conventional 8-bit PLA. For low power and area constraint applications, the proposed CPLA with EGDI is best in comparison to the static CMOS based ping lock round robin arbiter. Therefore, the results demonstrate that the proposed CPLA achieves low power and consumes less area than the existing ping lock arbiter.

Keywords: Network on chip, ping lock arbiter, effective gate diffusion input logic, CMOS Logic.

1. Introduction

The advancements in VLSI technology have allowed millions of transistors on a single die leading to system-on-chip designs. The major challenge to overcome in such devices is to offer for functionally correct, reliable operation of the interacting components [1]. In addition, with continuous decrease in feature size and increasing integration density, the global interconnect delay now dominates gate delays [2]. Interconnections have therefore been the dominant factor in determining the overall characteristics of a chip. Global interconnects cause severe on-chip synchronization errors, difficult timing closures, routing congestion and high power consumption [3].

*Corresponding author: sangeeta.singh@ieee.org

The communication network connects diverse functional blocks on a chip to form homogeneous, heterogeneous or a hybrid system [4]. This network connecting different functional blocks together is the back bone of any system. The amount of power and area associated with these interconnects keeps increasing with each scaling and thus the communication architecture has to be modified to achieve low area and high performance. Traditionally, the designs of systems were based on critical paths and clock trees which presented an increased amount of delay and power consumption. The emerging technology that targets such connections is called on-chip interconnection network, also known as Network on Chip [5]. The integration of massive number of storage blocks and processing elements (PEs) are done in a single chip is called as NoC. Network technology uses the NoC feature to create the data exchange within the chip. A typical NoC architecture contains Routers (Switches), Links, Network Interface (NI) and Processing Elements (PE) as shown in Fig.1.

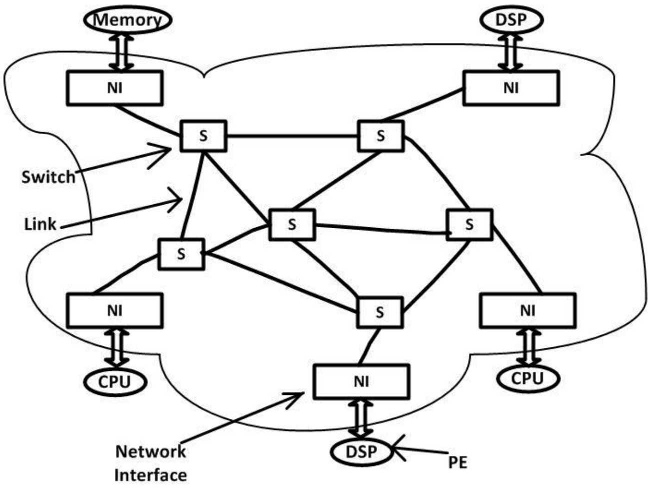


Fig. 1 - Basic NoC Architecture

Routers plays an imperative role in NoC. Routers handle the data transmission via network as packets [6]. Thus, to achieve high throughput, structural design of the router should experience lower delay with less area and power consumption [7].

Arbiter is a main building block of NoC router. An arbiter is used to determine the sequence of granting access to the shared resources when multiple input ports try to access the same output port. Different types of arbitration scheme available are round robin, lottery, token ring [24], and fixed priority [30]. Among all, round robin token scheme is the standard scheme which is used in Network on Chip for bus arbitration. The advantages associated with round robin arbitration scheme are low complexity, fairness in scheduling, prediction of worst-case wait time. Most of the designs used for arbiters are based on achieving the performance and scalability issues. But due to advances in technology, power consumption tends to dominate the design.

There are different types of Round Robin Arbiter (RRA) available to solve the conflicts among multiple input ports requesting for single output port. The names of few RRA based designs are: Programmable Priority Encoder Arbiter (PPE), Ping Pong Arbiter (PPA), Thermometer Coded Arbiter (TCA), Merged Arbiter and Multiplexer (MARX) [28], Switch Arbiter etc. Among all the round robin arbiters available, PPA provides binary tree-based structure to build higher order and complex arbiters [8].

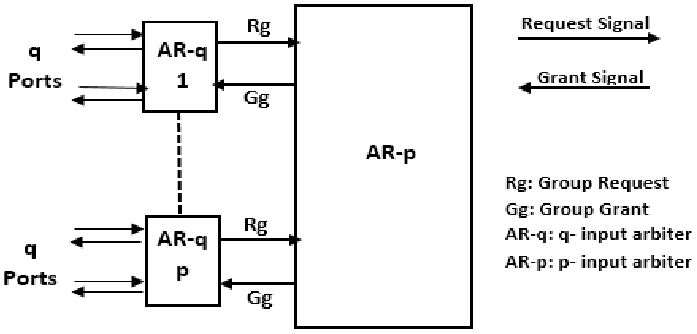


Fig. 2 - Tree Structure of PPA

The Ping Pong Arbiter (PPA) suffers from the drawback of unfair arbitration when the input requests are not uniformly distributed. Thus, a gate level Ping Lock Arbiter (PLA) was developed to overcome the limited fair arbitration in Improved Ping Pong Arbiter (IPPA) with less delay [9]. But this leads to increase in chip size and power consumption. This research work aims to design a low area, low power and high-speed Ping Lock Arbiter (PLA) for NoC router. The PLA is based on Ping Pong Arbiter (PPA) which is designed by connecting small arbiters in binary tree format [8]. The four-bit ping pong arbiter is developed by using a three 2-bit PPA with some additional gates. In addition, the internal feedback flag signal "fi" is given as the input through the feedback process. Mostly, the 2-bit Ping Pong Arbiter acquires a grant signal only if the next level subsequently provides a suitable grant signal in 2-bit PPA. Thus, the corresponding arbiter in each layer gets grant signals which are masked with grant signal of parent arbiter. The total arbitration time required is very low in PPA but it suffers from drawback of very high resource utilization.

In the proposed system, the Ping Lock Arbiter (PLA) can be used when greater than one request comes towards the particular port. The ports are arranged in the order of East, West, North, South and Local. It is based on round robin priority. This paper presents two types of ping lock arbiters are designed by using static CMOS technique and gate diffusion input logic (GDI) based CMOS technique [10,11]. Tanner tool is used to design the circuit and check the functionality of the arbiter for NoC router. The comparison between existing and the proposed arbiter is analyzed to check the resource utilization, performance and power consumption. In future, these arbiters can be incorporated into the Application Specific Network on Chip router (ASNoC).

The rest of the paper is arranged as follows: Section 2 describes various works related to arbiter. Section 3 presents the details of implementation of ping lock arbiter based on static CMOS technique. The design of proposed ping lock arbiter based on EGDI is demonstrated in section 4. The results and corresponding analysis is shown in section 5 and conclusion is given in section 6.

2. Related Work

The round robin arbiter is worked based on regular bus arbitration method. A token is used to attain an efficient allocation of resource in equal manner. An individual token is used for each and every user to create a grant signal at a meticulous time. At a time only one user will be receiving the grant signal from numerous requests in a cyclic manner. Currently, a lot of efficient resource allocation method are used for on chip communication. The researchers have concentrated on implementation and analysis of different types of arbiters available but to the best of our knowledge round robin arbitration is the best suitable method for many real time applications as it is less complex and provide fairness in scheduling for uniform traffic conditions. This section provides insights on various work done related to the implementation of arbiter.

In [12], the resource allocation policy is attained by using a fast crossbar scheduler. The multiplexing and de-multiplexing processes are performed in the crossbar switch of router. At a time more than one request is received from various input channel. However only one input channel which having highest priority is awarded a grant signal from a queue. Subsequently all other input channel obtains a grant signal in a regular interval of a clock cycle. In addition, the scheduler speed is improved by using pipelined scheme. Anyhow power consumption and area are high. In [13], the round robin arbiter with Programmable Priority Encoder (PPE) is used to set a routing policy and it leads to huge amount of gate delay. To overcome this problem, Parallel Round Robin Arbiter (PRRA) with binary tree structure is included for high performance and low complexity. In PRRA, the gate delay is lower than the crossbar scheduler but the power and resource utilization are very high.

Previously, a Dual Round Robin Arbiter (DRRA) is designed to split the transaction bus in SoC implementation [14]. The data bus arbiter and address bus arbiter are used to construct a non-multiplexed transaction bus. Every request is processed after certain bounded waiting time which is created by DRRA. In [15], the proposed circulated arbiter is created by using a circulated scheduler-based crossbar switch in order to analyze the result of router. The mask trail and compressed scheduler are used to create a fair crossbar control. The scheduling delay, critical delay, hardware cost and power consumption are decreased in the fair crossbar switch. In addition, they can predict the fairness of an arbiter. Currently, few other variations of arbiter are developed to improve the fairness than the existing arbiter.

In [16], a novel bit level algorithm and concurrent prefix adder logic are used to design a novel programmable priority arbiter. Consequently, it increases the complexity of the overall design. Furthermore, performance is high in novel programmable priority arbiter. In [17], the Edge Detector (ED) and two Fixed Priority Encoder (FPE) are used to construct parallel prefix arbiter based on thermo code. Various types of parallel prefix adders are used like Brent Kung, Ladner Fischer, Han Carison and Kogge Stone formations. A Carry Look Ahead Adder (CLA) is created by performing carry generation and carry propagation in three stages of adder. Carry generation is executed by performing AND operation. Similarly, carry propagation is performed by using OR operation. A low complexity parallel prefix adder is formed by reducing the number of stages. Therefore, it provides small resource utilization and high performance.

A novel sorting-based arbitration algorithm and a new Merged Arbiter and Multiplexer (MARX) structure are designed in [18]. Then it is integrated into the proposed merge switch allocation and traversal in NoC switches. This router is used in simple and high complexity arbitration strategy. Moreover, it offers low energy, delay and chip size. The Index Based Round Robin Arbiter (IRRA) is designed in [19]. The key is stored in index format to perform switch functions. The input ports are scaled by using a logarithmic operation. In addition, the IRRA provides low die size, low power and high performance. But it limits the fairness of router.

The design of concurrent pseudo round robin arbiter is presented in [20]. It enables several requests at a time via parallel processing and hence the performance is high. Anyhow the area and power utilization are very much elevated. A weighted round robin arbiter is built by using a generic algorithm-based optimization method [21]. Each and every input port is having a suitable weight. The grant signal is prearranged in which channel having supreme credence and thus the speed is increased. Anyhow the complexity of this arbiter is very high. The design of round robin arbiter with clock gating scheme is presented in order to achieve less delay and low power by shrinking the switching movement of the clock signal [22]. Whenever the clock signal and enable are high simultaneously only then the corresponding clock signal is activated. So the power consumption is low due to reducing the number of clock cycle required for arbiter.

An optimized buffered and greedy arbiter was presented in [23]. The complexity of the developed design increases with increase in number of inputs and the efficiency decreases under the condition of high activity where in all the inputs present are requesting for the access of resources or output ports. An arbitration method depending on token ring was presented in [24] which is suitable for on-chip CDMA based bus architectures. But it suffers with the disadvantage of starving among the input requestors. In addition, the efficiency of the designed technique was measured in terms of latency and throughput with no focus on power consumption.

In [25] a modified round robin arbiter was developed using clock gating technique which is power and area efficient but suffered with the drawback of low throughput. The proposed round robin block does not require the ring counter to choose the priority block. The clock signal is provided by the global ring oscillator, with optional dividers for slower clocking options.

According to [26], large number of arbiters are present in Network on Chip and it plays a critical role in defining the performance of the overall NoC. It presented the comparative analysis amongst various arbiters. The work concentrated on decreasing the response time of arbiters to the requesting input ports thereby increasing its speed. The comparison didn’t provide any insights on the amount of power and area consumption which are very crucial for arbiters.

The performance analysis between fixed priority arbiter, round robin arbiter and matrix arbiter was given in [27]. The results show that the fixed priority arbiter doesn’t support fairness in scheduling whereas matrix arbiter leads to large delay, thus making them less efficient when compared to round robin arbiter.

The fairness in granting resources to the requestors is an essential functionality of arbiter. A generic architecture of Dynamic Priority Arbiter was presented in [28] where in the priority was updated dynamically to achieve fast arbitration process. But this method increased the complexity of the designed arbiters and also led to high area and power consumption.

A technique for fair arbitration was developed to guarantee transmitter access for all the input channels in [29]. The proposed technique focused on improving timing towards granting access to the requested input channels without any reference to power and area consumption of the designed arbiter.

A high throughput arbiter based on fixed priority mechanism for applications related to Time Correlated Single Photon Counting system was presented in [30]. The major drawback of this technique is starvation of certain input requestors with more requests leading to increased wait time to get access to the resources.

The problem of constant wait time associated with a round robin arbiter was addressed in [31]. The proposed scheme increased the frequency of operation and throughput but led to increase in area occupation. A modified round robin arbiter was designed in [32] to overcome the problem of starvation where in the requestors need not wait for long to win the granting signal. The arbiter is made aware of the load present on buffer to decide priority of input requestors. This method suffers with the drawback of increased area and power consumption with reduced speed of operation.

The ping lock arbiter is designed in [9] with lock signal for good arbitration and fairness scheme. Therefore, the delay is less but area and power consumption are high. To overcome this problem, an Effective Gate Diffusion Input Logic based Ping Lock Arbiter (CPLA) is designed in the proposed system with a smaller number of transistors when compared to existing static CMOS logic-based ping lock arbiter. Thus, the proposed PLA offers small die size and low power consumption than the other state of art methods.

3. Existing Ping Lock Arbiter with Static CMOS Technique

Fair arbitration is attained in modified ping pong arbiter by altering the information about priority policy. The round robin arbiter is reorganized at the superior level to obtain the first priority sooner than every input is processed in single round of arbitration. The fair arbitration is received by processing the uppermost index-based request in starting meanwhile not updating the priority vector of midway arbiter.

The modified ping pong arbiter known as Ping Lock Arbiter uses a lock signal “l_0” to avoid change of priority register value until the arbiter grants access to the requesting input port. This helps to overcome the problem of unfair arbitration associated with the ping pong arbiter for non-uniform active requests.

The existing ping lock arbiter achieves fair arbitration for both uniform and non-uniform pattern of requesting signals. It’s efficiency in terms of area, power and delay decreases with increase in width of arbiter. For higher order arbiters, ping lock arbiter's performance is almost similar to ping pong arbiter. The basic architecture of ping lock arbiter is based on binary tree structure in which higher width arbiters can be constructed by using lower order arbiters.

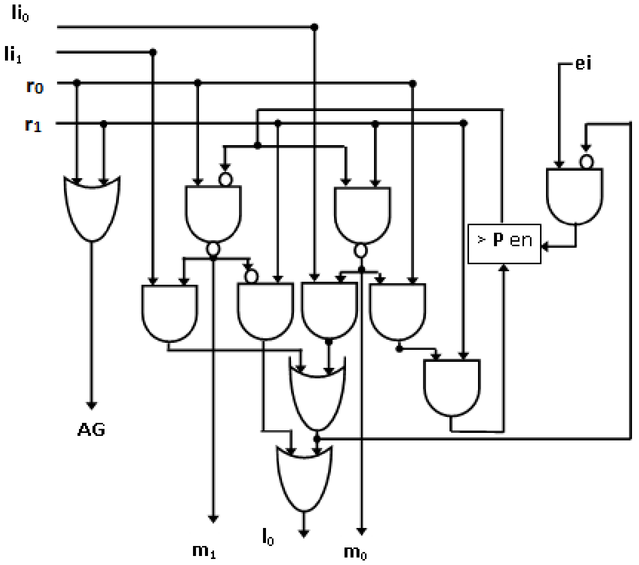


Fig. 3 - Structure of 2-bit Ping Lock Arbiter's Intermediate Arbiter [9]

The gate level structure of intermediate node of 2-bit ping lock arbiter is shown in Fig.3. It consists of two lock signals (li0, li1) corresponding to each requesting input port. The intermediate node produces an output lock signal "l_o" to be used by its parent arbiter.

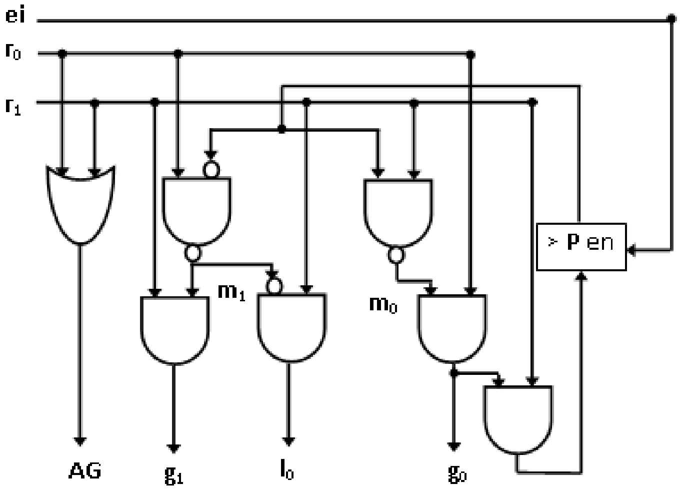


Fig. 4 - Structure of 2-bit Ping Lock Arbiter's Leaf Arbiter [9]

The gate level structure of leaf node corresponding to 2-bit ping lock arbiter is shown in Fig.4. It asserts the lock signal "l_0" only when it receives active requests from both r_0 and r_1 along with zero priority pointer. The leaf and intermediate cells presented consists of OR, NAND, AND gates, inverter and a register. The OR operation is performed between two request signals to produce the asserting signal (AG).

The priority register (P-reg) is designed asynchronously to make a glitch free circuit. Any unwanted signal produced at the output is called as glitch. The enable signal of P-reg is generated by using AND gate in which the output is feedback to give as input with error input (ei). When the enable is one, the grant signal is generated based on request inputs and p-register output. The input to output, input to register, register to output delay are calculated to find the arbiter maximum delay (MD). The Granting delay (GD) is defined as the maximum delay of arbiter to affirm the grant later than getting a request. The granting delay should be equal are lesser than the maximum delay of the arbiter.

The general structure of n-bit arbiter is depicted in Fig.5 where ‘r’ indicates request signals, ‘g’ as grant signals, ‘l_0’ is lock signal, ei and e_0 represent input and output enable signals. An n-bit ping lock arbiter (PLA_n) is constructed with the help of 2 $PLA_{n/2}$ and an intermediate PLA_2.

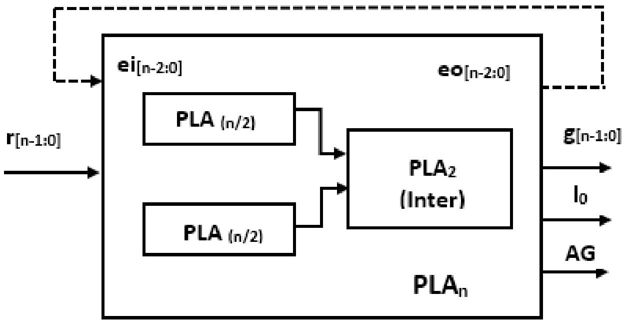


Fig. 5 - Structure of n-bit ping lock arbiter

The four-bit Ping Lock Arbiter (PLA) as shown in Fig.6 is created by combing two 2-bit PLA leaf and one 2-bit PLA intermediate structure (Fig.3 and Fig.4) along with four AND gates which is also called as new gate level compact round robin arbiter. The excellent grant updating policy is included in the PLA to attain fair arbitration. The lock out (lo), AG, error output (eo) and grant signal are generated by PLA. The output is obtained without any signal distortion with fair arbitration. Static Complementary Metal Oxide Semiconductor (S-CMOS) logic is used to design existing ping lock arbiter of basic and higher width. This leads to increased area and power consumption and the overall performance is quite similar to the conventional ping pong arbiter. Similarly, an 8-bit arbiter built using two 4-bit PLA and one intermediate PLA is also analyzed and compared with existing CMOS logic-based arbiter defined in [9].

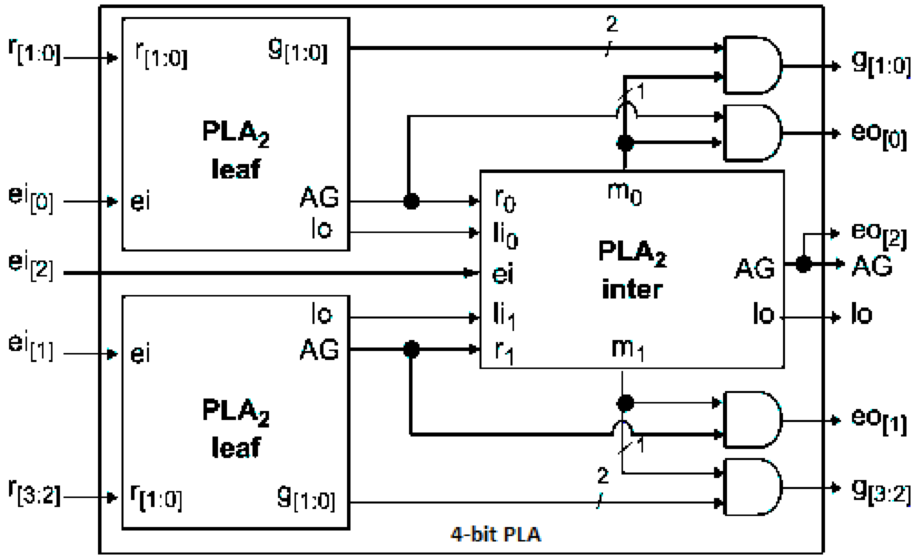
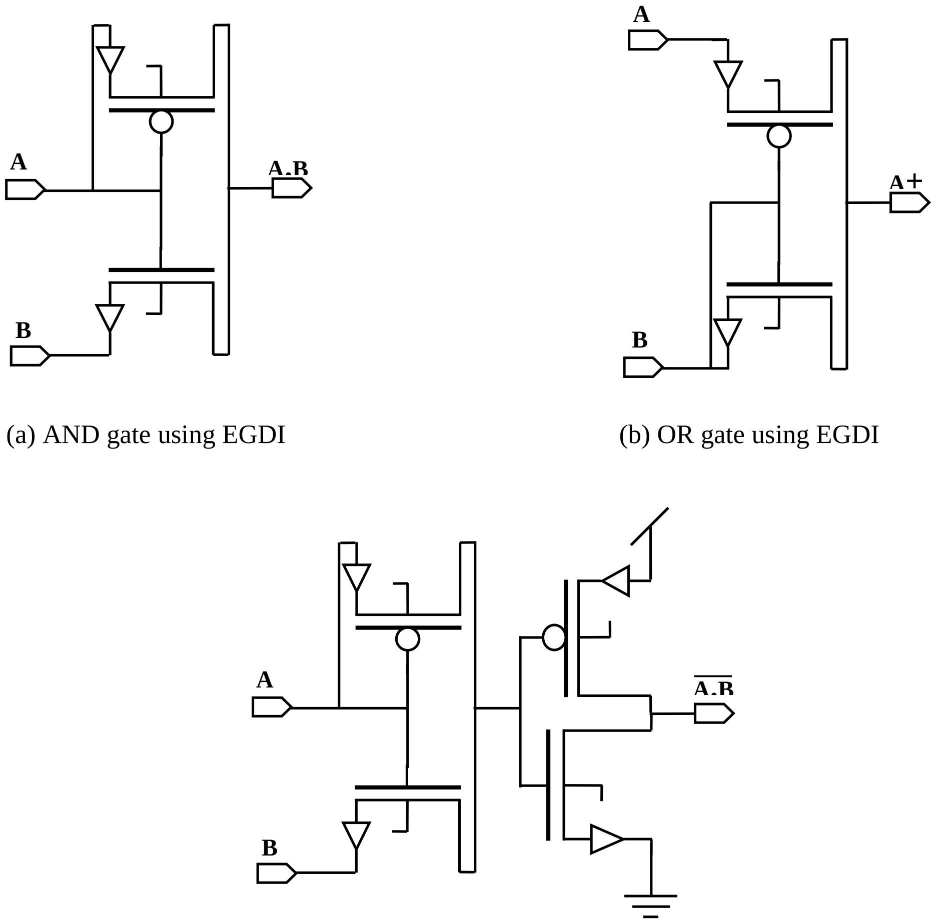


Fig. 6 - Structure of 4-bit ping lock arbiter

4. Proposed Ping Lock Arbiter with EGDI Logic

The existing ping lock arbiter achieves fairness in arbitration but suffers from the drawback of high power and area consumption. To overcome the drawback of existing PLA design, a modified PLA is proposed using GDI technique to reduce total area and power consumed by the arbiter. The effective GDI logic uses a smaller number of transistors to complete the basic logic gate operation compared to all other styles like Static CMOS logic, FINFET logic, Pass Transistor Logic (PTL), domino logic etc. The implementation of AND gate using EGDI logic is shown in Fig.7 (a). It needs only two transistors to perform the operation. In the gate diffusion logic, the drain and source terminal of MOSFET are used as one of the input terminal instead of the gate terminal. Hence the number of transistors are very less in EGDI logic. When A and B values are zero, the PMOS is ON and NMOS is OFF which is connected to A in drain. Thus, the values of A equal to zero is passed to the output. When A=0 and B=1, PMOS is ON and NMOS is OFF but the drain of

PMOS is connected to A. Hence the output is zero. Similarly, if A=1 and B=0, PMOS is OFF and NMOS is ON. Anyhow as the NMOS drain terminal is connected to B, the value of B is passed to output as zero. Similarly, when all the inputs are high, PMOS is OFF and NMOS is ON. The drain of NMOS is attached to B, making the output as B (i.e.) one.



(a) AND gate using EGDI

(b) OR gate using EGDI

(c) NAND gate using EGDI

Fig. 7 - Circuit diagram of AND, OR, NAND gates using EGDI logic

The circuit diagram of OR gate using EGDI is shown in Fig.7 (b). When both the inputs are zero, PMOS is ON and NMOS is OFF. The drain terminal of PMOS is attached to A makes the output as zero. When all the inputs are one, PMOS is OFF and NMOS is ON which is connected to B. Thus, the output is equal to one. If A=0 and B=1, NMOS is ON & PMOS is OFF, as the NMOS is connected to B, the output is pulled to one. The inverse operation is performed when A=1 and B=0. The EGDI logic-based NAND gate is shown in Fig.7 (c). The inverse operation of AND gate with EGDI is performed in NAND gate by adding the inverter in final stage.

A basic CMOS logic AND gate normally uses six transistors, whereas the same gate when implemented using EGDI logic uses only two transistors making it more area effective and less complex. Similarly, other basic gates when implemented using EGDI logic turn out to be area efficient and less complex when compared to static CMOS logic circuits.

The proposed compact ping lock round robin arbiter is designed by integrating EGDI based AND, OR, NAND gates instead of static CMOS logic gates in 4-bit and 8-bit ping lock arbiter. Initially the PLA2-leaf cell and PLA2-intermediate cell is designed using the basic gates designed using EGDI logic which uses a smaller number of transistors when compared to existing CMOS logic-based cells. The schematic of PLA2-leaf and PLA2-inter are presented in Fig.8 and Fig.9 respectively. These schematics are taken from the S-Edit window of Tanner tool.

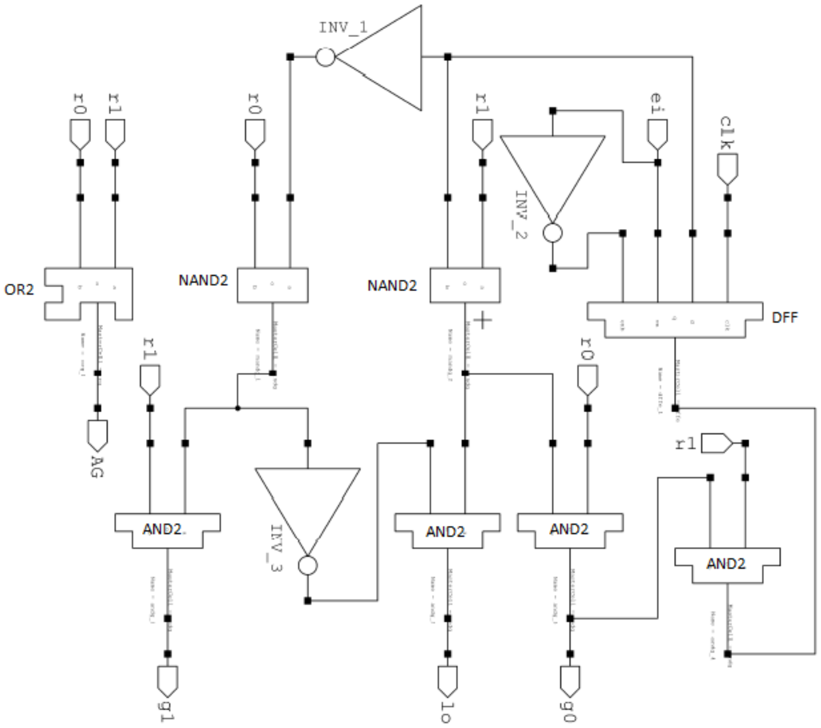


Fig. 8 - Schematic of EGDI based PLA2 Leaf Cell

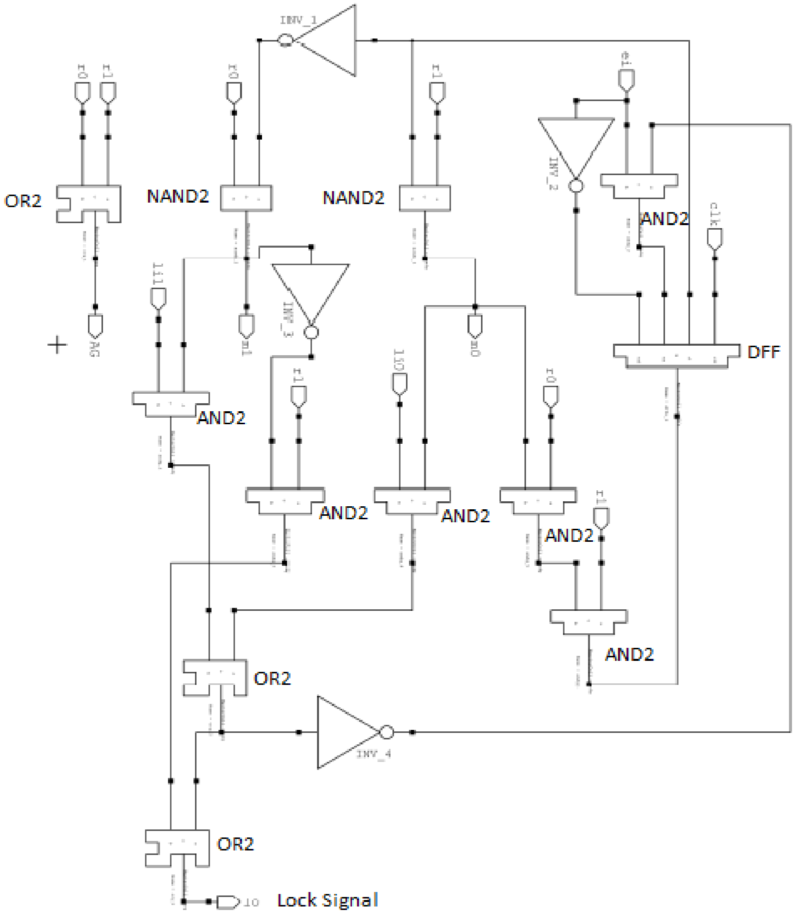


Fig. 9 - Schematic of EGDI based PLA2 Leaf Cell

The ping lock arbiter uses binary tree-based architecture to build higher order complex arbiter circuits. The 4-bit PLA is developed using an EGDI based PLA-leaf and PLA-intermediate cell as shown in Fig.10. It is designed by using PLA-2 inter and PLA-2 leaf structure. Finally, six AND gates are used to combine the outputs of PLA2 leaf and PLA2 inter to find the grant value and the error output.

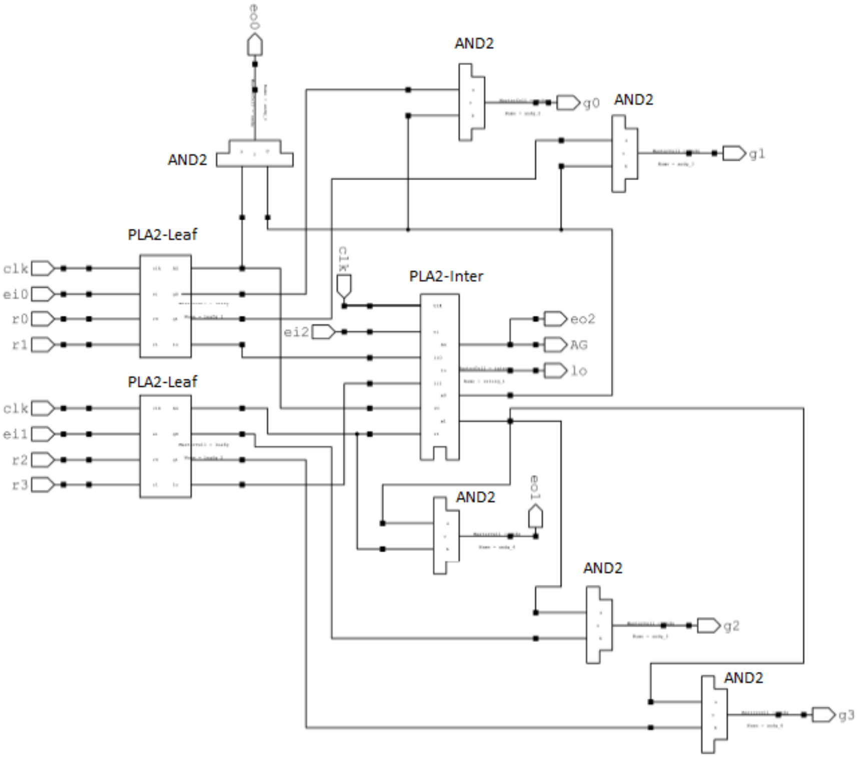


Fig. 10 - Schematic of proposed 4-bit PLRRA with EGDI logic

The 8-bit PLA is designed by using two 4-bit ping lock arbiter with 2-bit PLA intermediate structure as depicted in Fig.11. The proposed compact ping lock arbiter (CPLA) offers small die size, low delay and low power consumption than the other state of arts scheme.

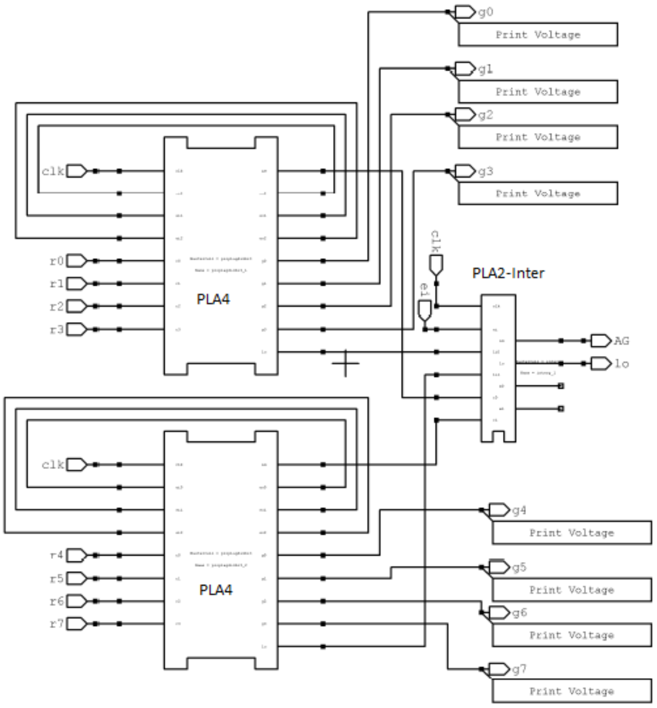


Fig. 11 - Schematic of proposed 8bit PLRRA with EGDI logic

5. Results and Discussion

In this work, two types of arbiter such as static CMOS based Ping Lock Round Robin Arbiter (PLRRA) and EGDI logic based PLRRA are designed by using tanner tool to verify the working principle of arbiters. Also the simulation results are taken to examine the number of transistors (area) needed along with the values of delay and power consumption. The value of power, delay and area compared for both conventional and proposed scheme to judge the efficiency of performance. The tanner tool W-edit is used to check the functionality of Design under Test (DUT). Also T-Spice is used to calculate the area, delay and power values.

Table 1 - Comparison between existing and proposed 4-bit PLRRA for 250nm

PLA 4-bit	Number of transistors	Delay (ns)	Power (mw)
With CMOS	252	0.9	3.53
With EGDI	148	1.12	0.51

In this work, two different types of technology are used for low power consumption. In the 250nm, 2.5V is an optimum supply voltage. Similarly, for 45nm, 0.5V optimum supply voltage is given for exact output. The Transistor sizing is done for both 250nm and 45nm technology. The value of L value is fixed, 0.25uv or 250nm for 250nm and 0.045uv or 45nm for 45nm technology. To find the value of width W, first the value of lambda is calculated by dividing the technology with 2.

Table 2 - Comparison between existing and proposed 8-bit PLRRA for 250nm

PLA 8-bit	No of transistors	Delay (ns)	Power (mw)
With static CMOS	592	1.62	7.88
With EGDI	348	3.18	4.38

The results of proposed 4-bit PLRRA is shown in Table.1. From the results, it shows that the proposed PLRRA with EGDI offers 41.22% area reduction and 85.55% power reduction than the existing PLA with static CMOS scheme. But the speed is high for static CMOS based ping lock arbiter. Similarly, results of 8-bit ping lock round robin arbiter are given in Table.2. From the obtained results, the existing static CMOS based PLRRA offers low delay but has high area and power consumption in comparison to the proposed EGDI. The proposed EGDI based 8-bit PLRRA provides 41.22% area reduction and 44.42% power reduction, when compared to existing static CMOS logic based 8-bit PLRRA.

Table 3 - Comparison between existing and proposed 4-bit PLRRA for 45nm

PLA 4-bit	Number of transistors	Delay (ns)	Power (uw)
With Static CMOS	252	1.78	8.41
With EGDI	148	3.22	7.145

From the results, it shows that the proposed 4-bit and 8-bit Ping Lock Round Robin Arbiter (PLRRA) with EGDI logic requires a smaller number of transistors and power than the conventional PLRRA with static CMOS scheme. The proposed 4-bit PLRRA offers 41.27% area reduction and 15% power reduction when compared to the conventional 4-bit PLRRA as shown in Table.3. The obtained results demonstrate that the proposed EGDI based ping lock arbiter is more efficient than existing CMOS logic-based arbiter.

Table 4 - Comparison between existing and proposed 8-bit PLRRA for 45nm

PLA 8-bit	Number of transistors	Delay (ns)	Power (uw)
With Static CMOS	592	4.563	20.45
With EGDI	348	6.754	16.15

Likewise, from the Table.4, the proposed 8-bit PLRRA gives 41.21% chip size reduction and 21% power reduction than the existing 8-bit PLRAA. However, the speed is high is low for existing static CMOS based 4-bit and 8-bit PLA, when compared to the proposed 4-bit and 8-bit PLA with EGDI.

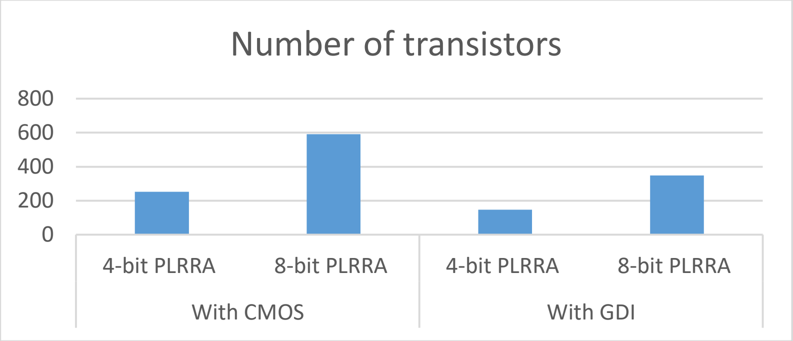


Fig. 12 - Comparison result for total count of transistors required for existing and proposed PLRRA

The comparison results for the total number of transistors required to design 4-bit PLRRA and 8-bit PLRRA using existing static CMOS and EGDI is shown in Fig.12. It depicts that EGDI based PLRRA requires a smaller number of transistors in comparison to static CMOS PLRRA and thus reduces the overall area consumption.

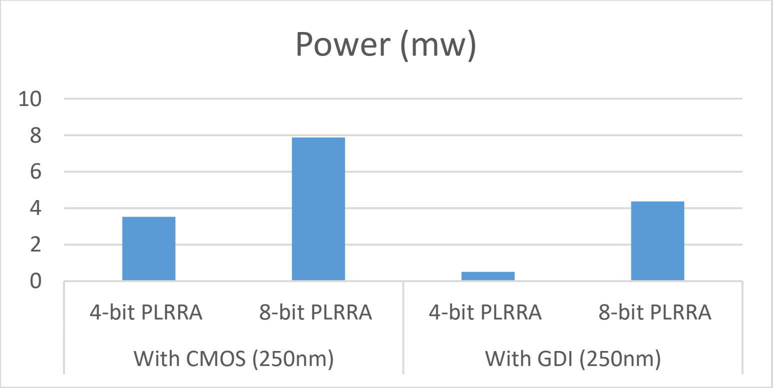


Fig. 13 - Comparison result for total power consumption in 250 nm for existing and proposed PLRRA

Similarly, the Fig.13 presents the comparison result corresponding to the value of total power consumption required for existing and proposed PLRRA in 250 nm technology. The graphical comparison presents that the total power consumption of EGDI PLRRA is very less when compared to the existing CMOS logic PLRRA and hence is more power efficient.

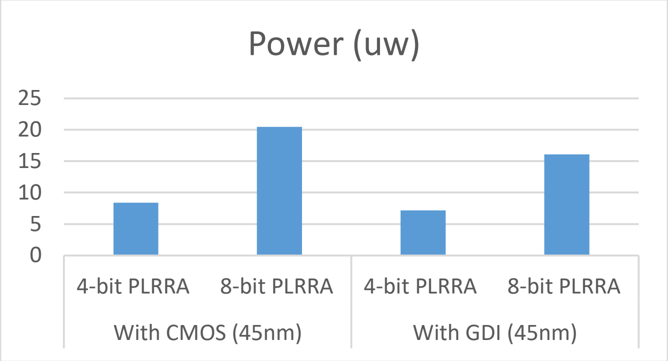
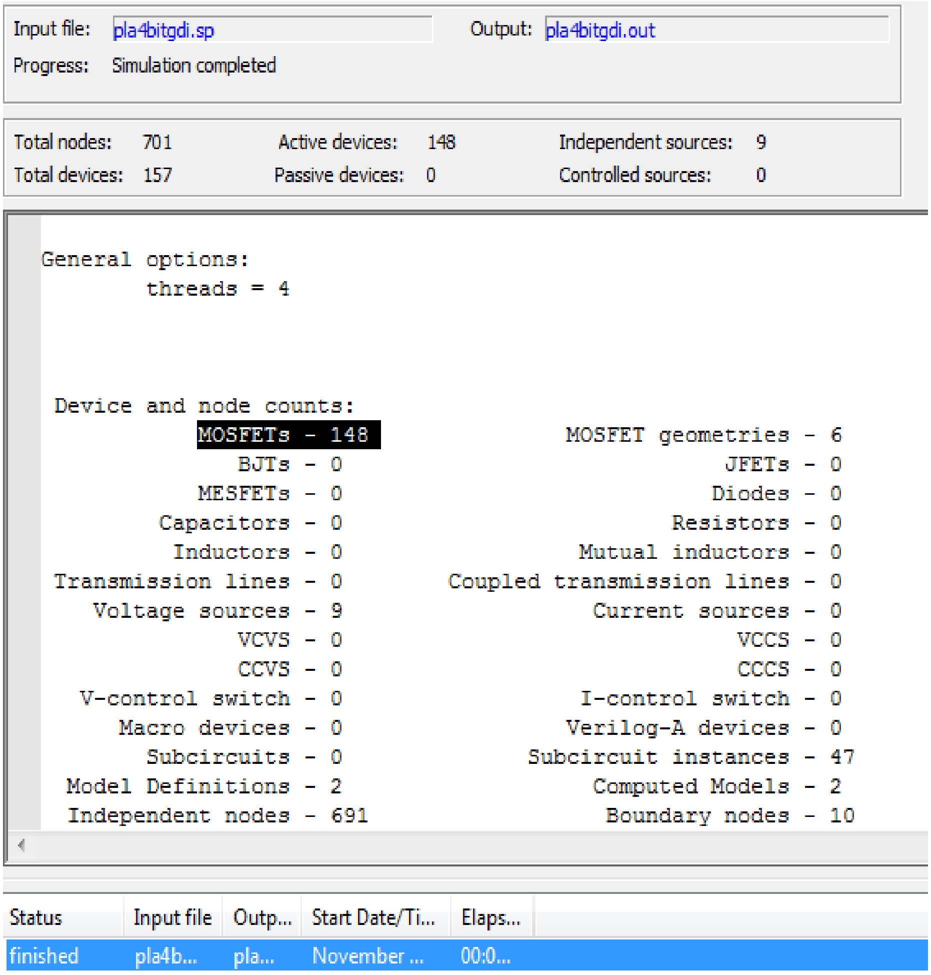


Fig. 14 - Comparison result for total power consumption in 45 nm for existing and proposed PLRRA

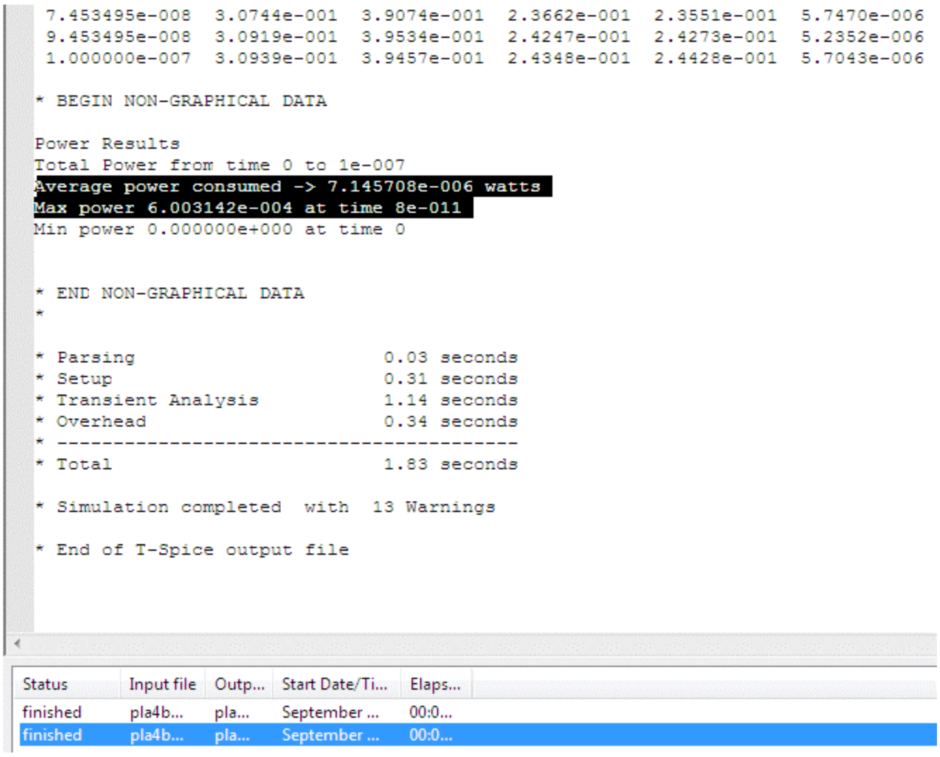
The Fig.14 demonstrates the comparison result corresponding to the value of total power consumption required for existing and proposed PLRRA in 45 nm technology. The output obtained for proposed 4-bit ping lock round robin arbiter related to area utilization given in Fig.15.



```
Input file:  pla4bitgdi.sp                    Output:  pla4bitgdi.out
Progress:    Simulation completed

Total nodes:     701        Active devices:    148        Independent sources:   9
Total devices:   157        Passive devices:   0          Controlled sources:    0

General options:
        threads = 4


 Device and node counts:
            MOSFETs - 148                    MOSFET geometries - 6
               BJTs - 0                                  JFETs - 0
            MESFETs - 0                                 Diodes - 0
         Capacitors - 0                              Resistors - 0
          Inductors - 0                      Mutual inductors - 0
 Transmission lines - 0            Coupled transmission lines - 0
    Voltage sources - 9                        Current sources - 0
               VCVS - 0                                   VCCS - 0
               CCVS - 0                                   CCCS - 0
   V-control switch - 0                       I-control switch - 0
     Macro devices - 0                       Verilog-A devices - 0
       Subcircuits - 0                    Subcircuit instances - 47
  Model Definitions - 2                        Computed Models - 2
  Independent nodes - 691                       Boundary nodes - 10

Status     Input file   Outp...   Start Date/Ti...   Elaps...
finished   pla4b...     pla...    November ...       00:0...
```

Fig. 15 - Resource utilization of proposed 4-bit PLRRA with EGDI logic

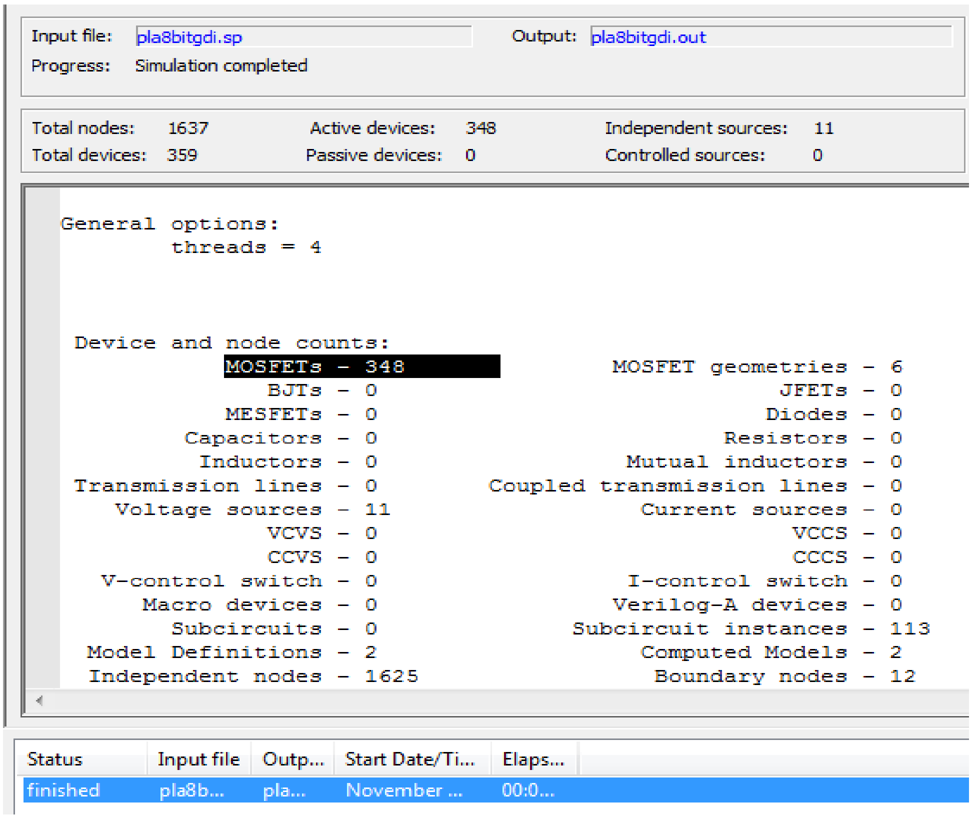
The number of transistors used by the proposed 4-bit PLRRA is 148 as shown in Fig.15 and similarly the value of power consumed obtained is presented in Fig.16 which is 7.14 µw.



```
 7.453495e-008  3.0744e-001  3.9074e-001  2.3662e-001  2.3551e-001  5.7470e-006
 9.453495e-008  3.0919e-001  3.9534e-001  2.4247e-001  2.4273e-001  5.2352e-006
 1.000000e-007  3.0939e-001  3.9457e-001  2.4348e-001  2.4428e-001  5.7043e-006

* BEGIN NON-GRAPHICAL DATA

Power Results
Total Power from time 0 to 1e-007
Average power consumed -> 7.145708e-006 watts
Max power 6.003142e-004 at time 8e-011
Min power 0.000000e+000 at time 0


* END NON-GRAPHICAL DATA
*

* Parsing                      0.03 seconds
* Setup                        0.31 seconds
* Transient Analysis           1.14 seconds
* Overhead                     0.34 seconds
* -----------------------------------------
* Total                        1.83 seconds

* Simulation completed  with  13 Warnings

* End of T-Spice output file
```

Status	Input file	Outp...	Start Date/Ti...	Elaps...
finished	pla4b...	pla...	September ...	00:0...
finished	pla4b...	pla...	September ...	00:0...

Fig. 16 - Power consumption of proposed 4-bit PLRRA with EGDI logic



Input file: pla8bitgdi.sp Output: pla8bitgdi.out

Progress: Simulation completed

Total nodes: 1637 Active devices: 348 Independent sources: 11

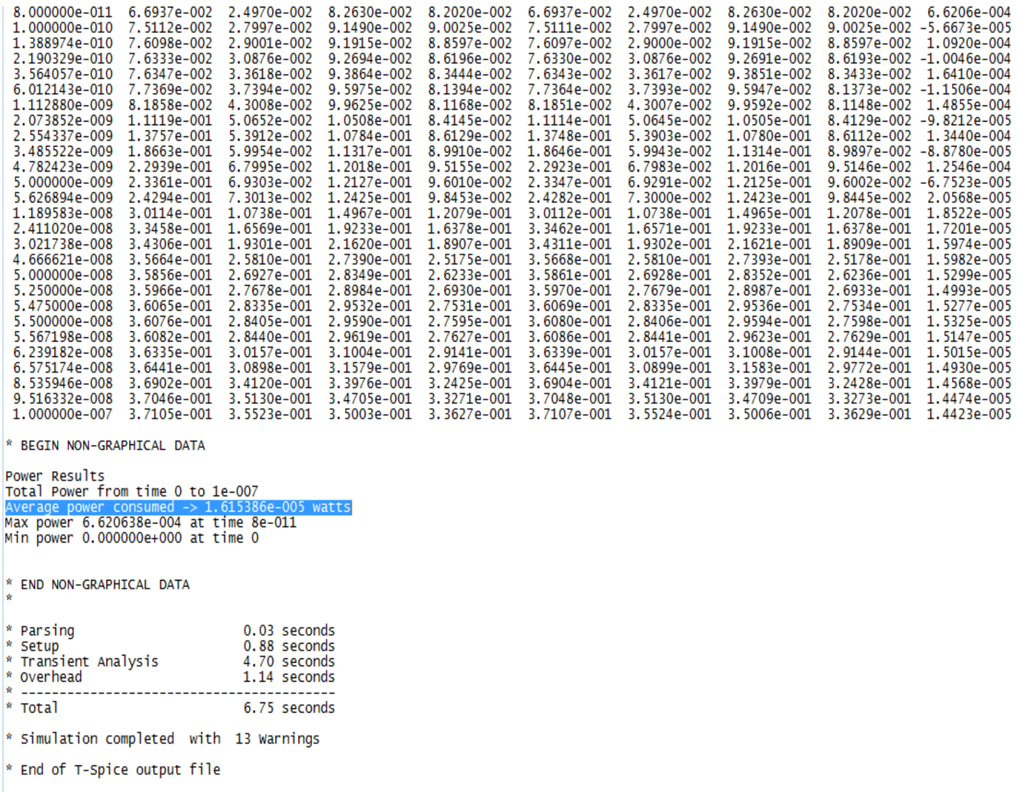
Total devices: 359 Passive devices: 0 Controlled sources: 0

```
General options:
        threads = 4


 Device and node counts:
            MOSFETs - 348               MOSFET geometries - 6
               BJTs - 0                             JFETs - 0
            MESFETs - 0                            Diodes - 0
         Capacitors - 0                         Resistors - 0
          Inductors - 0                 Mutual inductors - 0
 Transmission lines - 0       Coupled transmission lines - 0
    Voltage sources - 11                 Current sources - 0
               VCVS - 0                              VCCS - 0
               CCVS - 0                              CCCS - 0
   V-control switch - 0                 I-control switch - 0
      Macro devices - 0                 Verilog-A devices - 0
        Subcircuits - 0              Subcircuit instances - 113
  Model Definitions - 2                   Computed Models - 2
  Independent nodes - 1625                 Boundary nodes - 12
```

Status	Input file	Outp...	Start Date/Ti...	Elaps...
finished	pla8b...	pla...	November ...	00:0...

Fig. 17 - Resource utilization of proposed 8-bit PLRRA with EGDI logic

Similarly, the number of transistors required and power consumption of proposed 8-bit ping lock round robin arbiter are illustrated in Fig.17 and Fig.18 respectively.



```
 8.000000e-011  6.6937e-002  2.4970e-002  8.2630e-002  8.2020e-002  6.6937e-002  2.4970e-002  8.2630e-002  8.2020e-002  6.6206e-004
 1.000000e-010  7.5112e-002  2.7997e-002  9.1490e-002  9.0025e-002  7.5111e-002  2.7997e-002  9.1490e-002  9.0025e-002 -5.6673e-005
 1.388974e-010  7.6098e-002  2.9001e-002  9.1915e-002  8.8597e-002  7.6097e-002  2.9000e-002  9.1915e-002  8.8597e-002  1.0920e-004
 2.190329e-010  7.6333e-002  3.0876e-002  9.2694e-002  8.6196e-002  7.6330e-002  3.0876e-002  9.2691e-002  8.6193e-002 -1.0046e-004
 3.564057e-010  7.6347e-002  3.3618e-002  9.3864e-002  8.3444e-002  7.6343e-002  3.3617e-002  9.3851e-002  8.3433e-002  1.6410e-004
 6.012143e-010  7.7369e-002  3.7394e-002  9.5975e-002  8.1394e-002  7.7364e-002  3.7393e-002  9.5947e-002  8.1373e-002 -1.1506e-004
 1.112880e-009  8.1858e-002  4.3008e-002  9.9625e-002  8.1168e-002  8.1851e-002  4.3007e-002  9.9592e-002  8.1148e-002  1.4855e-004
 2.073852e-009  1.1119e-001  5.0652e-002  1.0508e-001  8.4145e-002  1.1114e-001  5.0645e-002  1.0505e-001  8.4129e-002 -9.8212e-005
 2.554337e-009  1.3757e-001  5.3912e-002  1.0784e-001  8.6129e-002  1.3748e-001  5.3903e-002  1.0780e-001  8.6112e-002  1.3440e-004
 3.485522e-009  1.8663e-001  5.9954e-002  1.1317e-001  8.9910e-002  1.8646e-001  5.9943e-002  1.1314e-001  8.9897e-002 -8.8780e-005
 4.782423e-009  2.2939e-001  6.7995e-002  1.2018e-001  9.5155e-002  2.2923e-001  6.7983e-002  1.2016e-001  9.5146e-002  1.2546e-004
 5.000000e-009  2.3361e-001  6.9303e-002  1.2127e-001  9.6010e-002  2.3347e-001  6.9291e-002  1.2125e-001  9.6002e-002 -6.7523e-005
 5.626894e-009  2.4294e-001  7.3013e-002  1.2425e-001  9.8453e-002  2.4282e-001  7.3000e-002  1.2423e-001  9.8445e-002  2.0568e-005
 1.189583e-008  3.0114e-001  1.0738e-001  1.4967e-001  1.2079e-001  3.0112e-001  1.0738e-001  1.4965e-001  1.2078e-001  1.8522e-005
 2.411020e-008  3.3458e-001  1.6569e-001  1.9233e-001  1.6378e-001  3.3462e-001  1.6571e-001  1.9233e-001  1.6378e-001  1.7201e-005
 3.021738e-008  3.4306e-001  1.9301e-001  2.1620e-001  1.8907e-001  3.4311e-001  1.9302e-001  2.1621e-001  1.8909e-001  1.5974e-005
 4.666621e-008  3.5664e-001  2.5810e-001  2.7390e-001  2.5175e-001  3.5668e-001  2.5810e-001  2.7393e-001  2.5178e-001  1.5982e-005
 5.000000e-008  3.5856e-001  2.6927e-001  2.8349e-001  2.6233e-001  3.5861e-001  2.6928e-001  2.8352e-001  2.6236e-001  1.5299e-005
 5.250000e-008  3.5966e-001  2.7678e-001  2.8984e-001  2.6930e-001  3.5970e-001  2.7679e-001  2.8987e-001  2.6933e-001  1.4993e-005
 5.475000e-008  3.6065e-001  2.8335e-001  2.9532e-001  2.7531e-001  3.6069e-001  2.8335e-001  2.9536e-001  2.7534e-001  1.5277e-005
 5.500000e-008  3.6076e-001  2.8405e-001  2.9590e-001  2.7595e-001  3.6080e-001  2.8406e-001  2.9594e-001  2.7598e-001  1.5325e-005
 5.567198e-008  3.6082e-001  2.8440e-001  2.9619e-001  2.7627e-001  3.6086e-001  2.8441e-001  2.9623e-001  2.7629e-001  1.5147e-005
 6.239182e-008  3.6335e-001  3.0157e-001  3.1004e-001  2.9141e-001  3.6339e-001  3.0157e-001  3.1008e-001  2.9144e-001  1.5015e-005
 6.575174e-008  3.6441e-001  3.0898e-001  3.1579e-001  2.9769e-001  3.6445e-001  3.0899e-001  3.1583e-001  2.9772e-001  1.4930e-005
 8.535946e-008  3.6902e-001  3.4120e-001  3.3976e-001  3.2425e-001  3.6904e-001  3.4121e-001  3.3979e-001  3.2428e-001  1.4568e-005
 9.516332e-008  3.7046e-001  3.5130e-001  3.4705e-001  3.3271e-001  3.7048e-001  3.5130e-001  3.4709e-001  3.3273e-001  1.4474e-005
 1.000000e-007  3.7105e-001  3.5523e-001  3.5003e-001  3.3627e-001  3.7107e-001  3.5524e-001  3.5006e-001  3.3629e-001  1.4423e-005

* BEGIN NON-GRAPHICAL DATA

Power Results
Total Power from time 0 to 1e-007
Average power consumed -> 1.615386e-005 watts
Max power 6.620638e-004 at time 8e-011
Min power 0.000000e+000 at time 0


* END NON-GRAPHICAL DATA
*

* Parsing                     0.03 seconds
* Setup                       0.88 seconds
* Transient Analysis          4.70 seconds
* Overhead                    1.14 seconds
* -----------------------------------------
* Total                       6.75 seconds

* Simulation completed  with  13 Warnings

* End of T-Spice output file
```

Fig. 18 - Power consumption of proposed 8-bit PLRRA with EGDI logic

6. Conclusion

In this paper, the compact PLRRA with high speed is constructed to attain fair arbitration. Many kinds of arbiters are premeditated for widespread resource sharing method. The arbiters can be designed using various available logical styles like static CMOS, Pseudo NMOS, Pass Transistor and Gate diffusion input etc. Out of all the styles, EGDI logic can implement circuits with a smaller number of transistors thereby reducing area and power consumption of the design. The two variations of arbiter like Ping Lock Arbiter (PLRRA) with static CMOS and Compact Ping Lock Arbiter (CPLRRA) with EGDI logic are designed to evaluate the die size, implementation delay and power values. The arbiters are designed by using tanner tool S-edit based on switch level (SL) execution. The functionality of every arbiter is demonstrated via tanner tool W-edit. T-spice is used to calculate the area, delay and power outputs. The proposed CPLRRA is best for compact and smart applications with glitch free and fair arbitration. The proposed 4-bit PLRRA offers 47.12% average Area and Power Product (APP) reduction than the existing 4-bit PLRRA. On the other hand, the proposed 8-bit CPLRRA requires 49.48% average APP product shrinkage when compared to conventional 8-bit PLRRA. For high-speed application, the static CMOS based ping lock round robin arbiter is best when compared to the proposed PLRRA with EGDI. In future, the proposed PLRRA will be integrated in the application specific NoC router, computer networks etc. and their performances can be compared.

References

[1] Luca Benini and G D Micheli, “Network on Chips: A New SoC Paradigm”, IEEE Comp. Society, vol.35, no. 1, Aug 2002.

[2] Sudeep Pasricha and Nikhil Dutt, “On-Chip Communication Architectures-System on Chip Interconnect”, Elsevier, 2008.

[3] Radu Marculescu, et al., “Outstanding Research Problems in NoC Design : System, Microarchitecture, and Circuit Perspectives,” IEEE Trans. VLSI Syst., Vol. 28, pp.12-21, 2009.

[4] Y. Jin, et al., “Communication-Aware Globally- Coordinated On-Chip Networks”, IEEE Trans. Parallel and Distrib. Syst., Vol. 23, no. 2, pp. 242-253, February 2012.

[5] J D. Owens, et al., “Research Challenges for on-Chip Interconnection Networks”, IEEE Trans. Micro., Vol. 27, no.5, Oct 2007.

[6] Sangeeta Singh et. al, “Power and Area Optimized FRA-CSLA for High-Speed NoC Applications”, International Journal of Advanced Trends in Computer Science and Engineering, Vol.8, no. 3, May-June 2019.

[7] Sangeeta Singh, et. al, “Power and Area Calibration of Switch Arbiter for High Speed Switch Control and Scheduling in Network-on-Chip”, in IEEE 13th International SoC Design Conference (ISOCC), pp. 5-6, Oct, 2016, Jeju, South Korea.

[8] H. Jonathan, et al., “A Fast Arbitration Scheme for Terabit Packet Switches”, Globecom, 1999.

[9] Alireza Monemi, Chia Yee Ooi, Maurizio Palesi, Muhammad N. Marsono, “Ping-lock round robin arbiter”, Microelectronics Journal, 63, 81–93, 2017.

[10] Sucharitha D.; Prudhvi Raj N.; Sravya Reddy B. and Sudheer Raja V., “GDI Logic Based Design of Hamming-Code Encoder and Decoder for Error Free Data Communication”, 3rd International Conference on Computing Methodologies and Communication (ICCMC), IEEE 2019.

[11] Arkadiy Morgenshtein; Idan Shwartz; Alexander Fish, “Gate Diffusion Input (GDI) logic in standard CMOS nanoscale process”, 26-th Convention of Electrical and Electronics Engineers in Israel, IEEE, Dec 2010.

[12] P.Gupta, N. McKeown, “Designing and implementing a fast crossbar scheduler”, IEEE Micro 19, no.1, pp.20–28, 1999.

[13] S. Zheng, M. Yang, J. Blanton, P. Golla, D. Verchere, “A simple and fast parallel round-robin arbiter for high-speed switch control and scheduling”, the 45th Midwest Symposium on Circuits and Systems, (MWSCAS-2002), vol. 2, pp. 671–674, 2002.

[14] J.Reed, N. Manjikian, “A dual round-robin arbiter for split-transaction buses in system-on-chip implementations”, in: Canadian Conference on Electrical and Computer Engineering, , vol. 2, IEEE, pp. 835–840, 2004.

[15] Shyue-Wen Yang, Ming-Hwa Sheu, Chun-Kai Yeh, Chih-Yuen Wen, Chih-Chieh Lin, Wen-Kai Tsai, “Fast Fair Crossbar Scheduler for On-chip Router”, IEEE, 2007.

[16] G.Dimitrakopoulos, N. Chrysos, K. Galanopoulos, “Fast arbiters for on-chip network switches”, IEEE International Conference on Computer Design (ICCD 2008), pp. 664–670, 2008.

[17] H.F. Ugurdag, O. Baskirt, “Fast parallel prefix logic circuits for n2n round-robin arbitration”, Micro electron. J. 43 (8), 573–581, 2012.

[18] G. Dimitrakopoulos, E. Kalligeros, K. Galanopoulos, “Merged switch allocation and traversal in network-on-chip switches”, IEEE Trans. Comput. 62 (10), 2013.

[19] M.Oveis-Gharan, G.N. Khan, “Index-based Round-Robin Arbiter for NoC Routers”, IEEE Computer Society Annual Symposium on VLSI, IEEE, pp. 62–67, 2015.

[20] R. Bashizade, H. Sarbazi-Azad, “P2R2: parallel Pseudo-Round-Robin arbiter for high performance NoCs”, Integration VLSI J. 50, 173–182, 2015.

[21] Fahimeh Jafari, Axel Jantsch, and Zhonghai Lu, “Weighted Round Robin Configuration for Worst-Case Delay Optimization in Network-on-Chip”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp.1–14, 2016.

[22] Biman Debbarma and Jeetu Debbarma, “Clock Gated Round Robin Arbiter for NoC Router”, International Journal of Science and Technology, No 3, pp. 113-124, 2017.

[23] Ned B and Rajit Manohar, “A Systematic Approach for Arbitration Expressions”, IEEE Trans. on Circuits and Systems, 2020.

[24] Tatjana R.NikolicSandra, et al., “Token Ring Arbitration Scheme for on-chip CDMA Architectures”, Microelectronics Journal, Vol.106, Dec 2020.

[25] Junran Pu, et al., “Power and Area Efficient Router with Automated Clock Gating for Neuromorphic Computing”, 32nd IEEE International System-on-Chip Conference (SOCC), 2019.

[26] Girish A. Subbarao, et al., “Design and Comparison of Synthesizable Fair Asynchronous Arbiter”, 18th IEEE International New Circuits and Systems Conference (NEWCAS), 2020.

[27] Rajeev Kamal, et al., “RTL implementation and analysis of fixed priority, round robin, and matrix arbiters for the NoC's routers”, International Conference on Computing, Communication and Automation (ICCCA), 29-30 April 2016.

[28] G. Dimitrakopoulos, A. Psarras and I. Seitanidis, "Microarchitecture of Network-on-Chip Routers", *Springer*, 2015.

[29] G.A. Subbarao and P.D. Häfliger, "Fair Arbitration for Multi-Instrument Space Probe", *IEEE New Circuits & Systems Conference*, Jun. 2017.

[30] T. Turko et al., "An Asynchronous Fixed Priority Arbiter for High Througput Time Correlated Single Photon Counting Systems", *International Conference on Electronics Circuits & Systems*, Dec. 2018.

[31] A.A Khan, et al., “Fault-Tolerant Buffer Aware Round Robin Arbiter Design for NoC Architectures”, International Journal of Computing and Digital Systems, 2019.

[32] A.A.Khan, et al., “Buffer aware arbiter design to achieve improved QoS for NoC”, TENCON, IEEE Region 10 Conference, 2017.

[33] Ashok Kumar K, et al., “Parallel Distributed Round Robin Arbiter for Network on Chip”, Journal of Computational and Theoretical Nanoscience 15(6):2064-2068, June 2018.