ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

Performance Analysis of Various Scheduling Algorithms using FPGA Platforms

Priya.V[1], Pujitha.S[2], Radhika.P[3], S.Vijay Murugan[4]

UG Scholar, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[1,2,3]

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[5]

ABSTRACT: This paper explores about the integration and implementation of various scheduling algorithms for Network on-Chip communication. Initially, these scheduling algorithms were established on ASIC platforms generally for shared bus based interconnection systems. Here we discuss about a comparative analysis by synthesizing and implementing various scheduling algorithms for sorting the crossbar in input queued switches. The implementation is carried out using various arbitration networks responsible for scheduling 8-bit input requests. This application targets Spartan6 FPGA family. The analysis concludes that the scheduling algorithm based on CLA based encoding network shows lower power delay product and lower area delay product and a reasonably lower resource utilization when implemented for speed optimization goal.

KEYWORDS: Network-on-chip, arbiter, scheduling, PPE.

I. INTRODUCTION

The data paths of the on-chip communication routers are made up of buffers and switching fabric, but the control paths of onchip communication routers are largely composed of arbiters and allocators. Allocators are used to allocate virtual channels, to packets and to allocate switch cycles to flits. Switching fabric comprising crossbar switches has a drawback that they make it difficult to provide the guaranteed quality of service. This is because the flits/cells arriving at the fabric must contend for access to the fabric with cells at both input and output. The time at which they leave the input queues and enter the crossbar switching fabric is dependent on the other traffic in the system, making it difficult to control when a cell will depart. One of the ways to mitigate this problem is to schedule the transfer of cells from inputs to outputs in a similar manner to that used in a timeslot interchanger, providing peak bandwidth allocation for reserved flows. The method has been implemented in many switches and on chip routers. The scheduling of the transfer of cells is achieved by scheduling algorithms used in arbiters, which resolve multiple requests for a single resource. The implementation complexity of an N-port crossbar switch increases with N2 making crossbars impractical for systems with a very large number of ports. Therefore majority of high-performance switches and routers today have only a relatively small number of ports (usually between 8 and 32). This is because the high performance devices are used at aggregation points where port density is low. In the past few years, with the concept of Network-on-Chip communication architecture, N oC has attracted lot of attentionby providing higher bandwidth and higher performance architectures for communication on chip. NoC can provide simple and scalable architectures if implemented on reconfigurable platforms . This work, therefore, focuses on the implementation of various algorithms for an arbiter with low port density. The port density we selected in this work is 8-bit. In this paper we carried out implementation and synthesis of various scheduling algorithms used in arbiters. A variable priority arbiter known as round robin arbiter is selected and implemented using various algorithms.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

II. RELATED WORK

"Designing and implementing a fast cross bar scheduler," Crossbar switches frequently function as the internal switching fabric of high performance network switches and routers. However, for fairness and high utilization, a crossbar needs an intelligent, centralized scheduler. In this article, we describe the design and implementation of a scheduling algorithm for configuring crossbars in input queued switches that support virtual output queues and multiple priority levels of unicast and multicast traffic. We carried out this design for Stanford University's Tiny Tera prototype, 1, 2 a fast, label-swapping packet switch. Its scheduler, designed to configure a crossbar once every 51 ns, implements the ESLIP scheduling algorithm, which consists of multiple round-robin arbiters.

"The Tiny Tera:1 A Packet Switch Core," In this paper, we present the Tiny Tera: a small packet switch with an aggregate bandwidth of 320Gb/s. The Tiny Tera is a CMOS-based input-queued, fixed-size packet switch suitable for a wide range of applications such as a highperformance ATM switch, the core of an Internet router or as a fast multiprocessor interconnect. Using off-the-shelf technology, we plan to demonstrate that a very highbandwidth switch can be built without the need for esoteric optical switching technology. By employing novel scheduling algorithms for both unicast and multicast traffic, the switch will have a maximum throughput close to 100%. Using novel highspeed chip-to-chip serial link technology, we plan to reduce the physical size and complexity of the switch, as well as the system pin-count "high-performance multi-queue buffers for VLSI communication switches," Small nxn switches are key components of multistage interconnection networks used in multiprocessors as well as in the communication coprocessors used in multicomputers. The design of the internal buffers in these switches is of critical importance for achieving high throughput low latency communication. We discuss several buffer structures and compare them in terms of implementation complexity and their ability to deal with variations in traffic patterns and message lengths. We present a new design of buffers that provide non-FIFO message handling and efficient storage allocation for variable size packets through the use of linked lists managed by a simple on-chip controller. We evaluate the new buffer design by comparing it to several alterative designs in the context of a multi-stage interconnection network. Our modeling and simulations show that the new buffer outperforms its "competition" and can thus be used to improve the performance of a wide variety of systems currently using less efficient buffers.

III. PROPOSED SYSTEM

In, proposed system to focuses on the implementation of various algorithms for an arbiter with low port density. The port density we selected in this work is 8-bit. In this paper we carried out implementation and synthesis of various scheduling algorithms used in arbiters. A variable priority arbiter known as round robin arbiter is selected and implemented using various algorithms. The lists of PPE design are:

1. Exhaustive (Exh) programmable priority encoder
2. Barrel shifter based programmable priority encoder
3. Ripple carry based programmable priority encoder
4. Carry look ahead based programmable priority encoder

Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. As the term is generally used, time slices (also known as time quanta)[3] are assigned to each process in equal portions and in circular order, handling all processes without priority (also known as cyclic executive). Round-robin scheduling is simple, easy to implement, and starvation-free. Round-robin scheduling can also be applied to other scheduling problems, such as data packet scheduling in computer networks. It is an operating system concept. The name of the algorithm comes from the round-robin principle known from other fields, where each person takes an equal share of something in turn.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

A.ARBITERS:

In figuring out how to make the arbiters fast, we first observed that a round-robin arbiter is equivalent to a programmable priority encoder (PPE) plus some state to store the round-robin pointer. A PPE differs from a simple priority encoder in that an external input dictates which input has the highest priority. Next, we'll take a detailed look at how we arrived at a design for a high speed, round-robin arbiter.

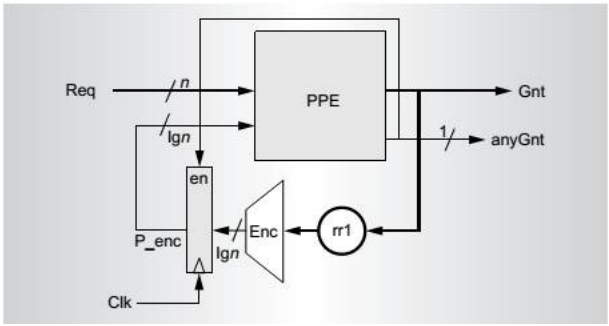


Figure 1: Block diagram of a round-robin arbiter.

Figure 1 shows a block diagram of a round robin arbiter. It has some state (the roundrobin pointer P_enc, of width lgn bits), which points to the current highest-priority input. In every arbitration cycle, it uses pointer P_enc to choose one among the n incoming requests, through a PPE.This PPE takes n 1-bit-wide requests and the lgn-bit-wide pointer (P_enc) as inputs. It then chooses the first nonzero request value beyond (and including) Req[P_enc], resulting in an n-bit grant. Clearly, this combinational block carries out the core function of contention resolution. The pointer-update mechanism is generally simple and can be performed in parallel. To minimize overall delay, we focus on minimizing the path from Req to Gnt, which is a pure combinational path passing through the PPE. Hence, the design of a fast round-robin arbiter reduces to designing a fast PPE. A fast PPE would work in any arbiter, regardless of the pointer update mechanism. The design of a fast, nonprogrammable priority encoder (smpl_PE) is well known. It is the presence of programmability that complicates the design of our PPE.

B.EXHAUSTIVE (EXH) PPE:

Figure 5 shows the first PPE design. It uses n duplicate copies of the simple priority encoder, smpl_PE, and uses the programmable priority input to select which priority encoder to use. In the figure, the multiplexer is an n-to-1 (n+1)-bit multiplexer, with P_enc as the select signal. The circles marked rri denote wiring connections (there is no logic) such that Req is rotated i positions to the right (the most significant bit is the leftmost bit) before being presented to the multiplexer. (The PPE could equivalently obtain anyGnt by a simple n-bit OR of the Req inputs. This observation is valid for all designs.)

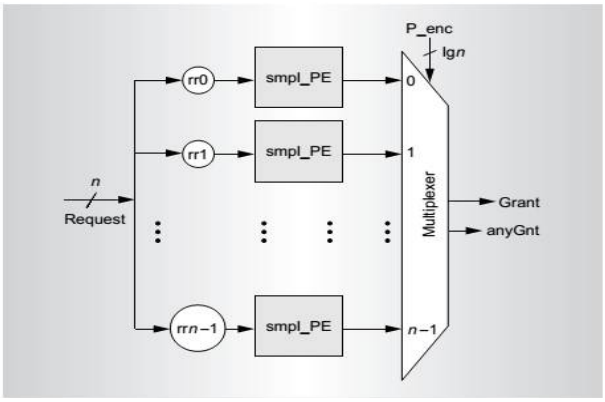


Figure 2: Design of the Exh PPE.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

This design is reasonably fast for small n: We can implement an n-way multiplexer in terms of a lgn-stage, 2-to-1 multiplexer tree, with total time delay = t(smpl_PE) + lgn × t(2:1 mux) and total area = n × area(smpl_PE) + area(n-way mux).

B.SHIFT ENCODER BASED PPE:

This is the most common design used in round-robin arbiters (see Figure 6). A barrel shifter first rotates the incoming requests by P_enc. It then rotates the output of one simple priority encoder by P_enc in the other direction to give the final output Gnt. This design simply transfers the programmability to the shifters. We can implement an n-bit barrel shifter by a series of lgn stages, stage i being a 2-to-1 multiplexer for each of the n bits, with P_enc[i] as the select signal.

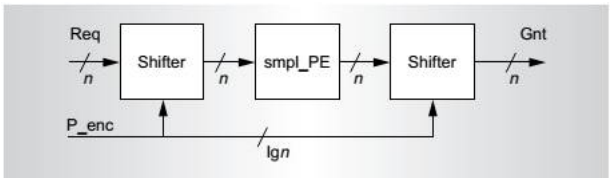


Figure 3: Design of the Shft_Enc PPE.

This design is slower than Exh because there are two shifters in the critical path. However, it is much better in terms of area. Still, the delay of 2 × lgn stages through the shifters dominates the critical path, limiting the speed at which this PPE design can operate.

C.BARREL SHIFTER:

A Barrel Shifter is a logic component that perform shift or rotate operations. Barrel shifters are applicable for digital signal processors and processors. This component design is for a natural size (4, 8, 16…) barrel shifters that perform shift right logical, rotate right, shift left logical, and rotate left operations depending on the instantiation parameters. The left and right operation is implemented through inversion of the input and output vectors, so the basic multiplexing function can perform both operations. The number of multiplexing stages is relative to the width of the input vector.

D.RIPPLE CARRY AND CLA BASED PPE:

These two designs are both based on the following observation: The first nonzero request beyond the current pointer value can be found in a series of at most n sequential operations. First, let's look at input P_enc. If Req[P_enc] is 1, then Gnt[i] will be 1 for i = P_enc and 0 otherwise. If, on the other hand, Req[P_enc] is 0, we look at the input numbered (P_enc + 1)(mod n), and continue this process until either we find a nonzero request or we are back to the input number we started from (P_enc). In the latter case, none of the input requests were 1, so Gnt[i] will be 0 for all i, and anyGnt will be 0. A subblock that we call mini_RPL (see Figure 7) carries out each step.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

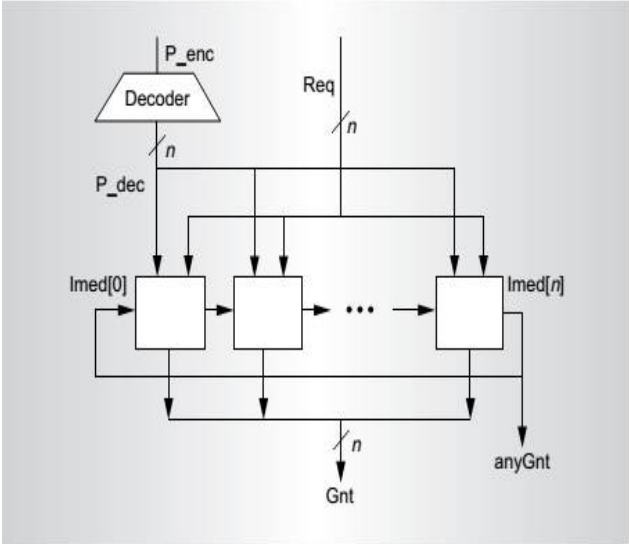


Figure 4: Design of the Ripple PPE.

The PPE obtains P_dec by decoding the input signal P_enc. The 1-bit signal Imed[i] indicates whether the search process has already found a 1; if so, Req[i] is ignored, Gnt[i] is 0, and Imed[i + 1] is 1. Imed[i] is ignored if P_dec[i] is 1, which means that this is the top-priority request and the starting point of the search process.One way to speed up the Ripple design is to use carry-look-ahead; we call the design that does this CLA. For example, a carry-look-ahead of 16 would decrease the rippling delay to about one-sixteenth of the original Ripple delay. We can trade off area and time in deciding the exact amount of look-ahead. However, we have found that the resulting design has two fundamental problems. First, the design has a combinational feedback loop. In fact, the loop is inherent to Ripple-based PPEs.

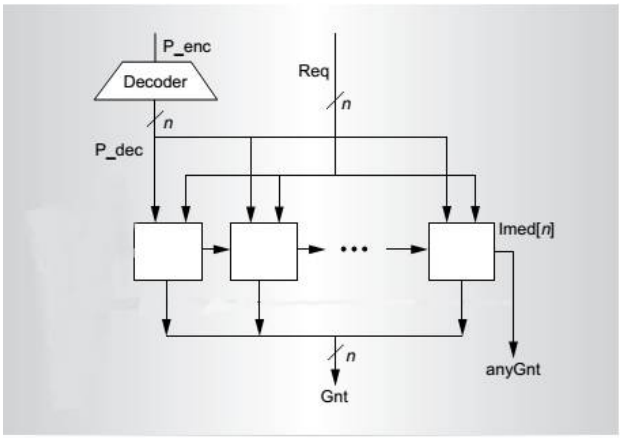


Figure 5: Design of carry look ahead based PPE

Advantages:

- Less power consumption.
-

Synthesis and Implementation

A.Methodology.

The implementation of this work is targeted for spartan6 FPGA family. Only LX series has been considered as it is appropriate for general logic applications. The implementation is carried out for input request of 8-bit

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

corresponding to the 8- bit grant output. The parameters considered are resource utilization, timing and average power dissipation. Resource utilization is considered in terms of on chip FPGA components used. Timing refers to the clock speed of the design and is limited by the setup time of the input and output registers, clock to the output time associated with the flip flops, propagation and routing delay associated with critical paths, skew between the launch and capture register. Timing analysis is carried out by providing appropriate timing constraints. The average power dissipation mainly composes of dynamic power dissipation besides negligible static power dissipation. In order to provide a fair comparison of the results obtained the test bench is provided with same clock period and same input statistics in all the topologies implemented. The constraints related to the period and offset are duly provided in order to ensure complete timing closure. The design synthesis, Implementation and mapping have been carried out in Xilinx ise 12.4 [15]. Power metrics are measured using Xpower analyzer. The simulator data base is observed to obtain clock period and operating frequency of the implemented design.

IV. EXPERIMENTAL RESULTS.

Resource utilization is actually FPGA resource utilization and Table I gives the comparison of the resource utilization of various scheduling algorithms implemented using various topologies discussed. The various resources utilized and thetotal resource utilization are plotted in figure 6. It can be seen that scheduling algorithm implemented using Exch topology based PPE utilizes the maximum on chip resources while as the ripple carry shift topology based PPE utilizes the least on chip resources. Table 2 provides a comparison of maximum achievable clock rates post implementation for an 8-bit input request. From table 2, we can see that the scheduling algorithm implemented using Barrel shift topology based PPE can operate at maximum frequency of 238.607 MHz, while as ripple carry based PPE proves to be slowest. Finally static and dynamic power dissipation for different scheduling algorithms is considered. The static power dissipation in an FPGA consists of device static power dissipation and the design static power dissipation. Design static power although is very small percentage of dynamic power dissipation. The dynamic power dissipation is function of input voltage y2. the clock freq (felk), the switching activity (a), load capacitance (CL) and number of elements used. Theanalysis was done on a constant supply voltage and at maximum operating frequency for each structure. To ensure a reasonable comparison, diverse input test vectors in post route simulation were selected to represent worst case switching activity. The physical constraint file and design node activityfile captured during the post route simulation were used for power analysis in Xpower analyzer tool. Table3. Shows the power dissipated in various algorithms. The power dissipated in the clocking resources varies with the clocking activity (clock frequency) as provided in the PCF. The □ower-delay product measured for the various schedulingalgonthms are plotted in figure 7. Similarly the area delay product is plotted in figure 8 below. It can be seen from figure 7 and figure 8 that barrel shifter based topology measures the minimum in the two parameters.

This paper carried out the performance analysis of various scheduling algorithms. The implementation was targeted for Spartan6 FPGA device. The resulting structures showed differences in the way of using resources available in the target FPGA device. The Exch topology based PPE uses the resources extensively and has large power delay product and largest area delay product metric. On the other hand the CLA based topology based PPE uses lesser resources than any other algorithm considered and also shows relatively lesser power
delay product and lesser area delay product metric. To sum up a judicious trade-off between area, power and throughput parameters, and the intended application will determine the correct approach for implementing the scheduling algorithms. Since the area delay product and the power delay product of
the design targeted for FPGA platforms are considered as important parameters. Hence implementation of above scheduling algorithms for on-chip communication architectures, for FPGA based platforms revealed that Barrel based scheduling as reasonably appropriate among the ones considered, for Network-on-chip (NoC) communication architectures.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

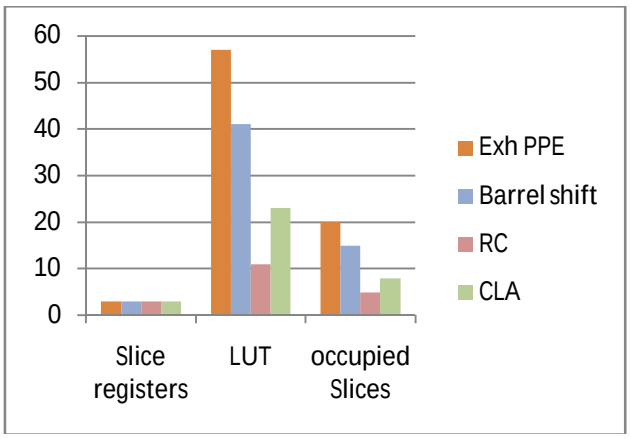
An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.



On chip Resource	Encoding topology			
	Exch	*Barrel shift*	*Ripple carry*	*CLA*
No of Slice registers	43	3	8	16
No of Slice LUTs	40	27	17	26
No. of occupied Slices	18	9	6	8

Table I. Resourse Utelizatlon Comparison For Different Encoding Topology

Timing Prameter	Encoding topology			
	Exch	*Barrel shift*	*Ripple carry*	*CLA*
Maximum frequency (MHz).	148.17	238.607	138.87	201.207
Min available offset-in (ns).	0.118	4.264	3.77	5.242
Min available offset-out (ns).	14.445	11.588	14.151	12.07

Table Ii. Timing Comparison Of Scheduling Algorithms Based On Different Encoding Topologies

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

FPGA Resource	Power dissipation (mW)			
	Encoding topology			
	Exch	*Barrel shift*	*Ripple carry*	*CLA*
Clock	0.49	0.85	0.85	0.91
logic	0.30	020	0.37	0.49
Signals	0.58	0.79	0.65	0.59
I/Os	23.67	73.13	28.18	20.75
Dynamic	25.05	74.97	30.05	22.74
Quiescent	13.95	14.48	14.00	13.92
Total	**38.99**	**89.45**	**44.05**	**36.67**

Table Iii. Power Dissipation For Dllfrent Scheduling Algorithms On Spartan6

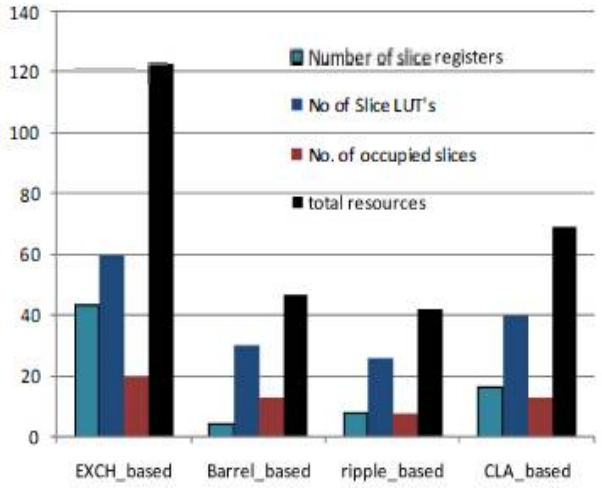


Figure 6. Plot of total resources used in various scheduling algorithms.

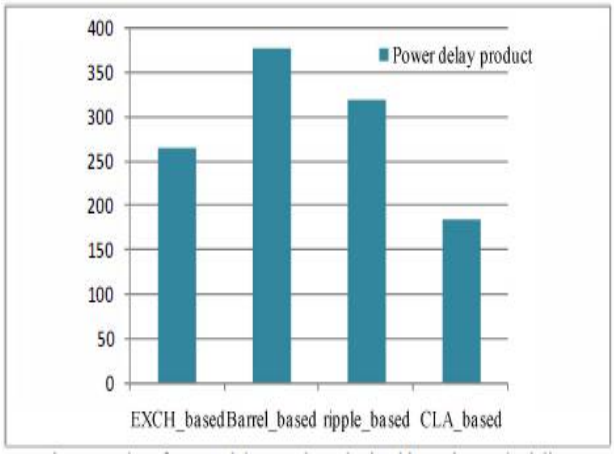


Figure 7. Plot of power delay product obtained in various scheduling algorithms.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

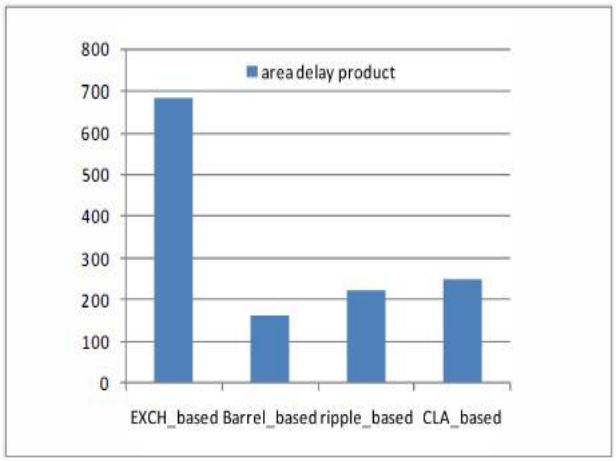


Figure 8. Plot of Area delay product in various scheduling algorithms.

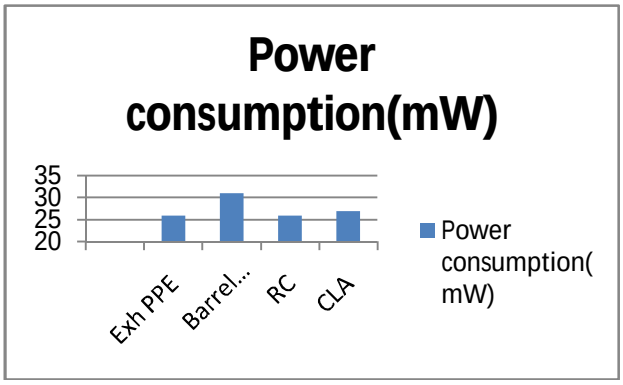


Figure 9. Plot of Power consumption

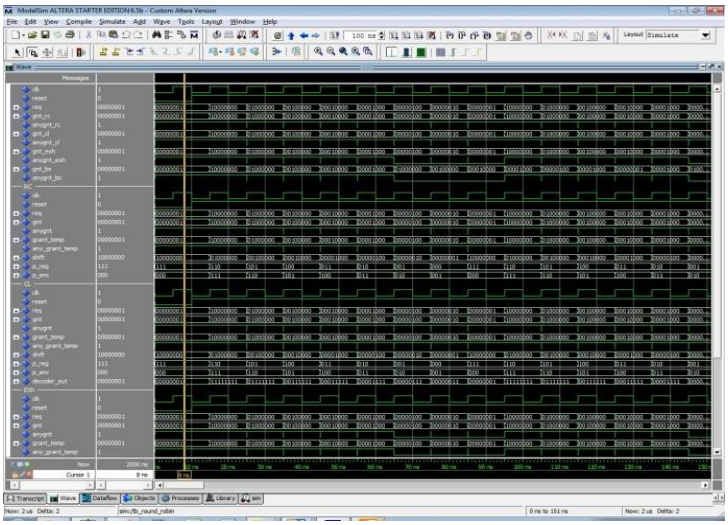


Figure 10.Simulation output

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

REFERENCES

[1]Ahmad R., Ismail W. (2017) Performance Comparison of the Improved Power-Throughput AES and Blowfish Algorithms on FPGA. In: Ibrahim H., Iqbal S., Teoh S., Mustaffa M. (eds) 9th International Conference on Robotic, Vision, Signal Processing and Power Applications. Lecture Notes in Electrical Engineering, vol 398. Springer, Singapore.

[2] Yuan-Ying Chang, Huang, Y. S.-c., Poremba, M. Narayanan, V.Yuan, Xie King, C, 'Title TS-Router: On maximizing the Quality-ofAllocation
in the On-Chip Network," in IEEE 19th International Symposium on High Performance Computer Architecture (HPCA2013),
pp 390-399, Feb 2013.

[3]Varis, N., & Manner, J. (2011). Performance of a software switch. In *Proceedings of IEEE HPSR 2011* (pp. 256–263).

[4]Thakur J, Kumar N (2011) DES, AES and Blowfish: symmetric key cryptography algorithms simulation based performance analysis. Int J Emerg Technol Adv Eng 1(2).

[5]B. Phanibhushana, K. Ganeshpure, S. Kundu, "Task model for on-chip communication infrastructure design for multi core systems," in proc of IEEE 29th International Conference on Computer Design (ICCD), pp 360-365, oct 2011.

[6] Akram Ben Ahmaed, Abderazek Ben Abdallah, Kenichi, Kuroda, "Architecture and Design of Efficient 3D Network-on-Chip (3D NoC) for custom multi core SoC," in International confrence on Broadband, Wireless Computing, communication and Application, FIT,Fukuoka, Japan, Nov 2010.

[7]Jin Ouyang, Yuan Xie, "LOFT: A High Performance Network-on-Chip Providing Quality-of-Service Support, " in 43rd Annual IEEE/ACM
International Symposium on Microarchitecture (MICRO),pp 409-420,
Dec 2010.

[8]B. Osterloh, H. Michalik, B. Fiethe, K. Kotarowski, "SoCWire: A Network-on-Chip Approach for Reconfigurable System-on-Chip Designs in Space Applications," in proc of NASA/ESA Conference on Adaptive Hardware and Systems, pp 51-56, june 2008.

[9] J. Guo, J. Yao, Laxmi Bhuyan, "An efficient packet scheduling algorithm in network processors," in proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies ,pp 807- 818, march 2005.

[10]K. Lee, Se-jee Lee, hoi-jun Yoo, "A Distributed Crossbar Switch Scheduler for On-Chip Networks," in Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE ,pp 671-674, Sept. 2003.

[11] William John Dally, Brain Towels, Principles and Practices of Interconnection Networks, 1st ed. Morgan Kaufmann publications, 2003.

[12] N. McKeown, V. Anantharam, and 1. Walrand, "Achieving 100% throughput in an input-queued switch," in Proc. IEEE INFOCOM '96,
San Francisco, CA, pp. 296-302.

[13] P. Gupta, N. Mckeown, " Designing and implementing a Fast Crossbar Scheduler," in proc of Micro, IEEE, vol 19, no 1, pp 20-28, Feb 1999.

[14] http://www.xilinx.com.