congestion can start to build and spread from a congested switch to source nodes, which grows into a congestion tree. It highly demeans the overall system performance, especially in real-time applications with strict latency requirements.

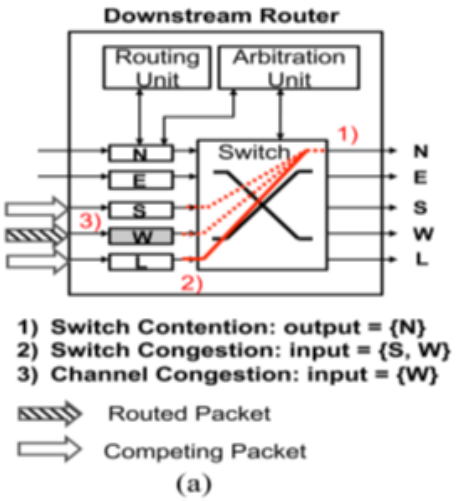

Figure 2: Switch contention, switch congestion, and channel congestion

Congestion-aware adaptive routing selects an output channel based on various types of network congestion information. Consequently, these selection functions can adjust path selection based on a time-variant congestion status. Congestion aware adaptive routing adopts two types of spatial information: local information and regional information. Local routing information considers local information such as the downstream buffer count and available flit slot to assess the traffic status

4. Block Diagram

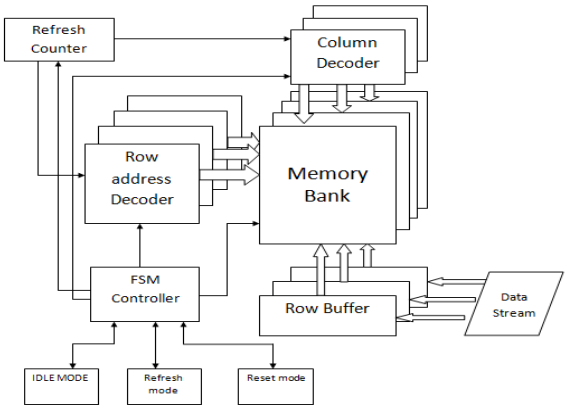

Figure 3: block diagram

Nios II is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGAs. Nios II architecture is a RISC soft core architecture which is implemented entirely in the programmable logic and memory blocks of Altera FPGAs. User-defined instructions accept values from up to two 32-bit source registers and optionally write back a result to a 32-bit destination register. DRAM plays vital role for system design such as RAM, cache memories etc. To increase row buffer locality, existing paper introduced a Thread row buffer to increase throughput and DRAM's overall performance.TRB increases row hit rate by reusing row that consists of same information's. Data access between DRAM banks are controlled by a logic controller consists of sequential elements. Logic controller is mostly accessed by every blocks in DRAM circuitry. Memory controllers contain the logic necessary to read and write to DRAM, and to "refresh" the DRAM. Without constant refresh, DRAM will drop off the data written to it as the capacitors leak their charge within a fraction of a second.Reading and writing to DRAM is performed by selecting the row and column data addresses of the DRAM as the inputs to the multiplexer circuit,wherethe demultiplexer on the DRAM uses the converted inputs to select the correct memory location and

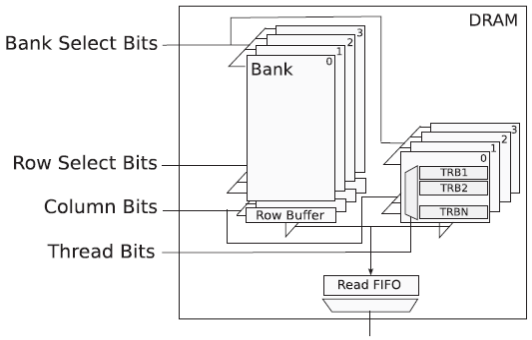

Figure 4: DRAM Bank

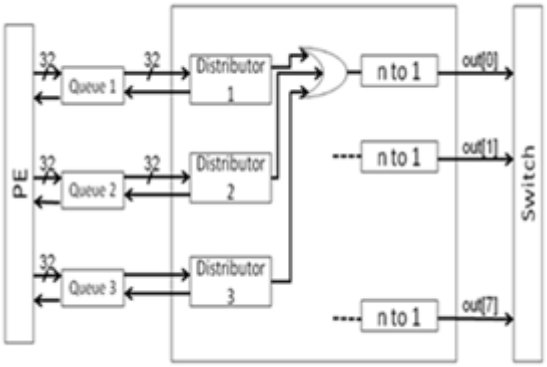

Figure 5: Network Interface

return the data, which is then passed back through a multiplexer to consolidate the data in order to reduce the required bus width for the operation.

5. Network Interface

The Network Interface (NI) is the module that acts as an abstraction layer in between the network protocol and internal UDN/MDN switch protocols. There are two types of NI: Input NI (INI) and Output NI (ONI). NI consists of a packetization unit (PU), a de-packetization unit (DU) and PE interface. NI is located between a router and a PE, decoupling the communication and computation. When a packet is injected into the switch, INI encapsulates (packetizes) them into U(M)DN packets, and carry them to the next router, in equally sized . ONI, in return, receives the U(M)DN flits, strips (depacketizes) the original packet, and ejects it from the switch. It offers a memory-mapped view on

all control registers in NI. That is, the registers in NI can be accessed using conventional bus interface. N to 1 bit serializers – one for each outgoing wire. Data Distributor to send data from output queues to one of the serializers. Each distributor can send data to each of the serializers. Not all the distributors are loaded all the time a single distributor can serve all the serializers. To that particular sending channel. As input to the 32-bit to 1-bit serializer, there is one OR gate to allow all the data distributors to be able to forward data to each serializer. The other OR gate and 1-bit output are hand-shaking signals between the data distributor and the serializer.The advantage of the new NI design is area savings.

6. Argo and Falp Method

In this paper we describe the use of asynchronous routers in a time-division-multiplexed (TDM) network-on-chip (NOC), Argo, that is being developed for a multi-processor platform for real-time systems.TDM need a common time reference, and existent TDM-based Network on chip designs are either synchronous or mesochronous. We use asynchronous to accomplish a simpler, smaller and more robust, self-timed design. Our design provide the fact that pipelined asynchronous circuits also act as ripple FIFOs. Thus, it keep off the need for explicit synchronization FIFOs between the routers. Argo has interesting timing properties that allow it to tolerate skew between the network interfaces (NIs).

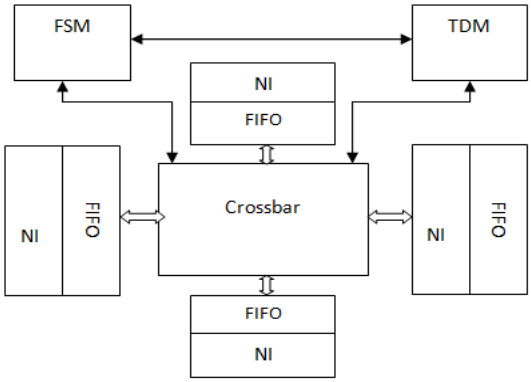

Figure 6: Argo Block Diagram

The paper presents Argo NOC-architecture and provides a quantitative analysis of its ability of absorb skew between the Nis by using a signal transition graph model and realistic component delays. Network-on-Chip (NoC) are known as the future communication infrastructure for many-core systems. They are capable of malfunction in the presence of the faults as technology sizes reduce proportionally the performance of the system. According to the results, links have failed 71% due to the Crosstalk fault. A new fault-adaptive and low power, calling FALP, method for Network on chip routers is presented in this article to extenuate their unexpected behavior through links. It reduces the switching power consumption overhead by keep track of the frequency and life time of faults. However, based on the VHDL execution of a Network on chip router, routing unit has a trifling area comparing to other components of an Network on chip router. In router architecture less than 11% of an NoC router area is occupied by routing component embedded with FALP technique.

7. Simulation Result

In this paper the analysis is based on typical and constant gate delay values. In fig 11 shows the output waveform of existing system. and fig 12 shows the waveform of proposed system. We have explored the use of min-max delay intervals and it leads to higher throughput values .the below table shows the performance evaluation table and therefore reduced skew tolerance.

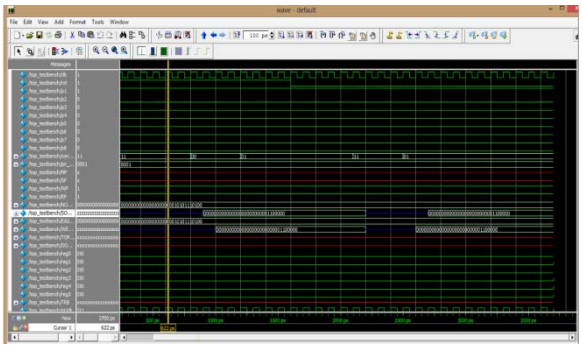

Figure 7: Output waveform for existing system

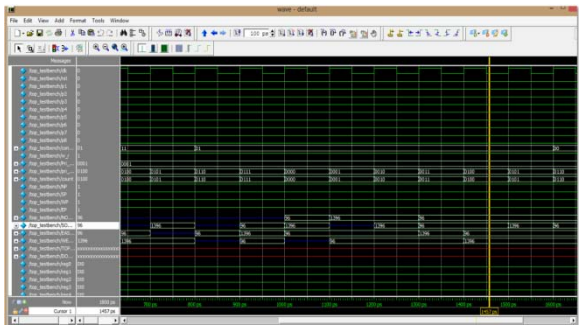

Figure 8: output waveform for proposed system

PARAMETERS	EXISTING SYSTEM	PROPOSED SYSTEM
AREA	1281	388
DELAY	0.0044ms	0.0041ms
TOTAL POWER	87.42mw	74.89mw
THROUGHPUT	7152	7744

Figure 9: Performance Evaluations Table

8. Conclusion

In this paper we emulated the performance of Path congestion aware routing and Argo on quartus II platform. The paper extended previous work on synchronous and mesochronous TDM-based NOCs by exploring the use of asynchronous routers that allow a truly GALS-style implementation of a NOC-based multi-core platform. We compared the performance of these two routing method in terms of number of resources utilized, throughput , area , power and delay. Path congestion aware routing consumes

more resources, which mean it utilizes more silicon area. Argo has high clock frequency than the Path congestion aware routing, which means Argo could process data more quickly. In this design, we should make a trade-off among the resource or silicon area, maximum clock frequency and delay and choose suitable arbitration mechanism according to that. We can propose a novel fault-tolerant method has been proposed to improve the reliability of NoC routerlinks.

This method is composed of fault detection and fault tolerant techniques, which is implemented in different zones of the network. We have not yet explored this property.

References

[1] En-Jui,Chan Hsien KaiHsin,(2014) ”Path Congestion Aware Adaptive Routing With a Contention Prediction Scheme for NOC Systems”, IEEE Transactions on computer-aided design of integrated circuits and systems, VOL. 33,pp 113-126.
[2] Ghiribaldi.A,Bertozzi.D,BNowick.S.M, (2013) “A transition-signaling bundled data NoC switch architecture for cost-effective GALS multicore systems,” Proceedings - Design, Automation, and Test in Europe Conference and Exhibition, pp. 332–337.
[3] Kasapaki.E, Sparsø.J, Sørensen.R, and Goossens.K,(2013) “Router Designs for an Asynchronous Time-Division-Multiplexed Network-on-Chip,” in Proc. of Euromicro Conference on Digital System Design (DSD), pp. 319–326.
[4] Panades.I.M,Greiner.A,Sheibanyrad.A, (2006)“A low cost network-on- chip with guaranteed service well suited to the GALS approach,” in 1st International Conference on Nano-Networks (Nano-Net), pp. 1–5..
[5] Schoeberl.M, Brandner.F, Sparsø.J, and Kasapaki.E,(2012) “A Statically Scheduled Time-Division-Multiplexed Network-on-Chip for Real-Time Systems,” in Proc. IEEE/ACM Intl. Symposium on Networks-on-Chip (NOCS). IEEE Computer Society Press, pp. 152–160.
[6] Sparsø.J,Kasapaki.E, Schoeberl.M,(2013) “An Area-efficient Network Interface for a TDM-based Network-on-Chip,” in Proc. Design Automation and Test in Europe (DATE), pp. 1044–1047.