INTERNATIONAL RESEARCH JOURNAL ON ADVANCED SCIENCE HUB
e-ISSN : 2582 - 4376
Open Access
RSP SCIENCE HUB
(The Hub of Research Ideas)
Available online at www.rspsciencehub.com

Special Issue of Second International Conference on Advancements in Research and Development (ICARD 2021)

Evaluation of Bufferless Network-On-Chip with Parallel Port Allocator

Cecil C Nachiar[1], *Dr.S.Sumathi* [2]
[1]*PG student, Dept. of ECE, Adhiyamaan College of Engineering,Hosur,Tamilnadu,India*
[2]*Head of the Department, Dept. of ECE , Adhiyamaan College of Engineering,Hosur,Tamilnadu,India.*
cecilchinnaraj@gmail.com[1]

Abstract

Multicore network-on-chip when scales upto hundred of nodes, energy consumption, design complexity and cost increases multifold owing to structure of interconnect. Many researches are being conducted to design novel architecture to build efficient networks-on-chips. Our paper proposes efficient bufferless design with deflection containment technique to eliminate buffers and latency. The high cost of buffers motivate us to go for bufferless design, however with increasing network loads, it become notorious with multiple deflection and flit loss between nodes. To overcome this, we have designed a bufferless architecture with local bypass ring within nodes to reduce deflection and packet loss. Deflection Containment with the use of local bypass ring shortens critical path and improves performance. Architecture of our designed bufferless NoC is analysed and RTL implementation of its components is done with Xillinx ISE design suite and its working is analysed in Modelsim SE. Our evaluation proves that bufferless routing with deflection containment technique reduces power dissipation without compromising on its performance.

Keywords: Multicore NoC ,Bufferless design, Deflection containment ,critical path

1. Introduction

Interconnection fabric becomes important design parameter in NoC on connecting on chip components. This ranks higher than traditional bus interconnection in scalability and bandwidth [1]. In design of NoCs, buffers consume more power and occupy larger area leading to high cost. Figure 1 presents the architecture of NoC with multicore system. As shown it connect all nodes within a chip with network interface and routers. Recent works on designing low cost NoCs by eliminating buffers has been discussed [2-10]. In case of buffered network whenever there is a congestion or destination port of that is busy, the packet remains idle in buffer and waits for acknowledgement to transmit. In this process we see that link bandwidth is not used unnecessarily. However in bufferless NoCs only pipeline registers are present which can hold only single flit of information. All the incoming flits are transmitted through pipeline registers and whenever destination port of particular flit is busy, it is routed to nearby port which is free.

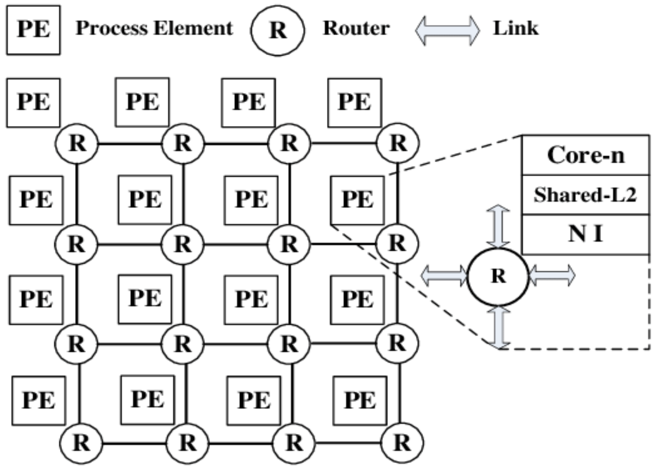


Fig.1.Typical NoC Architecture

When nearer ports are busy ,then it sends packet to whichever port is free irrespective of its distance to the destined port. So, whenever core increases packet injection rate also increases making deflection in bufferless router cumbersome reducing its performance [3]. Figure 2 shows the architecture of buffer and bufferless NoC.

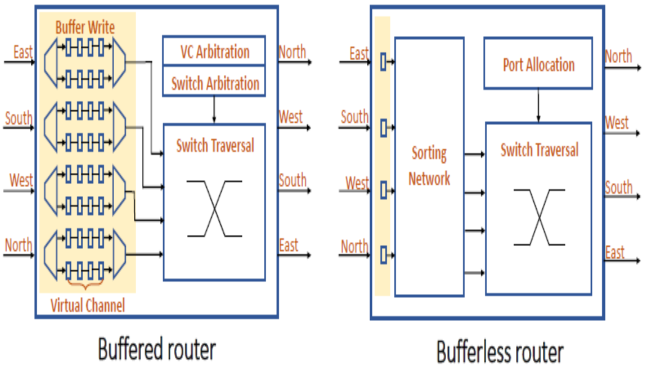


Fig.2.Architecture of Buffered and Bufferless NoC

However, various subsequent evaluation has brought out problems with adoption of bufferless routers [4,5]. Flit ranking and port allocation used so far in all designs has increased critical path delay as flits are allocated ports sequentially. Flint ranking method to avoid livelock during transmission such as oldest first etc shown in Table 1 has increased complexity in bufferless router design. As shown in Figure 3, Whenever network load increases , deflection rate increases causing flit contention reducing its performance. This leads to unnecessary multiple hops to a packet to reach its destination making deflection containment a stumbling block for building efficient routers as shown in Figure 3.

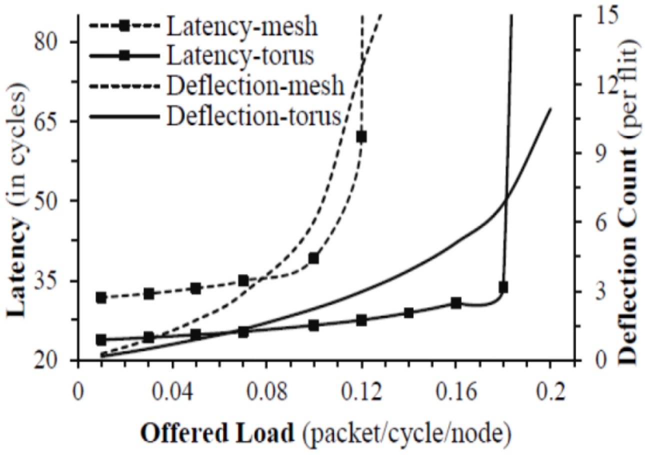


Fig.3. Performance of bufferless NoC with increasing loads.

ABNoC(Approximate Bufferless) [6], is a recent work in bufferless NoCs where network conflicts has been reduced and packet transmissions are done using approximate allocation mechanism. Though it improved latency and bandwidth utilisation as shown in Figure 4, with retransmission mechanism it makes the design more complex compared to other design.

Table.1. Various Flit ranking rules

Flit Ranking Rules	Comments
Oldest First	Older flits go before younger flits
Closest First	Flits closer to their destination before flits whose remaining distance is larger
Most Deflections First	Flits that have been deflected more before, flits deflected fewer times.
Round Robin	Flits from different input ports are ranked in round robin fashion.
Hybrid Policy	In odd cycles use Older first and for even cycles Round robin

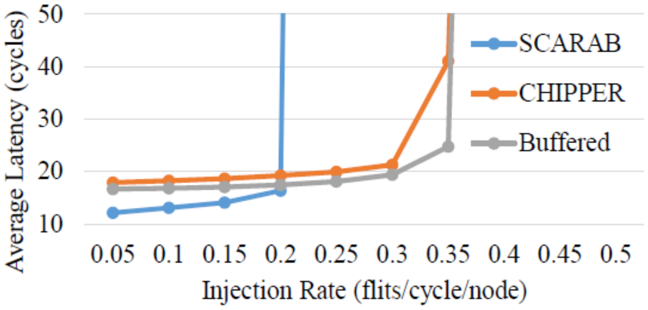


Fig.4.Average latency of Various Bufferless NoCs

BLESS (Bufferless NoCs) [7] was the first proposed bufferless NoC to eliminate buffers in conventional NoCs. Here each node, consists of an injection buffer and a reassembly buffer and when more than one flit contends for the same output, only oldest flit is transmitted and remaining flits goes to the port whichever is free. However, this age based priority used in BLESS is more deterministic on latency and expensive. CHIPPER [8], low complexity buffer has simplified router micro architecture by removing in router buffers and crossbars. It is based on Golden packet and retransmit once algorithms. The reassembly buffer size plays very important role in determining its efficiency which makes it less desirable. Ring based deflection Routers [9,10] is desirable for bufferless operation as once flits enters the ring it simply reaches its destined port however complicates traffic management. MinBD[11], DeBAR [12] and SLIDER [13] employs side buffers with two stage pipeline registers. The side buffers used in this design holds the flits that are to be transmitted to busy ports and get reinjected later to artbitrate to required output port.AFC[13] and FLEXI [14] uses a power gating technique to switch to bufferless mode along with conventional input buffering technique.

2.1Microarchitecture Of Proposed Bufferless Router

We have overcome the challenges in previous methods and made slight changes in their architecture which drastically improve the performance of the router. Figure 5 shows the proposed router architecture. It uses two stages pipeline where destination ports of incoming flits are calculated using conventional routing logic are calculated at the initial stage. Once destination ports of flits are known, they are passes to sorting network logic where flits are ranked based on priority. Sorting logic is designed in such a way that top channel is given highest priority. Once flits are sorted , it enters the second stage where port allocation is done in parallel depending upon the availability of destination ports. Unlike conventional bufferless NoCs , instead of deflection to other possible nodes we are sending contending flits to local bypass channel formed within subnetwork of that node. It contends for destined port after each clock cycle till it gets its turn for transmission. Injection and ejection are done till channels are occupied and winning flits moves to destined port over crossbar.

2.2 Flits Ranking By Priority

Flits ranking by priority becomes the first stage in arbitration process which is done by sorting network as shown in Figure 5. Flits form four ports namely West, East ,South and North are prioritised not including flits in bypass port for reducing hardware complexity. Since four ports are considered we are considering 2x2 partial permutation network with bitonic sorting. It is a parallel sorting algorithm where compare and exchange operation is done. Sorting is done such a way that top channel is with highest priority values. As shown in Figure 6, inputs a,b,c and d are inputs from west, east, north and south ports on sorting yield the output w,x,y and z. Example of bitonic sorting is shown in Figure 7.

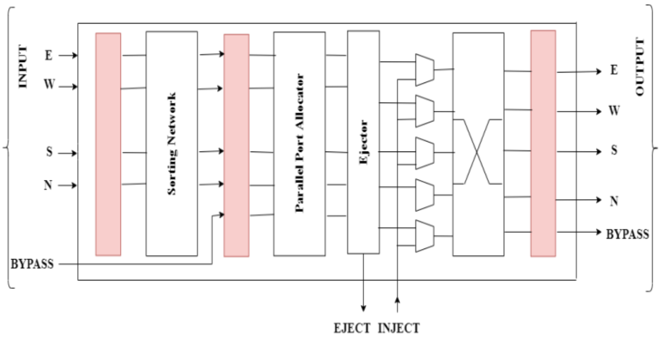


Fig.5. DeC Router micro architecture

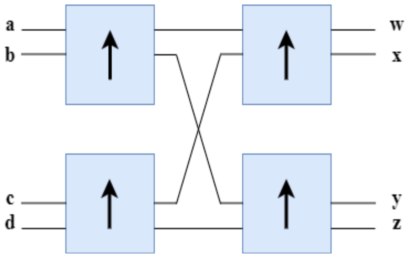


Fig.6. Partial permutation Network

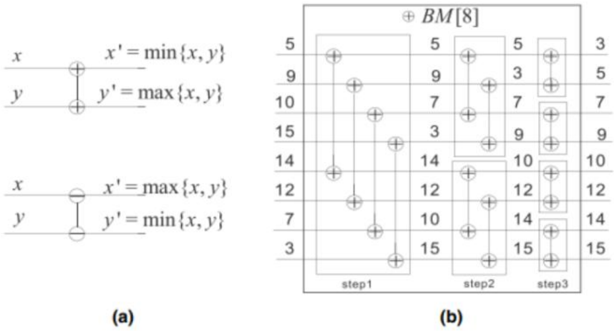


Fig.7.Bitonic sorting algorithm

2.3 Parallel Port Allocator

In earlier design, flits are allocated sequentially one by one in an order increasing delay in total transmission as shown in Figure 8. To overcome this, we are using now parallel port allocator where ports are assigned in parallel manner. This can be proved with RTL synthesis in results and discussion. It strictly follows two steps as shown below -

Step 1: Check for **status** of the destined port of particular flit. If that destined port is free without contention, then particular flit is transmitted to the destined port.

Step 2: When **the** destined port of particular flit is busy it follows Step 2. Depending upon all other flits and their destined port , available ports are calculated .With this we are building simple look up table to calculate the allocated port. Here we have designed it to have a predefined order Bypass, North, South ,East and West as shown in Figure 9.First available port is given to the flit present in the lower channel.

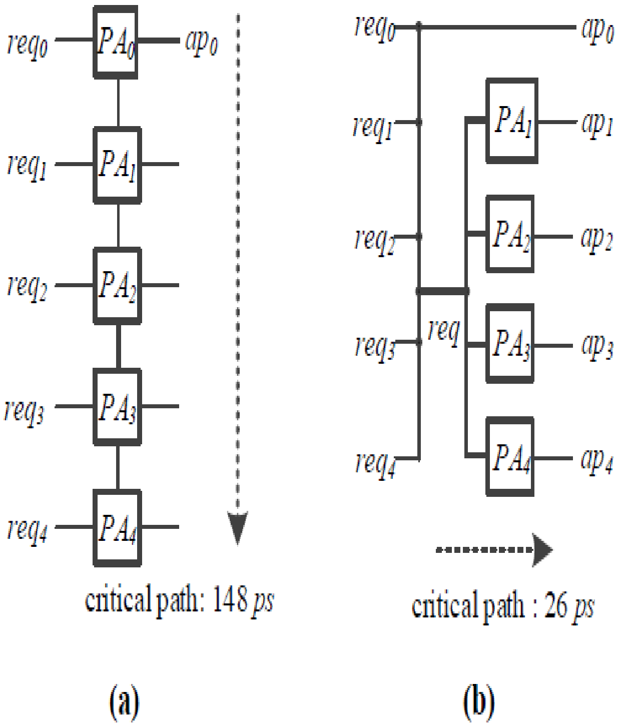


Fig.8.Port allocation (a) BLESS (b) Parallel port allocator

Consider an example where flit f_0 is to be moved to south, flit f_1 is to be moved to north f_1, flit f_2 is to be moved to east f_2, flit f_3 is to be moved to north. As per the flowchart in Figure 9, it first checks for non-contending ports. Here find f_0 and f_2 are non-contending and hence these flits are transmitted to south and east ports respectively. We also find flits f_1 anf f_3 are contending for north port. As per our design first contending flit f_0 moves to bypass and and f_3 moves to north port.Thus all flits get transmitted to destined port.

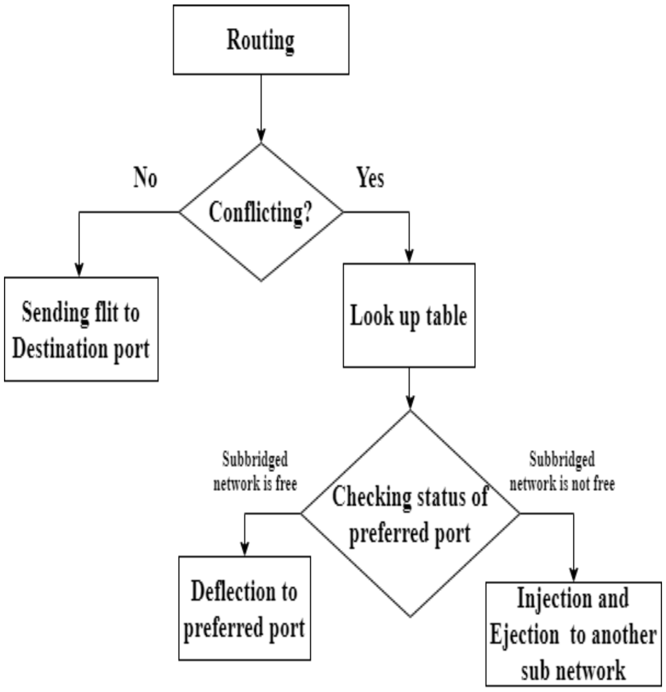

Routing
Conflicting?
No
Yes
Sending flit to Destination port
Look up table
Checking status of preferred port
Subbridged network is free
Subbridged network is not free
Deflection to preferred port
Injection and Ejection to another sub network

Fig.9.Flowchart of parallel port allocator

Consider an example where flit f_0 is to be moved to south , flit f_1 is to be moved to north f_1, flit f_2 is to be moved to east f_2, flit f_3 is to be moved to north. As per the flowchart in Figure 9, it first checks for non-contending ports. Here find f_0 and f_2 are non-contending and hence these flits are transmitted to south and east ports respectively. We also find flits f_1 anf f_3 are contending for north port. As per our design first contending flit f_0 moves to bypass and and f_3 moves to north port.Thus all flits get transmitted to destined port.

2.4 Ejection and Injection

So far we have discussed transmission and reception of data along four ports. As discussed local bypass ring calls for fifth port bypass port. To avoid hardware complexity local ejection and injection are done separately from crossbar operation. With every increasing cycle, data is ejected out and this continues till space in sub network i.e ports exist and if several ports are are free during transmission it follows the given order. To make our design more contention free we have used ejection before injection.

2.5 Deflection using local bypasss ring

To reduce the critical **delay** we partition a network present in every node to multiple sub networks bridged by bypass ring as shown in Figure 10. Partition is done in such a way that datapath width remains unchanged. This increases network path diversity reducing critical path delay. As shown in

Figure 10, each node consists of M subnetworks which are of same datapath width connected through local bypass ring which is unidirectional .Here we have used 2 sub networks for simplicity M= 2.

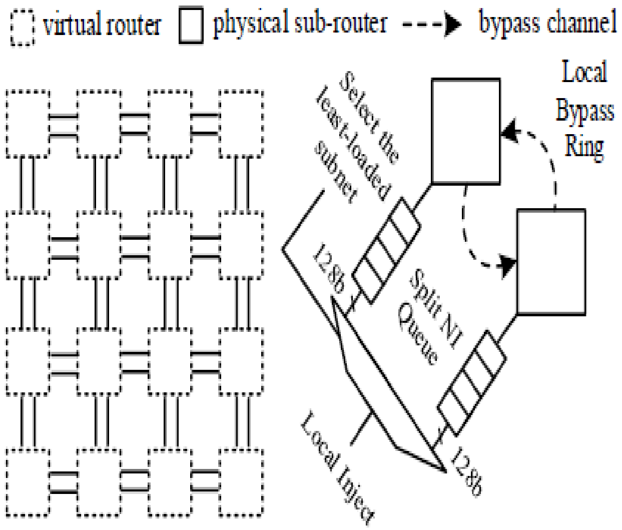


Fig.10. Deflection using local bypass ring with two sub networks

A contending flit which is to be deflected in conventional NoCs will be ejected to subnetwork through local bypass ring reducing multiple hops and improving latency.As shown in Figure 11, flits f_0 and f_1 are contending for east port(3) . Incase of BLESS f_0 is routed to east port while f_1 is routed to north port which again takes 2 hops to reach east port (3).So totally it takes 3 hops for complete data transmission f_0 and f_1. In our design, f_0 will be routed to east port and f_1 will be routed to bypass port which again in next cycle sends f_1 to east port totalling of 2 cycles of data transmission. Thus improving path diversity and latency of our proposed architecture.

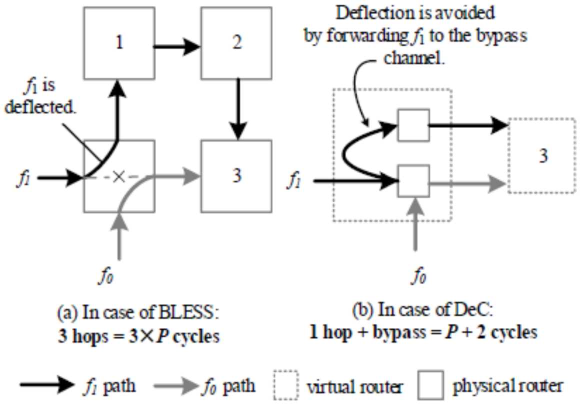
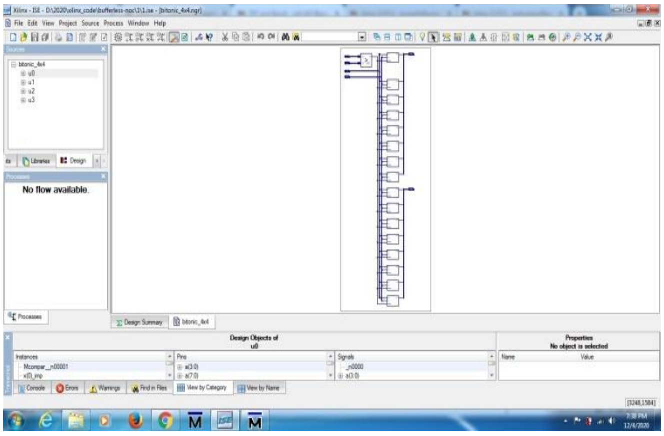

1
2
3
fi is deflected.
fi
fo
Deflection is avoided by forwarding fi to the bypass channel.
3
fi
fo
(a) In case of BLESS: 3 hops = 3×P cycles
(b) In case of DeC: 1 hop + bypass = P + 2 cycles
fi path
fo path
virtual router
physical router

Fig.11.Reducing hops with local bypass ring

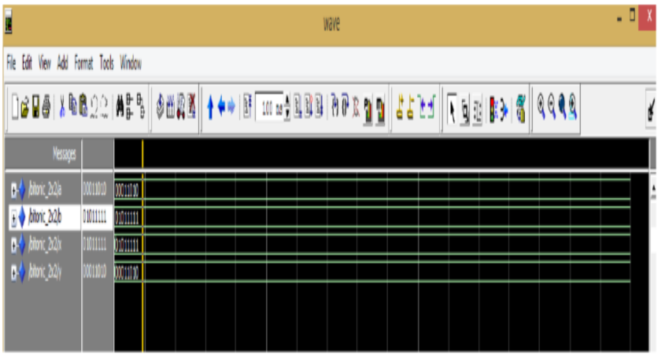
3. Results and Discussion

The proposed design is verified in ModelSim SE 6.3f .Here we have analysed flits ranking with Bitonic sorting , flits prioritisation with parallel port allocator. As discussed above the RTL implementation of 2x2 bitonic sorting is shown in Figure 12 while the output of 2x2 bitonic sorting is shown in Figure 13. Here we have taken 8 bits in which last 4 bits contains the priority bits. Two inputs “00011010” and “01011111” are given as inputs to a and b. By bitonic sorting , priority bits “1010” and “1111” are compared and output x is given with higher value (01011111) and output y is given with lower value (00011010).



No flow available.

Fig.12. RTL schematic of 2x2 bitonic sorting



Messages		
/bitonic_2x2/a	00011010	00011010
/bitonic_2x2/b	01011111	01011111
/bitonic_2x2/x	01011111	01011111
/bitonic_2x2/y	00011010	00011010

Fig.13. Simulation of 2x2 bitonic sorting

As seen in router micro architecture partial permutation network for flits priotization is built with four 2x2 bitonic nodes attached in cross bar format. Flits from east and west ports are given as input a and b respectively to u0 node whose output is a1 and b1,Flits from north and south ports are given as input c and d respectively to u1 node whose output is c1 and d1.The priority bits are compared and sorted and given to u2 and u3 whose outputs are (w,x),(y,z).Thus received flits from four ports are ranked and sorted. Its RTL implementation is shown in Figure 14. and its simulation is illustrated in Figure 15. From this Figure we can see that “0001011”, “00001000”, “00001101” and “00000111” flits to east, west, north and south port respectively are ranked in descending order of output ports w (0001101), x(00001011), y(00001000) and z(00000111).

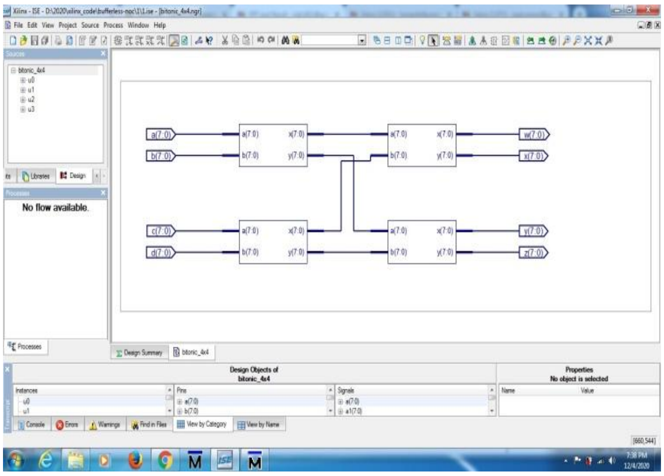


Fig.14.RTL implementation of Sorting network

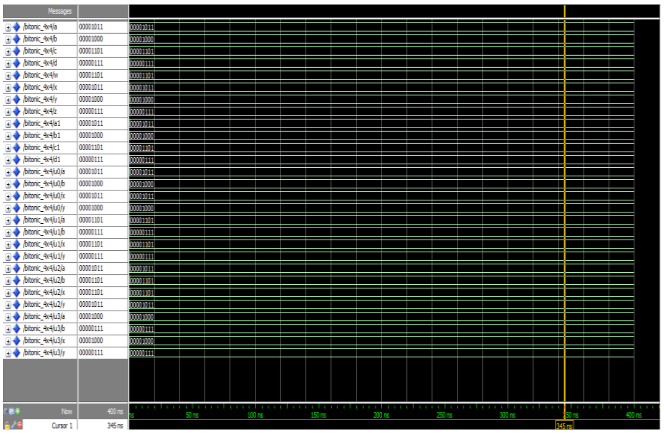


Fig.15.Simulation of Sorting network in our design

Each router has 4 input and 4 output ports (East, West, North, South) and one bypass port in sub network for deflection containment. Assuming 8 bit data last three bits indicate next hop of the flit. Consider flit f_0 is to be moved to south f0(10000**100**), f_1 is to be moved to north f_1(01000**011**), f_2 is to be moved to east f2(00100**001**), f_3 is to be moved to north f_3(00010**011**). Here , according to DeC design it first checks for non-contention i.e, f_0 can be moved to south port, f_2 can be moved to east port. Then we see that f_1 and f_3 are contending for moving to north port. Here f1 is moved to bypass port and f_4 is moved to north port.RTL implementation of parallel port allocator is shown in Figure 16 and its simulation is shown in Figure 17.

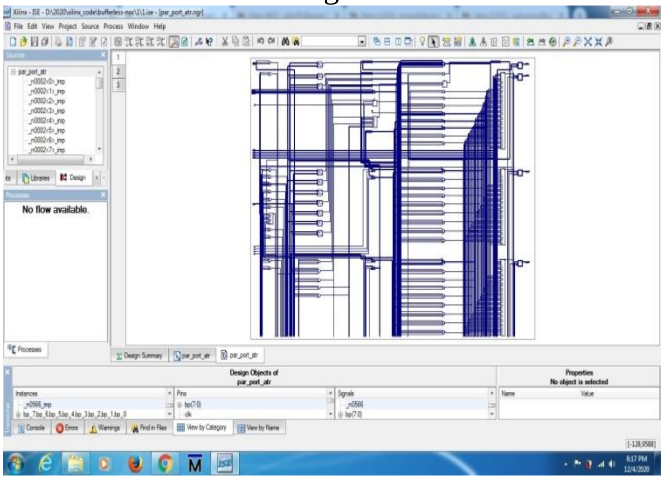


Fig.16.RTL schematic of Parallel port allocator in our design

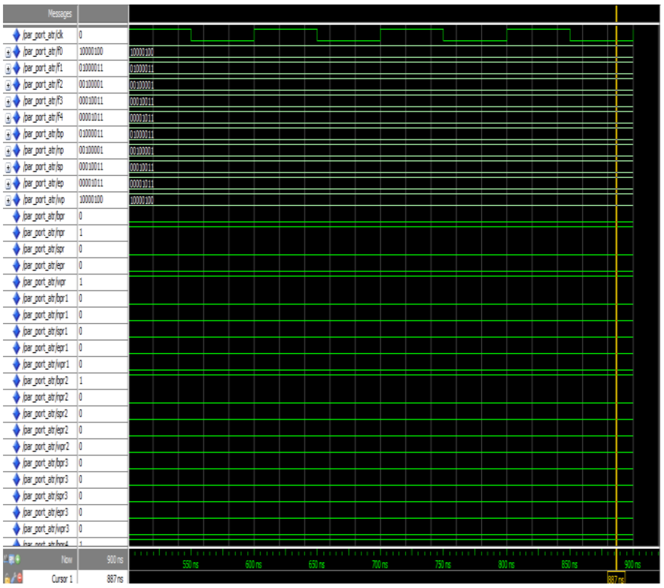
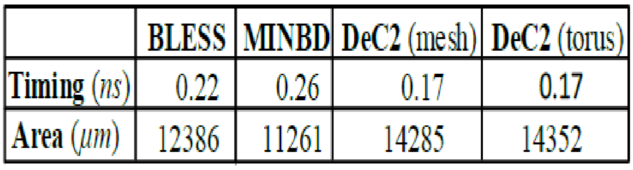


Fig.17: Simulation of parallel port allocator in our design

Table 2 presents comparison of various bufferless NoCs RTL implementation of router in 15nm process.



	BLESS	MINBD	DeC2 (mesh)	DeC2 (torus)
Timing (ns)	0.22	0.26	0.17	0.17
Area (um)	12386	11261	14285	14352

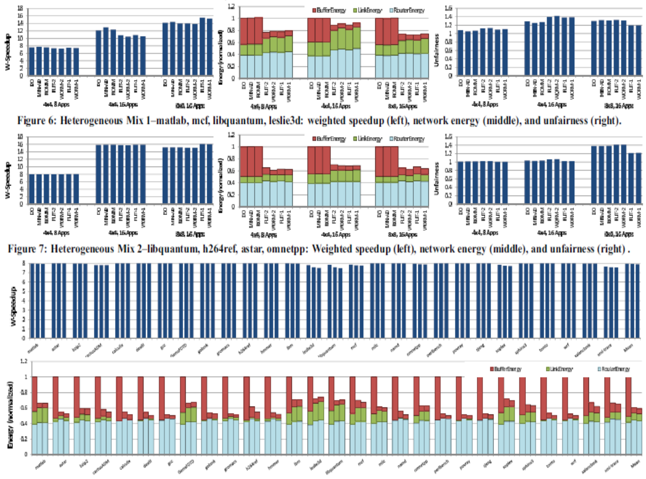


Figure 6: Heterogeneous Mix 1–matlab, mcf, libquantum, leslie3d: weighted speedup (left), network energy (middle), and unfairness (right).

Figure 7: Heterogeneous Mix 2–libquantum, h264ref, astar, omnetpp: Weighted speedup (left), network energy (middle), and unfairness (right).

Fig.18.Network energy performance of various design with our design of parallel port allocator

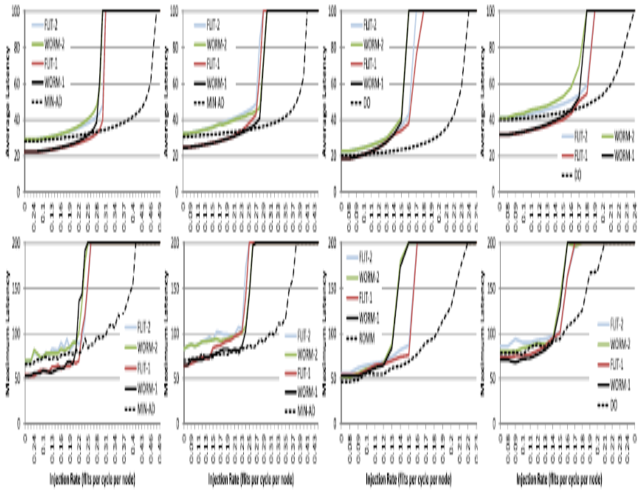


Fig.19.Latency of various design with our design of parallel port allocator

Conclusion

We make the case that bufferless NoCs can be effective way to simplify our design reducing latency and improving performance. With the use of simple bypass ring in every node divided into subnetworks, deflection can be reduced and unnecessary hops of flits is removed improving its performance. Moreover parallel port allocation along with bitonic sorting can reduce delay and improve network bandwidth by allocating ports in parallel instead of sequential allocation. With reduced time and improved bandwidth power consumption is reduced and with removal of buffers we can achieve scalability i.e reducing area.We have evaluated our design in Modelsim and also found that our proposed design is far more better than previous works.We have run simulation with 4x4 mesh topology which in future will be tested for more number of cores and various topologies.

References

Journals

[1] C. Nachiar, R. Poovendran and D. Saraswathi, "Architectural exploration of heterogeneous NoC with fault tolerant capacity," *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, 2020, pp. 1-5Geer, J., Hanraads, J. A. J., & Lupton, R. A. (2000). The art of writing a scientific article. *Journal of Science Communication, 163*, 51–59.

[2] H. P. E. Berman, L. Gravano, G. D. Pifarre, and J. L. C. Sanz. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks. *IEEE TPDS*, 12(5), 1994.

[3] R. Ausavarungnirun et al., “A Case for Hierarchical Rings with Deflection Routing: An Energy-Efficient On-Chip Communication Substrate,” Journal of Parallel Computing (PARCO), pp. 29-45,2016.

[4] S. Bhansali, W.-K. Chen, S. D. Jong, A. Edwards, R. Murray, M. Drinic, D. Mihocka, and J. Chau. Framework for instruction-level tracing and analysis of programs.

[5] D. Boggs et al. The microarchitecture of the Intel Pentium 4 processor on 90nm technology. *Intel Technology Journal*, 8(1), Feb. 2004.

[6] W.J. Dally, “Virtual-Channel Flow Control,” Proc.of 17thInt’l Symposium on Computer Architecture (ISCA), pp. 60-68, 1990.

[7] T. Moscibroda and O. Mutlu, “A Case for Bufferless Routing in On-Chip Networks,” Proc.of 36thInt’lSymposium on Computer Architecture (ISCA), pp. 196-207, 2009.

Book

[8] H. S. S. Borkar. Thousand core chips: A technology perspective. In DAC,2007.