

# DESIGN AND IMPLEMENTATION OF SHORTEST PATH AND FAULT TOLERANT ALGORITHM IN NOC

<sup>1</sup>SHRIDHAR S. BILAGI, <sup>2</sup>RAYMOND IRUDAYARAJ I., <sup>3</sup>ULAGANATHAN J., <sup>4</sup>ABDUL LATEEF HAROON P.S

<sup>1,2,3,4</sup>Assistant Professors, Dept. Of ECE, BITM/RMEC-Ballari-583104  
E-mail: <sup>1</sup>shridharbilagi875@gmail.com, <sup>2</sup>raymond.jhts@gmail.com, <sup>3</sup>ulgan.81@gmail.com, <sup>4</sup>abdulharoon27@gmail.com

**Abstract** - This paper describes an innovation scaling, dependability as became the main problem for network on chip (NOC). Numerous deficiencies tolerant steering calculations for NOC are produced to eliminate segments & gives solid transmission. In any case, proposed steering calculations don't give careful consideration to locate the most limited ways, which expands dormancy and force utilization. In this one, a shortcoming tolerant directing calculation utilizing new segment states dissemination technique in light of "Farthest Reachable Router (FRR)" is implemented. In this, calculation can decrease inactivity by identifying the limited paths among the source and destination switches. Tested simulation output confirms for the FRR directing calculation endure 50% shortcoming designs inside 3x3 and decrease inactivity by 16-44% contrasted & FON.

**Keywords** - Network on chip (NOC), Node, Router, FRR

## I. INTRODUCTION

The force of Moore's guidelines, more centers incorporated in a solitary chip, for On-Chip system it gives challenging task. On-Chip (NOC) gives more data transfer capacity & has high adaptability than conventional On-chip correspondence frameworks. Numerous center frameworks in view of NOC have generally looked into and built up. This work depends on system NOC along with cross section topology. Semiconductor technology as achieved profound submicron era, dependability of NOC is confronting more stretch. Generally on-chip components to be defective. Defects in NOC can be sorted in two forms: transient issue & permanent issue. Generally it is caused by force network vacillations, molecule hits & also crosstalk in network. Normally most of time, transient deficiency is recuperated by retransmission system or correcting error codes. In the conventional method, lasting deficiency is occurred by hardware problems. Deficiencies brought on by force restriction, warm administration can likewise be dealt with as lasting issue. To beat lasting flaw, steering ways of parcels ought to be rearranged to round the issue parts. In this manner flaw tolerant directing calculations have been broadly utilized.

### The Technique

A novel steering calculation went for decreasing idleness by accomplishing the most limited ways is proposed. This steering calculation utilizes new segment states dissemination system taking into account "Farthest Reachable Router (FRR)" algorithm, Tests the steering calculation is

- It endures all flaws in design maybe with a couple of shortcoming switches and 79% issue designs inside 3x3.
- And lessen the bundle inertness by 44% at most contrasted and FON

Deficiency tolerant steering algorithm scan be ordered into two principle sorts: deterministic and versatile directing calculation.

A deterministic steering calculation utilizes an altered way for every pair of hubs bringing about expanded parcel inertness particularly in congested systems.

In versatile steering calculations, bundles are not limited to a solitary way when crossed from a source switch to a destination switch. In this way, they can diminish the likelihood of steering bundles through congested regions and along these lines enhance the execution.

## II. METHODOLOGY

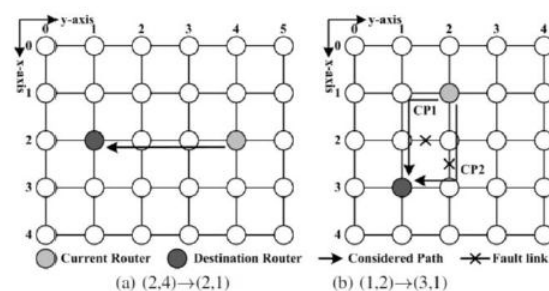


Fig 1: Generic path consideration in NOC

### 1) Node Analysis

In Node analysis it will compare source bits and destination bits. Source and destination bits are 14 bits. In source bits First 6 bits (MSB) are source address and rest 8 bits are data that to transfer to the destination. In destination same First 6 bits (MSB) are destination address and rest 8 bits are data in the destination.

Here source MSB bits compare with Destination MSB to find a shortest path in the network. For

example (0, 1) will be source address and (2, 7) is destination address. Here (0 < 2) therefore the flow in the network moves SOUTH. (1, 2)  $1 < 2$  the flow moves south. Here (2, 2) both  $s_x$  and  $d_x$  are equal. It will stop  $s-x$  &  $d_y$  comparison.

Now it compares source address and destination address, where destination address is 7 and source address is 1 ( $1 < 7$ ). Therefore the data packet moves towards EAST port. So this happens till it actually reaches the destination by check equality of the address.

Coordinate	Source	Condition	Destination	Towards port
X - Address	Source	Less than	Destination	South
	Source	Greater than	Destination	North
Y - Address	Source	Less than	Destination	East
	Source	Greater than	Destination	West

Fig 2: Packet passage algorithm table

**2) Network Analysis**

After Node analysis group of nodes will form a network and perform shortest path in network to reduce the time during execution. And also provide a fault tolerant to the system.

**III. RESULTS AND DISCUSSIONS**

**1) Individual node operation**

The architecture uses a 64 node (units when connected into a system) structure. Each node is connected to a subunit in a system. The node is capable of processing an 8-bit data packet at a time.

The node has 4 input ports and 4 output ports alike naming it to be "N, S, E & W port" for node to node connection. And to consider, it also employs an additional input-output port to which connects to the subunit to the network through the node and it is called as G-general input and PROCOUT-process output.

The system and node together uses a CLK-clock, where the operation of the system happens on both the clock cycle (for both positive and negative edge).

**Case 1: Data towards south port (Dest X > Src X)**

For considering the operation of a single node, let us take up the data to be transmitted to be "01010111" from the source node "00" ( $addr_x=000$  and  $addr_y=000$ ) to destination node "45" ( $addr_x=100$  and  $addr_y=101$ ). Shown in Fig 4.2



Fig 3: Data towards South Port with no Error

In this case, we assume errors on adjacent nodes to be zero. As per the algorithm the destination  $addr_x$  is compare with the source  $addr_x$  in the first place. Here in this case, the destination  $addr_x$  is larger than that of source. Hence the data packet is passed on to the south adjacent node. As shown in the figure 3.

**Case 2: Data towards north port (Dest X < Src X)**

For considering the operation of a single node for a different case, let us take up the data to be transmitted to be "00010111" from the source node "65" ( $addr_x=110$  and  $addr_y=101$ ) to destination node "23" ( $addr_x=010$  and  $addr_y=011$ ).

Every kind of operation happening depends on the comparison and move technique. Each node is responsible for this operation. The data reaches the destination by the node to node transmission till the packet actually reaches the destination.

Unlike the previous case, the consideration here is towards the north port. The destination and source address is chosen such that the case is satisfied on the requirement to be the north port data passage.

In this case, we again assume errors on adjacent nodes to be zero even for this case. As per the algorithm the destination  $addr_x$  is compare with the source  $addr_x$  in the first place. Here in this case, the destination  $addr_x$  is smaller than that of source. Hence the data packet is passed on to the north adjacent node. As shown in the figure 4.

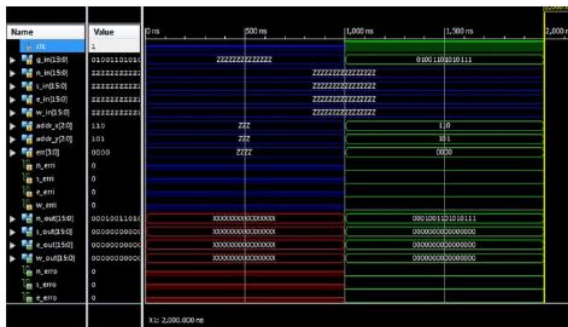


Fig 4: Data towards North Port

Case 3: Data should be towards east port (Dest X = Src X) with error it tends to be on north port

For considering the operation of a single node for a different case, let us take up the data to be transmitted to be “10101010” from the source node “04” (addr\_x=000 and addr\_y=100) to destination node “07” (addr\_x=000 and addr\_y=111).



Fig 5: Data towards East Port with Error

This time, for this case, we assume an error on adjacent east node for this case. As per the algorithm the destination addr\_x is compare with the source addr\_x in the first place. Since addr\_x for source and destination are equal, now on the second place the addr\_y of source and destination is compared and data movement happens.

Here in this case, the destination addr\_y is larger than that of source. Hence the data packet must be passed on to the east adjacent node. Since there is an introduced error on the east adjacent port, the data is not pushed towards east, but the data is pushed to the north port, where the force-y bit (15th bit of the data packet) is made high. As shown in the figure 5.

**2) Network operation**

In essence, the proposed network is made up of 64 nodes connected adjacently using the 4 ports of the nodes. Each node gets an address with clause addr\_x and addr\_y defining on which point of the network that node falls in or fits in.

The inputs concerned with the network are datain, clk, error (if needed) and one output for process output (PROCOUT). This list of inputs and output is for all the 64 nodes reflected with the node number.

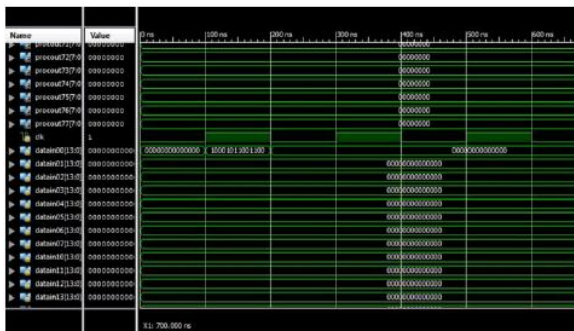


Fig 6: the list of inputs and output is for all the 64 nodes

As seen in the figure above, the network is triggered on both the edge of the cycle. And at a single instance of time, the input for 4 nodes is given at its datain (linked to general input of the node).

The 14-bits of data input on all levels are divided into 3 parts as destination addr\_x, destination addr\_y and the actual data of 8-bits. The data is from bit-0 till bit-7. And the addr\_y goes from bit-8 till bit-10. And the remaining bits are dedicated to addr\_x.

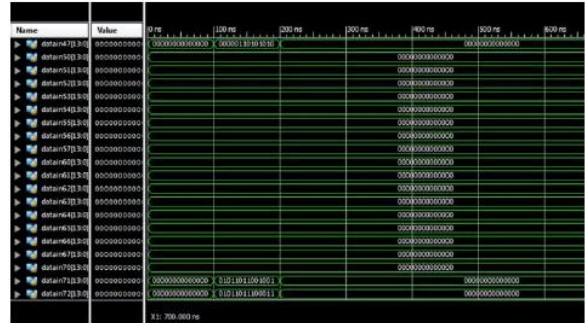


Fig 7: the list of inputs and output is for all the 64 nodes

The datain to procout for all 4 instances are shown below:

1st instance is from datain00 (source XY=>00) to procout42 (destination XY=>42). And the data is “11001100”. Hence the packet with address and data is “10001011001100”.

2nd instance is from datain47 (source XY=>47) to procout01 (destination XY=>01). And the data is “10101010”. Hence the packet with address and data is “00000110101010”.

3rd instance is from datain71 (source XY=>71) to procout26 (destination XY=>26). And the data is “11001001”. Hence the packet with address and data is “01011011001001”.

4th instance is from datain72 (source XY=>72) to procout26 (destination XY=>26). And the data is “11100011”. Hence the packet with address and data is “01011011100011”.

The respective outputs for the given inputs are shown in the below figures 8.

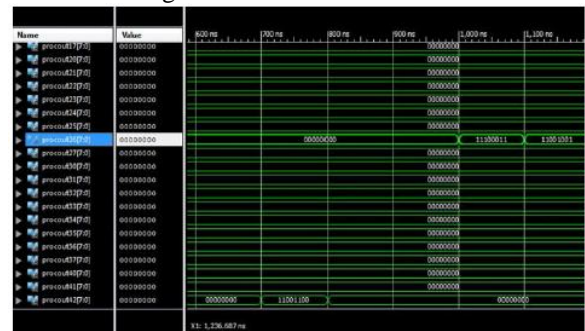


Fig 8: the list of inputs and output is for all the 64 nodes

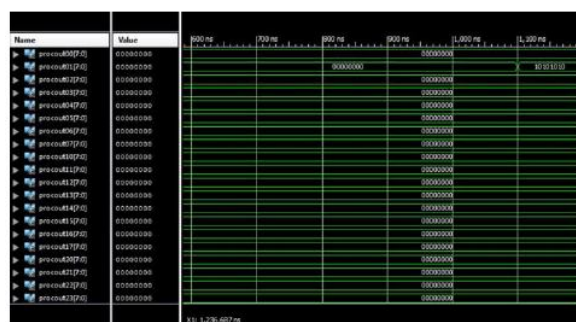


Fig 9: the list of inputs and output is for all the 64 nodes

All the four output are on different instant of time defining the clock cycle. But the data reaches the destination without making any duplication which in turn eliminates the overhead of the traffic in the network.

## CONCLUSIONS

Most remote Reachable Router flow tolerant steering algorithm for network is displayed in project. "Farthest Reachable Router (FRR)" utilizes to proliferate & capacity part states, here data deficiency and real time segments in mean time. Many segment parts called by switches are the key purpose behind FRR steering calculation to diminish normal way length. Test simulation confirm it that FRR can endure all deficiency designs inside two  $\times$  two and lessen the inactivity by 46% at most contrasting and FON directing calculation with PARSEC benchmark. The architecture uses a 64 node (units when connected into a system) structure. Each node is connected to a subunit in a system. The node is capable of processing an 8-bit data packet at a time.

The node has 4 input ports and 4 output ports alike naming it to be "N, S, E& W port" for node to node connection. And to consider, it also employs an additional input-output port to which connects to the subunit to the network through the node and it is called as G-general input and PROCOUT-process output. The system and node together uses a CLK-clock, where the operation of the system happens on both the clock cycle (for both positive and negative edge).

## FUTURE SCOPE

In future work, we will investigate wise directing system to handle more convoluted issue designs. Additionally, we will pay consideration on parity the expense of deficiency data dispersion and directing parcels (packets).

It can also be made as 8 ports instead of 4 ports. Where the port names can be as follows: NN – NE –

E – ES – S – SW – W – WN. This change can make the network faster indeed.

## ADVANTAGES AND APPLICATIONS

- Available on General IP cores. Easy to use.
- Logically No chances of data loss & data duplication which in turn avoids traffic.
- Provided shortest path algorithm even on transient and permanent errors.
- A total of 64 sub units can be connected to the network making it a huge advantage.
- Multiple data inputs instances can be made at one instance.
- Used in Video Processor Chip consisting of various Units.
- Employed in Complex algorithm computational units.
- Used in most advanced Network on Chip for communication between units
- On application requiring Low latency & Fast communication on chips.

## REFERENCES

- [1] Junshi Wang. "A Fault-tolerant Routing Algorithm for NoC Using Farthest Reachable Routers", 2013 IEEE 11th International Conference on Dependable, Autonomic & Secure Computing
- [2] Jarbas Silveira. "A Fault Prediction Module for a Fault Tolerant NoC Operation"
- [3] Letian Huang. "Low Cost Serious Failure Tolerance Solution in Unreliable NoC"
- [4] Thomas Hollstein. "Mixed-Criticality NoC Partitioning based on the NoC Dependability Technique"
- [5] T.G. Mattson, "The 48-core SCC processor: The programmer's view", Proc. High Performance Computing, Networking, SC, pp. 1-11
- [6] S. Bell, "TILE64 processor: A 64-core SoC with mesh interconnect", Proc. ISSCC, pp. 88-598
- [7] S. R. Vangal, "An 80-tile sub-100-W teraFLOPS processor in 65-nm CMOS", IEEE J. Solid-State Circuits, vol. 43, no. 1, pp. 29-41, 2008
- [8] N. E. Jerger, L.-S. Peh and M. H. Lipasti, "Circuit-switched coherence", Proc. ACM/IEEE Int. NoCs, pp. 193-202
- [9] Abousamra, A. K. Jones and R. Melhem, "Codesign of NoC and cache organization for reducing access latency in chip multiprocessor", IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 6, pp. 1038-1046, 2012
- [10] M. Modarressi, A. Tavakkol and H. Sarbazi-Azad, "Virtual point-to point connections for NoCs", IEEE Trans. Comput. - Aided Design Integr. Circuits Syst., vol. 29, no. 6, pp. 855-868, 2010
- [11] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks", Proc. DAC, pp. 684-689 [CrossRef]
- [12] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm", IEEE Trans. Comput., vol. 35, no. 1, pp. 70-78, 2002
- [13] U. Y. Ogras and R. Marculescu, "Energy-and performance-driven customized architecture synthesis using a decomposition approach", Proc. DATE, pp. 352-357
- [14] Pinto, "Efficient synthesis of networks on chip", Proc. ICCD, pp. 146-150