

A Novel Design of Multi-Port Cartesian Router

R.Anitha^{#1}, Dr.P.Renuga^{#2}

^{#1}Lecturer, Department of Electronics and Communication Engineering, PSNACET, Dindigul
¹anitha_rvs@yahoo.co.in

^{#2}Assistant Professor, Department of Electrical and Electronics Engineering, TCE, Madurai
²preee@tce.edu

Abstract: - Network-on-chip (NoC) architectures are emerging for the highly scalable, reliable, and modular on-chip communication infrastructure platform. The NoC architecture uses layered protocols and packet-switched networks which consist of on-chip routers, links, and network interfaces on a predefined topology. In this Paper we design network-on-chip which is based on the Cartesian network environment. This paper proposes the new Cartesian topology which is used to reduce network routing time, and it is a suitable alternate to network design and implementation.

The Cartesian Network-On-Chip can be modeled using Verilog HDL and simulated using Modelsim software.

Keywords: - Cartesian Networks, NoC, Routing algorithm, Multi-port Cartesian Router

I. Introduction

Cartesian routing is a fast packet routing mechanism intended for geographic addresses and can effectively accelerate the packet routing process within a local or metropolitan environment. The wide area Cartesian routing described in this paper is an extension of the Cartesian routing algorithms designed to make the exchange of internet work packets between geographical regions possible.. The proposed Internet is viewed as a hierarchy of networks consisting of routers. At the highest level of this hierarchy, major routers exchange packets between large geopolitical areas such as countries, states, or provinces. At the lowest level of the structure, packets are routed between local routers in small geographical regions ranging from an office to a small town The Cartesian Routing algorithm overcomes these problems by creating a hierarchical network consisting of two or more layers. Each network at a given layer encompasses one or more networks Each network, regardless of its layer, employs the Cartesian Routing algorithm for packet routing. Two extensions to the original Cartesian Routing algorithm are required: each network (except for the highest) requires an internet work router that can direct

packets destined for other networks “up” to the encompassing network. The address structure reflects the network structure, with specific fields in the address associated with each layer.

II. Cartesian Networks

A Cartesian network consists of a set of *collectors* and one or more *arterials*

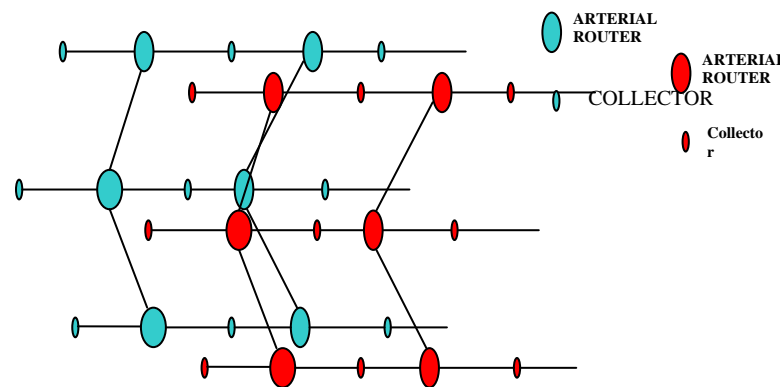


Fig.1. Cartesian Router

Each collector is a chain of *collector routers* running east-west and sharing common latitude. Collector routers have two side ports (east and west) to exchange packets' horizontally'. Each collector router also has a *bottom* Port which allows it to connect to a set of local hosts. Arterials exchange packets between collectors. Each arterial router, except the most northerly and the most southerly has, at least, four ports (north, south, east and west). Arterials need not share a common longitude. In a Cartesian network, the imposed topological structure relieves each router from maintaining routing tables. Each router is bound to a unique pair of addresses, the state information is minimal, and each router maintains the accessibility of arterials to its west and east.

2.1 Cartesian Network Initialization

In Cartesian routing, each arterial issue *Arterial This Way* (ATW) control packets during its initialization process. An ATW tells the receiving collector router if an arterial is accessible through the incoming port. An ATW also specifies what kind of connection is accessible via the incoming port: north, south, north and south or neither. Upon receiving an ATW, each collector router updates its *Arterial Direction Indicator* (ADI) and forwards the ATW to the opposite port. ATWs are also used to establish *virtual Arterials*, constructed in situations where it is physically impossible for an arterial to span two collectors. The ADI points in the direction of the arterial router (i.e., east or west) and indicates whether the arterial router has a connection to the north, the south, or both. Figure 1 illustrates a Cartesian network.

Packets can arrive on either a west or east port of a collector router. Packets intended for different latitude are forwarded out the opposite port from which they are received. The ADI determines the packet's initial direction on the collector router when a packet arrives on the bottom port collector router. In deciding a packet's initial direction, the router first compares the packet's destination address with its own address. The packet will be forwarded in the direction of the destination if the destination latitude is the same as the collectors. The packet is forwarded in the direction of the ADI if the destination is on different latitude.

2.2 4 Port Cartesian Routing Algorithm

2.1.2 Collector Router

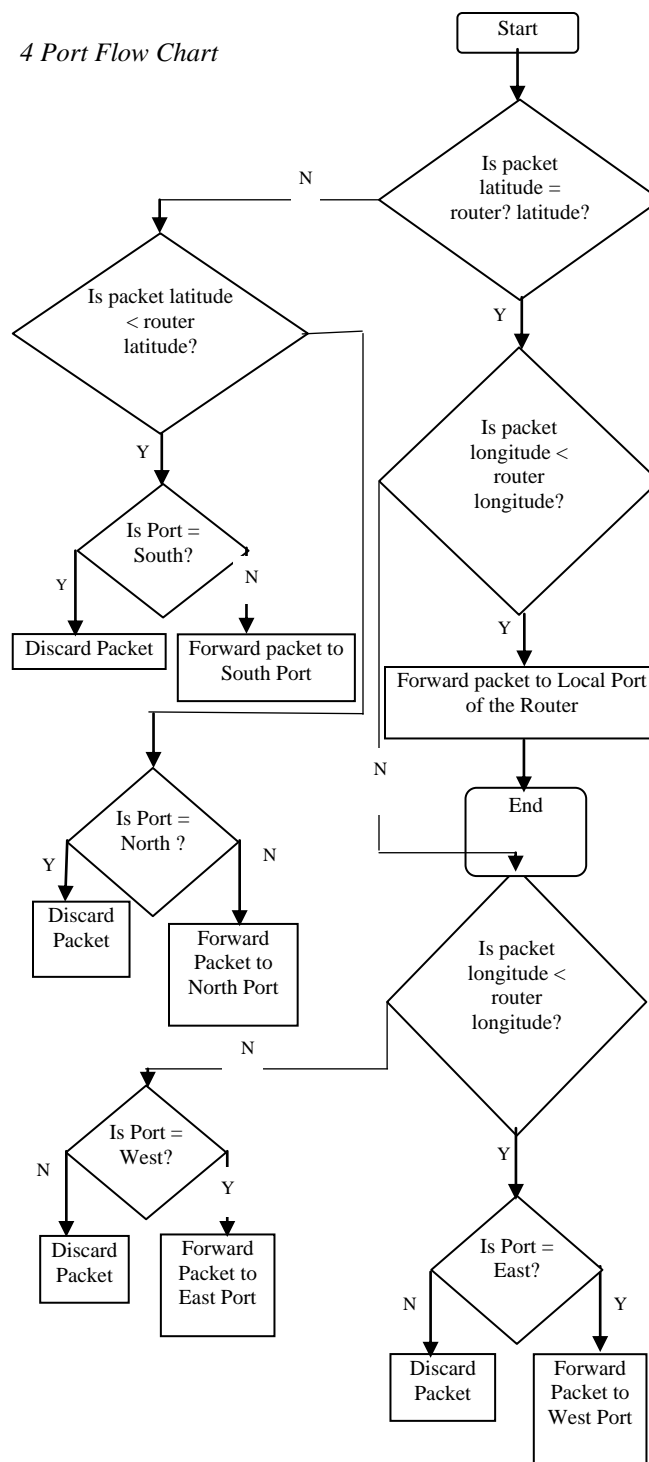
The Collector Router is responsible for routing a unicast packet on 'horizontal', collector networks. This router has tree ports: West, East and Local. When a packet arrives at a port; its longitude address is compared with the router's longitude address. If the packet's longitude is greater than that of router, the packet is forwarded to the east port. If less than, the packet is sent towards west, and if equal the packet belongs to the local port. The packet is discarded when a packet is determined to have a non-existent Cartesian address; this is to prevent packet circulation.

2.1.1 Arterial Routers

The Arterial router is responsible for routing packets from one collector network to another. An arterial router has four ports: North, East, South and West. When a packet arrives at an arterial router, its latitude is compared with the router's latitude.

If they are not equal, it implies that the packet either belongs to an 'upper' or 'lower' collector network; if packet's latitude is greater, the packet is forwarded north otherwise it is forwarded south. If the packet to west or east. Arterial routers also discard packets with addresses of non-existent destinations.

4 Port Flow Chart



| ADDRESS | FUNCTIONS |
|--------------------------------------|-----------------------------|
| Destination latitude=node_latitude | Router keeps the Packet |
| Destination_longitude=node_longitude | Packet routed to North Port |
| Destination_latitude>node_latitude | Packet routed to South Port |
| Destination_longitude>node_longitude | Packet routed to East Port |
| Destination_latitude<node_latitude | Packet Discarded |

Process flow for 4 Port Cartesian Network

Table 1:4-port Routing algorithm

The Routing algorithm for the Cartesian Network is if the Destination Latitude is equal to the node latitude and also the Destination longitude is equal to the node longitude means the router hold the packet. If the destination Latitude is greater than load latitude and also destination longitudes greater than node longitude means the packet routed to north port. If the destination latitude lesser than load latitude and also destination longitudes greater than node longitude means the packet routed to south port .If the destination latitude is equal to the node latitude and also destination longitude is greater than node longitude means the packet routed to East port. If the destination

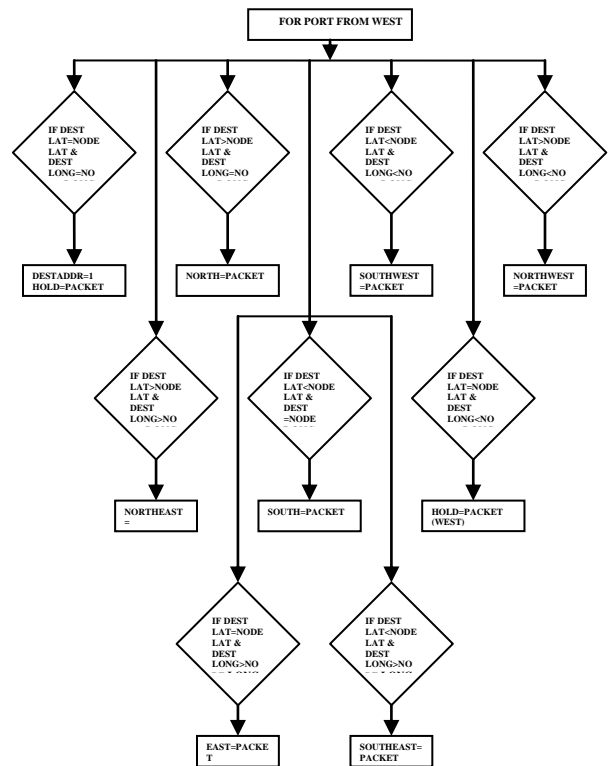
| ADDRESS | FUNCTIONS |
|--|----------------------------------|
| IF DEST LAT=NODE LAT & DEST LONG=NODE LONG | Router keeps the Packet |
| IF DEST LAT>NODE LAT & DEST LONG=NODE LONG | Packet routed to North Port |
| IF DEST LAT<NODE LAT & DEST LONG<NODE LONG | Packet routed to South West Port |
| IF DEST LAT>NODE LAT & DEST LONG<NODE LONG | Packet routed to North Port |
| IF DEST LAT>NODE LAT & DEST LONG>NODE LONG | Packet routed North East |
| IF DEST LAT<NODE LAT & DEST LONG=NODE LONG | Packet routed to South port |
| IF DEST LAT=NODE LAT & DEST LONG<NODE LONG | Router hold the Packet (WEST) |
| IF DEST LAT=NODE LAT & DEST LONG>NODE LONG | Packet routed to East port |
| IF DEST LAT<NODE LAT & DEST LONG>NODE LONG | Packet routed to South East port |

longitude is lesser than node longitude means the packet will discarded

Table2: 8 Port Routing algorithms

If destination latitude is equal to node latitude and destination longitude is equal to node longitude then router keeps the packet. If destination latitude is greater than to node latitude and destination longitude is equal to node longitude then packets routed to north port.

8-Port Flowchart:



2.3 Wide Areas Cartesian Networks

A Cartesian network provides a straightforward topological structure that relieves collector routers from the need to maintain routing tables. However, it would be unrealistic to implement a single worldwide Cartesian network. Such a widespread Cartesian network, for example, requires every packet destined for a router with the same latitude identifier as the source router's latitude identifier to visit all the collector routers. It is also necessary for such a network to have one collector for every possible latitude. These limitations suggest that implementing a single worldwide Cartesian network would be impractical. An alternative to a worldwide Cartesian network is to create a set of smaller Cartesian networks and implement a mechanism for interchanging packets between them. One

approach to interchanging packets between Cartesian networks is to forward packets towards their destinations.

When a packet reaches the boundary of a network it “falls off” the edge and is delivered to a special router to be forwarded towards the destination address. The process of routing a packet from one network to another using this approach becomes problematic when networks are *interleaved* or *overlapped*. Two networks are considered interleaved if there is at least one collector router on one of the networks where its longitude identifier lies between the longitude identifiers of two collectors from the other network and its latitude identifier lies between the latitude identifiers of two collectors from the other network. Figure illustrates two interleaved networks. Two networks are said to be overlapped if there is at least one collector router on one of the networks where its longitude identifier lies between the longitude identifiers of two collectors from the other network and all three of them share the same latitude identifier. Figure illustrates two overlapped networks.

An alternative method for delivering a packet to its destination is to find the destination network address and then to route the packet to the destination network by using Cartesian routing algorithms. This implies that each network must be identifiable using the packet’s destination address. If we assume that each network has a rectangular shape, recognizing the destination network is a matter of comparing the packet’s destination address with the network’s boundaries. However, there are a number of reasons to assume that it would be unrealistic to expect networks to have rectangular borders: geographical barriers and political jurisdictions, for example. Since Cartesian routing uses latitude and longitude pairs to identify the source and the destination addresses of packets, this information is not sufficient to determine to which network a collector/arterial belongs in the case of interleaved and overlapped networks. This, in turn, suggests that an additional set of information is required to identify to which network a collector or arterial is connected. To achieve this, the authors propose a hierarchical structure for Cartesian networks. In the next section the possibility of multiple-layer Cartesian networks as a solution for interchanging packets between arbitrary shaped interleaved and overlapped Cartesian networks are explained. In the remainder of this paper, the terms “wide area Cartesian networks” and “multiple-layer Cartesian networks” are used interchangeably.

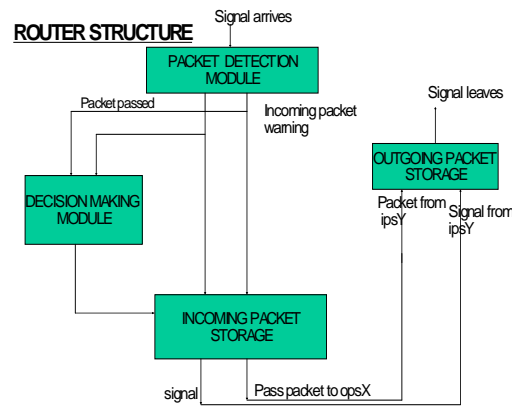


Fig 2: Router Structure

III. MULTIPLE-LAYER CARTESIAN NETWORKS

Multiple-layer Cartesian networks impose a new set of topological dependencies among a set of Cartesian networks, such that interchanging packets between networks is feasible without creating and maintaining routing tables. Generally, in multiple layer Cartesian networks, the idea of Cartesian networks is expanded in a larger scale using a hierarchical structure.

3.1 DMM

Multiple-layer Cartesian networks have a hierarchical structure. The highest layer of the hierarchy is a single Cartesian network. Each underlying layer consists of a set of mutually disjoint Cartesian networks (i.e., they are physically disjoint and share no collector or arterial router); however, networks in the same layer can be interleaved or overlapped.

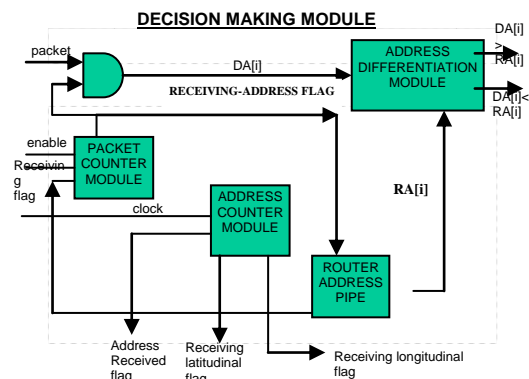


Fig 3: Decision Making Module

The PCM (Packet Counter Module) strips the destination address from the packet. It counts the incoming

packet bits and sets the RECEIVING-ADDRESS flag when the first bit of the address is read. The ACM (Address Counter Module) keeps the track of the number of address bits that have been read. It indicates which portion of the address is being received (latitude or longitude) and if the entire address is received, it sets the ADDRESS-RECEIVED flag. When the ADDRESS-RECEIVED flag is set, the IPS is told to keep the packet. The RAP stores the router address and pipes it out serially so that it can be compared with the incoming destination address.

The Router address is static value, and is kept in non-volatile memory for long term storage. The ADM simply compares the *i*-th bit of the destination address to the *i*-th bit of the router address as the destination address is read in to the DMM. If the DA does not equal to RA, the ADM tells the incoming packet storage (IPS) module to destroy the stored packet.

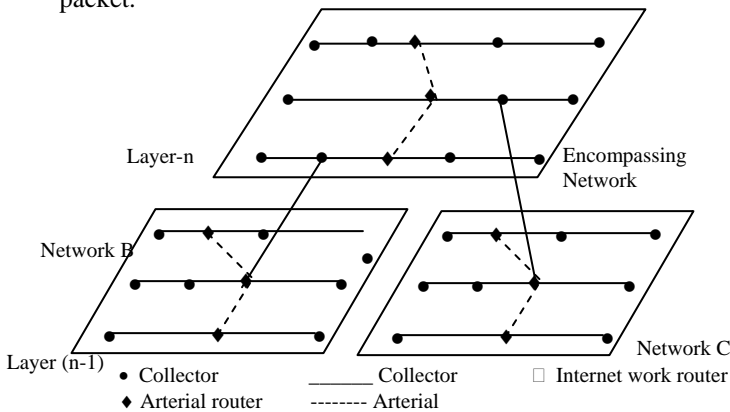


Figure 4: An encompassing network (A) and two encompassed networks (B and C)

3.2 Router Identification

In a Cartesian network, each router is bound to a Cartesian address. Whereas, in a multiple-layer Cartesian network, each router is bound to an *identifier*. The identifier of a router at layer-*m* of an *m*-layer Cartesian network is the same as its Cartesian address. An identifier of a router at lower layers is the identifier of its immediate encompassing router, followed by the Cartesian address of the router itself, meaning that an identifier is an ordered list of Cartesian addresses. For example, routers at layer-*(m-1)* maintain the Cartesian address of the router that represents the network to which they belong, followed by their own Cartesian address. In general, in an *m*-layer Cartesian network, each router at layer-*n* maintains a list of *(m-n+1)* ordered Cartesian addresses, where. For example, routers at the lowest layer of the hierarchy, layer-1, which are connected to local hosts through their bottom ports, are bound to *m* ordered Cartesian addresses: *m-1* correspond to the identifier of the encompassing router at layer-2, and one, the Cartesian

address of the router itself. Figure 5 illustrates the hierarchical addressing structure of an identifier for a router at layer-*n* of an *m*-layer Cartesian network. The hierarchical addressing structure overcomes the problems with both interleaved and overlapped networks since every router in the hierarchy has a unique address. It also enables a router in a network at layer to determine if a packet is local to the network or not. A packet is said to be *local* to a network if the network encompasses the destination address of the packet. A router can determine this by comparing the most significant *m-n* Cartesian addresses of the packet's destination address with the first *m-n* Cartesian addresses of its own identifier.

PACKET FORMAT

- SOP → Start to Packet (6 bits---6'b111111)
- EOP → End of Packet (6 bits---6'b111111)
- M → Cast (Unicast - 4'b0000, Multicast -b'0001)
- DLAT → Destination Latitude
- DLOK → Destination Longitude

| | | | | | | | |
|-----|---|------|------|------|-------|------|-----|
| SOP | M | DLAT | DLOK | SLAT | SLONG | DATA | EOP |
|-----|---|------|------|------|-------|------|-----|

3.3 Packet Routing

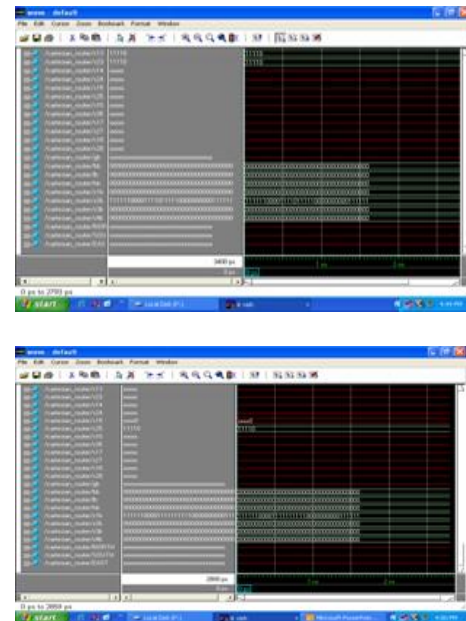
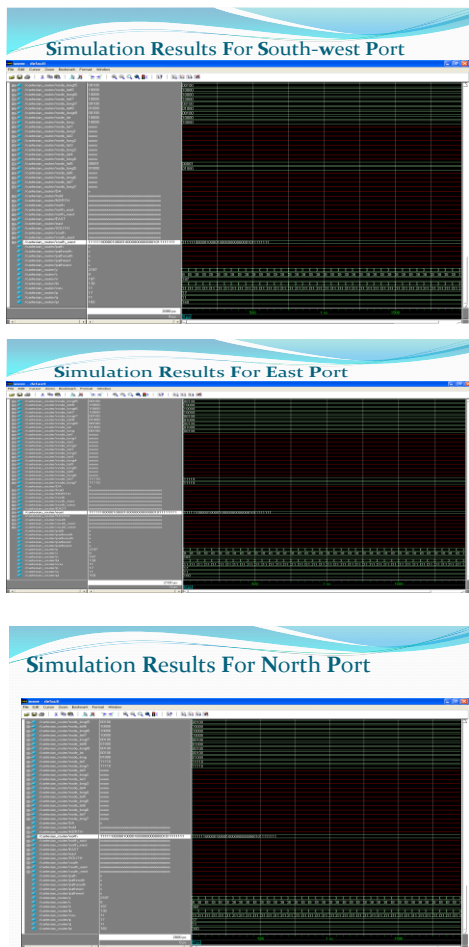
In an *m*-layer Cartesian network, each collector router at layer-*n* is bound to an identifier which is a list of *m-n+1* Cartesian addresses. A packet can enter a network at layer-*n* through the bottom port of a collector router or the top port of the network's IR. Packets received on the bottom port of a collector router are either local or non-local to the network, as described above. When a packet is found to be local, that is, the network encompasses the destination address, the router "tags" the packet as a local packet by setting a single bit of the packet's address called the *local bit*. When a packet is local to a network at layer-*n*, the *(m-n+1)* th Cartesian address of the packet's destination address is used to route the packet in the network using Cartesian routing algorithm. For example, at layer-*m*, the first Cartesian address is used for Cartesian routing, while at layer-1 the *m*th address is used. When a packet is received by a router on its bottom port, the address is Cartesian address of the encompassing router at layer-*(m-1)* inspected. Non-local packets must be sent towards the network's IR in order to be delivered to the encompassing network. The router clears the local bit and forwards the packet towards the IR. Forwarding a packet to the IR requires that each collector router and arterial maintains an *Internet work Router Direction Indicator* or IRDI. The IRDI determines if the internet work router is accessible through the west port, east port or neither, in the case of collector routers; or whether it is accessible through the north port, south port or neither, in the case of arterials. If the packet is determined to be non-local, it is forwarded in the direction specified by the IRDI.

When the IRDI indicates that IR is not accessible, the packet is dropped and a message is returned to the source notifying that the destination address is not reachable.

A collector router that receives a packet on its west or east port checks the local bit; if set the Cartesian routing algorithm is employed to route the packet, otherwise the packet is forwarded to the opposite port. When a packet enters a network through the top port of the network's IR, the packet is guaranteed to be local, since this has already been verified by the encompassing network. Upon receiving the packet, the IR sets the local bit and then applies the Cartesian routing algorithm on the $(m-n+1)$ th Cartesian address of the packet's destination address.

IV.SIMULATION RESULTS

4.1. Router Design simulation Result



V.CONCLUSION

In Previous work, they designed conventional router which uses a routing table to determine whether to keep, forward or discard the packets. As networks grow in size, the memory requirements of the routing tables increases proportionally. The average search time increases as the routing table increases. ASIC based router design.

In our Proposed work, the New Cartesian network is designed, which is Independent of routing table. It is the High speed network transmission when compared with existing work.

VI.REFERENCES

- [1].Low-Power Network-on-Chip for High-Performance SoC Design Kingman Lee, *Student Member, IEEE*, Se-Joong Lee, *Member, IEEE*, and Hoi-Jun Yoo, *Senior Member, IEEE* IEEE transactions On Very Large Scale Integration (VLSI) Systems," VOL. 14, NO. 2, February 2006.
- [2].W. Dally et al., "Route packets, not wires: On-chip interconnection networks," in Proc. Des. Autom. Conf., Jun. 2001, pp. 684–689.
- [3].L. Benini et al., "Networks on chips: A new SoC paradigm," IEEE Computer, vol. 36, no. 1, pp. 70–78, Jan. 2002.
- [4].D. Bertozzi et al., "Xpipes: A network-on-chip architecture for gig scale system-on-chip," IEEE Circuits Syst. Mag., vol. 4, no. 2, pp. 18–31, 2004.
- [5].E. Rijpkema et al., "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," in Proc. Des., Autom. Test Europe Conf., Mar. 2003, pp. 350–355.
- [6].V. Nollet et al., "Operating-system controlled network on chip," in Proc. Des. Autom. Conf., Jun. 2004, pp. 256–259.
- [1] A. Ivanov and G.D. Michele, "The Network-on-Chip Paradigm in Practice and Research," IEEE Design and Test of Computers, vol. 22, no. 5, pp. 399-403, Sept.-Oct. 2005.

- [2] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Milberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A Network on Chip Architecture and Design Methodology," Proc. IEEE CS Ann. Symp. VLSI, p. 117, 2002.
- [3] W.J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," Proc. 38th Design Automation Conf. (DAC '01), pp. 684-689, 2001.
- [4] F. Karim, A. Nguyen, and S. Dey, "An Interconnect Architecture for Networking Systems on Chips," IEEE Micro, vol. 22, no. 5, pp. 36-45, Sept.-Oct. 2002.
- [5] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a Switch for Network on Chip Applications," Proc. IEEE Int'l Symp. Circuits and Systems (ISCAS '03), vol. 5, pp. 217-220, May 2003.
- [6] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip," ACM Computing Surveys, vol. 38, no. 1, pp. 1-51, 2006.
- [7] P.P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-Offs for Networkon- Chip Interconnect Architectures," IEEE Trans. Computers, vol. 54, no. 8, pp. 1025-1040, Aug. 2005.
- [8] D. Linder and J. Harden, "An Adaptive and Fault-Tolerant Wormhole Routing Strategy for k-Ary n-Cubes," IEEE Trans. Computers, vol. 40, no. 1, pp. 2-12, Jan. 1991.