

Area and Power Budget Estimation of Hierarchical Network Topology in Comparison with 2D Mesh Topology for NOCs and its Design and Implementation Oriented Overview

Kulkarni Rashmi Manik, S Arulsevi, B Karthik

Abstract: Today's IC (Integrated Circuit) fabrication technology allows us, integration of billions of transistors on a single chip. The trend in design of SoCs (System-On-Chips) for mobile phones as well as high end servers is to integrate multiple processor cores with other peripheral cores for achieving higher performance. The other cores integrated along with processor cores are generally, graphics processing units, memory controllers, encryption/decryption cores, special functional units, accelerators and various types of interface controllers. As the number of processor cores (as a masters) increase beyond 16, the bus-based architecture poses bottleneck to overall performance. The network-on-chip based interconnects topology (using packet switching) promises higher performance for complex multi-core SoCs. The NoC (network-on-chip) interconnect design involves design of (a) network adaption logic for all cores, (b) on chip routers and (c) topology. The various network topologies have been proposed and analyzed for NoC. The problem of arriving at perfect Network topology and efficient router architecture for integrating 64 cores has been analyzed in this research work. As the low power SoC architectures are gaining importance, the reduction in power budget for NoC gains more attention of researchers. The power-wise exploration of topology with NoC power estimation tool can help in achieving low power targets. The proposed Modular Network Topology is compared with 2-D (Two Dimension) mesh topology. Standard Orion3 tool is used for estimating router power and area considering 45nm and 65nm technologies. Results are encouraging and better in performance than regular 2-D mesh topology. In this research article, we tried to give thought for gate level design and implementation of NOC. For that, we considered 64 processing elements connected with hierarchical networking topology. With implementation of GALS, feasibility study is carried out. Mainly we concentrated on cross bar router design which is a basic element of NOC.

Index Terms: Network on Chip (NoC), Network Topology, System-on-chip (SoC), Integrated Circuit (IC), ASM (Algorithmic State Machine), GALS (Globally Asynchronous Locally Synchronous), and NIC (Network Interface Component).

I. INTRODUCTION

The complexity of mobile application processor demands

Revised Manuscript Received on March 20, 2019.

Kulkarni Rashmi Manik, Research Scholar/ECE, Bharath Institute of Higher Education and Research, Chennai, India.

S Arulsevi, Associate Professor, Department of ECE, Bharath Institute of Higher Education and Research, Chennai, India.

B Karthik, Associate Professor, Department of ECE, Bharath Institute of Higher Education and Research, Chennai, India.

integration of many processor cores and other cores in to a single chip. The overall performance of an application processor depends not only on number of processing cores but the way in which all cores are interconnected. An efficient interconnects topology and router will impact on the performance of IC. Interconnects based on bus-based architecture do not provide desired performance. Network-on-chip (using packet switching protocol) has proven to be the only solution for better performance. Different network topologies have been proposed in earlier research work. The simple 2-D Mesh, spidergon [4] are common topology. Other topology like, square- octagon interconnects are in [1]. Binary tree, fat tree and mesh have been proposed [5]. Ran Manevich in [2] compared the hierarchical topology issues with 2-D mesh for suitability for many core architectures. The Quarc topology comparison with Spidergon is given in [3]. In [6] 2-D mesh topology with hierarchy is suggested. A central router for 64 ports is analyzed in [7]. In [8] the comparison of topologies is presented for 16 nodes. In our research paper, a new hierarchical network topology is proposed and same is compared with regular 2D mesh topology. The proposed topology and it's router are validated by using Orion3 analysis tool. The proposed topology is for interconnecting 64 cores. With suitable modification in the router, it can be adopted for interconnecting 100 cores or 144 cores. In the section 2, a basic router details are presented. Network topology for 64 cores is explained in section 3. Packet routing is explained in section 4. Comparison of 2D mesh and proposed topology (power and area wise, using Orion3 [9]tool) is presented in section 5. Implementation overview is given in section 6. The conclusion and extending topology for more number of cores is briefed in section 7.

II. INSIDE BASE ROUTER

Architecture of base router (BR) having 9 ports is shown in Fig. 1. The eight ports are connected to 8 processor/peripheral cores through network interface adapters and ninth port is for connecting to central-router (CR).



Area and Power Budget Estimation of Hierarchical Network Topology in Comparison with 2D Mesh Topology for NOCs and its Design and Implementation Oriented Overview

The all ports are bidirectional and packets can be routed from a port to any other port. The router supports 128-bit flit size for all ports and has buffers at input and output of all ports. Each port supports four virtual channels. Each buffer can accommodate four flits. Router has internal cross-bar switch and arbiter. Cross-bar switch is for routing packets from input port to

appropriate output port. Arbiter resolves contention of packets from different input ports to the same output port.

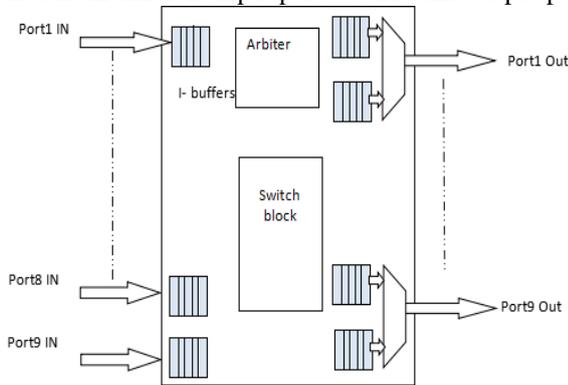


Fig.1. Base Router Architecture

III. NETWORK TOPOLOGY FOR 64 CORES

A Base Router interconnecting 8 processor cores is shown in Fig. 2. It forms a cluster. Each processor core has unique address for identification. The proposed network topology connecting 64 processor cores is shown in Fig. 3. The topology uses 8 numbers of base routers. The port 9 from each router is connected to central router. The central router (CR) has eight ports (Fig. 3). The router complexity in terms of area when implemented in 45 nm and 65 nm technology is estimated using Orion3 Tool.

Each Base Router has unique identification number ranging from 0 to 7 and each processor core has unique address ranging from 0 to 63. Base router 0 forms a cluster of 8 processor cores with address 0 to 7. Base router 1 forms a cluster of next 8 processor cores with address 8 to 15, and so on. The topology has eight clusters of eight processors having a base router in each cluster. The eight base routers are connected to a central router CR as shown in Fig. 3.

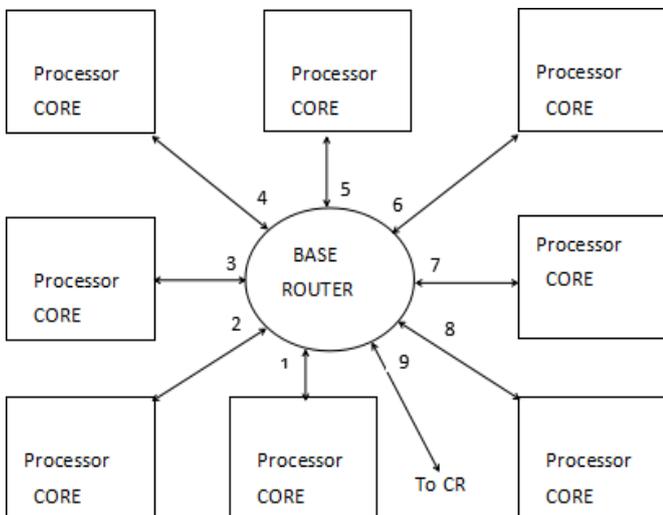


Fig.2. Base Router interconnected to 8 processor core

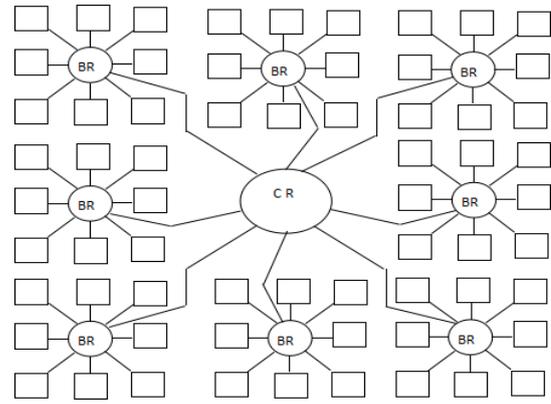


Fig.3. Network Topology connecting 64 cores

IV. PACKET ROUTING

Each processor core has unique address assigned from 0 to 63. The processor cores having addresses 0 to 7 are connected to base router BR0. The processor cores having addresses 8 to 15 are connected to base router BR1. Table. I show complete connectivity of base routers and processor cores.

The packets from source to destination are routed through Base Router and Central Router depending on the address of source and destination. Each base router has list of addresses of the processor cores connected to it. If destination address is within cluster, packet takes one hop to reach the destination. If destination address is not in base router's list, packet is forwarded to central router. The CR routes packet to appropriate base router depending on destination. The packet routed through CR takes three hops to reach the destination (BR - CR - BR).

TABLE I

BASE ROUTER AND CONNECTED CORES

Base Router ID	Addresses of the Processor Cores connected to Base Router
BR0	0, 1, 2, 3, 4, 5, 6 and 7
BR1	8, 9, 10, 11, 12, 13, 14 and 15
BR2	16, 17, 18, 19, 20, 21, 22 and 23
BR3	24, 25, 26, 27, 28, 29, 30 and 31
BR4	32, 33, 34, 35, 36, 37, 38 and 39
BR5	40, 41, 42, 43, 44, 45, 46 and 47
BR6	48, 49, 50, 51, 52, 53, 54 and 55
BR7	56, 57, 58, 59, 60, 61, 62 and 63

V. ROUTER AREA AND POWER COMPARISON WITH 2D MESH

The Orion3 tool is used for estimating area and power for Base and Central Router. It is then compared with 2-D Mesh router. In the 2D Mesh topology total 64 routers are required for connecting 64 processor cores and each router needs to have five ports.



The other router parameters such as input buffer size, output buffer size, number of virtual channels etc are kept same as per the 2-D mesh topology. Technology node of 65nm and 45 nm is used for generating the area and power values for operating frequencies from 100 Mhz to 1 GHz in steps of 100 Mhz. The number of input buffers and output buffers are set to 4 each and flit width is set to 128-bit. The area for 64 routers in 2-D mesh is estimated with Orion3. Similarly by configuring number of input ports and output ports in Orion3, the area for Base Router and Central Router is estimated and total area is calculated for proposed complete topology. Table II compares the proposed topology and a 2-D Mesh topology:-

The estimated power is compared at different frequencies for 45nm and 65nm technology as shown in Table. III and Table IV.

TABLE II
COMPARISON OF PROPOSED TOPOLOGY
AND 2-D MESH TOPOLOGY

Parameter	2-D Mesh Topology	Proposed Modular Topology
Routers required for connecting 64 cores	64 routers, each having 5 ports	8 base Routers each having 9 ports and 1 Central router having 8 ports
Maximum hops from source to destination	14	3
Router Area at 45 nm in sq.micron	For One router :-48659.0 For 64 routers :-3,114,176.0	For BR:- 131484.0 For CR:- 112964.0 Total area (for 8 BRs and 1 CR) :- 1,164,836.0
Router Area at 65 nm in sq. micron	One router:- 76385.0 For 64 routers:- 4,888,640.0	For BR:- 212075.0 For CR:- 181363.0 Total area (for 8 BRs and 1 CR) :-1,877,963.0

TABLE III
POWER COMPARISON AT 45NM

Topology	2-D Mesh		Proposed Topology		
Freq. MHz	One Router power in mW	Total Power in mW	BR power in mW	CR power in mW	Total power in mW
100	12.27	785.28	29.52	24.59	260.75
200	12.78	817.92	30.70	25.58	271.18
300	13.30	851.2	31.88	26.58	281.62
400	13.81	883.84	33.07	27.57	292.13
500	14.33	917.12	34.35	28.56	303.36
600	14.84	949.73	35.43	29.54	312.98
700	15.36	983.04	36.61	30.55	324.95
800	15.87	1015.68	37.80	31.55	333.95
900	16.39	1048.96	38.98	32.54	343.38
1000	16.91	1082.24	40.16	33.54	354.82

TABLE IV
POWER COMPARISON AT 65NM

Topology	2-D Mesh		Proposed Topology		
Freq. MHz	One Router power in mW	Total Power in mW	BR power in mW	CR power in mW	Total power in mW
100	25.63	1640.32	51.14	61.33	470.45
200	26.76	1712.64	53.26	63.83	489.91
300	27.88	1784.32	55.37	66.33	509.29
400	29.00	1856.00	57.49	68.83	528.75
500	30.12	1927.68	59.60	71.33	548.13
600	31.24	1999.36	61.72	73.82	567.58
700	32.36	2071.04	63.83	76.34	586.98
800	33.48	2142.72	65.95	78.83	606.43
900	34.60	2214.4	68.06	81.33	625.81
1000	35.73	2286.72	70.18	83.83	645.27

VI. DESIGN AND IMPLEMENTATION ORIENTED OVERVIEW

Heart of NOC or main component of NOC is router or cross bar switch. For implementation of NOC, design of router is important. Fig. 1 shows basic router, however for design purpose, we have taken reduced version of it. Following subsections describe abstract level design and pseudo-code for the same.

A. Basic Architecture

Basic architecture of complete system, integrating 64 processing elements:- In Fig. 4 it is explained that, how processing elements, summing processing elements and super summing processing elements are connected with the help of NOC.

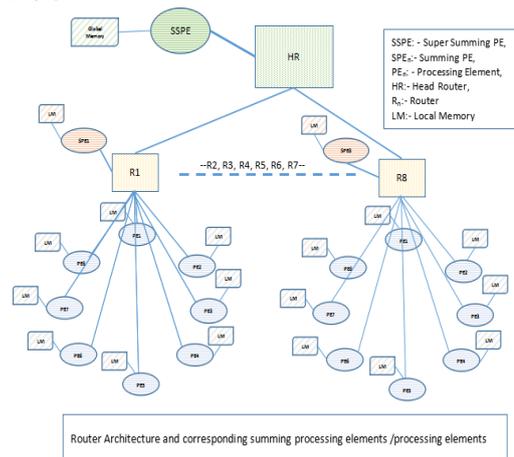


Fig.4. Architecture of complete system, integrating 64 PEs

B. Router Design - Block-wise

Crossbar switch/router design - Input, output buffers and arbiter (control unit):-

To start with simplest and basic design, Fig. 5 shows simple router with input, output buffers and control unit. Input buffer is of only one flit size and output buffer is of one complete data packet size. Input buffer will store only currently received flit, however output buffer will keep on receiving flits from input buffer for whole data packet. Simple router is designed for routing only one data packet at a time. Control unit will route data appropriately as specified in destination.

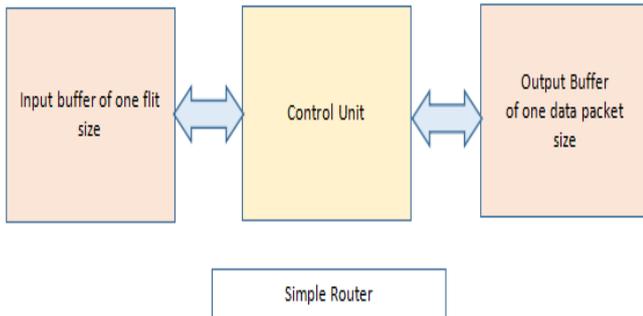


Fig.5. Crossbar switch/router design - I

C. Router Design - In Detail

Crossbar switch/router design – super multiplexer, super demultiplexer and arbiter:- Cross bar switch is at the heart of NOC. It does the job of packet routing. Crossbar switch is divided in to two basic parts, i.e. cross bar switch itself and arbiter. In the research article, we will call arbiter as a control unit. Looking at the basic function of cross bar switch, it is designed with super multiplexer and super demultiplexer. See Fig. 6.

Basic functioning of multiplexer is resolving multiple signal lines into one. Here, it is required to resolve multiple ports instead of single line. Therefore design is evolved as super multiplexer, resolving multiple ports into one port. This port becomes input to the super demultiplexer. Super Demultiplexer does opposite job i.e. resolving one port into multiple ports.

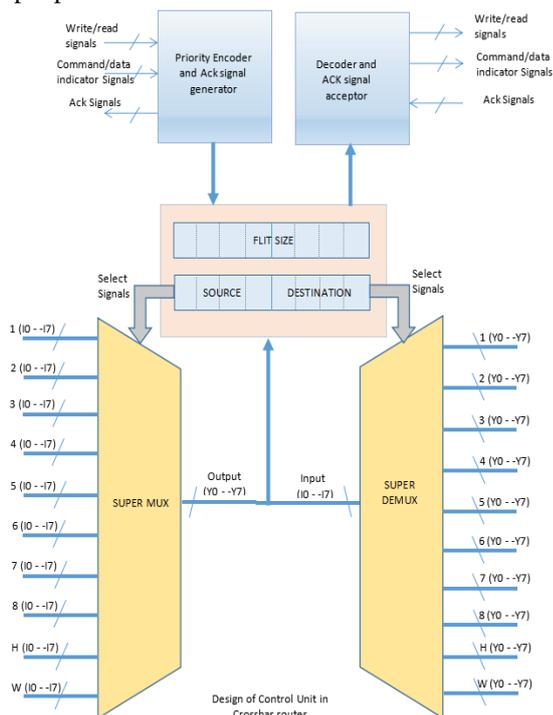


Fig.6. Crossbar switch/router design – II

Control unit decides which port should be connected to the output of the super multiplexer, depending on source. Here, source is router or processing element from which data packets are going to arrive at the cross bar switch. Next job of control unit is to decide output of super demultiplexer, depending on destination. Here, again destination is next router where data packet should go. Control unit of every crossbar switch have handshake signals (write_bar, read_bar and Ack) to implement GALS. Every crossbar switch and processing element will have separate clock and they will synchronized globally through handshake signals. With GALS, we need not have to distribute single clock across entire chip. This gives more robust and feasible design. Control unit is a priority encoder with fixed priority policy of NOC.

D. ASM Chart

ASM chart for router - This chart shows how router will work for every clock cycle, depending on work phase or flag settings. See fig. 7 and 8.

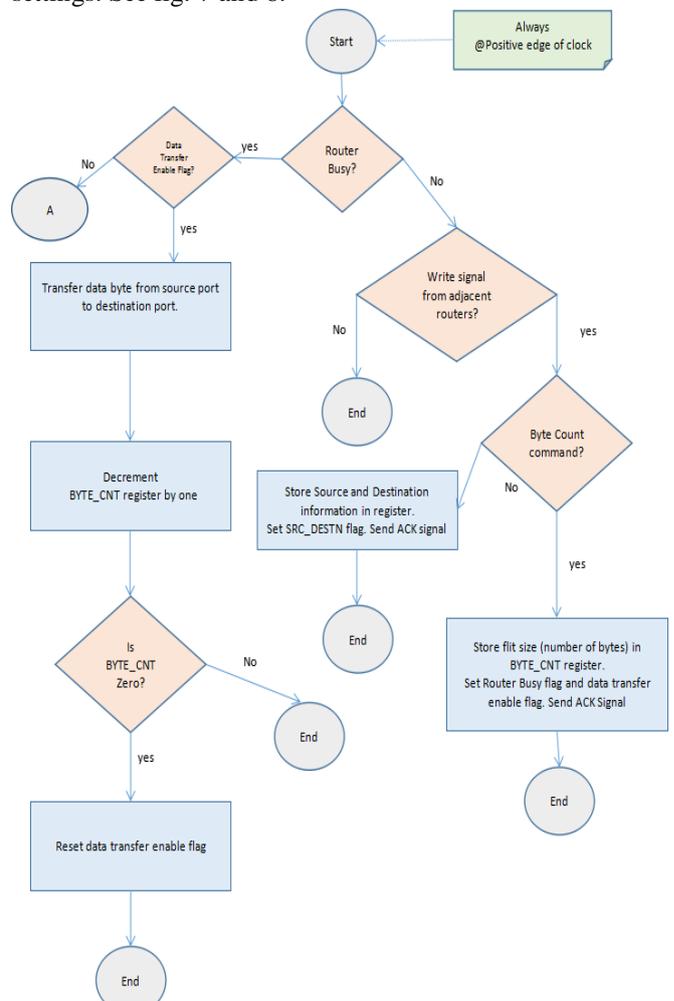


Fig.7. ASM chart for router - part 1

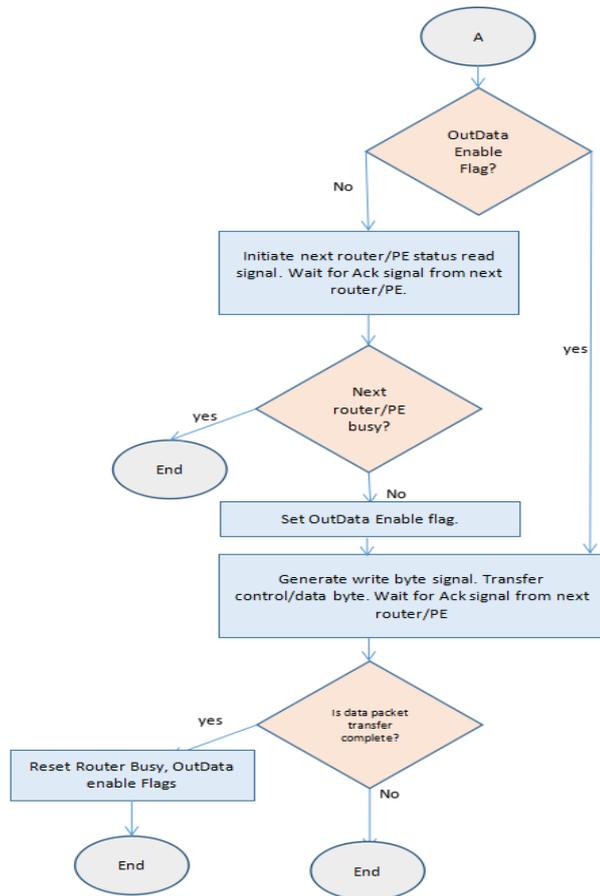


Fig.8. ASM chart for router - part 2

Router will work in three phases- 1. Accept packet header from previous router/PE. 2. Accept data flits and store them in output buffer. 3. Route received data packet to next appropriate router/PE as specified in the header. Every clock cycle one data byte will be received or transmitted. Since this globally asynchronous and locally synchronous NOC, if data byte reception or transmission is not successful through handshake signals router will introduce wait states until job is done.

E. Pseudo Code

Pseudo code for router - Pseudo code gives idea about how router architecture can be implemented using Verilog. It is included as Fig. 9 in this research article.

F. Interface Diagram

Interface between PE-router and router-router - In this NOC architecture, every NOC interface will carry two signals, i.e. actual data and handshake signals. Handshake signals will synchronize globally asynchronous devices in NOC. In interface between processing element and router, Network Interface Component (NIC) will have two parts, interface signals and corresponding software component. Interface signals again are divided into two parts i.e. 8 bit data bus and handshake signals. Processor will have two ports dedicated for these signals, data and control signals. Software component will control these ports and communication between router and PE. See Fig. 10 and 11. In interface between two routers, NIC software component is absent, as every thing is controlled by hardware in router.

```

Reg Busy;
Reg [8-0] SRC_DESTIN; Reg[3-0] SRC_DEST; MUX_DEMUX;
int COMMAND_W_1, 2, 3, 4, 5, 6, 7, 8, H, W, SRC_DESTIN_FLAG;
DATA_EN_FLAG, BYTE_CNT, FLIT_SIZE;
Wire WR_1, 2, 3, 4, 5, 6, 7, 8, H, W; Wire CDbar_1, 2, 3, 4, 5, 6, 7, 8, H, W;
Wire ACK_1, 2, 3, 4, 5, 6, 7, 8, H, W, CDbar;
//Invokes supermux and superdemux module
//Invokes priority encoder, decoder module
Always Begin
@ (posedge clk)
{ If (ROUTER_BUSY != 1) // SELECT I/p & O/p PORT
{ //code for fixed priority encoder
If (WR_1 == 0 & CDbar_1 == 1)
{ SRC_DESTIN = COMMAND_W_1;
SRC_DESTIN_FLAG = 1; ACK_1 = 1; Wait 20nsec;
ACK_1 = 0; }
If (WR_1 == 0 & CDbar_1 == 1 & SRC_DESTIN_FLAG == 1)
{ BYTE_CNT = COMMAND_W_1;
ROUTER_BUSY = 1; DATA_EN_FLAG = 1;
ACK_1 = 1; Wait 20nsec; ACK_1 = 0; }
// similar code for port 2, 3, 4, 5, 6, 7, 8, H, W
FLIT_SIZE = BYTE_CNT;
SRC = SRC_DESTIN[3-4];
DESTIN = SRC_DESTIN[5-0];
if (SRC == DESTIN) INVALID();
If (SRC == 1) MUX = (0000)b; // 1
If (SRC == 2) MUX = (0001)b; // 2
If (SRC == 3) MUX = (0010)b; // 3
If (SRC == 4) MUX = (0011)b; // 4
If (SRC == 5) MUX = (0100)b; // 5
If (SRC == 6) MUX = (0101)b; // 6
If (SRC == 7) MUX = (0110)b; // 7
If (SRC == 8) MUX = (0111)b; // 8
If (SRC == 9) MUX = (1000)b; // H
If (SRC == 10) MUX = (1001)b; // W
If (DESTIN == 1) DEMUX = (0000)b; // 1
If (DESTIN == 2) DEMUX = (0001)b; // 2
If (DESTIN == 3) DEMUX = (0010)b; // 3
If (DESTIN == 4) DEMUX = (0011)b; // 4
If (DESTIN == 5) DEMUX = (0100)b; // 5
If (DESTIN == 6) DEMUX = (0101)b; // 6
If (DESTIN == 7) DEMUX = (0110)b; // 7
If (DESTIN == 8) DEMUX = (0111)b; // 8
If (DESTIN == 9) DEMUX = (1000)b; // H
If (DESTIN == 10) DEMUX = (1001)b; // W
} //if router_busy != 1
} //always @ posedge
End
Always Begin @ (posedge clk)
{ CDbar = CDbar_1 | CDbar_2 | CDbar_3 | CDbar_4 | CDbar_5 |
CDbar_6 | CDbar_7 | CDbar_8 | CDbar_H | CDbar_W;
If (DATA_EN_FLAG == 1 & ROUTER_BUSY == 1)
{ If (FLIT_SIZE != 0 & CDbar == 0)
{ output_port[I] = input_port;
FLIT_SIZE = FLIT_SIZE - 1; I = I + 1; }
If (FLIT_SIZE == 0)
{ DATA_EN_FLAG = 0; } } }
End // similar way check next router busy flag
//Transfer data bytes to next router/PE as show in A-part of ASM
chart
//After transfer complete reset router busy flag
    
```

Fig.9. Pseudo Verilog Code listing of crossbar switch

G. Initialization of NOC

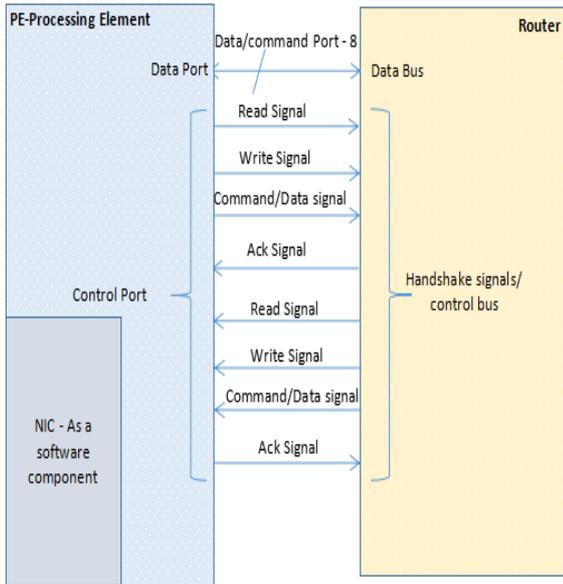
Initialization and Task Distribution in NOC - Static distribution of tasks is always simple, robust and less complex as compared to dynamic distribution. Here, in this research article, as a beginner, lets consider static distribution of tasks. Network start up will be as follows:-

- ✓ Reset signal will restart and synchronize (task wise) all routers and processing elements.
- ✓ Network start up program in PE will look for task download from summing PE.
- ✓ Once task distribution is done for all PEs, data distribution can be done as and when required or repeatedly.
- ✓ If task is complete, results will be uploaded to summing PE through NOC.
- ✓ Summing PE will have task distribution, data distribution and summing mode.
- ✓ Super summing PE will allot tasks to summing PE and summing PE will distribute tasks to leaf PEs through routers 1 to 8.
- ✓ Complete program will be stored in main memory. Later, it will be copied to local memories of individual PEs.
- ✓ Super summing PE will give us final result out of all tasks distributed.



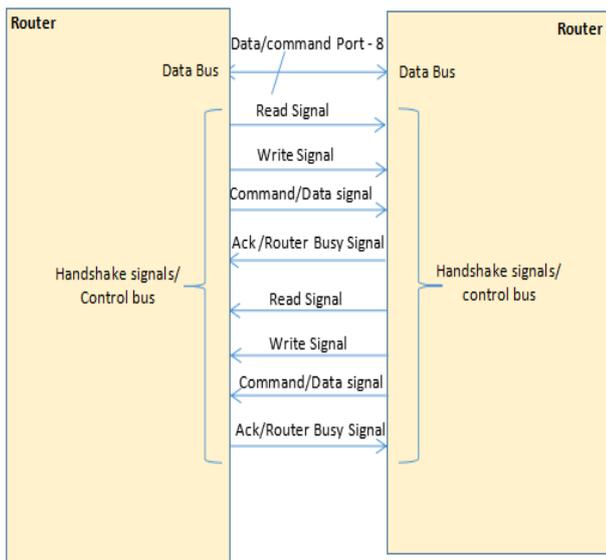
Area and Power Budget Estimation of Hierarchical Network Topology in Comparison with 2D Mesh Topology for NOCs and its Design and Implementation Oriented Overview

- ✓ Final results/reports will be stored in main memory by super summing PE.



Interface diagram between Processing Element and Router

Fig.10. Interface diagram -1



Interface diagram between two Routers

Fig.11. Interface diagram-2

H. Algorithm for NIC in PE

- ✓ Look for task download from summing PE
- ✓ Store task in local memory.
- ✓ Complete the task.
- ✓ Store results/report in local memory.
- ✓ Look for result/report upload to summing PE.
- ✓ Once data upload complete, look for next task download or data download.
- ✓ If it is just data download, complete processing as previous task.
- ✓ If it is task download, replace previous task for further data processing.

- ✓ In this design, router is just an another I/O device for PE. Policies for writing software for multiple PEs connected with NOC
- ✓ Looking at the NOC architecture proposed here, following policies will give best results:-
- ✓ Divide total task into small granular tasks
- ✓ Maximum communication should be between leaf PEs and summing PEs
- ✓ Communication between leaf PEs should be minimum
- ✓ Static distribution of tasks is better than dynamic task distribution
- ✓ Clear identification every software interface is must.
- ✓ Further study of software interfaces, network traffic, locality of data distribution, efficient use of resources, energy modelling can be done as bench-marking.

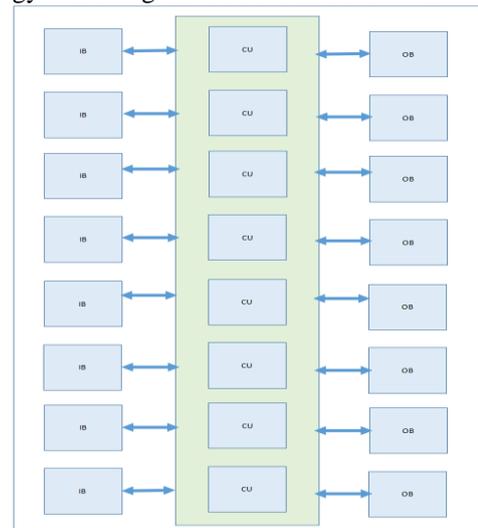


Fig.12. Parallel Arbiter

I. Scope for Upgradation

- ✓ In this research article, we looked for simple basic design.
- ✓ This can be further upgraded for smart energy distribution.
- ✓ Dynamic task distribution can be done for complex software.
- ✓ Networking priorities can be reshuffled intelligently.
- ✓ Presently it is virtual cut through network,
- ✓ it can be upgraded to warm hole routing.
- ✓ Topology proposed here, is best suited for multiple PE NOC,
- ✓ however one can go for further complex hybrid topology, like spidergone + fat tree + Hierarchical
- ✓ Here single arbiter is included; one can go for parallel arbitration for fast and heavy traffic. See Fig. 12.

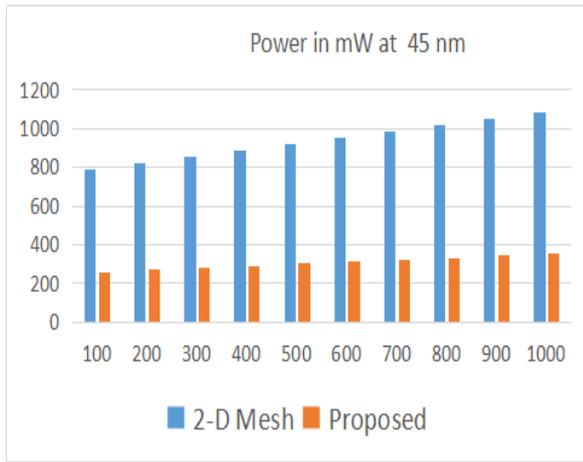


Fig.13. Power comparison

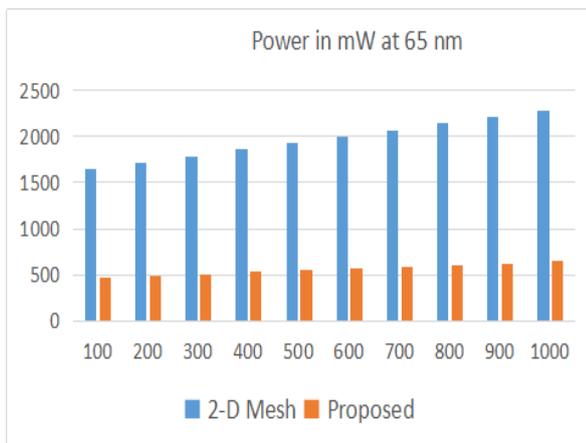


Fig.14. Power comparison

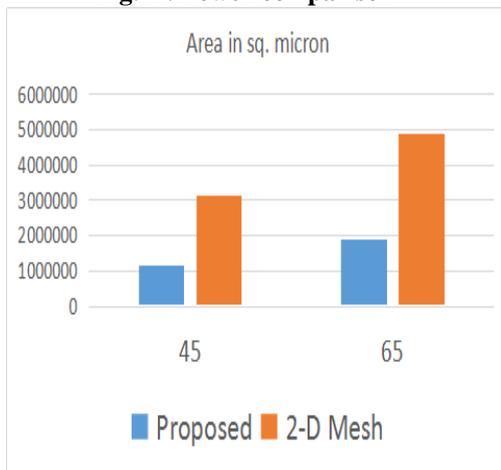


Fig.15. Area comparison at 45 nm & 65 nm (On X axis Technology in nano meter and On Y axis Area in Sq. micron)

VII. CONCLUSION

Proposed topology is estimated to have lower power and less area as compared to 2-D Mesh topology. Estimated power for proposed topology, at 45nm is around 37% of 2-D Mesh topology and at 65nm is around 28% of 2-D Mesh topology. In Fig. 13, power comparison at 45nm technology between proposed router topology and 2-D Mesh topology at different frequencies is shown and Fig. 14, power comparison at 65nm technology between proposed router topology and 2-D Mesh topology at different frequencies is

show. Fig. 15 illustrates area comparison. Proposed topology is well suited for multi-core SoC for low power application. Area required for routers in proposed topology is 38% of area required for 2-D mesh topology. Maximum number of hops required for packet routing in proposed topology is 3 compared to 14 in 2-D mesh. Proposed topology is upward scalable to interconnect 100 cores or 144 cores. For the expansion of 100 core, base router will connect to 10 cores and for 144, base router will connect to 12 cores.

Design and Implementation oriented overview gives gross level idea about how NOC can be implemented. In this section it shown that 64 PEs can be connected efficiently with hierarchical topology. ASM chart and pseudo code are given in support of proposed topology.

ACKNOWLEDGEMENT

We would like to thank Bharath University for encouraging us to write this research article. Orion 3 simulator is perfect tool for estimating power and area budget of the topology. We thank Mr. Anil Terkar, Scientist, D.R.D.O. for his valuable help and guidance.

REFERENCES

1. Meaad Fadhel Ali Qasem and Huaxi Gu, "Square-Octagon Interconnect Architecture for Network-on-chips", IEEE 2014
2. Ran Manevich, Israel Sidon and Avinoam Kolodny, "Design and Dynamic Management of Hierarchical NoCs", Microprocessor and Microsystems 40 (2016) pages 154-164.
3. Mahmoud Moadeli, Partha Maji, Wim Vander bau whedel, "Quarc: a High-efficiency Networ on-chip Architecture", 2009 International conference on Advanced Information Networking and Application
4. M. Coppola, R Locatelli, G. Maruccia, L., "Spider gon: a novel on-chip communication network," International Symposium on System-on-Chip, pp.15, Nov. 2004.
5. Giovanni De Michel, Luca Benini, "Network On Chips", Morgan Koufmann
6. YuanZhou, Gang Jian, Guodong Han, "Parametric Hierarchical Mesh Interconnected Structure for Network On Chip", IEEE 2015, 6th International Conference on Software Engineering and Service Science
7. Mostafa Khamis, Amir Zaytoun , Ahmed Shalaby, "Evaluating the feasibility of Centralized Router For Network on chip", IEEE 2015, 27th International Conference on Microelectronics.
8. A Q Ansari, M R Ansari, M A Khan, "Performances Evaluation of Various Parameters of Network-On-Chip (NoC) for Different Topologies", IEEE 2015 India Conference
9. Andrew B. Kahng, Bill Lin, Siddhartha Nath, "Orion 3.0 : A Comprehensive NOC Router Estimation Tool" Explicit Modeling of Control and Data for Improved NoC Router Estimation", IEEE Embedded Systems Letters (Volume 2, Issue 2, June 2015)