DESIGNING MULTI-CORE ARCHITECTURE USING FOLDED TORUS CONCEPT TO MINIMIZE THE NUMBER OF SWITCHES

A Thesis by

Sri RamyaChaturvedula

Bachelor of Engineering and Technology, Jawaharlal Nehru Technological University, 2009

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2011

DESIGNING MULTI-CORE ARCHITECTURE USING FOLDED TORUS CONCEPT TO MINIMIZE THE NUMBER OF SWITCHES

DEDICATION

ACKNOWLEDGEMENTS

I am grateful to my thesis advisor Dr. Abu Asaduzzaman for his support, encouragement, and supervision. He alwaysuzs thn mhnfor gnceuino euof his sy seand

my(a)4(ii)3(tuded)2571tm Dr. aPi(e)4(d(s)-10(e)4.)-596for his vaee advcHe sgeois and

asta(nc54(e)4witho)-21(m)-11(y)20( )-19(a)4czac9 5(s)-20ino utimi(e)4lmy320( )-19(manei)3(r.r)-6( )-

ABSTRACT

A multi-core system provides improved performance/power ratio than a single-core one. However, multi-core architecture suffers from thermal constraint and data inconsistency. Current multi-core system is not adequate to increase memory level parallelism and cache performance due to its poor core-to-core interconnection topology. In some architecture, like MIT Raw, each node/core has computing and switching components. Switching component of such a node consumes power while the node is only computing and vice versa. In this paper, we propose a design methodology to reduce the number of switches in multi-core architecture without compromising the performance. According to this method, nodes are separated between computing cores and network switches. Using folded torus topology, we develop a scheme to connect the components (cores and switches) in the multi-core architecture. We use multi-core architectures with various numbers of nodes (cores and switches) to evaluate the proposed methodology. Using synthetic workload, we obtain the core-to-core communication delay and total power consumption for MIT RAW, Triplet Based Architecture (TriBA), Logic Based Distributed Routing (LBDR), and the proposed architecture. Experimental results show that the proposed architecture outperforms Raw, TriBA, and LBDR by cutting down the need for the number of switches significantly. According to the results, proposed architecture reduces total power consumption approximately by 77% and average delay by 54%. Power reduction comes from the fact that number of switches is cut down. Average delay is reduced as each switch provides adequate communicate channels.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Network Topologies

,Q HYHU\GD\ VFHQDULR ³QHWZRUN´ LV WKH PRVW JHQ

Communication among different physical nodes is defined as Network. There exist various topologies for having connections among these nodes. Multi-core is such a technology where multiple cores communicate with each other to process a job. Here each core is considered as a node that is needed to be connected to other cores in a multi-core environment. With the advancement of Network on Chip (technology) chip network architecture can be explained through four parameters: topology, routing algorithm, flow control protocol and router micro architecture. Topology term in networking is defined as the how the links are connected between the nodes. Using topology of nodes all the possible paths from a particular source and destination pair can be determined. Using Routing algorithm the best path to from a source and destination pair can be identified. Using flow control protocol more details about the path selected from a source and destination pair is stored. The details include message traversal of the assigned route, when a message leaves a source node and also the time the path must be stored or buffered for future usage. Micro architecture of a networking component analyzes all the above parameters and uses it for network implementations.

In this we mainly concentrate on the topology parameter. A proper topology for a network is highly necessary for a better cost-performance on the whole network. The effect of topology while analyzing parameter is very important. Using topology of a network one can determine the number of hops a message from a source node should traverse before reaching the destination.

In general networking world different topologies like bus, mesh, ring topologies are extensively used. Figure 1.1 and Figure 1.2 shows the general ring and mesh topologies respectively.

Each topology has its own prototype for the nodes to be connected. In conventional architecture most of the designs have the cores connected in a mesh topology format. When these cores are connected in a multi-

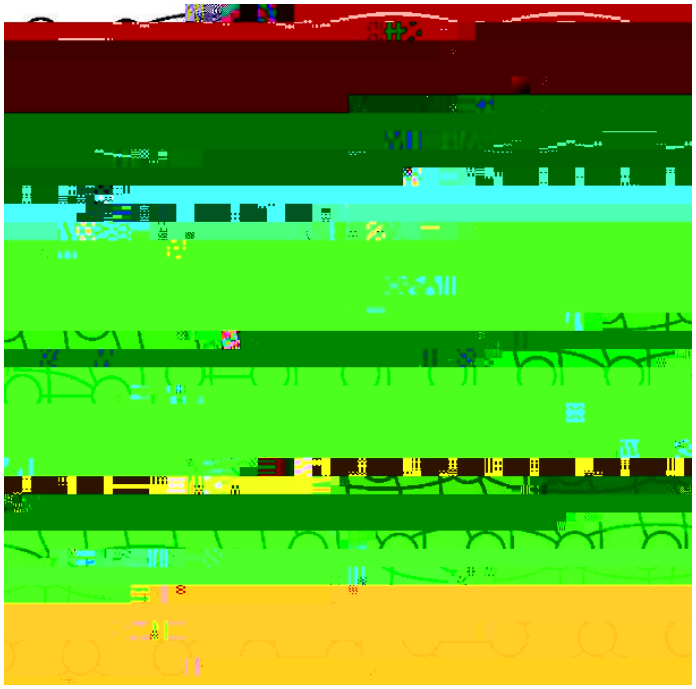

Figure 1.3:Folded Torus Network Topology [4]

As perFigure1.3, in folded torus topology every node has a link to its adjacent node in

processor improvises the computational capacity of the processors through parallel computing technology[13]. Multi-core architecture shares required resources like memory to process an application in a parallel manner. For successful and efficient processing of an application it is necessary that each core should have sufficient resources according to their respective tasks. Hence, design of multi-core architecture to utilize the available resources is very important. As discussed before nowadays multi-core designs uses bus topology to connect different cores. Designing of multi-core architecture can be done based on various components of cores in the architecture. For example designing can be done based on memory usage of architectures. In present days multi-core architectures adopt isomorphic architecture [14-16]. in this kind of architecture each core will have its own first level Cache, shared second level Cache through a bus. Also, Triplet Based Architecture [TriBA] is another kind of architecture where group of 3 cores will have common shared memory. Designs are proposed basing on the most common problem Deadlock. Deadlock is such a situation that occurs in multi-core environment where threads get stuck forever in a clash over access to shared resources or memory [17].

In [18] polling-transmission policy was discussed to solve the deadlock problem which occurs in intermediate nodes of a multi-core architecture network. This algorithm uses Hypercube topology for implementation.

Due to recent technological revolution, majority of the embedded systems are implementing more than one core for faster and efficient computations. When implementing multiple cores in a single chip it is very important to design the multi-core for efficient usage of chip volume. With this recent trends, billions of transistors are integrated on the same chip possibly. With the same capacity of chips designers are implementing multiple computing and memory cores on a single chip. This ensures computational tasks to be performed in efficient and fastest way. To design

Using that topology we are proposing a design for multi-core architecture to utilize more space in a given mesh and have an efficient communication between the cores with minimal number of switching components on the mesh of cores.

1.2.1 Switches in Multi-core Architecture

It is necessary in multi-core architecture cores should pass on information to other cores on the chip to process a single application. For the cores to communicate with each other networking components like switches or routers are necessary. In this thesis we use only the term switch for networking component. Switches will actually establish a communication channel between different cores. Depending on the source and destination parameters switches will transmit packets accordingly. Network on chip (NoC) is the famous term used in nowadays multi-core environment. Very active research is going on the same NoC technology.

expected to support the parallel communication patterns on demand to increase the data throughput [2]. Considering all these parameters and all the functionalities of a networking component in a multi-core environment, we are proposing a design based on number of switches that are utilized in folded torus based multi-core design. In the following sections of the document all the details about the design and other advantages of the proposed design are explained.

1.2.2 Raw Architecture

Raw Architecture from MIT is extensively analyzed for proposing this new design. This architecture uses mesh topology and processes an application. It considers the cores as tiles and each tile has a switching component, computing component and other components like cache main memory. More details about this architecture are discussed in the next chapter. The major disadvantage of this architecture in case of nxn mesh topology is that there exists n^2 number of switches and n^2 number of computing components. Having more number of switches will increment the energy consumption. Considering this disadvantage in the following section the Problem Description is discussed.

1.3 Problem Description

In the present designs of multi-core architectures it is observed that most of the widely used topologies are mesh, ring and bus based topologies. Following are general topological views of mesh and ring topologies.

These topologies have issues like High power consumption and latency. Also, due to large number of networking components in the architecture network complexity increases. Folded

torus topology is identified as a better topology for having multiple links among the nodes in th

1.5 Thesis Organization

In chapter 2, we presented some of the related architectures that are already existing and well approved from various published journals and conference papers.

In chapter 3, we explained the proposed multicore architecture and the approach to understand the methodology.

In chapter 4, we evaluated the proposed architecture by using synthetic workload and comparing with the selected existing architectures.

In chapter 5,

CHAPTER 2

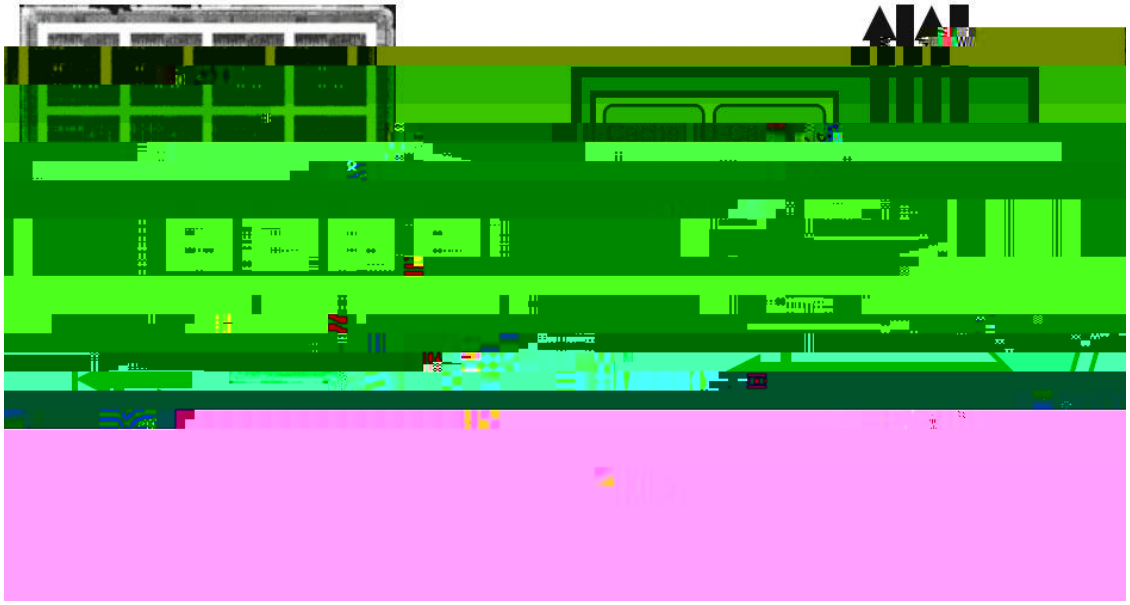

Figure 2.1: Raw Architecture Tile [52]

The tiles in this design are interconnected with several components like routers, programmable switch, switch instruction, data memory, ALU. Firstly, Raw architecture sees to implement fine grain parallelism in a more efficient way. Secondly, Raw architecture is designed to provide all the details about the hardware system in an architecture for the software system integrated with the architecture, such that scheduling and routing are taken care without any conflicts between the cores for shared resources. There has been some advancements in the Raw architecture [17, 25-27] proposed by Michael Bedford Taylor. This new architecture looked into concept of having static and dynamic networks for communication among the tiles. Static networks define a fixed communication channel before the compile time and the compiler exactly know where to send the message. In this static network communication each Raw tile is connected to its nearest neighbors through a series or separate, pipelines channels [17, 25-27]. In this advancement the behavior of the FPGA prototype is mainly taken care.

Also, Dynamic network-communication is proposed to avoid the situations wherein the memory requirement cannot be decided before the compilation time. It uses a header and implements some protocols for dynamic routing between the tiles [6].

In spite of all the above advantages due to more switching components in a multi-core architecture the heat dissipation has become the prime concern. In our proposal we are looking into that disadvantage by decrementing the number of switching components.

2.2 Triplet Based Architecture

Triplet Based Architecture is another design model for multi-core architecture which also looked into the drawbacks of having large number of switching components. TriBA a new idea in multi-core architectures and a direct interconnection network (DIN), is compared with 2D Mesh on single chip multi core architecture. TriBA consists of a 2D grid of small, programmable processing units, each physically connected to its three neighbors so that advantageous features of group locality can be fully and efficiently utilized for getting maximum out of an on-chip Interconnection of cores. Cores on the same chip are connected via a triplet based hierarchical interconnection network (THIN), which has simple topology and computing locality characteristic [7]. TriBA basically lo

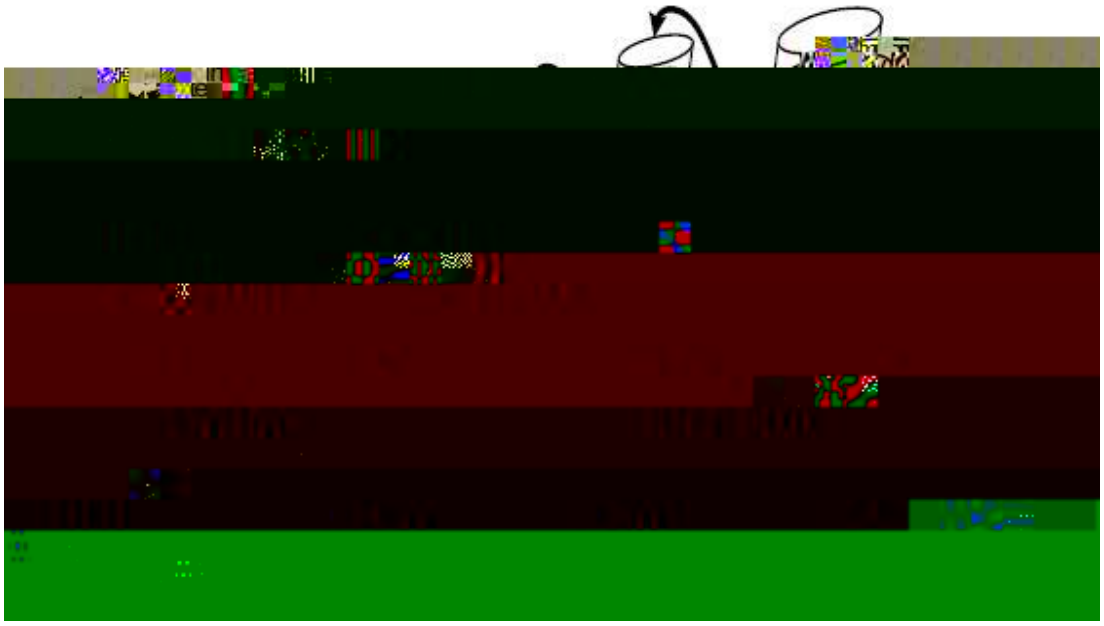

Figure 2.3:Memory Allocation Strategy in TriBA [22]

As shown in figure 2.3, three nodes (cell) are connected to each other in triangular pattern in TriBA. Each node has its local memory L1, while three nodes share a common L2 memory [25].

Prime motivation for our proposal is a new architecture. There exist wide varieties of network

Based Distributed Routing (LBDR). In the proposed design for LBDR, 4 cores are connected to a single switch and all switches are connected to each other.

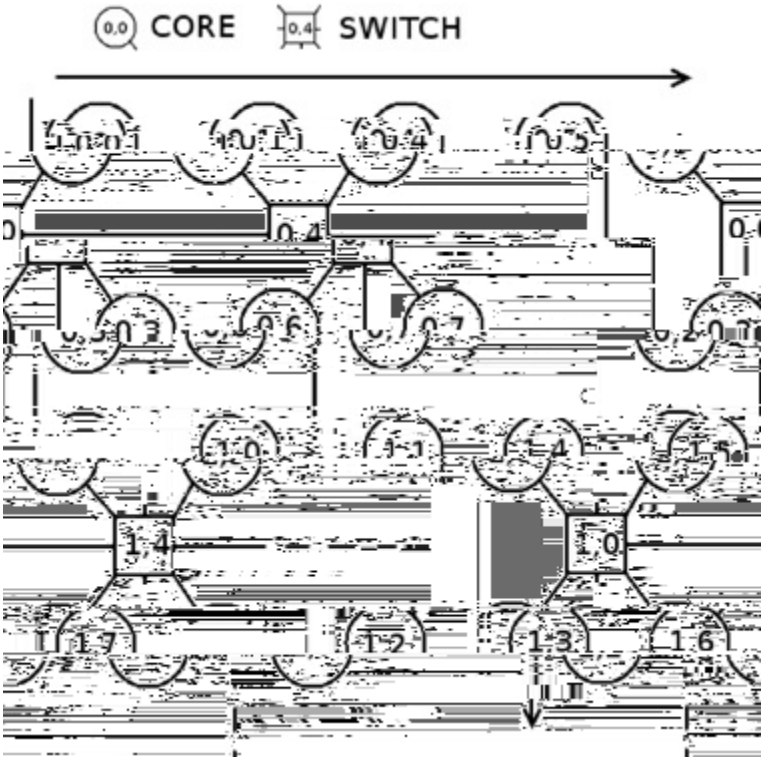

Figure 24:Architecture used to Implement LBDR3[2]

LBDR mechanism [2] was extended to support multiple cores per switch. Figure 2.4 shows the topology where the LBDR is implemented and has multiple cores connected to each switch. In the above design it is shown that 4 cores are connected to a single switch. Motivated from the same, we proposed a new design which has multiple cores connected to the switches instead of having switching and computing components in the same node.

each probe contains a header with 3 different fields. The first field in the header contains 2 bits. There are 2 bits in the priority field, one for the interlane priority and the other for the intra-lane

CHAPTER 3

PROPOSED MULTI-CORE ARCHITECTURE

In this proposed design main goal is to reduce the number of switching components and thereby reducing the power consumption and heat dissipation. The design is mainly based on Folded Torus based network topology. In present multi-core architectures each core consists of a switching component and a computing component. Thus if nxn mesh topology is considered there exists n^2 switching components and n^2 computing components. In this proposed design, basing Torus topology a novel design is implemented for multi-core architecture to reduce the number of switches and utilize maximum silicon area on a chip. Considering nxn mesh topology every third node in a column or a row is considered as switch, such that reducing the number of switches from n^2 to considerable number of switches by following an algorithm. All the remaining number of nodes is considered as computing components or cores. In this design it is made sure that all the cores have equal number of switching components connected to have proper communication among the cores.

3.1 Node Selection

In the Proposed design unlike Raw architecture, considering the nxn mesh topology a few nodes are considered to be the switches and a few considered to be exclusively computing nodes

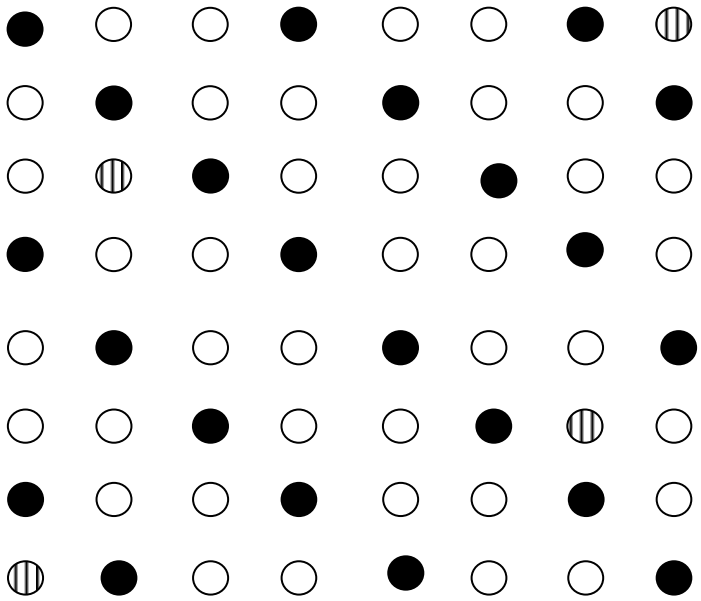

Figure 3.2: Selecting Computing Nodes and Special Node

In the Figure 3.2 the striped nodes are the special nodes that are used to connect 3 different layers in case of 8x8 mesh topology. The details about the connections between are discussed in the later sections.

3.2 Node Connections

The important part of this proposed design is the way the nodes are connected and the way they communicate for efficient processing of applications. The idea behind the Folded Torus Network Topology is used while defining the connections among nodes. It is important in any network topology to have proper connectivity among the switches. In multi-core architecture switches are the major components which synchronize all the computation data of the cores and provide a final result by collaborating with all the results obtained from each core involved in a process. Cores will not be able to communicate with their neighboring cores without networking

3.2.2 Connecting Computing Nodes

As mentioned in the previous sections, folded torus network topology idea is used to connect computing nodes and switching nodes. In folded torus topology each node will have a connection to a node at a distance of 2 units. Similarly in this proposed architecture each computing component will have a link to a switch that is at a distance of 2 units and at a distance 1 unit.

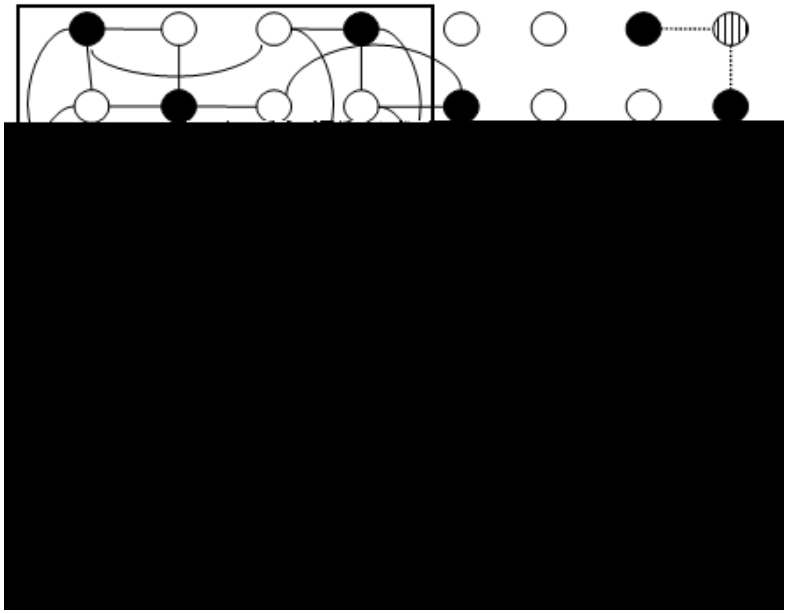

Figure 3.4: Connections between Switching Nodes and Computing Nodes

In Figure3.4 the nodes in the square box (higlighted) is an example of nodes having connections with the switches. Computing component will have links only to the switches. They will not have any link to other computing components.

3.2.3 Connecting Switching-Computing Nodes

As discussed earlier there are very few which acts as both computing and switching nodes. These nodes will have a direct link to its adjacent node at a distance of 1 unit. In the Figure 3.4 the dotted lines between the striped nodes and normal nodes shows the connection between the special nodes and computing nodes.

All the connections among the nodes in the proposed design can be summarized using the following algorithm

Connections are made using the following algorithm:

- A connection to every switch adjacent to core
- Starting from the initial node every third node in a column or a row is considered to be the switch
- Connection to a switch at a distance of 2 physical units until the number of connections to each core reaches 3
- There exists no core-to-core connection
- Also every switch is connected to its nearest neighboring switches
- Every node is identified with a proper location id on the network
- We can find all switches connected to each other in a cluster form.
- Exceptions for some nodes in making them as switching components to have all the switches in the network have proper communication
- Identifying those nodes according to the location.
- Every core has equal of number switches connected to have uniform resources available to each core.

Number of switches without exceptions in any kind of nxn mesh topology can be calculated using the following loop sequence.

```
i   number of rows in a mesh topology
j:number of columns in a mesh topology
k=0; //switch counter
        for (i=1;i< number of nodes in a row;i++)
        {
                for(j=1;j<number of nodes in a column;j++)
                        {
                                Counter=i%3;
                                if (j%3==Counter)
                                {
```

minimum of three connections to switches. This proposed design takes care that all the computing components have the uniform amount of network resources for efficient communication with the other computing components in the given architecture. All the nodes in the given nxn network are efficient used in the proposed design.

Figure 3.6: Distinguishing Different Layered Nodes

The Figure 3.6 shows all the different layers in the proposed design. All the switches which have the difference of 2 units between their positions belong to the same layer. For 8X8 mesh network there comes 3 layers. All the nodes of same layer are indicated with the same pattern in Figure 3.6. All the solid nodes indicate layer 1 nodes. As shown before striped nodes are the exception nodes to connect the different layered switches. Large checker patterned and 80 percent solid patterned nodes indicates layer 2 and layer 3 nodes respectively. The same layered nodes are

identified by the column they are present in the network. Following loop sequence is used to identify the switches of the same layer.

```
i: number of rows in a mesh topology
j:number of columns in a mesh topology
k=0; //switch counter
        for (i=0;i< number of nodes in a row;i++)
        {
                for(j=0;j<number of nodes in a column;j++)
                        {
                                Counter=i%3;
                                if (j%3==Counter)
                                {
                                sw[k] =a[i][j];
                                sw= sw+1; //counter for number of switches//
                                        k++;
                                if (Counter==0)
                                //sw[k] belongs to layer 1 with solid filling nodes
                                if(Counter==1)
                                //sw[k] belongs to layer 2 with checks filling nodes
                                if(Counter==2)
                        //sw[k] belongs to layer 3 with light dotted filling nodes color
                                }
                        }
                }
```

If a switch in a particular layer has to communicate with another switch in a different layer it has to communicate through the special node that is indicated with striped pattern. Exception acts as both switching component and computing as in RAW architecture. Basically it acts as a tile in the RAW architecture. Routing algorithms can be implemented depending on the layered structure and the position of the special nodes in the mesh. The proposed design is mainly focused on 8X8 mesh topology and algorithm is generated for a mesh network. When compared with other existing architecture this design has the capability of utilizing more si

CHAPTER 4

EVALUATION

4.1 Assumptions

For each design among Raw, TriBA, design used to implement LBDR and proposed design all the nodes are numbered rowwise for Raw and proposed design. The nodes in TriBA are numbered from top to bottom in triplets. For convenience the nodes in the design that is used for LBDR are numbered in a sequential manner in multiples of 5. We considered 16 nodes, 36 nodes and 64 nodes to calculate power consumption. While calculating results for each of nxn nodes 5 cases are considered. Each case shows the number of units of power consumption when one core tries to communicate with other core in the multi-core architecture for each of the designs that are analyzed in this paper.

Following figure shows an example of the numbering convention of nodes followed for Raw architecture and proposed design for calculating results.

The Figure 4.1 shows a 4x4 mesh which contains 16 nodes. With the increase in number of nodes the nodes are renumbered as per rows and columns in the mesh. While evaluating the power consumption and communication delay are calculated for different cases when cores at random places in a given topology are communicating. It is assumed that computing components consume more power when compared to switching components in a core environment. For Raw and TriBA designs it is considered that all the nodes have both switching and computing components. Hence, the power consumption by each in the path is 3 units. For switching component alone power consumption is considered as 1 unit. As discussed power consumption for computing component alone is considered as 2 units (>1). Thus, having all assumptions in place and considering different cases evaluation of new design is explained in the following sections.

4.2 Synthetic Work Load

In this section, all the different cases for each kind of topology of Raw, TriBA, LBDR and proposed are tabulated. These tables are used for evaluation of power consumption and communication delay for the 4 architectures.

Following table indicates has 6 columns and 5 rows. The first column indicates the case numbers. The second column indicates the source and destination nodes for that particular case. The other columns indicate the node numbers in each of the architectures, which a packet has to traverse from a source to destination. As assumed for Raw and TriBA architecture all the nodes are considered to have switching component and computing component. But in case of LBDR DQG 3URSRVHG WKH V\PERO ³VZ´ GLIIHUHQWLDWHV WKH FR ³VZ´ EHVLGH LW LV FRQVLGHUHG DFFRUGLQJO\ DOO WKH DVVXPSWLRQV DUH FKLQJ considered while evaluating.

Table 1a: Communication paths for Raw and TriBA in case of 16 nodes topology

Communication paths for 16 Nodes			
	Source-Destination	Raw	TriBA
Case 1	Node 2 – Node 15	2,3,7,11,15	2,4, 6, 13 ,15
Case 2	Node 3 – Node 14	3,2,6,10,14	3,2,4,6,13,14
Case 3	Node 7 – Node 15	7,11,15	7,8,6,13,15
Case 4	Node 2 – Node 10	2,6,10	2,4,5,10
Case 5	Node 8 – Node 14	8,12,16,15,14	8,6,13,14

Table 2a: Communication paths for Raw and TriBA in case of 25 nodes topology

Communication paths for 25 Nodes			
	Source Destination	Raw	TriBA
Case 1	Node 2 – Node 24	2,7,12,17, 22,23, 24	2,4,5,10,11, 22,24
Case 2	Node 3 – Node 23	3,8,13,18,23	3,2,4,5,10,11,22, 23
Case 3	Node 8 – Node 24	8,13,18,23, 24	8,6,5,10,11, 22,24
Case 4	Node 2 – Node 20	2,7,12,17,18,19,20	2,3,7,19,20
Case 5	Node 9 – Node 23	9,14,19, 24,23	9,8,6,5,10,11, 22,23

Table 2b: Communication paths for LBDR and Proposed Architectures in case of 25 nodes topology

Table 3a Communication paths for Raw and TriBA in case of 36 nodes topology

Communication paths for 36 Nodes			
	SourceDestination	Raw	TriBA
Case 1	Node 2 ±Node 35	2,8,14,20,26,32,33,34,35	2,4,6,13,15,31,33,35
Case 2	Node 3 Node 34	3,9,15,21,27,33,34	3,7,9,19,20,34
Case 3	Node 9 ±Node 35	9,15,21,27,33, 34,35	9,19,20,24,35
Case 4	Node 2 ±Node 30	2,8,14,20,26,27,28,29,30	2,4,6,13,14,28,30
Case 5	Node 12 ±Node 34	12,18,24,30,36, 34	12,14,15,17,18,20, 34

Table 4a: Communication paths for Raw and TriBA in case of 49 nodes topology

Communication paths for 49 Nodes			
	Source-Destination	Raw	TriBA
Case 1	Node 2 – Node 48	2,9,16,23,30,37,44,45,46,47,48	2,4,5,12,25,26,48
Case 2	Node 3 – Node 47	3,10,17,24,31,38,45,46,47	3,2,4,5,10,12,25,26,46,47
Case 3	Node 10 – Node 48	10,17,24,31,38,45,46,47,48	10,12,25,26,26,46,48
Case 4	Node 2 – Node 42	2,3,4,5,6,7,14,21,28,35,42	2,4,5,10,11,22,23,40,42
Case 5	Node 13 – Node 47	13,12,19,26,33,40,47	13,14,12,25,26,46,47

Table 4b: Communication paths for LBDR and proposed architectures in case of 49 nodes topology

Communication paths for 49 Nodes			
	Source-Destination	LBDR	Proposed
Case 1	Node 2 – Node 48	2,1(sw),6(sw),11(sw),41(sw),46(sw),48	2,22(sw),43(sw),46(sw),48
Case 2	Node 3 – Node 47	3,1(sw),6(sw),11(sw),16(sw),41(sw),46(sw),47	3,4(sw),7(sw),28(sw),49(sw),47
Case 3	Node 10 – Node 48	10,6(sw),11(sw),16(sw),41(sw),46(sw),48	10,17(sw),38(sw),41(sw),48
Case 4	Node 2 – Node 42	2,1(sw),6(sw),11(sw),16(sw),41(sw),42	2,4(sw),7(sw),28(sw),49(sw),42
Case 5	Node 13 – Node 47	13,11(sw),31(sw),36(sw),46(sw),47	13,12(sw),33(sw),47

Tables 4a and 4b shows the node-to-node communication in case of 49 nodes topologies for all the 4 architectures

4.3 Output Parameters

As mentioned in the assumption section, while evaluating power consumption, the power consumed by the switch is considered to be less when compared with power consumed by the core node. The logical reason for this kind of assumption is the basic behavior of a computing component and switching component. When a packet arrives a switching component it will just

Below is the graphical representation of the calculated results for power consumption in each case.

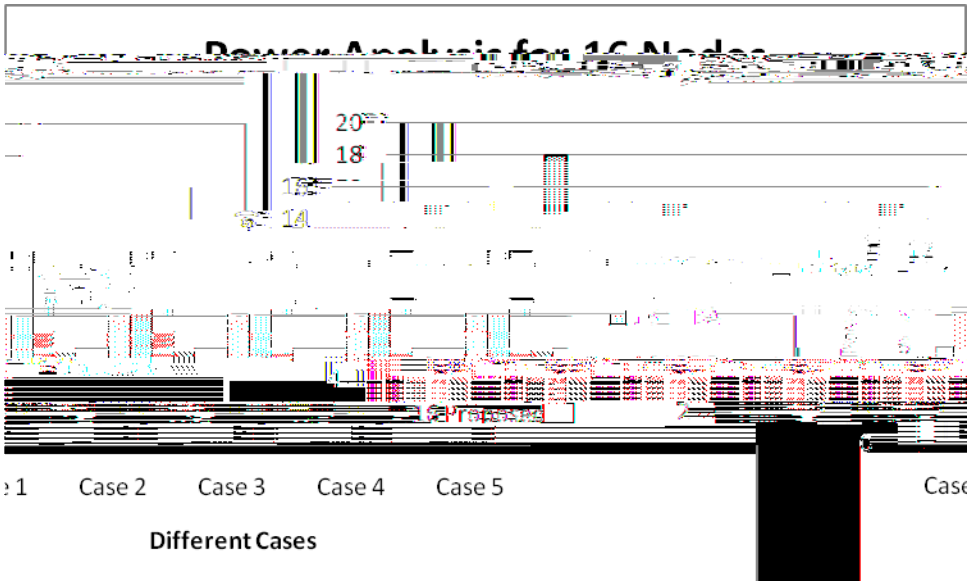

Figure 4.3:Power Analysis for 16 Nodes

The graph in Figure 4.3, indicates that the power consumption in case of proposed architecture is

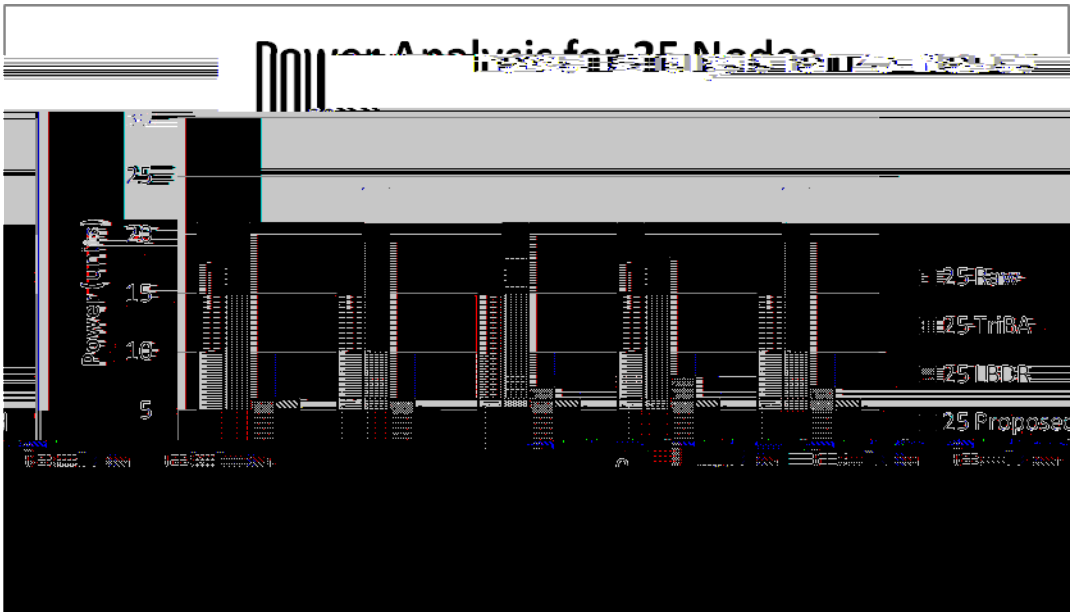

Figure 4.4:Power Analysis for 25 Nodes

The graph in

The graph in Figure 4.5, indicates the power consumption analysis for the tested 3 architectures and the proposed architecture in case of 36 nodes topology in each architecture.

The graph in Figure 4.7, indicates the power consumption analysis for the selected 3 architectures and the proposed architecture in case of 25 nm topology in each architecture.

Thus after going through all the graphs it can be observed that the power consumption in all the cases for all kinds of topologies is more efficient for proposed design when compared with remaining 3 architectures.

4.6 Comparison of Communication Delay

As described in the previous results section, in this section of results communication delay is calculated for different cases when cores at random places in a given topology are communicating. As discussed in the introduction latency is a term which refers to the delay for a message to reach its destination while it traverses the path between the source and destination. Hence, it can be inferred that the less number of hops a message traverses from source to destination the less would be delay. Therefore in the section of results communication delay is measured in terms of number of hops required for a core to communicate with the other core in each of the analyzed cases. 5 cases are considered for 16, 36, 64 nodes. In each case all the values of the obtained for the proposed design are compared with Raw, TriBA and with the design used for LBDR. Values obtained are the number of hops. Following graphs show the comparison of the same. The cases considered for c

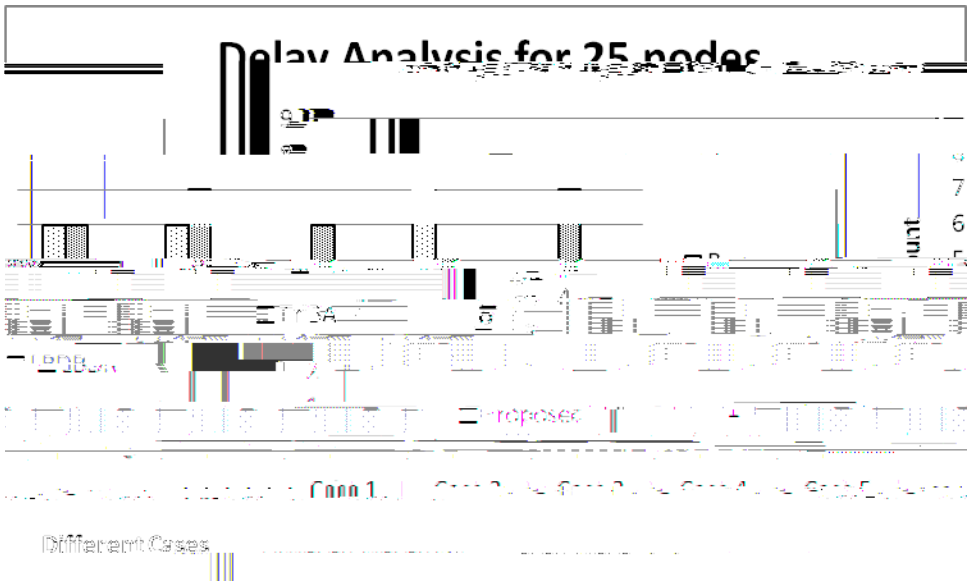

Figure 4.9:Delay Analysis for 25 Nodes

The graph in Figure 4.9, indicates the delay analysis for 25 nodes topology of all the 4 architectures.

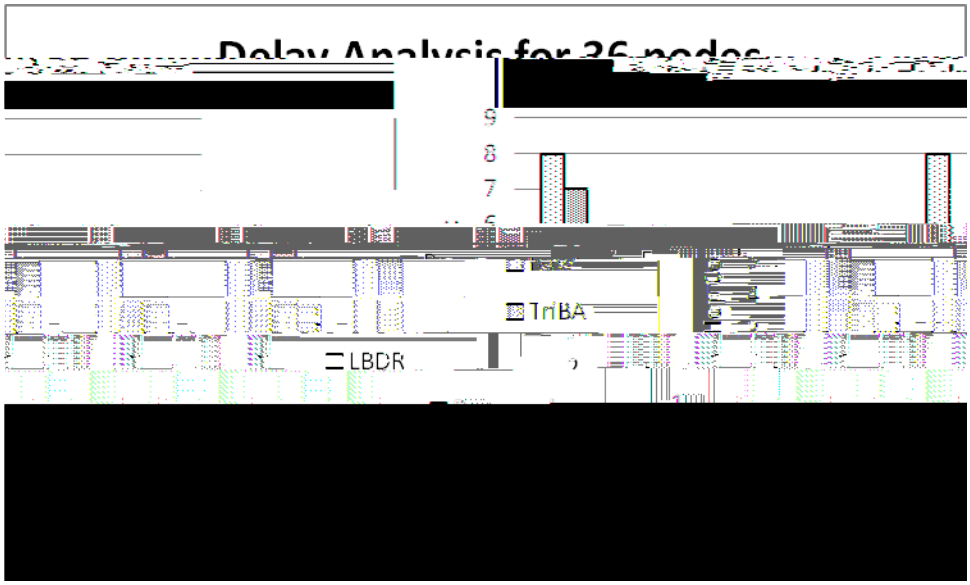

Figure 4.10: Delay Analysis for 36 Nodes

The graph inFigure4.10, indicates the delay analysis for 36 nodes topology of all the 4 architectures.

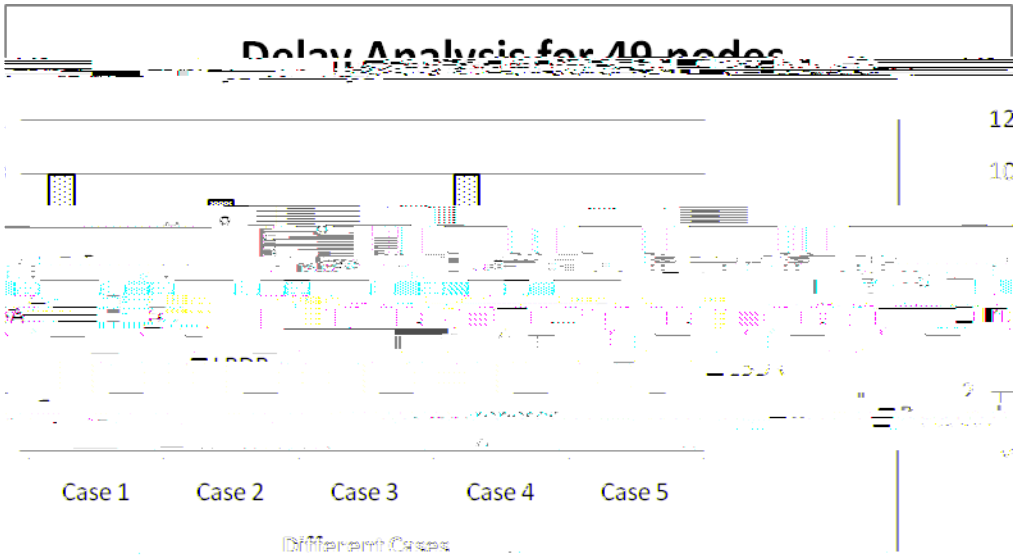

Figure 4.11: Delay Analysis for 49 Nodes

The graph inFigure4.11, indicates the delay analysis for 49 nodes topology of all the 4 architectures.

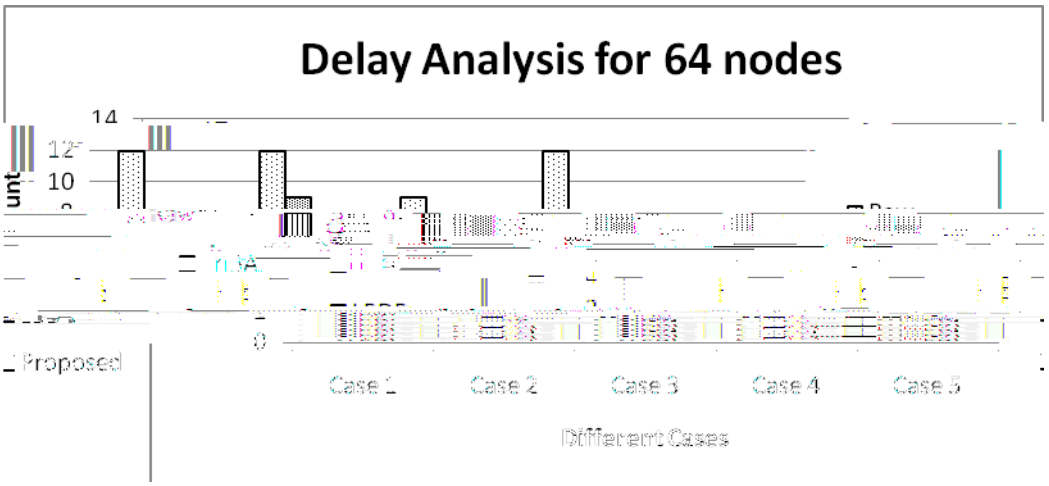

Figure 4.12:Delay Analysis for 64 Nodes

The graph in Figure 4.12, indicates the delay analysis for 64 nodes topology of all the 4 architectures.

From the above results it can be concluded that the communication delay in case of the proposed design is less when compared with remaining 3 designs in-order architecture.

4.7 Summary and Observations

From the above evaluations we can summarize that proposed architecture performs better than the other 3 selected architectures. After analyzing all the available values that are used for evaluation it is observed that the power consumption in case of proposed architecture is approximately 77% lesser than the Raw architecture from MIT in case of 64 nodes mesh topology. Similarly, it is observed that the communication delay in case of proposed architecture is 54% l

Table 6: Comparison of Proposed Architecture with RAW, TriBA, and LBDR

	RAW	TriBA	LBDR
Number of Switches	(+) 62.5	(+) 62.5	(-) 50
Power Consumption	(+)77		

CHAPTER 5

CONCLUSION AND FUTURE WORK

We hope the discussion presented in the thesis motivate interested scholars into considering research in the challenging but prosperous area of multi-core systems. Multi-core architecture is the future of all modern computing areas from server to desktop to embedded environments. With the appropriate architecture, the potential of multi-core systems can be enormous. Our contributions lead to solutions that overcome the disadvantages due to current point-to-point communication and the presence of caches in multi-core. In this chapter, we conclude our work and offer a list of possible future extensions of this work.

5.1 Conclusion

It is proven that multi-core architecture provides better performance/power ratio suitable for real-time applications. However, current multi-core system is not suitable to decrease power consumption and increase memory-level parallelism due to the wasteful core-to-core interconnection topology. For example, each node/core in MIT Raw architecture has computing and switching components. Computing component of such a node consumes power while the node is working (only) as a switching component and vice versa. Moreover, due to the presence of multiple level-1 caches (each core has its own private cache) multi-core architecture suffers from data inconsistency, power consumption, and heat dissipation.

In this paper, we propose a multi-core design methodology to reduce the number of switches without any negative impact on the performance. According to this method, nodes are separated

between computing cores and network switches. However, there are some special nodes (computing/switching nodes) with dual functionalities. Using folded torus concept, we develop an algorithm to determine the computing cores and network switches and how to connect them (cores and switches) in the multi-core architecture. Multi-core architectures with various numbers of nodes (cores and switches) are used to evaluate the proposed methodology. We obtain the core-to-core communication delay and total power consumption for MIT Raw, Triplet Based Architecture (TriBA), Logic-Based Distributed Routing (LBDR), and the proposed architecture using synthetic workload. In addition, we collaborate with other students to develop a simulation platform for multi-core systems.

According to the experimental results, the proposed architecture outperforms Raw, TriBA, and LBDR by cutting down the number of switches significantly. Average delay is decreased due to the fact that each switch provides adequate communication channels. Total power consumption is reduced as the number of switches is cut down. Based on the results, proposed architecture may reduce the total power consumption by up to 77% and average delay by up to 54%. It is also noted that the communication is more reliable in the proposed architecture because each computing core is connected to multiple switches.

5.2 Future Extensions

Our thesis contributions including the design methodology to reduce the number of switches in multi-core architectures can be extended to cope with the following important research areas

- Efficient routing algorithms for multi-core systems: Develop routing tables for the switches and propose efficient routing algorithms for multi-core systems for reliable communication with minimal delay.

- Multi-core modeling and simulation platform support: Modeling and simulation platforms are important to analyze multi-core systems. Proposed methodology can be extended to assist developing and/or evaluating multi-core modeling and simulation platforms.

- Evaluate core allocation strategies in multi-core: Effective core allocation in multi-core architecture may significantly reduce heat intensity of a multi-core chip. Proposed

REFERENCES

[12] Manira S. Rani ³ $Q (IILFLHQW DQG VFDODEOH FRUH DOORF
V\VWHPV´ 7KHVLV LQ 0DVWHUV RI 6FLHQFH)ORULGD $

[13] Jing-Mei Li; Ping Jiao; Chao-Guang Men; ³7KH +HWHURJHQHRXV -DUFKLW
&RUH UHVHDUFK´ MASS '09. International Conference on Management and
Service Science , 2009. doi: 10.1109/ICMSS.2009.5302477 Publication Year: 2009,
page(s): 1±6. 2009.

[14] McNairy C, Bhatia 5 0RQWHFLWR Re, Dual-Core Itanium
3URFHVVRU´ > - @ , ((2005 F U R

[15] Intel® Multi-Core Processor 2011.
www.intel.com/software/enterprise, (accessed in September, 2011).

[16] ,QWHO & RUH Œ 0 2011. R DUFKLWHFWXUH
www.intel.com/Multi-Core, (accessed in September, 2011).

[17] %U\DQ 2 6XOOLYDQ 'RQ 6WHZDUW DQG 2011. K Q *RHU]HQ
http://book.realworldhaskell.org/read/concurrent-and-multicore-programming.html.

[18] ³$Q (IIHFWLYH $SSURDFK- FRUH 0XOWL EDVHG WR ³<X[LQ :
Yuan; He Guo; Xinzhong Hui; Yuansheng Yang; Scalable Computing and
Communications; Eighth International Conference on Embedded Computing, 2009.
SCALCOM-EMBEDDEDCOM'09, page(s): 37±41. doi: 10.1109/EmbeddedCom-
ScalCom.2009.17 2009. T J

[19] Phi-Hung Pham; Phuong Mau; Chulwoo Kim; ³A 64-PE Folded-Torus Intra-chip
Communication Fabric for Guaranteed Throughput in Network-on-Chip Based
$SSOLFDWLRQV ³ &XVWRP ,QWHJUDWHG &LUFXLWV &RQ
doi: 10.1109/CICC.2009.5280748, Page(s): 645±648. 2009. 1

[20] . & &KDQJ - 6 6KHQ DQG 7) &KHQ´ &RQWHQWLRQ D
Circuit-Switched and Packet-Switched NOCs for Application-6 SHFLILF 62&´ '² B• rh—

[24] Raw Architecture Workstation. 2011. http://groups.csail.mit.edu/cag/raw/purpose.html (accessed in April, 2011)

[25] 0 L F K D H O % 7 D \ O R U : D O W H U / H H H W D O ³7 L O H G 0 X O W
Science+Business Media, LLC. doi: 10.1007/978-1-4419-0263-4, pp. 1-34. 2009.

[26] Michael Bedford Taylor $ % ³'HVLJQ 'HFLVLRQV LQ WKH ,PSOHP
$UFKLWHFWXUH :RUNVWDWLRQ´ 'DUWPRXWK &ROOHJH
Institute Of Technology. 2011.

[27] Haroon Ur- 5DVKLG 6KL)HQJ -L :HL [LQ -Core Interconnect 7ransaction HW %DVHG
1HWZRUN DQG LWV &RPSXWDWLRQDO (IILFLHQF\´ &RPS
ICIS 2009. Eighth IEEE/ACIS International Conference on, doi: 10.1109/ICIS.2009.137
page(s): 516-521. 2009.

[28] $E X $VDGX]]DPDQ HW DO -Power Characterization of Multi-Threaded Workloads
Homogeneous Multi& RUH 3URFHVVRUV´ ,(((QG ,QWHUQDWLRQDO
Microelectronics (ICM'10), page(s): 387-390. 2010.

[29] $E X $VDGX]]DPDQ HW DO ³0RGHOLQJ 0XOWLFRUH 'LVW
Performance, Power, and UHGLFWDEOLW\ XVLQJ 9LVXDO 6LP´ +XQ
Conference (HSC 2008) sponsored by SCS and hosted by AMSC, Huntsville, Alabama,
USA. 2008.

[30] 6WHSKHQ : .HFNOHU .XQOH 2OXNRWXQ DQG + 3HWHU
6\VWHPV´ 6SULQJHU 6FLHQFH+Business Media, LLC. 2009.
DOI: 10.1007/978-1-4419-0263-4