NOVA Interconnect for Dynamically Reconfigurable NoC Systems

Fernando Martinez Vallina, Nathan Jachimiec, and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois 60616

Abstract

Network on a Chip (NoC) topologies for interconnect offer the possibility of increasing the flexibility and performance of an embedded computational platform. NoC is also an enabling factor for the creation of dynamically reconfigurable single chip systems. In this paper, the NOVA interconnect topology is shown. NOVA is a hybrid topology for NoC interconnect targeted at an FPGA. This topology is able to efficiently support the communication workloads associated with multiprocessor, multi peripheral and reconfigurable systems. This paper presents simulation results on the performance and scalability of NOVA. Also, the performance of NOVA is compared to the star, torus, and hypercube topologies, and is shown to outperform these topologies.

1. Introduction

As a result of advancements in device integration and miniaturization, System on a Chip (SoC) designs contain an increasing number of processors and peripherals. This increase in the total number of possible system components has led to the creation of more complex SoCs with higher internal communication demands. One approach to reduce the impact of communication latency on performance has been the use of tiered buses and point-to-point links instead of a single communication bus.

The single bus is the easiest communication structure from both a conceptual and an implementation standpoint. However, this structure is hindered by its needs of arbitration to guarantee data integrity. The cost of arbitration on the worst case and average case data transfer times increases with the connection of more devices. Therefore, the performance of a bus degrades proportionally to the number of devices needing to use this communication medium.

Two approaches have been developed to improve upon the performance of the bus. These approaches are tiered busing and point-to-point links [1]. Although both approaches have better performance than the single bus, they still have the same problem in terms of scalability.

The tiered bus approach is the division of a single bus into N individual segments. In this structure, each segment handles communication over the shared medium independently of all other segments [1][2]. From an implementation point, each segment is a mini-bus. Communication between the mini-buses is accomplished through the use of bridging circuits which synchronize the data flow between buses. One of the drawbacks to this approach is the bridging delay cost. Also, the performance of each mini-bus will degrade as more devices are added. The only way to keep the mini-bus performance from degrading is to add more bus levels. This will result in the use of more bridging circuits and more hardware resources devoted entirely to communication.

2. Point-to-Point Links

The point-to-point link approach is the highest performance communication structure for a limited number of devices. One of the characteristics, which make this structure popular in SoC design, is its lack of arbitration. Data transfers happen as fast as the communication medium can handle and the maximum distance between nodes is fixed at 1. The lack of arbitration means that data can be consumed almost as soon as it is produced. Therefore, the data transfer rates between 2 nodes reaches a platform dependent maximum with this structure.

The main problem with point-to-point links is scalability. As devices are added to the system, more links are required to insure that all necessary communications paths exits. This has a negative impact on both the platform resources and on the individual computational nodes. At the computational node level, the addition of more nodes means that more communication links need to be monitored. The monitoring of communication links is done as an interrupt priority, round robin, or polling scheme [3]. Any of these monitoring techniques requires the computational node to devote more resources towards communication and to take them away from doing useful work for the user.

1-4244-0941-1/07/$25.00©2007 IEEE

Useful work is defined as all work required by user applications. Also, more platform resources have to be used in creating the point-to-point links. These resources could have been used to increase the computational power of the nodes in the system.

An alternative approach for SoC communication is the Network on a Chip (NoC) design paradigm [4]. NoC topologies can handle the communication loads of large systems and exhibit the scalability not found in other on-chip communication schemes.

3. NOVA

In this paper, the NOVA topology for NoC design is presented. NOVA is an augmented star communication topology, which is described in section 3. Like all NoC topologies, this topology uses a different communication model with respect to bus based topologies. The communication mode of an NoC is based on the communication methodology of computer networks. Using this data transfer paradigm, NoCs implement a minimal communication protocol to correctly route data. The actual routing protocol is implementation specific and dependent on the underlying interconnect topology. In this regard, NOVA is similar to other NoC approaches. The communication stack used in NOVA consists of only a destination address and a few bits of auxiliary information attached to each data packet. With this information, NOVA maintains a low routing overhead while at the same time allowing for upward scalability to *M* number of devices. The packet format used in NOVA is shown in Figure 1. It can be seen from this figure, that the packet consists of an 8-bit header and a 24-bit payload. The 8-bit headed is used to identify both the target cluster and the target tile. This packet size was chosen because it can traverse the 32 bit network links within a single cycle.

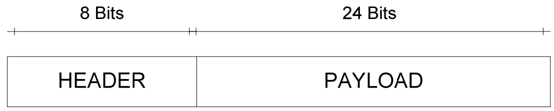

Figure 1. NOVA Packet Structure

Besides providing a communication framework for SoCs, NOVA has been designed to allow for system runtime reconfiguration. Dynamic reconfiguration of an SoC during execution is an enabling technology for the development of single chip polymorphic computing systems. In the embedded space, a single chip polymorphic system is desirable due its ability to execute many user applications within a reduced power budget. The computational flexibility and low power consumption are a result of reconfiguration, which allows for the instantiation of hardware only when it is needed.

Currently, NOVA is the interconnect topology for an FPGA based experimental polymorphic computing system called the Dynamically Reconfigurable Modular Processing Engine (DRMPE). To aid in the presentation of NOVA, a brief summary of the DRMPE is presented in the next section. After this section, the performance and scalability of NOVA is discussed in section 4. Simulation results for NOVA latency are also examined in section 4 for 8, 32, and 64 computational blocks. Furthermore, the performance of this proposed topology is compared to well known interconnects used in NoC and parallel computing platforms.

4. DRMPE Architecture

The DRMPE (shown in Figure 2) is a single chip polymorphic computing platform. This is a multiprocessor architecture with a reconfigurable acceleration fabric defined as a group of tiles, which can work individually or cooperatively on a task. The two processors used to control the DRMPE are the user and a coordinator processor. The coordinator processor is responsible for managing the network of processing tiles, while the user processor carries out user applications. The processing tile network creates a Network on a Chip (NoC) [5]., which extends the SoC by instantiating a series of processing modules that may communicate with one another and with the two microprocessors. All tiles in the network are of fixed size and occupy a rectangular region of FPGA resources. At the fundamental level, each tile is constructed from a number of “reconfigurable frames” as defined by the FPGA on-chip hardware reconfiguration capabilities [6][7].

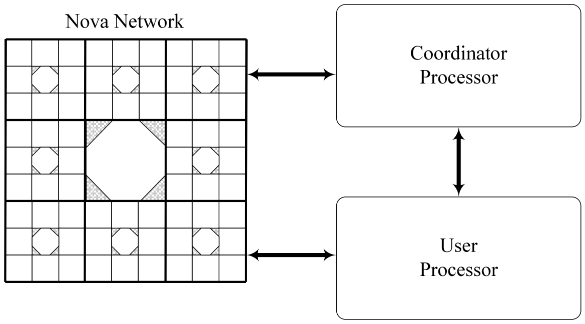

Figure 2. DRMPE Architecture Overview

A function tile is defined as several frames of FPGA resources. An FPGA frame is defined as the smallest reconfigurable region of the fabric. For example, in the Virtex^4 and Virtex^5 families, a column of logic is divided into several segments based on the device size [6][7]. Each segment is called a frame which can be reconfigured. All frames are addressable via hardware configuration and can be redefined without halting the system. The actual micro-architecture of the tile is based upon the underlying FPGA fabric. Since each tile is constructed from the FPGA fabric, it is a reconfigurable module and can take any number of possible functions. The tile is orientated such that several frames form a square, with three fixed communication ports. These ports allow bi-directional communication to and from the neighboring tiles and are constructed as FIFO links. From the point of view of the tile, the central switch node is just another tile in the fabric. Each tile is aware of its local neighbors and can send data to its neighbors via its two tile connections. If the destination tile is not one of those neighbors then it will send the packet to the central switch module that routes the packet to its correct destination. Each tile is also aware of its one and two-hop neighbors. This extra added level of network knowledge at the tiles, allows for the tile to avoid using the switch if it can send a packet to a neighboring tile and have that tile forward it to the true destination tile. This behavior allows for spreading out packet traffic within the tile cluster. As a result, the central switching node is no longer a communication bottleneck and dynamic fault tolerance is built into the macro-architecture.

5. NOVA Interconnect

The NOVA interconnect is the NoC topology developed for the DRMPE. This topology was developed after studying the problem of interconnect latency in existing network topologies. These topologies were evaluated to find the strengths and weaknesses of each one. The topologies tested include classical topologies such as the hypercube and torus, which are common in high performance computing clusters. Bus based topologies were not tested, because the poor scalability of this interconnect is well documented in literature [1][2][3]. From the experimental simulation work and genetic algorithm extrapolation, the basic star topology was evolved into an augmented star topology called NOVA (shown in Figure 3).

At the most basic level, NOVA is a cluster of eight tiles arranged around a dedicated switch module. The routing scheme used by the tiles allows multipath communication using either neighbor tiles or the central switch node. Using information on the location of the one and two hop neighbor, each tile can distribute traffic along different paths and reduce overall network latency. This creates a built-in system for data transfer congestion avoidance and management.

To take advantage of design regularity, the cluster switch and the central switch are the same. Both of the switch types are constructed using a Banyan type switch (see Figure 4) [8]. This type of switching structure was chosen because of its minimal latency for handling packet traffic. Also, the Banyan switch is a self routing topology which does not require lookup tables or an external control finite state machine. Therefore, the Banyan structure is ideal for replication in tile based architectures such as NOVA.

As mentioned before all intra-cluster communications can be carried out through neighbor tiles or through the tile switch. For inter-cluster communication, design space exploration showed that adequate performance was obtained by keeping all transfers through the switches. Variations including direct neighbor inter-cluster links were also examined. The results on these variations showed that a minimal performance gain was obtained at the cost of a more complex tile switch and a higher FPGA resource consumption. The performance gained through inter-cluster direct links in comparison with switch only data transfers did not justify the added complexity and implementation costs.

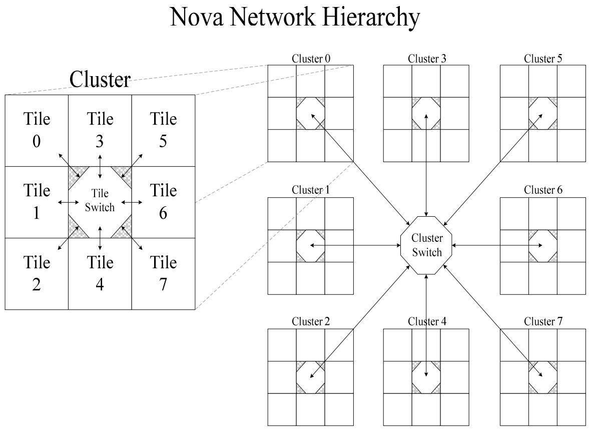

Figure 3. NOVA Interconnect

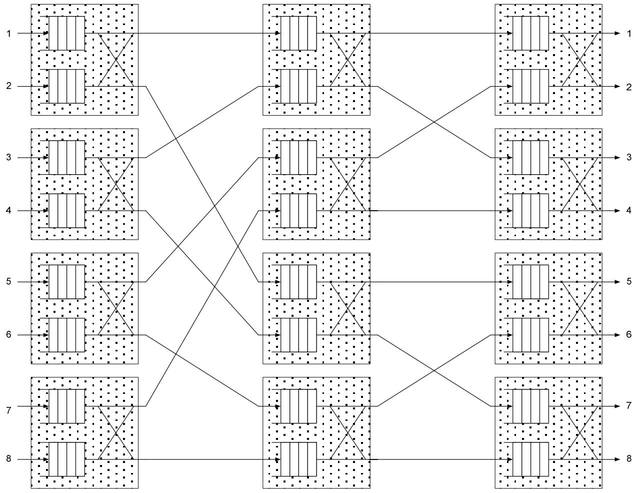

Figure 4. Central Switch Structure

6. NOVA Performance Evaluation

The NOVA interconnect topology was evaluated against well known topologies during its design. All topologies were evaluated on the premise of 8 communicating elements in terms of End-to-End (ETE) packet latency. This base case allowed for all topologies to be simulated with the possibility of physical implementation on an FPGA. The classical topologies used in the design and evaluation of NOVA are the single star, the dual star, the torus, and the hypercube topologies [9][10]. The results of all topology simulations are shown in Fig. 4. These simulations are carried out on an event driven simulator in units of time Δ.

From the theoretical standpoint, the traffic pattern is based on a random seed which guarantees stressing all topologies without biasing towards any particular topology.

Out of all the topologies evaluated, only the star based topologies can be efficiently implemented on an FPGA for large number of communicating elements. For FPGA implementation, both the torus and the hypercube are difficult to implement as a result of the dimensionality built into the topology. In the case of the torus, the wraparound links from edge to edge require the use of long wires within the FPGA. As the size of the torus increases, these wires get longer and the data transfer speeds on all links stops being equal. Also, implementing wraparound links in both the *x* and the *y* planes is resource intensive for FPGA architectures. In the case of the hypercube, the problem arises from the three dimensional nature of the topology. Like in the case of the torus, the creation of links between the sides of the cube involves long wires with uneven delay. These long wires will lead to unequal data transfer rates between the links. In an FPGA platform, the implementation of large torus or hypercube interconnects requires a lot of resources devoted only to communication. Also, these systems will have to be executed at lower frequencies to equalize the data flight time on all possible paths [9][10].

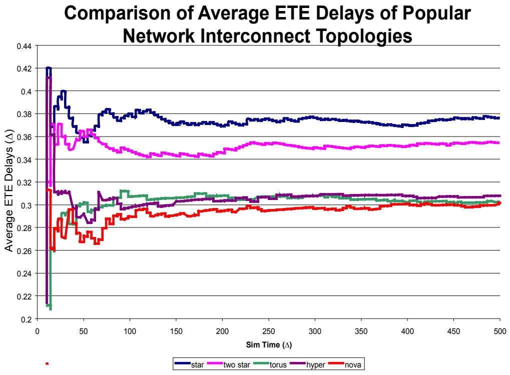

Figure 5. Topology Comparisons

It can be seen in Figure 5 that as expected the best performing classical topologies are the torus and the hypercube. Also, this figure shows that at steady state (network simulation time > 400Δ) NOVA matches the performance of the torus and hypercube. This means that the augmented star topology developed for FPGA implementation performs as well as the recognized best performing topologies in parallel computing. The results of Fig. 4 support the feasibility of NOVA as a high performance communication topology for NoC based systems. Also, these results validate the design decisions taken in NOVA.

The second part of the topology evaluation was to examine scalability as the number of communicating tiles increases. Using the 8 tile case of Figure 5 as a basis, NOVA was tested for both 32 and 64 tile cases. Unlike the 8 tile case, these two cases have to take into the account the central cluster switch. The impact of the cluster switch was studies by varying the intra-cluster packet traffic intensity. The cases tested were a 10%, 50% and 90% inter-cluster packet traffic intensity.

The results of the 32 tile case are shown in Figure 5. This figure shows that as expected the overall latency increases as the number of tiles increases. The interesting result is on the impact of the cluster switch on overall system latency. For the 10% case, the simulation results show that the 32 tile NOVA topology outperforms the 8 node star topologies shown in Figure 5. Also, as the amount of traffic going to the central cluster increases, the overall latency increases as a result of the loading effect on the central switch. In most operating conditions, the inter-cluster traffic intensity in NOVA should not exceed more than 50%. Inter-clusters traffic intensities over 50% show that the tasks were not properly mapped onto the NOVA network. Properly mapped tasks will have a runtime behavior showing a less than 50% inter-cluster communication since a majority of the work associated with a task will be mapped into a cluster.

A comparison of Figures 6 and 7 shows the scalability of the platform. The total latency increases by a linear factor while the computation power of the system is doubled. Also the system behavior under different inter-cluster intensities correlates to the behavior observed in the 32 tile case. These simulation results show the scalability of NOVA.

7. Conclusion

The NOVA topology for NoC systems is a scalable solution for varying numbers of computation tiles. Also the structured pattern of this interconnect allows for system runtime reconfiguration. The simulation results show that NOVA successfully incorporates the performance characteristics of the torus and hypercube in an FPGA centric framework.

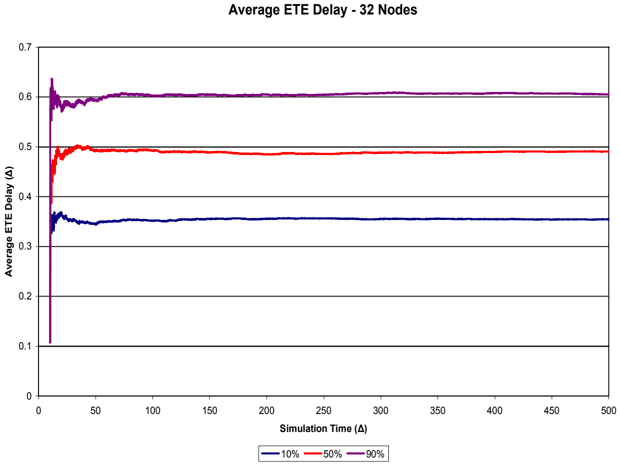

Figure 6. NOVA Performance with 32 Tiles

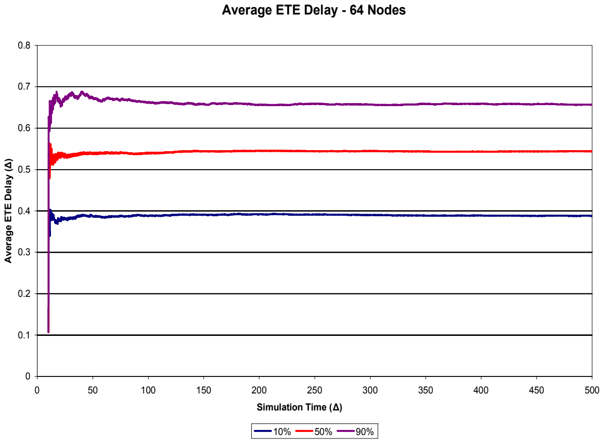

Figure 7. NOVA Performance with 64 Tiles

8. References

[1] K. Lahiri, A. Raghunathan and S. Dey, "Design space exploration for optimizing on-chip communication architectures", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, June 2004.

[2] C. Shin, Y.T. Kim, E-Y Chung, K-M Choi, J-T Kong, S-K Eo, "Fast exploration of parameterized bus architecture for communication-centric SoC design," *Design, Automation and Test in Europe*, 2004.

[3] T. Seceleanu, V. Leppanen, J. Suomi, O. Nevalainen, "Resource Allocation Methodology for the Segmented Bus Platform," *IEEE International SoC Conference*, Sept. 2005.

[4] R. Saleh,, "An approach that will NoC your SoCs off!", *Journal of Design & Test of Computers*, IEEE vol. 22, issue 5, pp. 488, Sept.-Oct. 2005.

[5] M. Forsell,, "A Scalable High-Performance Computing Solution for Networks on Chips", *IEEE Micro*, vol. 22, issue 5, pp. 46-55, Sept/Oct 2005.

[6] Xilinx, "Virtex-4 Configuration Guide", UG071, v1.4, Xilinx Corp., Jan. 2006.

[7] Xilinx, "Virtex-5 Configuration Guide", UG191, v1.2, Xilinx Corp. July 2006.

[8] A. Huang and S. Knaurer, "Starlite: A Wideband Digital Switch," *Proc. Globecom '84*, pp. 121-125, 1984.

[9] D.A. Kearney, G. Veldman,, "Evaluation of Network Topologies for a Runtime Re-Routable Network on a Programmable Chip", *Proceedings of Field-Programmable Technology (FPT)*, pp. 178-185, Dec. 2003.

[10] N. Bansal, S. Gupta,, N. Dutt, and A. Nicolau, "Network Topology Exploration of Mesh-Based Coarse-Grain Reconfigurable Architectures", *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, pp. 474-479, Feb. 2004.