

# Reconfigurable Network on Chip Router for Image Processing based Multiprocessor Applications

Jonathan Joshi  
University of Southern  
California

Om Prakash Vyas  
Jodhpur Institute Of Engineering  
Technology

Sanjay Gaur  
Jodhpur Institute Of Engineering  
Technology

## ABSTRACT

Real time Image processing (I.P.) systems, involving on board multiprocessor communication, use standard bus based communication. The load on the system to deliver the output towards real time standards call for high speeds, but for data intensive application such as IP algorithms require constant transfer of data between the logic cores. This would need either dedicated connections or additional bus controllers. Networks-On-Chip (NoC) provide a structured way of realizing interconnections on silicon, and obviate the limitations of a bus-based solution. This paper deals with the design and implementation of a NoC router targeted for an Image processing system consisting of different modules. All the cores have been designed targeting real time frame rates. The design has been prototyped on a Virtex II FPGA. The timings are given in comparison to a standard DMA controller

**Index Terms:** NoC, Virtex II, DMA, Router

## 1. INTRODUCTION

Modern Systems contain multiple processors, dedicated hardware processing units and peripherals. Such a distributed architecture is required for reasons of performance and energy-efficiency, but it also introduces the requirement of an efficient system-level communication. As technology advances with ever increasing processor speed, global wires spanning across significant portion of board size will dominate the propagation delay [1], which becomes a performance bottleneck for systems design. In recent years, significant research has demonstrated that an on-chip packet interconnection network is a better candidate for handling on chip communication [2]. System modules communicate to one another by sending packets across the network. This approach has the advantages of both performance and modularity. In another example [3], researchers implemented such a reconfigurable interconnection network on FPGA for improved hardware-software multitasking.

The system level components include, besides the on-chip network, also embedded software. Some communication networks that target general-purpose multi processors are the J-Machine [4] and Smart Memory [5]. However, very little research has been done on modeling the on-chip communication architecture and integrating the communication network with processor units in a single environment. Architectural exploration of a network should be done in the early stages of the design, using system-level simulation. This exploration is required because the communication requirements of a system are often determined by the target application.

Our application presented here is the first step towards the implementation of the different components of a NoC. The router presented here is designed such that the FPGA based solution can alleviate the challenges faced by bus controllers when it comes to handling high speed communication. A

comparative study of a system based on a bus controller and the NoC based on our designed router is presented.

## 2. FPGA AS IMPLEMENTATION TECHNOLOGY

Over 10 years ago, Xilinx Corporation introduced the first generation of Field Programmable Gate Arrays, or FPGAs [6]. These chips were designed to allow hardware manufacturers to include simple control logic in their products without having to resort to custom circuits. Essentially, the technology allows engineers to use software tools to specify hardware circuits. Although the technology was originally developed as an alternative to PALs and used for glue logic, there were early visionaries who perceived that the potential for FPGA technology was much greater than that. Even in the early stages of FPGA development, Papers were published that suggested this technology could be used for complex applications such as imaging. The key to FPGA technology is that it is reconfigurable. At any time, new software can be loaded into the chip that completely changes its character and function. Although the original FPGAs were relatively simple devices, this class of chip has grown in size and complexity to the point that today, complex algorithms can be implemented using FPGAs. The programming tools for these products, however, have not advanced to the same level as other, more mature technologies. As a result, creating software to run in an FPGA environment requires a high level of skill. Developers create schematics or a Hardware Description Language (HDL) representation of a design. The design is then compiled into a bit-stream which is loaded into the chip, rather than building a physical circuit. Advancements in FPGA technology have allowed it to become a viable alternative to other general purpose and specialized processors. FPGA represents the next step in computer design and control. For real-time computation, FPGA technology provided even more specialization and power. FPGAs continue to advance this process. For many applications, the use of FPGAs offers a faster, less expensive solution that is easier to upgrade as technology continues to move forward. In addition to higher speed and lower costs, the implementation of an FPGA solution requires fewer chips on a board. This allows a smaller footprint to be achieved as well as creating a highly customizable product. In addition, FPGA based system can be upgraded in the field by simply sending new code to run on the chips.

## 3. RELATED WORK

In [2], Benini and De Micheli present network on chip (NoC) as a new paradigm for SOC design based on an approach similar to the micro-network stack model [7]. They discuss the design problems and possible solutions for each level of the stack from the application level to the physical level through the topology and protocol levels. The standard solution of the topology

selection problem is the use of a single bus, but this may turn out quite inefficient from a power consumption viewpoint. Instead, [2] suggests using packet-switching architectures. They focus on providing some examples of known topologies and do not discuss the problem of selecting an optimum one. A methodology centered on the simulation of traces is proposed in [8]: the resulting communication architecture is an interconnection of well-characterized communication structures similar to buses. Finally, in [9] the interconnection structure between computation blocks is fixed (a grid) and predictable. Information is routed in the communication network by means of dedicated switches. Constraint-driven communication synthesis (CDCS) [10] follows an approach that is inherently different from the previous ones because it aims to derive communication architecture as the union of heterogeneous sub networks that together satisfy the original communication constraints given by the designer.

#### 4. NETWORK STRUCTURE

Each processor core is connected to a dedicated router for communication into and out of the network as illustrated in Figure 1. These routers are addressable for communication among processors. The network uses a deterministic routing algorithm in the form of a lookup table inside each router for routing to the neighboring node. Although flow control is not supported, a deterministic routing approach significantly reduces hardware complexity and overhead. Reconfiguration of the network topology and placement of processing units only requires a modification to the routing table. Designers can arbitrarily instantiate multiple 1D- or 2D-routers library block to build a dedicated network. Furthermore, they can reconfigure internal buffer size of each router, and in this way, trade area for speed. These two features allow creation of a network topology that is matched to the traffic patterns of a special purpose system. It also allows for more efficient use of routing resources, which is an important design factor in system design.

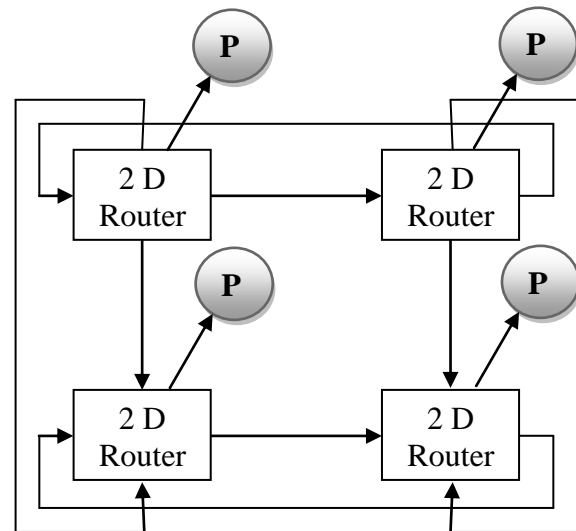
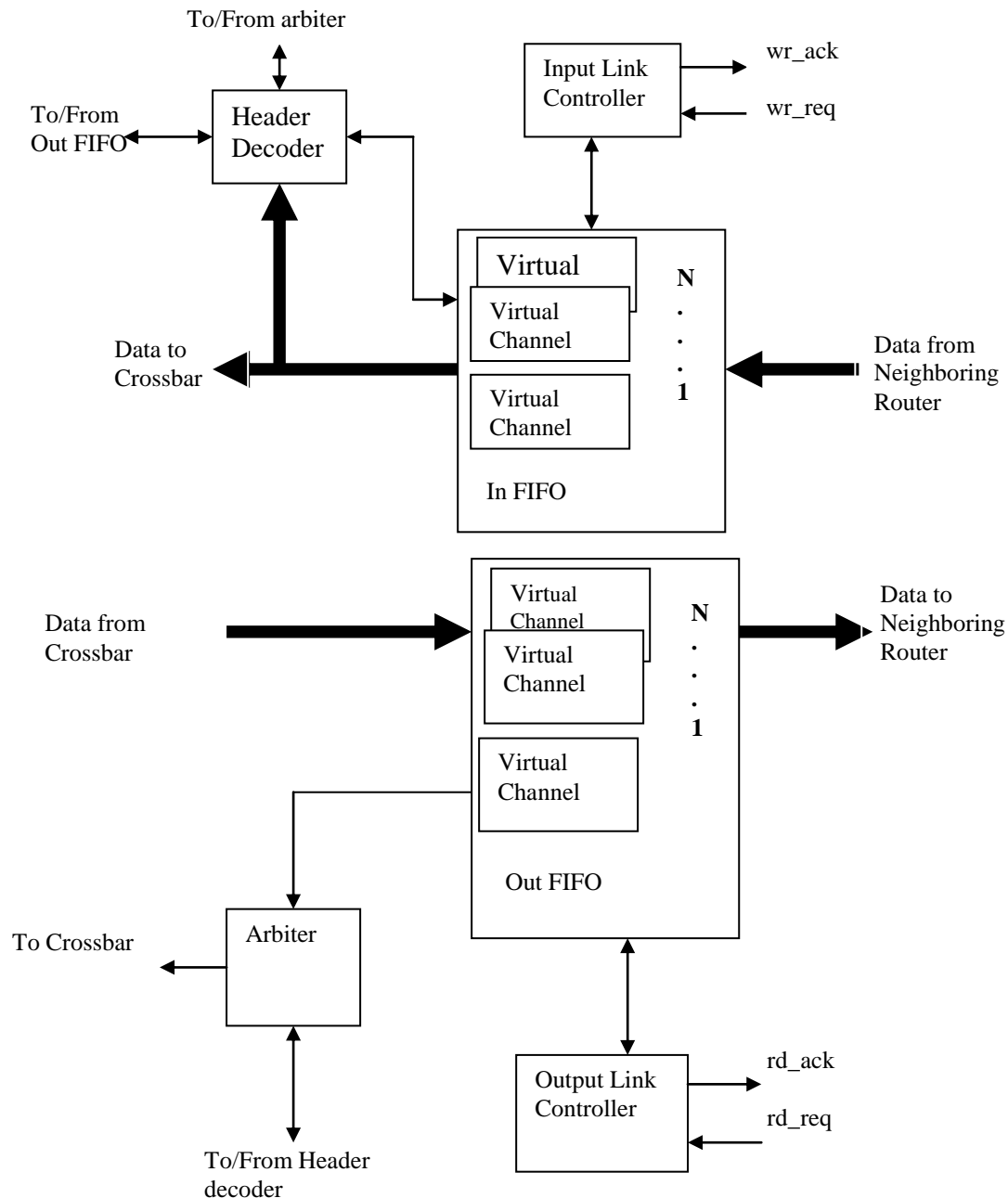


Figure 1. System Architecture

#### 5. ROUTER INTERFACES AND PACKET FORMAT

The 2D router shown in Figure 2 has data flowing in two directions. Each router has three input interfaces and three output interfaces dealing with synchronized communication between routers and the network interaction with processors. The communication reliability is guaranteed through a two-way handshake for each packet transmission. Each router performs wormhole routing. The transmission does not make any assumption on maximum message size or on the message data type as long as the proper packet format is abided. The first 2 bits of each packet contain control information indicating a header packet, a tail packet or a normal packet. The header packet will contain additional information on destination port.



**Figure 2. Router Architecture**

### 5.1 Router Architecture

As illustrated in Figure 2, the router contains three concurrent controllers: an input controller, a router output controller and the arbiter. The input controller handles simultaneous input requests from neighboring routers and the processors. Priority is given to router inputs because the processor interfaces are driven by software, which is typically slower. A round-robin scheme is employed to arbitrate requests of equal priority. The router output controller and two virtual channels handle communication to neighboring routers. The two virtual channels can avoid deadlocks in a two dimensional torus network topology [11].

Finally, the arbiter interfaces with the processor core to receive packets from the network. Because the communication between network and processor is handled in a blocking-send and receive manner, an additional output buffer is added between the routing channel and the processor output to relieve possible congestion caused by the blocking. A 1-D router has a similar structure but with a reduced interface and reduced number of virtual channels. A routing table is used to determine the subsequent routing path of each packet.

## 6. IMPLEMENTATION RESULTS

The architecture was prototyped on a Virtex II FPGA, the hardware occupancy of the system in terms of FPGA slices has been provided in table 1. Another system based on a DMA bus controller had been implemented. The three ports of the NoC had processors and the fourth had a shared memory. The reduction in timing shows the advantage of using a NoC based system for multiprocessor communication. The processors have been exchanging eight and sixteen bit data with the shared memory. The timings have been registered for different amounts of data in table 2.

**Table 1. Hardware Occupancy**

Module	No. of Slices
Network Interface	120
Router	344

**Table 2. Comparative Timing Results**

System	Data			
	8 Bit		16 Bit	
	10 Bytes	1Kb	10 Bytes	1Kb
NoC Based	10.23 $\mu$ S	4.56 mS	17.65 $\mu$ S	9.45 mS
DMA Based	25 $\mu$ S	100 mS	45.3 $\mu$ S	214.6 mS

## 7. CONCLUSION

An FPGA based Network on Chip system is presented. The design focus has been on the router design with further work to be done on the network interface architecture. The timing results have been documented for the NoC and the DMA based systems respectively. The results show the advantage of using the NoC based system over standard bus controllers whilst using heavy amounts of data. Further research can be done on the improvising the router latency by using parallel processing techniques but maintaining the hardware occupancy optimization. We are currently developing a methodology to explore and select different on-chip reconfigurable network architectures. The performance can also be improved by selecting the routing algorithm that best suits the application under observation. An attempt is also being made to design an application specific architecture suitable for signal processing applications.

## 8. REFERENCES

- [1] S. Charles, "Let's Route Packets Instead of Wires," Proc. 6th MIT Conf. 1990, Advance Research in VLSI, pp. 133-138.
- [2] L. Benini and G. Micheli, "Networks on Chips: A New SoC Paradigm," IEEE Computer 35(1) 2002, pp. 70-78.
- [3] T. Marescaux, A. Bartic, D. Verkest, S. Vernalde, R. Lauwereins, "Interconnection Networks Enable Fine-Grain Dynamic Multi-Tasking on FPGAs," FPL, Sep, 2002.
- [4] W. Dally, "The J-Machine Network" Proc Intl Conf on Computer Design. IEEE VLSI in Computer & Processor, Oct 1992, pp 420-423.
- [5] K. Mai, "Smart Memories: A Modular Reconfigurable Architecture," Proc ISCA, June 2000, pp. 161-71.
- [6] Trimmerger S. M, "Field – Programmable Gate Array Technology", Kluwer Academic Publishers, 1995.
- [7] J. Walrand and P. Varaija. "High Performance Communication Networks". Morgan Kaufmann, San Francisco, 2000.
- [8] K. Lahiri, A. Raghunathan, and S. Dey. "Efficient exploration of the soc communication architecture design space". In Proc. Intl. Conf. on Computer-Aided Design, pages 424–430, 2000.
- [9] W. J. Dally and B. Towles. "Route packets, not wires". In Proc. of the Design Automation Conf., pages 684–689, 2001.
- [10] A. Pinto, L. P. Carloni, and A. L. Sangiovanni Vincentelli. Constraint-Driven Communication Synthesis. In Proc. of the Design Automation Conf., pages 783–788. IEEE, June 2002.
- [11] W. Dally, "Virtual-Channel Flow Control," IEEE Transaction on Parallel and Distributive System, vol 3, no 2, March 1992.