POWER AND LATENCY ANALYSIS OF MASKED NOC ARCHITECTURE

***Dayakar C. V., **Janarthanan M**
**Assistant Professor, Dept of EIE*
Velammal Engg College, Chennai, India
***Assistant Professor, Dept of ECE*
SRM University, Ramapuram
Chennai, India

ABSTRACT

A Network-on-chip (NoC) is a new paradigm in complex system-on-chip (SoC) designs that provide efficient on chip communication networks. It allows scalable communication and allows decoupling of communication and computation. The data is routed through the networks in terms of packets. The routing of data is mainly done by routers. So the architecture of router must be an efficient one with a lower latency and higher throughput. And Wormhole flow control, also called wormhole switching or wormhole routing is a system of simple flow control in computer networking based on known fixed links. It is a subset of flow control methods called Flit-Buffer Flow Control. In this project we designed, implemented and analyzed crossbar router architectures for a network on chip communication in a FPGA. The routers have five ports, four ports connected to other ports in four different directions and the fifth port connected to the processing element through a network interface. Our Proposed architecture contains 4x4 crossbar switch, switch allocator, path and channel request, data ram and 4 i/o ports.

Keywords— Networkonchip, Latency, Field Programmable Gate Arrays (FPGA).

I. INTRODUCTION

Modern field-programmable gate-arrays (FPGAs) consist of several million logic cells [1], an assortment of specialized hard blocks, such as RAM, multipliers and processor cores, and embedded hard interfaces, such as DDRx memory interfaces and Peripheral component interconnection express (PCIe) transceivers. These capabilities make FPGAs a strong programmable platform for implementing large complex systems; however, it is still difficult to complete FPGA designs. One of the main difficulties in designing for FPGAs is creating the system-level interconnect; currently, this interconnect consists of multiplexer-based soft buses constructed out of the FPGA fabric. It is challenging to create these soft buses that must often connect a hard interface running 10× faster than the FPGA fabric. Because the soft bus is much slower than these interfaces, it must be very wide to transport the incoming data bandwidth. For example, a single 64-bit DDR3 933-MHz interface requires both a 576-bit wide input and a 576-bit output bus running at over 200 MHz, and these buses often span much of the chip. To design that very wide bus for 200 MHz is a challenge that often necessitates multiple design iterations. In addition, these huge buses rapidly consume a large fraction of the FPGA's resources, both area and power.

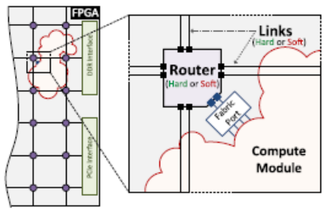


Fig. 1 : Mesh NoC implemented on a FPGA

We propose augmenting the FPGA's conventional interconnect with a high-speed embedded network-on chip (NoC) for the purpose of handling global communication between I/O interfaces, hard blocks, and the FPGA fabric (Fig. 1). The NoC abstraction can simplify design and speed up compilation [2], [3]. Our recent work shows that hard NoCs have compelling area and delay advantages over soft NoCs [2], [4]; however, power is a major concern: does this higher level of interconnect abstraction come at an unacceptable power cost? How do NoCs compare with the multiplexer-based soft buses that are currently used for interconnection? In answering these questions, we investigate both how to design an energy-efficient NoC in the FPGA context and how the power of this NoC compares with that of the conventional fabric.

II. NETWORK ARCHITECTURE

NoCs consist of routers and links. Routers perform distrib-uted buffering, arbitration, and switching to decide how data moves across a chip, and links are the physical connections that carry data between routers.

On modern FPGAs, communication bandwidth demands are high. In particular, FPGAs interface to many high-speed I/Os, such as DDRx, PCIe, gigabit Ethernet, and serial transceivers. To keep up with these high-throughput data streams and move data across the FPGA with low latency, we base our NoCs on a high-performance packet-switched router. This packet-switched router includes a superset of the components that are used in building any NoC. Because we analyze each subcomponent separately, studying this full-featured router yields a more complete analysis of the design space.

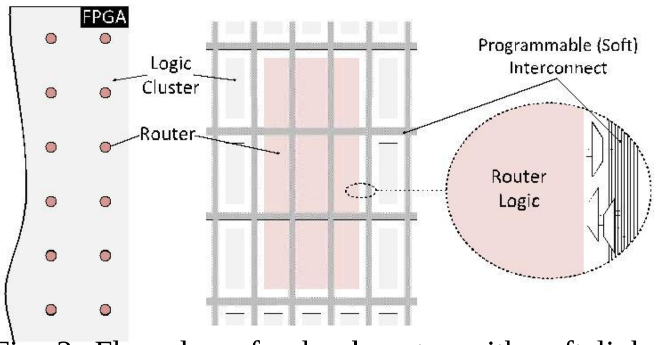


Fig. 2. Floorplan of a hard router with soft links embedded in the FPGA fabric. Drawn to a realistic scale.

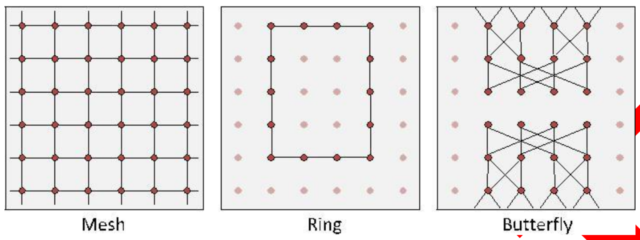


Fig. 3. Examples of different topologies that can be implemented using the soft links in a mixed NoC.

We investigate the design of NoCs on FPGAs; as shown in Fig. 1, both routers and links can be either soft or hard. Soft implementation means configuring the NoC out of the conventional FPGA fabric, whereas hard implementation refers to embedding the NoC as unchangeable logic on the FPGA chip. We compare the power of soft NoCs to that of several possible hard NoCs. Note that, a 64-node version of a hard NoC adds ~1% area to a large FPGA, making it a highly practical addition [2].

III. EXISTING SYSTEM

The input ports buffer input flits and send requests to the allocators. The routing computation module determines the output port based on the routing algorithm. After the route computation, a free output VC (OVC) in the next router is assigned to the input VC (IVC) by sending request to the VC allocator. If an OVC is successfully assigned, then another allocation request will be sent to the switch allocator. The crossbar is then configured to send the desired flit to the output port if the switch allocation request is granted. In order to send requests to the switch allocator, the available space in the next router buffer must be known.

IV. PROPOSED SYSTEM

To reduce the communication latency while maintaining good throughput, a router needs to perform several stages such as route computation, VC allocation, and switch allocation in parallel. In the proposed NoC router architecture, any request which has been granted service by the switch allocator is able to pass a flit to the output port successfully. An efficient masking technique is proposed to filter all switch allocation requests that are not able to pass flits to the output port, either due to the lack of free space in assigned VC or due to the lack of free VC in the output port for non assigned VC requests. Our proposed technique has minimal impact in timing and area overhead of an NoC router. It is also fully parameterizable in terms of number of VCs, buffer width, and flit width.

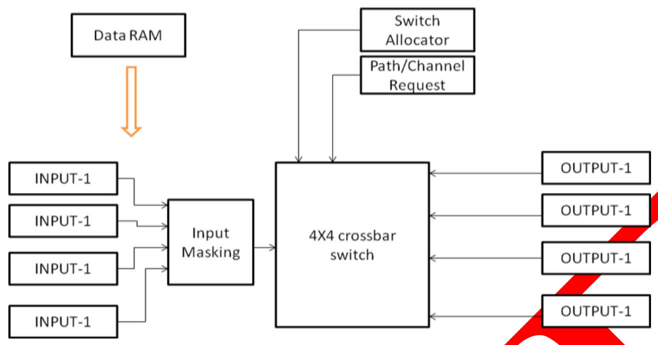
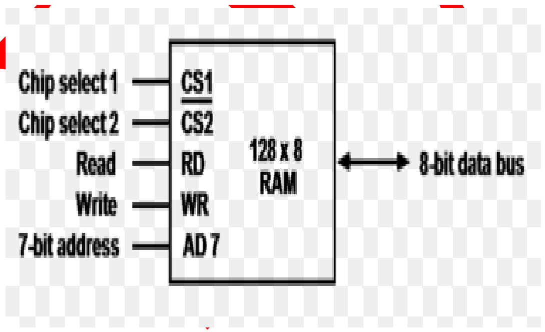


Fig. 2 : Block diagram of the proposed system

V. SYSTEM ARCHITECTURE

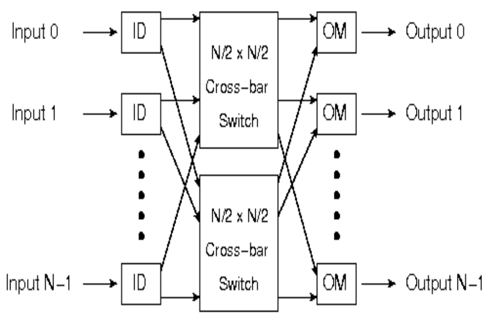
Data RAM

Dataram RAM Disk takes all of this into account: launch the configuration utility, set a size for your RAM disk, then switch to the Load and Save tab – tick both load and save options to ensure the data within your RAM disk is stored on your hard drive, and tick AutoSave for additional protection.



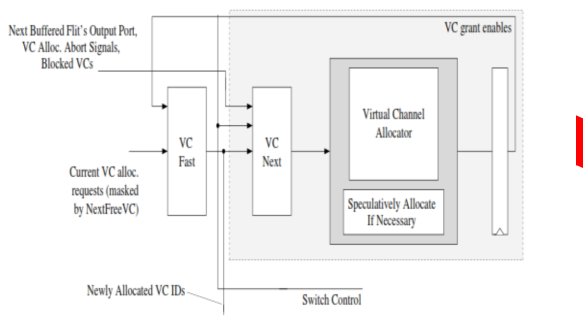
Crossbar Switch

A crossbar switch (cross-point switch, matrix switch) is a collection of switches arranged in a matrix configuration. A crossbar switch has multiple input and output lines that form a crossed pattern of interconnecting lines between which a connection may be established by closing a switch located at each intersection, the elements of the matrix.



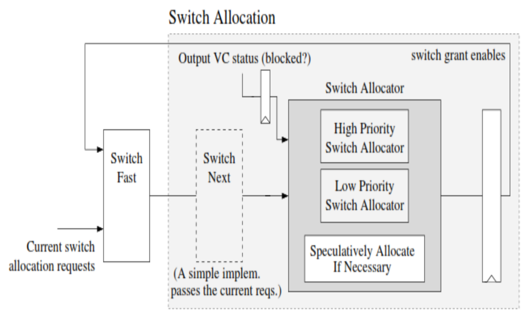
Virtual Channel Router

Virtual channel router (VCR) is a router which uses wormhole network flow control with virtual channels.This router architecture has 5 input and output ports. Four of them are connected to neighbor routers and one is for router's local core. Each input port has 4 virtual channels which are de-multiplexed and buffered in FIFOs. After FIFOs the virtual channels are multiplexed again to a single channel that goes to a crossbar. Routing operations in the crossbar are controlled by an arbitration unit (AU). Arbitration unit also takes care that there are no conflicts between virtual channels and that the arbitration is fair.


Next Buffered Flit's Output Port, VC Alloc. Abort Signals, Blocked VCs
VC grant enables
VC Fast
VC Next
Virtual Channel Allocator
Speculatively Allocate If Necessary
Current VC alloc. requests (masked by NextFreeVC)
Newly Allocated VC IDs
Switch Control

Switch Allocation

Each packet maintains state indicating the availability of buffer space at their assigned output VC. When flits are waiting to be sent, and buffer space is available, an input VC will request access to the necessary output channel via the router's crossbar. On each cycle the switch allocation logic matches these requests to output ports, generating the required crossbar control signals.


Switch Allocation
Output VC status (blocked?)
switch grant enables
Switch Allocator
High Priority Switch Allocator
Low Priority Switch Allocator
Speculatively Allocate If Necessary
Switch Fast
Switch Next
Current switch allocation requests
(A simple implem. passes the current reqs.)

Allocation

After masking the IVC requests, these requests are sent to the switch allocator. Due to having two levels of arbitrations in the switch allocator, arbiter delay is an important parameter in defining the NoC critical path. Hence, to minimize the arbitration delay, fast arbiter proposed.

VC allocation

This stage assigns an empty VC in the neighboring router connected to the output port. Since several header flits may send requests for the same VC, arbitration is required. The routing computation as well as the VC allocation only requires the header flit. The body and tail flits will follow their respective header flit.

Switch allocation

If VC allocation is successful, the third stage sends request to the switch allocator to allocate the output port. Each packet maintains state indicating the availability of buffer space at their assigned output VC. When flits are waiting to be sent, and buffer space is available, an input VC will request access to the necessary output channel via the router's crossbar

Separable input-first allocator (Masking)

A comprehensive analysis on the following allocators shows that separable input-first allocators have the advantage of lower communication delay, area overhead, and power consumption compared to other schemes. Hence, the separable input-first allocator has been chosen to be implemented in our low latency NoC router. A separable input-first allocator consists of two levels of arbitrations.

Routing Algorithm

Routing algorithm determines the output port which a packet must be sent to reach its destination. Deterministic routings act well when dealing with uniform traffic where congestion has been distributed equally across all links in an NoC. However, the nature of NoC traffic is bursts which results in imbalanced distribution of traffic across all links.

Hence, deterministic routing results in poor performance for such traffic. As packets can be sent to multiple ports, a port selection module is required to select the desired output port among them. In the case of look-ahead deterministic routing algorithm, only single output port is selected and it can be directly used in our proposed design.

VI. SOFTWARE IMPLEMENTATION

In the proposed system, the software implementation plays a major role while retrieving the sensor data and updating it to the server.

ModelSim

ModelSim is a multi-language HDL simulation environment by Mentor Graphics, for simulation of hardware description languages such as VHDL, Verilog and System C, and includes a built-in C debugger. ModelSim can be used independently, or in conjunction with Altera Quartus or Xilinx ISE. Simulation is performed by GUI, or automatically using scripts.

Xilinx ISE

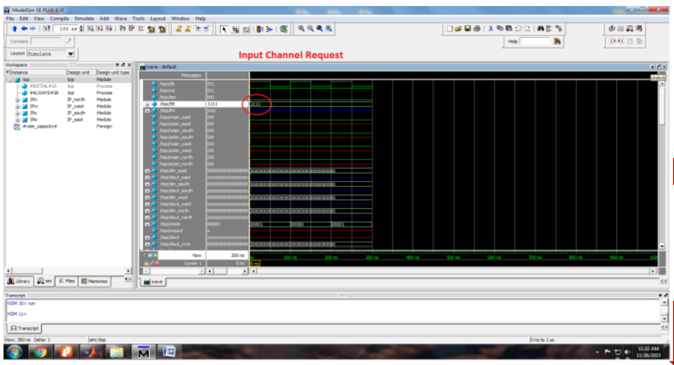
The Xilinx ISE is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors.

ISE Design Suite Highlights

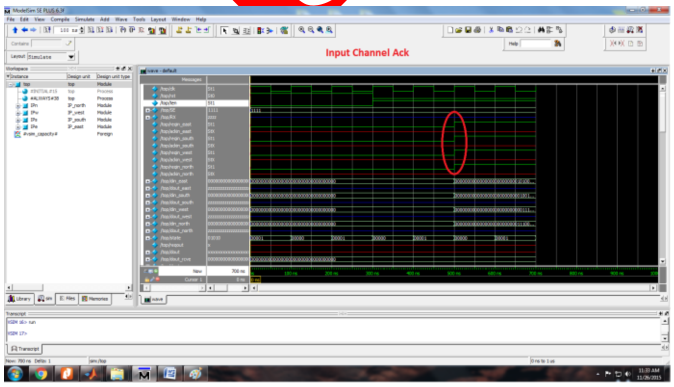
Power

Up to 30% dynamic power reduction using innovative automated clock gating technology. Building on a well-known but often under-utilized power-optimizing design methodology called "clock-gating", ISE 12 introduces the first automated, intelligent clock-gating technology for FPGA designs.

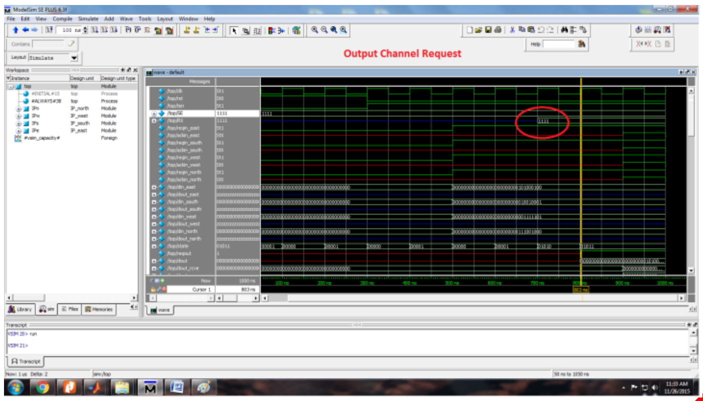
VII. SIMULATION RESULTS



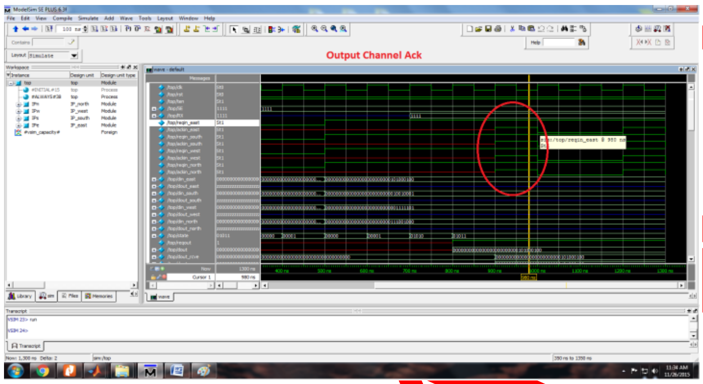
- In the first step in simulation process we select the CLK pulse
- Then we choose the RESET pin to one after we select the TEN to zero
- In the next step we choose the SENDER input
- Then we select the RECEIVER input
- After that we run the simulation in some particular cycle
- Then we start the process from the beginning
- But, in the second time we choose the RESET pin to zero
- After we select the TEN to one
- In the next step RUN the simulation
- The red round in the output refer to the INPUT channel request



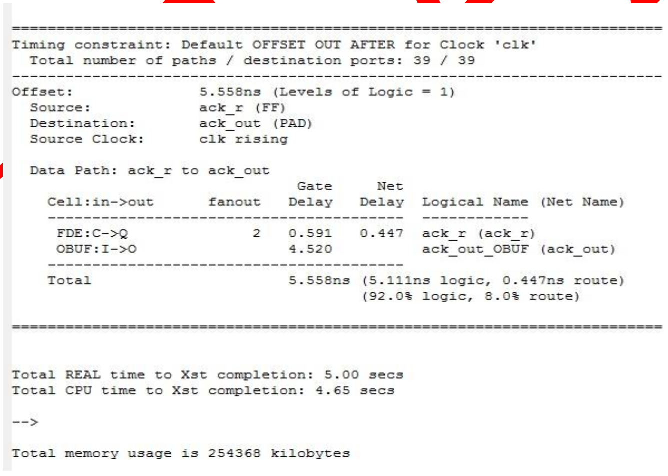
The above image refers to input channel ACK.



The above image refers to output of channel Request.



The above image refer to ACK of output channel.



```
========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 39 / 39
------------------------------------------------------------------------
Offset:              5.558ns (Levels of Logic = 1)
  Source:            ack_r (FF)
  Destination:       ack_out (PAD)
  Source Clock:      clk rising

  Data Path: ack_r to ack_out
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDE:C->Q              2   0.591   0.447  ack_r (ack_r)
     OBUF:I->O                 4.520          ack_out_OBUF (ack_out)
    ----------------------------------------
    Total                      5.558ns (5.111ns logic, 0.447ns route)
                                       (92.0% logic, 8.0% route)

========================================================================


Total REAL time to Xst completion: 5.00 secs
Total CPU time to Xst completion: 4.65 secs

-->

Total memory usage is 254368 kilobytes
```

VIII. CONCLUSION

In this work a Network-on-chip (NoC) which is a new paradigm in complex system-on-chip (SoC) designs that provide efficient on chip communication networks was proposed. It allows scalable communication and allows decoupling of communication and computation. In this project we designed, implemented and analyzed crossbar router architectures for a network on chip communication in a FPGA. The architecture is optimized in five main criteria, which are 4x4 crossbar switch, switch allocator, path and channel request, data ram and 4 I/O ports compared to existing works.

REFERENCES

[1] Xilinx Inc. (2014). " Xilinx Ships First Virtex UltraScale FPGA and Expands Industry's Only 20 nm High-End Family for 500 G on a Single Chip".
[2] M. S. Abdelfattah and V. Betz, "Design tradeoffs for hard and soft FPGA-based networks-on-chip,"
[3] E. S. Chung, J. C. Hoe, and K. Mai, "CoRAM: An in-fabric memoryarchitecture for FPGA-based computing,"
[4] M. S. Abdelfattah and V. Betz, "The case for embedded networks on chip on field-programmable gate arrays," Jan./Feb. 2014.
[5] B. Sethuraman, P. Bhattacharya, J. Khan, and R. Vemuri, "LiPaR: A light-weight parallel router for FPGA-based networks-on-chip," 2005.
[6] M. K. Papamichael and J. C. Hoe, "CONNECT: Re-examining conventional wisdom for designing NoCs in the context of FPGAs," 2012.
[7] Y. Huan and A. DeHon, "FPGA optimized packet-switched NoC using split and merge primitives," 2012.
[8] R. Francis and S. Moore, "Exploring hard and soft networks-on-chip for FPGAs", Dec. 2008,
[9] K. Goossens, M. Bennebroek, J. Y. Hur, and M. A. Wahlah, "Hardwired networks on chip in FPGAs to unify functional and configuration interconnects," 2008,
[10] G. Guindani, C. Reinbrecht, T. Raupp, N. Calazans, and F. Moraes, "NoC power estimation at the RTL abstraction level," ,Apr. 2008,
[11] R. Mullins, "Minimising dynamic power consumption in on-chip networks,"Nov. 2006.