High Throughput NoC System with Low Latency and Interfacing using Wishbone

Priya. K
M.Tech, Dept. of Electronics and Communication
Sir.M Visvesvaraya Institute of Technology ,
INDIA

Krishnapriya Sharma S
Associate.Prof. Dept. of Electronics and communication
Sir.M Visvesvaraya Institute of Technology,
INDIA

***Abstract*— A Network on Chip is an interconnection network in modern on chip design. Network on Chip is flexible and scalable communication architecture for design of System on Chip. In this paper an FPGA based NoC computing system is designed with the aim of reducing the latency, increasing the throughput and decreasing the area overhead when compared to other papers. The NoC system is designed using a routing algorithm by which there is a reduction in area and latency. The NoC is optimized for Xilinx FPGA and the NoC operates at a desired frequency in Spartan FPGA.**

The main goal of this project is reduced latency, high throughput and reduced area overhead. The next goal is designing a standard bus protocol such as Wishbone that can be easily interfaced. The next goal is to maintain a certain amount of flexibility depending on the choice of topology. The wishbone interface is designed such that it is easily compatible with the NoC architecture. In wishbone protocol we are designing a master and slave using cross bar switch interconnection.

Keywords—NoC Router, Routing Algorithm, SoC Buses, Wishbone Interface, Wishbone Bus

I. INTRODUCTION,

Network on Chip (NoC) is a new paradigm for System on Chip (SoC) design .With the increasing integration and growing complexity, the commonly used interconnection techniques for SoC architecture, bus structure, poses practical physical problems. In NoC paradigm, cores are connected to each other through a network of routers and they communicate among themselves through packet-switched communication. The protocols used in NoC are generally simplified versions of general communication protocols used in data networks. This makes it possible to use accepted and mature concepts of communication networks such as routing algorithms, switching techniques, flow and congestion control etc. in Network-on- chip architecture. It allows significant reuse of resources and provides highly scalable and flexible communication infrastructure for SoC design. Data communications between segments of chip are packetized and transferred through the network. The network consists of wires and routers. Processors, memories and other IP-blocks (Intellectual property) are connected to routers. A routing algorithm plays a significant role on network's operational performance.

Network on Chip (NoC) is a new archetype to formulate the interconnections in a System on Chip (SoC) system. With encroachment in NoC technology the bus structure which often results in traffic congestion, is replaced with an integrated network. Today a NoC consists of devices that use the network routers directing traffic among devices and wires analogous to the Internet. Thus efficient network topology and routing algorithm are imperative to the NoC design. There are different routing algorithms available today; the choice of one is often system dependant. This paper presents a modified XY routing algorithm combined with a scheduler to be used on NoCs. The proposed method is a fast way to transferring data via a specific path between two nodes in the network and the scheduler further helps to avoid collision. More latency and fewer throughputs are obtained with contention in network. As mesh size increases latency and throughput increases accordingly

The Wishbone specification defines the Wishbone bus as the System-on-Chip (SoC) architecture which is a handy edge for use with semiconductor IP cores. It is intended to be used as an internal bus for SoC applications with the aim of alleviating SoC integration problems by nurturing design reuse. It can be concluded that a SoC these days has turned into an IC that implements most or all the functions of a complete electronic system. The WISHBONE design was strongly influenced by three factors. First, there was a requirement for a good, reliable System-on-Chip integration solution. Second, there was a need for a common interface specification to facilitate structured design methodologies on large project teams.

.

This paper attempts chiefly to present a review of the WISHBONE bus architecture developed by the SILICORE. The paper endeavours to evaluate the features like the bus topologies, communication protocols, arbitration methods, bus-widths, and types of data transfers in these SoC organized communication architectures.

II BACKGROUND

Networks-on-chip has been a popular research area for some time now. An early paper that discusses the advantages of an ASIC based NoC is [2] when compared to more traditional approaches.

FPGA based NoCs

While the majority of NoC publications are discussing ASIC based NoC, there are some publications that explicitly deal with FPGA based NoCs, an early one is [5] [6] where a packet switched NoC is studied on the Virtex-II Pro FPGA. One of the main goals of this NoC is that it should be able to be used in a dynamically reconfigurable system. A recent example of an FPGA based NoC is described by Kapre, Mehta, et al in [6] in which the authors describe a packet switched NoC running on a Virtex-II and compares it to a statically scheduled NoC on the same FPGA. A circuit switched NoC for FPGAs named PNoC is described by C.Hilton and Nelson in [7].

Another recent example of an FPGA based NoC is No-Cem by Schelle and Grunwald in [8] which is a NoC aimed at multicore processors in an FPGA. The source code for NoCem is also available on the Internet [9].The interested reader can find a survey of some additional FPGA based NoCs in [10]. It also includes comparison with other interconnect architectures such as buses.

The WISHBONE specification document [10] defines the WISHBONE bus as the System-on- Chip (SoC) architecture which is a portable interface for use with semiconductor IP cores. It is intended to be used as an internal bus for SoC applications with the aim of alleviating SoC integration problems by fostering design reuse. This objective is achieved by creating a common interface between IP cores [11]. It improves the portability, reliability of the system, and results in faster time-to-market for the end user [12]. However, the cores can be integrated more quickly International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.2, April 2012 109and easily by the end user if a standard interconnection scheme is adopted. The WISHBONE bus helps the end user to accomplish all these objectives at one platform. Intended as a general purpose interface, WISHBONE bus defines a standard set of signals and bus cycles to be used between IP core modules making no attempt to regulate the application specific functions of the IP core. The Wishbone defines two types of interfaces, called master and slave. Master interfaces are IPs which is capable of initiating bus cycles, while slave interfaces are capable of accepting bus cycles [10]. The hardware implementations support various types of interconnection topologies such as: point-to-point connection used for direct connection of two participants that transfer data according to some handshake protocol dataflow interconnection used in linear systolic array architectures for implementation of DSP algorithms shared bus typical for MPSoCs organized around single system bus
Crossbar switch interconnection usually used in MPSoCs when more than one masters can simultaneously access several different slaves. The master requests a channel on the switch, once this is established; data is transferred in a point-to-point manner.

III RELATED WORK

Network on chip has been a rapidly developed concept in recent years to tackle the crisis with focus on network based communication. Buses and point to point connections were main mean of transmission. But new technologies advances further, problems related to bus appeared the most. First, buses do not scale as the number of communication partners connected become higher. Second, long and global wires and buses become undesirable due to their low and unpredictable performance, high power consumption and noise phenomenon. Third, due to the unpredictability of the communication performance, designing and verifying a large bus based communication networks is very hard. Fourth, every system has a different communication structure, making its reuse difficult [10]. Micro-architectural configurations of buffers in routers have a significant impact on the overall performance of an on-chip network (NoC). This buffering can be at the inputs or the outputs of a router, OBRs(0utput buffer router) are attractive because they have higher throughput and lower queuing delays under high loads than IBRs(input buffer router). However, a direct implementation of OBRs requires a router speedup equal to the number of ports, making such a design prohibitive given the aggressive clocking and power budgets of most NoC applications.so based on distributed shared buffer efficient pipelining and novel flow control is achieved. In XY routing algorithm, a packet must always be routed along horizontal or X axis until it reaches the same column as that of destination. Then it should be routed along vertical or Y axis and towards the location of destination resource.

The routing algorithm, which defines the path taken by a packet between the source and the destination, is a main task in network layer design of NoC. According to where routing decisions are taken, it is possible to classify the routing as source and distributed routing [6]. Routing algorithm can be classified on the basis of adaptivity, such as deterministic or adaptive. In deterministic routing, the path from source to destination is completely determined in advance by the source and destination address. Examples are XY routing. In adaptive routing, multiple paths from source to destinations are possible [7]. Analysis of 2-Dimensional Mesh routing algorithms which restrict certain paths for communication with deadlock restrictions. Examples are Odd even routing. They are simple and easy to implement compared to adaptive routing algorithm. The routing algorithm that uses shortest path for communication is called minimal routing. The routing algorithm which uses longer paths for communication though shorter paths exist is known as non-minimal routing. Non-minimal routing has some advantages over minimal routing including possibility of balancing network load and fault tolerance. In static routing, the path cannot be changed after a packet leaves the source. In dynamic routing, a path can be altered anytime depending upon the network conditions. Routing algorithms can also be defined based on their implementation: lookup table and Finite State Machine (FSM).

IV XY ROUTING ALGORITHM

Wang Zhang and Ligang Hou proposed Classic XY(Static XY OR XY) routing algorithm which is one kind of distributed deterministic routing algorithms .XY routing is one of the type of Dimension order routing (DOR) which is a typically a minimal turn algorithm and is more suitable for networks using mesh or torus topology. XY routing algorithm routes packets first in x-direction (or horizontal direction) to the correct column and then in y- direction (or vertical direction) to the receiver. In XY routing the addresses of the routers are their xy-coordinates. One of the advantages of XY routing is that it never runs into deadlock or livelock [8]. XY routing from router A to router B in 4x4 NOC .However despite the advantages of simplified calculation and removal of deadlock or livelock problem, there are some problems in the traditional XY routing. One important issue with traditional XY routing algorithm is that the algorithm causes the biggest load in the middle of the network which does not extend the traffic regularly over the whole network. Thus there is a need for algorithms which equalize the traffic load over the whole network and are yet simple and efficient.

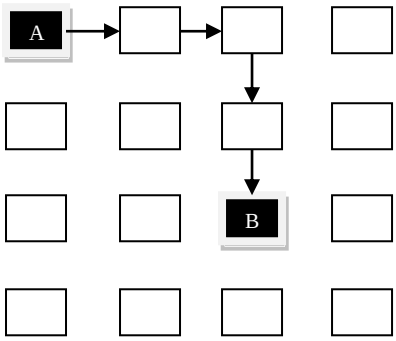

Fig 1. Routing from source to destination

The XY routing algorithm is one kind of distributed deterministic routing algorithm. XY routing never runs into deadlock or livelock [8]. For a 2-Dimesion mesh topology NoC, each router can be identified by its coordinate (x, y) The XY routing algorithm compares the current router address (Cx,Cy) to the destination router address (Dx,Dy) of the packet, stored in the header flit [9]. Flits must be routed to the core port of the router when the (Cx, Cy) address of the current router is equal to the (Dx, Dy) address.

If this is not the case, the Dx address is firstly compared to the Cx (horizontal) address. Flits will be routed to the East port when Cx<Dx, to West when Cx>Dx and if Cx=Dx the header flit is already horizontally aligned. If this last condition is true, the Dy (vertical) address is compared to the Cy address. Flits will be routed to South when Cy<Dy, to North when Cy>Dy. If the chosen port is busy, the header flit as well as all subsequent flits of this packet will be blocked. The routing request for this packet will remain active until a connection is established in some future execution of the procedure in this router.

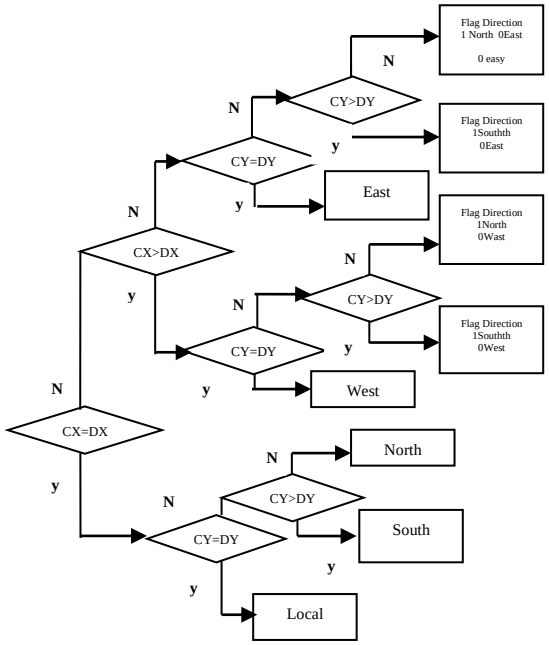

FIG 2: XY Routing Algorithm

The following text is the XY routing algorithm:/* XY routing Algorithm */ /*Source router: (Sx, Sy);destination router: (Dx,Dy); current router: (Cx,Cy).*/

```
begin
if (Dx>Cx) //eastbound messages
return E;
else
if (Dx<Cx) //westbound messages
return W;
else
if (Dx=Cx) {
//currently in the same column as
//destination
if (Dy<Cy) //southbound messages
return S;
else
if (Dy>Cy) //northbound messages
return N;
else
if (Dy=Cy) //current router is the destination router return C;
}
end
```

The implementation of XY routing algorithm is simple. However, it is deterministic routing algorithm, which means this routing algorithm only provides a routing path for a pair of source and destination. Moreover, XY routing algorithm cannot avoid from deadlock appearance.

V WISHBONE

In the Wishbone specification it is said that the main idea with Wishbone is that it offers IP cores a common and flexible interface for communication. This is said to lead to faster integration of different IP cores, since it would lead to a plug-

and-play manner when connecting IP cores. The goal with getting a plug-and-play manner is to be able to remove the time needed to implement and test new IP cores and instead use old and already tested IP cores to create new on-chip systems. Another reason to use Wishbone is that it offers a variety of bus transfer cycles for IP cores to use. As stated in Wishbone is only _intended as a general purpose interface that defines the standard data exchange between IP cores. It is not intended to interfere with application specific functions of the IP cores. In other words this means that Wishbone only bothers about how data is transferred between IP cores and not what the data does mean for the different IP cores. The final reason to use Wishbone is that it is an open interconnection architecture, which means that it is free to use for any developer without having to pay a fee. This is hopefully thought to lead to that many different IP Core suppliers choose to use Wishbone and thereby drive down prices and improving the quality of IP cores supporting it, as the organization behind Wishbone formulates it in.

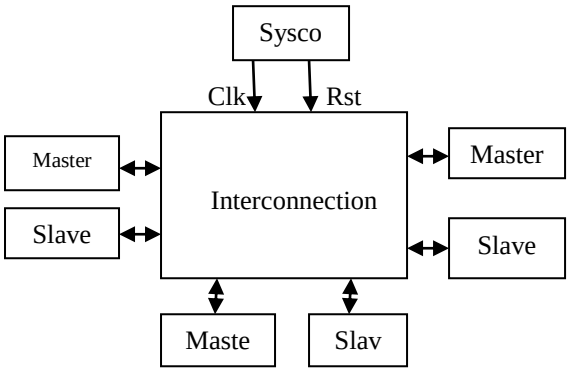

FIG 3: WISHBONE INTERCONNECTION

Wishbone utilizes master and slave architectures which are connected to each other through an interface called Intercon. Master is an IP core that initiates the data transaction to the slave IP core. Master starts transaction providing an address and control signal to slave. Slave in turn responds to the data transaction with the master with the specified address range. . Intercon is a module that interconnects wishbone MASTER and SLAVE interfaces. Syscon is a wishbone module that drives the system clock [clk_o] and reset [rst_o] signals. The WISHBONE interconnection itself is nothing more than a large, synchronous circuit. It must be designed to logically operate over a nearly infinite frequency range. However, every integrated circuit has physical characteristics that limit the maximum frequency of the circuit. Transmitter (Master) Receiver (Slave) Wishbone Bus Interface the WISHBONE interconnection itself is nothing more than a large, synchronous circuit. I will be designing it to logically operate over a nearly infinite frequency range. However, every integrated circuit has physical characteristics that limit the maximum frequency of the circuit. This means that a WISHBONE compatible circuit will theoretically function normally at any speed, but that its maximum speed will always be limited by the process technology of the integrated circuit. It can interconnect between soft-core, solid-core and hard-core. Wishbone bus structure is relatively simple, it just defines a high-speed bus and offers four different IP interconnect structure namely point to point, dataflow, shared bus and cross bar switch architectures.

VI EVALUTION

The system is evaluated by designing Verilog HDL and simulated using modelsim and synplified on FPGA. To verify Routing Algorithm, we first use the Modelsim which is a very famous commercial simulation tool in the electronics industry. Make clear that the data packet can be transmitted to their destination output-port based on the destination addresses which it takes.

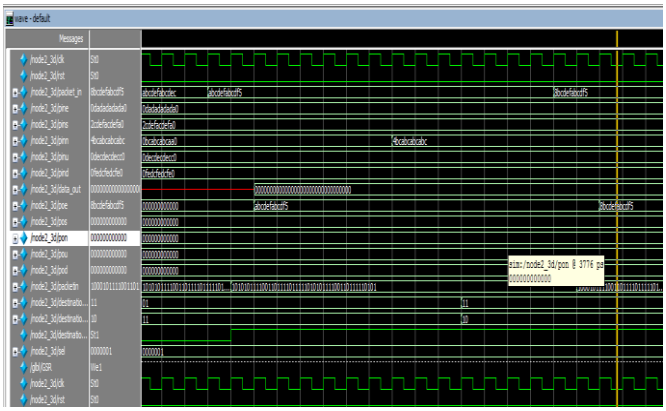

FIG 4: SIMULATED WAVEFORM

Here in this paper considering for 4*4 Mesh topology results are analyzed. Depending on the usage of slices on FPGA respective module have been created for slice usage which permits to know the results of evaluated structure.

VII CONCLUSION

When compared to other system architectures, our paper has a reduced latency, increased throughput and area overhead .Wishbone protocol is designed in such a way that it is flexible and easily compatible with other architectures .Designing of wishbone protocol is further in process.

REFERANCES

[1] W. Dally and B. Towles, “Principles and Practices of Interconnection Networks”, Morgan Kaufmann, 2004.
[2] W. J. Dally and B. Towles, “Route packets, not wires: On chip interconnection networks,” pp. 684–689. In Design Automation Conference, 2001,
[3] K. Goossens, J. Dielissen, and A. Radulescu, “Aethereal network on chip: concepts, architectures, and Implementations,” vol. 22, Design & Test of Computers, IEEE.
[4] D. Bertozzi and L. Benini, “Xpipes: a network-on-chip architecture for giga scale systems-on-chip,” vol. 4, Circuits and Systems Magazine, IEEE, 2004.
[5] A. Ehliar and D. Liu, "An FPGA based open source network-on-chip architecture", 17th International Conference on Field Programmable Logic and Applications, FPL, pp.800-.803, IEEE Amsterdam, Holland, 2007.
[6] N. Kapre, N. Mehta, M. deLorimier, R. Rubin, H. Barnor, M. J.Wilson, M. Wrighton, and A. DeHon, “Packet switched vs. time multiplexed FPGA

overlay networks," Symposium on Field programmable Custom Computing Machines, IEEE 2006.
[7] C. Hilton and B. Nelson, "PNoC: a flexible circuit switched NoC for fpga-based systems," vol. 153, Computers and Digital Techniques, IEEE Proceedings- 2006.
[8] G. Schelle and D. Grunwald, "On chip interconnect exploration for multicore processors utilizing FPGA's," 2nd Workshop on Architecture Research using FPGA Platforms, 2006.
[9] "NoCem – network on chip emulator," http://www.opencores.com/projects.cgi/web/nocem/overview
[10] T. Mak, P. Sedcole, P. Y. Cheung, and. Luk, "On- FPGA communication architectures and design factors," 16th International Conference on Field Programmable Logic and Applications, 2006.
[11] D. Wiklund and D. Liu, "SoCBUS: switched network on chip for hard real time embedded systems," Parallel and Distributed Processing Symposium. Proceedings. International, 2003.
[12] "Wishbone system-on-chip (SoC) interconnection architecture for portable IP cores," http://www.opencores.org/, 2002.