

Design of Router Architecture Based on Wormhole Switching Mode for NoC

L.Rooban, S.Dhananjeyan

Abstract - Network on Chip (NoC) is an approach to designing communication subsystem between intelligent property (IP) cores in a system on chip (SoC). Packet switched networks are being proposed as a global communication architecture for future system-on-chip (SoC) designs. In this project, we propose a design and implement a wormhole router supporting multicast for Network-on-chip. Wormhole routing is a network flow control mechanism which decomposes a packet into smaller flits and delivers the flits in a pipelined fashion. It has good performance and small buffering requirements. The implementations are at the RT level using VHDL and they are synthesizable. First, based on virtual cut through router model, a unicast router is implemented and validated and based on the wormhole switching mode the multicast router architecture is designed and implemented. A Wormhole input queued 2-D mesh router is created to verify the capability of our router.

Index Terms - Network on chip, router architecture, wormhole switching

1 INTRODUCTION

More and more processor cores and large reusable components have been integrated on a single silicon die, which has become known under the label System-on-Chip (SoC). Buses and point-to-point connections were the main means to connect the components. But as silicon technology advances further, problems related to buses have appeared. 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. So researches in systematic approaches to the design of the communication part of SoC are needed at all levels from the physical to architectural to the operating system and the application level. So Network-on-Chip (NoC) is used mostly in a very broad meaning, encompassing the hardware communication infrastructure, the middleware, operating system communication services, the design methodology and tools to map application onto a NoC. All these together can be called a NoC platform. Platform based design methods accelerate time-to-market through extensive reuse of an architectural platform. Such design methods decouple computation from communication concerns, simplify problems. A packet switched network which delivers messages between communicating components has been proposed as the solution for SoC design. NoC is a new paradigm for integrating a large number of IP's cores for implementing a SoC. In NoC paradigm a router-based network is used for packet switched communication among on chip cores. Since the communication infrastructure as well as the cores from one design can be easily reused for a new product, NoC provides high possibility for reusability. Networks are composed of routers, links between routers, and Network Interfaces (NI), which implement the interface to the IP modules. We use the wormhole switching mode is a variant of the virtual cut

Through mode that avoids the need for large buffer spaces. A packet is transmitted between switches in units called flits (flow control digits the smallest unit of flow control). Only the header flit has the routing information. Thus, the rest of the flits that compose a packet must follow the same path reserved for the header. We use a deterministic routing; the path is uniquely defined by the source and target addresses. This paper describes a new set of router components that can be used to form different routers with a varying number of ports, routing algorithm, data widths and buffer depths. Router architecture with low latency, high speed and high maximal peak performance is designed using wormhole switching mode which is used as a variant of the virtual cut through and comparisons between different routing structures is performed and hardware implementation of wormhole router architectures with 4x4 nodes is proposed.

2 RELATED WORKS

The large number of processor cores in chip multi-processors, 2D mesh has been gaining wide acceptance for inter core on chip communication. Program performance is more sensitive to the router latency than to the link bandwidth. Adaptive System-on-a-Chip (aSoC) is used as a backbone for power-aware video processing cores. Adaptive System-on-a-Chip, by nature of its statically scheduled mesh interconnect, performs up to 5 times faster than bus-based architectures. Additionally interconnect usage for typical digital signal processing applications is fewer than 20%. This leaves significant interconnect bandwidth to accommodate the control communications required by the power-aware features of modern cores. ASoC's ability to provide dynamic voltage and frequency scaling is critical to future portable digital signal processing applications. This will allow SoC implementations to exploit the inevitable mismatches in core utilization, due to data content variations or user requirements, to reduce power consumption. Communication between nodes takes place via pipelined, point-to-point connections. The XY algorithms are used to route packet from source node to a destination node. The

entire flow leverages the flexibility of a fully reusable and scalable network components library called Xpipes. The latency of its router is 7 cycles. The disadvantage of this NoC is that the reception of the packets is not guaranteed when flits take different paths. The ANoC communication architecture is composed of nodes links between nodes and computations resources. The global topology of the architecture is not determined. HERMES is a 2D-Mesh NoC topology satisfies the requirement of implementing a low area and low latency communication for system on chip modules. The flit size is parametrable and the XY algorithm is used for routing the packet from source node to a destination node. The width of link is parametrable (16 bits data + 2 bits control for prototype) and it uses a deterministic algorithm with virtual cut through switching modes. The output queue is the used buffering techniques and the interface with the IP core is custom.

2.1 Virtual Cut through Switching Mode

In computer networking, cut-through switching is a method for packet switching systems, wherein the switch starts forwarding a frame (or packet) before the whole frame has been received, normally as soon as the destination address is processed. This technique reduces latency through the switch, but decreases reliability. Switches do not necessarily have cut-through and store-and-forward "modes" of operation. As stated earlier, cut-through switches usually receive a predetermined number of bytes, depending on the type of packet coming in, before making a forwarding decision. The switch does not move from one mode to the other as dictated by configuration, speed differential, congestion, or any other condition.

Virtual cut-through switching is the most sophisticated and expensive technique where,

1. Messages are split into packets and router has buffers for the whole packet as in SF switching.
2. Instead of waiting for the whole packet buffered, the incoming header flit is cut through into the next router as soon as the routing decision was made and the output channel is free (see Figure (b)).
3. Every further flit is buffered whenever it reaches the router, but it is also immediately cut-through to the next router if the output channel is free (see Figure (c)).
4. In case the header cannot proceed, it waits in the current router and all the following flits subsequently draw in, possibly releasing the channels occupied so far (see Figure (d)).
5. In case of no resource conflicts along the route, the packet is effectively pipelined through successive routers as a loose chain of flits (see Figure (e)). All the buffers along the routing path are blocked for other communication requirements.

A packet of length M transmitted to distance d takes time $t_{VCT} = d(t_r + t_w + t_m) + \max(t_w, t_m)M$. We assume that there is no time penalty for cutting through. Only the header

experiences routing delay as well as switching and inter-router latency. Once the header flit reaches the destination, the cycle time of the pipeline of packet flits is determined by the maximum of the switching time and inter-router latency. This is because we assume that channels have both input and output buffers. In case of input buffering only, for example, we would have $t_w + t_m$ instead of $\max(t_w, t_m)$. Only the header flit contains routing information and therefore each incoming data flit is simply forwarded along the same output channel as its predecessor. Therefore, transmission of different packets cannot be interleaved or multiplexed over one physical channel.

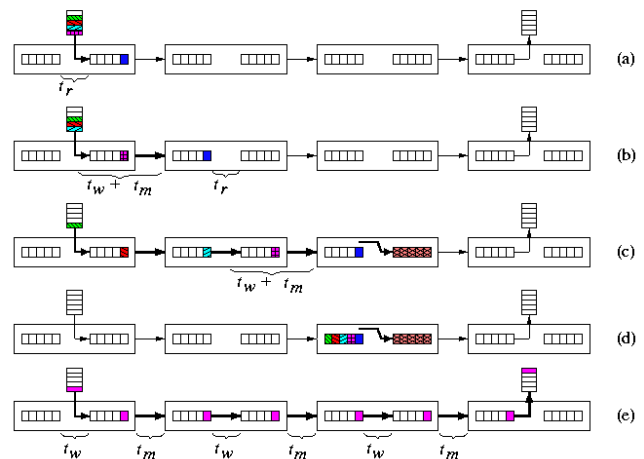


Figure 2.1: Virtual cut-through switching of a packet. (a) The header flit has moved into the output buffer of the first router. (b) The header has cut through into the second router while subsequent flits are following its path. (c) Pulling a chain of flits behind, the header has cut through into a router where its output buffer is reserved for another Communication. (d) The whole chain of packet flits has contracted and the whole packet gets buffered in the first router releasing all previously allocated links. (e) Flit pipeline moving towards the destination.

3 ROUTER ARCHITECTURE

The design of router consists of the following parts: the input controllers, routing functions, crossbar, credit switcher and global switch arbiter. The flow of messages across the physical channel between adjacent routers is implemented by the input controller. We use in each input port of our router the module called input controller which is divided into three sub modules called respectively: link controller, FIFO and output controller. The link controller has the role of receiving a packet flits from the Output port of the adjacent router and stores them in FIFO buffer and manages the flow control (credit based) between adjacent router. It allows the pipelining between packets. It can accept a new packet P when the packet $P-1$ is not totally routed. The output controllers have the role of reading flits of packet from input FIFO and forward him when the output port requested by

the routing function is granted by the global switch arbiter. The routing function implements the routing algorithm, which determines the output port that should be taken by the header and the remaining flits of the packet. The computing of the output port is based on destination address filed from the header of packet. The global Switch Arbiter which contains 5 arbiters receives requests from the routing function of each input port and assigns available output ports to the requestors. Arbiters used in this work 13 employ the round Robin algorithm. The crossbar switch directly connects the 5 inputs port to 5 outputs port of router with no intermediate stages. In effect, such a switch contains 5 multiplexers 4:1, one for each output port. The crossbar is controlled by the global switch arbiter module and it contains 5 Mux. The credit switcher module allows linking of credit signals of the neighbor routers to current router.

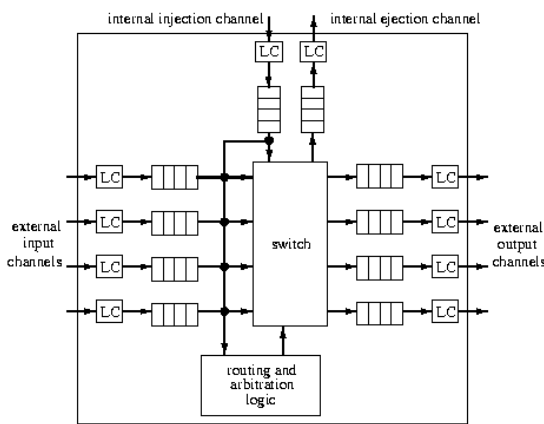


Figure 3.1: Router Architecture

3.1 Routing function

The main role of routing function is to determine from the destination address field of the header flit the path that the packet must follow. Each routing function sends 4 requests to the global switch arbiter. If the output port has denied access, the routing function must maintain its requests. The Switch Allocator manages access between input controllers and the output ports by asserting correspondent grants signals. Our routing module use deterministic routing algorithm which provides a minimal path between any two nodes called XY. In this type of routing algorithms, the decision is independent of the state of the network. According to the wormhole, the router can contain only information of one packet. In this fact, the input port cannot receive a new packet. To avoid this problem, our routing function can accept a new packet to be buffered in the input controller and the output port of the new packet is computed and its requests are temporary memorized until the current packet will be totally routed. From communication point of view, input controllers manages the data flow of input ports which it is dependent and it establishes the communication with the other neighboring blocks. It is mainly composed of three modules called respectively: link controller, FIFO and output controller. Each router in our case contains five input controller to the maximum. One input controller for each

input port. The input signals of this module are: 4 grant signals provided by the global switch arbiter. The Link Controller FSM starts when BOP and req are asserted and the current credit is in high level. Its tasks are summarized in two principal tasks: - To receive packets sent by the output controller of the sender and to write packets received in FIFO. We present the interface of this component and represent the various interexchange signals. The output Controller is like a bridge that connects the FIFO and the output port destination. Its tasks are to read data from the FIFO, detect if EOP is in high level or not in each data and to determine the last flit of each packet. In fact, The EOP signal is asserted in output pin when it was detected and it is also connected to the RF to indicate the end of the routing. In reality the EOP is also asserted when one of all credit inputs and when one of all grants inputs are activated. The global Switch Arbiter which contains 5 arbiters receives requests from the routing function of each input port and assigns available output ports to the requestors, resolving contention. Credit switcher is to connect the credit signals provided by the input controllers of neighboring routers to current router.

3.2 Wormhole switching mode

Nodes in a direct network communicate by passing messages from one node to another. A message enters the network from a source node and is routed towards its destination through a series of intermediate nodes. Four types of switching techniques are usually used for this purpose: circuit switching, packet switching, virtual cut-through switching, and wormhole switching. In circuit switching, a dedicated path is established between the source and the destination before the data transfer initiates. The message is never blocked during transfer. In order to improve performance, packet switching is used. In packet switching, a message is divided into packets that are independently routed towards their destination. The destination address is encoded in the header of each packet. The entire packet is stored at every intermediate node and then forwarded to the next node in its path. In order to reduce the time to store the packets at each node, virtual cut-through switching is introduced. In this technique, a message is stored at an intermediate node only if the next channel required is occupied by another packet.



Figure 3.2: Wormhole switching

Wormhole switching is a variant of the virtual cut-through technique that avoids the need for large buffers for saving messages. In wormhole switching, a packet is transmitted between the nodes in units of flits, the smallest units of a message on which flow control can be performed. The head flit of a message contains all the necessary routing

information and all the others flit contain the data elements. The flits of the message are transmitted through the network in a pipelined fashion. It is shown in figure 3.2. The main advantage of wormhole switching derives from the pipelined message flow, since transmission latency is insensitive to the distance between the source and the destination. Moreover, since the message moves flit by flit across the network, each node needs to store only one flit. The reduction of buffer requirements at each node has a major effect on the cost and the size of systems. The main disadvantage of wormhole switching comes from the fact that only the head flit has the routing information. If the head flit cannot advance in the network due to resource allocation, all the trailing flits will be blocked along the path and these blocked messages can block other messages. This reduces network performance drastically and this chained blocking can also lead to deadlock, as shown in figure 3.3. In this situation, each message wants to turn left, but the buffer that it wants to use is occupied by another one. Messages wait for each other in a cycle and hence no message can advance any further. Prevention of deadlock in wormhole switching is usually accomplished by a suitable choice of routing algorithm that selectively prohibits the message from taking all the variable paths, thus preventing cycles in the network.

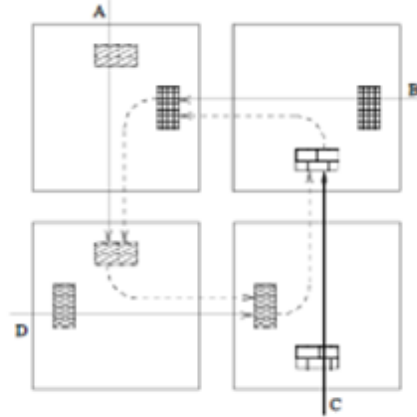


Figure 3.3: Deadlock

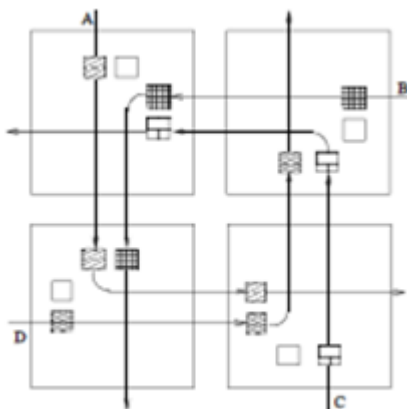


Figure 3.4: Virtual Channels

Virtual channels (VCs) can be used in wormhole-switched networks to prevent deadlock and reduce effects of channel blocking. A virtual channel is a logical abstraction of a physical lane. All the virtual channels associated with a physical channel have individual flit buffers and are time-multiplexed for message transmission using the physical channel. An example is shown in figure 3.4.

3.3 Unicast router

It has p physical channels (PCs) and v lanes per PC. A packet passes the router through four stages: Routing, lane allocation, flit scheduling and arbitration. Consider that a packet consists of three flits, one head flit, one body flit and one tail flit. When the head flit arrives, according to its flit type field, the lane sends out the routing request and enters routing stage. Then the routing logic calculates the output path by its destination field according to routing algorithm. After receiving the grant from the routing logic, the lane sends out the lane allocation request and goes to lane allocation stage. The lane allocator finds out an available lane on its output path in the next hop and associates that lane to the lane in this hop together, so that other packets cannot use them until this association is released. If the lane allocation succeeds, the lane will receive a grant from the lane allocator, send out an arbitration request, go to the arbitration stage and wait for the transfer; otherwise the lane will stay in the allocation stage. The arbitration includes two levels. The first level is about the lane sharing the same physical channel to the crossbar. The second level is for the crossbar traversal. If one lane wins all the two levels of the arbitration, the lane will receive a grant from the arbitration. Then the flit in this lane is transmitted through the crossbar which is controlled by the arbitration. Hence the data or trail flit arrives, according to its flit type field; the lane enters the flit scheduling stage. If there is a buffer available in the next hop lane associated by the head flit, a grant will be sent out from the flit scheduler to the lane and the lane will enter the arbitration stage and perform the same as the head flit, otherwise it will stay here till success. When the tail flit leaves the router, the router will release the lane association. Then the head flits of other packet can use the lane. The flits from different packets will not be interleaved in a lane since the lane is kept being associated to the previous lane while transmitting the whole packet. In the admission part, there are a lot of lanes used for admitting packets into the network. Their length is the maximum number of flits that a packet can contain. If an uploading lane is available, a packet will be split into smaller flits and the flits are put into one uploading lane in order. Then the packet will also pass the router through four stages. Routing, lane allocation, flit scheduling, and arbitration, the same as the lane at the input channels (channels are connected to outputs of other routers). The model in which the packet in one admission lane can be routed to any output channel is called ideal admission model or decoupled admission model which was shown in the figure. If the packet in one admission lane can only be routed to one specific output channel, the model is called

coupled admission model. This scheme alleviates the complexity of the router, especially in the crossbar, and achieves approximately the same performance as the decoupled admission model.

4 SIMULATION RESULTS

In the implementation of one to one unicast router architecture 4x4 nodes are used, in which the values or packets are transmitted from one node to the adjacent node in any manner they are connected to the other node. Once the clock pulse are received the transmission of packets from one node to the adjacent node are received uniformly making use of buffer in the routing path. Using the Modelsim software it is simulated to provide the output.

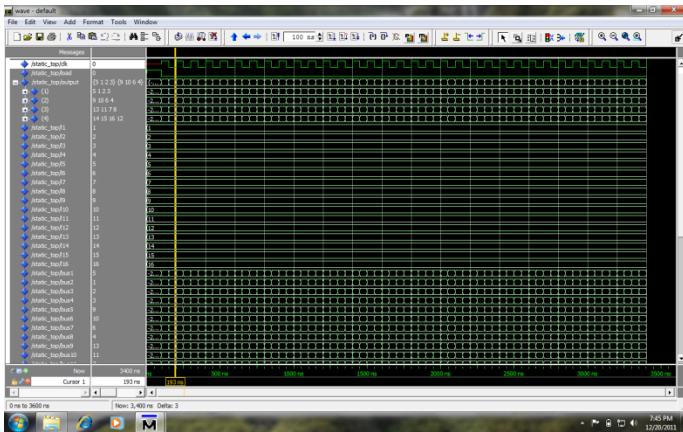
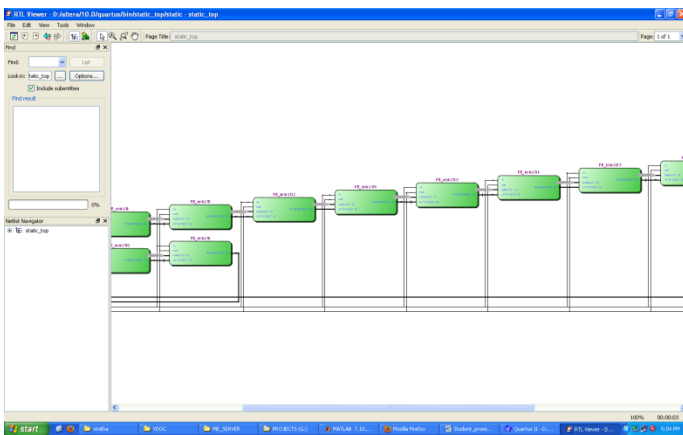


Figure 4.1: One to one unicast router output

4.1 RTL schematic

The one to one unicast router architecture with 4x4 nodes where the transfer of packets can be viewed in the RTL schematic as shown below.



5 CONCLUSION

The one to one unicast router architecture making use of virtual cut through switching mode was designed and the outputs were verified using simulation results. The desired wormhole switching mode multicast router architecture would be designed and implemented making use of FPGA in later stages, and the performances of both the switching modes output is compared.

REFERENCES

- [1] A Modular Router Architecture Design for network on chip - a conference paper on system signals& devices
- [2] Adaptive system on a chip (ASOC): A backbone for power-aware signal processing cores. Andrew Laffely, Jian Liang, Russell Tessier, Wayne Bureson.
- [3] W. J. Dally, Virtual-channel flow control, inProc. 17th Annu. Int. Symp. Comput. Architecture, May 1990.
- [4] S. Felperin, P. Raghavan, and E. Upfal, A theory of wormhole Routing, Proceeding IEEE Transaction on Computer, June 1996, Vol. 45, no. 6, Pages: 704-713.
- [5] Fundamentals of digital logic with VHDL design, Stephen Brown, McGraw-Hill Higher Education, 2004.
- [6] VHDL programming by example, Douglas L. Perry, McGraw-Hill, 2002.
- [7] F. Moraes, N. Calazars, A. Mello, L. M. oller, and L. Ost, HERMES: an infrastructure for low area overhead packet switching networks on chip, INTEGRATION, the VLSI journal 2004. VOL.38,Pages: 6993.

AUTHORS BIOGRAPHY



L. Rooban received his B.E (Electronics and Communication) Degree from Yellamma Dasappa Institute of Technology, Bangalore - Affiliated to Visvesvaraya Technological University, Belgaum. Currently he is pursuing M.E Degree in Vel Tech Multi Tech Dr.RR Dr.SR Engineering College, Chennai, affiliated to Anna University. His Area of interest includes Embedded System and VLSI Design.



S.Dhananjeyan received his B.Tech (Electrical and Electronics) from Sri Manakula Vinayagar Engineering College, Pondicherry and M.E degree from Vel Tech Engineering College, Chennai. He is currently working as an Assistant Professor in Electrical and

Electronics Engineering at Vel Tech Multi Tech Dr.RR Dr.SR Engineering College, affiliated to Anna University, Chennai. He has 2 years of teaching experience in Engineering College. He has been teaching the subjects: Solid State Drives Electromagnetic Field Theory, Electrical Machines, and Micro Controller. His area of interest includes Electrical Machines, Power Electronics, and Embedded Control System .